

อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกล  
ด้วยสัญญาณคลื่นไฟฟ้าจากสมอง  
DEEP LEARNING ALGORITHM FOR ROBOT ARM CONTROL  
USING EEG SIGNAL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกล  
ด้วยสัญญาณคลื่นไฟฟ้าจากสมอง  
DEEP LEARNING ALGORITHM FOR ROBOT ARM CONTROL  
USING EEG SIGNAL



นายกรธรรม ศรีธนธรรม 61010022  
นางสาวณัฐชยา ศรีคำ 61010323

อาจารย์ที่ปรึกษา  
ผศ. ดร.ตุลยา ลิมปิติ  
รศ. ดร.จิรสุดา โกษียาภรณ์

ปริญญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกลด้วยสัญญาณคลื่นไฟฟ้าจากสมอง

DEEP LEARNING ALGORITHM FOR ROBOT ARM CONTROL USING EEG SIGNAL

ผู้จัดทำ

- |                 |           |          |
|-----------------|-----------|----------|
| 1. นายกรธรรม    | ศรีชนธรรม | 61010022 |
| 2. นางสาวณัฐชยา | ศรีคำ     | 61010323 |

*Tulaya Limpi*  
..... อาจารย์ที่ปรึกษา  
( ผศ. ดร.ตุลยา ลิมปิติ )

*Kesong*  
..... อาจารย์ที่ปรึกษาร่วม  
( รศ. ดร.จิรัสตา โกษิยาวรณ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาานิพนธ์เรื่อง “อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกลด้วยสัญญาณคลื่นไฟฟ้าจากสมอง” จะไม่สามารถสำเร็จลุล่วงไปได้ หากไม่ได้รับความอนุเคราะห์อย่างยิ่งจากอาจารย์ที่ปรึกษาปริญญาานิพนธ์ คือ ผศ. ดร.ตุลยา ลิมปิติ, รศ. ดร.จิรสุดา โกษียาภรณ์ และ ผศ. ดร.ณัฐกานต์ พุทธิรักษ์ ที่กรุณาให้คำปรึกษา และแนวทางการแก้ไขปัญหาตลอดระยะเวลาในการจัดทำปริญญาานิพนธ์ รวมทั้งสนับสนุนสถานที่ เครื่องมือ และอุปกรณ์ต่างๆ ที่จำเป็นต้องใช้ในระหว่างการจัดทำปริญญาานิพนธ์ ขอขอบพระคุณในความห่วงใยและความหวังดีที่มีให้แก่คณะผู้จัดทำเป็นอย่างยิ่ง

ขอขอบคุณอาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอนและประสิทธิ์ประสาทวิชาความรู้ให้แก่คณะผู้จัดทำ

ขอขอบคุณผู้มีส่วนเกี่ยวข้องทุกท่าน อาทิ บิดา มารดา และเพื่อนนักศึกษา ที่คอยสนับสนุน แนะนำช่วยเหลือ และให้กำลังใจแก่คณะผู้จัดทำเสมอมา จนกระทั่งปริญญาานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

นายกรรธรรม ศรีธนธรรม

นางสาวณัฐชยา ศรีคำ

ผู้จัดทำ

อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกล  
ด้วยสัญญาณคลื่นไฟฟ้าจากสมอง  
DEEP LEARNING ALGORITHM FOR ROBOT ARM CONTROL  
USING EEG SIGNAL

โดย นายกรธรรม ศรีธรรม 61010022  
นางสาวณัฐชยา ศรีคำ 61010323

อาจารย์ที่ปรึกษา ผศ. ดร.ตุลยา ลิ้มปิติ  
อาจารย์ที่ปรึกษาร่วม รศ. ดร.จิรสุดา โกษียาภรณ์

#### บทคัดย่อ

อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกลด้วยสัญญาณคลื่นไฟฟ้าจากสมองเป็นหนึ่งในงานที่ได้รับความนิยมในด้าน Brain-computer-interface (BCI) เพื่อช่วยเหลือผู้สูงอายุหรือผู้ป่วยที่ไม่สามารถขยับร่างกายได้ด้วยตัวเอง โดยอัลกอริธึมการเรียนรู้เชิงลึกนั้นมีโครงสร้างที่สามารถนำมาใช้ได้มากกว่าหนึ่งโครงสร้าง จุดประสงค์ของปริญญาโทครั้งนี้จึงต้องการที่จะค้นคว้าและพัฒนาอัลกอริธึมเพื่อประมวลผลสัญญาณคลื่นไฟฟ้าจากสมอง โดยศึกษาการเตรียมข้อมูล การแยกคุณลักษณะสัญญาณ และวิเคราะห์ประสิทธิภาพของโครงสร้างอัลกอริธึมการเรียนรู้เชิงลึกแบบต่างๆ เพื่อพัฒนาโมเดลการเรียนรู้เชิงลึกที่เหมาะสมในการนำไปใช้ควบคุมแขนกลจำลองต่อไป

#### ABSTRACT

Deep learning algorithm for robot arm control using electroencephalographic (EEG) signal is one of the popular applications in the field of Brain-Computer Interface (BCI) to help elderly or patients who have trouble moving on their own. Various deep learning structures have been proposed. The objective of this thesis is to research and develop an algorithm for processing EEG signals. After pre-processing, the extracted features of the EEG signal are used to train and test different deep learning structures. The best algorithm is used to control the simulated robot arm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	XVI
<b>บทที่ 1</b>	<b>บทนำ</b>
	1.1 ความเป็นมาและความสำคัญของปัญหา
	1.2 วัตถุประสงค์
	1.3 ขอบเขตของปริญญาานิพนธ์
<b>บทที่ 2</b>	<b>ทฤษฎีและหลักการที่เกี่ยวข้อง</b>
	2.1 คลื่นไฟฟ้าจากสมอง (EEG Signal)
	2.2 Filter Bank Common Spatial Pattern (FBCSP)
	2.3 การแปลงเวฟเลตแบบต่อเนื่อง (Continuous Wavelet Transform)
	2.4 การแปลงเวฟเลตแบบดิสครีต (Discrete Wavelet Transform)
	2.5 อัลกอริธึมการเรียนรู้เชิงลึก (Deep Learning Algorithm)
	2.6 เฟรมเวิร์คสำหรับการเรียนรู้เชิงลึก (Deep Learning Framework)
	2.7 อุปกรณ์ Unicorn Hybrid Black
	2.8 Lab Streaming Layer (LSL)
	2.9 OpenViBE
	2.10 CoppeliaSim
	2.11 แขนกลจำลอง
<b>บทที่ 3</b>	<b>การออกแบบและการจัดทำปริญญาานิพนธ์</b>
	3.1 การออกแบบ
	3.2 เครื่องมือที่ใช้ในการทดลอง
	3.3 การจัดเก็บผลการทดลอง

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 4 ผลการทดลอง</b>	
4.1 การทดสอบการเตรียมข้อมูล	90
4.2 การทดสอบอัลกอริธึม Common Spatial Pattern กับสัญญาณ EEG	97
4.3 การทดสอบการแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG	104
4.4 การทดสอบการแปลงเวฟเลตแบบต่อเนื่องของสัญญาณ EEG	112
4.5 การทดสอบการรวมคุณลักษณะ	113
4.6 ระยะเวลาที่ใช้ในประมวลผลการแยกคุณลักษณะและการจำแนกด้วยการเรียนรู้เชิงลึก	124
4.7 การทดสอบการจำแนกคลาสด้วยอัลกอริธึมการเรียนรู้เชิงลึกด้วยชุดข้อมูลชุดทดสอบจาก BCI Competition IV 2a	125
4.8 การทดสอบการจำแนกคลาสด้วยอัลกอริธึมการเรียนรู้เชิงลึกด้วยชุดข้อมูลคลื่นไฟฟ้าจากสมองของผู้จัดทำ	130
4.9 การทดสอบระบบ Offline System Testing สำหรับข้อมูล BCI Competition IV 2a	131
4.10 การทดสอบการทำงานของระบบโดยรวมแบบ Real-Time System Testing	134
<b>บทที่ 5 สรุปผลและข้อเสนอแนะ</b>	
5.1 สรุปผล	137
5.2 ข้อเสนอแนะ	137
<b>บรรณานุกรม</b>	<b>138</b>

## สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบของเซลล์ประสาท	2
2.2 เครื่อง Electroencephalogram	3
2.3 ช่วงความถี่ของสัญญาณคลื่นไฟฟ้าจากสมอง	3
2.4 ตำแหน่งของอิเล็กโทรดที่ใช้วัดสัญญาณ EEG	5
2.5 ตัวอย่าง ERD/ERS ใน Mu-band (a) อิเล็กโทรด C3 (b) อิเล็กโทรด Cz (c) อิเล็กโทรด C4	5
2.6 ตัวอย่างสถาปัตยกรรมของอัลกอริธึม Filter Bank Common Spatial Pattern (FBCSP) สำหรับขั้นตอนการฝึกอบรมและการประเมิน	6
2.7 ตัวอย่างของการแจกแจงข้อมูล 2 คลาส (a) ก่อนกรอง และ (b) หลังกรอง ด้วย CSP ใน 2 มิติ	8
2.8 คุณสมบัติของเวฟเลตแบบต่อเนื่องใน MATLAB	9
2.9 การแบ่งกลุ่มย่อยของสัญญาณโดยใช้ DWT	10
2.10 เวฟเลตแบบดิสครีตประเภทต่างๆ	11
2.11 เวฟเลต Db4 Filter Bank ในมิติของความถี่	11
2.12 เวฟเลต Db4 Filter Bank ในมิติของเวลา	12
2.13 ตัวอย่างตัวกรองสำหรับแยกสัญญาณของเวฟเลต Db4 (a) ตัวกรองความถี่ต่ำ (b) ตัวกรองความถี่สูง	13
2.14 สถาปัตยกรรมของ MultiLayer Perceptron	14
2.15 สถาปัตยกรรมของ LeNet-5	15
2.16 สถาปัตยกรรมของ ResNet18	16
2.17 สถาปัตยกรรมของโครงข่ายประสาทเทียมเชิงลึกแบบผสม	18
2.18 โครงสร้างของโมเดล CNN_1	19
2.19 โครงสร้างของโมเดล CNN_2 ซึ่งประกอบด้วย 3 Convolution Layer และ 3 Max Pooling Layer	19
2.20 เฟอร์มิเว็คของเซลล์ LSTM โดย $C_t$ แสดงถึงสถานะของเซลล์ LSTM ที่โมเมนต์ปัจจุบัน	20
2.21 Long Short-Term Memory (LSTM) Network ที่ประกอบด้วยเซลล์ LSTM จำนวน 4 เซลล์ LSTM	20

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.22 โครงสร้างของ Fully-Connected (FC) Neural Network สำหรับ Transfer Learning	21
2.23 Unicorn Hybrid Black แบบสวมหัว 8 Channel	22
2.24 สถาปัตยกรรมซอฟต์แวร์ CoppeliaSim	26
2.25 ตำแหน่งบนแกนบนแขนของหุ่นยนต์	27
2.26 เฟรมของพิกัดลิงก์และพารามิเตอร์ข้อต่อ	28
2.27 แขนกลจำลอง Jaco Arm 2	30
3.1 อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกลด้วยสัญญาณคลื่นไฟฟ้าจากสมอง	32
3.2 บล็อกไดอะแกรมส่วนการวิเคราะห์สัญญาณ EEG (EEG Signal Analysis)	32
3.3 คำสั่งในการใช้งาน EEGLAB Extension	33
3.4 หน้าต่างแรกของ EEGLAB Extension GUI	33
3.5 เหตุการณ์ใน 1 การทดลองย่อยของข้อมูล Stroke Dataset	34
3.6 มิติของข้อมูลในชุดข้อมูลของผู้ป่วยโรคหลอดเลือดสมองหลังได้รับการบำบัด	34
3.7 มิติข้อมูลหลังการเพิ่มเหตุการณ์ที่เกิดการกระตุ้น	35
3.8 ตัวอย่างการนำชุดข้อมูลเข้าโปรแกรม EEGLAB	35
3.9 รายละเอียดของข้อมูล P1_post_training_raw	36
3.10 ตัวอย่างสัญญาณ EEG จากชุดข้อมูลที่นำเข้า	36
3.11 ตัวอย่างการตั้งค่าการดึงเหตุการณ์ Trigger ให้กับชุดข้อมูล	37
3.12 รายละเอียดของข้อมูล P1_post_training_raw หลังดึงเหตุการณ์ Trigger	38
3.13 ตัวอย่างสัญญาณ EEG ของข้อมูล P1_post_training_raw ที่แสดงเหตุการณ์ Trigger ในช่วง 0-20 วินาที	38
3.14 ตัวอย่างการตั้งค่าการแยกเหตุการณ์การจินตนาการขยับมือขวา	39
3.15 รายละเอียดของข้อมูล P1_post_training_raw หลังแยกเหตุการณ์การจินตนาการขยับมือขวา	39
3.16 ตัวอย่างสัญญาณ EEG ของข้อมูล P1_post_training_raw หลังแยกเหตุการณ์การจินตนาการขยับมือขวา 5 Epochs	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.17 ตัวอย่างสัญญาณ EEG ของข้อมูล P1_post_training_raw หลังแยกเหตุการณ์การจินตนาการการขยับมือขวา ใน Epoch แรก	40
3.18 สเปกตรัมของสัญญาณ (a) ก่อนผ่านการกรอง และ (b) หลังผ่านการกรอง	41
3.19 ตัวอย่างการตั้งค่าการกรองสัญญาณใน EEGLAB (a) การตั้งค่าความถี่ตัดของตัวกรองความถี่สูงผ่านที่ 8 Hz และ (b) การตั้งค่าความถี่ตัดของตัวกรองความถี่ต่ำผ่านที่ 30 Hz	42
3.20 การตั้งค่าการกรองสัญญาณด้วยตัวกรอง Notch ที่ความถี่ 50 Hz	42
3.21 เหตุการณ์ที่เกิดขึ้นใน 1 การทดลองย่อยของข้อมูล BCI Competition IV 2a	43
3.22 คำสั่งที่ใช้ประมาณค่าข้อมูลที่ขาดหายด้วยวิธี Cubic Spline Interpolation	44
3.23 ตัวอย่างที่ 1 ของโปรแกรม CSP (a) การสร้างชุดข้อมูลตัวอย่าง และ (b) คำสั่งที่ใช้ในการวิเคราะห์ผลลัพธ์ของอัลกอริธึม CSP	45
3.24 ฟังก์ชัน CSP	46
3.25 ผลลัพธ์ของอัลกอริธึม CSP บนตัวอย่างที่ 1 (a) การกระจายของชุดข้อมูลตั้งต้น และ (b) การกระจายของรูปของชุดข้อมูลหลังผ่านอัลกอริธึม CSP	46
3.26 ผลลัพธ์ของอัลกอริธึม CSP บนตัวอย่างที่ 2 (a) การกระจายของชุดข้อมูลตั้งต้น และ (b) การกระจายของรูปของชุดข้อมูลหลังผ่านอัลกอริธึม CSP	47
3.27 ผลลัพธ์ของอัลกอริธึม CSP บนตัวอย่างที่ 3 (a) การกระจายของชุดข้อมูลตั้งต้น และ (b) การกระจายของรูปของชุดข้อมูลหลังผ่านอัลกอริธึม CSP	47
3.28 คำสั่งการแปลงเวฟเลตแบบดิสครีต	48
3.29 คำสั่งการแปลงเวฟเลตแบบต่อเนื่อง	48
3.30 บล็อกไดอะแกรมการรวมคุณลักษณะการแปลงเวฟเลตแบบดิสครีตและ CSP	49
3.31 บล็อกไดอะแกรมการรวมคุณลักษณะการแปลงเวฟเลตแบบต่อเนื่องและ CSP	49
3.32 คำสั่งในการสร้างโมเดลการเรียนรู้เชิงลึกแบบ MLP	50
3.33 คำสั่งที่ใช้ในการตั้งค่าส่วนประกอบในการสร้างโมเดล	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.34 คำสั่งที่ใช้ในการฝึกสอนโมเดล	50
3.35 คำสั่งที่ใช้ในการหาจำนวนนิเวรอนที่ดีที่สุด	50
3.36 คำสั่งที่ใช้ในการนำเข้าและปรับขนาดของรูปภาพคุณลักษณะ	51
3.37 คำสั่งที่ใช้ในการระบุโมเดล, ปรับแต่งโมเดลและฝึกสอนโมเดล	51
3.38 หน้าต่างแอปพลิเคชัน Experiment Manager	52
3.39 คำสั่ง Setup Function ที่ใช้ในการหาค่า Hyperparameter ที่ดีที่สุด	52
3.40 คำสั่งที่ใช้ในการแบ่งชุดข้อมูลสำหรับการฝึกสอนและตรวจสอบ	53
3.41 คำสั่งที่ใช้ในการเรียนรู้เชิงลึกแบบ LeNet-5 (a) การสร้างโมเดลการเรียนรู้เชิงลึก และ (b) การตั้งค่าส่วนประกอบในการสร้างโมเดลการเรียนรู้เชิงลึกแบบการเรียนรู้เชิงลึก	54
3.42 คำสั่งที่ใช้หาค่า Hyperparameter ที่ดีที่สุดโดยใช้วิธี Bayesian Optimization	54
3.43 คำสั่งที่ใช้ในการลด Learning Rate ใน Callbacks API ของ Keras	54
3.44 คำสั่งที่ใช้ในการฝึกสอนโมเดล	54
3.45 คำสั่งที่ใช้ในการประเมินความแม่นยำของโมเดล	54
3.46 คำสั่งการเรียนรู้เชิงลึกแบบผสม (a) การสร้างโมเดลการเรียนรู้เชิงลึกแบบผสม และ (b) การตั้งค่าส่วนประกอบในการสร้างโมเดลการเรียนรู้เชิงลึกแบบผสม	56
3.47 บล็อกไดอะแกรมของระบบการทดลองแบบ System Training	58
3.48 แผนผังการทำงานของระบบการทดลองแบบ System Training	59
3.49 บล็อกไดอะแกรมของระบบการทดลองแบบ System Testing	60
3.50 แผนผังการทำงานของระบบการทดลองแบบ System Testing	61
3.51 การทดลองเบื้องต้นเพื่อบันทึกและแสดงผลคลื่นไฟฟ้าจากสมอง	62
3.52 การออกแบบการทดลองสำหรับ System Training	63
3.53 กรอบเวลาของกระบวนการรันแบบ System Training ใน 1 การทดลองย่อย	64
3.54 คำสั่งที่ใช้ในการทดลองเก็บคลื่นไฟฟ้าจากสมอง	66
3.55 หน้าจอสีดำ	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.56 หน้าจอรูปร่างเครื่องหมายบวก	67
3.57 หน้าจอแสดงตัวอย่างของลูกศร	67
3.58 กรอบเวลาของกระบวนการที่ค้นแบบ System Testing ใน 1 การทดลองย่อย	68
3.59 บล็อกที่ใช้ในระบบ Offline System Testing สำหรับข้อมูล BCI Competition IV 2a	69
3.60 คำสั่งที่ใช้ในการเตรียมข้อมูลของหนึ่งการทดลองย่อย	69
3.61 ตัวอย่างข้อมูลจากหนึ่งการทดลองย่อย	70
3.62 คำสั่งที่ใช้ในการควบคุมระบบ Offline System Testing	70
3.63 ฟังก์ชัน cwt_bci ที่ใช้ในการแปลง CWT	71
3.64 ฟังก์ชัน predict คำสั่งที่ใช้ในการจำแนกรูปภาพ CWT	71
3.65 บล็อกที่ใช้ในระบบ Real-Time System Testing สำหรับข้อมูลคลื่นไฟฟ้าจากสมองผู้จัดทำ	72
3.66 คำสั่งที่ใช้ในการควบคุมระบบ Real-Time System Testing สำหรับคลื่นไฟฟ้าจากสมองผู้จัดทำ	73
3.67 ตำแหน่งของแต่ละ Electrode บนหมวก	74
3.68 การเชื่อมต่อ Electrode กับหมวก	75
3.69 การเชื่อมต่อ Sticky Electrode กับ Reference Electrode	75
3.70 ลักษณะการสวมหมวกที่ต่อ Unicorn Brain Interface ไว้ที่ศีรษะ	75
3.71 การลือคหมวกกับศีรษะด้วยสายรัดคาง	76
3.72 การติด Sticky Electrode บริเวณหลังใบหู	76
3.73 การเชื่อมต่อ Bluetooth Dongle กับคอมพิวเตอร์	77
3.74 การเปิดการทำงานของอุปกรณ์ Unicorn Brain Interface	77
3.75 หน้าต่างในการเพิ่มและ Activate License บน Unicorn Suite	78
3.76 หน้าต่าง My Unicorn บน Unicorn Suite	78
3.77 หน้าต่าง Unicorn Recorder ที่ส่วนของการปรับการแสดงผลสัญญาณและส่วนของสัญญาณในแต่ละ Electrode โดยเรียงจาก Electrode 1-8 ตามลำดับ	79
3.78 ไชริงและ Unicorn Gel	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.79 การหยอด Unicorn Gel ลงบน Electrode	80
3.80 หน้าต่าง Unicorn Recorder ของสัญญาณที่มีความเสถียรแล้ว	80
3.81 LSL Interface	81
3.82 หน้าต่างโปรแกรม Unicorn LSL	81
3.83 หน้าต่างโปรแกรม OpenViBE Acquisition Server	82
3.84 โปรแกรม OpenViBE Designer	82
3.85 การขยับมือของ (a) Reset Stage (b) Left Hand Stage และ (c) Right Hand stage	83
3.86 ฟังก์ชันที่ใช้ในการขยับการเคลื่อนไหวของ Jaco Arm 2 ใน Coppeliasim	84
3.87 คำสั่งเลือกคอลัมน์ที่ต้องการ	85
3.88 คำสั่งในการหา Index ของแต่ละ Trigger	86
3.89 คำสั่งที่ใช้ในการเตรียมสัญญาณ	87
4.1 สัญญาณของ MI คลาส 1 ในโดเมนเวลาของวิธีการกรองสัญญาณ 3 วิธี	91
4.2 Frequency Response ของตัวกรองความถี่แถบผ่านแบบ Butterworth ของ (a) ฟังก์ชัน bp_filter และ (b) ฟังก์ชัน TESA ใน EEGLAB (TESA Butterworth Filter)	91
4.3 PSD ของสัญญาณหลังผ่านการกรองสัญญาณทั้ง 3 วิธี	92
4.4 ตัวอย่างของค่าข้อมูลที่ถูกประมาณค่าในช่วงด้วยวิธี Cubic Spline Interpolation	92
4.5 ผลลัพธ์สัญญาณก่อนผ่าน (a) และหลังผ่าน (b) ตัวกรองความถี่แถบผ่านแบบ Butterworth	93
4.6 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 1 บน Channel (a) C3 (b) C4	94
4.7 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 2 บน Channel (a) C3 (b) C4	94
4.8 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 3 บน Channel (a) C3 (b) C4	94

## สารบัญญรูป (ต่อ)

รูปที่	หน้า
4.9 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 4 บน Channel (a) C3 (b) C4	95
4.10 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 1 บน Channel (a) C3 (b) C4	95
4.11 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 2 บน Channel (a) C3 (b) C4	95
4.12 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 3 บน Channel (a) C3 (b) C4	96
4.13 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 4 บน Channel (a) C3 (b) C4	96
4.14 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 5 บน Channel (a) C3 (b) C4	96
4.15 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 6 บน Channel (a) C3 (b) C4	97
4.16 ฟังก์ชันในการแบ่งข้อมูลสำหรับเรียนรู้และทดสอบอัลกอริธึม CSP	97
4.17 คำสั่งที่ใช้ในการวิเคราะห์ผลลัพธ์ของอัลกอริธึม CSP	98
4.18 ฟังก์ชัน CSP	98
4.19 ผลลัพธ์ของการกระจายของชุดข้อมูลก่อนเข้าอัลกอริธึม CSP (a) สัญญาณ ที่ถูกกรองด้วยตัวกรองแถบผ่านที่ความถี่ 8-13 Hz (b) สัญญาณที่ถูกกรอง ด้วยตัวกรองแถบผ่านที่ความถี่ 8-30 Hz	100
4.20 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component แรกของเมทริกซ์ W (a) อินพุตของอัลกอริธึม CSP คือชุด ข้อมูลเรียนรู้ (b) อินพุตของอัลกอริธึม CSP คือชุดข้อมูลทดสอบ	101
4.21 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component สุดท้ายของเมทริกซ์ W (a) อินพุตของอัลกอริธึม CSP คือ ชุดข้อมูลเรียนรู้ (b) อินพุตของอัลกอริธึม CSP คือชุดข้อมูลทดสอบ	102

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.22 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component แรกของเมทริกซ์ W เมื่ออินพุตของอัลกอริธึม CSP คือชุดข้อมูลเรียนรู้	103
4.23 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component สุดท้ายของเมทริกซ์ W เมื่ออินพุตของอัลกอริธึม CSP คือชุดข้อมูลเรียนรู้	103
4.24 Eigenvalue ของเมทริกซ์ W จาก Channel (a) C3 (b) C4	104
4.25 DWT ของสัญญาณ EEG และความถี่ที่เกี่ยวข้องแต่ละชั้น	104
4.26 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C3 (a) สัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	105
4.27 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C4 (a) สัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	105
4.28 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือขวา อิเล็กโทรด C3 (a) สัญญาณ EEG คลาสมือขวา อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	106
4.29 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือขวา อิเล็กโทรด C4 (a) สัญญาณ EEG คลาสมือขวา อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	106

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.30 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสลีน อิเล็กโทรด C3 (a) สัญญาณ EEG คลาสลีน อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณ ชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	107
4.31 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสลีน อิเล็กโทรด C4 (a) สัญญาณ EEG คลาสลีน อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณ ชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	107
4.32 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสเท้า อิเล็กโทรด C3 (a) สัญญาณ EEG คลาสเท้า อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณ ชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	108
4.33 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสเท้า อิเล็กโทรด C4 (a) สัญญาณ EEG คลาสเท้า อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณ ชั้นที่ 4 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4	108
4.34 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสมือซ้าย (a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4	109
4.35 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสมือขวา (a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4	109
4.36 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสลีน (a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4	110
4.37 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสเท้า (a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4	110
4.38 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสมือซ้าย	111
4.39 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสมือขวา	111
4.40 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสลีน	112
4.41 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสเท้า	112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญรูป (ต่อ)

รูปที่	หน้า
4.42 Scalogram ของสัญญาณ EEG สำหรับ MI (a) มือซ้าย (b) มือขวา	113
4.43 ตัวอย่าง Scalogram ที่ได้หลังจากการเรียงข้อมูลใน Channel C3, Cz และ C4	113
4.44 สัมประสิทธิ์ละเอียดขั้นที่ 3 ของ DWT คลาสมือซ้ายของข้อมูลชุดฝึกสอน เมื่อผ่านการแปลง CSP	114
4.45 สัมประสิทธิ์ละเอียดขั้นที่ 3 ของ DWT คลาสมือขวาของข้อมูลชุดฝึกสอน เมื่อผ่านการแปลง CSP	114
4.46 สัมประสิทธิ์ละเอียดขั้นที่ 3 ของ DWT คลาสมือซ้ายของชุดทดสอบเมื่อผ่านการแปลง CSP	115
4.47 สัมประสิทธิ์ละเอียดขั้นที่ 3 ของ DWT คลาสมือขวาของชุดทดสอบเมื่อผ่านการแปลง CSP	115
4.48 สัมประสิทธิ์ละเอียดขั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดฝึกสอนเมื่อผ่านการแปลง CSP	116
4.49 สัมประสิทธิ์ละเอียดขั้นที่ 4 ของ DWT คลาสมือขวาของชุดฝึกสอนเมื่อผ่านการแปลง CSP	116
4.50 สัมประสิทธิ์ละเอียดขั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดทดสอบเมื่อผ่านการแปลง CSP	117
4.51 สัมประสิทธิ์ละเอียดขั้นที่ 4 ของ DWT คลาสมือขวาของชุดทดสอบเมื่อผ่านการแปลง CSP	117
4.52 สัมประสิทธิ์ละเอียดขั้นที่ 3 รวมกับขั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดฝึกสอนเมื่อผ่านการแปลง CSP	118
4.53 สัมประสิทธิ์ละเอียดขั้นที่ 3 รวมกับขั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดฝึกสอนเมื่อผ่านการแปลง CSP	118
4.54 สัมประสิทธิ์ละเอียดขั้นที่ 3 รวมกับขั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดทดสอบเมื่อผ่านการแปลง CSP	119
4.55 สัมประสิทธิ์ละเอียดขั้นที่ 3 รวมกับขั้นที่ 4 ของ DWT คลาสมือขวาของชุดทดสอบเมื่อผ่านการแปลง CSP	119

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.56 ภาพสัมพันธ์การแปลงเวฟเลตแบบต่อเนื่อง (a) คลาสมือซ้าย และ (b) คลาสมือขวา	120
4.57 ภาพสัมพันธ์การแปลงเวฟเลตแบบต่อเนื่องของคลาสมือซ้ายเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกของเมทริกซ์ W	121
4.58 ภาพสัมพันธ์การแปลงเวฟเลตแบบต่อเนื่องของคลาสมือขวาเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกของเมทริกซ์ W	121
4.59 ภาพสัมพันธ์การแปลงเวฟเลตแบบต่อเนื่องของคลาสมือซ้ายเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกและสองคอลัมน์สุดท้ายของเมทริกซ์ W	122
4.60 ภาพสัมพันธ์การแปลงเวฟเลตแบบต่อเนื่องของคลาสมือขวาเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกและสองคอลัมน์สุดท้ายของเมทริกซ์ W	122
4.61 สัมประสิทธิ์การแปลงเวฟเลตแบบต่อเนื่องของคลาสมือซ้ายเมื่อผ่านการแปลง CSP โดยใช้ทุกคอลัมน์ของเมทริกซ์ W	123
4.62 สัมประสิทธิ์การแปลงเวฟเลตแบบต่อเนื่องของคลาสมือขวาเมื่อผ่านการแปลง CSP โดยใช้ทุกคอลัมน์ของเมทริกซ์ W	123
4.63 Confusion Matrix การจำแนกชุดข้อมูลทดสอบ	131
4.64 CWT ของ MI มือซ้าย	132
4.65 ผลการควบคุมแขนกลจำลองของ MI มือซ้าย	132
4.66 CWT ของ MI มือขวา	133
4.67 ผลการควบคุมแขนกลจำลองของ MI มือขวา	133
4.68 หน้าจอแสดงจุดสีแดงระยะเวลา 2 วินาที	134
4.69 หน้าจอแสดงจุดสีเขียวระยะเวลา 3 วินาที	135
4.70 สัญญาณคลื่นไฟฟ้าจากสมองบน Channel C3	135
4.71 สัญญาณคลื่นไฟฟ้าจากสมองบน Channel C3 เมื่อผ่านการ Normalize	135
4.72 ผลลัพธ์การแปลง CWT จากคลื่นไฟฟ้าจากสมอง MI ผู้จัดทำ	136
4.73 ผลลัพธ์การจำแนกของโมเดล	136
4.74 ผลการควบคุมแขนกลด้วยคลื่นไฟฟ้าจากสมอง	136

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
2.1 ช่วงความถี่ที่เกี่ยวข้องกับ Motor Imagery	4
2.2 สถาปัตยกรรมของ ShuffleNet	17
2.3 DH Parameter	30
2.4 การกำหนดค่าในแนวตั้งใน Kinova Convention	31
2.5 การกำหนดค่าในแนวตั้งใน DH Convention	31
2.6 การแปลงพิกัดจาก DH เป็น Kinova	31
2.7 การแปลงพิกัดจาก Kinova เป็น DH	31
3.1 การกำหนดค่าเริ่มต้นของพิกัดใน DH Parameter	84
4.1 เวลาที่ใช้ในการจำแนกข้อมูลด้วยอัลกอริธึมการเรียนรู้เชิงลึก	124
4.2 มิติของข้อมูลหลังผ่านการแยกคุณลักษณะด้วยวิธีต่างๆ	125
4.3 ผลลัพธ์การจำแนกคลาสข้อมูลของ Subject ที่ 9 โดยใช้คุณลักษณะทั้ง 9 แบบด้วยอัลกอริธึมการเรียนรู้เชิงลึก	125
4.4 ผลการทำ 5-Fold Cross Validation ของการฝึกสอนโมเดลโครงสร้าง ResNet18 และ HDNN	129
4.5 ผลการจำแนกคลาสด้วยอัลกอริธึมการเรียนรู้เชิงลึกด้วยชุดข้อมูลชุดทดสอบของผู้จัดทำ	130

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัญญาประดิษฐ์กลายเป็นส่วนสำคัญในระบบอุตสาหกรรมต่างๆ และงานวิจัยเกี่ยวกับ Brain-Computer Interface (BCI) มีความนิยมมากขึ้นในปัจจุบัน ปัญหานี้จึงมีจุดประสงค์เพื่อศึกษาการประมวลผลสัญญาณคลื่นไฟฟ้าจากสมองด้วยอัลกอริธึมการเรียนรู้เชิงลึกแบบต่างๆ เพื่อนำไปควบคุมแขนกลจำลอง เพื่อเป็นการปูพื้นฐานให้ผู้จัดทำในการนำ BCI มาประยุกต์ใช้ในระบบการดูแลสุขภาพเพื่อช่วยเหลือผู้พิการและผู้สูงอายุในการทำกิจกรรมประจำวันต่อไป

### 1.2 วัตถุประสงค์

- 1) ศึกษาอุปกรณ์และวิธีการบันทึกสัญญาณคลื่นไฟฟ้าจากสมอง (Electroencephalographic Data: EEG) และการเตรียมสัญญาณ
- 2) ศึกษาการแยกคุณลักษณะข้อมูล (Feature Extraction)
- 3) ศึกษาอัลกอริธึมการเรียนรู้เชิงลึกเพื่อจำแนกสัญญาณ EEG
- 4) ศึกษาการเขียนโปรแกรมเพื่อควบคุมการทำงานของแขนกล

### 1.3 ขอบเขตของปัญญานิพนธ์

ปัญญานิพนธ์นี้ทำการศึกษาและพัฒนาอัลกอริธึมการเรียนรู้เชิงลึกโดยใช้สัญญาณคลื่นไฟฟ้าจากสมอง (EEG) เพื่อควบคุมแขนกล โดยส่วนของข้อมูล EEG จะใช้ข้อมูลจากฐานข้อมูลออนไลน์และข้อมูลที่ทำการบันทึกด้วยตัวเองด้วยเซนเซอร์ ข้อมูลที่ได้จะถูกนำไปถูกจัดเตรียมเบื้องต้นดึงคุณลักษณะ ทำการฝึกสอนโมเดลและสุดท้ายปรับแต่งพารามิเตอร์ เมื่อได้โมเดลที่พร้อมใช้งานแล้วจะนำไปประยุกต์ใช้ในการควบคุมแขนกลจำลองในคอมพิวเตอร์

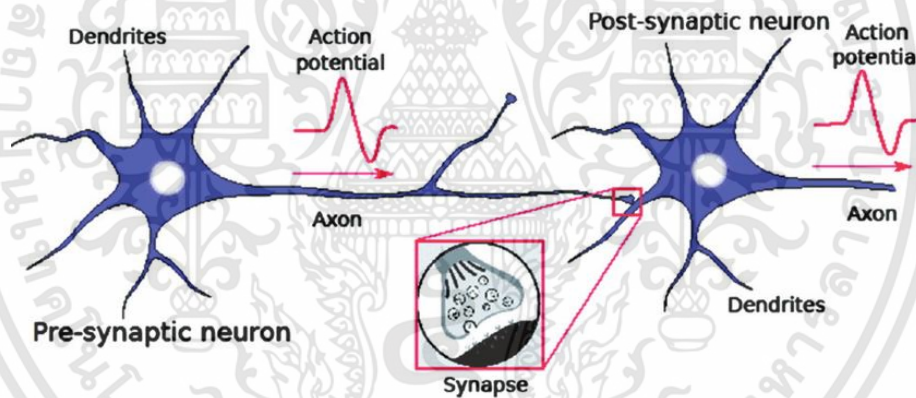
## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 คลื่นไฟฟ้าจากสมอง (EEG signal)

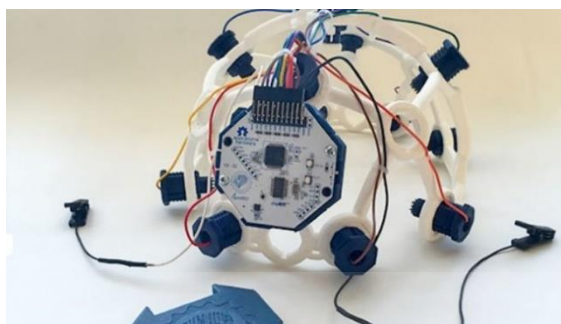
สมองประกอบด้วยเซลล์หลายพันล้านเซลล์ซึ่งครึ่งหนึ่งเป็นเซลล์ประสาท (Neurons) และอีกครึ่งหนึ่งช่วยอำนวยความสะดวกในเรื่องกิจกรรมของเซลล์ประสาท เซลล์ประสาทเหล่านี้มีการเชื่อมต่อหนาแน่นผ่านไซแนปส์ (Synapse) หรือจุดประสานประสาท ดังแสดงในรูปที่ 2.1 [1]

การส่งผ่านไซแนปส์นั้นจะสร้างแรงกระตุ้นไฟฟ้าที่ละเอียดอ่อนเรียกว่า Postsynaptic Potential ซึ่งการวัดจากเซลล์ประสาทเซลล์เดียวโดยไม่ไปวัดโดยตรงนั้นจะทำการวัดอย่างแม่นยำได้ยาก แต่เมื่อเซลล์ประสาทหลายพันเซลล์เชื่อมต่อกันและเกิดการส่งพร้อมกันจะเกิดสนามไฟฟ้าที่แข็งแกร่งพอที่จะแพร่กระจายผ่านเนื้อเยื่อ กระดูกและกะโหลกจนสามารถวัดผ่านหนังศีรษะได้ [2] โดยสัญญาณที่เกิดขึ้นคือคลื่นไฟฟ้าจากสมอง (Electroencephalographic Signal หรือ EEG Signal)



รูปที่ 2.1 ส่วนประกอบของเซลล์ประสาท

ความแรงของสัญญาณ EEG นั้นจะมีค่าค่อนข้างน้อยซึ่งถูกวัดอยู่ในหน่วย Microvolts ( $\mu V$ ) [3] อุปกรณ์ที่ใช้วัดคลื่นไฟฟ้าจากสมองมีชื่อว่า Electroencephalogram โดยอุปกรณ์นี้ประกอบไปด้วย อิเล็กโทรด เจลนำไฟฟ้า แอมพลิฟายเออร์และ ADC โดยอุปกรณ์แสดงดังรูปที่ 2.2 [4] อิเล็กโทรดหรือตัวนำถูกใช้ในการนำกระแสไฟฟ้าที่เกิดขึ้นบนหนังศีรษะโดยอิเล็กโทรดจะติดตั้งไว้บนหนังศีรษะพร้อมด้วยเจลนำไฟฟ้า (Ag-Cl) จำนวนเล็กน้อย [5]

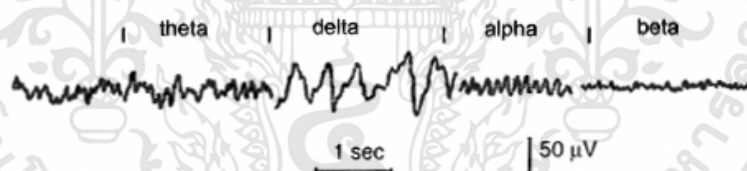


รูปที่ 2.2 เครื่อง Electroencephalogram

ความถี่ของสัญญาณคลื่นไฟฟ้าจากสมองของมนุษย์นั้นสามารถจำแนกออกมาได้ 5 แถบ ความถี่ดังนี้

1. เดลตา (Delta) : มีความถี่อยู่ในช่วงที่ต่ำกว่า 3 Hz
2. ทีตา (Theta) : มีความถี่อยู่ในช่วงระหว่าง 4 ถึง 7 Hz
3. แอลฟา (Alpha) : มีความถี่อยู่ในช่วงระหว่าง 8 ถึง 13 Hz
4. เบตา (Beta) : มีความถี่อยู่ในช่วงระหว่าง 13 ถึง 30 Hz
5. แกมมา (Gamma) : มีความถี่อยู่ในช่วงที่มากกว่า 30 Hz

ตัวอย่างสัญญาณในช่วงความถี่ต่างๆ แสดงดังรูปที่ 2.3 [6]



รูปที่ 2.3 ช่วงความถี่ของสัญญาณคลื่นไฟฟ้าจากสมอง

### 2.1.1 มโนภาพการเคลื่อนไหว (Motor Imagery)

มโนภาพการเคลื่อนไหว (Motor Imagery; MI) เป็นสัญญาณคลื่นไฟฟ้าจากสมองที่เกิดขึ้นตามธรรมชาติจากการที่ผู้ใช้งานจินตนาการถึงการเคลื่อนไหว (Motor Movement) เช่น การจินตนาการว่ากำลังขยับแขน เป็นต้น โดย MI ในงานวิจัยส่วนใหญ่คือการจินตนาการถึงการเคลื่อนไหว แขนขวา แขนซ้าย ขาซ้าย ขาขวา ขาทั้งสองข้างและลิ้น [7]

ช่วงความถี่ที่เกี่ยวข้องกับ MI นั้นจะมีอยู่ในช่วง Alpha (8-12 Hz), Beta (16-24 Hz), และ Gamma (30-35 Hz) โดยสัญญาณคลื่นไฟฟ้าจากสมองในช่วง Alpha นั้นจะเรียกว่า Mu-band โดยช่วงความถี่ของสัญญาณคลื่นไฟฟ้าสมองที่เกี่ยวข้องกับ MI แสดงดังตารางที่ 2.1 [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ช่วงความถี่ที่เกี่ยวข้องกับ Motor Imagery

Band name	Frequency band (Hz)	MI related frequency band (Hz)
Delta	1-3	-
Theta	4-7	-
Alpha	8-13	8-12
Beta	13-30	16-24
Gamma	Above 30	30-35

ในช่วงของการจินตนาการถึงการขยับมือซ้ายหรือมือขวานั้นจะเกิด 1) การลดลงของกำลังในช่วง Mu-band และ Beta-band ของสัญญาณ โดยจะเรียกปรากฏการณ์นี้ว่า Event-Related Desynchronization (ERD) และ 2) กำลังของสัญญาณทั้งสองช่วงความถี่นั้นได้ฟื้นฟูกลับมาหลังจากจบการจินตนาการแล้ว เรียกปรากฏการณ์นี้ว่า Event-Related Synchronization (ERS) ปริมาณที่ใช้ในการแสดง ERD และ ERS ที่เกิดขึ้นใน MI คืออัตราส่วน ERD/ERS [9]

ERD/ERS นั้นสามารถวัดได้จากการเปลี่ยนแปลงของกำลังในสัญญาณ EEG เมื่อเกิด MI เปรียบเทียบกับช่วงเวลาอ้างอิง (Reference Interval) หรือช่วงเวลาก่อนที่จะเริ่มการจินตนาการ โดยสามารถคำนวณได้จากอัตราการเปลี่ยนแปลงของกำลังเทียบกับสัญญาณอ้างอิง [10] ดังสมการที่ (2.1) – (2.3)

$$EEG_{avg}(j) = \frac{1}{N} \sum_{i=1}^N s_{i,j}^2 \quad (2.1)$$

$$EEG_{ref} = \frac{1}{k} \sum_{j=1}^k EEG_{avg}(j) \quad (2.2)$$

$$\frac{ERD}{ERS}(j)(\%) = \left( \frac{EEG_{avg}(j) - EEG_{ref}}{EEG_{ref}} \right) \times 100\% \quad (2.3)$$

โดย  $N$  คือ จำนวนของการทดลองย่อย (Trial)

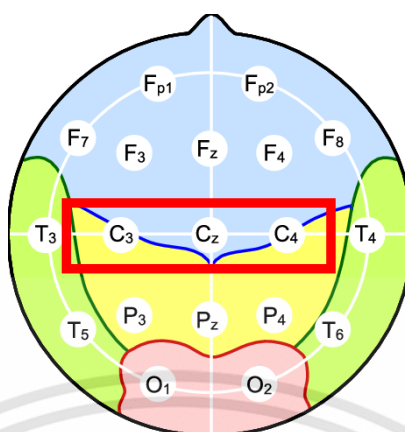
$k$  คือ จำนวนข้อมูลในช่วงอ้างอิง

$s_{ij}$  คือ จุดข้อมูลลำดับที่  $j$  ที่การทดลอง  $i$  ของสัญญาณ EEG ที่ผ่านตัวกรองความถี่แถบผ่าน

$EEG_{avg}(j)$  คือ กำลังเฉลี่ยของสัญญาณ MI EEG จุดข้อมูลลำดับที่  $j$  ในทุกการทดลอง

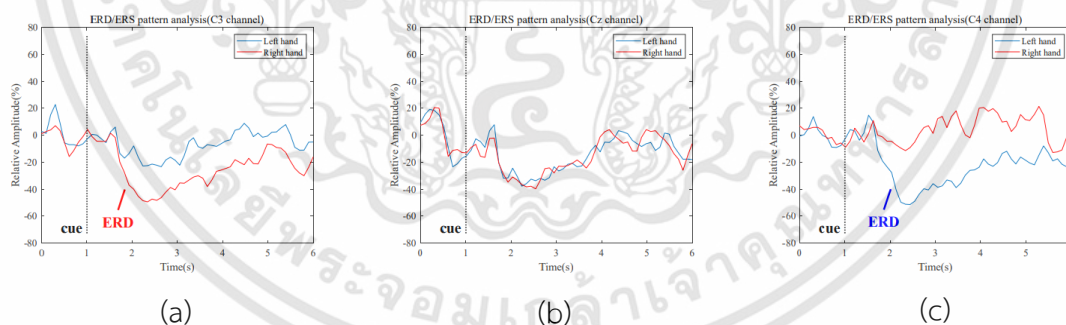
$EEG_{ref}$  คือ กำลังเฉลี่ยของสัญญาณ MI EEG ที่ถูกวัดในช่วงอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตำแหน่งของอิเล็กโทรดที่ใช้วัดสัญญาณ EEG

รูปที่ 2.4 แสดงตำแหน่งของอิเล็กโทรดที่ใช้วัดสัญญาณ EEG [11] โดยปกติแล้วรูปแบบ ERD/ERS นั้นสามารถสังเกตได้ในบริเวณส่วน Motor Cortex ซึ่งอยู่ใกล้อิเล็กโทรด C3, Cz, และ C4 ตัวอย่างรูปแบบ ERD/ERS ใน Mu-band ซึ่งถูกตรวจวัดจากอิเล็กโทรด C3, Cz, และ C4 แสดงในรูปที่ 2.5 (a)-(c) ตามลำดับ [12] โดยแอมพลิจูดนั้นแสดงค่าที่ถูกคำนวณมาจากสมการที่ (2.3) จะเห็นว่ารูปแบบ ERD จะเกิดขึ้นระหว่างที่ทำ MI (วินาทีที่ 2-5) โดยการจินตนาการถึงการเคลื่อนไหวมือนั้น จะเกิด ERD ที่อิเล็กโทรด C3 และการจินตนาการถึงการเคลื่อนไหวมือซ้ายนั้นเกิด ERD ที่อิเล็กโทรด C4 อย่างไรก็ตาม ERD/ERS ใน Cz นั้นไม่สามารถมองเห็นได้ชัดเจนเมื่อเทียบกับอิเล็กโทรดอื่นๆ



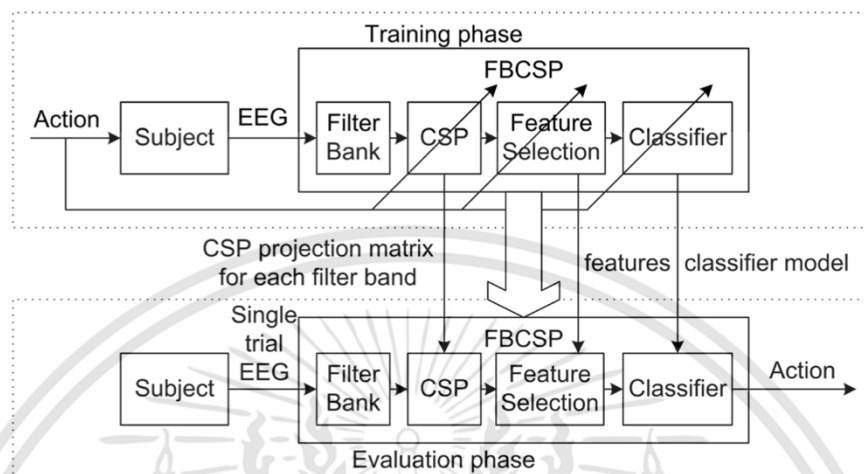
รูปที่ 2.5 ตัวอย่าง ERD/ERS ใน Mu-band (a) อิเล็กโทรด C3 (b) อิเล็กโทรด Cz (c) อิเล็กโทรด C4

## 2.2 Filter Bank Common Spatial Pattern (FBCSP)

อัลกอริธึม Filter Bank Common Spatial Pattern (FBCSP) มีสถาปัตยกรรมดังรูปที่ 2.6 โดย FBCSP ประกอบด้วย 4 ขั้นตอนของการประมวลผลสัญญาณ (Signal Processing) และการเรียนรู้ของเครื่อง (Machine Learning) บนข้อมูล EEG ได้แก่ 1) Filter Bank ซึ่งประกอบไปด้วยการกรองด้วยตัวกรองแถบผ่านชนิด Chebyshev Type II, 2) การกรองเชิงพื้นที่ (Spatial Filtering) โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้อัลกอริธึม Common Spatial Pattern (CSP), 3) การเลือกคุณลักษณะของสัญญาณ และ 4) การจำแนกสัญญาณจากคุณลักษณะ [13]



รูปที่ 2.6 ตัวอย่างสถาปัตยกรรมของอัลกอริธึม Filter Bank Common Spatial Pattern (FBCSP) สำหรับขั้นตอนการฝึกอบรมและการประเมิน

### 2.2.1 Filter Bank

ขั้นตอนแรกของอัลกอริธึม FBCSP คือการใช้ Filter Bank เพื่อแยกสัญญาณ EEG ออกเป็นหลายความถี่แถบผ่านโดยใช้ตัวกรองชนิด Chebyshev Type II ซึ่งจะได้ความถี่แถบผ่านในช่วง 4-40 Hz ความกว้าง 4 Hz ทั้งหมด 9 ช่วง [13]

### 2.2.2 Common Spatial Pattern (CSP)

ขั้นตอนถัดจากการแบ่งสัญญาณออกเป็นหลายช่วงความถี่แถบผ่านคือขั้นตอนการกรองเชิงพื้นที่ ซึ่งการกรองเชิงพื้นที่ที่ได้มาจากอัลกอริธึม CSP โดยการกรองเชิงพื้นที่ที่ได้มักนิยมใช้เพื่อแปลงสัญญาณดิบ MI EEG อัลกอริธึมนี้ประสบความสำเร็จเป็นอย่างสูงในการคำนวณตัวกรองเชิงพื้นที่สำหรับการตรวจจับ ERD/ERS [14-16] อัลกอริธึม CSP เป็นเทคนิคหนึ่งที่ใช้วิเคราะห์ข้อมูลแบบ 2 คลาสที่มีหลายช่องสัญญาณโดยผลลัพธ์ของอัลกอริธึมจะแยกสัญญาณ EEG ออกเป็นส่วนด้วยเมทริกซ์  $W_b \in \mathbb{R}^{c \times c}$  เพื่อ Project จากสัญญาณ  $X_{b,i} \in \mathbb{R}^{c \times t}$  เป็น  $Z_{b,i} \in \mathbb{R}^{c \times t}$  โดยใช้สมการที่ (2.4) [17]

$$Z_{b,i} = W_b^T X_{b,i} \quad (2.4)$$

เมื่อ  $X_{b,i}$  คือ สัญญาณ EEG จากตัวกรองแถบผ่านช่วงที่  $b$  ของ Trial ที่  $i$

$Z_{b,i}$  คือ สัญญาณ  $X_{b,i}$  หลังผ่านการกรองเชิงพื้นที่

$W_b$  คือ Projection Matrix ของ CSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $c$  คือ จำนวนของช่องสัญญาณ  
 $t$  คือ จำนวนจุดข้อมูล EEG ในแต่ละช่องสัญญาณ  
 $T$  คือ โอเปอเรเตอร์ทรานสโพส

จากสมการที่ (2.4) เมื่อต้องการแบ่งคลาสของสัญญาณ EEG จะต้องเลือก  $N_w$  คอลัมน์แรกและท้ายของ  $W_b$  เพื่อเปลี่ยนให้อยู่ในรูปตัวกรองเชิงพื้นที่ใหม่  $\bar{W}_b \in \mathbb{R}^{c \times (2 \times N_w)}$  ซึ่งคอลัมน์แรกและท้ายของ  $W_b$  จะสอดคล้องกับค่าเจาะจง (Eigenvalue) ที่มากที่สุดและน้อยที่สุดตามลำดับ ดังนั้นการแปลงสัญญาณ EEG  $X_{b,i}$  ที่ถูกรองด้วยตัวกรองแถบผ่านโดยการคูณ  $\bar{W}_b$  จะได้เอาต์พุตของตัวกรองเชิงพื้นที่ดังสมการที่ (2.5) เมื่อ  $\bar{Z}_{b,i} \in \mathbb{R}^{(2 \times N_w) \times t}$  [18]

$$\bar{Z}_{b,i} = \bar{W}_b^T X_{b,i} \quad (2.5)$$

ในกรณีของการจำแนกประเภทแบบ  $k$  คลาสเมื่อ  $k \geq 3$  กระบวนการ One-Versus-Rest (OVR) จะถูกใช้กับอัลกอริธึมนี้ ส่งผลให้ค่าของ  $c$  จะเท่ากับ  $k \times 2 \times N_w$  และจำนวนมิติของเอาต์พุตจะเท่ากับ  $(k \times 2 \times N_w) \times t$  [18] เมทริกซ์ตัวกรองเชิงพื้นที่  $W_b \in \mathbb{R}^{c \times c}$  สามารถคำนวณได้จากการหาค่าที่ดีที่สุดของ Objective Function ดังสมการที่ (2.6) [18] ภายใต้ข้อจำกัดของสมการที่ (2.7) และ (2.8)

$$J(w_{b,i}) = \frac{w_{b,i}^T \Sigma_{b,1} w_{b,i}}{w_{b,i}^T \Sigma_{b,2} w_{b,i}} \quad (2.6)$$

$$w_{b,i}^T \Sigma_{b,1} w_{b,i} + w_{b,i}^T \Sigma_{b,2} w_{b,i} = 1 \quad (2.7)$$

$$w_{b,i} = \log(\text{var}(\bar{Z}_{b,i})) \quad (2.8)$$

เมื่อ  $\Sigma_{b,1}$  และ  $\Sigma_{b,2}$  คือ ค่าประมาณของเมทริกซ์ความแปรปรวนร่วมของ EEG ที่เกี่ยวกับ MI คลาส 1 และ 2 ตามลำดับ

อัลกอริธึม CSP จะคำนวณการแปลงเมทริกซ์ เพื่อให้ค่าความแปรปรวนของลักษณะเฉพาะเหมาะสมสำหรับการแบ่งคลาสของการวัด EEG [16] จากสมการที่ (2.7) ปัญหาการเพิ่มประสิทธิภาพของข้อจำกัดนี้สามารถแปลงเป็นการแก้ปัญหาค่าเจาะจง (Eigenvalue Decomposition Problem) ได้โดยใช้สมการที่ (2.9) [18]

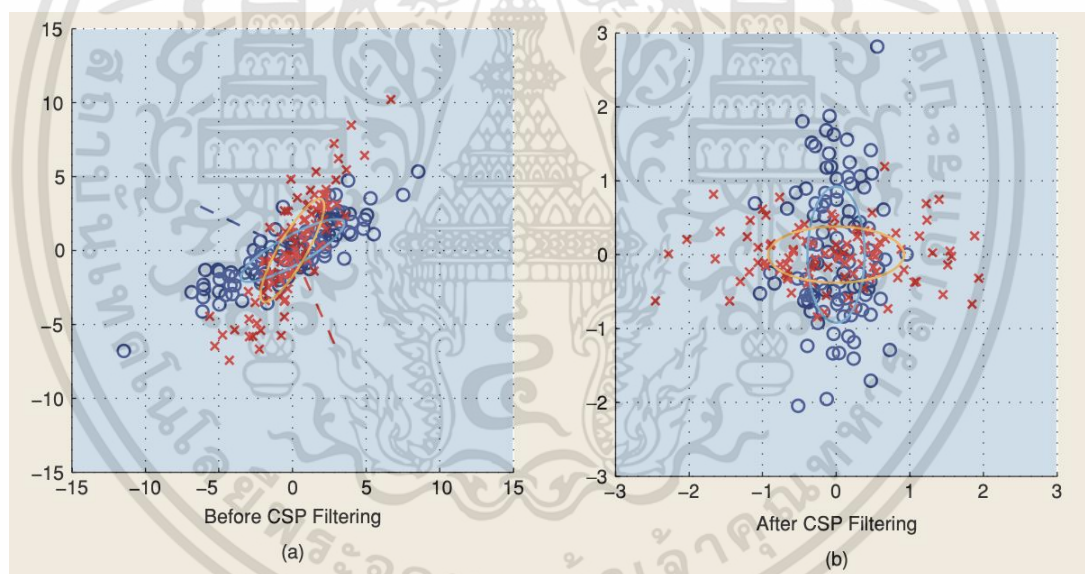
$$\Sigma_{b,1} W_b = (\Sigma_{b,1} + \Sigma_{b,2}) W_b D_b \quad (2.9)$$

เมื่อ  $D_b$  คือ เมทริกซ์ทแยงมุมที่ประกอบไปด้วยค่าเจาะจงของ  $\Sigma_{b,1}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางเทคนิค  $W_b$  สามารถคำนวณในโปรแกรม MATLAB ได้โดยการใช้ฟังก์ชัน  $eig()$  ตั้งคำสั่ง  $W = eig(S1, S1 + S2)$  เมื่อ  $W$ ,  $S1$  และ  $S2$  หมายถึง  $W_b$ ,  $\Sigma_{b,1}$  และ  $\Sigma_{b,2}$  ตามลำดับ

ค่าของสมการที่ (2.6) ที่มีค่าเข้าใกล้ 1 แสดงว่าตัวกรองเชิงพื้นที่สอดคล้องกับค่าความแปรปรวนสูงในสถานะที่เป็นคลาสแรกหรือเชิงบวกและให้ค่าความแปรปรวนต่ำในสถานะที่เป็นคลาสสองหรือเชิงลบตามลำดับ โดยค่าความแปรปรวนสามารถสลับคลาสกันได้ ดังนั้นความแตกต่างของสองคลาสนี้จะนำไปใช้แบ่งคลาสของสัญญาณได้ ซึ่งตัวอย่างของการแบ่งคลาสโดยการใช้อัลกอริทึม CSP ในแบบ 2 มิติแสดงดังรูปที่ 2.7 ตัวอย่างของข้อมูล 2 คลาส (กากบาทสีแดงและวงกลมสีน้ำเงิน) มาจากการสุ่มด้วยการแจกแจงแบบเกาส์เซียนสองชุด อัลกอริทึม CSP จะ Map ตัวอย่างจากรูปที่ 2.7 (a) ไปเป็นรูปที่ 2.7 (b) ซึ่งในรูปที่ 2.7 (b) จะเห็นว่าความสัมพันธ์ที่หนาแน่นระหว่างแกนเดิมทั้งสองจากรูปที่ 2.7 (a) จะถูกกระจายและกำจัดความสัมพันธ์ นอกจากนี้การแจกแจงของทั้งสองแกนจะแตกต่างกันมากที่สุดตามแกนใหม่ เส้นประในรูปที่ 2.7 (a) หมายถึงทิศทางของการคาดคะเนของ CSP (CSP Projection) สังเกตว่าทั้งสองเวกเตอร์นั้นเกือบจะตั้งฉากกัน [17]



รูปที่ 2.7 ตัวอย่างของการแจกแจงข้อมูล 2 คลาส (a) ก่อนกรอง และ (b) หลังกรอง  
ด้วย CSP ใน 2 มิติ

### 2.3 การแปลงเวฟเลตแบบต่อเนื่อง (Continuous Wavelet Transform; CWT)

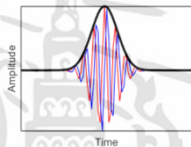
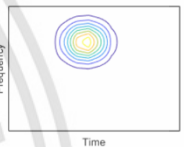
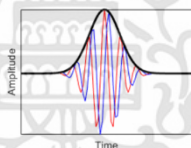
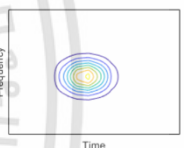
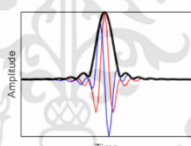
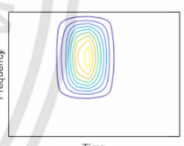
การแปลงเวฟเลตแบบต่อเนื่อง [19] ทำให้สามารถสร้างภาพสองมิติที่อยู่ในคุณสมบัติมิติของเวลาและความถี่ โดยความถี่ของ CWT จะมีค่าเฉลี่ยเป็นศูนย์และเป็นคลื่นที่จำกัดในเวลา โดยตัวแปรในการขยายและการเลื่อนสัญญาณในมิติของเวลานั้นเป็นส่วนสำคัญในการวิเคราะห์สัญญาณ สมการของ CWT แสดงในสมการที่ (2.10) [20-21]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$CWT = \frac{1}{\sqrt{|s|}} \int f(t) \psi^* \left( \frac{t - \omega}{s} \right) dt \quad (2.10)$$

- เมื่อ  $\psi$  คือ Mother Wavelet  
 $\psi^*$  คือ Complex Conjugate ของ Mother Wavelet  
 $\omega$  คือ ตัวแปรในการเลื่อน  
 $s$  คือ ตัวแปรในการขยาย

ใน MATLAB มีเวฟเลตแบบต่อเนื่องให้เลือกใช้หลักๆ อยู่ 3 ประเภทคือ Generalized Morse Wavelet, Analytic Morlet (Gabor) Wavelet และ Bump Wavelet โดยมีลักษณะและคุณสมบัติแสดงดังรูปที่ 2.8

Wavelet	Features	Name	Time Domain	Time-Frequency Domain
Generalized Morse Wavelet	Can vary two parameters to change time and frequency spread	'morse' (default)		
Analytic Morlet (Gabor) Wavelet	Equal variance in time and frequency	'amor'		
Bump Wavelet	Wider variance in time, narrower variance in frequency	'bump'		

รูปที่ 2.8 คุณสมบัติของเวฟเลตแบบต่อเนื่องใน MATLAB

## 2.4 การแปลงเวฟเลตแบบดิสครีต (Discrete Wavelet Transform; DWT)

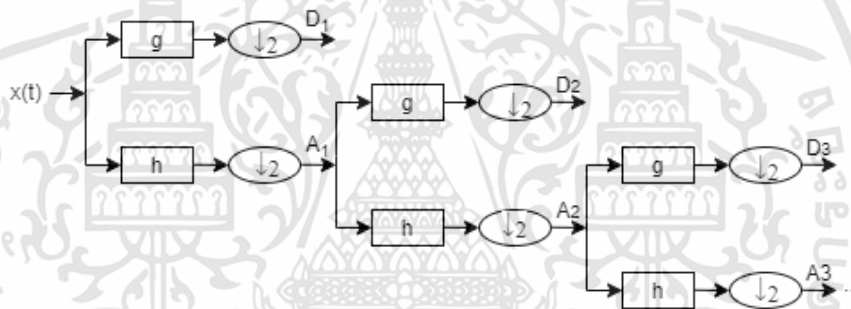
เนื่องจากสัญญาณ EEG เป็นสัญญาณแบบ Non-Stationary และ Non-Linear ดังนั้นการแปลงเวฟเลตจะมีข้อดีในการเก็บข้อมูลในเชิงมิติของเวลาและความถี่ การแปลงเวฟเลตเกิดจากการขยายและเลื่อนของ Basis Wavelet การแปลงเวฟเลตแบบดิสครีตสามารถแก้ปัญหาความความซ้ำซ้อนของการแปลงเวฟเลตแบบต่อเนื่องโดยตัวแปรในการขยายจะเป็นแบบดิสครีตและตัวแปรในการเลื่อนจะเป็นสัดส่วนโดยตรงกับตัวแปรในการขยาย [22] สมการทั่วไปของการแปลงเวฟเลตแบบดิสครีตแสดงในสมการที่ (2.11)

$$DWT_x(a, \tau) = \int_{-\infty}^{\infty} x(t) 2^{-a/2} \psi(2^{-a/2}t - \tau) dt \quad (2.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย  $\psi$  คือ Basis Wavelet Function  
 $a$  คือ ลำดับชั้นจากการขยาย  
 $\tau$  คือ ตำแหน่งการเลื่อน

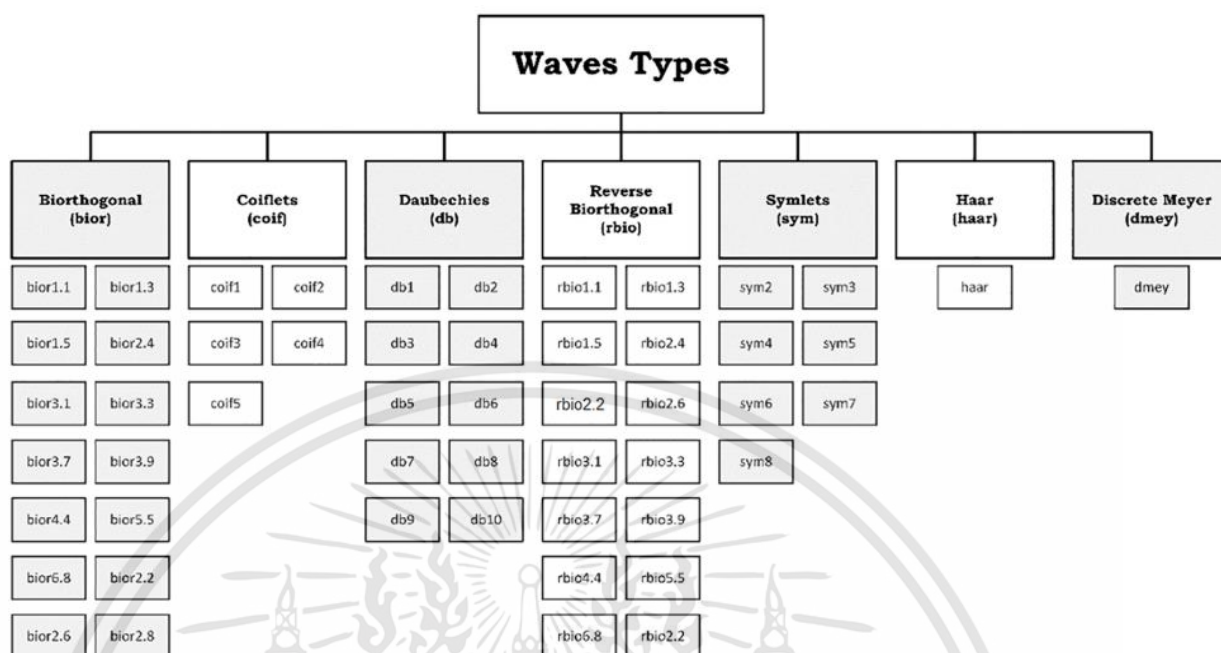
DWT แยกสัญญาณออกเป็นกลุ่มย่อยโดยกรองสัญญาณผ่านตัวกรองความถี่สูงและตัวกรองความถี่ต่ำในมิติของเวลา แสดงดังรูปที่ 2.9 กำหนดให้  $x(t)$  แทนสัญญาณในโดเมนเวลา ตัวกรองความถี่สูง  $g$  คือ Mother Wavelet แบบดิสครีต และตัวกรองความถี่ต่ำ  $h$  คือการสะท้อนของ Mother Wavelet โดยสัญญาณหลังผ่านการกรองและการสุ่มในขั้นแรกจะถูกเรียกว่า ค่าสัมประสิทธิ์ประมาณ  $A_1$  และ ค่าสัมประสิทธิ์ละเอียด  $D_1$  เมื่อนำ  $A_1$  ไปผ่านตัวกรองความถี่และการสุ่มจะได้ค่าสัมประสิทธิ์ในขั้นต่อไป จำนวนขั้นในการแยกสัญญาณโดยใช้ DWT ขึ้นอยู่กับช่วงความถี่ของสัญญาณ [23]



รูปที่ 2.9 การแบ่งกลุ่มย่อยของสัญญาณโดยใช้ DWT

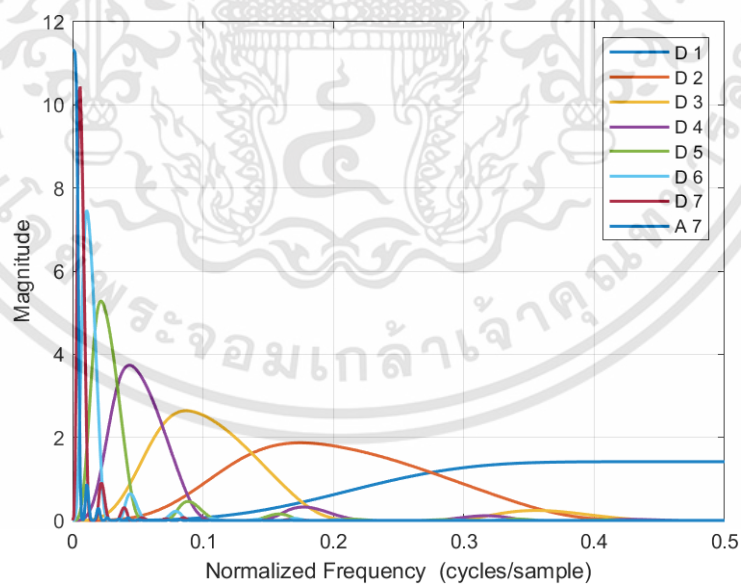
DWT ถูกนิยามโดย Wavelet Function  $\psi(t)$  เช่น Mother Wavelet และ Scaling Function  $\varphi(t)$  ในมิติของเวลา โดยสามารถแบ่งได้ออกเป็นสองแบบคือ Orthogonal และ Biorthogonal Wavelets โดย Orthogonal Wavelets จะมีจำนวน Scaling Function และ Wavelet Function เพียงอันเดียว ในขณะที่ Biorthogonal Wavelets จะมี Scaling Function และ Wavelet Function สองอันสำหรับการแยกและรวมสัญญาณตามลำดับ [24]

เวฟเลตแบบดิสครีตถูกแบ่งประเภทออกเป็นตระกูลโดยพิจารณาจากส่วนประกอบความถี่เป็นหลัก ตัวอย่างการแบ่งประเภทของเวฟเลตแบบดิสครีตแสดงดังรูปที่ 2.10 [25] โดย Daubechies-4 (Db4) ได้รับความนิยมสูงสุดกับสัญญาณ EEG เพราะสามารถตรวจจับการเปลี่ยนแปลงของสัญญาณได้ดี [26]



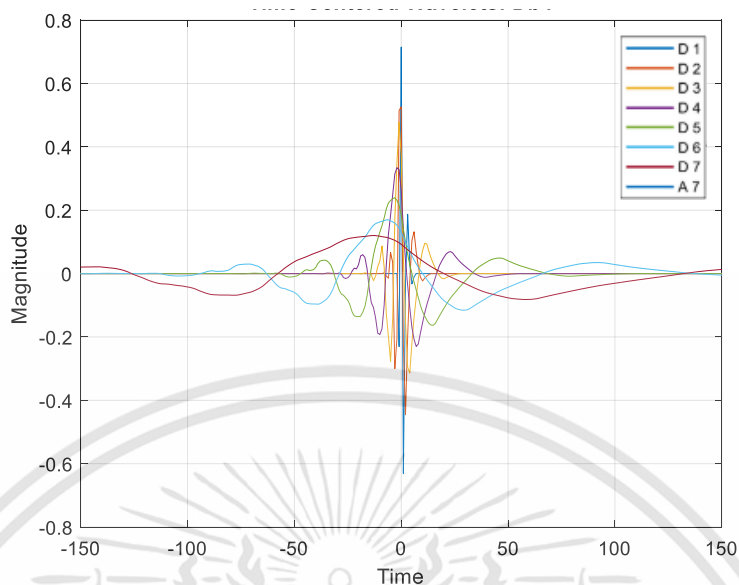
รูปที่ 2.10 เวฟเลตแบบดิสครีตประเภทต่างๆ

DWT Filter Bank ในมิติของความถี่และมิติของเวลาแต่ละชั้นของ Daubechies 4 หรือ Db4 แสดงดังรูปที่ 2.11 และ 2.12 ตามลำดับ



รูปที่ 2.11 เวฟเลต Db4 Filter Bank ในมิติของความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

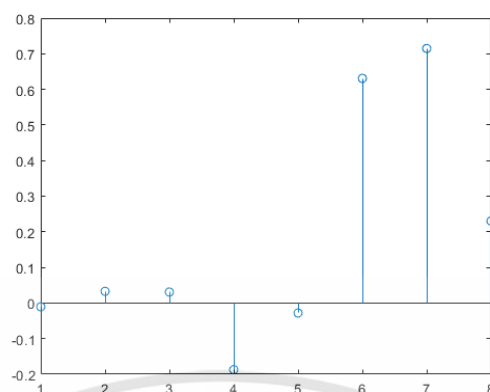


รูปที่ 2.12 เวฟเลต Db4 Filter Bank ในมิติของเวลา

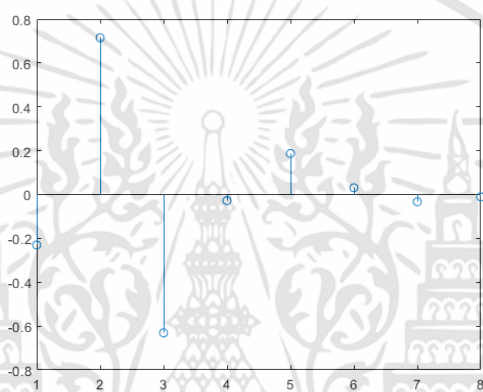
เมื่อนำ Scaling Function ของมาแปลงผ่านความสัมพันธ์ Twin-Scale [26] สามารถสร้าง FIR ฟิลเตอร์สำหรับการแยกสัญญาณและการสร้างคืนสัญญาณ ตัวอย่างฟิลเตอร์กรองความถี่ต่ำและฟิลเตอร์กรองความถี่สูงสำหรับแยกสัญญาณของ Db4 แสดงดังรูปที่ 2.13 และสามารถคำนวณหาจำนวนค่าสัมประสิทธิ์ในแต่ละชั้นได้โดยใช้สมการที่ (2.12)

$$M = \text{floor}\left(\frac{\text{length}(x) - 1}{2}\right) - n \quad (2.12)$$

เมื่อ  $M$  คือ จำนวนสัมประสิทธิ์ A และ D  
 $x$  คือ สัญญาณหรือค่าสัมประสิทธิ์อินพุต  
 $n$  คือ จำนวนที่มีค่าเป็นครึ่งหนึ่งของความยาวฟิลเตอร์



(a)



(b)

รูปที่ 2.13 ตัวอย่างตัวกรองสำหรับแยกสัญญาณของ Db4

(a) ตัวกรองความถี่ต่ำ (b) ตัวกรองความถี่สูง

## 2.5 อัลกอริธึมการเรียนรู้เชิงลึก (Deep Learning Algorithm)

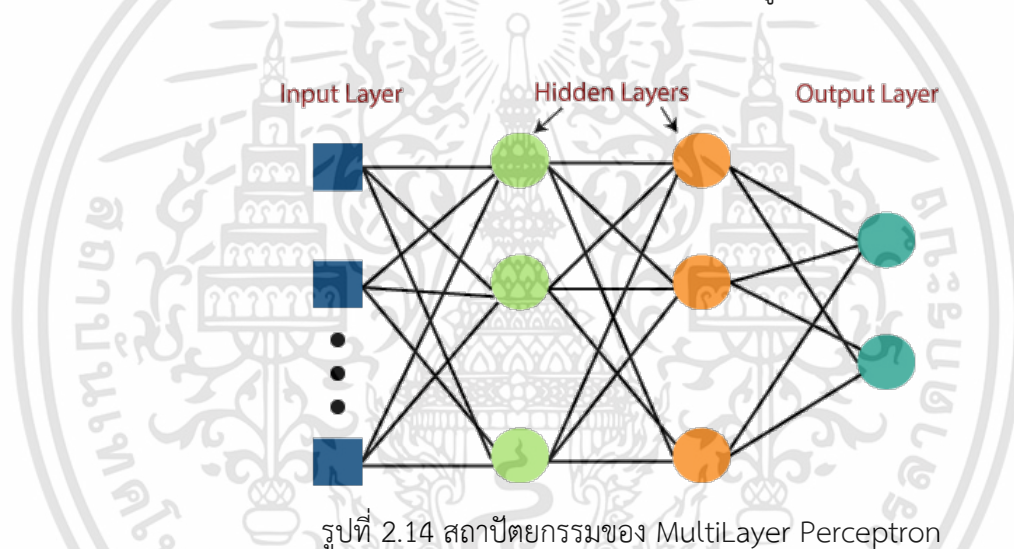
การเรียนรู้เชิงลึก (Deep Learning; DL) [27] คือ ชุดคำสั่งที่สร้างขึ้นมาเพื่อการเรียนรู้ของเครื่องคอมพิวเตอร์โดยเลียนแบบการวิเคราะห์ของสมองมนุษย์ ชุดคำสั่งนี้จะทำให้เครื่องคอมพิวเตอร์สามารถประมวลผลข้อมูลจำนวนมาก สถาปัตยกรรมที่เป็นตัวแทนของการเรียนรู้เชิงลึกประกอบด้วย โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Networks; CNN) โครงข่ายประสาทเทียมแบบกำเนิด (Generative Neural Networks; GNN) โครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Networks; RNN) และ โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Networks; DNN) สำหรับการประยุกต์ใช้ด้าน BCI การเรียนรู้เชิงลึกจะถูกใช้เปรียบเทียบบ่อย่างกว้างๆ กับเทคโนโลยีการเรียนรู้เครื่องจักร (Machine Learning) เป็นหลัก เนื่องจากปัจจุบันส่วนใหญ่งานวิจัยของการเรียนรู้เครื่องมุ่งเน้นไปที่ข้อมูลคงที่ (Static Data) ซึ่งไม่ได้เป็นวิธีการที่ดีในการจัดหมวดหมู่ของสัญญาณสมองที่มีการเปลี่ยนแปลงอย่างรวดเร็ว [28]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.1 สถาปัตยกรรมของ MultiLayer Perceptron (MLP)

สถาปัตยกรรม MultiLayer Perceptron (MLP) หรือโครงข่ายแบบป้อนไปข้างหน้าโดยมีการเรียนรู้แบบย้อนกลับนั้นถูกพัฒนาในปี 1970 โดยปัจจุบันสถาปัตยกรรม Back-Propagation หรือแบบแพร่ย้อนกลับนี้เป็นที่นิยมสูงสุดและยังมีประสิทธิภาพมาก รวมถึงยังมีความเรียบง่ายสำหรับการเป็นต้นแบบสำหรับโครงข่ายประสาทเทียมแบบหลาย Layer

MLP นั้นมีลักษณะต้นแบบคือจะมีจำนวนหนึ่งชั้นอินพุต (Input Layer) หนึ่งชั้นเอาต์พุต (Output Layer) และอย่างน้อยหนึ่งชั้นซ่อน (Hidden Layer) ซึ่งลักษณะของโครงข่ายประสาทเทียม นั้น ไม่มีข้อจำกัดทางทฤษฎีต่อจำนวนของชั้นซ่อน แต่ตามแบบต้นฉบับจะมีเพียงหนึ่งหรือสองชั้นเท่านั้น โดยบางการทำงานเพื่อแก้ปัญหาที่ซับซ้อนจะต้องมีอย่างน้อยที่สุดสี่ชั้น (สามชั้นซ่อนกับหนึ่งชั้นเอาต์พุต) แต่ละชั้นเชื่อมต่อกับชั้นที่ตามมา ตัวอย่างสถาปัตยกรรม MLP แสดงดังรูปที่ 2.14 [29]



### 2.5.2 สถาปัตยกรรมของ LeNet-5

LeNet-5 คือโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ประกอบด้วย 7 Layer ไม่รวม Input Layer ทุก Layer มี Trainable Parameter (Weight) Input Layer ของโมเดลนี้มีขนาด 32x32 พิกเซล Convolution Layer ของโมเดลนี้จะถูกระบุด้วยตัวอักษร Cx, Sub-Sampling Layer ถูกระบุด้วยตัวอักษร Sx และ Fully-Connected Layer ถูกระบุด้วย Fx เมื่อ x คือลำดับของ Layer โดยรายละเอียดของสถาปัตยกรรมแสดงดังรูปที่ 2.15 โดยทุก Layer มี Activation Function เป็น Hyperbolic Tangent Function แต่ใน Output Layer ใช้ Activation Function เป็น Sigmoid Function

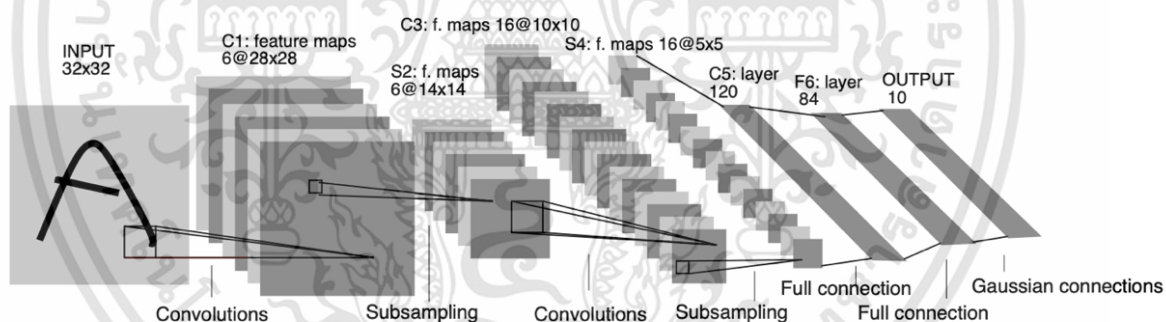
C1 Layer คือ Convolution Layer ที่มี 6 Feature Map ซึ่งแต่ละโหนดในแต่ละ Feature Map จะถูกเชื่อมกับอินพุตใกล้เคียงขนาด 5x5 พิกเซล ขนาดของ Feature Map คือ 28x28 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S2 Layer คือ Sub-Sampling Layer ที่มี 6 Feature Map ขนาด  $14 \times 14$  พิกเซล ซึ่งแต่ละโหนดในแต่ละ Feature Map ถูกเชื่อมกับอินพุตใกล้เคียงขนาด  $2 \times 2$  พิกเซล ตามที่สอดคล้องกับ Feature Map ใน C1 โดย 4 อินพุตโหนดใน S2 จะถูกเพิ่มเข้ามา หลังจากนั้นจะถูกคูณด้วย Trainable Coefficient และถูกบวกด้วย Trainable Bias ผลลัพธ์ที่ได้จะถูกนำไปผ่าน Sigmoid Function ช่องรับขนาด  $2 \times 2$  พิกเซล จะไม่ซ้อนทับกัน ดังนั้น Feature Map ใน S2 จะมีจำนวนของแถวและหลักเป็นครึ่งหนึ่งเมื่อเทียบกับขนาดของ Feature Map ใน C1 เลเยอร์ C3 และ S4 มีรายละเอียดคล้ายกับ C1 และ S2 ตามลำดับ

C5 Layer คือ Convolution Layer ที่มี 120 Feature Map ซึ่งแต่ละโหนดจะถูกเชื่อมกับอินพุตใกล้เคียงทุก 16 Feature Map ใน S4 Layer เนื่องจากขนาดของ S4 เท่ากับ  $5 \times 5$  พิกเซล และขนาดของ Feature Map ใน C5 Layer เท่ากับ  $1 \times 1$  พิกเซล ซึ่งตัวเลข 16 นี้คือจำนวนการเชื่อมต่อที่สมบูรณ์ระหว่าง S4 และ C5 เลเยอร์ C5 ถูกเรียกว่าเป็น Convolution Layer แทนที่จะเป็น Fully-Connected Layer เพราะหากอินพุต LeNet-5 มีขนาดใหญ่ขึ้นโดยที่ทุกอย่างคงที่ มิติของ Feature Map จะมีขนาดใหญ่กว่า  $1 \times 1$  พิกเซล

F6 Layer ประกอบด้วย 84 โหนด และทุกโหนดจะถูกเชื่อมต่อจาก C5 Layer [30]



รูปที่ 2.15 สถาปัตยกรรมของ LeNet-5

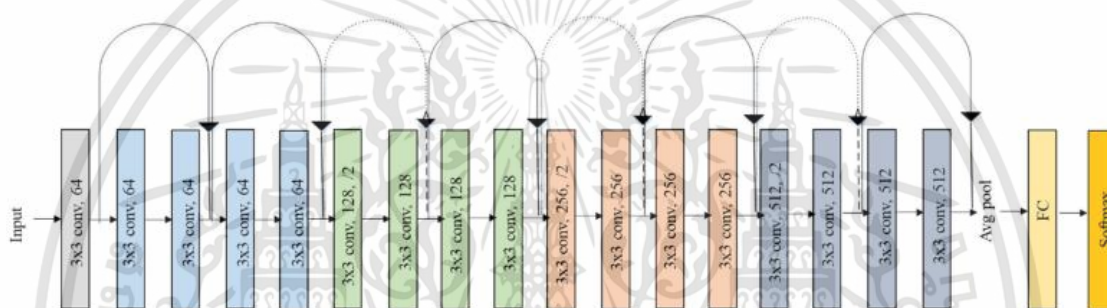
## 2.5.3 สถาปัตยกรรมของ Pre-Trained Model

### 2.5.3.1 สถาปัตยกรรมของ ResNet18

ResNet เป็นโครงสร้างของการเรียนรู้เชิงลึกที่คิดค้นโดย Kaiming He et al. [31] ในปี 2015 ซึ่งเป็นผู้ชนะเลิศครั้งแรกในงาน ImageNet Competition Classification Task เนื่องด้วยมีความเรียบง่ายและสามารถใช้งานได้จริง มีวิธีการสร้างโครงสร้างมากมายโดยใช้พื้นฐานของ ResNet เช่น ResNet50 (มีทั้งหมด 50 Layers) หรือ Resnet101 (มีทั้งหมด 101 Layers) ตัวเลขด้านหลัง ResNet เป็นตัวเลขที่ระบุจำนวน Layers ทั้งหมดของคอนโวลูชันในโครงสร้างนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมของ ResNet18 แสดงในรูปที่ 2.16 โดยจะมีจำนวนชั้น Layer ทั้งหมด 18 Layers (17 Convolution Layers, 1 Fully-Connected Layer และ Softmax Layer ที่ใช้ในการจำแนก) โดย Convolution Layer ใช้ฟิลเตอร์ขนาด 3x3 พิกเซลและโครงข่ายถูกออกแบบมาให้เอาต์พุตของ Feature Map มีขนาดเท่ากับ ดังนั้น Layer จะมีจำนวนเท่ากับจำนวนฟิลเตอร์ อย่างไรก็ตามฟิลเตอร์จะมีขนาดเพิ่มขึ้นสองเท่าถ้าเอาต์พุตของ Feature Map มีจำนวนลดลงครึ่งหนึ่ง การสุ่มแบบลดลงนั้นเกิดขึ้นบน Convolution Layer ที่มีค่า Stride เท่ากับ 2 และสุดท้ายจะมี Average-Pooling ตามด้วย Fully-Connected Layer ที่ใช้ Softmax Layer โดยจะมี Residual Shortcut เชื่อมต่อระหว่าง Layer เพื่อแก้ปัญหาเรื่อง Vanishing Gradient [32]



รูปที่ 2.16 สถาปัตยกรรมของ ResNet18

### 2.5.3.2 สถาปัตยกรรมของ ResNet50

สถาปัตยกรรมของ ResNet50 เหมือนกับสถาปัตยกรรมของ ResNet18 แต่จะมีจำนวนชั้น Layer ที่มากกว่า โดยจะมีจำนวนเท่ากับ 50 Layers (49 Convolution Layers, 1 Fully-Connect Layer และ Softmax Layer ที่ใช้ในการจำแนก)

### 2.5.3.3 สถาปัตยกรรมของ ShuffleNet

ShuffleNet นั้นใช้วิธีการดำเนินการ 2 วิธี คือ Pointwise Group Convolution และ Channel Shuffle เพื่อลดเวลาในการคำนวณ ในขณะที่เดียวกันก็ยังคงรักษาความแม่นยำในการจำแนกได้ โดยโครงข่าย ShuffleNet นั้นประกอบไปด้วย ShuffleNet Units ที่ถูกรวมเป็นกลุ่มทั้งหมดสามช่วง จำนวนของ Bottleneck Channels ถูกกำหนดเป็น  $\frac{1}{4}$  ของจำนวนเอาต์พุต Channel สำหรับในแต่ละ ShuffleNet Unit และ Scale Factor ใช้ในการกำหนดจำนวนของ Channel โดยสถาปัตยกรรมของ ShuffleNet แสดงในตารางที่ 2.2 [33]

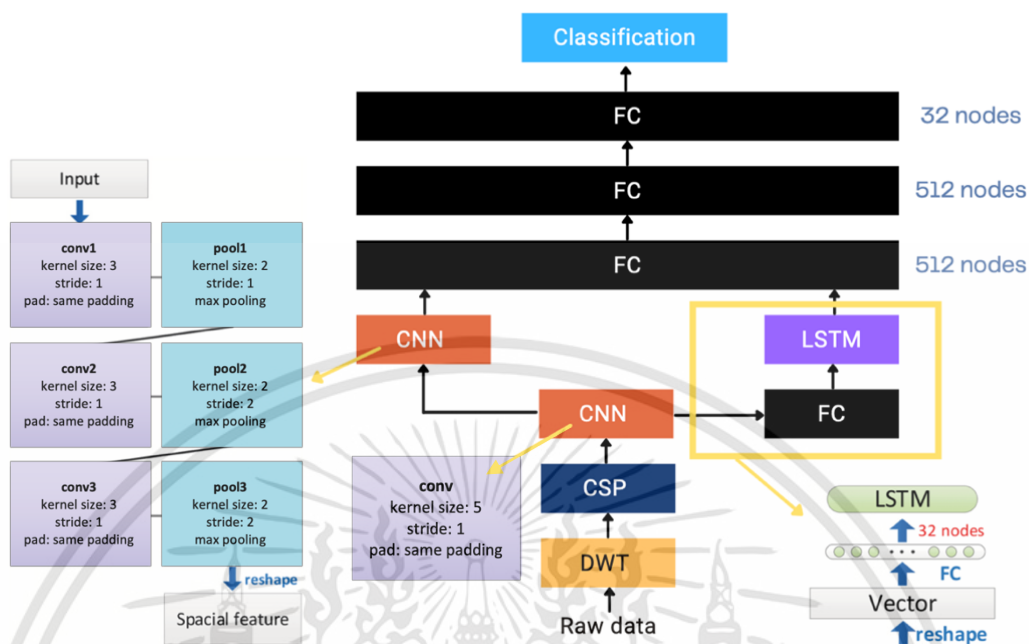
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 สถาปัตยกรรมของ ShuffleNet

Layer	Output Size	K Size	Stride	Repeat	Output Channel (groups)	
					g = 1	g = 4
Image	224 x 224				3	3
Conv1	112 x 112	3 x 3	2			
MaxPool	28 x 28	3 x 3	2	1	24	24
Stage2	14 x 14		2	1	144	272
	14 x 14		1	3	144	272
Stage3	7 x 7		2	1	288	544
	7 x 7		1	7	288	544
Stage4	1 x 1		2	1	576	1088
			1	3	576	1088
GlobalPool	1 x 1	7 x 7				
FC					1000	1000

#### 2.5.4 สถาปัตยกรรมของโครงข่ายประสาทเทียมเชิงลึกแบบผสม (Hybrid Deep Neural Network; HDNN)

สถาปัตยกรรมของโครงข่ายประสาทเทียมเชิงลึกแบบผสม [30] แสดงดังรูปที่ 2.17 โดยโครงสร้างของ Convolution Neural Network (CNN) และ Long Short-Term Memory (LSTM) และ Fully-Connected (FC) จะกล่าวถึงในหัวข้อที่ 2.5.4.1-2.5.4.3 ตามลำดับ

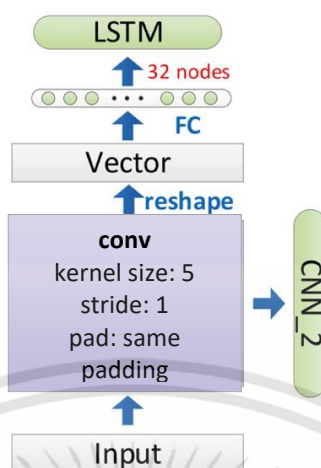


รูปที่ 2.17 สถาปัตยกรรมของโครงข่ายประสาทเทียมเชิงลึกแบบผสม

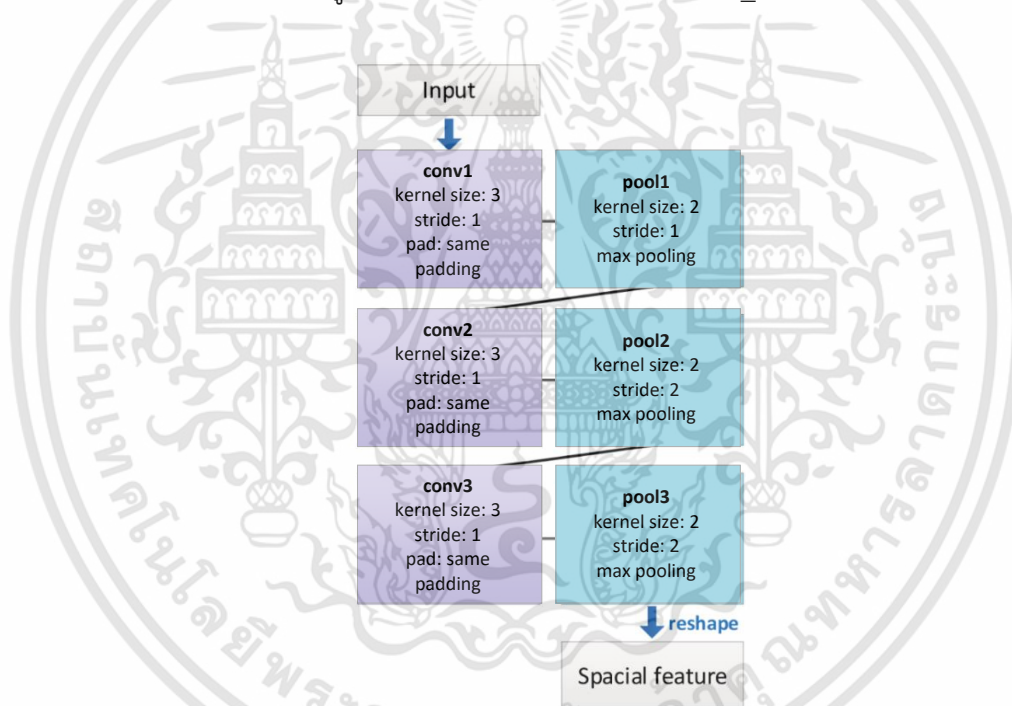
#### 2.5.4.1 Convolution Neural Network (CNN)

HDNN ที่นำเสนอมี CNN ที่ประกอบด้วย 2 Sub-CNNs CNN\_1 และ CNN\_2 CNN\_1 เป็น Mono Layer ที่มี 1 Convolution Layer และ FC Layer แสดงดังรูปที่ 2.18 ซึ่งพารามิเตอร์ของ 4 CNN\_1 มีค่าเหมือนกัน CNN\_2 มี 3 Hidden Layer ซึ่งแสดงรายละเอียดดังรูปที่ 2.19 ในแต่ละ Convolution Layer จะมี Convolution Kernel ขนาด 3x3 พิกเซล และ Activation Function เป็น Rectified Linear Unit (ReLU) ซึ่งถูกใช้เพื่อสกัดคุณลักษณะเชิงพื้นที่ของสัญญาณ MI

Max Pooling Layer มีขนาดของ Kernel คือ 2x2 พิกเซล ซึ่งถูกใช้เพื่อลดขนาดของ Feature Map ของเอาต์พุตแต่ละ Convolution Layer นอกจากนี้เทคนิค Zero-Padding ถูกนำไปใช้ใน Convolution Layer เพื่อให้แน่ใจว่าขนาดของเอาต์พุตจะสอดคล้องกับอินพุตและหลีกเลี่ยงการสูญเสียข้อมูลของ Spatial Feature Map [34]



รูปที่ 2.18 โครงสร้างของโมเดล CNN\_1



รูปที่ 2.19 โครงสร้างของโมเดล CNN\_2 ซึ่งประกอบด้วย 3 Convolution Layer และ 3 Max Pooling Layer

#### 2.5.4.2 Long Short-Term Memory (LSTM)

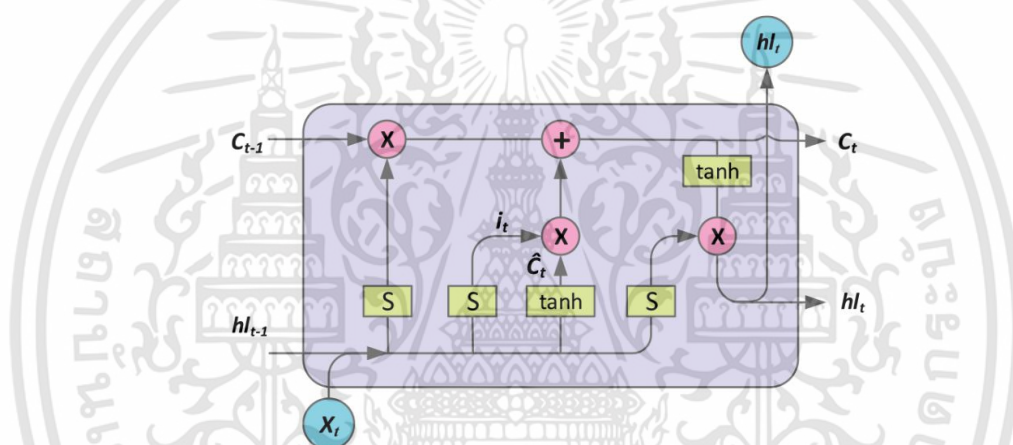
การดึงคุณลักษณะเชิงเวลา (Temporal Feature) ของสัญญาณ EEG มีความสำคัญพอๆ กับคุณลักษณะเชิงพื้นที่ (Spatial Feature) LSTM ซึ่งเป็นโครงข่ายประสาทของ Recurrent Neural Network (RNN) เป็นวิธีที่นิยมในการแสดงให้เห็นถึง Internal Temporal Correlation ของสัญญาณอนุกรมทางเวลา ดังนั้นจึงมีการประยุกต์ใช้ LSTM Neural Network ควบคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

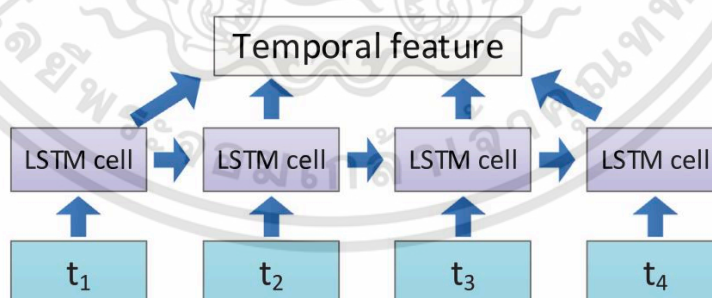
ไปกับ CNN เพื่อดึง Temporal Feature เซลล์ LSTM เป็นหน่วยพื้นฐานของ LSTM ในการประมวลผลสัญญาณอินพุตของทุกๆ ขั้นตอน ดังแสดงในรูปที่ 2.20 โดยที่

- **S** หมายถึง Sigmoid Activation Function
- **tanh** หมายถึง Hyperbolic Tangent Activation Function
- **+** หมายถึง บวก
- **x** หมายถึง การคูณ

เอาต์พุตของทุกขั้นตอนใน CNN\_1 จะถูกป้อนเป็นสัญญาณอินพุตเข้าสู่เซลล์ LSTM ตามลำดับเพื่อให้ได้ Temporal Feature ดังแสดงในรูปที่ 2.20 พารามิเตอร์ของทั้ง 4 เซลล์ LSTM เหมือนกันและเฟรมเวิร์คของเซลล์ LSTM มีโครงสร้างดังแสดงในรูปที่ 2.21



รูปที่ 2.20 เฟรมเวิร์คของเซลล์ LSTM โดย  $C_t$  แสดงถึงสถานะของเซลล์ LSTM ที่โมเมนต์ปัจจุบัน



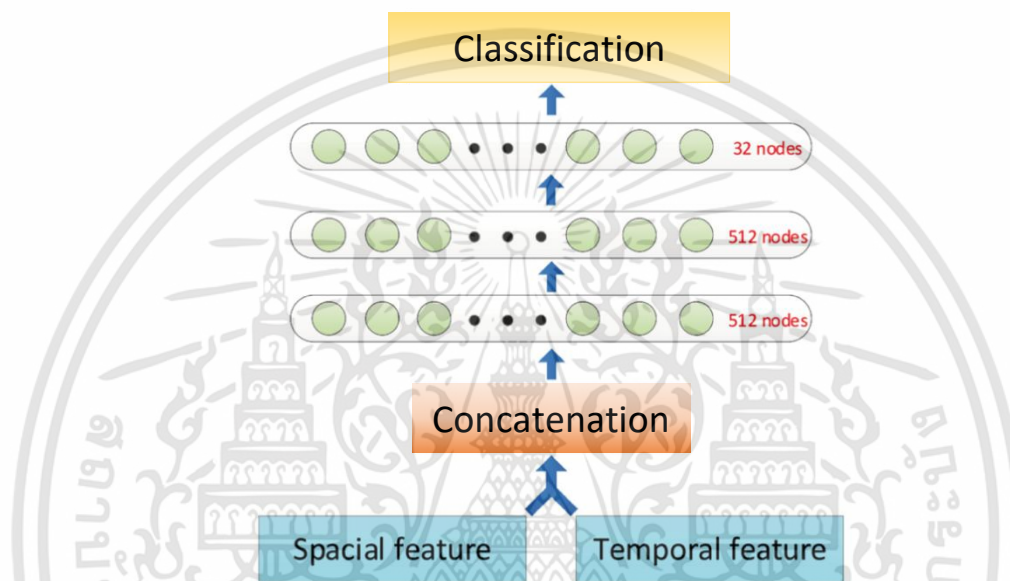
รูปที่ 2.21 Long Short-Term Memory (LSTM) Network ที่ประกอบด้วยเซลล์ LSTM จำนวน 4 เซลล์

#### 2.5.4.3 Fully-Connected Neural Network สำหรับ Transfer Learning

หลังจากดึงคุณลักษณะของ Spatial Feature และ Temporal Feature โดย

ใช้ CNN และ LSTM ของสัญญาณ MI จะใช้ Fully-Connected (FC) Neural Network เพื่อวิเคราะห์เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติที่ได้รับมาโดยใช้การสังเคราะห์และจำแนกประเภทผลลัพธ์ โครงสร้างของ FC แสดงดังรูปที่ 2.22 ซึ่งประกอบไปด้วย 3 Hidden Layer แต่ละ Hidden Node มี 512, 512, และ 32 โหนด ตามลำดับ FC ถูกนำมาใช้โดย Feed-Forward Neural Network (FNN) 3 Layers ใช้ ReLU เป็น Activation Function ในแต่ละ Hidden Layer และใช้ Softmax Function เพื่อแสดงการกระจายของความน่าจะเป็นแบบเลขชี้กำลังระหว่างคลาสต่างๆ ใน Output Layer [34]



รูปที่ 2.22 โครงสร้างของ Fully-Connected (FC) Neural Network สำหรับ Transfer Learning

## 2.6 เฟรมเวิร์กสำหรับการเรียนรู้เชิงลึก (Deep Learning Framework)

### 2.6.1 TensorFlow

TensorFlow คือไลบรารีแบบ Open Source ที่ใช้ในการฝึกสอนและพัฒนาโมเดลการเรียนรู้ของเครื่องจักร TensorFlow ถูกใช้เพื่อสร้างกราฟการผ่านของข้อมูลโดยมีกระบวนการเกิดขึ้นในแต่ละโหนด โดยโหนดเหล่านี้เป็นการดำเนินการทางคณิตศาสตร์ และการเชื่อมต่อแสดงถึง Tensor หรืออาร์เรย์ข้อมูลหลายมิติ กระบวนการเหล่านี้เป็นกระบวนการที่มีการเรียนรู้ของเน็ตเวิร์กเกิดขึ้น โดยกระบวนการทางคณิตศาสตร์นี้เกิดขึ้นบน C++ ซึ่งเป็นภาษาที่มีประสิทธิภาพและเป็นไบนารีความสามารถสูง โดยไพธอนเป็นเพียงภาษาที่เชื่อมต่อทุกอย่างเข้าด้วยกัน [35]

### 2.6.2 Keras

Keras คือไลบรารีการเรียนรู้เชิงลึกที่มีประสิทธิภาพสูงซึ่งดำเนินการอยู่บนไลบรารีการเรียนรู้ของเครื่องแบบ Open Source ต่างๆ เช่น TensorFlow และ Keras นั้นยังเป็นไลบรารีแบบ Open Source เช่นเดียวกัน Keras จะไม่รองรับการคำนวณระดับต่ำ แต่จะส่งไปยังไลบรารีอื่นแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Keras มีวัตถุประสงค์เพื่อสร้างไลบรารีการเรียนรู้ของเครื่องโดยสามารถทำงานแยกและสามารถทำงานร่วมกับเฟรมเวิร์คอื่นได้เช่นกัน [35]

## 2.7 อุปกรณ์ Unicorn Hybrid Black

Unicorn Hybrid Black เป็นอุปกรณ์บันทึกข้อมูล EEG แบบสวมหัว 8 ช่องสัญญาณ (Channel) ที่มีคุณภาพสูง สำหรับการใช้งานแอปพลิเคชันในด้าน BCI เพื่อวัดสัญญาณคลื่นไฟฟ้าจากสมอง [36] ชุดอุปกรณ์ตามรูปที่ 2.23 ประกอบไปด้วย



รูปที่ 2.23 Unicorn Hybrid Black แบบสวมหัว 8 Channel

- Unicorn Brain Interface Hybrid Black: มี 8 Channel, Sampling Rate 250 Hz, 24 Bit และใช้ระบบในการส่งสัญญาณแบบ Bluetooth
- Unicorn Hybrid EEG Electrodes: มีทั้งหมด 8 ชิ้นสำหรับการบันทึกคลื่นสัญญาณแบบแห้งหรือบันทึกคลื่นสัญญาณแบบเปียกโดยใช้ Unicorn Gel
- Unicorn Sticky Electrodes: มีทั้งหมด 50 ชิ้น โดยจะใช้แปะที่หลังหูทั้งฝั่งซ้ายและขวา
- Unicorn Cap: หมวกที่ใช้แปะ Electrode ไซส์ M
- Unicorn USB Charging Cable: โดยใช้เป็น Micro USB
- Unicorn Bluetooth Dongle: อุปกรณ์ที่ใช้รับข้อมูลจาก Unicorn Brain Interface มายังคอมพิวเตอร์
- Unicorn Suite Hybrid Black Software Environment: ประกอบไปด้วยโปรแกรม Recorder, C API, .NET API และแอปพลิเคชัน Dev Tools

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8 Lab Streaming Layer (LSL)

Lab Streaming Layer (LSL) คือระบบสำหรับรวบรวมข้อมูลอนุกรมเวลา (Time Series) ในการทดลองเชิงวิจัยที่สามารถใช้ได้ทั้งกับ Networking, Time-Synchronization และ (Near-) Real-Time Access เช่นเดียวกันกับการรวบรวมเข้าสู่ศูนย์กลาง, การแสดงผลข้อมูลและการบันทึกข้อมูลลงหน่วยความจำ [37]

LSL Distribution ประกอบด้วยไลบรารีหลักและชุดของเครื่องมือที่สร้างขึ้นบนไลบรารีไลบรารีหลักที่ใช้ในการส่งข้อมูลคือ liblsl และอินเทอร์เฟซภาษาของไลบรารี (C, C++, Python, Java, C#, MATLAB) ไลบรารีสามารถใช้ได้ในหลากหลายวัตถุประสงค์และใช้ได้หลากหลายแพลตฟอร์ม เช่น Window / Linux / MacOS / Android / iOS; Architecture Support: x86 / amd64 / arm

ชุดของเครื่องมือประกอบไปด้วยโปรแกรมที่ใช้ในการบันทึก, ตัวนำเข้าไฟล์และแอปพลิเคชันที่สร้างข้อมูลจากอุปกรณ์ที่สามารถใช้ได้บน Lab Network เช่น เสียง, EEG, หรือการตรวจจับการเคลื่อนไหว

### 2.8.1 Streaming Layer API

ไลบรารี liblsl จัดเตรียม Abstractions เหล่านี้ให้สำหรับโปรแกรมผู้ใช้

- Sample: การวัดของทุก Channel จากอุปกรณ์เรียกว่า Sample
- Chunk: Sample สามารถแลกเปลี่ยนด้วยตัวเองเพื่อพัฒนา Latency หรือการส่งหลายๆ Sample เพื่อพัฒนา Throughput
- Metadata: นอกจากข้อมูลดิบ ข้อมูลสำหรับการ Stream นั้นถูกจัดเก็บและส่งในรูปแบบของ XML
- Stream: การรวบรวมของข้อมูลสุ่มจากอุปกรณ์ด้วย Metadata เรียกว่า Stream โดย Stream นั้นจะมีอัตราการสุ่มอย่างปกติ (เช่น ข้อมูลเสียงจะถูกสุ่มที่ 44100 Hz, ข้อมูลวิดีโอจะมีอัตราเฟรมที่ 24 หรือ 60 Hz) หรืออัตราการสุ่มอย่างไม่ปกติ (เช่น การกดปุ่ม, อีเว้นท์ของการทดลอง) และ Channel เดี่ยวหรือมากกว่า (เช่น 2 Channel สำหรับไฟล์เสียงแบบสเตอริโอ, 32 / 64 Channel สำหรับการบันทึกข้อมูล EEG, 1920x1080 Channel สำหรับการจับหน้าจอแบบ FULL HD) ข้อมูลทุกอย่างภายใน Stream ต้องเป็นข้อมูลประเภทเดียวกัน (Integers, Floats, Doubles, Strings)
- Stream Outlet: เพื่อสร้าง Stream ข้อมูลอนุกรมเวลาบน Lab Network ข้อมูลจะถูกส่งแบบ Sample-By-Sample หรือ Chunk-By-Chunk ไปยัง Outlet โดยในการสร้าง Outlet นั้น Stream จะถูกทำให้ตรวจพบได้กับกลุ่มของคอมพิวเตอร์ โดยสามารถติดตามได้โดยค้นหาผ่าน Resolver และเชื่อมต่อกับ Stream Inlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Stream Inlet: ใช้สำหรับการรับข้อมูลอนุกรมเวลาจาก Single Connected Outlet โดย Stream Inlet จะอนุญาตให้ดึงข้อมูล Samples จาก Stream นอกจาก Samples แล้วยังสามารถดึงข้อมูลจาก Metadata ได้
- Resolver: LSL มีการจัดเตรียมฟังก์ชันที่ใช้สำหรับการจัดการ Streams ที่ถูกแสดงบน Lab Network โดยสอดคล้องกับ Content-Based Queries โดย Service Discovery Features จะไม่ขึ้นอยู่กับ Service ภายใน เช่น Zeroconf และมีวัตถุประสงค์เพื่อลดความซับซ้อนของการตั้งค่าเครือข่ายการรวบรวมข้อมูล
- Built-In Clock: อนุญาตให้ประทับเวลา Sample ที่ส่งเพื่อให้สามารถซิงโครไนซ์ร่วมกันได้

### 2.8.2 ความน่าเชื่อถือ (Reliability)

ฟีเจอร์ความน่าเชื่อถือต่อไปนี้ถูกใช้โดยไลบรารี

- การส่งข้อมูลมีความน่าเชื่อถือแบบเดียวกับ TCP เป็น Message-Oriented และ Type Safe
- ไลบรารีจัดเตรียมการฟื้นฟูความผิดพลาดแบบอัตโนมัติจากแอปพลิเคชันหรือการ Crash ของคอมพิวเตอร์เพื่อลดความสูญเสียของข้อมูลให้น้อยที่สุด โดยสิ่งนี้สามารถแทนที่คอมพิวเตอร์ระหว่างการบันทึกข้อมูลโดยไม่จำเป็นต้องเริ่มต้น Data Collection ใหม่
- ข้อมูลจะถูก Buffer ทั้งฝั่งส่งและฝั่งรับเพื่อทนต่อความล้มเหลวของเครือข่ายเป็นระยะ
- การส่งข้อมูลเป็นแบบ Type Safe และรองรับการแปลงประเภทตามความจำเป็น

### 2.8.3 Time Synchronization

LSL ใช้หลักการ Built-in Time Synchronization กับทุกข้อมูลที่ถูกบันทึกโดยมีจุดประสงค์ในการส่งข้อมูลที่แม่นยำในระดับมิลลิวินาทีบนเครือข่ายคอมพิวเตอร์ Local โดย Built-in Time Synchronization ถูกออกแบบหลังจากมีการใช้อย่างกว้างขวางของ Network Time Protocol (NTP) และถูกนำไปใช้ในไลบรารี LSL

### 2.8.4 File Format

Transport API ไม่ได้รับรองหรือจัดเตรียมรูปแบบไฟล์เฉพาะ แต่เป็นโปรแกรมบันทึกที่ให้มา (LabRecorder) โดยบันทึกไปยังรูปแบบไฟล์ XDF (Extensible Data Format) XDF ถูกออกแบบพร้อมกับเลย์เออร์การสตรีมแล็บและรองรับชุดฟีเจอร์เต็มรูปแบบ LSL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9 OpenViBE

OpenViBE คือแพลตฟอร์มซอฟต์แวร์ที่สามารถออกแบบ ทดสอบและใช้งาน BCI โดยสามารถใช้สำหรับการรับ, ฟิลเตอร์, ประมวลผล, จำแนกและแสดงคลื่นไฟฟ้าจากสมองแบบ Real-Time โดย OpenViBE นั้นเป็นแพลตฟอร์มที่สามารถใช้งานได้ทั้งบนระบบปฏิบัติการ Windows และ Linux [38]

OpenViBE นั้นถูกพัฒนาบางส่วนให้สามารถส่งข้อมูลระหว่างแอปพลิเคชัน LSL-Aware อื่นๆ ได้ เช่น LSL สามารถถูกใช้เพื่อนำสัญญาณจากหลายหลากอุปกรณ์ที่ไม่ถูกสนับสนุนเข้าสู่ OpenViBE ได้ เพียงแค่ส่งข้อมูลจากอุปกรณ์ไปยัง LSL Stream หรือสามารถส่งข้อมูลจาก OpenViBE ไปยังแอปพลิเคชัน LSL-Aware อื่นๆ [39]

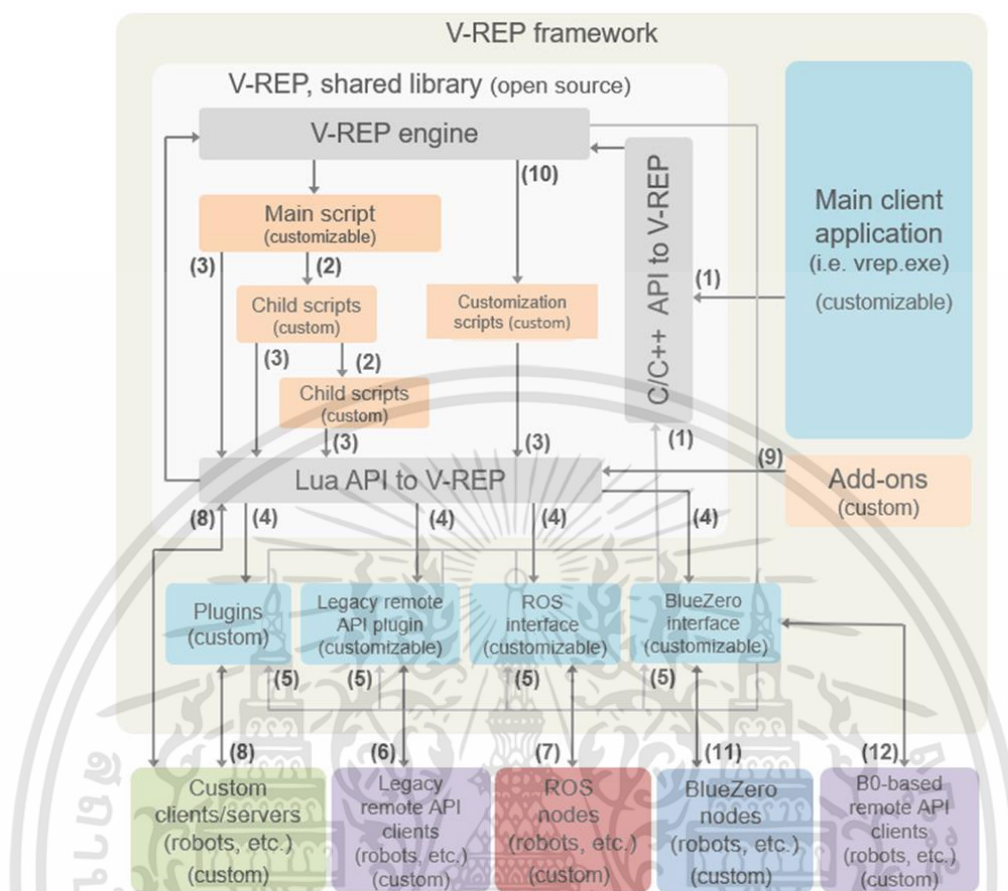
## 2.10 CoppeliaSim

โปรแกรม CoppeliaSim ถูกพัฒนาต่อจากโปรแกรม V-REP โดย CoppeliaSim เป็นโปรแกรมจำลองหุ่นยนต์ที่มาพร้อมกับสภาพแวดล้อมการพัฒนาแบบบูรณาการที่ถูกอ้างอิงตามสถาปัตยกรรมการควบคุมแบบกระจาย โดยแต่ละ Object หรือโมเดลสามารถควบคุมแบบแยกกันผ่าน Embedded Script, Plug-In, ROS Node, Remote API Client หรือกำหนดโซลูชันเอง ซึ่งการทำงานในแบบที่กล่าวไป ช่วยให้ CoppeliaSim ใช้งานได้หลากหลายและเหมาะสมสำหรับแอปพลิเคชันที่ใช้หลายหุ่นยนต์ Controller สามารถเขียนได้ทั้งภาษา C/C++, Python, Java, Lua, MATLAB หรือ Octave

โปรแกรม CoppeliaSim ถูกใช้สำหรับการพัฒนาอัลกอริธึมที่รวดเร็ว การจำลองการทำงานอัตโนมัติของโรงงาน การสร้างต้นแบบและตรวจสอบอย่างรวดเร็ว การศึกษาที่เกี่ยวข้องกับจำพวกหุ่นยนต์ การควบคุมระยะไกล การตรวจสอบความปลอดภัยแบบสองชั้นซึ่งเป็น Digital Twin และอื่นๆ [40]

### 2.10.1 การเชื่อมต่อระหว่าง MATLAB และ CoppeliaSim

สถาปัตยกรรมซอฟต์แวร์ CoppeliaSim แสดงดังรูปที่ 2.24 [41] โดยโปรแกรม CoppeliaSim จะทำการโต้ตอบกับระบบการคำนวณจำลองผ่านคำสั่งของภาษา Lua ที่มีการสร้างไว้ภายใน CoppeliaSim และหลังจากนั้นจะเป็นการเขียนฟังก์ชันในการควบคุมโมเดลหุ่นยนต์จำลอง ในขณะเดียวกัน Remote API จะโต้ตอบกับ Lua API ภายในซึ่งจะทำให้โปรแกรมภายนอกอื่นๆ สามารถควบคุมโมเดลหุ่นยนต์ในโปรแกรม CoppeliaSim ได้ผ่าน Remote API โดยจะทำการโต้ตอบผ่าน Socket



รูปที่ 2.24 สถาปัตยกรรมซอฟต์แวร์ CoppeliaSim

ไฟล์ที่จำเป็นที่ใช้ในการควบคุม CoppeliaSim ผ่านโปรแกรม MATLAB มีดังนี้

1. remoteApiProto.m
2. remApi.m
3. remoteApi.dll

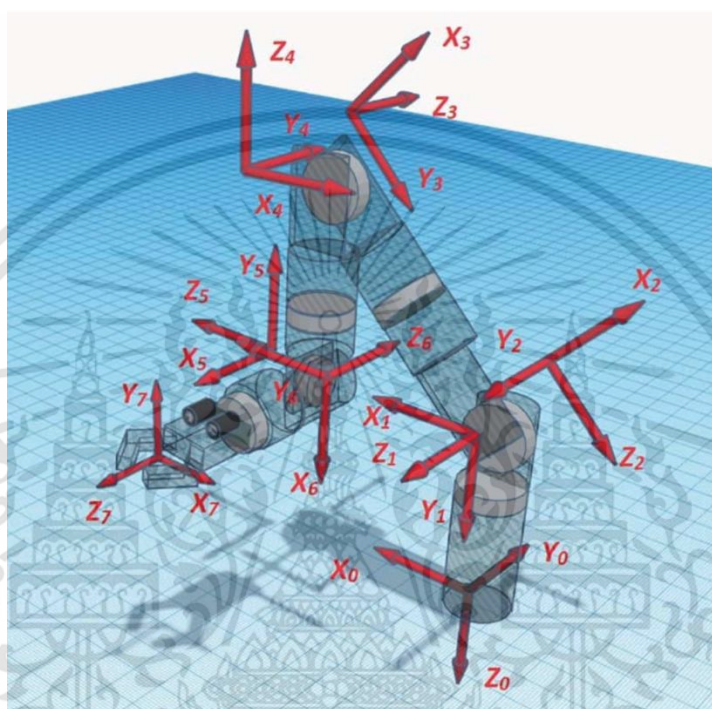
โดยไฟล์เหล่านี้ถูกจัดเก็บไว้ในโฟลเดอร์ของ CoppeliaSim และต้องทำการคัดลอกไปยังโฟลเดอร์เดียวกับโมเดลหุ่นยนต์และโปรแกรม MATLAB

## 2.11 แขนกลจำลอง

จลศาสตร์หุ่นยนต์หมายถึงการศึกษาเชิงวิเคราะห์ในการเคลื่อนที่ของหุ่นยนต์จัดการ (Robot Manipulator) เช่น การศึกษาโซ่จลศาสตร์ (Kinematic Chain) ที่มีการสร้างโครงสร้างมาจากระบบหุ่นยนต์ (Robotic System) สมการจลศาสตร์จะ Map พารามิเตอร์ของข้อต่อ (Joint Parameter) กับการกำหนดค่าของระบบหุ่นยนต์ จลศาสตร์ทางตรง (Forward Kinematic) จะใช้สมการของจลศาสตร์เพื่อคำนวณตำแหน่งของ End-Effector จากค่าที่ถูกระบุของพารามิเตอร์ของข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อ แต่หากพารามิเตอร์ของข้อต่อถูกคำนวณสำหรับตำแหน่งที่ระบุของ End-Effector จะเรียกอีกแบบว่า จลศาสตร์ผกผัน (Inverse Kinematic) ในเนื้อหาต่อไปนี้จะกล่าวถึงการวิเคราะห์ของจลศาสตร์ทางตรงของแขนกล Jaco Arm 2 ซึ่งเริ่มต้นพิกัดแกนจะถูกวางบนแต่ละข้อต่อของแขนกลดังรูปที่ 2.25 ตามข้อกำหนดของผู้ผลิต [42]



รูปที่ 2.25 ตำแหน่งบนแกนบนแขนของหุ่นยนต์

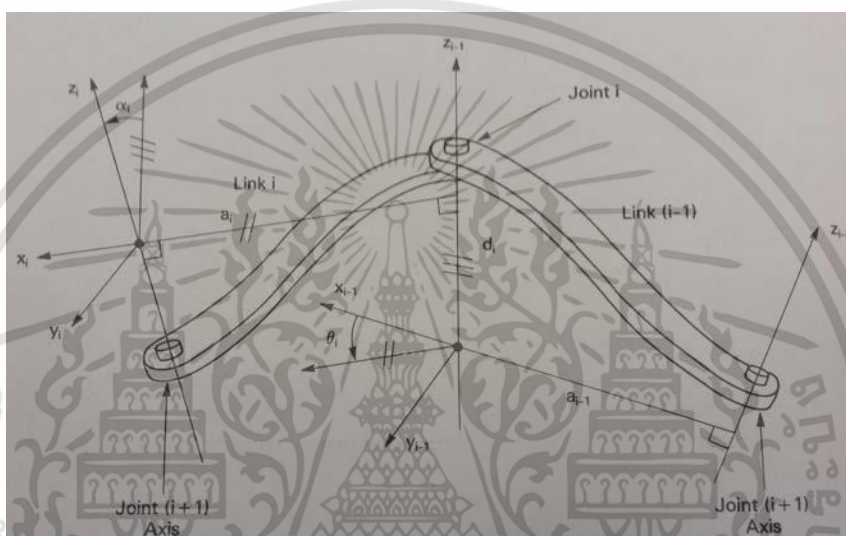
### 2.11.1 Denavit-Hartenberg (DH) Parameter

Denavit-Hartenberg (DH) Parameter เป็นพารามิเตอร์ที่ใช้กำหนดการเคลื่อนที่สำหรับการเคลื่อนที่แบบจลศาสตร์ทางตรงซึ่งอีกนัยความหนึ่งคือเป็นการจำลองพฤติกรรมของหุ่นยนต์ [43] ตามคำจำกัดความของพารามิเตอร์ DH นั้นได้อธิบายว่าเป็นการอธิบายตำแหน่งหรือทิศทางของลิงก์ (Link) และข้อต่อ (Joint) ที่ประกอบรวมกันเป็นแขนกล พารามิเตอร์ DH ถูกนำไปใช้ในกับโพลีจลศาสตร์ของแขนซึ่งลิงก์จะถูกเชื่อมต่อกับข้อต่อและขับเคลื่อนโดยใช้ Acuator การเชื่อมต่อของลิงก์กับข้อต่อจะเรียกว่า Degree of Freedom (DOF) เพราะฉะนั้นหากหุ่นยนต์นั้นเป็นแบบ 6-DOF นั้นหมายถึงมี 6 ลิงก์และ 6 ข้อต่อ ข้อต่อนั้นจะมีการกำหนดหมายเลขแบบเรียงลำดับโดยเริ่มจากฐานจนถึง End-Effector แต่ส่วนของฐาน (ลิงก์ 0) จะไม่ได้นับเป็น DOF เนื่องจากลิงก์ดังกล่าวเป็นตัวรักษาความสัมพันธ์ที่คงที่ระหว่างข้อต่อ [43]

ตัวอย่างพิกัดของลิงก์ DH และพารามิเตอร์ของข้อต่อแสดงดังรูปที่ 2.26 [43] ซึ่งมีรายละเอียดของโครงสร้างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนของโครงสร้างลิงก์
  - $a_i$  : ความยาวของลิงก์
  - $\alpha_i$  : การหมุนของลิงก์
2. ส่วนของตำแหน่งสัมพัทธ์ระหว่างลิงก์
  - $d_i$  : ระยะห่างระหว่างแกน x และแกน z ปกติของสองข้อต่อ
  - $\theta_i$  : มุมของระหว่างแกน x ของสองข้อต่อ



รูปที่ 2.26 เฟรมของพิกัดลิงก์และพารามิเตอร์ข้อต่อ

พารามิเตอร์ DH ถูกใช้เพื่อเชื่อมโยงระหว่างลิงก์ปัจจุบันและลิงก์ข้างเคียงและเพื่อระบุการจับคู่ของแต่ละลิงก์กับข้อต่อ หลังจากที่เฟรมอ้างอิงของแต่ละลิงก์ถูกกำหนดแล้ว พารามิเตอร์ที่กล่าวถึงเหล่านั้นจะใช้เมทริกซ์ในการแปลงเพื่อเชื่อมเฟรมต่อไปตามลำดับโดยที่หลักการสร้างเฟรมพิกัดของลิงก์มี 3 กฎ คือ

1. แกน  $z_{i-1}$  จะอยู่ตามแนวแกนของการเคลื่อนที่ของข้อต่อที่  $i^{th}$
2. แกน  $x_i$  จะเป็นปกติสำหรับแกน  $z_{i-1}$  ที่มุ่งไปยังแกน  $z_i$
3. แกน  $y_i$  ถูกกำหนดโดย Cross Product ของ  $z_i$  และ  $x_i$  แล้วทั้งสามแกนจะถูกฟอร์มให้เป็นไปตามกฎมือขวา (Right-Hand System)

เมื่อมีการกำหนดพารามิเตอร์ลิงก์และเฟรมอ้างอิงแล้ว ก็จะสามารถเชื่อมโยงเฟรมที่  $i-1$  ถึง  $i$  ได้อย่างต่อเนื่องกันผ่านการแปลงและการหมุนเพื่อสร้างความสัมพันธ์กันระหว่างสองเฟรมอ้างอิง ซึ่งสมการที่ (2.13) เป็นตัวอย่างของหลักการคำนวณเมทริกซ์  $A$  [43]

$$\begin{aligned}
A_{(i-1),i} &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \\
&= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)
\end{aligned}$$

เมทริกซ์  $A$  นั้นแปลงจากข้อต่อหนึ่งไปอีกข้อต่อโดยการหมุน (Rotation) รอบ  $z$  แปลง (Translation) ตาม  $d$  พารามิเตอร์อื่นแปลงตาม  $a$  และสุดท้ายหมุนรอบแกน  $x$  ซึ่งเมื่อพิจารณาเมทริกซ์จากสมการที่ (2.13)  $a_i$  และ  $\alpha_i$  เป็นค่าคงที่ ในขณะที่  $d_i$  และ  $\theta_i$  คือตัวแปรร่วม, ส่วนของกรอบสีฟ้าหมายถึงการหมุน, กรอบสีส้มหมายถึงการแปลง, กรอบสีแดงคือ Perspective Transform และกรอบสีเขียวหมายถึง Scaling Factor สุดท้ายเมื่อสามารถสร้างเฟรมพิกัดและทราบค่าของ DH เมทริกซ์จะสามารถกำหนดตำแหน่ง End-Effector ของ Jaco Arm ที่สัมพันธ์กับฐานได้ดังสมการที่ (2.14) [43]

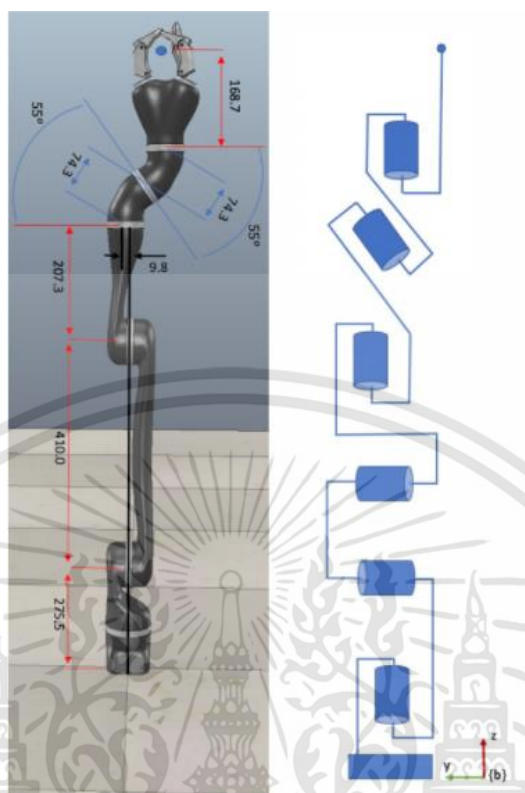
$$T = A_{0n} = A_{01}A_{12}A_{23}\cdots A_{(n-1)n} \quad (2.14)$$

เนื่องจาก Kinova Jaco Arm เป็นแบบ 7-DOF ดังนั้นเมทริกซ์การแปลงโดยรวมจะเป็นดังสมการที่ (2.15) [42]

$$T = A_{01}A_{12}A_{23}A_{34}A_{45}A_{56}A_{67} \quad (2.15)$$

### 2.11.2 แขนกลจำลอง Jaco Arm 2

แขนกลจำลองที่ผู้จัดทำเลือกใช้ในปริญญาณิพนธ์คือ Jaco Arm 2 โดย Kinova ซึ่ง Jaco Arm เป็นหุ่นยนต์ส่วนบุคคลที่นิยมใช้สำหรับงานวิจัยและการฟื้นฟูรักษาผู้ป่วย มีทั้งหมด 6 แขนและ 3 นิ้วมือ สามารถควบคุมได้ 3 แขน และมีเซนเซอร์ที่เกี่ยวข้องคือ เซนเซอร์วัดแรง เซนเซอร์วัดตำแหน่ง และเซนเซอร์วัดความเร็ว แขนกลจำลอง Jaco Arm 2 แสดงดังรูปที่ 2.25 [44] โดยตัวหุ่นยนต์เป็นแบบ 7 Degree of Freedom (DOF)



รูปที่ 2.27 แขนกลจำลอง Jaco Arm 2

พารามิเตอร์ที่อธิบายถึงการเคลื่อนที่ของหุ่นยนต์ของ Jaco Arm 2 แสดงดังตารางที่ 2.3 [45] โดยที่ความยาวของข้อต่อ  $d$  จะขึ้นอยู่กับข้อกำหนดของผู้ผลิต และมุมของข้อต่อจะถูกคำนวณตามเวลาจริงจากการเคลื่อนไหวของแขนกล [42]

ตารางที่ 2.3 DH Parameter

joint	$a$ (m)	$\alpha$ ( $^{\circ}$ )	$d$ (m)	$\theta$ ( $^{\circ}$ )
1	0	$90^{\circ}$	0.2755	$\theta_1$
2	0	$90^{\circ}$	0	$\theta_2$
3	0	$90^{\circ}$	-0.410	$\theta_3$
4	0	$90^{\circ}$	-0.0098	$\theta_4$
5	0	$90^{\circ}$	-0.3111	$\theta_5$
6	0	$90^{\circ}$	0	$\theta_6$
7	0	$0^{\circ}$	0.2638	$\theta_7$

ตารางที่ 2.4 และ 2.5 แสดงการกำหนดค่าของ  $\theta$  ในแนวตั้งของ Kinova Convention และ DH Convention ตามลำดับ โดยการแปลงพิกัดจาก Kinova เป็น DH convention หรือสลับกันจะต้องเปลี่ยน  $\theta$  ตามตารางที่ 2.6 และ 2.7 ตามลำดับ [45]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 การกำหนดพิกัดในแนวตั้งใน Kinova Convention

joint	
1	$\theta_{1\text{Jaco}} = -90^\circ$
2	$\theta_{2\text{Jaco}} = 180^\circ$
3	$\theta_{3\text{Jaco}} = 0^\circ$
4	$\theta_{4\text{Jaco}} = 180^\circ$
5	$\theta_{5\text{Jaco}} = 0^\circ$
6	$\theta_{6\text{Jaco}} = 180^\circ$
7	$\theta_{7\text{Jaco}} = 0^\circ$

ตารางที่ 2.5 การกำหนดพิกัดในแนวตั้งใน DH Parameter

joint	
1	$\theta_{1\text{DH}} = 90^\circ$
2	$\theta_{2\text{DH}} = 0^\circ$
3	$\theta_{3\text{DH}} = 0^\circ$
4	$\theta_{4\text{DH}} = 180^\circ$
5	$\theta_{5\text{DH}} = 0^\circ$
6	$\theta_{6\text{DH}} = 0^\circ$
7	$\theta_{7\text{DH}} = 0^\circ$

ตารางที่ 2.6 การแปลงพิกัดจาก DH เป็น Kinova

joint	
1	$\theta_{1\text{Jaco}} = -\theta_{1\text{DH}}$
2	$\theta_{2\text{Jaco}} = \theta_{2\text{DH}} + 180^\circ$
3	$\theta_{3\text{Jaco}} = \theta_{3\text{DH}}$
4	$\theta_{4\text{Jaco}} = \theta_{4\text{DH}}$
5	$\theta_{5\text{Jaco}} = \theta_{5\text{DH}}$
6	$\theta_{6\text{Jaco}} = \theta_{6\text{DH}} + 180^\circ$
7	$\theta_{7\text{Jaco}} = -\theta_{7\text{DH}}$

ตารางที่ 2.7 การแปลงพิกัดจาก Kinova เป็น DH

joint	
1	$\theta_{1\text{DH}} = -\theta_{1\text{Jaco}}$
2	$\theta_{2\text{DH}} = \theta_{2\text{Jaco}} - 180^\circ$
3	$\theta_{3\text{DH}} = \theta_{3\text{Jaco}}$
4	$\theta_{4\text{DH}} = \theta_{4\text{Jaco}}$
5	$\theta_{5\text{DH}} = \theta_{5\text{Jaco}}$
6	$\theta_{6\text{DH}} = \theta_{6\text{Jaco}} - 180^\circ$
7	$\theta_{7\text{DH}} = -\theta_{7\text{Jaco}}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

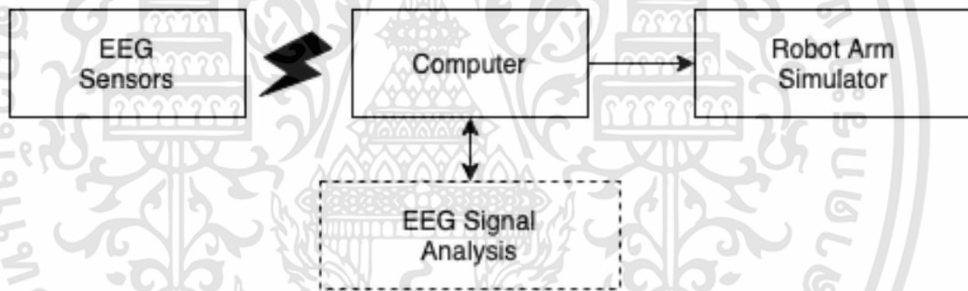
### บทที่ 3

## การออกแบบและการจัดทำปฏิญานิพนธ์

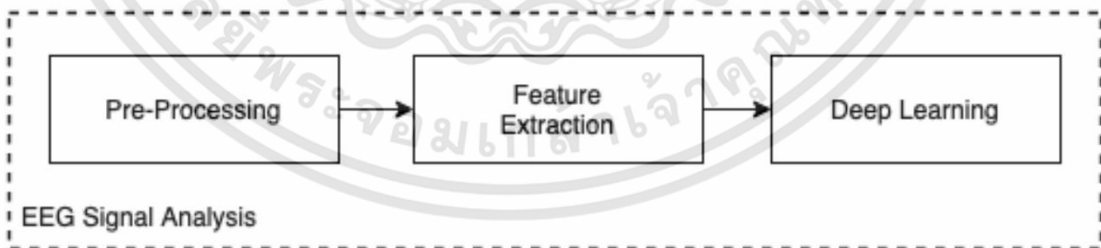
### 3.1 การออกแบบ

#### 3.1.1 การออกแบบระบบ

ปฏิญานิพนธ์นี้ต้องการนำอัลกอริธึมการเรียนรู้เชิงลึกที่ใช้ในการประมวลสัญญาณคลื่นไฟฟ้าจากสมองมาใช้ในการควบคุมแขนกลจำลอง ซึ่งเป็นตัวอย่างการนำ BCI มาใช้ในระบบการดูแลสุขภาพเพื่อช่วยเหลือผู้พิการและผู้สูงอายุในการทำกิจกรรมประจำวันต่อไป โดยบล็อกไดอะแกรมของระบบแสดงดังรูปที่ 3.1 ระบบเริ่มต้นจากการบันทึกสัญญาณคลื่นไฟฟ้าจากสมองผ่านเซนเซอร์และนำไปประมวลผล โดยมีกระบวนการย่อยแสดงดังรูปที่ 3.2 ได้แก่ การเตรียมข้อมูล, การสกัดคุณลักษณะข้อมูล, การฝึกสอนข้อมูล, การปรับค่าพารามิเตอร์ของอัลกอริธึมการเรียนรู้เชิงลึก สุดท้ายเมื่อมีโมเดลที่พร้อมใช้งานแล้วจะนำไปใช้ในการควบคุมแขนกลจำลองในคอมพิวเตอร์ต่อไป



รูปที่ 3.1 อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกลด้วยสัญญาณคลื่นไฟฟ้าจากสมอง



รูปที่ 3.2 บล็อกไดอะแกรมส่วนการวิเคราะห์สัญญาณ EEG (EEG Signal Analysis)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 ชุดข้อมูลตัวอย่างและการเตรียมข้อมูล (Pre-Processing)

#### 3.1.2.1 การเตรียมข้อมูลด้วยโปรแกรม EEGLAB

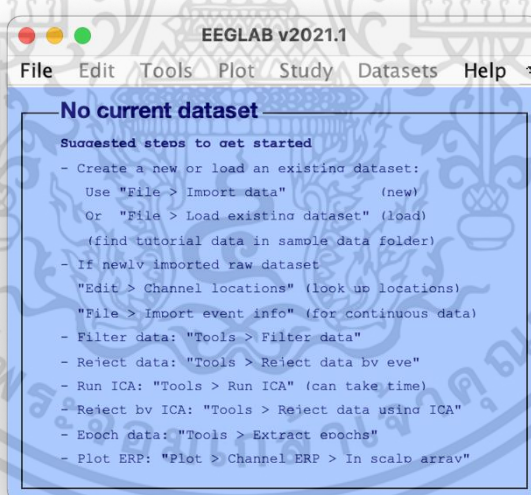
EEGLAB เป็นโปรแกรมถูกใช้อย่างแพร่หลายในการวิเคราะห์สัญญาณ EEG ของมนุษย์และวิเคราะห์ข้อมูลที่เกี่ยวข้องอื่นๆ โดยโปรแกรมนี้ได้รับการสนับสนุนจากโปรแกรมเมอร์ ผู้เขียนคู่มือ Plug-In และผู้ใช้หลายสิบคน [46]

ในการใช้งาน EEGLAB Extension จะรันผ่าน Command บนโปรแกรม MATLAB โดยเขียนคำสั่งดังรูปที่ 3.3 หลังจากนั้นจะปรากฏหน้าต่าง GUI ของโปรแกรม EEGLAB ดังรูปที่ 3.4 ซึ่งหากต้องการโปรแกรมที่ใช้ทั้งหมดใน EEGLAB สามารถบันทึกไฟล์โดยไปที่คำสั่ง File > History scripts > Save session history script



Command Window  
fx >> eeglab|

รูปที่ 3.3 คำสั่งในการใช้งาน EEGLAB Extension

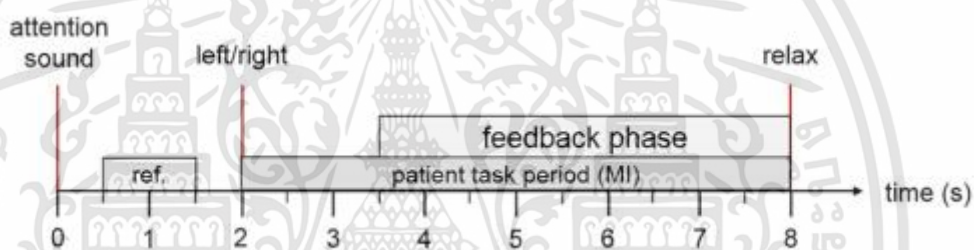


รูปที่ 3.4 หน้าต่างแรกของ EEGLAB Extension GUI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2.2 การเตรียมข้อมูลจากชุดข้อมูลของผู้ป่วยโรคหลอดเลือดสมอง (Stroke Dataset)

Stroke Dataset [47] ที่ใช้ในการศึกษาเป็นชุดข้อมูลจากงานแข่งขัน Hackathon ที่จัดขึ้นโดย br4in.io ซึ่งเป็นข้อมูลแบบส่วนตัว ชุดข้อมูลนี้ถูกจัดเก็บมาจากผู้ป่วยที่เป็นโรคหลอดเลือดในสมองจำนวน 3 คน โดยเป็นชุดข้อมูลก่อนได้รับการฝึกและหลังการฝึก MI โดยเหตุการณ์ที่เกิดขึ้นในหนึ่งการทดลองย่อยแสดงดังรูปที่ 3.5 [46] โดยประกอบไปด้วย MI สองประเภท คือ MI ของมือซ้ายและ MI ของมือขวา ในการทดลองย่อยนั้นจะเริ่มต้นด้วยเสียงสั้นๆ หลังจากนั้น วินาทีที่ 2 จะมีลูกศรบนจอมอนิเตอร์ที่ชี้ทางขวาหรือซ้ายให้ผู้ทำการทดลองจินตนาการการยกมือขวาหรือซ้าย วินาทีที่ 3.5 เป็นการป้อนกลับ และจบการทดลองในวินาทีที่ 8 โดยแต่ละการทดลองย่อยจะมีช่วงเวลาพักสั้นๆ หลังจากช่วงจินตนาการ



รูปที่ 3.5 เหตุการณ์ใน 1 การทดลองย่อยของข้อมูล Stroke Dataset

จากชุดข้อมูลของผู้ป่วยโรคหลอดเลือดสมองหลังได้รับการบำบัดมี Sampling Rate จำนวนที่เกิดเหตุการณ์และจำนวนตัวอย่างทั้งหมดดังรูปที่ 3.6 โดย





- Sampling Rate (fs) : 256 Hz
- จำนวนเหตุการณ์ที่กระตุ้น (Trig) : 197343
- จำนวนตัวอย่างทั้งหมด (y) : 197343x16 โดยเลข 16 หมายถึงจำนวนของช่องสัญญาณ

fs	256
trig	197343x1 double
y	197343x16 double

รูปที่ 3.6 มิติของข้อมูลในชุดข้อมูลของผู้ป่วยโรคหลอดเลือดสมองหลังได้รับการบำบัด

ผู้จัดทำได้เพิ่มตัวแปรของเหตุการณ์ที่กระตุ้นให้เป็นมิติที่ 17 ของตัวแปร เหตุการณ์ทั้งหมดโดยจำนวนมิติของข้อมูลแสดงดังรูปที่ 3.7 ในตัวแปร data

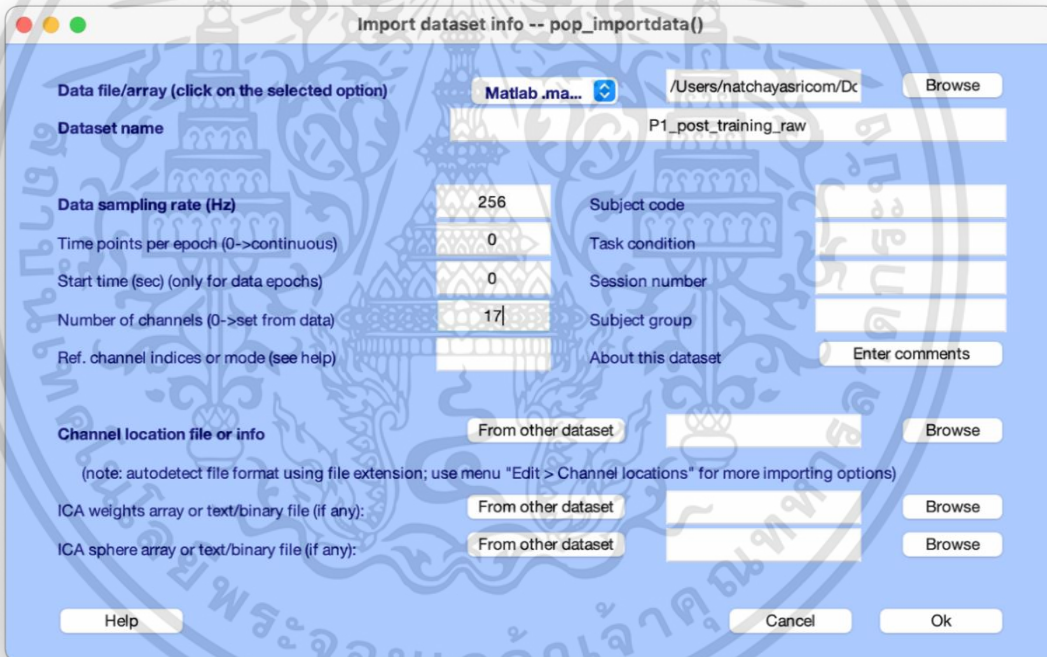
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	data	17x197343 double
	fs	256
	trig	197343x1 double
	y	197343x16 double

รูปที่ 3.7 มิติข้อมูลหลังการเพิ่มเหตุการณ์ที่เกิดการกระตุ้น

### 1) การนำเข้าชุดข้อมูล

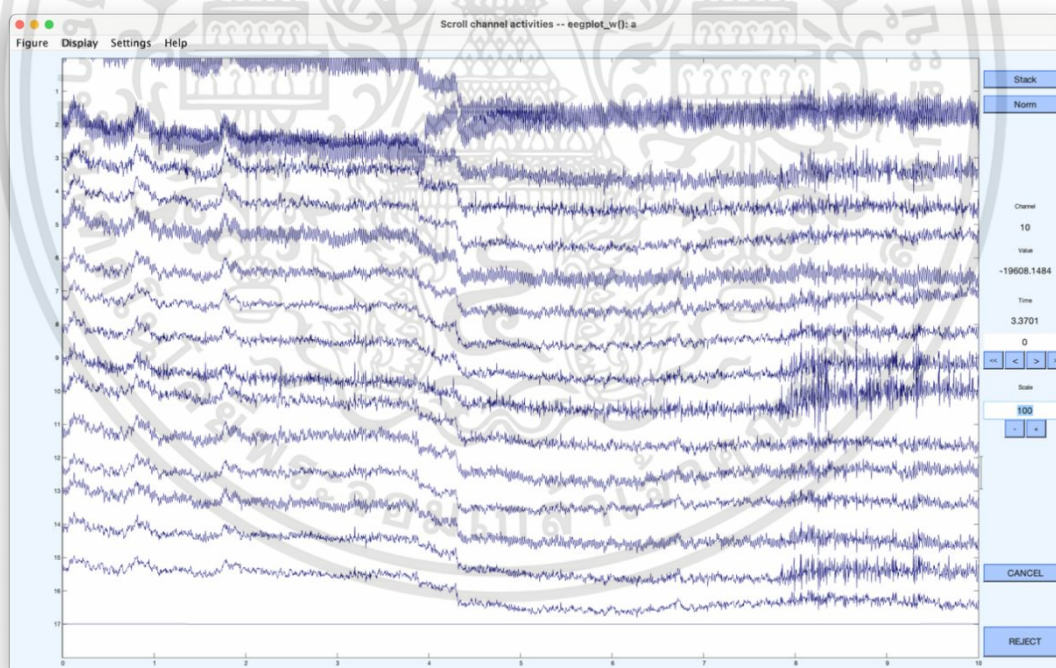
นำข้อมูลที่ทำการเพิ่มเหตุการณ์ที่เกิดการกระตุ้นแล้วในรูปที่ 3.7 มาเข้าโปรแกรม EEGLAB ตามคำสั่งในหัวข้อ 3.1.2.1 หลังจากนั้นเข้า File > Import data > Using EEGLAB functions and plugins > From ASCII/float or Matlab array โดยตั้งชื่อชุดข้อมูลเป็น P1\_post\_training\_raw และตั้งค่าตัวแปรอื่นๆ ดังรูปที่ 3.8 หลังจากนั้นรายละเอียดของชุดข้อมูลจะแสดงผ่าน GUI ดังรูปที่ 3.9 และรูปสัญญาณ EEG แสดงดังรูปที่ 3.10



รูปที่ 3.8 ตัวอย่างการนำเข้าชุดข้อมูลเข้าโปรแกรม EEGLAB



รูปที่ 3.9 รายละเอียดของข้อมูล P1\_post\_training\_raw

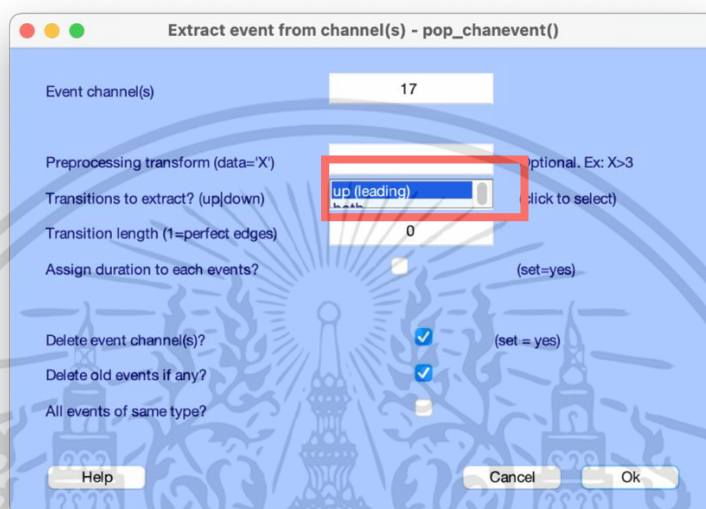


รูปที่ 3.10 ตัวอย่างสัญญาณ EEG จากชุดข้อมูลที่นำเข้า

2) การตั้งเหตุการณ์เกิดการกระตุ้น (Trigger) ของข้อมูล

ทำการตั้งเหตุการณ์เกิดการกระตุ้นจากข้อมูลมิติที่ 17 ออกจากข้อมูล EEG โดยไปที่ File > import event info > From data channel โดยทำการตั้งค่าตัวแปรดังรูปที่ 3.11 ซึ่ง up เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการตั้งค่า Transition หมายถึงเปลี่ยนแปลงค่าตัวแปรจากข้อมูลเหตุการณ์ -1 เป็น 1 รายละเอียดของชุดข้อมูลจะแสดงผ่าน GUI ดังรูปที่ 3.12 และสัญญาณที่ทำการเพิ่มเหตุการณ์จะแสดงดังรูปที่ 3.13 ซึ่งเส้นสีแดงในกรอบสีแดงหมายถึงการจินตนาการการขยับมือขวาและเส้นสีเขียวในกรอบสีเขียวหมายถึงการจินตนาการการขยับมือซ้าย



รูปที่ 3.11 ตัวอย่างการตั้งค่าการตั้งเหตุการณ์ Trigger ให้กับชุดข้อมูล

### 3) การแยกเหตุการณ์ของข้อมูล

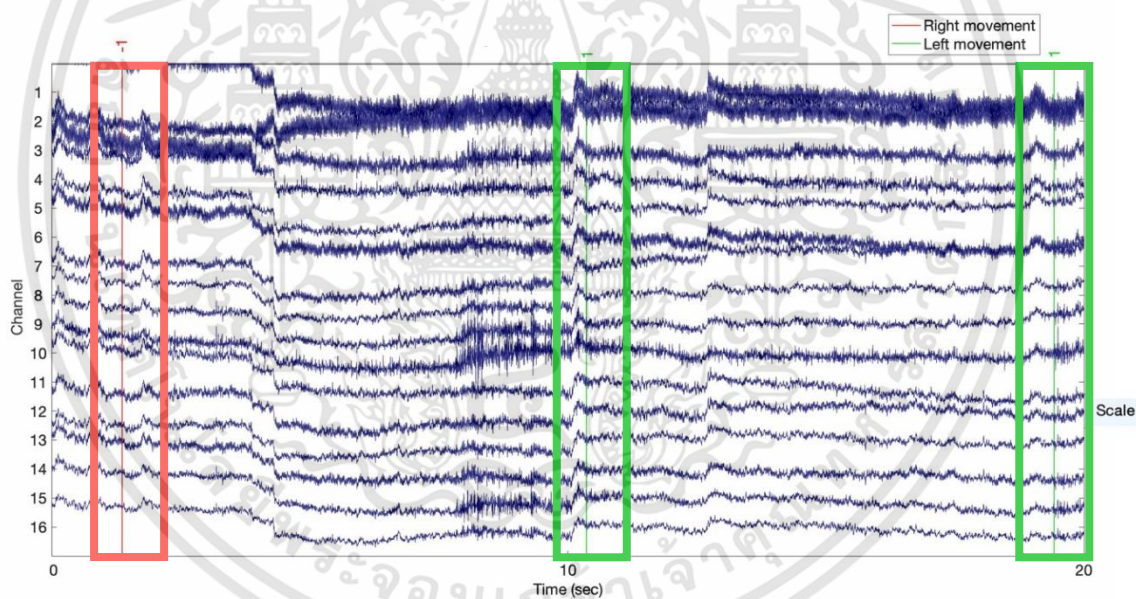
ทำการแยกเหตุการณ์แต่ละคลาส เช่น หากสนใจเฉพาะการเคลื่อนไหวด้วยมือขวา (Right-Hand Movement) สามารถไปที่ Tools > Extract epochs และตั้งค่าตัวแปรดังรูปที่ 3.14 ซึ่ง

- Time-locking event type(s) หมายถึง เหตุการณ์ที่ต้องการแยก ในที่นี้คือ -1
- Epoch limits [start, end] in seconds หมายถึง ช่วงเวลาเริ่มและจบที่เกิดเหตุการณ์

รายละเอียดของชุดข้อมูลจะแสดงผ่าน GUI ดังรูปที่ 3.15 และสัญญาณ  $\pm 1$  วินาที จากเหตุการณ์ Trigger จะแสดงดังรูปที่ 3.16 และ 3.17

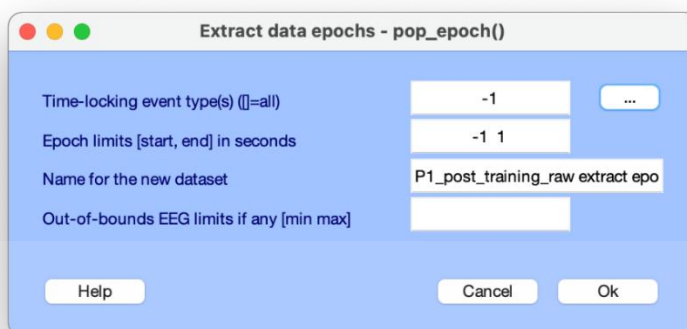


รูปที่ 3.12 รายละเอียดของข้อมูล P1\_post\_training\_raw หลังดึงเหตุการณ์ Trigger

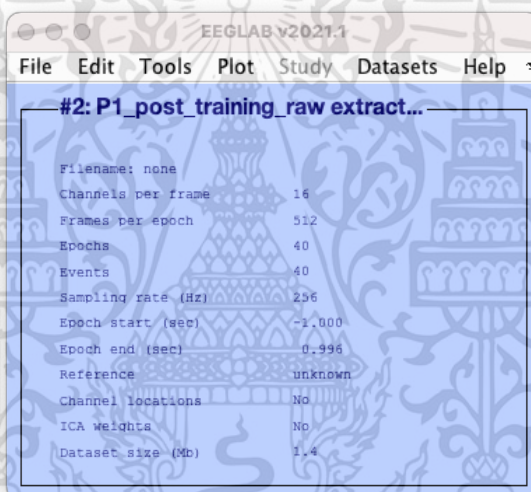


รูปที่ 3.13 ตัวอย่างสัญญาณ EEG ของข้อมูล P1\_post\_training\_raw ที่แสดงเหตุการณ์ Trigger ในช่วง 0-20 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

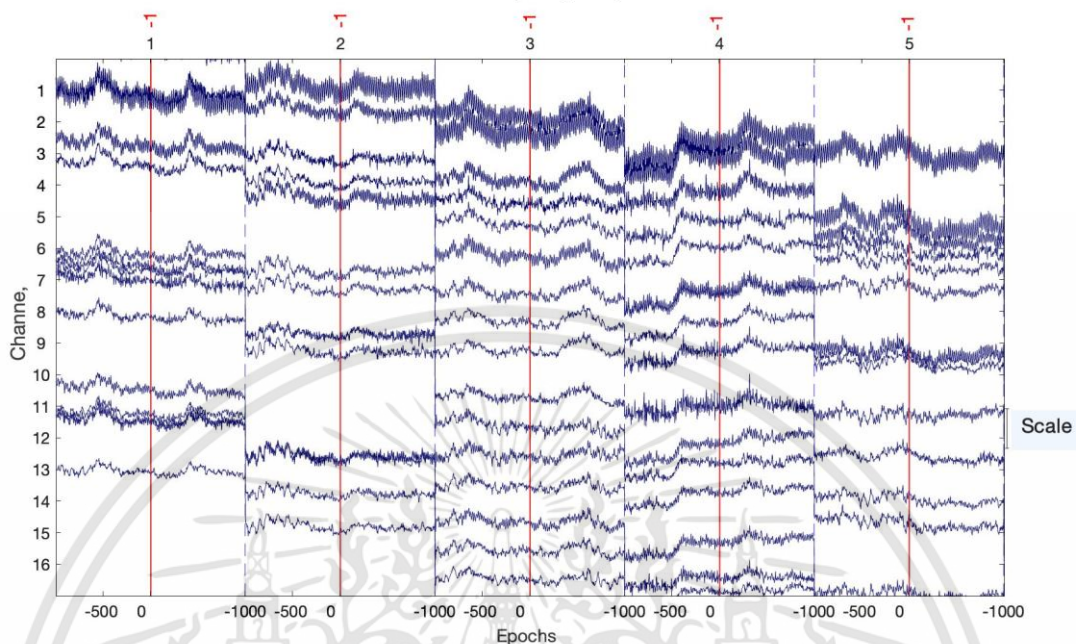


รูปที่ 3.14 ตัวอย่างการตั้งค่าการแยกเหตุการณ์การจินตนาการขยับมือขวา

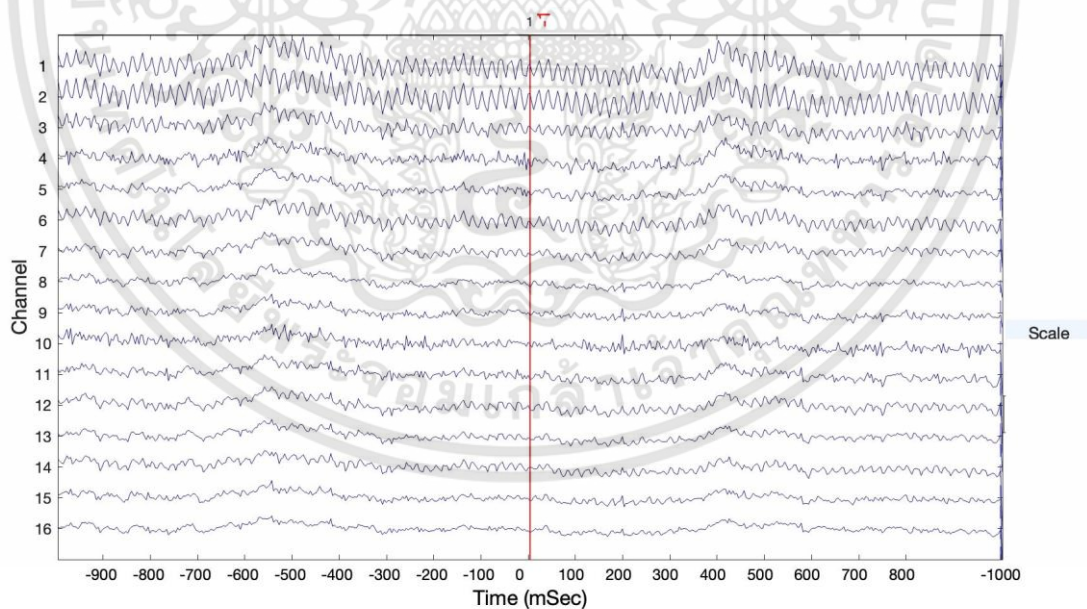


รูปที่ 3.15 รายละเอียดของข้อมูล P1\_post\_training\_raw หลังแยกเหตุการณ์การจินตนาการการขยับมือขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ตัวอย่างสัญญาณ EEG ของข้อมูล P1\_post\_training\_raw หลังแยกเหตุการณ์การจินตนาการการขยับมือขวา 5 Epochs



รูปที่ 3.17 ตัวอย่างสัญญาณ EEG ของข้อมูล P1\_post\_training\_raw หลังแยกเหตุการณ์การจินตนาการการขยับมือขวา ใน Epoch แรก

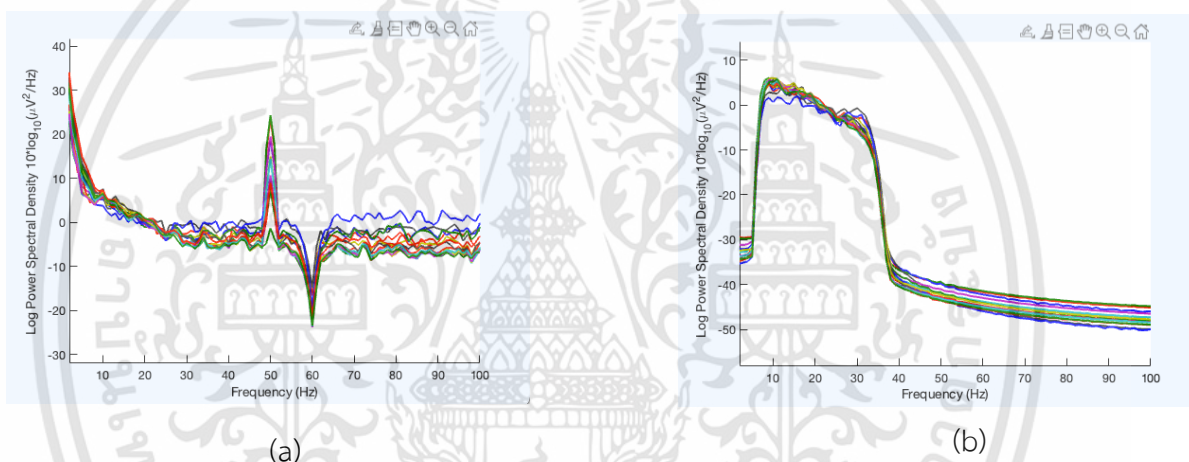
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4) การกรองความถี่ของสัญญาณ

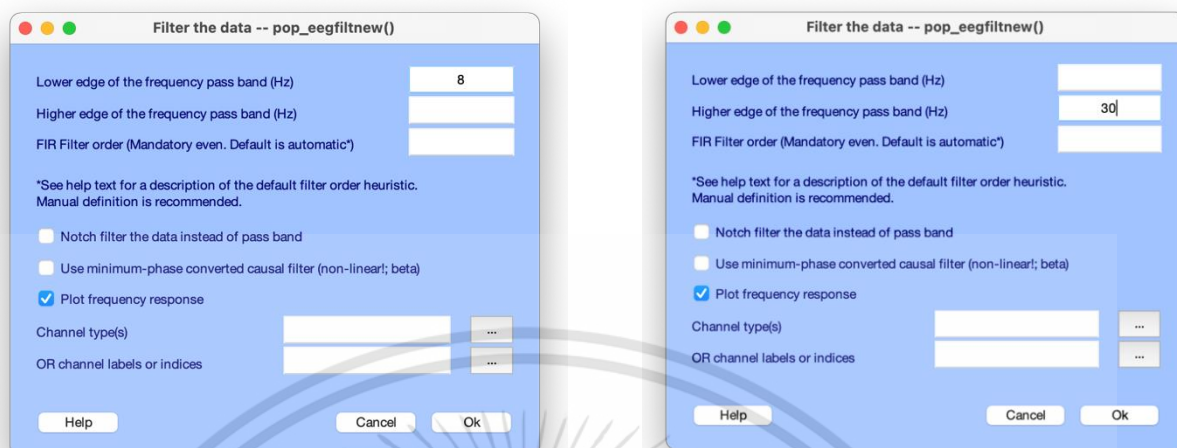
หลังทำการแยกเหตุการณ์ จะนำสัญญาณที่ได้ไปกรองเพื่อให้ได้ช่วงความถี่ที่ต้องการรวมถึงกรองสัญญาณรบกวน โดยใช้ FIR Filter ของ EEGLAB Extension ซึ่งรูปที่ 3.18 แสดงสเปกตรัมของสัญญาณก่อนและหลังผ่านการกรอง โดยไปที่ Tools > Filter the data > Basic FIR filter (new, default) และตั้งค่าตัวแปรดังรูปที่ 3.19 โดยในการทดลองนี้จะตั้งค่าตัวแปรคือ

- ตั้งค่าความถี่สูงผ่านที่ 8 Hz ดังรูปที่ 3.19 (a)
- ตั้งค่าความถี่ต่ำผ่านที่ 30 Hz ดังรูปที่ 3.19 (b)

และทำการกรองสัญญาณรบกวนที่ 50 Hz ออกด้วยตัวกรอง Notch โดยไปที่ Tools > Cleanline และตั้งค่าตามรูปที่ 3.20 เนื่องจากจะเห็นได้ว่ารูปที่ 3.18 (a) ที่ความถี่ 50 Hz มีค่า PSD สูง



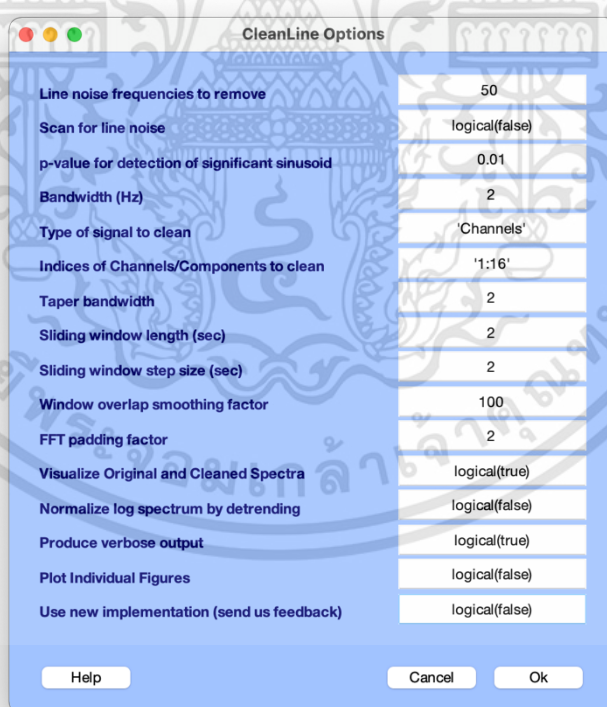
รูปที่ 3.18 สเปกตรัมของสัญญาณ (a) ก่อนผ่านการกรอง และ (b) หลังผ่านการกรอง



(a)

(b)

รูปที่ 3.19 ตัวอย่างการตั้งค่าการกรองสัญญาณใน EEGLAB  
 (a) การตั้งค่าความถี่ตัดของตัวกรองความถี่สูงผ่านที่ 8 Hz และ  
 (b) การตั้งค่าความถี่ตัดของตัวกรองความถี่ต่ำผ่านที่ 30 Hz



รูปที่ 3.20 การตั้งค่าการกรองสัญญาณด้วยตัวกรอง Notch ที่ความถี่ 50 Hz

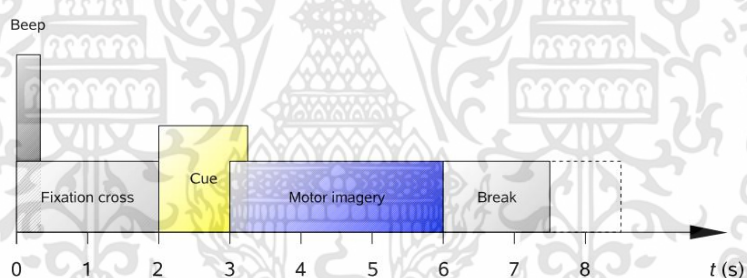
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2.3 การเตรียมสัญญาณจากชุดข้อมูล BCI Competition IV 2a

ข้อมูล BCI Competition IV 2a [47] ประกอบไปด้วยข้อมูลคลื่นไฟฟ้าจากสมองของผู้เข้าร่วมทดลองทั้งหมด 9 คนที่มีสุขภาพแข็งแรง โดยผู้เข้าร่วมการทดลองทุกคนนั้นถนัดมือขวา มีการมองเห็นที่ปกติ รูปที่ 3.21 [48] แสดงถึงเหตุการณ์ที่เกิดขึ้นใน 1 การทดลองย่อยของ Motor Imagery ซึ่งประกอบด้วยข้อมูลจาก 4 Class ได้แก่

- Class 1 : การจินตนาการการขยับแขนซ้าย
- Class 2 : การจินตนาการการขยับแขนขวา
- Class 3 : การจินตนาการการขยับเท้าทั้งสองข้าง
- Class 4 : การจินตนาการการขยับลิ้น

โดยการทดลองย่อยนั้นจะเริ่มต้นด้วยเครื่องหมายบวกและมีการส่งเสียงสั้นๆ หลังจากนั้นจะมีภาพ Cue แสดงถึงสิ่งที่จะต้องทำเป็นเวลา 1.25 วินาที หลังจากนั้นผู้เข้าร่วมทดลองต้องจินตนาการการขยับมือขวา, มือซ้าย, ลิ้นหรือเท้า (ตามที่แสดงใน Cue) เป็นเวลา 4 วินาที โดยแต่ละการทดลองย่อยจะมีช่วงเวลาพักสั้นๆ หลังจากช่วงจินตนาการ



รูปที่ 3.21 เหตุการณ์ที่เกิดขึ้นใน 1 การทดลองย่อยของข้อมูล BCI Competition IV 2a

จัดเตรียมข้อมูลเพื่อเลือกช่วงเวลาและกรองช่วงสัญญาณที่ต้องการโดยการเขียนโปรแกรมตัดช่วงเวลาที่เกิด MI คือ 2-5 วินาทีหลังเริ่มการทดลองและกรองสัญญาณในช่วงความถี่ 8-30 Hz ซึ่งแบ่งออกเป็น 5 ส่วนคือ

- (1) โหลดข้อมูลโดยจะอยู่ในรูปของตัวแปร  $s$  ตามต้นแบบตัวแปรเดิม
- (2) ทำการเติมข้อมูลที่เป็น NaN ด้วย 0
- (3) เก็บข้อมูลที่จำเป็นเช่น Sampling Rate และเวลาของสัญญาณโดยเก็บไว้ในตัวแปรที่เป็นโครงสร้างชื่อ `eeg`
- (4) ทำการแบ่งคลาสของ MI ซึ่งการจินตนาการการขยับมือซ้ายหมายถึง Class 1 และทำการเลือกช่วงเวลาระหว่าง 2-5 วินาทีหลังเริ่มการทดลองในช่วงสัญญาณที่ 8, 10 และ 12 ซึ่งตรงกับอิเล็กโทรด C3 Cz และ C4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (5) นำข้อมูลที่ผ่านการเลือกช่วงเวลาผ่านตัวกรองแถบผ่านแบบ Butterworth ซึ่งเป็นตัวกรองชนิดหนึ่งในการประมวลผลสัญญาณที่ถูกออกแบบให้มีการตอบสนองความถี่เรียบที่สุดเท่าที่จะเป็นไปได้ในช่วงความถี่แถบผ่าน 8-30 Hz

### 3.1.2.4 การประมาณค่าในช่วง (Interpolation)

เนื่องจากชุดข้อมูลที่นำมาศึกษาในปริญญาานิพนธ์มีค่าตัวแปรบางช่องสัญญาณและบางช่วงเวลาที่ขาดหายไป (NaN) เนื่องจากการเก็บหรือบันทึกสัญญาณ EEG มีปัญหาจากเหตุผลบางประการ เช่น การรบกวนของ Power Line (50/60 Hz), อิมพีแดนซ์มีความผิดปกติ (มากเกินไปหรือน้อยเกินไป), หน้าสัมผัสสวดของเครื่องมือขาดหรือไม่สามารถใช้งานได้อย่างปกติ, อิเล็กโทรดวางตำแหน่งไม่ถูกต้องหรือสัมผัสกับหนังศีรษะไม่ดี เจลที่ใช้กับอิเล็กโทรดมากเกินไป เป็นต้น ซึ่งทำให้ข้อมูลจากช่องสัญญาณบางช่องนั้นไม่สมบูรณ์ [49] ดังนั้นการประมาณค่าในช่วงจึงถูกใช้เพื่อสร้างสัญญาณที่ขาดหายจากปรากฏการณ์ดังกล่าว

ในปริญญาานิพนธ์นี้เลือกใช้ Cubic Spline Interpolation ซึ่งใช้เงื่อนไข Not-a-knot End การประมาณค่าในช่วงที่ต้องการจะใช้การประมาณค่าลูกบาศก์ (Cubic Interpolation) โดยใช้จุดกริดที่อยู่ใกล้เคียงในมิติที่เกี่ยวข้อง [50] โดยมีสมการที่ใช้ในการแทนค่าแสดงดังสมการที่ (3.1) เมื่อ  $i = 2, 3, \dots, n - 1$  และ  $k_1 = k_n = 0$  [51] ซึ่งคำสั่งที่ใช้ในการประมาณค่าในช่วงด้วยวิธี Cubic Spline Interpolation แสดงดังรูปที่ 3.22

$$k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1}) = 6 \left( \frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}} \right) \quad (3.1)$$

เมื่อ  $(x_{i-1}, y_{i-1})$ ,  $(x_i, y_i)$  และ  $(x_{i+1}, y_{i+1})$  คือ คู่อันดับ  $x$  และ  $y$  ที่  $i - 1$ ,  $i$  และ  $i + 1$

```
% Collect NaN values each channels
for i=1:22
    data = eeg.data(i,:); %Data
    x = find(isnan(eeg.data(i,:))); %Find NaN position
    pos = (0:length(data)-1); %total points in data

    % Interpolate Data using spline methods
    interp_data = spline(pos(~isnan(data)), data(~isnan(data)), pos(isnan(data)));

    % Replace NaN values
    data(x) = interp_data;

    eeg.data_new(i,:) = data;
end
```

รูปที่ 3.22 คำสั่งที่ใช้ประมาณค่าข้อมูลที่ขาดหายด้วยวิธี Cubic Spline Interpolation

### 3.1.3 การเขียนโปรแกรมสกัดคุณลักษณะ (Feature Extraction)

#### 3.1.3.1 การเขียนโปรแกรมสำหรับอัลกอริธึม CSP

รูปที่ 3.23 แสดงตัวอย่างที่หนึ่งของโปรแกรม CSP ซึ่งได้พัฒนาขึ้น [52] ซึ่งแบ่งออกเป็น (a) การสร้างชุดข้อมูลตัวอย่างที่นำมาทดสอบ ซึ่งเป็นข้อมูลแบบเกาส์เซียนเช่นเดียวกับข้อมูลในรูปที่ 2.7 และ (b) ส่วนของคำสั่งอัลกอริธึม CSP โดยที่อัลกอริธึมใช้ฟังก์ชันตามรูปที่ 3.24 ผลลัพธ์ของอัลกอริธึมแสดงดังรูปที่ 3.25 ตัวอย่างที่สองและสามแสดงดังรูปที่ 3.26 และ 3.27 ตามลำดับ โดยตัวอย่างของชุดข้อมูลที่สองคือ Sine Waves โดยคลาสแรกคือ Sine Wave ที่มีความถี่ 10 Hz เฟส 0 และคลาสสองคือ Sine Wave ที่มีความถี่ 10 Hz ที่ถูกเลื่อนเฟสไป  $\pi/8$  ในส่วนของตัวอย่างที่สาม ชุดข้อมูลในคลาสหนึ่งและสองเหมือนกับตัวอย่างที่สองแต่มีการเพิ่มสัญญาณรบกวนที่ SNR เท่ากับ 0 dB เมื่อพิจารณาจากรูปที่ 3.25-3.27 พบว่าสำหรับทุกตัวอย่างการกระจายของข้อมูลสองคลาสหลังผ่านอัลกอริธึมมีทิศทางที่ตั้งฉากกัน เป็นการยืนยันว่าโปรแกรมที่เขียนขึ้นสามารถทำงานได้อย่างถูกต้อง

```
N = 500;
mu = [0,0];
sigma = [6,1];
rot1 = eye(2); % Rotation for data1
theta = 15*pi/180; % Angle of rotation for data2
rot2 = [cos(theta) -sin(theta); sin(theta) cos(theta)];
data1 = (rot1*(repmat(mu,N,1)+ randn(N,2).*repmat(sigma,N,1)))';
data2 = (rot2*(repmat(mu,N,1)+ randn(N,2).*repmat(sigma,N,1)))';
```

(a)

```
% CSP
X1 = data1'; % Positive class data: X1~[C x T]
X2 = data2'; % Negative class data: X2~[C x T]
[W,l,A] = csp(X1,X2);
X1_CSP = W'*X1;
X2_CSP = W'*X2;
```

(b)

รูปที่ 3.23 ตัวอย่างที่ 1 ของโปรแกรม CSP (a) การสร้างชุดข้อมูลตัวอย่าง และ

(b) คำสั่งที่ใช้ในการวิเคราะห์ผลลัพธ์ของอัลกอริธึม CSP

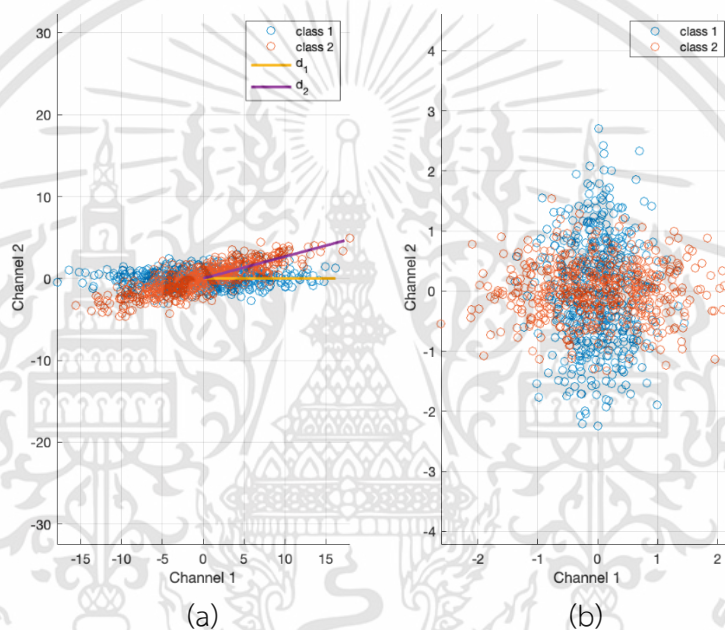
```

function [W, lambda, A] = csp(X1, X2)
% Compute the covariance matrix of each class
S1 = cov(X1') % S1~[C x C]
S2 = cov(X2') % S2~[C x C]

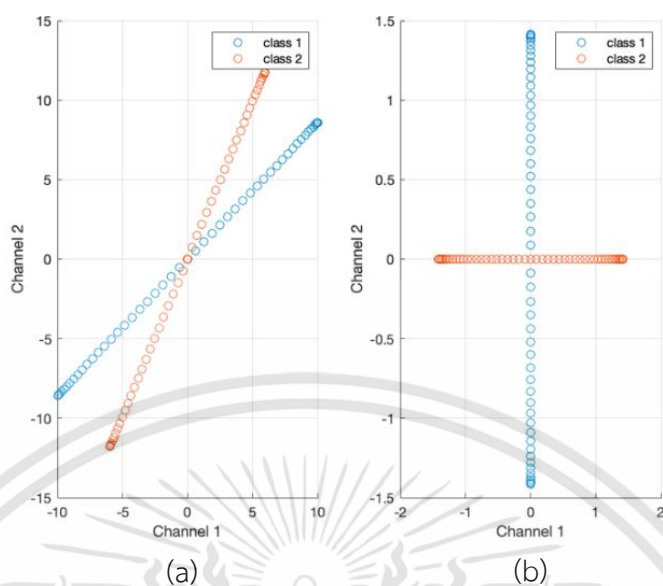
% Solve the eigenvalue problem S1·W = l·S2·W
[W,L] = eig(S1, S1 + S2); % Mixing matrix W
                               % (spatial filters are columns)
lambda = diag(L); % Eigenvalues
A = (inv(W))'; % Demixing matrix
end

```

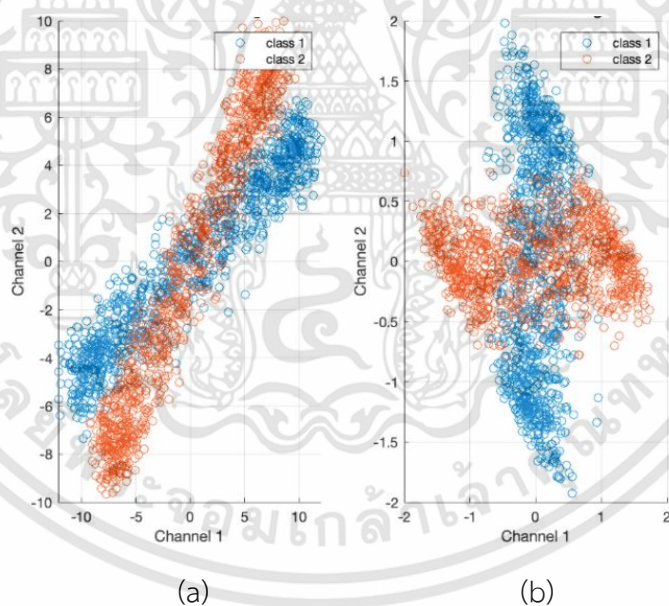
รูปที่ 3.24 ฟังก์ชัน CSP



รูปที่ 3.25 ผลลัพธ์ของอัลกอริธึม CSP บนตัวอย่างที่ 1 (a) การกระจายของชุดข้อมูลตั้งต้น และ (b) การกระจายของรูปของชุดข้อมูลหลังผ่านอัลกอริธึม CSP



รูปที่ 3.26 ผลลัพธ์ของอัลกอริธึม CSP บนตัวอย่างที่ 2 (a) การกระจายของชุดข้อมูลตั้งต้น และ (b) การกระจายของรูปของชุดข้อมูลหลังผ่านอัลกอริธึม CSP



รูปที่ 3.27 ผลลัพธ์ของอัลกอริธึม CSP บนตัวอย่างที่ 3 (a) การกระจายของชุดข้อมูลตั้งต้น และ (b) การกระจายของรูปของชุดข้อมูลหลังผ่านอัลกอริธึม CSP

### 3.1.3.2 การเขียนโปรแกรม Discrete Wavelet Transform (DWT)

ผู้จัดทำเขียนโปรแกรมสำหรับการแปลงเวฟเลตแบบดิสครีต (DWT) ดังรูปที่ 3.28 โดยใช้ Wavelet Toolbox ใน MATLAB โดย wname คือชื่อของเวฟเลตแบบดิสครีตที่จะนำมาแปลง และใช้ฟังก์ชัน wmaxlev เพื่อหาจำนวนชั้นที่มากที่สุดที่สามารถแปลงได้กับสัญญาณที่จะนำมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงและเก็บจำนวนชั้นที่มากที่สุดไว้ในตัวแปร level ในฟังก์ชัน wavedec เป็นการแปลงสัญญาณด้วยเวฟเลตแบบดิสครีตด้วยจำนวน Level และได้ตัวแปร c และ l ซึ่งเป็นตัวแปรที่เก็บค่าการแปลงเวฟเลตแบบดิสครีตเอาไว้ และนำสองตัวแปร c และ l ไปหาค่าสัมประสิทธิ์แบบประมาณและแบบละเอียดผ่านฟังก์ชัน appcoef และ detcoef ตามลำดับ

```
wname = 'db8';
level = wmaxlev(length(test_signal), wname);
[c,l] = wavedec(test_signal,level,wname);
approx = appcoef(c,l,wname);
cd = detcoef(c,l, "dcells");
```

รูปที่ 3.28 คำสั่งการแปลงเวฟเลตแบบดิสครีต

### 3.1.3.3 การเขียนโปรแกรม Continuous Wavelet Transform (CWT)

ผู้จัดทำเขียนโปรแกรมสำหรับการแปลงแบบเวฟเลตต่อเนื่อง (CWT) ดังรูปที่ 3.29 โดยใช้ Wavelet Toolbox ใน MATLAB โดยการแปลงเวฟเลตแบบต่อเนื่องนั้นต้องป้อนสัญญาณ EEG ไปที่ละการทดลองย่อยและทีละ Channel และใช้ฟังก์ชัน cwt ในการแปลงเวฟเลตแบบต่อเนื่อง โดยต้องใช้ตัวแปรสัญญาณที่จะนำไปแปลงและความถี่ในการสุ่ม โดยตัวแปรที่ได้คือ cfs และ frq ซึ่งเป็นค่าสัมประสิทธิ์ในแต่ละความถี่และความถี่ตามลำดับ

```
sigx = eeg.data.lh(trial_idx, :, channel_idx);
[cfs,frq] = cwt(sigx, eeg.fs);
```

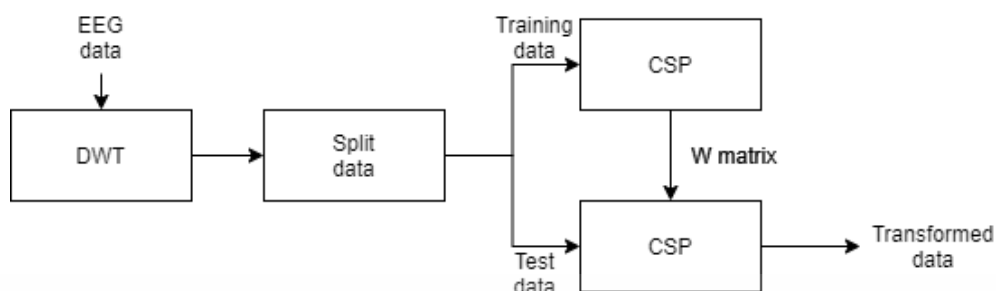
รูปที่ 3.29 คำสั่งการแปลงเวฟเลตแบบต่อเนื่อง

### 3.1.3.4 การรวมคุณลักษณะ (Feature Combination)

1) การรวมคุณลักษณะของการแปลงเวฟเลตแบบดิสครีต (Discrete Wavelet Transform; DWT) และ Common Spatial Pattern (CSP)

นำสัญญาณ EEG มาทำการแปลงเวฟเลตแบบดิสครีตและนำค่าสัมประสิทธิ์ที่ได้มาทำการแบ่งออกเป็นสองส่วนคือส่วนสำหรับการฝึกสอนและส่วนที่ใช้สำหรับการทดสอบ โดยอัตราส่วนแบ่งคือ 90% สำหรับการฝึกสอนและ 10% สำหรับการทดสอบ นำค่าสัมประสิทธิ์ที่ใช้สำหรับการฝึกสอนมาทำการแปลง CSP แล้วนำเมทริกซ์ W ของชุดฝึกสอนไปทำการแปลง CSP บนข้อมูลชุดทดสอบ ผลลัพธ์ที่ได้จะนำไปใช้สำหรับฝึกสอนและทดสอบการจำแนกประเภทสัญญาณด้วยการเรียนรู้เชิงลึก บล็อกไดอะแกรมการรวมคุณลักษณะแสดงดังรูปที่ 3.30

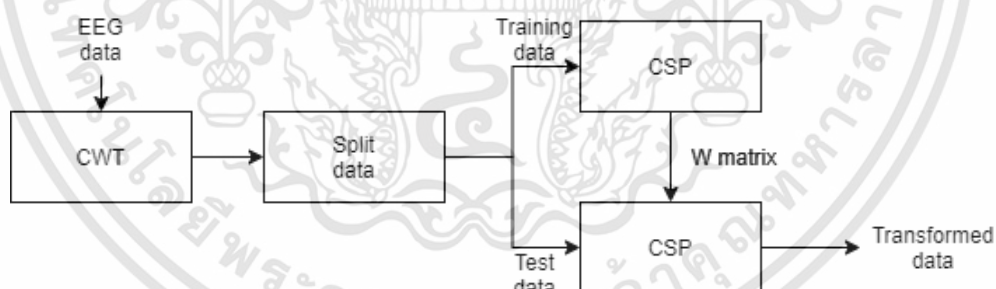
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 บล็อกไดอะแกรมการรวมคุณลักษณะการแปลงเวฟเลตแบบดิสครีตและ CSP

2) การรวมคุณลักษณะของการแปลงเวฟเลตแบบต่อเนื่อง (Continuous Wavelet Transform; CWT) และ Common Spatial Pattern (CSP)

นำสัญญาณ EEG มาทำการแปลงเวฟเลตแบบต่อเนื่องโดยเลือกความถี่ในช่วง 8 – 30 Hz และเลือกอิเล็กโทรด C3, Cz, C4 นำค่าสัมประสิทธิ์ที่ได้มาทำการแบ่งออกเป็นสองส่วนคือ ส่วนสำหรับการฝึกสอนและส่วนที่ใช้สำหรับการทดสอบ โดยอัตราการแบ่งคือ 90% สำหรับการฝึกสอน และ 10% สำหรับการทดสอบ นำค่าสัมประสิทธิ์ที่ใช้สำหรับการฝึกสอนมาทำการแปลง CSP แล้วนำเมทริกซ์  $W$  ของชุดฝึกสอนไปทำการแปลง CSP บนข้อมูลชุดทดสอบ ผลลัพธ์ที่ได้นำไปใช้สำหรับฝึกสอนและทดสอบการจำแนกประเภทสัญญาณด้วยการเรียนรู้เชิงลึก บล็อกไดอะแกรมการรวมคุณลักษณะแสดงดังรูปที่ 3.31



รูปที่ 3.31 บล็อกไดอะแกรมการรวมคุณลักษณะการแปลงเวฟเลตแบบต่อเนื่องและ CSP

### 3.1.4 การเขียนโปรแกรมการเรียนรู้เชิงลึก

#### 3.1.4.1 การเขียนโปรแกรมการเรียนรู้เชิงลึกแบบ MLP

รูปที่ 3.32 แสดงคำสั่งที่ใช้สร้างโมเดลการเรียนรู้เชิงลึกแบบ MLP โดย unit คือ จำนวนนิวรอน (Neuron) ในแต่ละ Layer โดยสามารถปรับแต่งจำนวน Layer และจำนวนนิวรอน เพื่อให้โมเดลเรียนรู้จากคุณลักษณะที่ป้อนเข้าไปในโมเดลระหว่างเรียนรู้ได้ดีที่สุด ใช้คำสั่งในรูปที่ 3.33 เพื่อทำการตั้งค่าส่วนประกอบที่สำคัญในการสร้างโมเดลและคำสั่งในการฝึกสอนโมเดลแสดงในรูปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.34 และรูปที่ 3.35 แสดงคำสั่งในการหาจำนวนนิเวรอนที่ทำให้ค่า Validation Loss น้อยที่สุดเพื่อใช้ในการปรับแต่งโมเดลให้เหมาะสมกับคุณลักษณะที่ป้อนเข้าไปมากที่สุด

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(unit1, activation="relu"),
    tf.keras.layers.Dense(unit2, activation="relu"),
    tf.keras.layers.Dense(unit3, activation="relu"),
    tf.keras.layers.Dense(1)])
```

รูปที่ 3.32 คำสั่งในการสร้างโมเดลการเรียนรู้เชิงลึกแบบ MLP

```
model.compile(loss='binary_crossentropy',
              optimizer=keras.optimizers.SGD(learning_rate=1e-4, momentum=0.4),
              metrics=['accuracy'])
```

รูปที่ 3.33 คำสั่งที่ใช้ในการตั้งค่าส่วนประกอบในการสร้างโมเดล

```
history = model.fit(X_train, y_train, epochs=300,
                   verbose = 1, batch_size = 32,
                   validation_split=0.1)
```

รูปที่ 3.34 คำสั่งที่ใช้ในการฝึกสอนโมเดล

```
def build_model(hp):
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Dense(
        units=hp.Int('units', min_value=8, max_value=512, step=8),
        activation='relu', input_shape=[dim]))
    model.add(tf.keras.layers.Dense(
        units=hp.Int('units2', min_value=8, max_value=512, step=8), activation='relu'))
    model.add(tf.keras.layers.Dense(
        units=hp.Int('units3', min_value=8, max_value=512, step=8), activation='relu'))
    model.add(tf.keras.layers.Dense(1))
    hp_learning_rate = hp.Choice('learning_rate', values=[1e-3, 1e-4, 1e-4, 1e-5, 1e-6])
    optimizer = tf.keras.optimizers.SGD(learning_rate=hp_learning_rate, momentum=0.5)
    model.compile(loss="mse", optimizer=optimizer)
    return model

tuner = keras_tuner.RandomSearch(build_model,
                                objective='val_loss', max_trials=20,
                                executions_per_trial=3, directory='my_dir',
                                project_name='MLP_4')
```

รูปที่ 3.35 คำสั่งที่ใช้ในการหาจำนวนนิเวรอนที่ดีที่สุด

#### 3.1.4.2 การเขียนโปรแกรมการเรียนรู้เชิงลึกแบบ Pre-Trained Models

ในปริณญาณิพนธ์นี้ผู้จัดทำได้ศึกษา Pre-Trained Models 3 โครงสร้าง ได้แก่ ResNet18, ResNet50 และ ShuffleNet ดังรายละเอียดในหัวข้อ 2.5.3 รูปที่ 3.36 แสดงคำสั่งที่ใช้ในการนำเข้ารูปภาพคุณลักษณะจากโฟลเดอร์ แบ่งรูปภาพออกเป็นสามส่วนคือ ชุดสำหรับฝึกสอน, ชุดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการตรวจสอบ และชุดสำหรับการทดสอบ โดยอัตราแบ่งคือ 80% สำหรับการฝึกสอน 10% สำหรับการตรวจสอบและ 10% สำหรับการทดสอบ และทำการปรับขนาดของรูปภาพให้เหมาะสมกับ Pre-Trained Model ที่จะนำมาใช้ในการฝึกสอน โดยคำสั่งสำหรับการเรียนรู้เชิงลึกแบบ Pre-Trained Model นั้นจะใช้โปรแกรม MATLAB และใช้เฟรมเวิร์ค Deep Learning Toolbox ตัวอย่างการระบุโมเดลที่ใช้ซึ่งในตัวอย่างเลือกใช้โครงสร้าง ResNet18 และตั้งค่าส่วนประกอบต่างๆ ที่ใช้ในการฝึกสอนสุดท้ายคือคำสั่งที่ใช้ในการฝึกสอนโมเดลแสดงดังรูปที่ 3.37

```
sizeim = 224;
imds = imageDatastore("C:\Users\User\Desktop\4-th year project\test\Project_code\week\Case1\CWT\Data", ...
    "IncludeSubfolders",true,"LabelSource","foldernames");
[trainImgs, validImgs, testImgs] = splitEachLabel(imds,0.8,0.1);
trainReImg = augmentedImageDatastore([sizeim sizeim],trainImgs);
validReImg = augmentedImageDatastore([sizeim sizeim],validImgs);
testReImg = augmentedImageDatastore([sizeim sizeim],testImgs);
```

รูปที่ 3.36 คำสั่งที่ใช้ในการนำเข้าและปรับขนาดของรูปภาพคุณลักษณะ

```
net = resnet18;
lgraph = layerGraph(net);
newfc = fullyConnectedLayer(2,"Name","new_fc","BiasLearnRateFactor",10,"WeightLearnRateFactor",10);
newout = classificationLayer("Name","new_out");
lgraph = replaceLayer(lgraph,"fc1000",newfc);
lgraph = replaceLayer(lgraph,"ClassificationLayer_predictions",newout);
options = trainingOptions('adam', ...
    'InitialLearnRate',0.001, ...
    'MaxEpochs',15, ...
    'LearnRateSchedule','piecewise', ...
    'MiniBatchSize',32, ...
    'Shuffle','every-epoch', ...
    'Verbose',false, ...
    'ValidationData',validReImg, ...
    'ValidationFrequency',5);

net = trainNetwork(trainReImg,lgraph,options)

net =
  DAGNetwork with properties:
    Layers: [71x1 nnet.cnn.layer.Layer]
    Connections: [78x2 table]
    InputNames: {'data'}
    OutputNames: {'new_out'}
```

รูปที่ 3.37 คำสั่งที่ใช้ในการระบุโมเดล, ปรับแต่งโมเดลและฝึกสอนโมเดล

การปรับแต่งโมเดลเพื่อหาค่า Hyperparameter สามารถทำได้โดยใช้แอปพลิเคชัน Experiment Manager ของ MATLAB รูปตัวอย่างแอปพลิเคชันและชุดคำสั่ง Setup Function ที่ใช้ในการปรับแต่งโมเดลแสดงในรูปที่ 3.38 และ 3.39 ตามลำดับ

## Hyperparameters

Strategy: Bayesian Optimization

Name	Range	Type	Transform
Period	[10 50]	integer	none
InitialLearnRate	[1e-4 1e-1]	real	log
L2Regularization	[1e-10 1e-1]	real	log
Factor	[1e-1 8e-1]	real	none

 Add  Delete

## Bayesian Optimization Options

Name	Value
Maximum time (in seconds)	Inf
Maximum number of trials	70

## Setup Function

BayesOptExperiment\_setup1

 New  Edit

รูปที่ 3.38 หน้าต่างแอปพลิเคชัน Experiment Manager

```

im_size = 224;
train = imageDatastore("C:\Users\User\Desktop\4-th year project\test\Project_code\Last_data\SUB 9\Fold 1\Train", ...
    "IncludeSubfolders",true,"LabelSource","foldernames");
imds = imageDatastore("C:\Users\User\Desktop\4-th year project\test\Project_code\Last_data\SUB 9\Fold 1\Test", ...
    "IncludeSubfolders",true,"LabelSource","foldernames");
[validImgs, notuse] = splitEachLabel(imds,0,5);
trainReImg = augmentedImageDatastore([im_size im_size],train);
validReImg = augmentedImageDatastore([im_size im_size],validImgs);

net = resnet18;
lgraph = layerGraph(net);
newfc = fullyConnectedLayer(2,"Name","new_fc",'WeightLearnRateFactor',10,'BiasLearnRateFactor',10);
newout = classificationLayer("Name","new_out");
lgraph = replaceLayer(lgraph,"fc1000",newfc);
lgraph = replaceLayer(lgraph,"ClassificationLayer_predictions",newout);
options = trainingOptions('adam', ...
    'InitialLearnRate',params.InitialLearnRate, ...
    'MaxEpochs',params.Epoch, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',params.Period, ...
    'LearnRateDropFactor',params.Factor, ...
    'MiniBatchSize',params.Batch, ...
    'L2Regularization',params.L2Regularization, ...
    'Shuffle','every-epoch', ...
    'Verbose',false, ...
    'ValidationData',validReImg, ...
    'ValidationFrequency',5);
end

```

รูปที่ 3.39 คำสั่ง Setup Function ที่ใช้ในการหาค่า Hyperparameter ที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4.3 การเขียนโปรแกรมการเรียนรู้เชิงลึกแบบ LeNet-5

ก่อนทำการฝึกสอนและประเมินโมเดล ผู้จัดทำได้ทำการ Normalization หรือ Feature Scaling ชุดข้อมูลในแต่ละการทดลองย่อยหลังจากสกัดคุณลักษณะข้อมูลด้วยวิธีต่างๆ โดยใช้คำสั่งดังรูปที่ 3.40 แบ่งชุดข้อมูลสำหรับฝึกสอน (Train) และตรวจสอบ (Validate) เป็น 80% และ 20% ตามลำดับ

```
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train,
                                                test_size=0.2,
                                                random_state=seed_value)
```

รูปที่ 3.40 คำสั่งที่ใช้ในการแบ่งชุดข้อมูลสำหรับการฝึกสอนและตรวจสอบ

รูปที่ 3.41 (a) แสดงคำสั่งที่ใช้ในการสร้างโมเดลการเรียนรู้เชิงลึกแบบ LeNet-5 ซึ่งมีสถาปัตยกรรมของโมเดลดังกล่าวไว้ในหัวข้อ 2.4.1 โดยมีการใช้ค่าต่างๆ ดังนี้

- Kernel Initializer: GlorotUniform Initializer
- Kernel Regularization: Ridge Regression หรือ L2 Regularization ถูกใช้เพื่อลด Generalization Error ของโมเดลจากปัญหาเรื่อง Overfitting โดยกำหนดให้ค่า  $\lambda$  หรือ Weight Decay เท่ากับ 0.01 ตาม Default ของ Keras [53]

คำสั่งที่ใช้ในการตั้งค่าส่วนประกอบในการสร้างโมเดลแสดงดังรูปที่ 3.41 (b) โดยใช้วิธีการคำนวณค่าความผิดพลาดแบบ Binary Cross Entropy ใช้ Optimizer ชนิด Adam และคำนวณค่าความแม่นยำของโมเดลแบบ Accuracy การปรับแต่งโมเดลเพื่อหาค่า Hyperparameter ที่ดีที่สุดใช้คำสั่งดังรูปที่ 3.42 โดยใช้วิธี Bayesian Optimization ซึ่งจุดประสงค์ในการปรับแต่งโมเดลเพื่อให้ค่า Validation Accuracy (ตัวแปร val\_accuracy) มีค่ามากที่สุดและทำการ Tune ทั้งหมด 20 รอบ (ตัวแปร max\_trials) Hyperparameter ที่ทำการปรับแต่งประกอบด้วย

- Learning rate: 0.01, 0.001 และ 0.0001
- Batch size: 24 – 84 โดยเพิ่มขึ้นทีละ 10

รูปที่ 3.43 คือ Callbacks API ใน Keras มีจุดประสงค์เพื่อลด Learning Rate ในขณะที่ฝึกสอนโมเดลโดยพารามิเตอร์ในคำสั่งคือให้ลดค่า Learning rate ครั้งละ 0.1 เมื่อค่าของ Validation Loss ไม่มีการเปลี่ยนแปลงติดกัน 10 ครั้ง เมื่อได้ Hyperparameter ที่ดีที่สุด จะนำตัวแปรที่ได้มาใช้คำสั่งรูปที่ 3.44 ในการฝึกสอนโมเดล 500 Epoch และประเมินความแม่นยำของโมเดลได้ใช้คำสั่งดังรูปที่ 3.45

```
def build_lenet5(hp):
    initializer = tf.keras.initializers.GlorotUniform(seed=seed_value)

    ip = keras.Input(shape=(57,750,1), name='Input')
    x = layers.Conv2D(6, 5, strides=1, activation='tanh', name='C1',
                    padding='same', kernel_initializer=initializer, kernel_regularizer='l2')(ip)
    x = layers.AveragePooling2D(name='S2')(x)
    x = layers.Conv2D(16, 5, strides=1, activation='tanh', name='C3',
                    padding='valid', kernel_initializer=initializer, kernel_regularizer='l2')(x)
    x = layers.AveragePooling2D(name='S4')(x)
    x = layers.Flatten(name='Flatten')(x)
    x = layers.Dense(120, activation='tanh', name='C5', kernel_initializer=initializer,
                    kernel_regularizer='l2')(x)
    op = layers.Dense(1, activation='sigmoid', name='Output')(x)

    model = keras.Model(inputs=ip, outputs=op, name="Lenet5")
    hp_lr = hp.Choice('learning_rate', [1e-2, 1e-3, 1e-4])
```

(a)

```
model.compile(loss='binary_crossentropy',
              optimizer=Adam(learning_rate=hp_lr),
              metrics=['accuracy'])
return model
```

(b)

รูปที่ 3.41 คำสั่งที่ใช้ในการเรียนรู้เชิงลึกแบบ LeNet-5 (a) การสร้างโมเดลการเรียนรู้เชิงลึก และ (b) การตั้งค่าส่วนประกอบในการสร้างโมเดลการเรียนรู้เชิงลึกแบบการเรียนรู้เชิงลึก

```
class MyTuner(keras.tuners.BayesianOptimization):
    def run_trial(self, trial, *args, **kwargs):
        kwargs['batch_size'] = trial.hyperparameters.Int('batch_size', 24, 84, step=10)
        return super(MyTuner, self).run_trial(trial, *args, **kwargs)

tuner = MyTuner(build_lenet5, objective=keras.Objective('val_accuracy', 'max'),
               max_trials=20, project_name=name, seed=seed_value, directory='Lenet5_dir')

tuner.search(x1_train, y1_train, validation_data = (x1_val, y1_val), callbacks=[reduce_lr], epochs=500)
best_hp = tuner.get_best_hyperparameters(1)[0]
```

รูปที่ 3.42 คำสั่งที่ใช้หาค่า Hyperparameter ที่ดีที่สุดโดยใช้วิธี Bayesian Optimization

```
reduce_lr = ReduceLR0nPlateau(monitor='val_loss', factor=0.1, patience=10, min_lr=0)
```

รูปที่ 3.43 คำสั่งที่ใช้ในการลด Learning Rate ใน Callbacks API ของ Keras

```
model = tuner.hypermodel.build(best_hp)
history = model.fit(x1_train, y1_train, validation_data = (x1_val, y1_val),
                  batch_size=best_hp.values['batch_size'],
                  epochs=500, verbose=0, callbacks=[reduce_lr])
```

รูปที่ 3.44 คำสั่งที่ใช้ในการฝึกสอนโมเดล

```
_, acc = model.evaluate(x1_test, y_test)
```

รูปที่ 3.45 คำสั่งที่ใช้ในการประเมินความแม่นยำของโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4.4 การเขียนโปรแกรมการเรียนรู้เชิงลึกแบบการเรียนรู้เชิงลึกแบบผสม (HDNN)

รูปที่ 3.46 (a) แสดงคำสั่งที่กล่าวไว้ในการสร้างโมเดลการเรียนรู้เชิงลึกแบบการเรียนรู้เชิงลึกแบบผสมซึ่งมีสถาปัตยกรรมของโมเดลดังกล่าวไว้ในหัวข้อ 2.5.4 โดยมีการใช้ Kernel Initializer และ Kernel Regularization เหมือนกับการฝึกสอนโมเดลการเรียนรู้เชิงลึกแบบ LeNet-5 คำสั่งที่ใช้ในการตั้งค่าส่วนประกอบในการสร้างโมเดลแสดงดังรูปที่ 3.46 (b) โดยใช้วิธีการคำนวณค่าความผิดพลาดแบบ Binary Cross Entropy ใช้ Optimizer ชนิด Adam และคำนวณค่าความแม่นยำของโมเดลแบบ Accuracy การปรับแต่งโมเดลเพื่อหาค่า Hyperparameter ที่ดีใช้วิธี Bayesian Optimization เหมือนดังรูปที่ 3.42 และรายละเอียดเหมือนวิธีหา Hyperparameter ในโมเดล LeNet-5 แต่มีเพิ่มส่วนของ Units หรือโหนดใน LSTM Layer เป็น 8, 16 และ 32 โหนด และมีการใช้ Callbacks API ใน Keras เหมือนโมเดล LeNet-5 แล้วใช้คำสั่งในการฝึกสอนโมเดลและประเมินความแม่นยำของโมเดลดังรูปที่ 3.44 และ 3.45 ตามลำดับ

```

def get_cnn_lstm(hp):
    initializer = tf.keras.initializers.HeUniform(seed=seed_value)

    # CNN
    ip = keras.Input(shape=(57,750,1), name='Input')

    #CNN_1
    CNN_1 = layers.Conv2D(5, 3, padding='same', activation='relu',
                        kernel_initializer=initializer,
                        kernel_regularizer=regularizers.l2(1e-2))(ip)

    #CNN_2
    n_1 = layers.Conv2D(3, 3, padding='same', activation='relu',
                      kernel_initializer=initializer, kernel_regularizer='l2')(CNN_1)
    mp_1 = layers.MaxPooling2D(pool_size=(2,2), strides=2, padding='same')(n_1)
    n_2 = layers.Conv2D(3, 3, padding='same', activation='relu',
                      kernel_initializer=initializer, kernel_regularizer='l2')(mp_1)
    mp_2 = layers.MaxPooling2D(pool_size=(2,2), strides=2, padding='same')(n_2)
    n_3 = layers.Conv2D(3, 3, padding='same', activation='relu',
                      kernel_initializer=initializer, kernel_regularizer='l2')(mp_2)
    mp_3 = layers.MaxPooling2D(pool_size=(2,2), strides=2, padding='same')(n_3)
    s_f = layers.Flatten()(mp_3)

    #LSTM_side
    input_fc_lstm = layers.Flatten(name='Flatten')(CNN_1)
    input_fc_lstm = layers.Reshape((1,-1))(input_fc_lstm)
    lstm = layers.LSTM(hp.Choice('units', [8,16,32]), name='lstm_1',
                      kernel_initializer=initializer,
                      kernel_regularizer='l2')(input_fc_lstm)
    s_t = layers.Flatten()(lstm)

    #concat and FC
    concat = tf.keras.layers.Concatenate()([s_f, s_t])
    f_c = layers.Dense(512, activation='relu', kernel_initializer=initializer,
                      kernel_regularizer='l2')(concat)
    f_c = layers.Dense(512, activation='relu', kernel_initializer=initializer,
                      kernel_regularizer='l2')(f_c)
    f_c = layers.Dense(32, activation='relu', kernel_initializer=initializer,
                      kernel_regularizer='l2')(f_c)
    op = layers.Dense(1, activation='sigmoid', name='Output')(f_c)

    model = keras.Model(inputs=ip, outputs=op, name='cnn_lstm')

```

(a)

```

hp_lr = hp.Choice('learning_rate', [1e-2, 1e-3, 1e-4])
# hp_lr = hp.Float("learning_rate", 1e-4, 1e-1, sampling="log")
model.compile(loss='binary_crossentropy',
              optimizer=Adam(learning_rate=hp_lr),
              metrics=["accuracy"])
return model

```

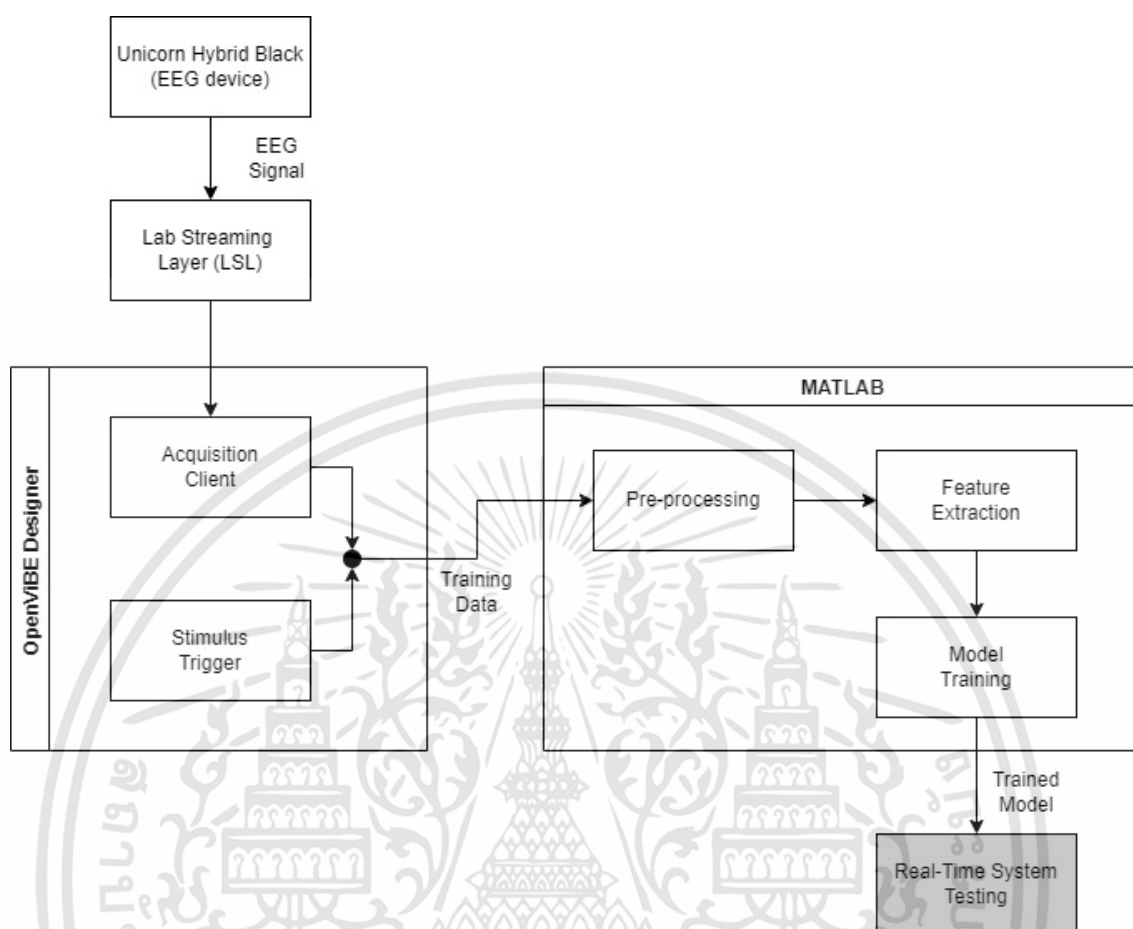
(b)

รูปที่ 3.46 คำสั่งการเรียนรู้เชิงลึกแบบผสม (a) การสร้างโมเดลการเรียนรู้เชิงลึกแบบผสม และ (b) การตั้งค่าส่วนประกอบในการสร้างโมเดลการเรียนรู้เชิงลึกแบบผสม

### 3.1.5 การออกแบบระบบ BCI

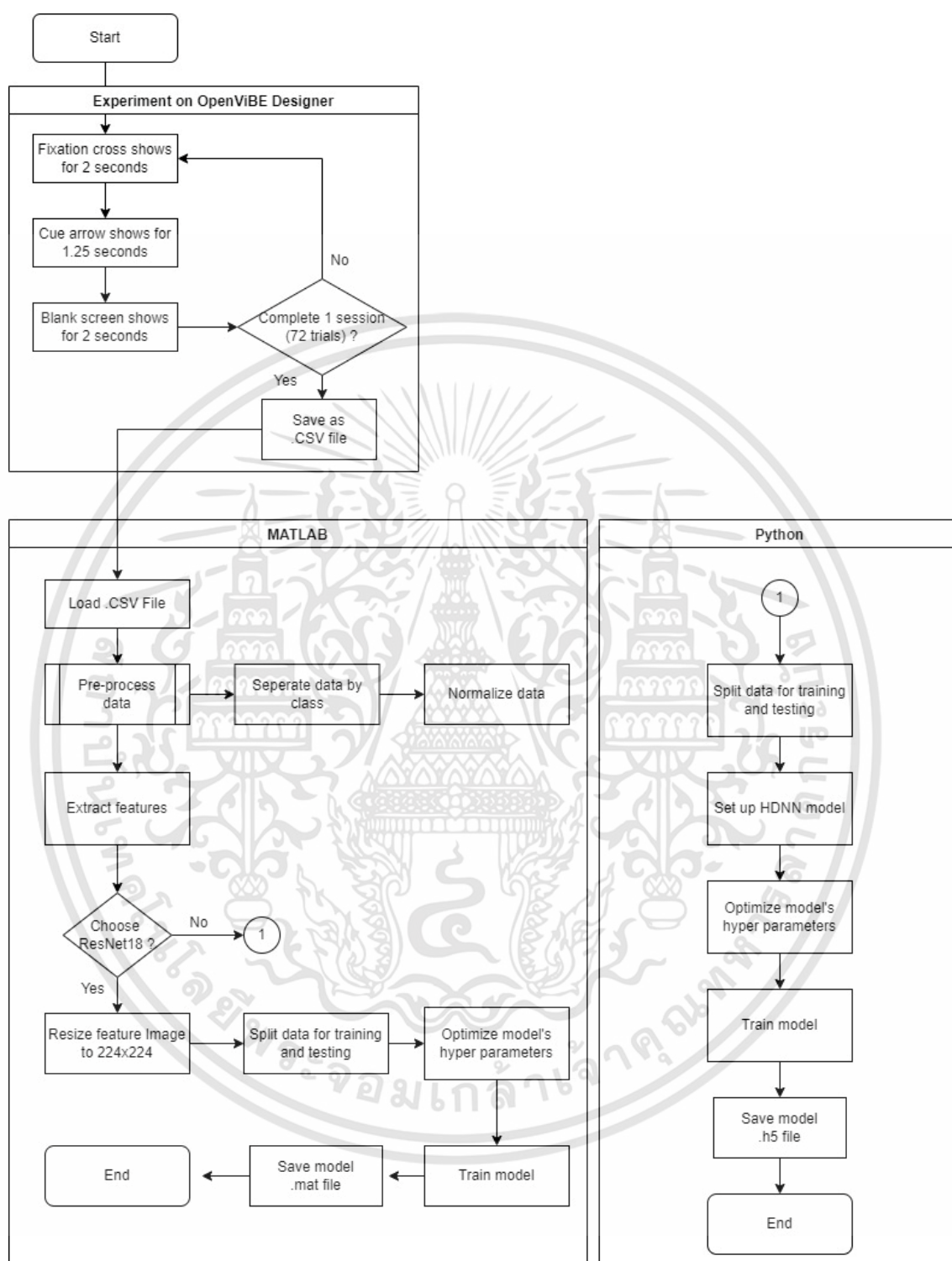
ระบบสำหรับการทดลองแบบ System Training แสดงดังบล็อกไดอะแกรมในรูปที่ 3.47 ซึ่งประกอบไปด้วยการเก็บสัญญาณคลื่นไฟฟ้าจากสมองด้วยอุปกรณ์ Unicorn Hybrid Black ผ่าน Lab Streaming Layer (LSL) ไปยัง Acquisition Client ในโปรแกรม OpenViBE Designer หลังจากเริ่มการทดลองในกระบวนการทดสอบ จะมีการส่ง Stimulus Trigger ซึ่งเป็นสัญญาณที่บอกถึง Event ที่เกิดขึ้นในการทดลองควบคู่กับการบันทึกสัญญาณคลื่นไฟฟ้าจากสมอง โดยจะจัดเก็บเป็นไฟล์ .CSV เพื่อนำไปประมวลผลสัญญาณเบื้องต้นในโปรแกรม MATLAB และนำสัญญาณที่ผ่านการประมวลผลไปทำการแยกคุณลักษณะโดยใช้ CWT และ CSP และฝึกสอนโมเดลการเรียนรู้เชิงลึกด้วยโครงสร้าง ResNet18 และ HDNN โดยโมเดลที่ฝึกสอนจะถูกนำไปใช้ในการทดลองแบบ Real-Time System Testing ดังรายละเอียดในหัวข้อที่ 3.1.9

แผนผังแสดงการทำงานของระบบการทดลองแบบ System Training สามารถเขียนได้ดังรูปที่ 3.48 เมื่อเริ่มต้นการทดลอง ช่วงแรกคือส่วนของการแสดงกระบวนการทดสอบใน 1 การทดลองย่อย โดยรายละเอียดขั้นตอนคือ เริ่มต้นหน้าจอของ Fixation Cross จะแสดงบนหน้าจอ 2 วินาที หลังจากนั้นหน้าจอของ Cue จะแสดงต่อเป็นเวลา 1.25 วินาที เมื่อสิ้นสุดหน้าจอ Cue หน้าจอสีดำจะปรากฏเป็นเวลา 2 วินาที ตรวจสอบเงื่อนไขว่าทำการแสดงหน้าจอครบ 1 Session หรือ 72 การทดลองย่อยหรือไม่ หากไม่ครบให้กลับไปทำขั้นตอนเริ่มต้นแสดงหน้าจออีก เมื่อครบเงื่อนไขระบบจะบันทึกข้อมูลเป็นไฟล์ .csv จากนั้นทำการโหลดข้อมูลที่ได้จากขั้นตอนบันทึกสัญญาณ ทำการแบ่งจำนวนของข้อมูลสำหรับการฝึกสอน (Train Data) และข้อมูลสำหรับการทดสอบ (Test Data) แบบสุ่ม 80% และ 20% ตามลำดับ และทำการหา Hyperparameter ที่ดีที่สุดด้วยวิธี Bayesian Optimization โดยมีเป้าหมายในการ Optimize คือการหาค่าความแม่นยำของชุดข้อมูลสำหรับการทดสอบความถูกต้องขณะฝึกสอน (Validation Accuracy) มากที่สุด และนำโมเดลสุดท้ายไปใช้ในระบบ Real-Time System ต่อไป



รูปที่ 3.47 บล็อกไดอะแกรมของระบบการทดลองแบบ System Training

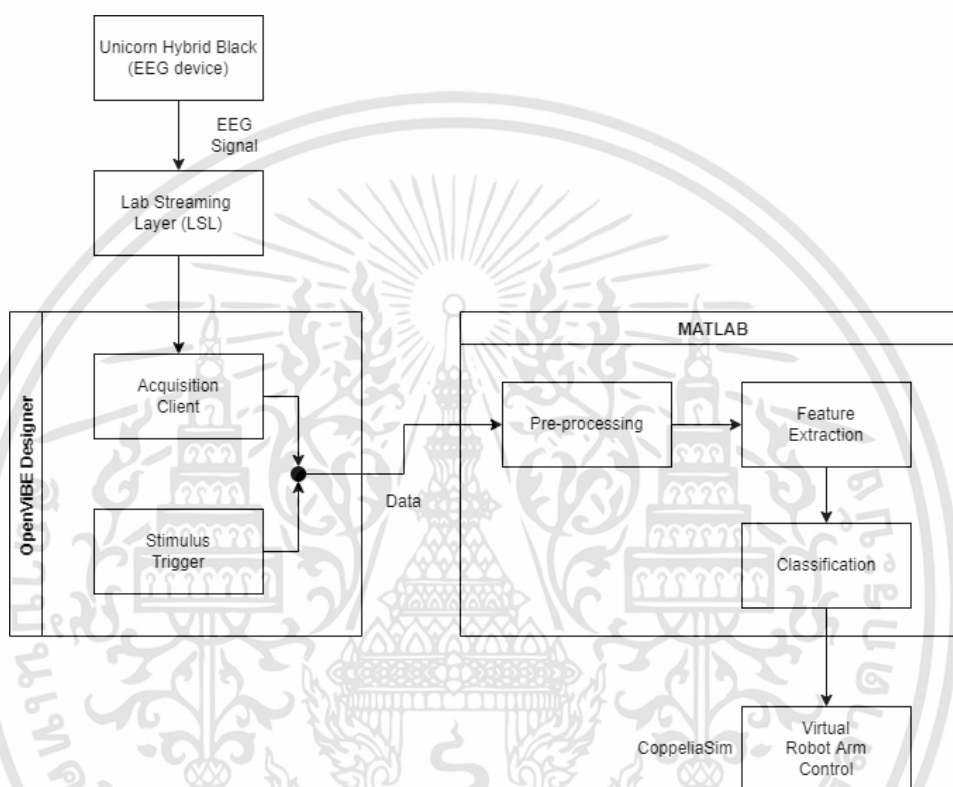
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



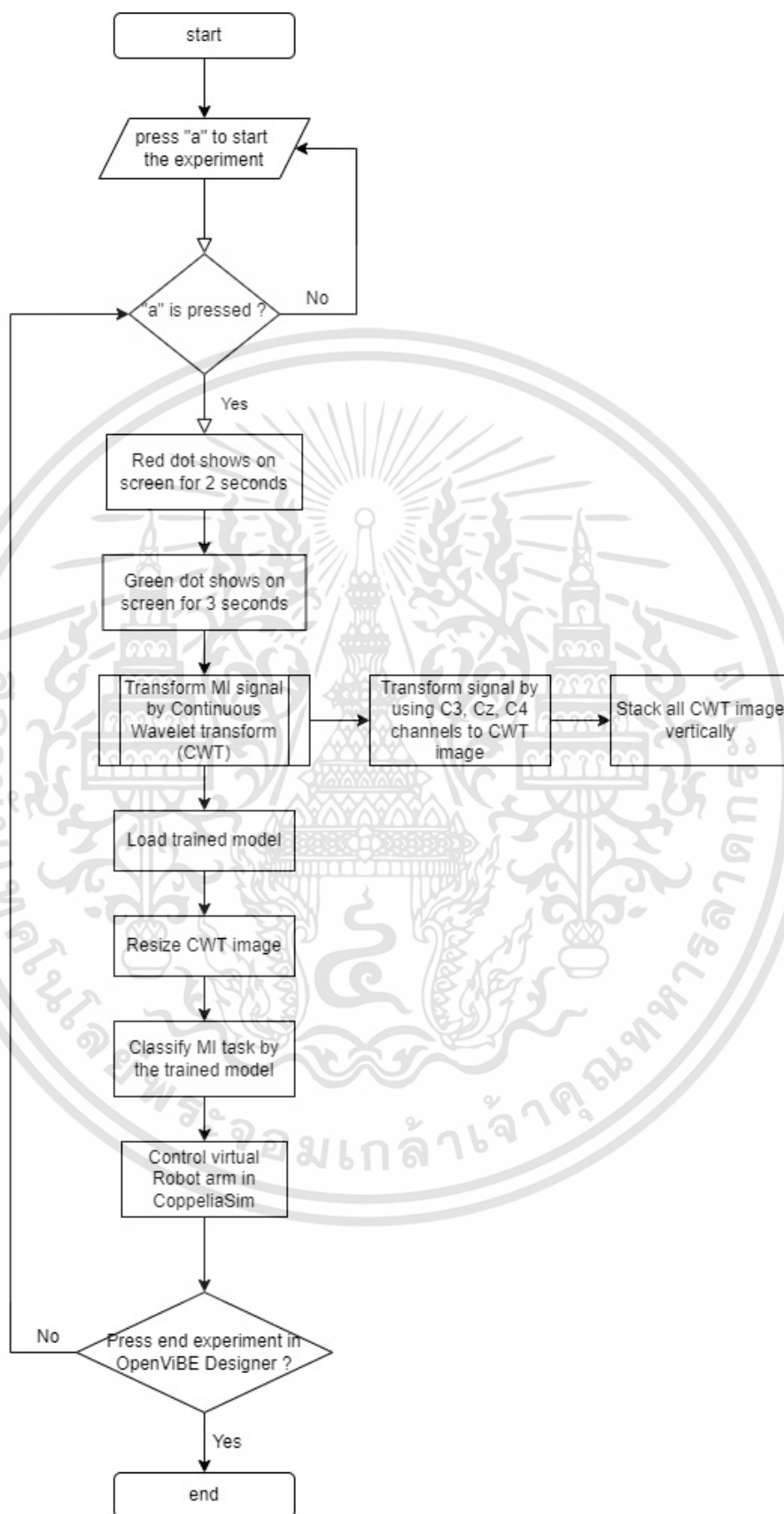
รูปที่ 3.48 แผนผังการทำงานของระบบการทดลองแบบ System Training

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.49 แสดงระบบการทดลองแบบ System Testing โดยนำโมเดลที่ได้บันทึกจากรูปที่ 3.48 มาทำการจำแนกประเภทของคุณลักษณะและนำผลที่ได้ไปทำการควบคุมแขนกลจำลองในโปรแกรม CoppeliaSim โดยแผนผังแสดงการทำงานของระบบการทดลองแบบ System Testing สามารถเขียนได้ดังรูปที่ 3.50



รูปที่ 3.49 บล็อกไดอะแกรมของระบบการทดลองแบบ System Testing



รูปที่ 3.50 แผนผังการทำงานของระบบการทดลองแบบ System Testing

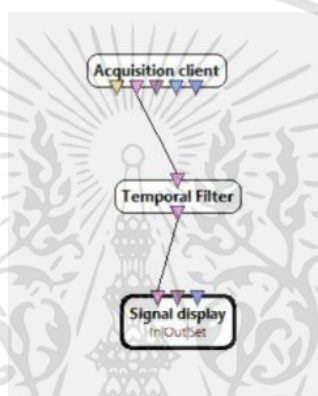
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.6 การออกแบบการทดลองบน OpenViBE Designer

#### 3.1.6.1 การออกแบบการทดลองเพื่อบันทึกและแสดงผลสัญญาณ

การออกแบบการทดลองเบื้องต้นเพื่อบันทึกและแสดงผลสัญญาณคลื่นไฟฟ้าจากสมองแบบ Real-Time สามารถออกแบบได้ดังรูปที่ 3.51 ซึ่งประกอบไปด้วย

1. Acquisition Client: รับสัญญาณที่ได้จาก Acquisition Server
2. Temporal Filter: กรองสัญญาณในช่วงความถี่ 1-30 Hz
3. Signal Display: หน้าต่างแสดงผลสัญญาณ



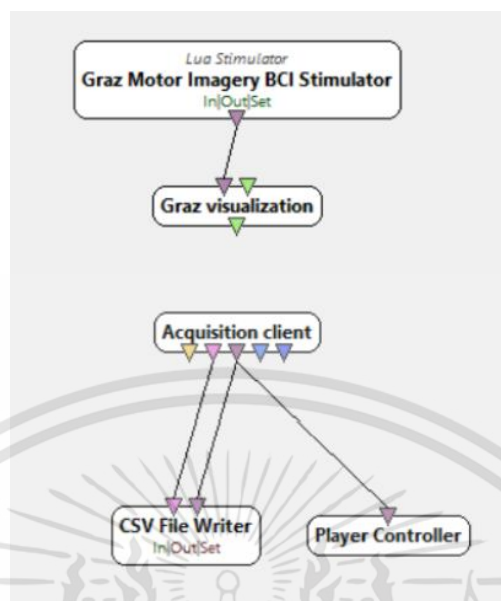
รูปที่ 3.51 การทดลองเบื้องต้นเพื่อบันทึกและแสดงผลคลื่นไฟฟ้าจากสมอง

#### 3.1.6.2 การออกแบบการทดลองสำหรับ System Training

รูปที่ 3.52 แสดงการออกแบบการทดลองที่ใช้สำหรับ System Training ซึ่งประกอบไปด้วย

1. Lua Stimulator: ทำการสร้าง Stimulation โดยอิงจากภาษา Lua ที่ผู้จัดทำเขียนขึ้น โดยกล่องเครื่องมือนี้จะทำการถอดรหัสสัญญาณที่เข้ามาและจะทำการเข้ารหัสสัญญาณพร้อมกับตัว Stimulation ออกไป [54]
2. Graz Visualization: แสดงหน้าจอโดยอ้างอิงจากกระบวนการที่ค้นของ Graz BCI ซึ่งใช้สำหรับการทดลอง BCI Motor Imagery โดยจะเป็นการแสดงผลสัญญาณลูกศรซ้ายหรือขวาเพื่อให้ผู้ทดลองทำตาม [55]
3. Acquisition Client: รับข้อมูลสัญญาณที่ส่งผ่านมาจาก Acquisition Server
4. CSV File Writer: ทำการเขียนไฟล์ข้อมูลทั้ง Stimulation และสัญญาณคลื่นไฟฟ้าจากสมองในรูปแบบไฟล์ .csv [56]
5. PLayer Controller: เพิ่มการตั้งค่าเพื่อควบคุม pLayer ต่างๆ เช่น Graz Visualization [57]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.52 การออกแบบการทดลองสำหรับ System Training

### 3.1.7 การออกแบบกระบวนการทดลองแบบ System Training

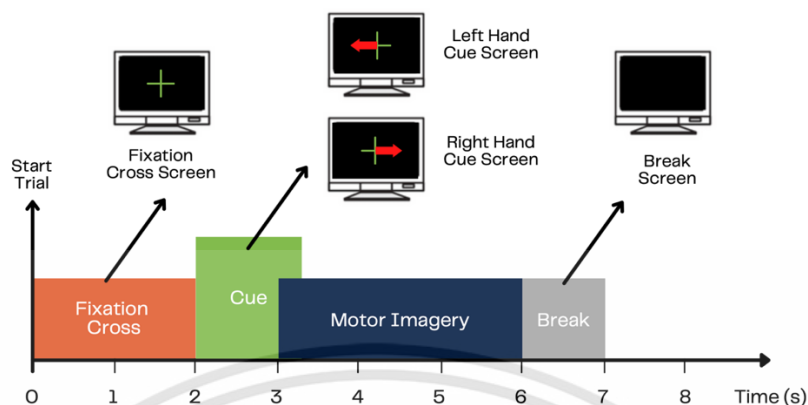
การศึกษาชุดข้อมูลออนไลน์ของคลื่นไฟฟ้าจากสมองนั้นได้ใช้ชุดข้อมูลของ BCI Competition IV 2a เป็นหลักในการทำการประมวลผลสัญญาณ, การแยกคุณลักษณะสัญญาณ และการจำแนกสัญญาณ ดังนั้นการออกแบบกระบวนการทดลองแบบ System Training จึงกำหนดรายละเอียดการทดลองโดยอ้างอิงจากชุดข้อมูลดังกล่าวซึ่งมีรายละเอียดดังนี้

- จำนวนของ MI Class: 2 Class โดยเป็นการจินตนาการการขยับมือซ้ายและมือขวา
- จำนวนของวันที่ทำการทดลอง: 2 วัน
- จำนวนของการทดลองย่อย: 12 การทดลองย่อย
- จำนวนของ Session: 6 Session

หน้าจอและช่วงเวลาของกระบวนการทดลองในแต่ละการทดลองย่อย แสดงดังรูปที่ 3.53 [58] โดยรายละเอียดของแต่ละรอบเวลามีดังนี้

- Fixation Cross: หน้าจอปรากฏภาพเครื่องหมายบวกเพื่อบ่งบอกว่าเริ่มการทดลองย่อย
- Cue: หน้าจอปรากฏภาพลูกศรเพื่อบ่งบอกว่าให้เริ่มจินตนาการ
- Motor Imagery: ช่วงการจินตนาการ MI
- Break: หน้าจอปรากฏภาพสีดำเพื่อบ่งบอกว่าจบการทดลองย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.53 กรอบเวลาของกระบวนการทศน์แบบ System Training ใน 1 การทดลองย่อย

### 3.1.8 การเขียนโปรแกรมสำหรับกระบวนการทศน์การทดลองแบบ System Training

ผู้จัดทำเขียนฟังก์ชันสำหรับกระบวนการทศน์การทดลองแบบ System Training ในแต่ละการทดลองย่อย ดังรูปที่ 3.54 ด้วยภาษา Lua ในโปรแกรม OpenViBE โดยในแต่ละส่วนจะมี Stimulation Trigger ต่างกันเพื่อระบุในไฟล์ที่บันทึกข้อมูลว่า Index ไหนเป็น Event ประเภทอะไร [59] รายละเอียดในแต่ละส่วนแสดงดังนี้

(a) ส่วนเริ่มการทดลอง : ผู้ทดลองจะนั่งอยู่นิ่งๆ โดยจะเห็นหน้าจอสีดำเป็นเวลา 20 วินาที ดังรูปที่ 3.55 โดยมี Simulation Trigger คือ

- OVTK\_StimulationID\_ExperimentStart คือ เริ่มการทดลอง มีเลขเป็น 32769
- OVTK\_StimulationID\_BaselineStart คือ เริ่ม Baseline มีเลขเป็น 32775
- OVTK\_StimulationID\_BaselineStop คือ สิ้นสุด Baseline มีเลขเป็น 32776
- OVTK\_StimulationID\_Beep คือ เสียง Beep มีเลขเป็น 33282

(b) ส่วนเริ่มการทดลองย่อย : ผู้ทดลองจะเห็นเครื่องหมายบวกสีเขียวกลางหน้าจอเป็นเวลา 2 วินาที ตัวอย่างหน้าจอแสดงดังรูปที่ 3.56 เพื่อบอกผู้ทดลองว่าจะเริ่มทำการทดลอง โดยมี Simulation Trigger คือ

- OVTK\_GDF\_Start\_Of\_Trial คือ เริ่มการทดลองแต่ละการทดลองย่อย มีเลขเป็น 768
- OVTK\_GDF\_Cross\_On\_Screen คือ แสดงเครื่องหมายบวกสีเขียวบนหน้าจอ มีเลขเป็น 786

(c) ส่วนเริ่ม MI Task : ผู้ทดลองจะเห็นลูกศรสีแดงบนหน้าจอเป็น 1.25 วินาที เพื่อกำหนดลักษณะการจินตนาการ ลูกศรหันหัวไปทางซ้ายหมายถึงให้จินตนาการการขยับมือซ้ายและลูกศรหันหัวไปทางขวาหมายถึงให้จินตนาการการขยับมือขวา ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลูกศรจะปรากฏขึ้นแบบสุ่มในแต่ละการทดลองย่อย ตัวอย่างของหน้าจอแสดงดังรูปที่ 3.57 โดยมี Simulation Trigger คือ

- OVTK\_GDF\_Left คือ ลูกศรสีแดงที่หันหัวไปทางซ้าย (Task มือซ้าย) มีเลขเป็น 769
- OVTK\_GDF\_Right คือ ลูกศรสีแดงที่หันหัวไปทางขวา (Task มือขวา) มีเลขเป็น 770

(d) ส่วนการจินตนาการ MI Task : ผู้ทดลองจะต้องจินตนาการการขยับมือตามทิศทางของหัวลูกศรเป็นเวลา 3.75 วินาที โดยมี Simulation Trigger คือ

- OVTK\_GDF\_Feedback\_Continuous คือ ให้จินตนาการการขยับมือ มีเลขเป็น 781

(e) ส่วนจบการทดลองย่อย : ผู้ทดลองจะเห็นหน้าจอสีดำซึ่งเป็นการบ่งบอกว่าได้จบการทดลองย่อยเป็นเวลา 2 วินาที โดยมี Simulation Code คือ

- OVTK\_GDF\_End\_Of\_Trial คือ จบการทดลองแต่ละการทดลองย่อย มีเลขเป็น 800

(f) ส่วนจบการทดลอง : ผู้ทดลองจะเห็นหน้าจอสีดำดังรูปที่ 3.55 เป็นเวลา 5 วินาที เพื่อบ่งบอกว่าเป็นการจบการทดลองในแต่ละ Session โดยมี Simulation Trigger คือ

- OVTK\_GDF\_End\_Of\_Session คือ จบการทดลองแต่ละ Session มีเลขเป็น 1010
- OVTK\_StimulationID\_ExperimentStop คือ จบการทดลอง มีเลขเป็น 32770

จากการออกแบบระบบในหัวข้อที่ 3.1.7 ส่วนของ (b) – (e) จะรันทั้งหมด 12 ครั้งเพื่อให้ได้ข้อมูลรวมทั้งหมด 12 การทดลองย่อย

```

function process(box)

    local t=0

    -- manages baseline

    box:send_stimulation(1, OVTk_StimulationId_ExperimentStart, t, 0)
    t = t + 5

    box:send_stimulation(1, OVTk_StimulationId_BaselineStart, t, 0)
    box:send_stimulation(1, OVTk_StimulationId_Beep, t, 0)
    t = t + baseline_duration

    box:send_stimulation(1, OVTk_StimulationId_BaselineStop, t, 0)
    box:send_stimulation(1, OVTk_StimulationId_Beep, t, 0)
    t = t + 5

    -- manages trials

    for i = 1, number_of_trials*2 do

        -- first display cross on screen
        box:send_stimulation(1, OVTk_GDF_Start_Of_Trial, t, 0)
        box:send_stimulation(1, OVTk_GDF_Cross_On_Screen, t, 0)
        t = t + 2

        -- display cue
        box:send_stimulation(1, sequence[i], t, 0)
        t = t + 1.25

        -- provide feedback
        box:send_stimulation(1, OVTk_GDF_Feedback_Continuous, t, 0)
        t = t + 3.75

        -- ends trial

        box:send_stimulation(1, OVTk_GDF_End_Of_Trial, t, 0)
        t = t + 2
    end

    -- send end for completeness
    box:send_stimulation(1, OVTk_GDF_End_Of_Session, t, 0)
    t = t + 5

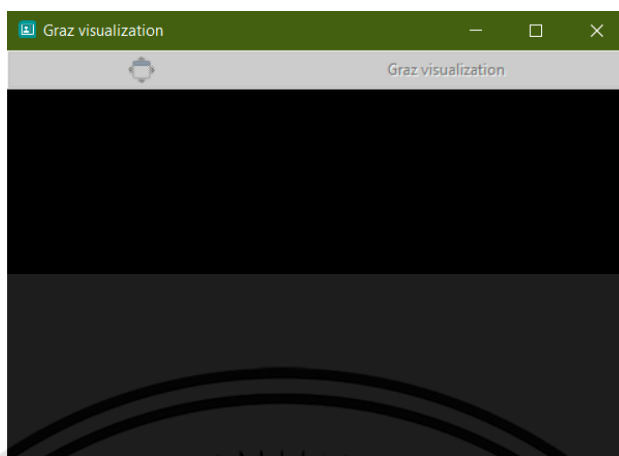
    -- used to cause the acquisition scenario to stop
    box:send_stimulation(1, OVTk_StimulationId_ExperimentStop, t, 0)

end

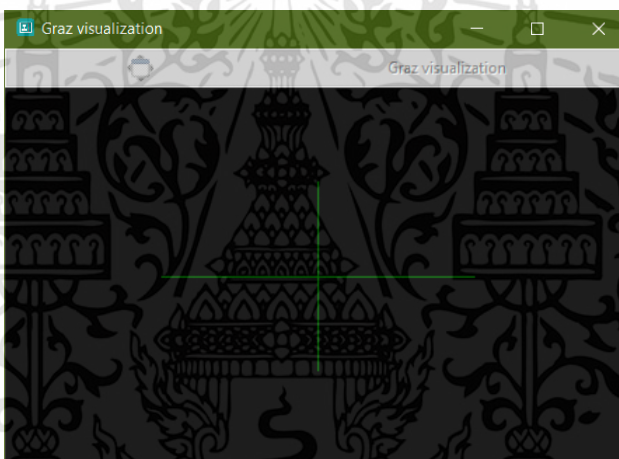
```

รูปที่ 3.54 คำสั่งที่ใช้ในการทดลองเก็บคลื่นไฟฟ้าจากสมอง

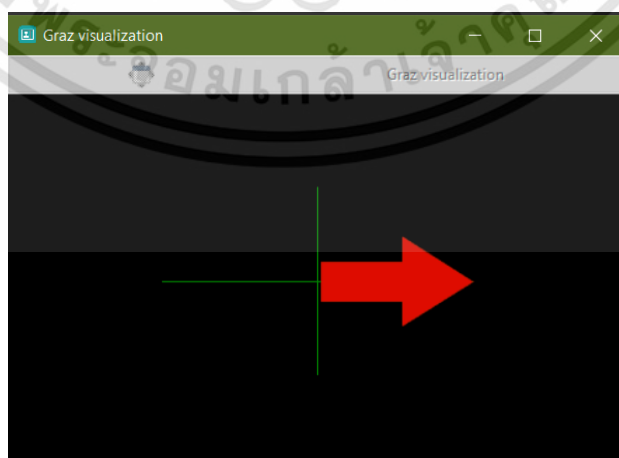
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.55 หน้าจอสีดำ



รูปที่ 3.56 หน้าจอรูปเครื่องหมายบวก



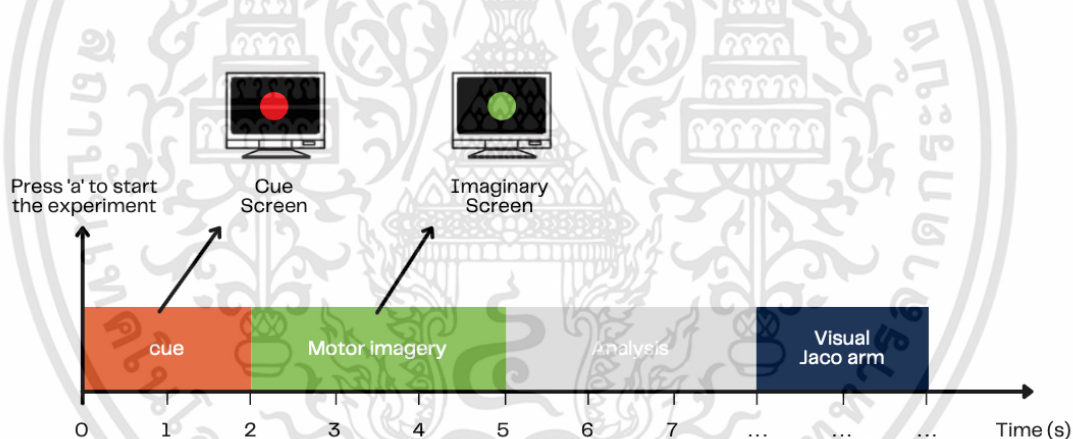
รูปที่ 3.57 หน้าจอแสดงตัวอย่างของลูกศร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.9 การออกแบบกระบวนการทดสอบแบบ System Testing

ผู้จัดทำได้ออกแบบกระบวนการทดสอบแบบ System Testing ใน 1 การทดลองย่อย ดังรูปที่ 3.58 โดยมีรายละเอียดขั้นตอนของแต่ละกรอบเวลาดังนี้

1. กดปุ่ม a บนคีย์บอร์ดเพื่อเริ่มทำการทดลองใน 1 การทดลอง
2. หลังจากกดปุ่ม a เริ่มกรอบเวลาของ Cue โดยที่หน้าจอจะแสดงจุดตรงกลางสีแดง เป็นเวลา 2 วินาทีเพื่อบ่งบอกให้ผู้ทดลองเตรียมทำการจินตนาการการขยับมือ
3. เริ่มกรอบเวลาของ Motor Imagery โดยหน้าจอจะแสดงจุดตรงกลางสีเขียวเป็นเวลา 3 วินาทีเพื่อบ่งบอกให้ผู้ทดลองเริ่มทำการจินตนาการการขยับมือ ซึ่งผู้ทดลองสามารถจินตนาการขยับมือซ้ายหรือมือขวาก็ได้
4. กรอบเวลาของ Analysis โดยเป็นช่วงที่ระบบทำการประมวลผลสัญญาณ, การแยกคุณลักษณะสัญญาณ และการจำแนกสัญญาณเพื่อส่งผลลัพธ์ไปยัง CoppeliaSim
5. กรอบเวลาของ Visual Jaco Arm โดยเป็นช่วงที่นำผลลัพธ์จากการจำแนกสัญญาณมาขยับแขนกล Jaco Arm 2



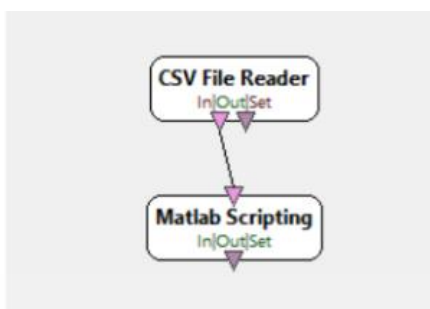
รูปที่ 3.58 กรอบเวลาของกระบวนการทดสอบแบบ System Testing ใน 1 การทดลองย่อย

### 3.1.10 การเขียนโปรแกรมสำหรับกระบวนการทดสอบแบบ System Testing

#### 3.1.10.1 ระบบ Offline System Testing สำหรับข้อมูล BCI Competition

IV 2a

โปรแกรมการทดสอบ Offline System Testing ซึ่งออกแบบใน OpenViBE Designer แสดงรูปที่ 3.59 ซึ่งประกอบไปด้วยบล็อก CSV File Reader ใช้ในการอ่านไฟล์ CSV และบล็อก Matlab Scripting ใช้ในการเขียนฟังก์ชันในการควบคุมระบบโดยรวมตามหัวข้อที่ 3.1.9



รูปที่ 3.59 บล็อกที่ใช้ในระบบ Offline System Testing สำหรับข้อมูล BCI Competition IV 2a

การเตรียมข้อมูลหนึ่งการทดลองย่อยใน CSV File Reader สามารถทำได้โดยต้องให้อยู่ใน Format ที่ OpenViBE Designer กำหนดไว้ โดยคำสั่งที่ใช้ในการเตรียมข้อมูลแสดงดังรูปที่ 3.60 และเมื่อเปิดด้วยโปรแกรม Excel จะแสดงดังรูปที่ 3.61

คำสั่งที่ใช้ในการควบคุมในบล็อก Matlab Scripting แสดงดังรูปที่ 3.62 ซึ่งประกอบไปด้วยฟังก์ชันย่อยคือ cwt\_bci, predict และ control\_jaco ตามลำดับ โดยฟังก์ชัน cwt\_bci จะทำการแปลงสัญญาณ sig\_x ซึ่งเป็นสัญญาณที่อ่านจาก CSV File Reader ด้วยวิธี CWT ซึ่งผลที่ได้จะเป็นรูปภาพของ Channel C3, Cz, C4 ที่มีการเรียงต่อกันในแนวตั้งตามลำดับและทำการบันทึกไว้ในโพลเดอร์ที่ผู้จัดทำกำหนดไว้ โดยฟังก์ชัน predict จะนำเข้าโมเดลที่ถูกฝึกสอนไว้แล้วสำหรับ Subject นี้มาใช้ในการจำแนกรูปภาพดังกล่าว ฟังก์ชัน predict จะส่งค่ากลับคือ Task โดยถ้าผลการจำแนกคือ Right ค่าที่ส่งกลับมาจะเท่ากับ 0 และ 1 ถ้าผลการจำแนกคือ Left หลังจากจำแนกเรียบร้อยแล้วจะส่งเข้าสู่ฟังก์ชัน control\_jaco ดังรายละเอียดในหัวข้อที่ 3.1.12 ซึ่งจะทำการเชื่อมต่อ API ไปยังโปรแกรม CoppeliaSim และทำการควบคุมแขนกลจำลอง Jaco Arm 2 ในทิศทางที่กำหนดโดยคำสั่งสำหรับฟังก์ชัน cwt\_bci และ predict แสดงดังรูปที่ 3.63 และ 3.64 ตามลำดับ

```

%%
Time_keep = ((0:750-1)/250).';
Epoch_keep = zeros(750, 1);
%%
load('lk.mat');
tr = 2;
Ch1 = left_keep(tr, :, 1).';
Ch2 = left_keep(tr, :, 2).';
Ch3 = left_keep(tr, :, 3).';
%%
load('rk.mat');
tr = 2;
Ch1 = right_keep(tr, :, 1).';
Ch2 = right_keep(tr, :, 2).';
Ch3 = right_keep(tr, :, 3).';
%%
E_id = zeros(750, 1);
E_date = zeros(750, 1);
E_du = zeros(750, 1);
%%
T = table(Time_keep, Epoch_keep, Ch1, Ch2, Ch3, E_id, E_date, E_du);
T.Properties.VariableNames = {'Time:250Hz', 'Epoch', 'Channel 1', 'Channel 2', 'Channel 3', 'Event Id', ...
    'Event Date', 'Event Duration'};
writetable(T, 'test.csv');
  
```

รูปที่ 3.60 คำสั่งที่ใช้ในการเตรียมข้อมูลของหนึ่งการทดลองย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	A	B	C	D	E	F	G	H
1	Time:250Hz	Epoch	Channel 1	Channel 2	Channel 3	Event Id	Event Date	Event Duration
2	0	0	13.18359	4.6875	15.625	0	0	0
3	0.004	0	15.72266	6.103516	15.82031	0	0	0
4	0.008	0	10.69336	4.638672	13.81836	0	0	0
5	0.012	0	8.544922	8.300781	13.76953	0	0	0
6	0.016	0	-0.24414	2.001953	5.810547	0	0	0
7	0.02	0	-11.5723	-2.24609	-8.83789	0	0	0
8	0.024	0	-18.457	0.585938	-12.1582	0	0	0
9	0.028	0	-10.4492	10.74219	-7.71484	0	0	0
10	0.032	0	-4.78516	18.75	-6.78711	0	0	0
11	0.036	0	1.611328	24.56055	-1.70898	0	0	0
12	0.04	0	8.300781	25.83008	-0.04883	0	0	0
13	0.044	0	10.64453	22.16797	0.976563	0	0	0
14	0.048	0	7.568359	12.35352	-4.3457	0	0	0
15	0.052	0	6.982422	6.738281	-6.78711	0	0	0
16	0.056	0	3.027344	-2.68555	-9.08203	0	0	0
17	0.06	0	11.03516	0.830078	1.904297	0	0	0
18	0.064	0	19.87305	8.544922	15.82031	0	0	0
19	0.068	0	18.60352	5.322266	17.91992	0	0	0
20	0.072	0	18.11523	5.615234	25.39063	0	0	0
21	0.076	0	15.72266	1.220703	24.51172	0	0	0
22	0.08	0	4.785156	-9.22852	16.45508	0	0	0
23	0.084	0	3.710938	-11.7676	17.08984	0	0	0
24	0.088	0	1.318359	-16.2109	10.44922	0	0	0
25	0.092	0	-3.17383	-19.1406	1.5625	0	0	0
26	0.096	0	-2.83203	-18.8477	-4.98047	0	0	0

รูปที่ 3.61 ตัวอย่างข้อมูลจากหนึ่งการทดลองย่อย

```
function box_out = matlab_Process_bci(box_in)|
for i = 1: OV_getNbPendingInputChunk(box_in,1)
[box_in, start_time, end_time, matrix_data] = OV_popInputBuffer(box_in,1);
L = box_in.inputs{1}.header.nb_samples_per_buffer; % Length of signal
% Time vector
sig_x = matrix_data(:,1:L);
%%CWT_PART
cwt_bci(sig_x)
%%MODEL_PREDICTION_PART
task = predict();
%%ROBOT_PART
control_jaco(task);
end
box_out = box_in;
end
```

รูปที่ 3.62 คำสั่งที่ใช้ในการควบคุมระบบ Offline System Testing

```

function [] = cwt_bci(signal)
    for channel_idx=1:3
        usedChannel = {'c3','cz','c4'};
        cd('C:\Users\User\Desktop\2_project\OpenViBE');
        [cfs,frq] = cwt(squeeze(signal(channel_idx,:)), 250);
        time = (0:750-1);
        t = time/250;
        pcolor(t, frq, abs(cfs));shading 'interp';
        set(gca,'xtick',[],'ytick',[]);
        colormap bone;
        set(gca,'LooseInset',get(gca,'TightInset'));
        ylim([8 30]);
        exportgraphics(gca, sprintf('%s.jpg',usedChannel{channel_idx}));
    end
    f1 = imread(sprintf('%s.jpg',usedChannel{1}));
    f2 = imread(sprintf('%s.jpg',usedChannel{2}));
    f3 = imread(sprintf('%s.jpg',usedChannel{3}));
    figx = [f1;f2;f3];
    cd('C:\Users\User\Desktop\2_project\OpenViBE');
    imwrite(figx, 'test.jpg');
end

```

รูปที่ 3.63 ฟังก์ชัน cwt\_bci ที่ใช้ในการแปลง CWT

```

function [task] = predict()
    net = load('Model.mat');
    net = net.net;
    img = imresize(imread('C:\Users\User\Desktop\2_project\OpenViBE\Test.jpg'), ...
    [224 224]);
    predicted = classify(net,img);
    if predicted == 'Right'
        task = 0;
    else
        task = 1;
    end
    disp(predicted)
end

```

รูปที่ 3.64 ฟังก์ชัน predict คำสั่งที่ใช้ในการจำแนกรูปภาพ CWT

### 3.1.10.2 ระบบ Real-Time System Testing สำหรับคลื่นไฟฟ้าจากสมอง

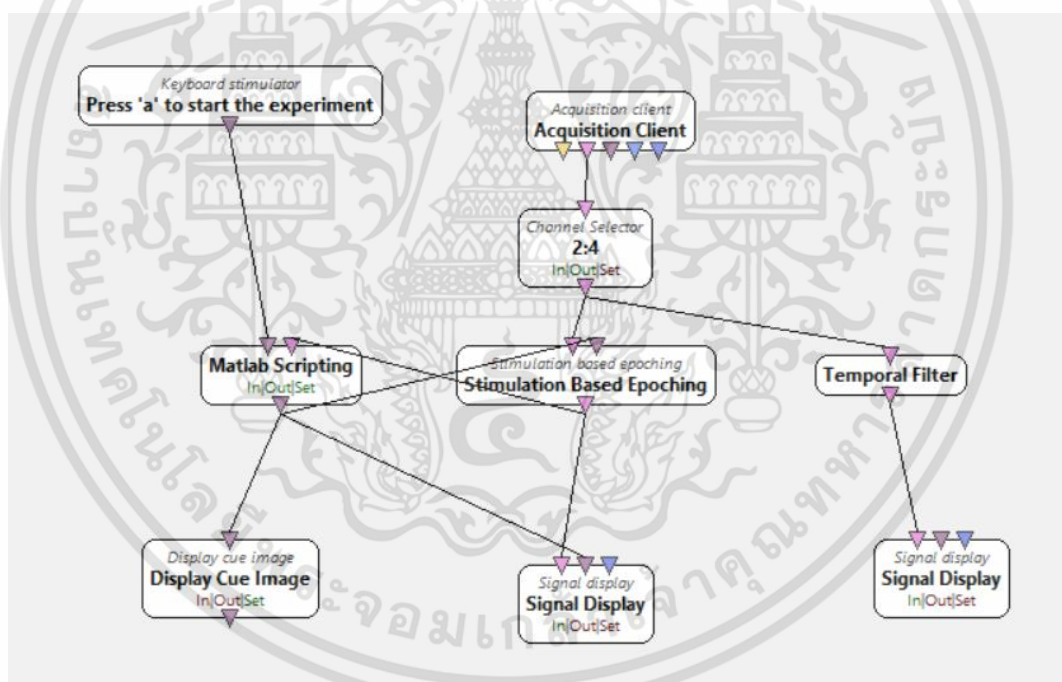
ผู้จัดทำ

โปรแกรมการทดสอบ Real-Time System Testing ซึ่งออกแบบใน OpenViBE Designer แสดงดังรูปที่ 3.65 โดยบล็อกที่ใช้ในระบบมีหน้าที่ต่างๆ ดังนี้

1. Keyboard Stimulator: ทำการรับการกดปุ่ม a บนคีย์บอร์ดเพื่อส่งสัญญาณ Stimulus Trigger ไปยังบล็อก Matlab Scripting
2. Acquisition Client: เป็นสื่อกลางระหว่าง Unicorn Hybrid Black ซึ่งเชื่อมต่ออยู่กับ LSL กับ OpenViBE Designer เพื่อนำคลื่นไฟฟ้าจากสมองมาใช้ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Channel Selector: ทำการเลือก Channel ที่ต้องการใช้ โดยผู้จัดทำเลือก แชนแนลที่ 2, 3 และ 4 ซึ่งตรงกับ Channel C3, Cz และ C4 ตามลำดับ
4. MATLAB Scripting: ใช้ในการเขียนฟังก์ชันในการควบคุมระบบโดยรวม
5. Stimulation Based Epoching: ใช้ในการกำหนดช่วงเวลาที่ต้องการใน สัญญาณผ่าน Stimulus Trigger ที่ส่งมา ซึ่งผู้จัดทำเลือกช่วงเวลาคือ 3 วินาทีหลังจากจุดสีเขียวขึ้นที่หน้าจอ
6. Display Cue Image: ทำการแสดง Cue ต่างๆตาม Stimulus Trigger ที่ ส่งมา โดยในระบบนี้มีจุดสีแดงและจุดสีเขียวเป็น cue ที่ใช้
7. Temporal Filter: ทำการกรองสัญญาณในช่วงความถี่ 1-30 Hz เพื่อใช้ในการ Visualize สัญญาณขณะทำการทดลอง
8. Signal Display: หน้าต่างแสดงสัญญาณ



รูปที่ 3.65 บล็อกที่ใช้ในระบบ Real-Time System Testing สำหรับข้อมูลคลื่นไฟฟ้าจากสมองผู้จัดทำ

คำสั่งที่ใช้ในการควบคุมในบล็อก Matlab Scripting แสดงดังรูปที่ 3.66 โดยรายละเอียดและขั้นตอนการทำงานของคำสั่งมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function box_out = matlab_Proces(box_in)

% we iterate over the pending chunks on input 2 (STIMULATIONS)
for i = 1: OV_getNbPendingInputChunk(box_in,1)
[box_in, start_time, end_time, stim_set] = OV_popInputBuffer(box_in,1);
if(numel(stim_set) >= 3) % at least one stim in the set.
    fprintf('Received stimulation code %i at time %f\n', stim_set(1), stim_set(2))
    if stim_set(1) == box_in.settings(1).value
        stim_x = [33025 box_in.clock 0];
        box_in = OV_addOutputBuffer(box_in,1,start_time,end_time,stim_x);
        box_in.user_data.trigger = true;
        box_in.user.time = stim_set(2);
    end
end

if(box_in.user_data.trigger)
    if box_in.clock == box_in.user.time+2
        stim_y = [33026 box_in.clock 0];
        box_in = OV_addOutputBuffer(box_in,1,start_time,end_time,stim_y);
        box_in.user_data.trigger = false;
    end
end

for i = 1: OV_getNbPendingInputChunk(box_in,2)
% we increment the matrix count
[box_in, start_time, end_time, matrix_data] = OV_popInputBuffer(box_in,2);
L = box_in.inputs(2).header.nb_samples_per_buffer; % Length of signal
sig_x = matrix_data(:,1:L);
%%CWT_PART
CWT_start = tic;
cwt_bci(sig_x)
CWT_end = toc(CWT_start);
%%MODEL_PREDICTION_PART
Model_start = tic;
task = predict();
Model_end = toc(Model_start);
%%ROBOT_PART
Robot_start = tic;
control_jaco(task);
Robot_end = toc(Robot_start);
fprintf('Time of CWT : %d \n',CWT_end)
fprintf('Time of Model Prediction : %d \n',Model_end)
fprintf('Time of Controlling : %d \n',Robot_end)
stim_y = [33027 box_in.clock 0];
box_in = OV_addOutputBuffer(box_in,1,start_time,end_time,stim_y);
end
box_out = box_in;
end

```

รูปที่ 3.66 คำสั่งที่ใช้ในการควบคุมระบบ Real-Time System Testing สำหรับคลื่นไฟฟ้าจากสมอง  
ผู้จัดทำ

1. คำสั่งจะทำการตรวจเช็คว่ามีกรกดปุ่ม a บนคีย์บอร์ดหรือไม่ หากมีการกดปุ่มดังกล่าวจะทำการส่ง Stimulus Trigger รหัส 33025 ซึ่งจะเป็นรหัสที่ใช้ในการแสดงจุดสีแดงบนหน้าจอในบล็อก Display Cue Image
2. ทำการตรวจเช็คจุดสีแดงบนหน้าจอแสดงครบ 2 วินาทีแล้วหรือยัง หากแสดงครบแล้วจะทำการส่ง Stimulus Trigger รหัส 33026 เพื่อแสดงจุดสีเขียวบนจอ และในขณะเดียวกันในบล็อก Stimulus Based Epoching

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะคอยตรวจเช็คหากมีการส่ง Stimulus Trigger รหัส 33026 มาที่บล็อก จะทำการเลือกสัญญาณ 3 วินาทีหลังจากนั้นและส่งสัญญาณดังกล่าว กลับไปยังบล็อก Matlab Scripting

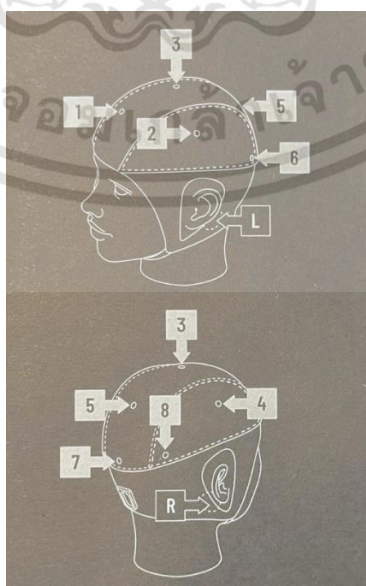
3. สัญญาณที่ส่งมาจะนำไปทำการแปลง CWT ผ่านฟังก์ชัน `cwt_bci` และนำรูปภาพที่จากการแปลงนำไปจำแนกด้วยฟังก์ชัน `predict` และนำผลการจำแนกไปทำการควบคุมแขนกลจำลองด้วยฟังก์ชัน `control_jaco` ดังรายละเอียดในหัวข้อที่ 3.1.12
4. ทำการส่ง Stimulus Trigger รหัส 33027 ซึ่งเป็นคำสั่งที่ใช้ในการลบจุดบนหน้าจอให้เป็นหน้าจอเปล่า
5. รอรับการกดปุ่ม a บนคีย์บอร์ดเพื่อเริ่มต้นการควบคุมครั้งต่อไป

### 3.1.11 การวัดสัญญาณและการเชื่อมต่ออุปกรณ์เข้าสู่ระบบ

#### 3.1.11.1 การเชื่อมต่อ Unicorn Hybrid Black เข้าสู่ระบบ

การเตรียมอุปกรณ์ Unicorn Hybrid Black เพื่อใช้ในการวัดสัญญาณและเชื่อมต่อเข้าสู่ระบบมีขั้นตอนดังนี้

1. นำ Electrode ของ Unicorn Brain Interface มาประกอบกับหมวก โดยให้หมายเลขของแต่ละ Electrode ตั้งแต่หมายเลข 1-8 ตรงกับตำแหน่งของแต่ละ Electrode บนหมวกดังรูปที่ 3.67 และ 3.68 โดยใช้วิธีสไลด์อย่างระมัดระวังและทำการติด Reference Electrode L และ R โดยใช้ Sticky Electrode ดังรูปที่ 3.69

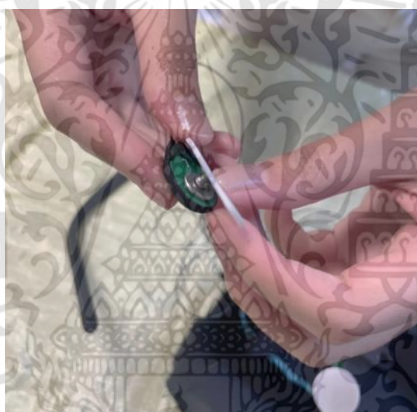


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.67 ตำแหน่งของแต่ละ Electrode บนหมวก



รูปที่ 3.68 การเชื่อมต่อ Electrode กับหมวก



รูปที่ 3.69 การเชื่อมต่อ Sticky Electrode กับ Reference Electrode

2. นำ Unicorn Brain Interface ส่วนอะแดปเตอร์เชื่อมต่อแม่เหล็กต่อกับด้านหลังหมวกและสวมหมวกไว้ที่ศีรษะของผู้ทดลองดังรูปที่ 3.70 แล้วทำการลือคส่วนของหมวกกับศีรษะโดยใช้สายรัดคางดังรูปที่ 3.71



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.70 ลักษณะการสวมหมวกที่ต่อ Unicorn Brain Interface ไว้ที่ศีรษะ



รูปที่ 3.71 การลือคหมวกกับศีรษะด้วยสายรัดคาง

3. ทำการติดส่วน Reference Electrode L และ R ที่ถูกติดกับ Sticky Electrode บริเวณหลังใบหูด้านซ้ายและด้านขวาดังรูปที่ 3.72 ทำการตรวจสอบว่าตำแหน่งของ Electrode ที่อยู่ตรงกลางตรงกับส่วนกลางของ ศีรษะและทุก Electrode ของหมวกสัมผัสหนังศีรษะ



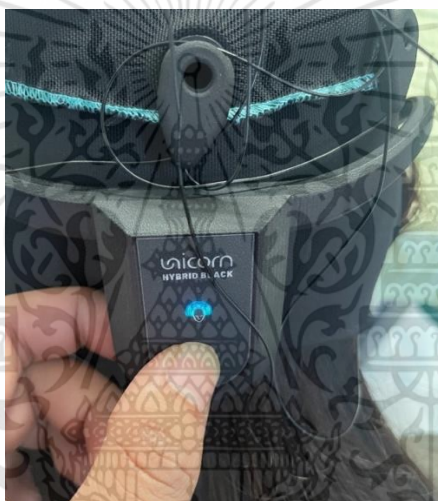
รูปที่ 3.72 การติด Sticky Electrode บริเวณหลังใบหู

4. เชื่อมต่อ Unicorn Brain Interface กับคอมพิวเตอร์ โดยเสียบ Bluetooth Dongle ที่คอมพิวเตอร์ดังรูปที่ 3.73 และเปิดเครื่อง Unicorn Brain Interface โดยกดค้ำที่ปุ่มเปิดไว้ประมาณ 2 วินาทีดังรูปที่ 3.74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

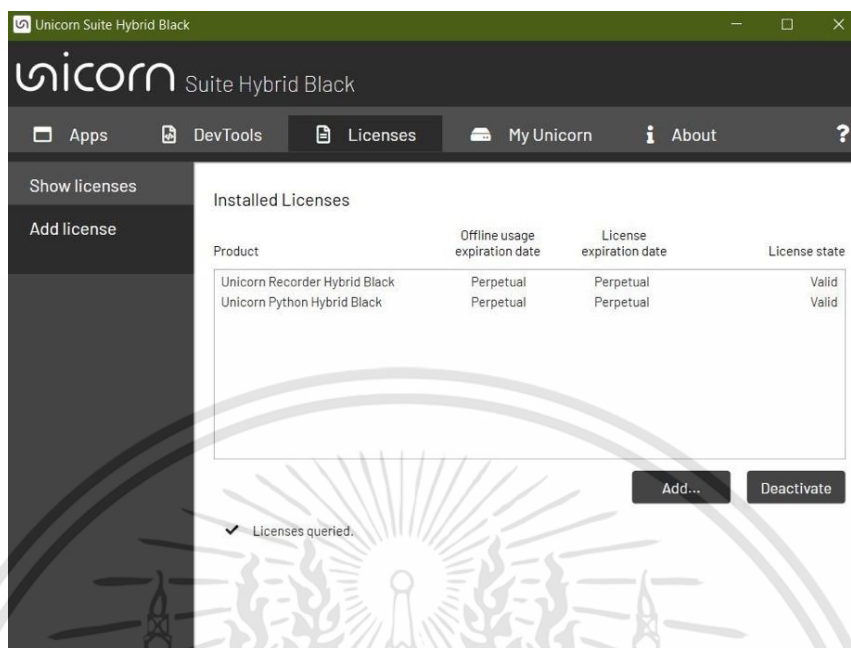


รูปที่ 3.73 การเชื่อมต่อ Bluetooth Dongle กับคอมพิวเตอร์

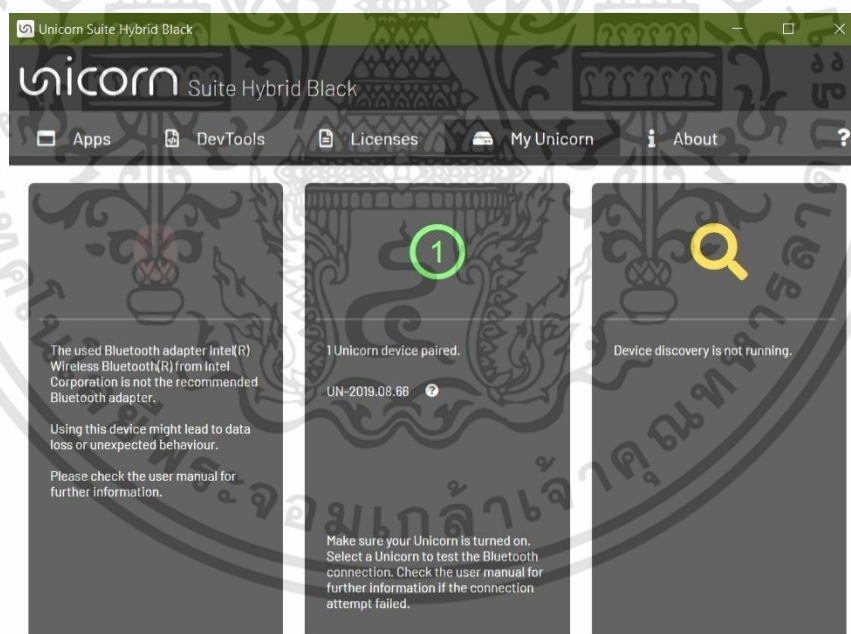


รูปที่ 3.74 การเปิดการทำงานของอุปกรณ์ Unicorn Brain Interface

5. เปิดโปรแกรม Unicorn Suite บนเครื่องคอมพิวเตอร์และทำการเพิ่ม License ของ Unicorn Recorder แล้วทำการ Activate อุปกรณ์ ดังรูปที่ 3.75 หลังจากนั้นไปที่หน้า My Unicorn เพื่อเชื่อมต่อ Bluetooth กับ Unicorn Brain Interface หากอุปกรณ์สามารถเชื่อมต่อกับคอมพิวเตอร์ได้ หมายเลขและชื่อของอุปกรณ์จะแสดงดังรูปที่ 3.76



รูปที่ 3.75 หน้าต่างในการเพิ่มและ Activate License บน Unicorn Suite



รูปที่ 3.76 หน้าต่าง My Unicorn บน Unicorn Suite

- สัญญาณคลื่นไฟฟ้าจากสมองในแต่ละ Electrode แสดงดังรูปที่ 3.77 โดยการแสดงผลสัญญาณสามารถเลือกช่วงสัญญาณ, ช่วงของความถี่แถบผ่าน และความถี่ที่ใช้กับ Notch Filter ซึ่งสัญญาณคลื่นไฟฟ้าจะเสถียรก็ต่อเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดสีเหลืองของแต่ละ Electrode เปลี่ยนเป็นสีเขียว ซึ่งใช้เวลาประมาณ 2-3 นาที



รูปที่ 3.77 หน้าต่าง Unicorn Recorder ที่ส่วนของการปรับการแสดงผลสัญญาณและ ส่วนของสัญญาณในแต่ละ Electrode โดยเรียงจาก Electrode 1-8 ตามลำดับ

7. ในการเพิ่มคุณภาพของสัญญาณสามารถใช้การเก็บคลื่นไฟฟ้าจากสมองแบบเปียกได้ โดยใช้ Unicorn Gel ซึ่งจะหยอด Unicorn Gel โดยใช้ไซริงดังรูปที่ 3.78 ลงบน Electrode ดังรูปที่ 3.79 เพื่อเป็นตัวนำให้ Electrode สัมผัสกับศีรษะได้ดีขึ้น



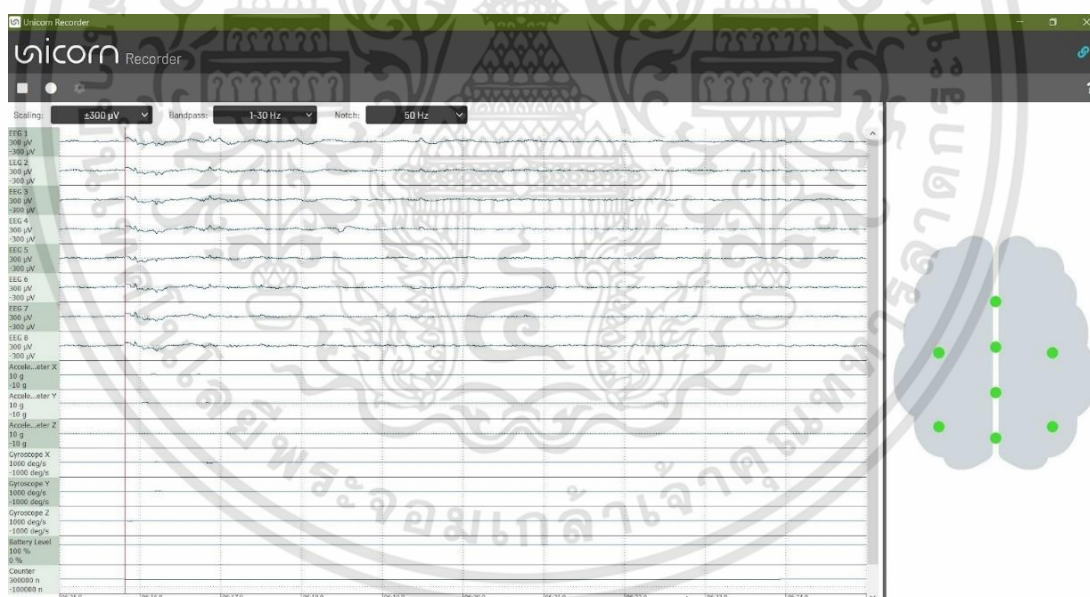
รูปที่ 3.78 ไซริงและ Unicorn Gel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.79 การหยอด Unicorn Gel ลงบน Electrode

8. รอให้สถานะของสัญญาณแต่ละ Electrode เป็นสีเขียวดังรูปที่ 3.80 โดยมีการปรับช่วงสัญญาณเป็นช่วงแอมพลิจูดที่  $\pm 300 \mu\text{V}$ , กรองความถี่เฉพาะช่วง 1-30 Hz และใช้ Notch Filter ที่ 50 Hz

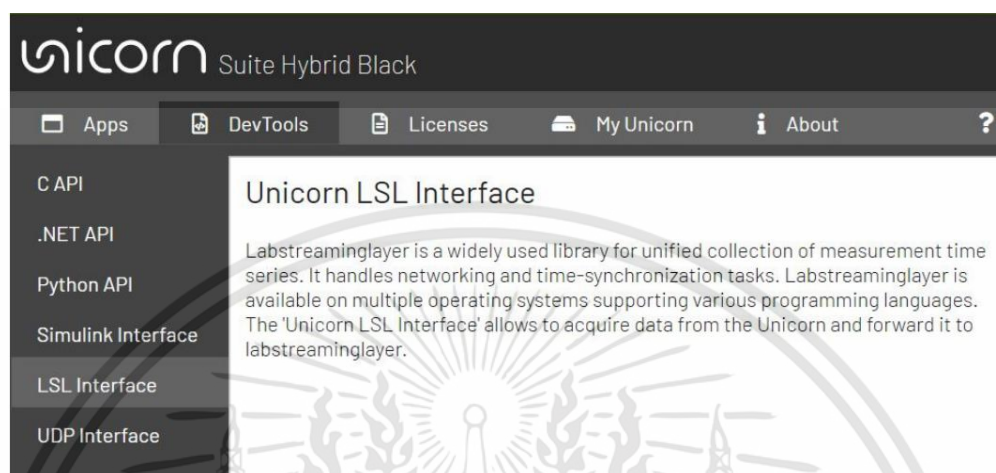


รูปที่ 3.80 หน้าต่าง Unicorn Recorder ของสัญญาณที่มีความเสถียรแล้ว

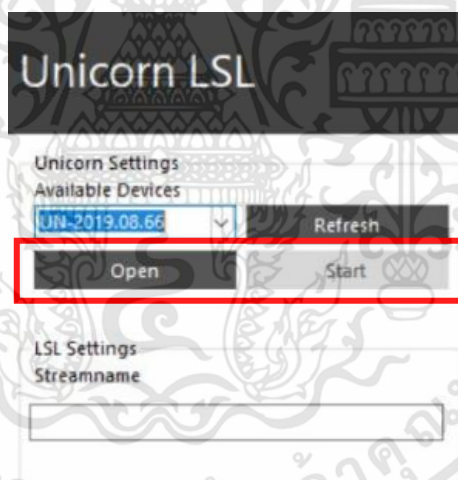
### 3.1.11.2 การเชื่อมต่อระหว่าง Unicorn Hybrid Black และ OpenViBE

การเชื่อมต่อระหว่างอุปกรณ์ Unicorn Hybrid Black และโปรแกรม OpenViBE ต้องทำการเชื่อมต่อผ่าน Lab Streaming Layer (LSL) โดยขั้นตอนการเชื่อมต่อจะต้องใช้โปรแกรม Unicorn Hybrid Black Suite ซึ่งเป็นโปรแกรมของอุปกรณ์ Unicorn Hybrid Black โดยตัวโปรแกรมนั้นจะมี Unicorn LSL Interface ดังรูปที่ 3.81 เป็นตัวสื่อกลางในการเชื่อมต่ออุปกรณ์เข้าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สู่ LSL เมื่อเปิด Unicorn LSL Interface จะพบหน้าต่างการเชื่อมต่อดังรูปที่ 3.82 โดยสามารถเชื่อมต่ออุปกรณ์เข้าสู่ LSL โดยทำการกดปุ่ม Open และ Start ตามลำดับ



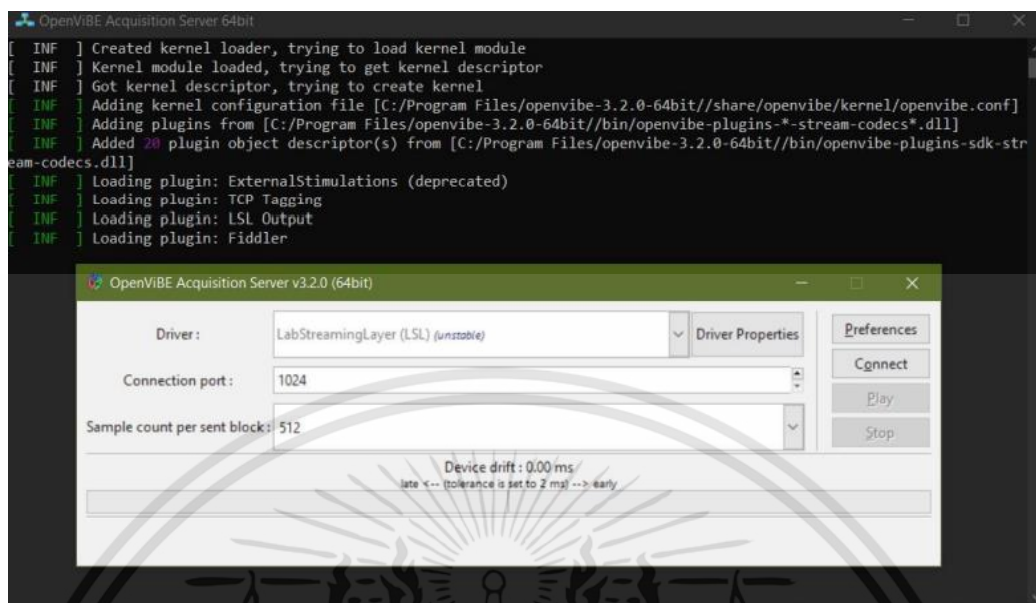
รูปที่ 3.81 LSL Interface



รูปที่ 3.82 หน้าต่างโปรแกรม Unicorn LSL

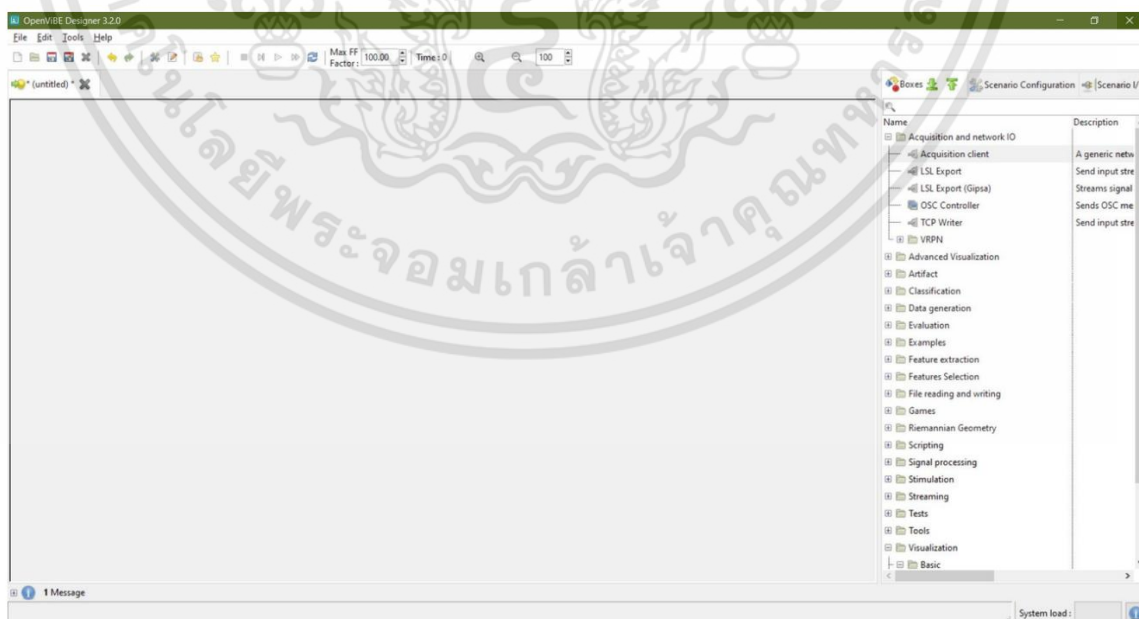
ขั้นตอนต่อไปจะเป็นการเชื่อมต่อ LSL เข้ากับโปรแกรม OpenViBE โดยสามารถเชื่อมต่อได้ผ่านโปรแกรม OpenViBE Acquisition Server ซึ่งจะเป็นตัวกลางที่จะรับอุปกรณ์ที่ส่งผ่าน LSL นำไปใช้ในโปรแกรมต่างๆ ของ OpenViBE โดยรูปที่ 3.83 แสดงตัวอย่างของโปรแกรม OpenViBE Acquisition Server โดยเลือก Driver เป็น Lab Streaming Layer (LSL) และกดปุ่ม Connect เพื่อเชื่อมต่อ LSL เข้าสู่ตัว Acquisition Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.83 หน้าต่างโปรแกรม OpenViBE Acquisition Server

เมื่อทำการเชื่อมต่ออุปกรณ์ผ่าน LSL เข้าสู่ OpenViBE Acquisition Server แล้วผู้จัดทำได้ออกแบบระบบเบื้องต้นโดยใช้โปรแกรม OpenViBE Designer ซึ่งเป็นโปรแกรมที่ใช้ในการออกแบบการทดลองผ่านเครื่องมือต่างๆ เช่น การกรองสัญญาณและการแสดงผลสัญญาณ เป็นต้น ตัวอย่างหน้าต่างของโปรแกรม OpenViBE Designer แสดงดังรูปที่ 3.84



รูปที่ 3.84 โปรแกรม OpenViBE Designer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.12 การออกแบบทิศทางการขยับของแขนกลจำลอง Jaco Arm 2

ในปริญญาณิพนธ์นี้ผู้จัดทำได้ออกแบบการทำงานของแขนกล Jaco Arm 2 ออกเป็น 3

Stage คือ

- Reset Stage : Stage ที่ไม่มีการส่งผลการทำนายการจินตนาการ MI หรือผลการทำนายไม่ตรงกับการจินตนาการการขยับมือซ้ายและมือขวา
- Left Hand Stage : Stage ที่มีการส่งผลการทำนายเป็นการจินตนาการการขยับมือซ้าย
- Right Hand Stage : Stage ที่มีการส่งผลการทำนายเป็นการจินตนาการการขยับมือขวา

รูปที่ 3.85 แสดงท่าการขยับมือของ แต่ละ Stage โดยการกำหนดค่าเริ่มต้นของพิกัดใน DH Parameter ของแต่ละ Stage แสดงดังตารางที่ 3.1 และรูปที่ 3.86 แสดงฟังก์ชันที่ใช้ในการขยับแขนกลจำลองใน CoppeliaSim โดยใช้ภาษา MATLAB ซึ่งการเชื่อมต่อระหว่าง MATLAB และ CoppeliaSim ต้องมีการกำหนด Port เพื่อให้สามารถทำการเชื่อมต่อ API ของทั้งสองโปรแกรมได้ ดังนั้นในปริญญาณิพนธ์นี้กำหนดไว้ที่ Port 19997



รูปที่ 3.85 การขยับมือของ (a) Reset Stage (b) Left Hand Stage และ (c) Right Hand stage

```

function [t, q, q_act] = control_jaco(task)

addpath('functions_coppelia/');
addpath('functions_matlab/');
porta = 19997; % default V-REP port
tf = 2; % final time
Ts = 0.02; % sampling time
t = 0:Ts:tf; % time vector
N = length(t); % number of points of the simulation
n = 7; % joint number
q = zeros(n,N); % q(:,i) collects the joint position for t(i)
q_jaco = zeros(n,N); % q_jaco(:,i) collects the joint position for t(i) in Kinova convention
dq = zeros(n,N); % dq(:,i) collects the joint velocity for t(i)
q(:,1) = [77 -17 0 43 -94 0 74]'/180*pi; % approximated home configuration
q_jaco(:,1) = mask_q_DH2Jaco(q(:,1));
state = 0;
% msgbox interface
CreateStruct.Interpreter = 'tex';
CreateStruct.WindowStyle = 'modal';

clc
fprintf('-----');
fprintf('\n simulation started ');
fprintf('\n trying to connect...\n');
[clientID, vrep ] = StartVrep(porta);

handle_joint = my_get_handle_Joint(vrep,clientID); % handle to the joints
my_set_joint_target_position(vrep, clientID, handle_joint, q_jaco(:,1)); % first move to q0
q_act(:,1) = my_get_joint_target_position(clientID,vrep,handle_joint,n); % get the actual joints angles from v-rep
% Kinova conversion -> DH
q_act(:,1) = mask_q_DH2Jaco(q_act(:,1));

% Conditions
if task == 0
    ang = [0 0 0 0 -1 0]'/180*pi;
elseif task == 1
    ang = [0 0 0 0 1 0]'/180*pi;
else
    waitfor(msgbox('\fontsize{14}\rm Invalid task', 'Error', 'error', CreateStruct));
    state = 1;
end
% Move to target oustion
if state == 0
    for i=2:N
        q(:,i) = q(:,i-1) + ang;
        % DH -> Kinova conversion
        q_jaco(:,i) = mask_q_DH2Jaco(q(:,i));
        my_set_joint_target_position(vrep, clientID, handle_joint, q_jaco(:,i));
        % get the actual joints angles from v-rep
        q_act(:,i) = my_get_joint_target_position(clientID,vrep,handle_joint,n);
        % Kinova conversion -> DH
        q_act(:,i) = mask_q_Jaco2DH(q_act(:,i));
    end
    waitfor(msgbox('\fontsize{14}\rm Press OK ( or ENTER ) to continue', 'Success', CreateStruct));
end
end

```

รูปที่ 3.86 ฟังก์ชันที่ใช้ในการขยับการเคลื่อนไหวของ Jaco Arm 2 ใน CoppeliaSim

ตารางที่ 3.1 การกำหนดค่าเริ่มต้นของพิกัดใน DH Parameter

Home Configuration in DH convention ( $\theta$ )			
Joint	Reset Stage	Left Hand Stage	Right Hand Stage
1	77	77	77
2	-17	-17	-17
3	0	0	0
4	43	43	43
5	-94	-94	-94
6	0	90	-90
7	74	74	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.13 การเตรียมสัญญาณจากอุปกรณ์ Unicorn Hybrid Black

ในปริณญาณิพนธ์นี้เลือกใช้โปรแกรม MATLAB ในการเตรียมสัญญาณ โดยข้อมูลแต่ละคอลัมน์ของไฟล์สกุล .CSV ที่ได้จากการเก็บสัญญาณคลื่นไฟฟ้าสมองคือ

- Time\_250\_Hz: Sampling Frequency 250 Hz
- Epoch: ลำดับของการทดลองย่อย
- Channel1 - Channel17: สัญญาณในแต่ละ Channel
- Event Id: Stimulation Trigger
- Event Date: วันที่ทำการทดลอง
- Event Duration: ระยะเวลาของเหตุการณ์

ในการทดลองผู้จัดทำเก็บสัญญาณคลื่นไฟฟ้าสมองทั้งหมด 8 Channel หรือ 8 Electrode จึงเขียนคำสั่งเพื่อดึงข้อมูลเฉพาะคอลัมน์ที่ 1-10 และ 20-22 ดังรูปที่ 3.87

```
raw_data = raw_data{:, [1:10, 20:22]};
```

รูปที่ 3.87 คำสั่งเลือกคอลัมน์ที่ต้องการ

รูปที่ 3.88 แสดงตัวอย่างของคำสั่งในการหาแต่ละ Trigger โดยใช้ตัวเลขของ Trigger ในหัวข้อ 3.1.8 มาอ้างอิง ตัวอย่างคำสั่งที่ใช้เลือกช่วงสัญญาณที่เกิด MI คือ 2-7 วินาที และเลือกเฉพาะ Channel C3, Cz และ C4 (คอลัมน์ที่ 3, 4 และ 5) แสดงดังรูปที่ 3.89 เนื่องจากข้อมูลที่บันทึกมามีค่า Amplitude ที่สูง จึงต้อง Normalize ข้อมูลก่อน แล้วนำข้อมูลที่ผ่านการเลือกช่วงเวลาผ่านตัวกรองแถบผ่านแบบ Butterworth Order 4 และใช้ช่วงความถี่ที่ 8-30 Hz

```
start_trial = find(raw_data(:,11)==768)
```

```
start_trial = 2x1
    7493
    38998
```

```
left_idx = find(raw_data(:,11)==769)
```

```
left_idx = 12x1
    1
    2
    3
    4
    5
    6
    7
    8
    10243
    12497
    14744
    16997
    19249
    23741
    28246
    32750
```

```
right_idx = find(raw_data(:,11)==770)
```

```
right_idx = 12x1
    7995
    21499
    25998
    30497
    34999
    39504
    41754
    44005
    48501
    50753
    ...
```

```
feedback = find(raw_data(:,11)==781)
```

```
feedback = 24x1
    8308
    10557
    12810
    15056
    17307
    19560
    21811
    24060
    26311
    28559
    ...
```

```
stop_trial = find(raw_data(:,11)==800)
```

```
stop_trial = 24x1
    9495
    11746
    13993
    16247
    18498
    20748
    22999
    25248
    27497
    29751
    ...
```

รูปที่ 3.88 คำสั่งในการหา Index ของแต่ละ Trigger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% Left hand
for i=1:length(left_idx)
    x = i+trial+day;
    eeg.raw.lh(n_sub,x,,:) = [raw(left_idx(i)+eeg.fs:left_idx(i)+5*eeg.fs, [5,6,7])];
    eeg.norm.lh(n_sub,x,,:) = normalize(eeg.raw.lh(n_sub,x,,:));
    eeg.fil.lh(n_sub,x,,:) = bp_filter(4, 8, 30, eeg.fs, eeg.norm.lh(n_sub,x,,:));
end

```

รูปที่ 3.89 คำสั่งที่ใช้ในการเตรียมสัญญาณ

## 3.2 เครื่องมือที่ใช้ในการทดลอง

ในปริณญาณิพนธ์นี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลองดังนี้

### 3.2.1 MATLAB R2021b

ใช้สำหรับการเขียนโปรแกรมโดยใช้ภาษา MATLAB ในการประมวลผลสัญญาณ การแยกคุณลักษณะ การจำแนกข้อมูลและการประเมินความแม่นยำของโครงสร้างการเรียนรู้เชิงลึก

### 3.2.2 OpenVIBE

ใช้สำหรับการจัดเตรียมกระบวนการทาง BCI เช่น System Training, Offline System Testing และ Real-Time System Testing

### 3.2.3 CoppeliaSim

ใช้สำหรับการควบคุมแขนกลจำลอง Jaco Arm 2

### 3.2.4 Unicorn Hybrid Black

ใช้สำหรับการบันทึกคลื่นไฟฟ้าสมอง

### 3.2.5 Visual Code Studio

ใช้สำหรับเขียนโปรแกรมโดยใช้ภาษา Python ในการฝึกสอนโมเดล, การจำแนกข้อมูล และประเมินความแม่นยำของ HDNN Model

### 3.2.6 Nvidia Geforce RTX 2060

GPU ที่ใช้ในการฝึกสอน ResNet18

## 3.3 การจัดเก็บผลการทดลอง

### 3.3.1 การทดสอบการเตรียมข้อมูล

เป็นการทดลองเพื่อเตรียมข้อมูล BCI Competition IV 2a เพื่อให้ได้สัญญาณที่พร้อมใช้ในการสกัดคุณลักษณะโดยมีการเปรียบเทียบระหว่างการใช้ Tool ของ EEGLAB และการเขียนฟังก์ชันด้วยตัวเอง

### 3.3.2 การทดสอบอัลกอริธึม Common Spatial Pattern กับสัญญาณ EEG

เป็นการทดลองเพื่อแสดงผลลัพธ์ของเอาต์พุตหลังผ่านการสกัดคุณลักษณะด้วยวิธี CSP ของชุดข้อมูล BCI Competition IV 2a และพิจารณาคุณสมบัติของการสกัดคุณลักษณะด้วยวิธีดังกล่าว

### 3.3.3 การทดสอบการแปลงเวฟเลตแบบดิสครีตกับสัญญาณ EEG

เป็นการทดลองเพื่อแสดงผลลัพธ์ของเอาต์พุตหลังผ่านการสกัดคุณลักษณะด้วยวิธี DWT ของชุดข้อมูล BCI Competition IV 2a และพิจารณาคุณสมบัติของการสกัดคุณลักษณะด้วยวิธีดังกล่าว

### 3.3.4 การทดสอบการแปลงเวฟเลตแบบต่อเนื่องกับสัญญาณ EEG

เป็นการทดลองเพื่อแสดงผลลัพธ์ของเอาต์พุตหลังผ่านการสกัดคุณลักษณะด้วยวิธี CWT ของชุดข้อมูล BCI Competition IV 2a และพิจารณาคุณสมบัติของการสกัดคุณลักษณะด้วยวิธีดังกล่าว

### 3.3.5 การทดสอบการรวมคุณลักษณะ

เป็นการทดลองเพื่อแสดงผลลัพธ์ของเอาต์พุตหลังผ่านการสกัดคุณลักษณะด้วยวิธี CWT กับ CSP และ DWT กับ CSP ของชุดข้อมูล BCI Competition IV 2a และพิจารณาคุณสมบัติของการสกัดคุณลักษณะด้วยวิธีดังกล่าว

### 3.3.6 ระยะเวลาที่ใช้ในประมวลผลการแยกคุณลักษณะและการจำแนกด้วยอัลกอริธึมการเรียนรู้เชิงลึก

เป็นการทดสอบระยะเวลาที่ใช้ประมวลผลการแยกคุณลักษณะ CWT+CSP และการจำแนกด้วยอัลกอริธึมการเรียนรู้เชิงลึกโครงสร้าง ResNet18, ResNet50, ShuffleNet, MLP, LeNet-5 และ HDNN

### 3.3.7 การทดสอบการจำแนกคลาสด้วยอัลกอริธึมการเรียนรู้เชิงลึกด้วยชุดข้อมูลชุดทดสอบ BCI Competition IV 2a

เป็นการทดสอบการจำแนกคลาสชุดข้อมูลทดสอบ BCI Competition IV 2a ด้วยโครงสร้างอัลกอริธึมการเรียนรู้เชิงลึกทั้ง 6 โครงสร้างที่เลือกมาศึกษา โดยนำวิธีสกัดคุณลักษณะและอัลกอริธึมที่ให้ค่าความแม่นยำของระบบมากที่สุดไปใช้กับชุดข้อมูลคลื่นไฟฟ้าจากสมองของผู้จัดทำ

### 3.3.8 การทดสอบการทำงานของ System Training ด้วยชุดข้อมูลคลื่นไฟฟ้าจากสมองของผู้จัดทำ

เป็นการทดสอบการสกัดคุณลักษณะและการจำแนกคลาสจากชุดข้อมูลคลื่นไฟฟ้าจากสมองของผู้จัดทำโดยใช้ระบบ System Training โดยนำวิธีสกัดคุณลักษณะ, อัลกอริธึมการเรียนรู้เชิงลึก และ Subject ที่ให้ค่าความแม่นยำของระบบมากที่สุดไปใช้กับ Real-Time System Testing

### 3.3.9 การทดสอบการทำงานของ Offline System Testing ด้วยชุดข้อมูลชุดทดสอบ BCI Competition IV 2a

เป็นการทดสอบระบบโดยนำโมเดลที่ถูกฝึกสอนด้วยชุดข้อมูล BCI Competition IV 2a ของ Subject ที่ 9 มาทดสอบการควบคุมแขนกลจำลอง

### 3.3.10 การทดสอบการทำงานของ Real-Time System Testing

เป็นการทดสอบการทำงานของระบบโดยรวมแบบ Real-Time System Testing ของ Subject ที่ให้ค่าความแม่นยำมากที่สุดจากข้อสรุปในหัวข้อที่ 3.3.8

## บทที่ 4

### ผลการทดลอง

ผู้จัดทำได้ทำการเก็บผลการทำงานของระบบ โดยแบ่งการทดลองและจัดเก็บผลการทดลองเป็นส่วนๆ ดังต่อไปนี้

#### 4.1 การทดสอบการเตรียมข้อมูล

เนื่องจากการเก็บสัญญาณ EEG นั้นจะมีส่วนประกอบของความถี่อื่นที่ไม่เกี่ยวข้องกับช่วงความถี่ที่ต้องการและเมื่อพิจารณาข้อมูลพบว่าในบางช่วงเกิดการขาดหายของสัญญาณ ดังนั้นก่อนที่จะทำการสกัดคุณลักษณะของข้อมูลจำเป็นต้องกรองความถี่เพื่อให้ได้ช่วงของสัญญาณที่ต้องการและทำการประมาณค่าของข้อมูลในส่วนที่ขาดหายไป

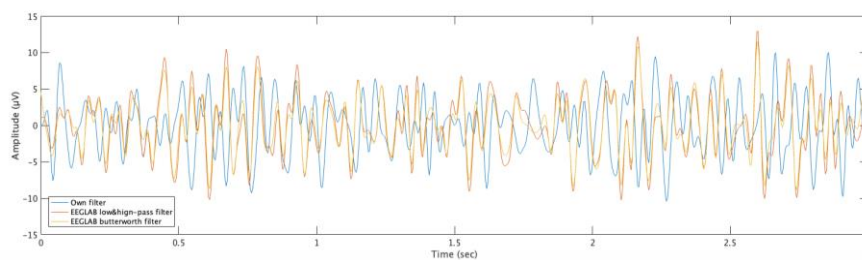
##### 4.1.1 ผลลัพธ์สัญญาณของข้อมูลหลังผ่านตัวกรอง

รูปที่ 4.1 เป็นตัวอย่างของชุดข้อมูลในโดเมนเวลาของการกรองสัญญาณในช่วงความถี่ 8-30 Hz โดยใช้ 3 วิธี ได้แก่

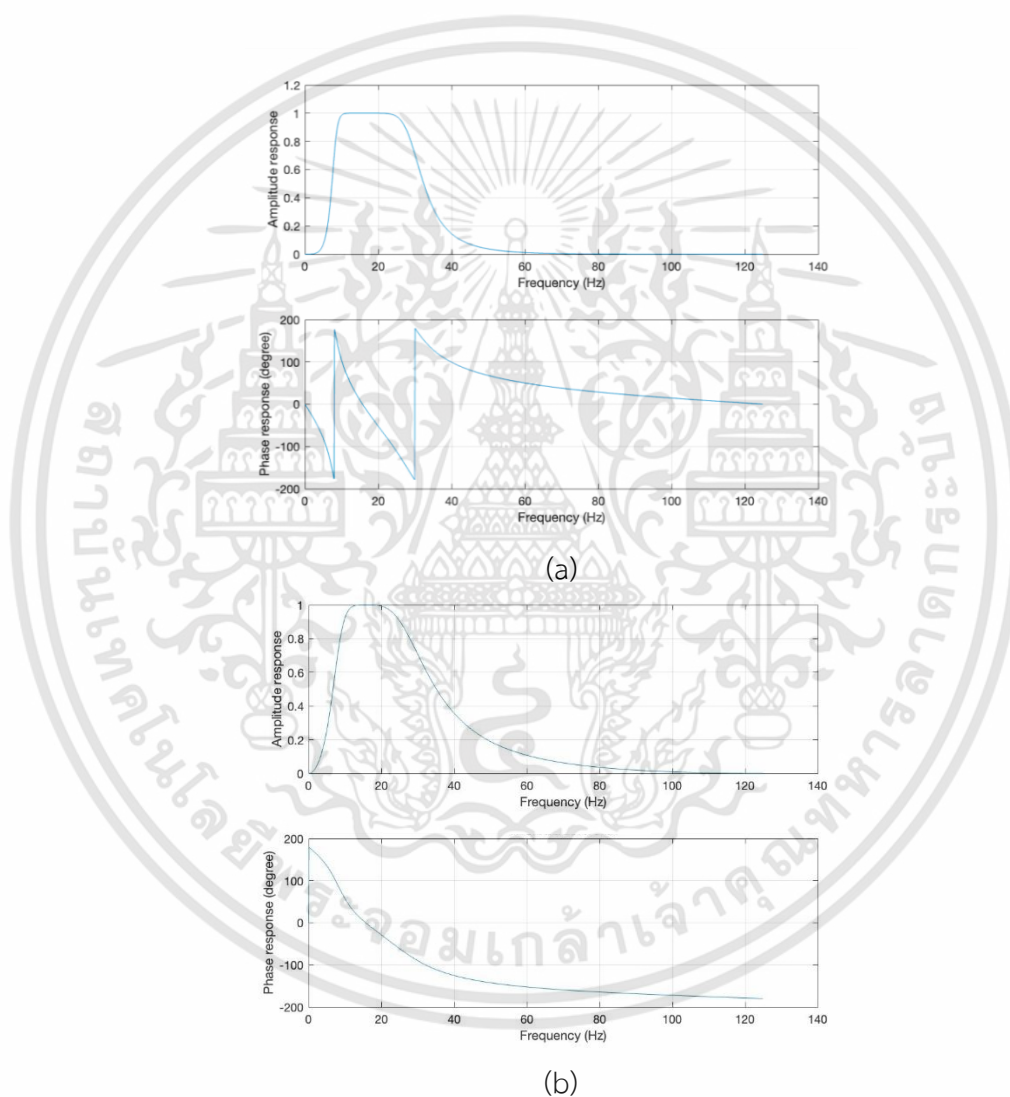
- วิธีการกรองสัญญาณด้วยฟังก์ชัน bp\_filter (Own filter)
- วิธีการกรองสัญญาณด้วยตัวกรองความถี่ต่ำผ่านแล้วกรองด้วยตัวกรองความถี่สูงผ่านใน EEGLAB plug-in (EEGLAB low&high-pass filter)
- วิธีการกรองสัญญาณด้วยฟังก์ชันของ TESA (pop\_tesa\_filtbutter) (EEGLAB butterworth filter)

รูปที่ 4.1 แสดงสัญญาณของ MI คลาส 1 ในโดเมนเวลาหลังผ่านการกรองสัญญาณแถบผ่านทั้ง 3 แบบจะเห็นว่าการเหลื่อมของสัญญาณที่เกิดขึ้นในรูปที่ 4.1 เกิดจากความแตกต่างของ Phase Response ระหว่างฟังก์ชัน bp\_filter และ TESA Butterworth Filter ที่นำมาใช้งาน ดังรูปที่ 4.2 โดยเฟสที่เกิดขึ้นของ bp\_filter เป็น Linear Phase ในช่วงความถี่ 8-30 Hz แต่เฟสของ TESA Butterworth Filter ไม่ได้เป็น Linear Phase ซึ่งในปริยญาณิพนธ์นี้จะเลือกใช้ฟังก์ชัน bp\_filter ในการกรองสัญญาณ

รูปที่ 4.3 แสดง Power Spectral Density (PSD) ของแต่ละชุดข้อมูล เมื่อพิจารณารูปที่ 4.1 และ 4.3 ชุดข้อมูลที่ถูกกรองผ่านโปรแกรม EEGLAB และฟังก์ชันของ TESA ให้สัญญาณในโดเมนเวลาและ PSD ที่คล้ายกัน และจากรูปที่ 4.3 ในช่วงบางความถี่ ชุดข้อมูลที่กรองด้วยฟังก์ชัน bp\_filter มีค่า PSD น้อยกว่าวิธีการกรองสัญญาณด้วย EEGLAB

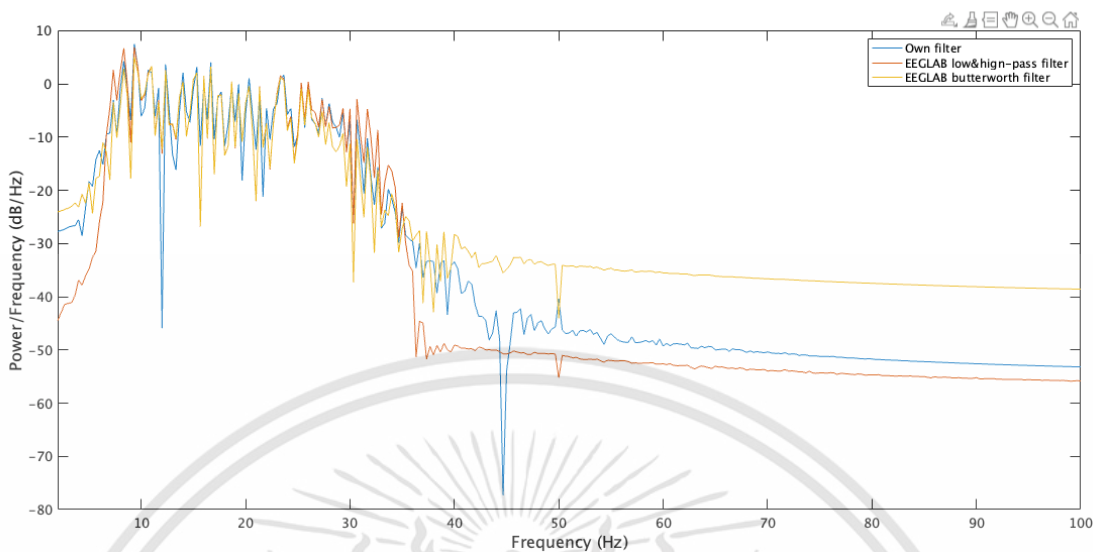


รูปที่ 4.1 สัญญาณของ MI คลาส 1 ในโดเมนเวลาของวิธีการกรองสัญญาณ 3 วิธี



รูปที่ 4.2 Frequency Response ของตัวกรองความถี่แถบผ่านแบบ Butterworth ของ (a) ฟังก์ชัน bp\_filter และ (b) ฟังก์ชัน TESA ใน EEGLAB (TESA Butterworth Filter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 PSD ของสัญญาณหลังผ่านการกรองสัญญาณทั้ง 3 วิธี

#### 4.1.2 ผลลัพธ์ของข้อมูลหลังการประมาณค่าในช่วง

รูปที่ 4.4 แสดงตัวอย่างของข้อมูลที่ถูกประมาณค่าในช่วงด้วยวิธี Cubic Spline Interpolation ซึ่งเป็นตัวอย่างของช่องสัญญาณที่ 22 โดยมีข้อมูล NaN ทั้งหมด 1230 จุด โดย pos\_NaN\_value หมายถึง ตำแหน่งที่เป็นข้อมูล NaN

```
pos_NaN_value = find(isnan(eeg.data(22,:)))
pos_NaN_value = 1x1230
    54    917    924    996   1001   1005   1043   1088   1099

size(pos_NaN_value)
ans = 1x2
    1   1230

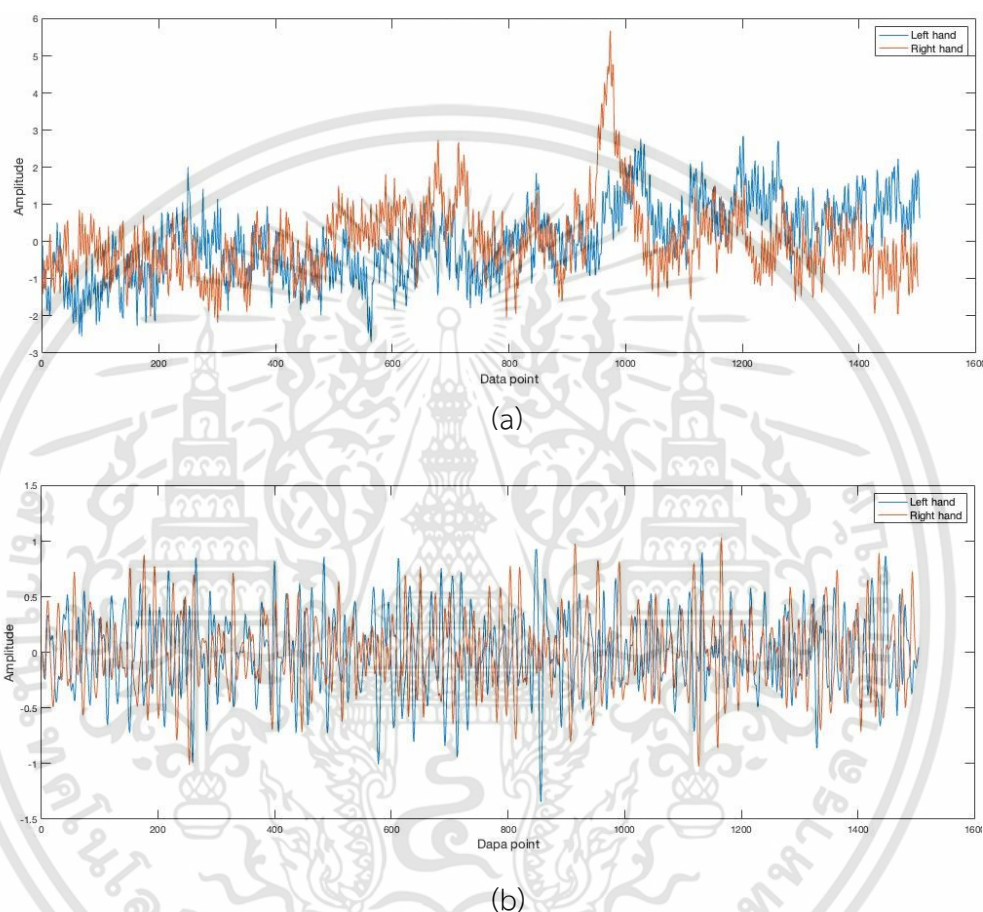
old_data = eeg.data(22,916:925)
old_data = 1x10
    53.1738    NaN    28.7598    5.0293    8.6914    0.2930   -21.5332    2.6855    NaN   -22.6562

new_data = eeg.data_new(22,916:925)
new_data = 1x10
    53.1738    50.0711    28.7598    5.0293    8.6914    0.2930   -21.5332    2.6855   -18.0630   -22.6562
```

รูปที่ 4.4 ตัวอย่างของค่าข้อมูลที่ถูกประมาณค่าในช่วงด้วยวิธี Cubic Spline Interpolation

#### 4.1.3 ผลลัพธ์ของสัญญาณคลื่นไฟฟ้าจากสมองของผู้จัดทำหลังผ่านตัวกรอง

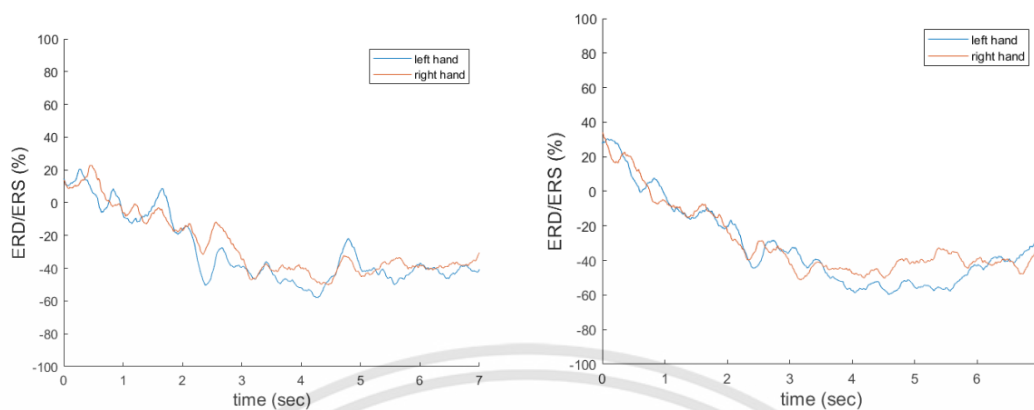
ตัวอย่างผลลัพธ์ของสัญญาณคลื่นไฟฟ้าจากสมองของผู้จัดทำจาก MI การขยับมือซ้ายและมือขวาหลังผ่านการ Normalize, การเลือกช่วงเวลาผ่านตัวกรองแถบผ่านแบบ Butterworth Order 4 และใช้ช่วงความถี่ที่ 8-30 Hz ของ Channel C3 แสดงดังรูปที่ 4.5



รูปที่ 4.5 ผลลัพธ์สัญญาณ (a) ก่อนผ่าน และ (b) หลังผ่านตัวกรองความถี่แถบผ่านแบบ Butterworth

#### 4.1.4 ERD/ERS ของสัญญาณคลื่นไฟฟ้าจากสมองของผู้จัดทำ

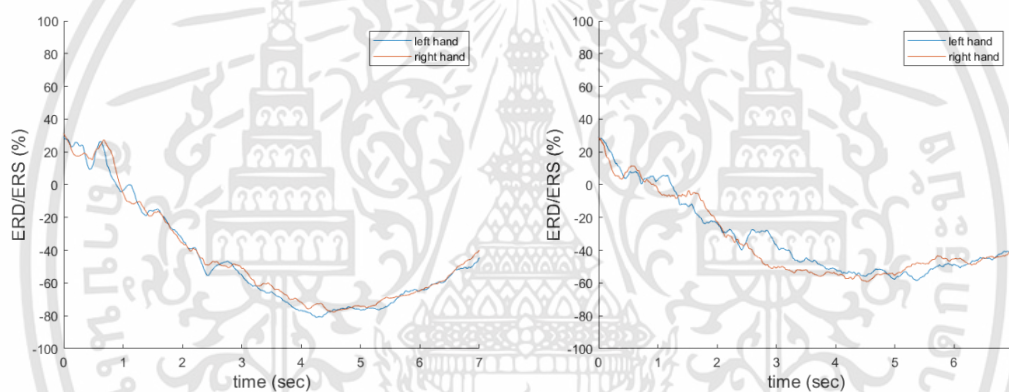
เมื่อนำคลื่นไฟฟ้าจากสมองของผู้จัดทำมาคำนวณหา ERD/ERS ด้วยสมการที่ (2.3) กราฟ ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 1, 2, 3 และ 4 แสดงดังรูปที่ 4.6, 4.7, 4.8 และ 4.9 ตามลำดับ และกราฟ ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 1, 2, 3, 4, 5 และ 6 แสดงดังรูปที่ 4.10, 4.11, 4.12, 4.13, 4.14 และ 4.15 ตามลำดับ ซึ่ง Subject 1 ไม่ได้ทำการเก็บข้อมูลใน Session 5 และ 6 โดยรายละเอียดของกระบวนการที่ค้นพบในแต่ละวินาทีแสดงในรูปที่ 3.53 จากรูปที่ 4.6 – 4.15 นั้น จะพบว่า ERD/ERS ของ MI มือซ้ายและ MI มือนขวานั้นมีความต่างกันในช่วง Session ตั้งแต่วินาทีที่ 2 เป็นต้นไป ซึ่งเป็นวินาทีที่ผู้จัดทำทำการทดลองเริ่มจินตนาการ



(a)

(b)

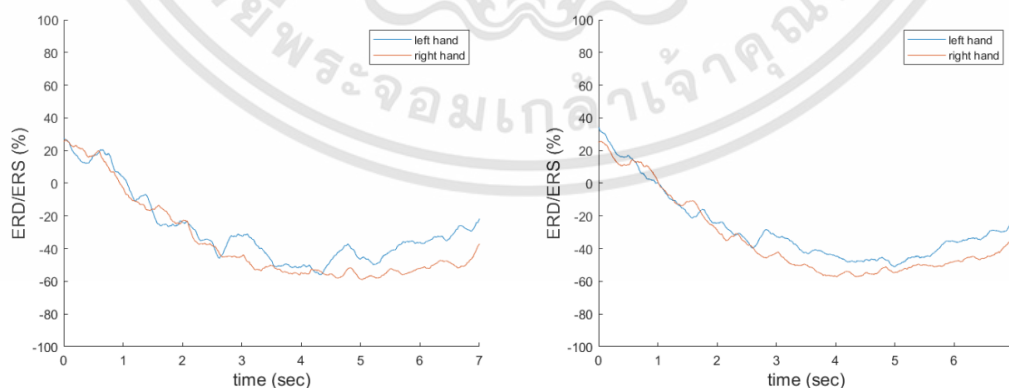
รูปที่ 4.6 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 1 บน Channel (a) C3 (b) C4



(a)

(b)

รูปที่ 4.7 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 2 บน Channel (a) C3 (b) C4

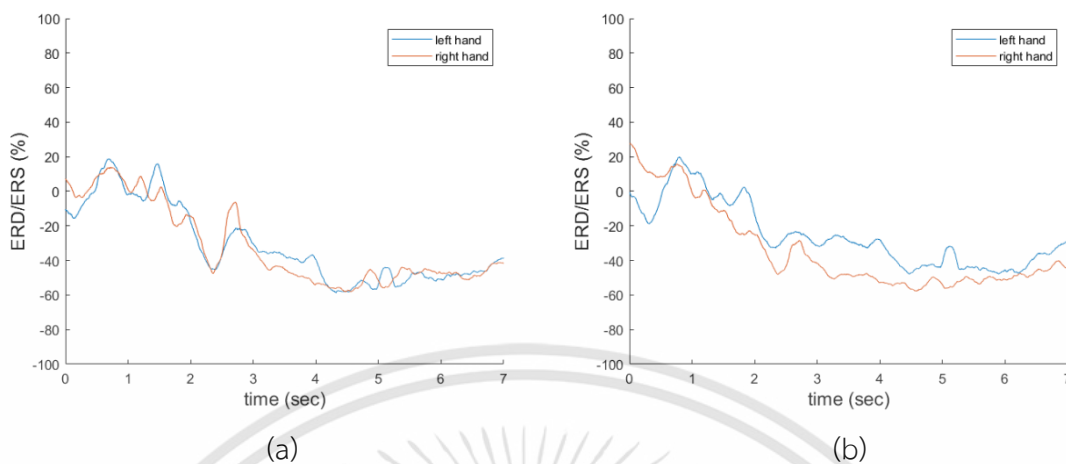


(a)

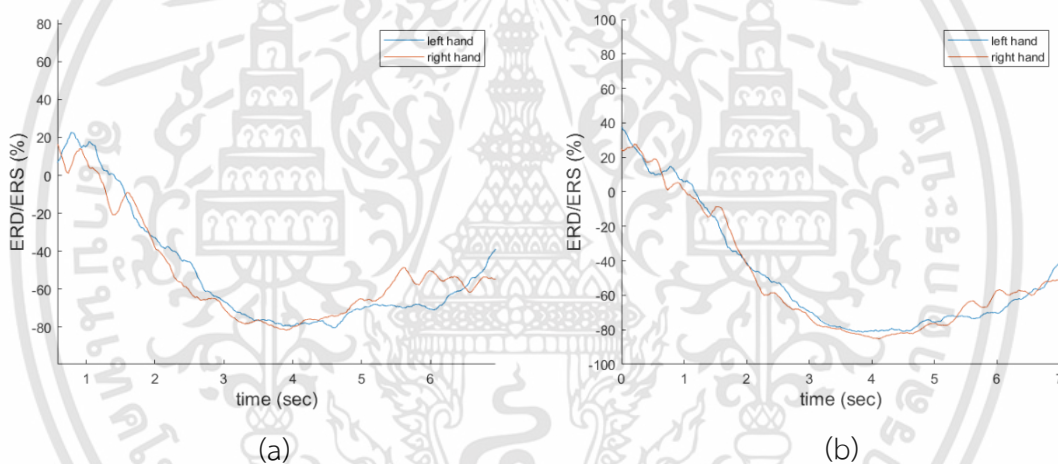
(b)

รูปที่ 4.8 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 3 บน Channel (a) C3 (b) C4

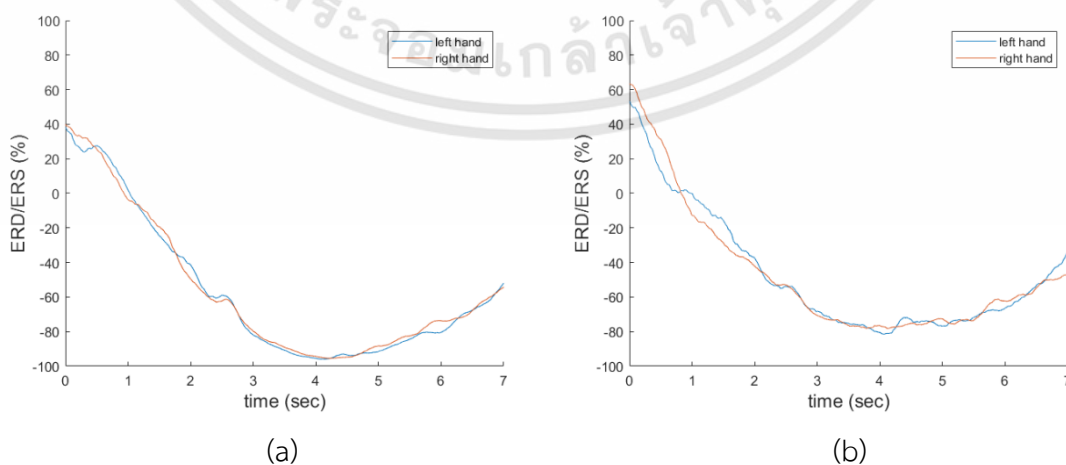
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ERD/ERS ของ Subject ที่ 1 ใน Session ที่ 4 บน Channel (a) C3 (b) C4

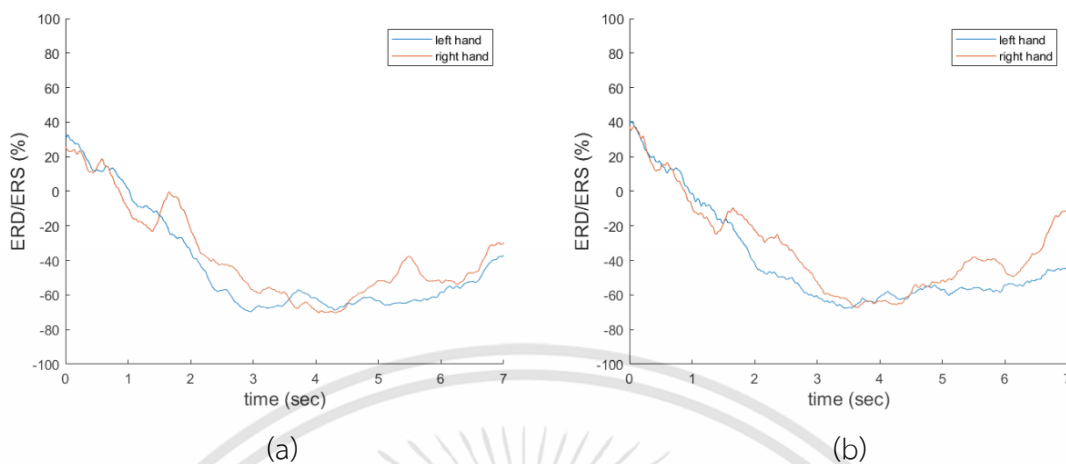


รูปที่ 4.10 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 1 บน Channel (a) C3 (b) C4

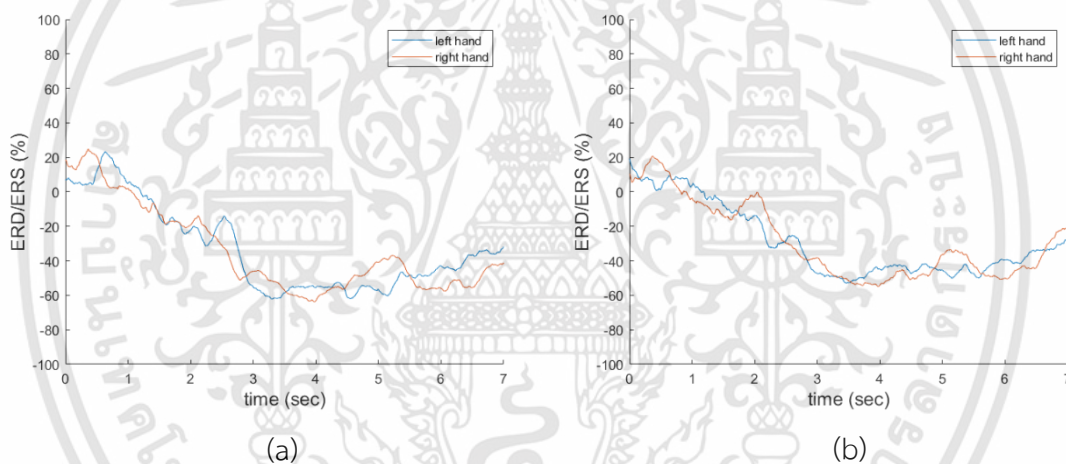


รูปที่ 4.11 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 2 บน Channel (a) C3 (b) C4

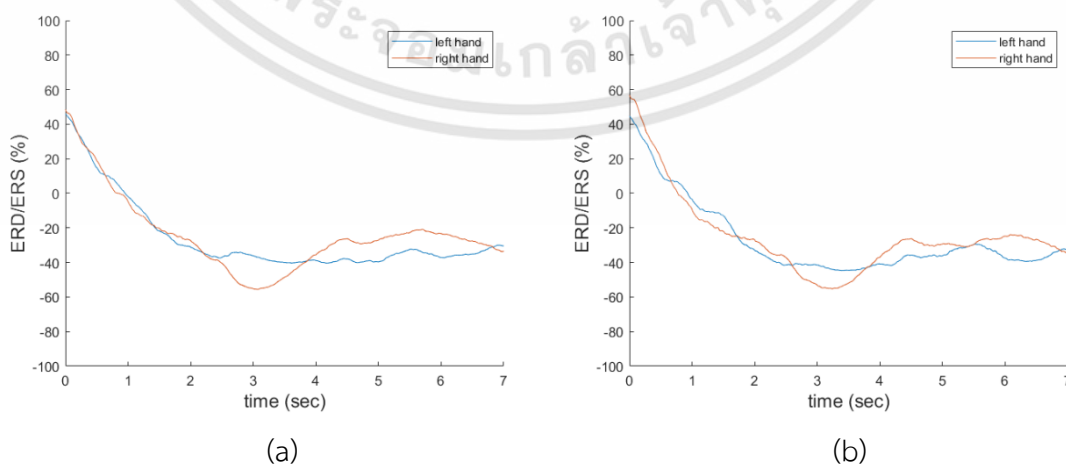
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 3 บน Channel (a) C3 (b) C4

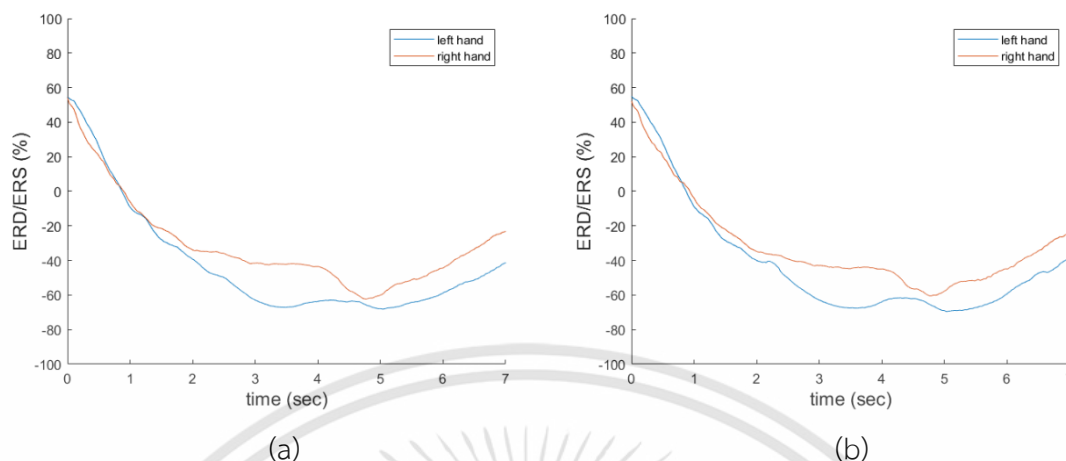


รูปที่ 4.13 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 4 บน Channel (a) C3 (b) C4



รูปที่ 4.14 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 5 บน Channel (a) C3 (b) C4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 ERD/ERS ของ Subject ที่ 2 ใน Session ที่ 6 บน Channel (a) C3 (b) C4

## 4.2 การทดสอบอัลกอริธึม Common Spatial Pattern กับสัญญาณ EEG

### 4.2.1 ผลการทดสอบอัลกอริธึม CSP กับชุดข้อมูล BCI Competition IV 2a

ในการทดสอบอัลกอริธึม CSP ต้องแบ่งจำนวนการทดลองย่อยของข้อมูลสำหรับการเรียนรู้และสำหรับทดสอบ โดยฟังก์ชันที่ใช้ในการแบ่งชุดข้อมูลแสดงดังรูปที่ 4.16

```
function [train, test] = split_one_class(x, percent)
data_size = size(x);
i = data_size(1);           % i = number of trials
i_tr = round(i*percent);
i_te = i - i_tr;

train = x(1:i_tr, :, :);
test = x(i_tr+1:i_tr+i_te, :, :);
end
```

รูปที่ 4.16 ฟังก์ชันในการแบ่งข้อมูลสำหรับเรียนรู้และทดสอบอัลกอริธึม CSP

การเขียนโปรแกรมที่ใช้อัลกอริธึม CSP แสดงดังรูปที่ 4.17 ซึ่งแบ่งออกเป็น

- สร้าง Projection Matrix หรือเมทริกซ์  $W$  โดยใช้ชุดข้อมูลสำหรับเรียนรู้ของแต่ละคลาสเป็นอินพุต
- นำชุดข้อมูลของสองคลาสมาท่อกันเพื่อใช้เป็นอินพุตในการคูณกับเมทริกซ์  $W$
- เลือก  $N_w = 2$  ซึ่งหมายถึงเลือกช่องสัญญาณสองช่องแรกและสองช่องสุดท้ายในมิติที่สอง
- หาเอาต์พุตของแต่ละ Trial โดยฟังก์ชันอัลกอริธึม CSP เมื่อข้อมูลแต่ละคลาสมี่จำนวนหลายการทดลองย่อย โดยใช้ฟังก์ชัน CSP ในรูปที่ 4.18

ผลลัพธ์ของการกระจายของชุดข้อมูลก่อนเข้าอัลกอริธึม CSP แสดงดังรูปที่ 4.19 โดยแกน  $x$  หมายถึงอิเล็กโทรด C3 และแกน  $y$  หมายถึงอิเล็กโทรด C4 ซึ่งเป็นช่องสัญญาณที่มองเห็นความแตกต่างระหว่างสองคลาสได้มากที่สุด รายละเอียดของชุดข้อมูลก่อนผ่านอัลกอริธึมประกอบด้วยจำนวนของช่องสัญญาณ 22 ช่องสัญญาณและจำนวน Sample 750 ตำแหน่ง ผลลัพธ์การกระจายของข้อมูลของสองคลาสหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component แรกของเมทริกซ์  $W$  ของชุดข้อมูลเรียนรู้และทดสอบ กรณีสัญญาณผ่านตัวกรองแถบผ่านที่ช่วงความถี่ 8-13 Hz แสดงดังรูปที่ 4.20 (a) และ (b) ตามลำดับ และผลลัพธ์การกระจายของข้อมูลของสองคลาสหลังผ่านอัลกอริธึม CSP ใน 2 Component สุดท้ายของเมทริกซ์  $W$  ของชุดข้อมูลเรียนรู้และทดสอบ กรณีสัญญาณผ่านตัวกรองแถบผ่านที่ช่วงความถี่ 8-13 Hz แสดงดังรูปที่ 4.21 (a) และ (b) ตามลำดับ โดยรูปด้านซ้ายหรือสัญลักษณ์สีฟ้า หมายถึงการจินตนาการการขยับมือซ้ายและโดยรูปด้านขวาหรือสัญลักษณ์สีแดง หมายถึงการจินตนาการการขยับมือขวา

```
[W, lambda, A] = csp(eeg.train.lh,eeg.train.rh,3); (a)
X = vertcat(eeg.train.lh,eeg.train.rh); (b)
Wb = W(:, [1,2,21,22]); (c)
data_size = size(X);
for i=1:data_size(1)
    x = X(i,:,:);
    x = squeeze(x);
    Z = Wb'*x;
    eeg.csp(i,:,:)= Z;
end (d)
```

รูปที่ 4.17 คำสั่งที่ใช้ในการวิเคราะห์ผลลัพธ์ของอัลกอริธึม CSP

```

function [W, lambda, A] = csp(x1, x2, dim)

% Compute the covariance matrix
if dim==2
    samp_cov_s1 = cov(x1') % s1~[c x c]
    samp_cov_s2 = cov(x2') % s2~[c x c]
else
    data_size = size(x1);
    num_trial = data_size(1); % i = number of trials
    sum_s1 = 0; sum_s2 = 0;
    for i=1:num_trial
        samp1 = x1(i,:,:); samp1 = squeeze(samp1);
        samp2 = x2(i,:,:); samp2 = squeeze(samp2);
        s1 = cov(samp1'); % s1~[c x c]
        s2 = cov(samp2'); % s2~[c x c]
        sum_s1 = sum_s1 + s1;
        sum_s2 = sum_s2 + s2;
    end

    % Average covariance of each class
    samp_cov_s1 = sum_s1/(num_trial-1);
    samp_cov_s2 = sum_s2/(num_trial-1);
end

dim_s1 = size(samp_cov_s1)
dim_s2 = size(samp_cov_s2)

% Solve the eigenvalue problem s1(W) = D(s2)(W)
[W,L] = eig(samp_cov_s1, samp_cov_s1+samp_cov_s2); % Mixing matrix W (Spatial filters are columns)
[lambda,ind] = sort(diag(L)); % Sort eigenvalue in descending order
W = W(:,ind);
A = (inv(W))';
end

```

รูปที่ 4.18 ฟังก์ชัน CSP

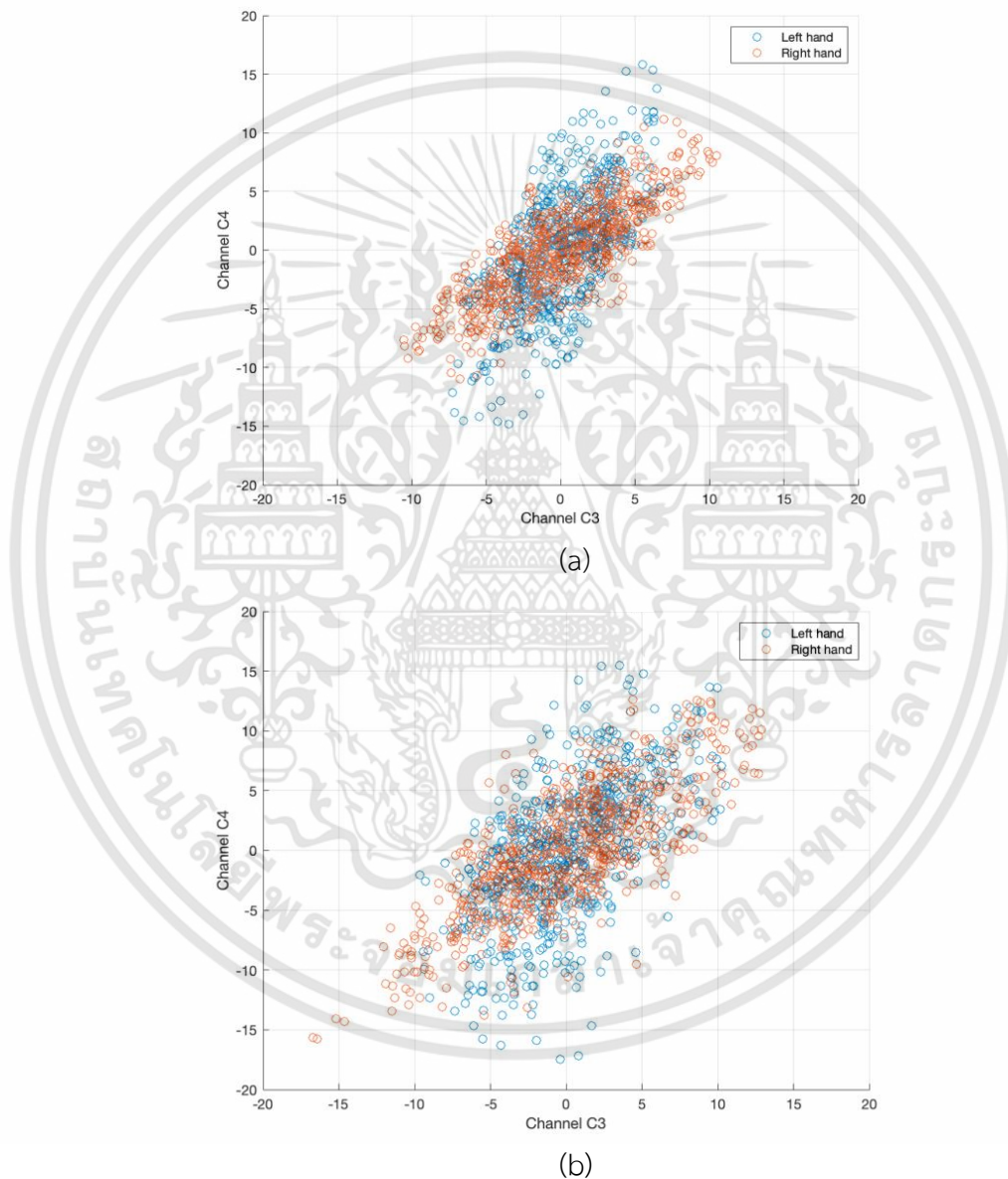
ผลลัพธ์การกระจายของข้อมูลทั้งสองคลาสหลังจากอัลกอริธึม CSP โดยใช้ 2 Component แรกของ W ของชุดข้อมูลเรียนรู้และทดสอบ กรณีสัญญาณผ่านตัวกรองแถบผ่านที่ช่วงความถี่ 8-13 Hz ซึ่งเป็นความถี่ในช่วงของ Mu-band แสดงดังรูปที่ 4.20 และผลลัพธ์การกระจายของข้อมูลของสองคลาสหลังจากอัลกอริธึม CSP โดยใช้ 2 Component สุดท้ายของเมทริกซ์ W ของชุดข้อมูลเรียนรู้และทดสอบ กรณีสัญญาณผ่านตัวกรองแถบผ่านที่ช่วงความถี่เดียวกัน แสดงดังรูปที่ 4.21 โดยรูปด้านซ้ายหรือสัญลักษณ์สีฟ้าหมายถึงการจินตนาการการขยับมือซ้ายและรูปด้านขวาหรือสัญลักษณ์สีแดงหมายถึงการจินตนาการการขยับมือขวา ซึ่งการกระจายของข้อมูลในชุดข้อมูลเรียนรู้สามารถแยกคลาสได้ดีกว่าชุดข้อมูลทดสอบทั้งแบบใช้ 2 Component แรก และ 2 Component สุดท้าย

ความแตกต่างระหว่างการใช้ 2 Component แรก และ 2 Component สุดท้ายคือ 2 Component แรกเห็นการกระจายของข้อมูลในแต่ละคลาสคล้ายคลึงกับทฤษฎีในหัวข้อ 2.2.2 รูปที่ 2.7 มากกว่า 2 Component สุดท้าย โดยคลาสการจินตนาการการขยับมือซ้ายมีการกระจายของข้อมูลในแนวนอน ส่วนคลาสการจินตนาการการขยับมือขวามีการกระจายในแนวตั้ง

ผลลัพธ์การกระจายของข้อมูลทั้งสองคลาสหลังจากอัลกอริธึม CSP โดยใช้ 2 Component แรกของ W ของชุดข้อมูลเรียนรู้และทดสอบ กรณีสัญญาณผ่านตัวกรองแถบผ่านที่ช่วงความถี่ Mu-band และ Beta-band แสดงดังรูปที่ 4.22 และผลลัพธ์การกระจายของข้อมูลของสองคลาสหลังจากอัลกอริธึม CSP โดยใช้ 2 Component สุดท้ายของเมทริกซ์ W ของชุดข้อมูลเรียนรู้และทดสอบ กรณีสัญญาณผ่านตัวกรองแถบผ่านที่ช่วงความถี่เดียวกัน แสดงดังรูปที่ 4.23 โดยรูปด้านซ้ายเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

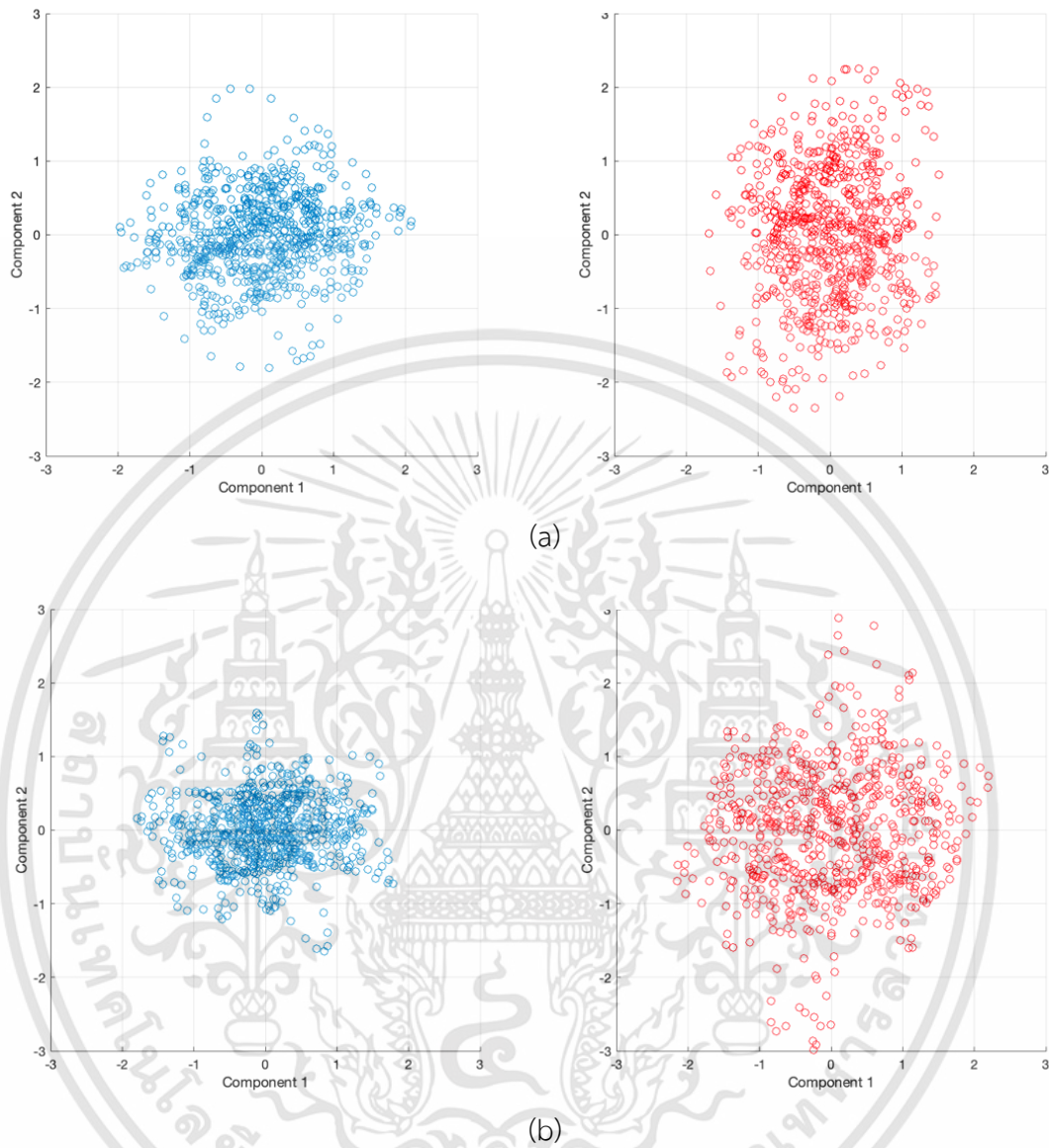
หรือสัญลักษณ์สีฟ้าหมายถึงการจินตนาการการขยับมือซ้ายและรูปด้านขวาหรือสัญลักษณ์สีแดงหมายถึงการจินตนาการการขยับมือขวา

จากผลลัพธ์ในการกรองสัญญาณที่ใช้ตัวกรองแถบผ่านช่วงความถี่ต่างกัน พบว่าสัญญาณที่ถูกกรองด้วยช่วงความถี่ Mu-band (8-13 Hz) จะเห็นการกระจายของข้อมูลแต่ละคลาสหลังผ่านอัลกอริธึม CSP ได้ชัดกว่าใช้ตัวกรองแถบผ่านที่มีความถี่ช่วง Mu-band และ Beta-band (8-30 Hz)

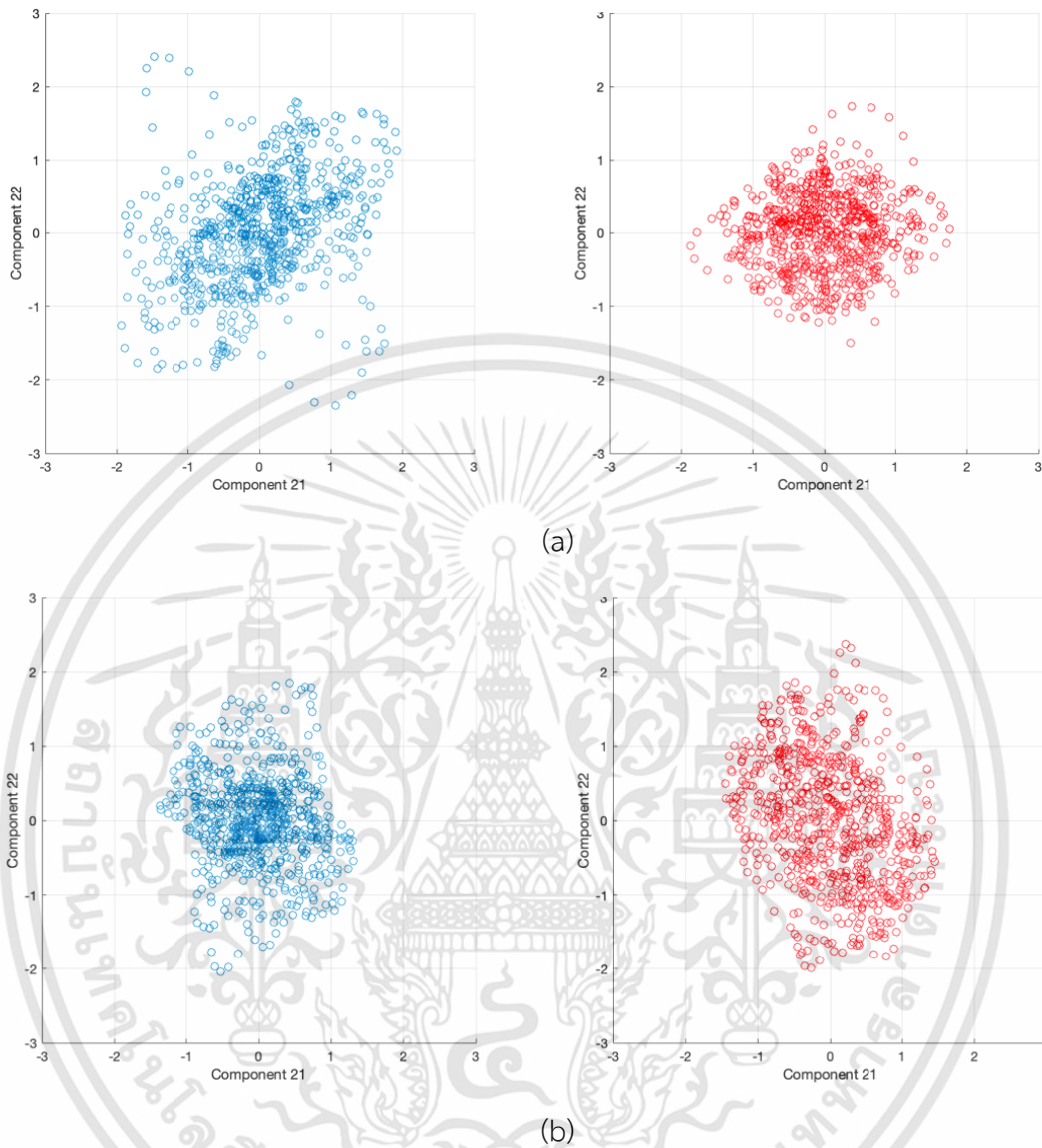


รูปที่ 4.19 ผลลัพธ์ของการกระจายของชุดข้อมูลก่อนเข้าอัลกอริธึม CSP (a) สัญญาณที่ถูกกรองด้วยตัวกรองแถบผ่านที่มีความถี่ 8-13 Hz (b) สัญญาณที่ถูกกรองด้วยตัวกรองแถบผ่านที่มีความถี่ 8-30 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



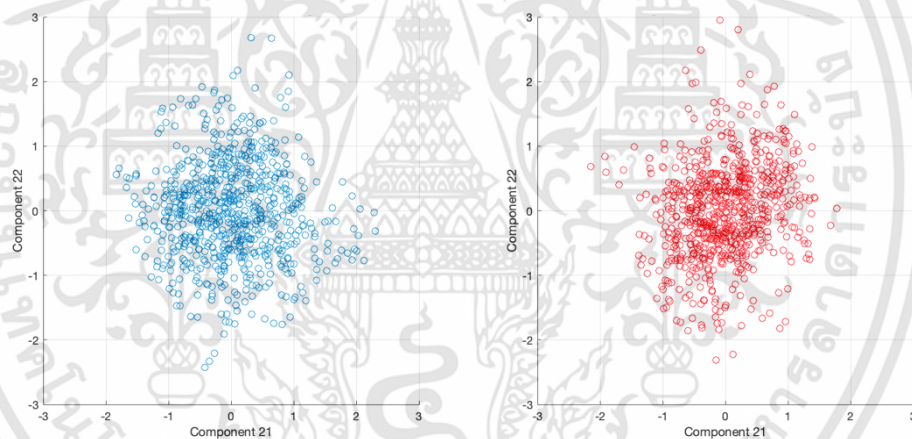
รูปที่ 4.20 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component แรกของเมทริกซ์  $W$  (a) อินพุตของอัลกอริธึม CSP คือชุดข้อมูลเรียนรู้ และ (b) อินพุตของอัลกอริธึม CSP คือชุดข้อมูลทดสอบ



รูปที่ 4.21 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component สุดท้ายของเมทริกซ์  $W$  (a) อินพุตของอัลกอริธึม CSP คือชุดข้อมูลเรียนรู้ และ (b) อินพุตของอัลกอริธึม CSP คือชุดข้อมูลทดสอบ



รูปที่ 4.22 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component แรกของเมทริกซ์  $W$  เมื่ออินพุตของอัลกอริธึม CSP คือชุดข้อมูลเรียนรู้

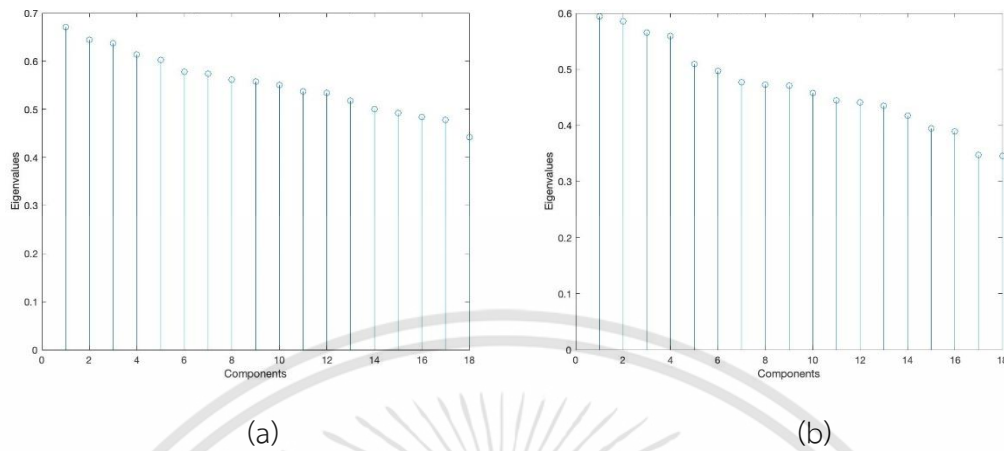


รูปที่ 4.23 ผลลัพธ์ของการกระจายของชุดข้อมูลหลังผ่านอัลกอริธึม CSP โดยใช้ 2 Component สุดท้ายของเมทริกซ์  $W$  เมื่ออินพุตของอัลกอริธึม CSP คือชุดข้อมูลเรียนรู้

#### 4.2.2 ผลการทดสอบอัลกอริธึม CSP กับชุดข้อมูลของผู้จัดทำ

การพิจารณา Eigenvalue ของเมทริกซ์  $W$  ซึ่งคำนวณจากข้อมูลใน Channel C3 และ C4 ในรูปที่ 4.24 พบว่ามีค่าใกล้เคียงกันหมด ซึ่งหมายความว่าไม่มี Component ใดเป็น Component หลักในการอธิบายข้อมูล ดังนั้นในการทดลองการสกัดคุณลักษณะด้วย CSP จึงเลือกทุก Component ของ  $W$  มาใช้ในการคำนวณ

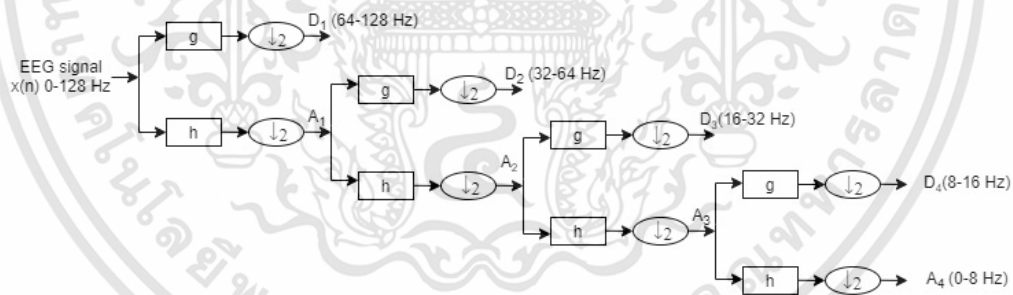
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 Eigenvalue ของเมทริกซ์  $W$  จาก Channel (a) C3 (b) C4

### 4.3 การทดสอบการแปลงเวฟเลตแบบดิสคริตของสัญญาณ EEG

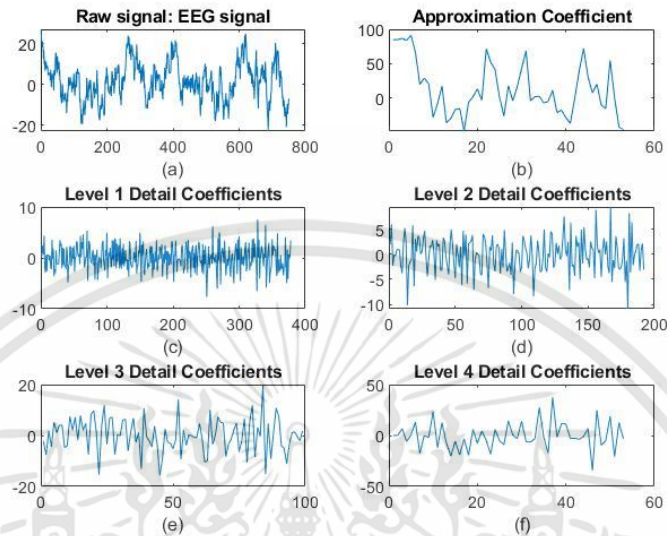
เมื่อทำการเตรียมสัญญาณ EEG ของชุดข้อมูล BCI Competition IV 2a เรียบร้อยแล้ว นำสัญญาณของ 1 อิเล็กโทรด 1 การทดลองย่อยในแต่ละคลาสมาแปลงเวฟเลตแบบดิสคริตเพื่อให้ได้ค่าสัมประสิทธิ์ในแต่ละชั้น โดยแต่ละชั้นของสัญญาณที่ถูกแยกออกมาจะเกี่ยวข้องกับความถี่ในช่วงต่างๆ [60] แสดงดังรูปที่ 4.25



รูปที่ 4.25 DWT ของสัญญาณ EEG และความถี่ที่เกี่ยวข้องแต่ละชั้น

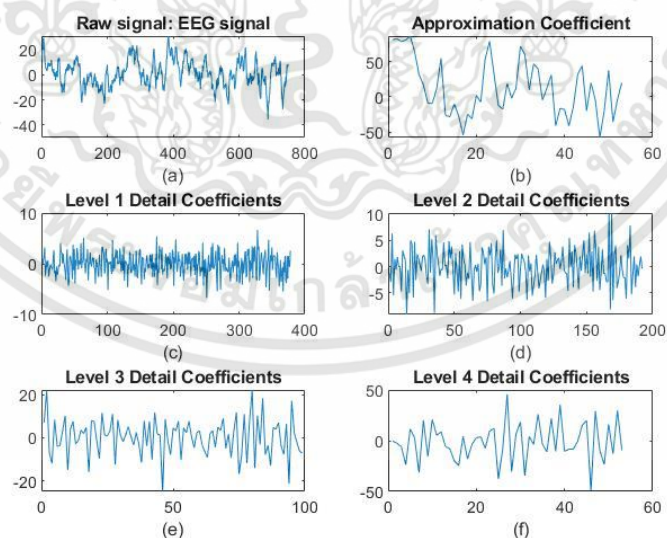
จากนั้นทำการทดลองโดยนำสัญญาณ EEG ในแต่ละคลาสมาแปลงเวฟเลตแบบดิสคริตโดยใช้ Db4 จำนวน 4 ชั้น โดยค่าสัมประสิทธิ์ของสัญญาณจากคลาสที่ 1 (มือซ้าย) บนอิเล็กโทรด C3 และ C4 แสดงในรูปที่ 4.26 และรูปที่ 4.27 ตามลำดับ คลาสที่ 2 (มือขวา) บนอิเล็กโทรด C3 และ C4 แสดงในรูปที่ 4.28 และรูปที่ 4.29 ตามลำดับ คลาสที่ 3 (ลิ้นขวา) บนอิเล็กโทรด C3 และ C4 แสดงในรูปที่ 4.30 และรูปที่ 4.31 ตามลำดับและคลาสที่ 4 (เท้า) บนอิเล็กโทรด C3 และ C4 แสดงในรูปที่ 4.32 และรูปที่ 4.33 ตามลำดับ จากผลการทดลองพบว่าค่าสัมประสิทธิ์ในชั้นที่ 4 จะมีขนาดมากที่สุดและ

ตามด้วยค่าสัมประสิทธิ์ในชั้นที่ 3 ซึ่งเป็นสัมประสิทธิ์ที่อยู่ในช่วงความถี่ 8-16 Hz และ 16-32 Hz ตามลำดับ ซึ่งความถี่ดังกล่าวเป็นช่วงความถี่ที่เกี่ยวข้องกับ MI



รูปที่ 4.26 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C3

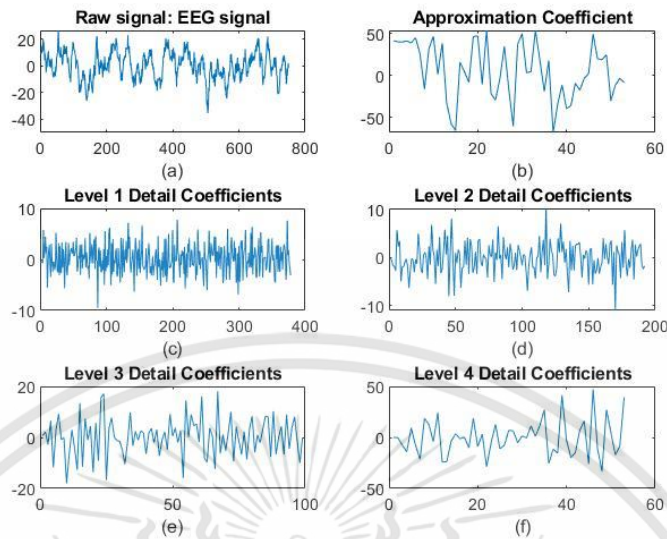
- (a) สัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
(c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
(f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4



รูปที่ 4.27 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C4

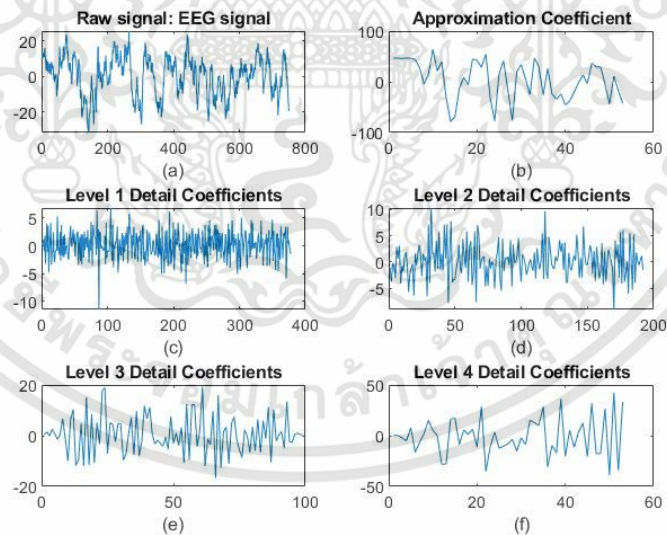
- (a) สัญญาณ EEG คลาสมือซ้าย อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
(c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
(f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



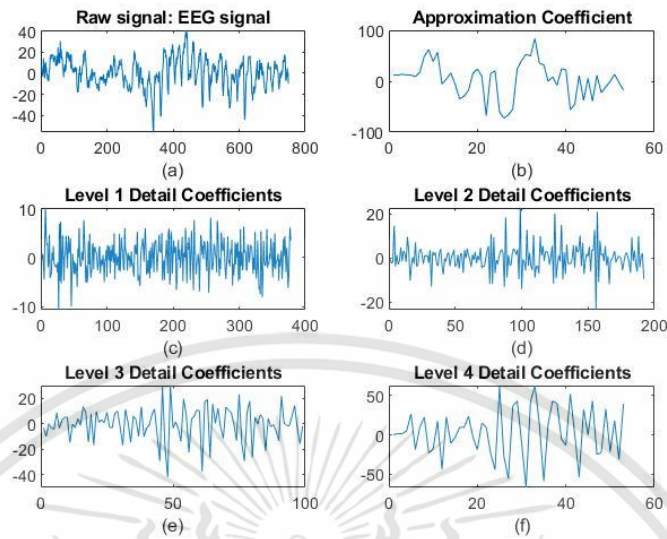
รูปที่ 4.28 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือขวา อิเล็กโทรด C3

- (a) สัญญาณ EEG คลาสมือขวา อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4



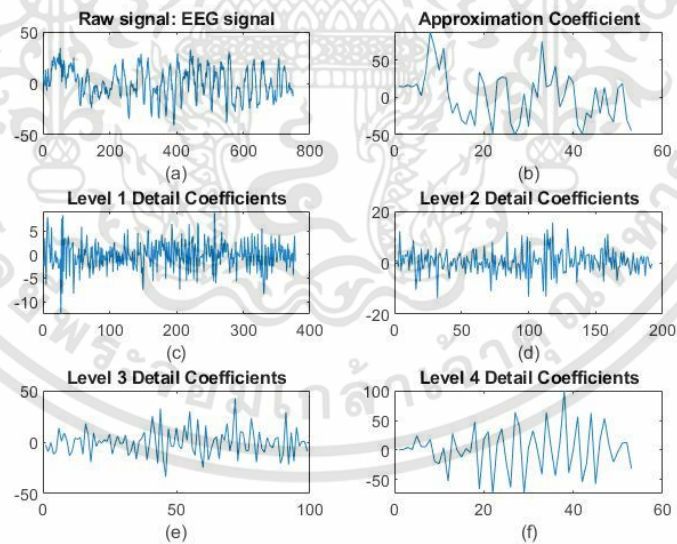
รูปที่ 4.29 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสมือขวา อิเล็กโทรด C4

- (a) สัญญาณ EEG คลาสมือขวา อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4



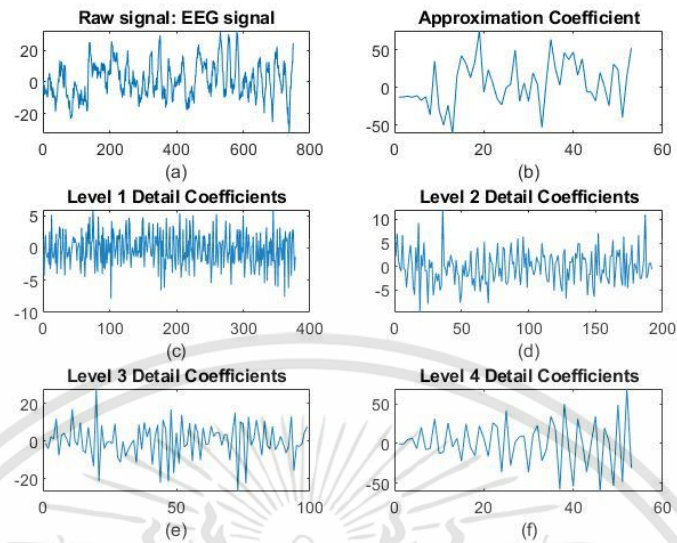
รูปที่ 4.30 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสลิ่ง อิเล็กโทรด C3

- (a) สัญญาณ EEG คลาสลิ่ง อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4



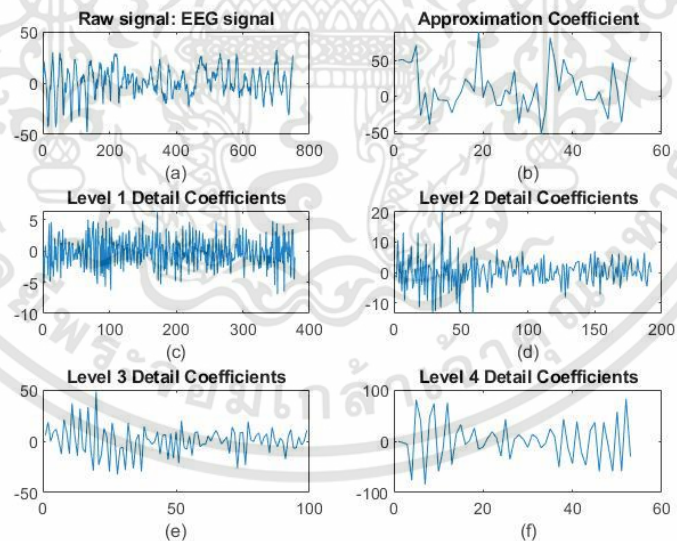
รูปที่ 4.31 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสลิ่ง อิเล็กโทรด C4

- (a) สัญญาณ EEG คลาสลิ่ง อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4



รูปที่ 4.32 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสเท้า อิเล็กโทรด C3

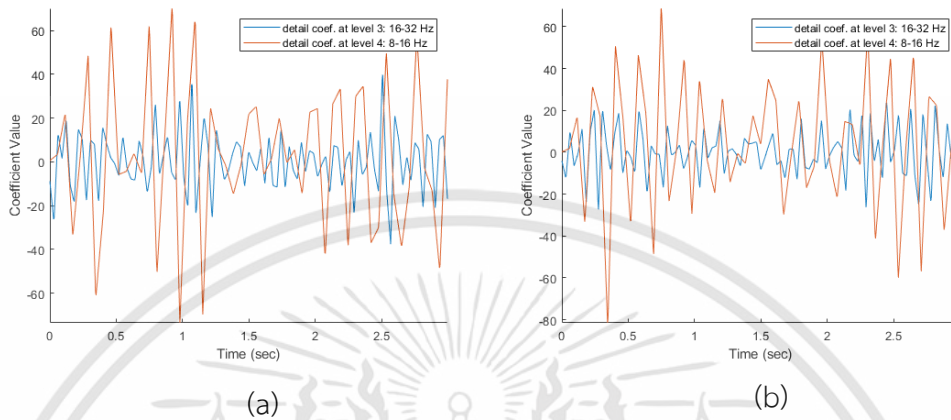
- (a) สัญญาณ EEG คลาสเท้า อิเล็กโทรด C3 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4



รูปที่ 4.33 การแปลงเวฟเลตแบบดิสครีตของสัญญาณ EEG คลาสเท้า อิเล็กโทรด C4

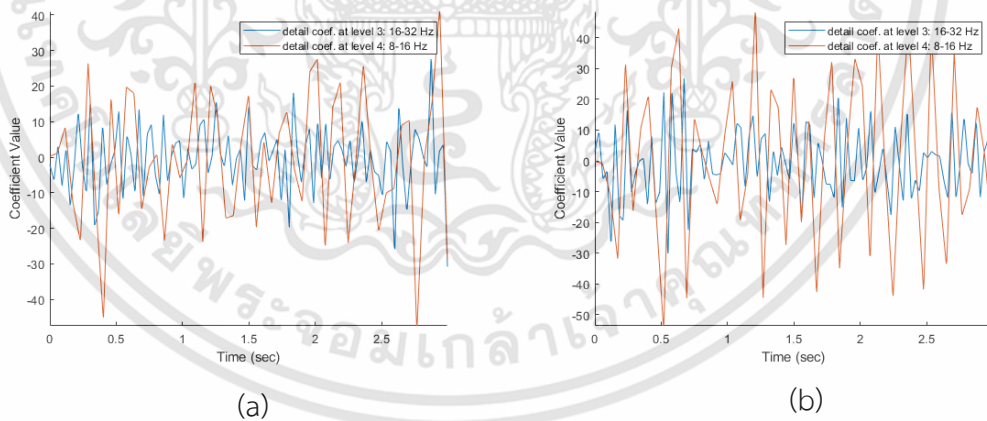
- (a) สัญญาณ EEG คลาสเท้า อิเล็กโทรด C4 (b) ค่าสัมประสิทธิ์ประมาณชั้นที่ 4  
 (c) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 1 (d) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 2 (e) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3  
 (f) ค่าสัมประสิทธิ์ละเอียดชั้นที่ 4

ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสมือซ้ายของอิเล็กโทรด C3 และอิเล็กโทรด C4 แสดงในรูปที่ 4.34 (a) และรูปที่ 4.34 (b) ตามลำดับ



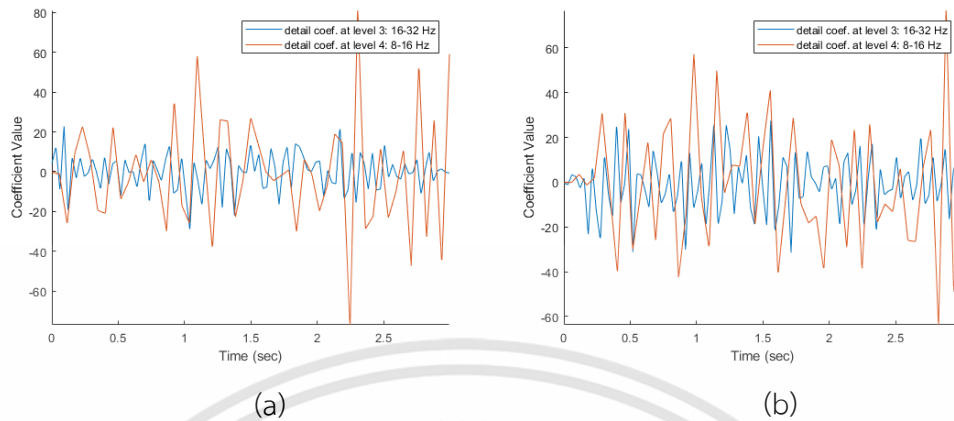
รูปที่ 4.34 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสมือซ้าย  
(a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4

ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสมือขวาของอิเล็กโทรด C3 และอิเล็กโทรด C4 แสดงในรูปที่ 4.35 (a) และรูปที่ 4.35 (b) ตามลำดับ



รูปที่ 4.35 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสมือขวา  
(a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4

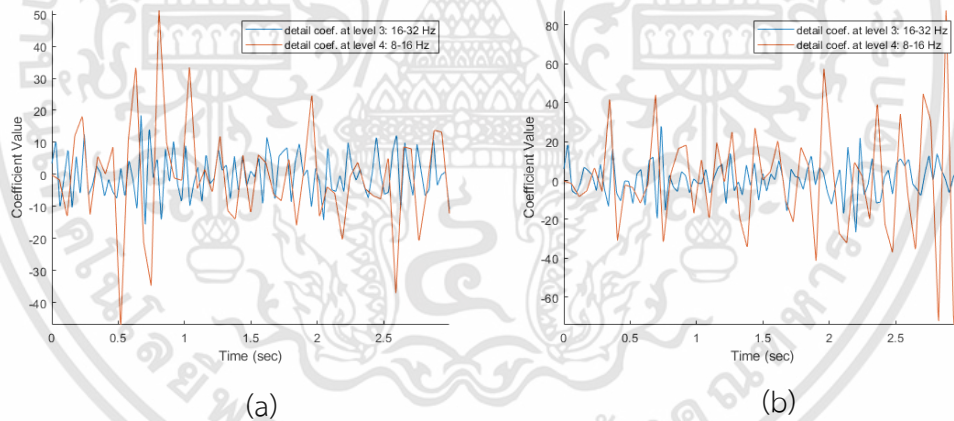
ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสลิ้นของอิเล็กโทรด C3 และอิเล็กโทรด C4 แสดงในรูปที่ 4.36 (a) และรูปที่ 4.36 (b) ตามลำดับ



รูปที่ 4.36 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสลีน

(a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4

ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสเท้าของอิเล็กโทรด C3 และอิเล็กโทรด C4 แสดงในรูปที่ 4.37 (a) และรูปที่ 4.37 (b) ตามลำดับ

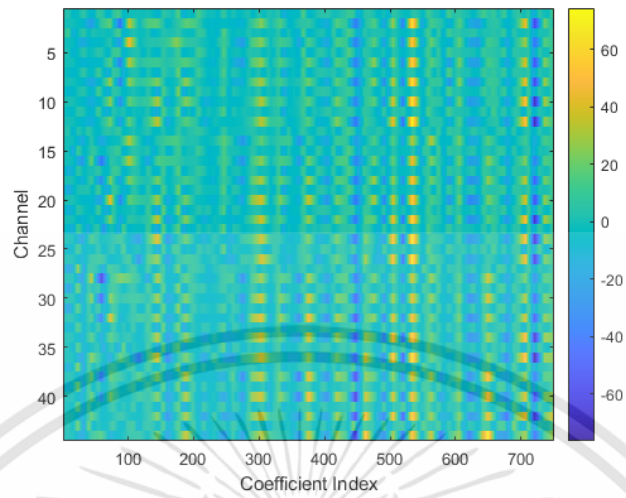


รูปที่ 4.37 ค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และ 4 ของคลาสเท้า

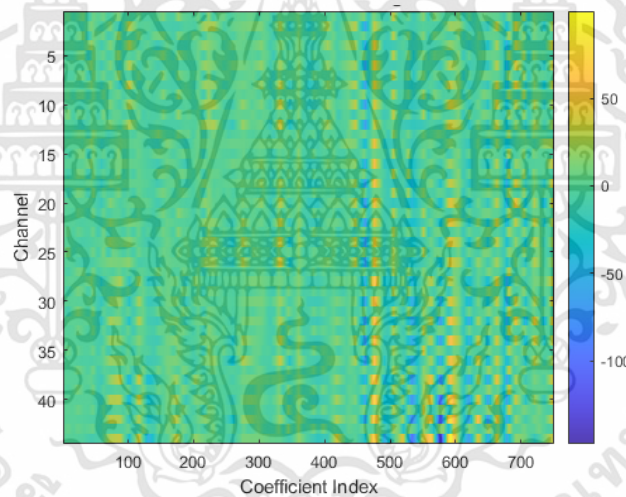
(a) อิเล็กโทรด C3 (b) อิเล็กโทรด C4

การสร้างภาพคุณลักษณะจากค่าสัมประสิทธิ์ละเอียดของ DWT ทำได้โดยให้แกน y เป็นค่าของสัมประสิทธิ์จากอิเล็กโทรดช่องที่ 1-22 จากบนลงล่างตามลำดับและมีการจัดเรียงสลับระหว่างค่าสัมประสิทธิ์ละเอียดชั้นที่ 3 และชั้นที่ 4 แกน x คือจำนวนสัมประสิทธิ์ในระยะเวลา 3 วินาที ซึ่งเมื่อคูณกับความถี่สุ่มแล้วจะได้ 750 ค่า ค่าของสัมประสิทธิ์ถูกแสดงในรูปแบบของสี ภาพคุณลักษณะจากค่าสัมประสิทธิ์ละเอียดของ DWT สำหรับคลาสมือซ้าย, มือขวา, ลีนและเท้า แสดงดังรูปที่ 4.38, 4.39, 4.40 และ 4.41 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

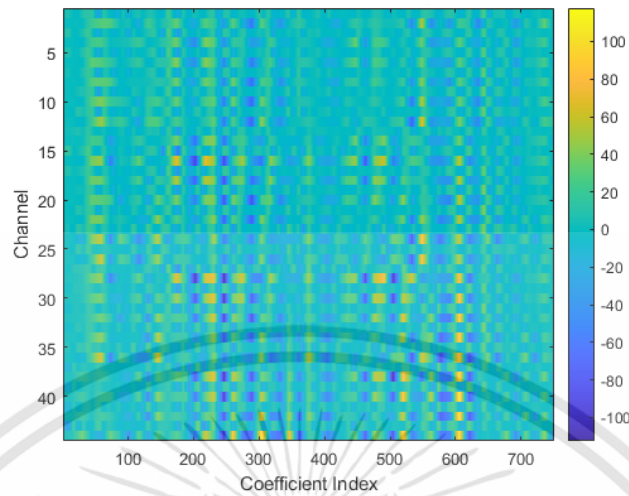


รูปที่ 4.38 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสมือซ้าย

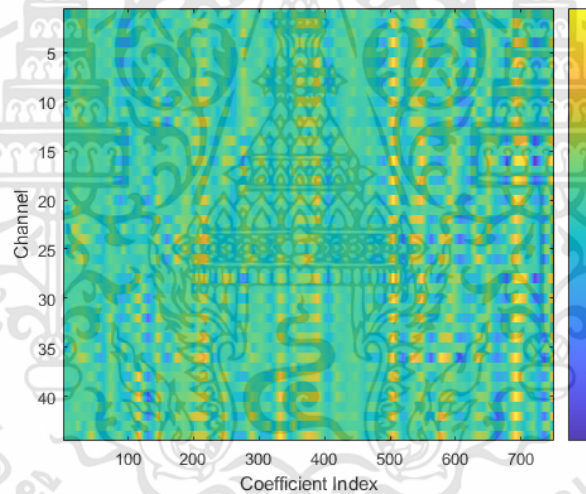


รูปที่ 4.39 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสมือขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสสิน

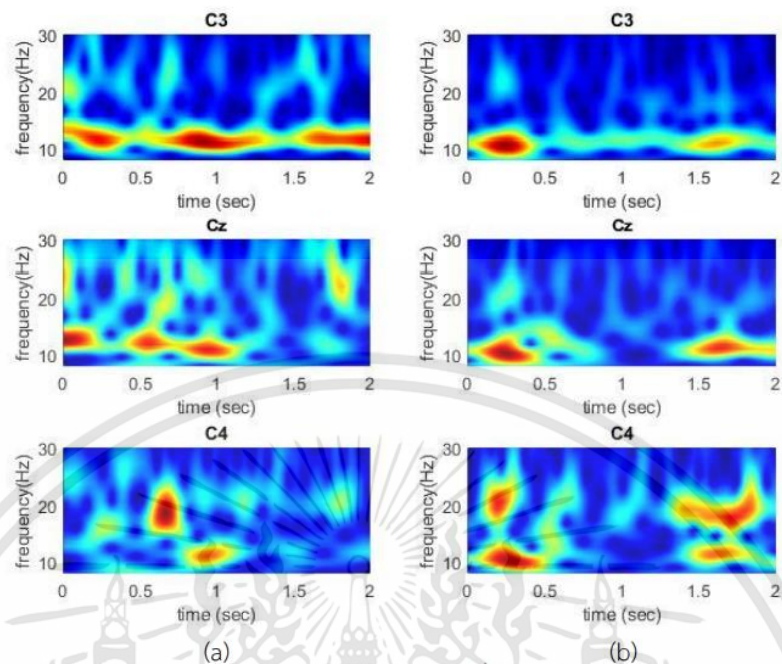


รูปที่ 4.41 ภาพคุณลักษณะจากสัมประสิทธิ์ละเอียดของคลาสเท้า

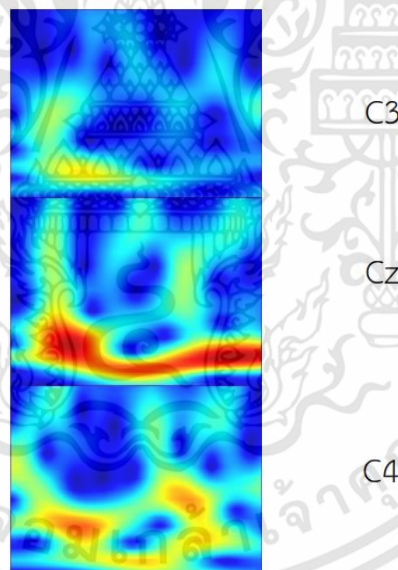
#### 4.4 การทดสอบการแปลงเวฟเลตแบบต่อเนื่องของสัญญาณ EEG

เนื่องจากสัญญาณ EEG เป็นสัญญาณแบบ Non-Stationary และ Non-Linear ดังนั้น CWT จะมีข้อดีของการเก็บข้อมูลเชิงสเปกตรัมในเชิงมิติของเวลา ซึ่งในปริภูมิตวินนี้เลือกใช้ Generalized Morse Wavelet โดยเมื่อแปลงบนชุดข้อมูล BCI Competition IV 2a แล้วจะได้ตัวอย่างผลลัพธ์ดังรูปที่ 4.42 สำหรับ MI ประเภทมือซ้ายและมือขวา โดยรูปภาพผลลัพธ์ที่ได้หลังจากการแปลงแบบต่อเนื่องนั้นจะเรียกว่า Scalogram ซึ่งรูปภาพที่ใช้สำหรับการเรียนรู้โดยอัลกอริธึมการเรียนรู้เชิงลึกนั้นจะเป็นรูปภาพที่นำ Scalogram จาก 3 Channel (C3, CZ และ C4) มาเรียงต่อกันดังรูปที่ 4.43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.42 Scalogram ของสัญญาณ EEG สำหรับ MI (a) มือซ้าย (b) มือขวา



รูปที่ 4.43 ตัวอย่าง Scalogram ที่ได้หลังจากการเรียงข้อมูลใน Channel C3, Cz และ C4

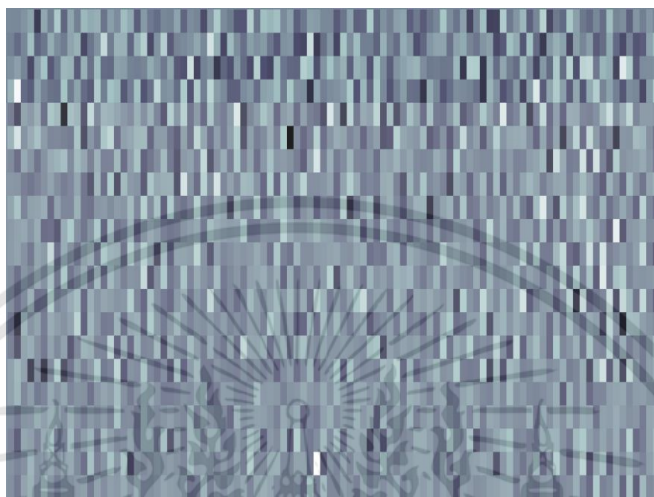
## 4.5 การทดสอบการรวมคุณลักษณะ

### 4.5.1 ผลลัพธ์การรวมคุณลักษณะของ DWT และ CSP

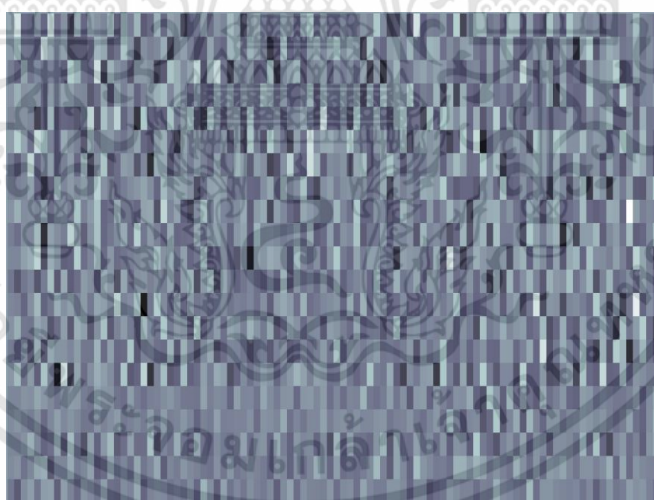
ในการทดลองนี้จะใช้ข้อมูลสัญญาณคลื่นไฟฟ้าจากสมองของ BCI Competition IV 2a ในการแสดงตัวอย่างผลลัพธ์การรวมคุณลักษณะของ DWT และ CSP โดยภาพของสัมประสิทธิ์ละเอียดชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 3 ของ DWT เมื่อผ่านการแปลง CSP ของคลาสมือซ้ายและมือขวาของชุดข้อมูลสำหรับฝึกสอนแสดง  
ในรูปที่ 4.44 และ 4.45 ตามลำดับ



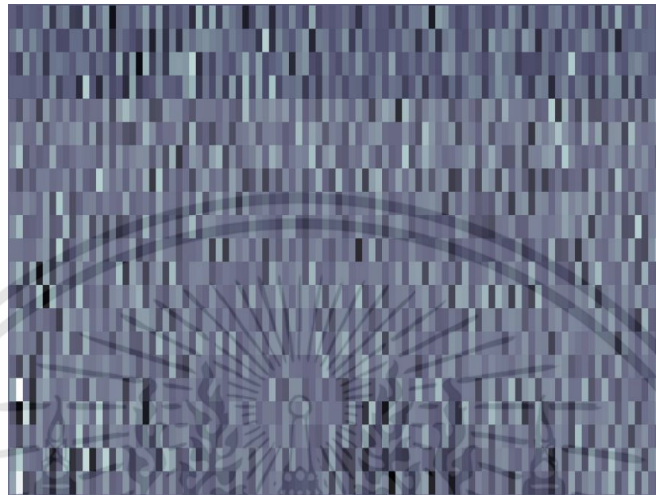
รูปที่ 4.44 สัมประสิทธิ์ละเอียดระดับที่ 3 ของ DWT คลาสมือซ้ายของข้อมูลชุดฝึกสอน  
เมื่อผ่านการแปลง CSP



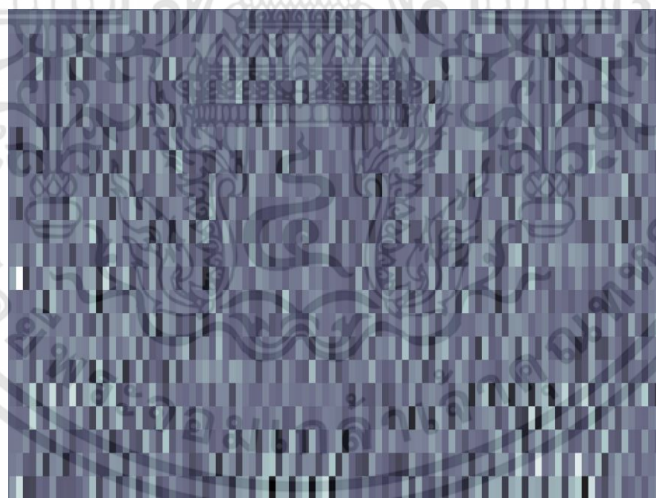
รูปที่ 4.45 สัมประสิทธิ์ละเอียดระดับที่ 3 ของ DWT คลาสมือขวาของข้อมูลชุดฝึกสอน  
เมื่อผ่านการแปลง CSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของสัมประสิทธิ์ละเอียดขั้นที่ 3 ของ DWT เมื่อผ่านการแปลง CSP ของคลาสมือซ้าย และมือขวาของชุดข้อมูลสำหรับทดสอบแสดงในรูปที่ 4.46 และ 4.47 ตามลำดับ



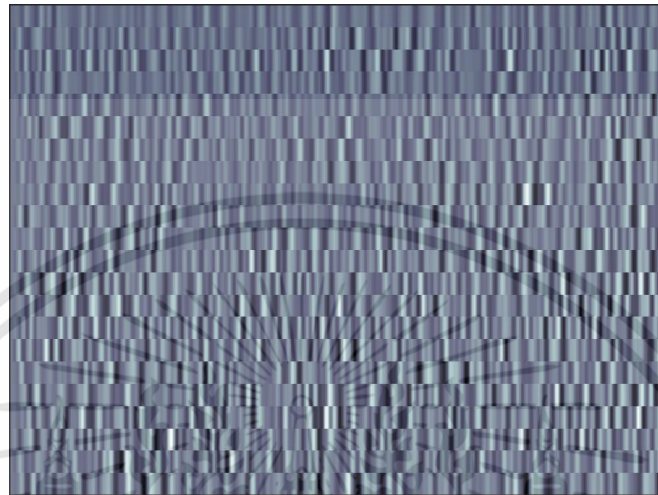
รูปที่ 4.46 สัมประสิทธิ์ละเอียดขั้นที่ 3 ของ DWT คลาสมือซ้ายของชุดทดสอบ  
เมื่อผ่านการแปลง CSP



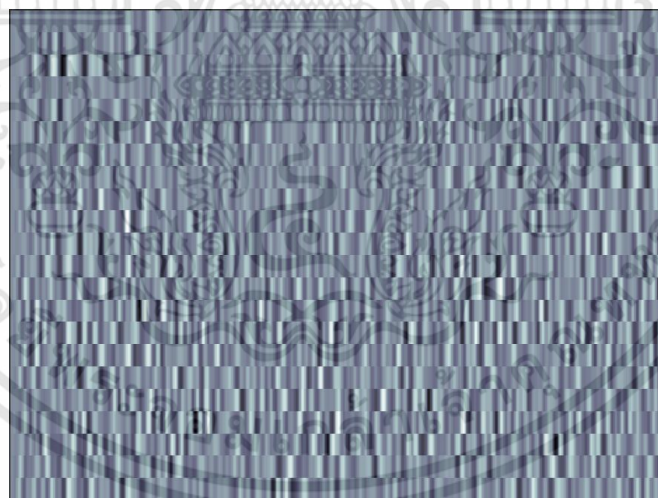
รูปที่ 4.47 สัมประสิทธิ์ละเอียดขั้นที่ 3 ของ DWT คลาสมือขวาของชุดทดสอบ  
เมื่อผ่านการแปลง CSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของสัมประสิทธิ์ละเอียดชั้นที่ 4 ของ DWT เมื่อผ่านการแปลง CSP ของคลาสมือซ้าย และมือขวาของชุดข้อมูลสำหรับฝึกสอนแสดงในรูปที่ 4.48 และ 4.49 ตามลำดับ



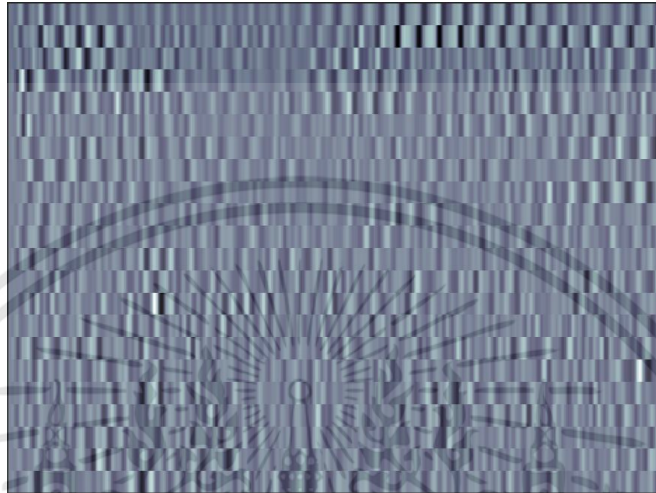
รูปที่ 4.48 สัมประสิทธิ์ละเอียดชั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดฝึกสอน  
เมื่อผ่านการแปลง CSP



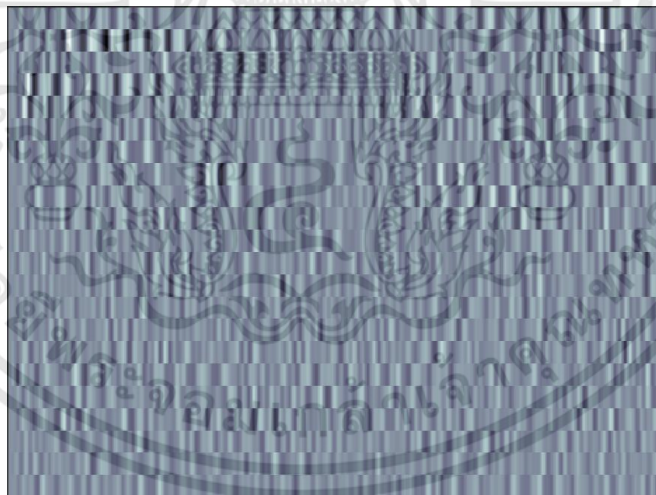
รูปที่ 4.49 สัมประสิทธิ์ละเอียดชั้นที่ 4 ของ DWT คลาสมือขวาของชุดฝึกสอน  
เมื่อผ่านการแปลง CSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของสัมประสิทธิ์ละเอียดขั้นที่ 4 ของ DWT เมื่อผ่านการแปลง CSP ของคลาสมือซ้าย และมือขวาของชุดข้อมูลสำหรับทดสอบแสดงในรูปที่ 4.50 และ 4.51 ตามลำดับ



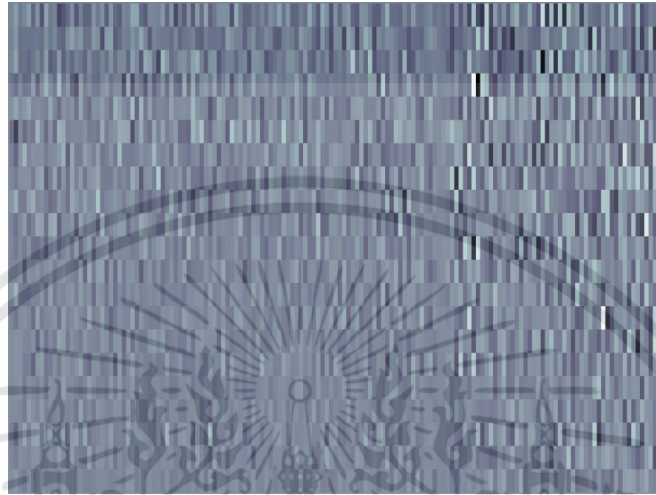
รูปที่ 4.50 สัมประสิทธิ์ละเอียดขั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดทดสอบ เมื่อผ่านการแปลง CSP



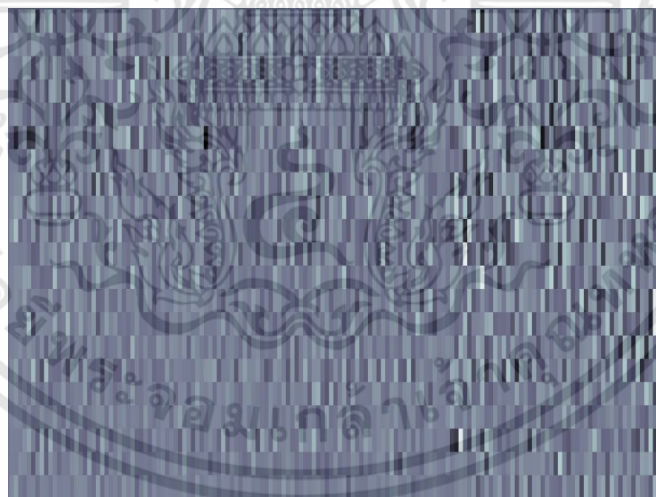
รูปที่ 4.51 สัมประสิทธิ์ละเอียดขั้นที่ 4 ของ DWT คลาสมือขวาของชุดทดสอบ เมื่อผ่านการแปลง CSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของสัมประสิทธิ์ละเอียดชั้นที่ 3 รวมกับชั้นที่ 4 ของ DWT เมื่อผ่านการแปลง CSP ของคลาสมือซ้ายและมือขวาของชุดข้อมูลสำหรับฝึกสอนแสดงในรูปที่ 4.52 และ 4.53 ตามลำดับ



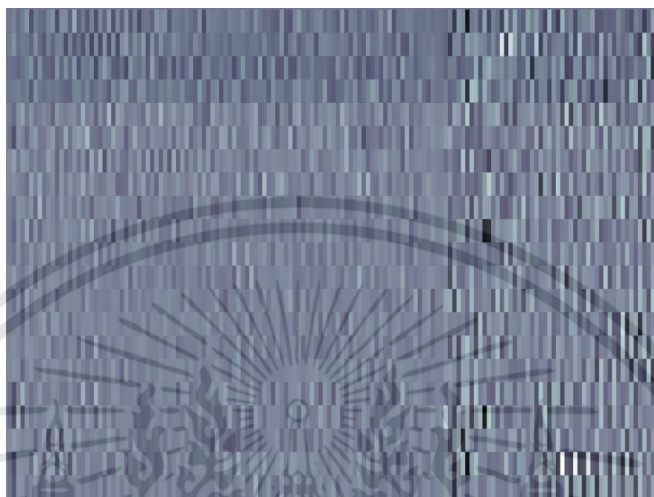
รูปที่ 4.52 สัมประสิทธิ์ละเอียดชั้นที่ 3 รวมกับชั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดฝึกสอน เมื่อผ่านการแปลง CSP



รูปที่ 4.53 สัมประสิทธิ์ละเอียดชั้นที่ 3 รวมกับชั้นที่ 4 ของ DWT คลาสมือขวาของชุดฝึกสอน เมื่อผ่านการแปลง CSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของสัมประสิทธิ์ละเอียดชั้นที่ 3 รวมกับชั้นที่ 4 ของ DWT เมื่อผ่านการแปลง CSP ของคลาสมือซ้ายและมือขวาของชุดข้อมูลสำหรับทดสอบแสดงในรูปที่ 4.54 และ 4.55 ตามลำดับ



รูปที่ 4.54 สัมประสิทธิ์ละเอียดชั้นที่ 3 รวมกับชั้นที่ 4 ของ DWT คลาสมือซ้ายของชุดทดสอบ เมื่อผ่านการแปลง CSP

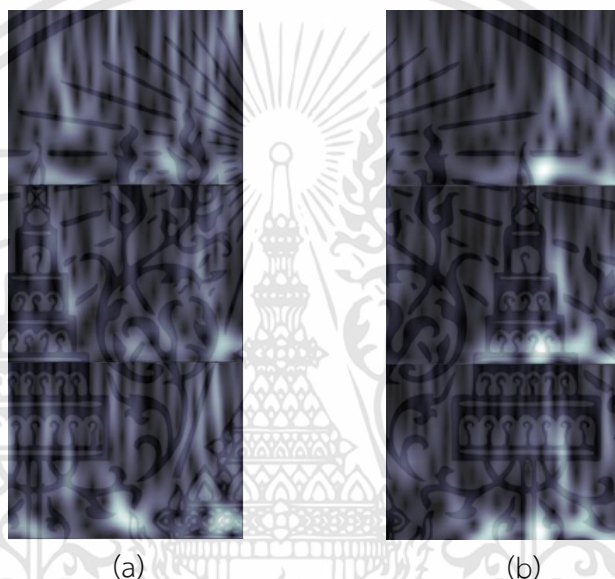


รูปที่ 4.55 สัมประสิทธิ์ละเอียดชั้นที่ 3 รวมกับชั้นที่ 4 ของ DWT คลาสมือขวาของชุดทดสอบ เมื่อผ่านการแปลง CSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

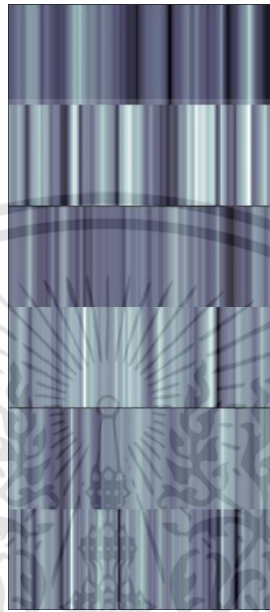
#### 4.5.2 ผลลัพธ์การรวมคุณลักษณะของ CWT และ CSP

ในการทดลองนี้จะใช้ข้อมูลสัญญาณสมองจาก BCI Competition IV 2a ในการแสดงตัวอย่างในการแสดงผลการรวมคุณลักษณะของ CWT และ CSP ซึ่งการสร้างภาพคุณลักษณะจากสัมประสิทธิ์การแปลงแบบต่อเนื่องทำโดยนำภาพสัมประสิทธิ์ของแต่ละอิเล็กโทรดมาเรียงต่อกัน ส่วนบนคือสัมประสิทธิ์จากอิเล็กโทรด C3 ส่วนกลางคือสัมประสิทธิ์จากอิเล็กโทรด Cz และส่วนล่างคือสัมประสิทธิ์จากอิเล็กโทรด C4 ภาพสัมประสิทธิ์การแปลงแบบต่อเนื่องของคลาสมือซ้ายและมือขวาแสดงดังรูปที่ 4.56 (a) และ 4.56 (b) ตามลำดับ



รูปที่ 4.56 ภาพสัมประสิทธิ์การแปลงแบบต่อเนื่อง (a) คลาสมือซ้าย และ (b) คลาสมือขวา

ภาพของสัมประสิทธิ์ CWT เมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกของเมทริกซ์ W สำหรับชุดข้อมูลฝึกสอนของคลาสมือซ้ายและมือขวาแสดงดังรูปที่ 4.57 และ 4.58 ตามลำดับ



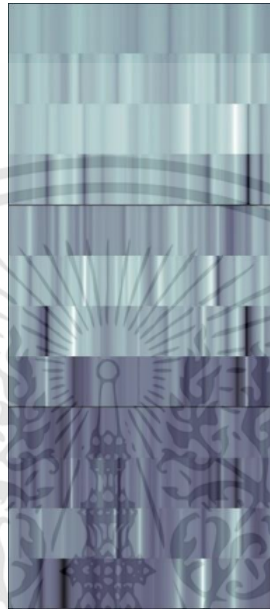
รูปที่ 4.57 ภาพสัมประสิทธิ์การแปลงแบบต่อเนื่องของคลาสมือซ้ายเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกของเมทริกซ์ W



รูปที่ 4.58 ภาพสัมประสิทธิ์การแปลงแบบต่อเนื่องของคลาสมือขวาเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกของเมทริกซ์ W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของสัมประสิทธิ์ CWT เมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกและสองคอลัมน์สุดท้ายของเมทริกซ์  $W$  สำหรับชุดข้อมูลฝึกสอนของคลาสมือซ้ายและมือขวาแสดงดังรูปที่ 4.59 และ 4.60 ตามลำดับ



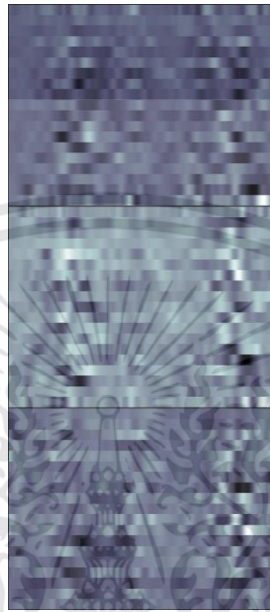
รูปที่ 4.59 ภาพสัมประสิทธิ์การแปลงแบบต่อเนื่องของคลาสมือซ้ายเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกและสองคอลัมน์สุดท้ายของเมทริกซ์  $W$



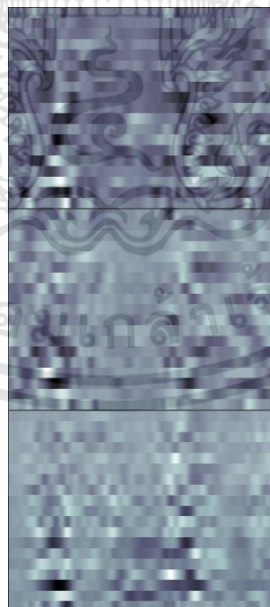
รูปที่ 4.60 ภาพสัมประสิทธิ์การแปลงแบบต่อเนื่องของคลาสมือขวาเมื่อผ่านการแปลง CSP โดยใช้สองคอลัมน์แรกและสองคอลัมน์สุดท้ายของเมทริกซ์  $W$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของสัมประสิทธิ์ CWT เมื่อผ่านการแปลง CSP โดยใช้ทุกคอลัมน์ของเมทริกซ์ W สำหรับชุดข้อมูลฝึกสอนของคลาสมือซ้ายและมือขวาแสดงดังรูปที่ 4.61 และ 4.62 ตามลำดับ



รูปที่ 4.61 สัมประสิทธิ์การแปลงแบบต่อเนื่องของคลาสมือซ้ายเมื่อผ่านการแปลง CSP โดยใช้ทุกคอลัมน์ของเมทริกซ์ W



รูปที่ 4.62 สัมประสิทธิ์การแปลงแบบต่อเนื่องของคลาสมือขวาเมื่อผ่านการแปลง CSP โดยใช้ทุกคอลัมน์ของเมทริกซ์ W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.6 ระยะเวลาที่ใช้ในประมวลผลการแยกคุณลักษณะและการจำแนกด้วยการเรียนรู้เชิงลึก

เวลาที่ใช้ในส่วนของการแยกคุณลักษณะนั้น CSP กับ CWT ใช้เวลาในการประมวลผล 0.24 วินาทีและใช้เวลาในการส่งออกรูปภาพหนึ่งรูปภาพ 0.60 วินาทีรวมแล้วใช้เวลา 0.87 วินาที ในขณะที่ CSP กับ DWT ใช้เวลาในการประมวลผล 0.37 วินาทีและใช้เวลาในการส่งออกรูปภาพหนึ่งรูปภาพ 0.2 วินาทีรวมแล้วใช้เวลา 0.57 วินาที ในส่วนของเวลาที่ใช้ในการจำแนกข้อมูลด้วยอัลกอริธึมการเรียนรู้เชิงลึกแบบต่างๆ แสดงดังตารางที่ 4.1

ตารางที่ 4.1 เวลาที่ใช้ในการจำแนกข้อมูลด้วยอัลกอริธึมการเรียนรู้เชิงลึก

Classification methods	Computation time (s)	
Pre-Trained Model	ResNet18	0.05
	ResNet50	0.08
	ShuffleNet	0.03
MLP	0.06	
LeNet-5	0.12	
HDNN	0.47	

## 4.7 การทดสอบการจำแนกคลาสด้วยอัลกอริธึมการเรียนรู้เชิงลึกด้วยชุดข้อมูลชุดทดสอบจาก BCI Competition IV 2a

### 4.7.1 มิติข้อมูล

ผู้จัดทำได้เตรียมชุดข้อมูลสำหรับฝึกสอนและทดสอบประสิทธิภาพของโมเดล โดยมีมิติของข้อมูลจากวิธีการแยกคุณลักษณะในแบบต่างๆ รวม 9 แบบแสดงดังตารางที่ 4.2 โดย CWT ( $\cdot^{\wedge}2$ ) หมายถึงการยกกำลังสองค่าสัมประสิทธิ์ CWT

ตารางที่ 4.2 มิติของข้อมูลหลังผ่านการแยกคุณลักษณะด้วยวิธีต่างๆ

Feature extraction methods	Dimension of each trial (pixels)	
CSP	22 x 750	
DWT	cd4	22 x 53
	cd3	22 x 99
	cd3 + cd4	22 x 152
DWT + CSP	cd4	22 x 53
	cd3	22 x 99
	cd3 + cd4	22 x 152
CWT + CSP	57 x 750	
CWT( $\cdot^{\wedge}2$ ) + CSP	57 x 750	

### 4.7.2 ผลการทดสอบการจำแนกคลาสของข้อมูลด้วยอัลกอริธึมการเรียนรู้เชิงลึก

คุณลักษณะทั้ง 9 แบบในตารางที่ 4.3 ถูกนำไปใช้เป็นอินพุตเพื่อทดสอบโครงสร้างของอัลกอริธึมการเรียนรู้เชิงลึกทั้ง 6 โครงสร้าง ผลการทดสอบอัลกอริธึมการเรียนรู้เชิงลึกบนข้อมูลจาก Subject ที่ 9 ภายหลังมีการปรับพารามิเตอร์ให้ดีที่สุดได้ผลลัพธ์แสดงดังตารางที่ 4.3

ตารางที่ 4.3 ผลลัพธ์การจำแนกคลาสข้อมูลของ Subject ที่ 9 โดยใช้คุณลักษณะทั้ง 9 แบบด้วยอัลกอริธึมการเรียนรู้เชิงลึก

Feature Extraction Methods	Classification Methods	Accuracy (%)
CSP	MLP	50.00
	LeNet-5	39.29

ตารางที่ 4.3 ผลลัพธ์การจำแนกคลาสข้อมูลของ Subject ที่ 9 โดยใช้คุณลักษณะทั้ง 9 แบบด้วย อัลกอริธึมการเรียนรู้เชิงลึก

Feature Extraction Methods	Classification Methods	Accuracy (%)
CSP	ResNet18	57.14
	ResNet50	57.14
	ShuffleNet	50.00
	HDNN	82.14
	MLP	52.86
DWT	LeNet-5	64.29
	ResNet18	89.29
	ResNet50	82.14
	ShuffleNet	82.14
	HDNN	57.14
	MLP	61.43
	LeNet-5	60.71
	ResNet18	89.29
	ResNet50	82.14
	ShuffleNet	75.00
DWT + CSP	MLP	51.43
	LeNet-5	50.00
	ResNet18	82.14
	ResNet50	78.57
	ShuffleNet	82.14
	MLP	52.86
	LeNet-5	50.00
DWT + CSP	ResNet18	82.14
	ResNet50	82.14
	ResNet18	82.14
	ResNet50	82.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 (ต่อ) ผลลัพธ์การจำแนกคลาสข้อมูลของ Subject ที่ 9 โดยใช้คุณลักษณะทั้ง 9 แบบด้วย อัลกอริธึมการเรียนรู้เชิงลึก

Feature Extraction Methods	Classification Methods	Accuracy (%)
cd4	ShuffleNet	82.14
	HDNN	78.57
	MLP	52.86
	LeNet-5	50.00
cd3	ResNet18	82.14
	ResNet50	82.14
DWT + CSP	ShuffleNet	82.14
	MLP	51.43
	LeNet-5	39.29
	ResNet18	78.57
cd3 + cd4	ResNet50	82.14
	ShuffleNet	75.00
	HDNN	85.71
	MLP	51.40
CWT + CSP	LeNet-5	78.57
	ResNet18	<b>92.86</b>
	ResNet50	82.14
	ShuffleNet	85.71
	HDNN	75.00
	MLP	55.00
CWT(.^2) + CSP	LeNet-5	78.57
	ResNet18	83.33
	ResNet50	78.57
	ShuffleNet	78.57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 (ต่อ) ผลลัพธ์การจำแนกคลาสข้อมูลของ Subject ที่ 9 โดยใช้คุณลักษณะทั้ง 9 แบบด้วย อัลกอริธึมการเรียนรู้เชิงลึก

Feature Extraction Methods	Classification Methods	Accuracy (%)
CWT(. <sup>^</sup> 2) + CSP	HDNN	<b>89.29</b>
	MLP	54.06
	LeNet-5	68.97
	ResNet18	87.71
CWT	ResNet50	78.57
	ShuffleNet	75.00
	HDNN	75.86
	MLP	52.84
CWT(. <sup>^</sup> 2)	LeNet-5	68.97
	ResNet18	89.28
	ResNet50	89.29
	ShuffleNet	82.14
	HDNN	65.52

จากผลการทดสอบพบว่าเมื่อใช้ CSP ร่วมกับ CWT และใช้การเรียนรู้เชิงลึกโครงสร้าง ResNet18 และ CSP ร่วมกับ CWT กำลังสอง และใช้การเรียนรู้เชิงลึกประเภท HDNN ให้ค่าความแม่นยำในการจำแนกที่ดีที่สุดคือ 92.86% และ 89.29% ตามลำดับ ผู้จัดทำจึงนำกระบวนการทั้งสองแบบข้างต้นไปทำ 5-Fold Cross Validation เพิ่มเติม ซึ่งผลการทดสอบแสดงในตารางที่ 4.4 โดยเมื่อทำ 5-Fold Cross Validation พบว่าการใช้ CSP ร่วมกับ CWT และใช้การเรียนรู้เชิงลึกโครงสร้าง ResNet18 ได้ค่าความแม่นยำในการจำแนกเฉลี่ยของผู้เข้าร่วมการทดลองทั้ง 9 คนเท่ากับ 60.25% ซึ่งสูงกว่าการใช้ CSP ร่วมกับ CWT กำลังสองและใช้การเรียนรู้ประเภท HDNN ซึ่งมีค่าความแม่นยำในการจำแนกเฉลี่ยเท่ากับ 59.55%

ตารางที่ 4.4 ผลการทำ 5-Fold Cross Validation ของการฝึกสอนโมเดลโครงสร้าง ResNet18 และ HDNN

		5-Fold Cross Validation					
Classification Methods	Subject	cv1	cv2	cv3	cv4	cv5	Average
HDNN	S1	42.86	35.71	60.71	64.28	60.71	52.85
	S2	57.14	46.43	46.43	60.71	60.71	54.28
	S3	71.43	71.43	71.43	78.57	85.71	75.71
	S4	42.86	71.43	46.43	57.14	50.00	54.47
	S5	50.00	46.43	60.71	53.57	53.57	53.57
	S6	50.00	50.00	50.00	50.00	50.00	50.00
	S7	60.71	53.57	50.00	67.85	42.86	42.32
	S8	53.57	71.43	46.43	60.71	46.43	55.71
	S9	71.43	89.28	71.43	71.43	78.57	76.43
		<b>Average</b>	<b>57.14</b>	<b>60.71</b>	<b>57.65</b>	<b>64.28</b>	<b>61.22</b>
ResNet18	S1	55.36	57.14	41.07	55.36	51.79	52.14
	S2	50.00	55.36	44.64	51.79	41.07	48.22
	S3	80.36	82.14	66.07	76.79	83.93	77.86
	S4	51.79	57.14	51.79	55.36	48.21	52.86
	S5	58.93	57.14	62.50	48.21	39.29	53.21
	S6	66.07	62.50	51.79	64.29	51.79	59.29
	S7	46.43	50.00	51.79	64.29	51.79	53.58
	S8	66.07	50.00	53.57	62.50	62.50	61.16
	S9	73.21	85.71	85.71	82.14	92.86	83.93
		<b>Average</b>	<b>62.28</b>	<b>65.3</b>	<b>56.55</b>	<b>62.3</b>	<b>58.14</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.8 การทดสอบการจำแนกคลาสด้วยอัลกอริธึมการเรียนรู้เชิงลึกด้วยชุดข้อมูลคลื่นไฟฟ้าจากสมองของผู้จัดทำ

จากตารางที่ 4.4 อัลกอริธึมการเรียนรู้เชิงลึกที่มีประสิทธิภาพในการจำแนกคลาสข้อมูลดีที่สุดใน 2 โมเดล คือ ResNet18 และ HDNN ดังนั้นปริภูมิตดงนี้จึงเลือกใช้ 2 โมเดลดังกล่าวในการจำแนกคลาสของสัญญาณคลื่นไฟฟ้าจากสมองของผู้จัดทำ โดยผลของการทดสอบการจำแนกคลาสโดยใช้พารามิเตอร์ที่ดีที่สุดแสดงดังตารางที่ 4.5

ตารางที่ 4.5 ผลการจำแนกคลาสด้วยอัลกอริธึมการเรียนรู้เชิงลึกด้วยชุดข้อมูลชุดทดสอบของผู้จัดทำ

Subject	Feature Extraction	Classification	Accuracy (%)			
			ss 1+2	ss 3+4	ss 1+2+3+4	ss 5+6
S1	CWT	ResNet18	<b>68.97</b>	63.79	58.62	-
	CWT + CSP		60.34	56.90	57.76	-
	CWT	HDNN	48.28	48.28	51.72	-
	CWT + CSP		58.62	60.34	52.59	-
S2	CWT	ResNet18	51.72	56.90	54.31	53.45
	CWT + CSP		55.17	51.72	56.00	<b>58.62</b>
	CWT	HDNN	41.38	48.28	51.72	48.28
	CWT + CSP		50.00	51.72	50.00	48.28

จากตารางที่ 4.5 พบว่า Subject ที่ 1 และ Subject ที่ 2 มีค่าความแม่นยำมากที่สุดคือ 68.97% และ 58.62% ตามลำดับ เนื่องจากผลของค่าความแม่นยำใน Subject ที่ 1 มีค่ามากที่สุด จึงเลือก Subject ที่ 1 เป็นผู้ทดลองในระบบ Real-Time System Testing นอกจากนี้พบว่าในทุก Session ของทั้งสอง Subject โมเดล ResNet18 ให้ค่าความแม่นยำที่สุดในส่วนของการจำแนกข้อมูล ดังนั้นในปริภูมิตดงนี้จึงเลือกโมเดลดังกล่าวในการใช้งานในระบบ Real-Time System Testing โดยใช้การสกัดคุณลักษณะด้วย CWT

## 4.9 การทดสอบระบบ Offline System Testing สำหรับข้อมูล BCI Competition IV 2a

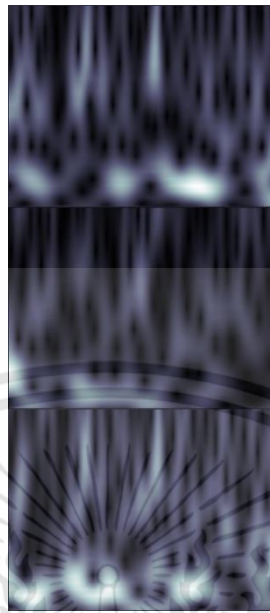
ผู้จัดทำเลือกข้อมูลจาก BCI Competition IV 2a โดยเลือก Subject ที่ 9 ซึ่งเป็น Subject ที่เมื่อทำการฝึกสอนโมเดลการเรียนรู้เชิงลึกด้วย CWT แล้วได้ผลการจำแนกถูกต้องที่สุด โดยทำการแบ่งข้อมูลในการฝึกสอน 80% ซึ่งเท่ากับ 115 การทดลองย่อยและข้อมูลสำหรับการทดสอบ 20% เท่ากับ 29 การทดลองย่อยต่อ Task

ในส่วนการสกัดคุณลักษณะผู้จัดทำได้เลือกใช้ CWT ในการสร้าง Feature Image และเลือกใช้โมเดล ResNet18 ในการฝึกสอนและทดสอบ ซึ่งค่าความแม่นยำในการจำแนกของชุดข้อมูลทดสอบได้มีค่าเท่ากับ 84.48% Confusion Matrix ของการจำแนกชุดข้อมูลทดสอบแสดงดังรูปที่ 4.63



รูปที่ 4.63 Confusion Matrix ของการจำแนกชุดข้อมูลทดสอบ

เมื่อนำโมเดลที่ได้ไปใช้ในระบบ Offline System Testing โดยนำสัญญาณข้อมูลหนึ่งการทดลองย่อยของ MI มือซ้ายและมือขวาเป็นอินพุตให้ระบบเพื่อทดสอบ พบว่าระบบสามารถควบคุมแขนกลได้ เมื่อทดลองนำ MI มือซ้าย 1 การทดลองย่อยมาใส่เป็นอินพุตของระบบแล้ว ผลลัพธ์การแปลง CWT จะได้ดังรูปที่ 4.64 และเมื่อโมเดลมีการจำแนกรูป CWT แล้วจะส่งผลการจำแนกไปควบคุมแขนกลจำลองได้ถูกต้องดังรูปที่ 4.65 โดยผลลัพธ์การแปลง CWT และรูปภาพการควบคุมแขนกลจำลองของการใช้ MI มือขวา 1 การทดลองย่อยเป็นอินพุตแสดงดังรูปที่ 4.66 และ 4.67 ตามลำดับ

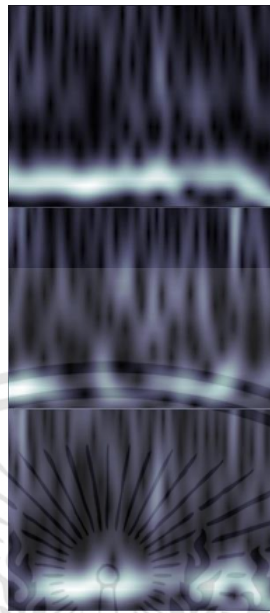


รูปที่ 4.64 CWT ของ MI มือซ้าย

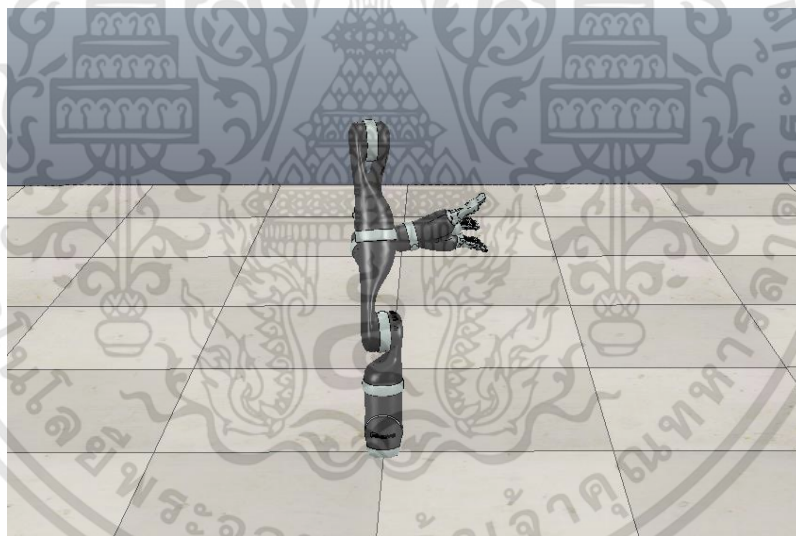


รูปที่ 4.65 ผลการควบคุมแขนกลจำลองของ MI มือซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.66 CWT ของ MI มือขวา



รูปที่ 4.67 ผลการควบคุมแขนกลจำลองของ MI มือขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

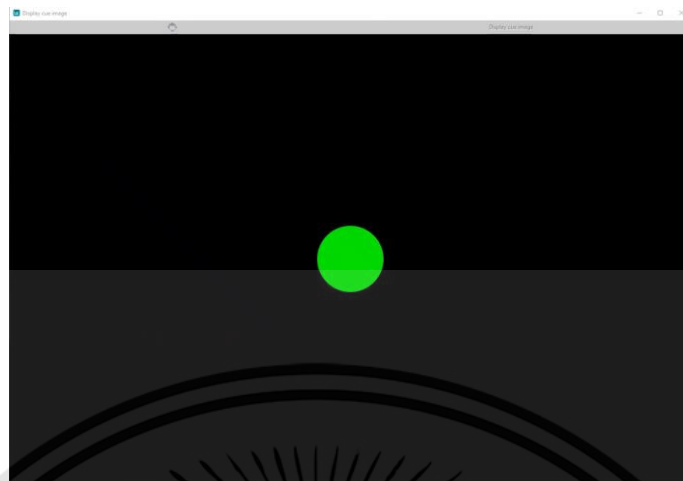
#### 4.10 การทดสอบการทำงานของระบบโดยรวมแบบ Real-Time System Testing

ผู้จัดทำนำโมเดลที่ดีที่สุดจากหัวข้อ 4.8 มาใช้ในระบบ Real-Time System Testing โดยใช้ข้อมูลคลื่นไฟฟ้าจากสมองของ Subject ที่ 1 มาใช้ในการทดสอบระบบ โดยในการทดสอบทั้งหมด 50 ครั้ง พบว่าโมเดลทำนายถูกทั้งหมด 31 ครั้ง ซึ่งคิดเป็น 62 % เมื่อพิจารณาเรื่องเวลาที่ใช้ในแต่ละส่วน พบว่าส่วนในการแปลง CWT ใช้เวลาเฉลี่ย 2.01 วินาที ในส่วนของการจำแนกรูปภาพ CWT ใช้เวลาเฉลี่ย 0.99 วินาที และในส่วนของการควบคุมแขนกลใช้เวลาเฉลี่ย 2.36 วินาที โดยเวลาเฉลี่ยหลังจากมีการจินตนาการเคลื่อนไหวเกิดขึ้น ระบบใช้เวลาทั้งหมด 5.36 วินาที

ในการทดลองเมื่อทำการเริ่มต้นการทดลองโดยกดปุ่ม a บนคีย์บอร์ด หน้าจอจะแสดงจุดสีแดงเป็นระยะเวลา 2 วินาทีดังรูปที่ 4.68 และหลังจากนั้นจะแสดงจุดสีเขียวบนหน้าจอเป็นระยะเวลา 3 วินาทีซึ่งเป็นช่วงการจินตนาการ MI ดังรูปที่ 4.69 ซึ่งเมื่อจินตนาการเสร็จแล้วจะได้สัญญาณบน Channel C3 ที่ยังไม่ผ่านการ Normalize ดังรูปที่ 4.70 และเมื่อทำการ Normalize สัญญาณแล้วจะได้ดังรูปที่ 4.71 หลังจากนั้นนำสัญญาณที่ผ่านการ Normalize แล้วไปทำการแปลง CWT จะได้ดังรูปที่ 4.72 และโมเดลจะทำการจำแนกรูปภาพ CWT ดังกล่าว โดยผลการจำแนกในหน้าต่าง Command-Line Interface ของ OpenViBE Designer แสดงดังรูปที่ 4.73 ซึ่งผลการจำแนกคือ Right หรือ MI มือขวา และผลการควบคุมแขนกลจำลองแสดงดังรูปที่ 4.74



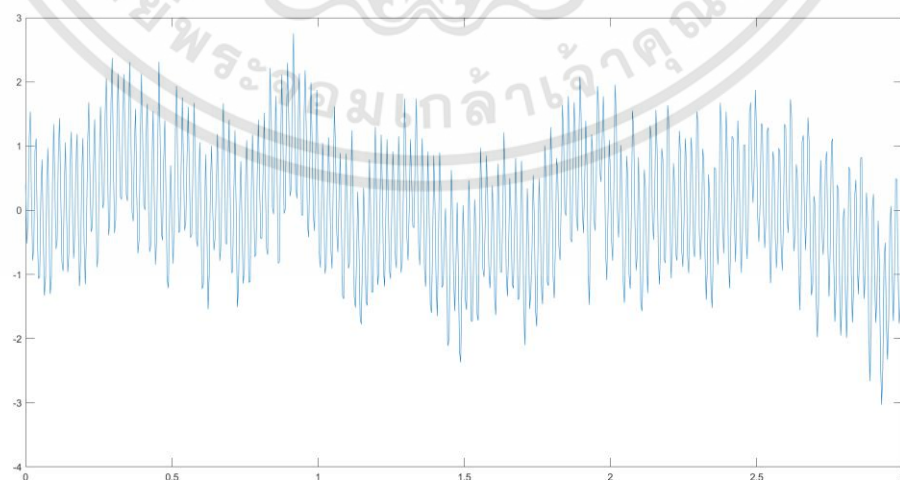
รูปที่ 4.68 หน้าจอแสดงจุดสีแดงระยะเวลา 2 วินาที



รูปที่ 4.69 หน้าจอแสดงจุดสีเขียวระยะเวลา 3 วินาที

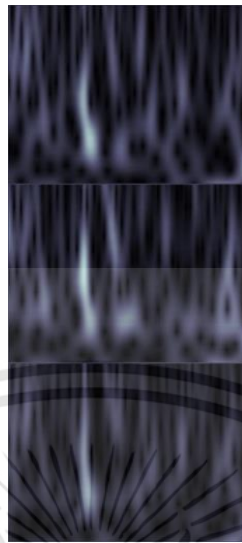


รูปที่ 4.70 สัญญาณคลื่นไฟฟ้าจากสมองบน Channel C3



รูปที่ 4.71 สัญญาณคลื่นไฟฟ้าจากสมองบน Channel C3 เมื่อผ่านการ Normalize

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.72 ผลลัพธ์การแปลง CWT

```

OpenVIBE Designer 64bit 455.000000
---- MATLAB BUFFER - INFO ----
Received stimulation code 0 at time 457.531250
[ _TIME_ ] At time 464.633 sec <box algorithm::(0x00007026, 0x000014a0) aka Matlab Scripting>
---- MATLAB BUFFER - INFO ----
Right
N =
    101
-----
simulation started
trying to connect...
Note: always make sure you use the corresponding remoteApi library
(i.e. 32bit Matlab will not work with 64bit remoteApi, and vice-versa)
clientID =
    0
remote API server connected successfully
Time of CWT : 1.583879e+00
Time of Model Prediction : 1.125293e+00
Time of Controlling : 2.604125e+00

```

รูปที่ 4.73 ผลลัพธ์การจำแนกของโมเดล



รูปที่ 4.74 ผลลัพธ์การควบคุมแขนกลด้วยคลื่นไฟฟ้าจากสมอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

ปริญญานิพนธ์นี้ประสบความสำเร็จในการศึกษาอัลกอริธึมการเรียนรู้เชิงลึกเพื่อจำแนกสัญญาณคลื่นไฟฟ้าจากสมองประเภท MI ผู้จัดทำได้ออกแบบการจัดการข้อมูล การทดสอบอัลกอริธึมสำหรับการประมวลผลคลื่นไฟฟ้าจากสมอง การจำแนกด้วยการเรียนรู้เชิงลึกและระบบรวมสำหรับการควบคุมแขนกลจำลอง ในส่วนของการประมวลผลนั้นประกอบไปด้วย 2 ส่วนคือ 1) การประมวลสัญญาณเบื้องต้นและการแยกคุณลักษณะด้วยการใช้ DWT, DWT กับ CSP, CWT และ CWT กับ CSP ในส่วนของการเรียนรู้เชิงลึกนั้นผู้จัดทำเลือกโครงสร้างทั้งหมด 6 โครงสร้าง ได้แก่ ResNet18, ResNet50, ShuffleNet, MLP, LeNet-5 และ HDNN มาศึกษา พบว่าโมเดล ResNet18 ร่วมกับคุณลักษณะ CWT ของคลื่นไฟฟ้าจากสมองได้ผลลัพธ์การจำแนกข้อมูลที่แม่นยำที่สุด ผู้จัดทำได้ทำการออกแบบระบบแบบ System Training ที่ใช้ในการจัดเก็บข้อมูลในการฝึกสอนโมเดลและระบบแบบ Real-Time System Testing เพื่อใช้ในการควบคุมแขนกลจำลองโดยได้ทำการจัดเก็บข้อมูลสัญญาณคลื่นไฟฟ้าสมองด้วยอุปกรณ์ Unicorn Hybrid Black และเชื่อมต่อระบบผ่านโปรแกรม OpenViBE Designer และ CoppeliaSim โดยแขนกลจำลองที่ผู้จัดทำเลือกใช้คือ Jaco Arm 2 ในส่วนของ Offline System Testing กับชุดข้อมูล BCI Competition IV 2a พบว่าโมเดลสามารถทำงานและจำแนกรูปภาพได้ด้วยความแม่นยำ 84.48% ผลการทดสอบใน Real-Time System Testing จำนวน 50 ครั้ง พบว่าโมเดลจำแนกคลื่นไฟฟ้าจากสมองได้ 31 ครั้ง ซึ่งคิดเป็น 62 % และเวลาเฉลี่ยทั้งหมดที่ระบบใช้เท่ากับ 5.36 วินาที จากการศึกษาในปริญญานิพนธ์นี้ ได้ข้อสรุปว่า สัญญาณคลื่นไฟฟ้าจากสมองสามารถนำไปประยุกต์ใช้ในการควบคุมแขนกลได้

#### 5.2 ปัญหาและข้อเสนอแนะ

อัลกอริธึมการเรียนรู้เชิงลึกเพื่อควบคุมแขนกลด้วยสัญญาณคลื่นไฟฟ้าจากสมองที่ได้ทำการศึกษาในปริญญานิพนธ์นี้สามารถนำไปปรับปรุงและพัฒนาให้มีความแม่นยำในการจำแนกให้สูงมากยิ่งขึ้นได้ โดยอัลกอริธึมและโมเดลที่ได้นั้นยังมีข้อบกพร่องในการจำแนก ซึ่งผลลัพธ์ด้านความแม่นยำในการจำแนกที่ได้ยังมีค่าไม่สูง เนื่องจากผู้จัดทำยังไม่มีประสบการณ์ในการจินตนาการแบบ Motor Imagery ซึ่งทำให้สัญญาณเมื่อสกัดคุณลักษณะมาแล้วยังไม่ดีพอ ส่งผลโดยตรงต่อประสิทธิภาพของโมเดล อีกทั้งยังมีข้อบกพร่องในเรื่องของเวลาที่ใช้ในระบบที่ค่อนข้างมาก อาจเป็นเพราะมีการเชื่อมต่อของโปรแกรมที่จำเป็นต้องใช้ ซึ่งหากทำให้ระบบสามารถทำงานภายในโปรแกรมเดียวกันได้ก็จะสามารถช่วยลดเวลาในการทำงานลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] Huang, Anping, Xinjiang Zhang, Runmiao Li and Yu Chi., “Memristor Neural Network Design,” 2017, doi:10.5772/intechopen.69929.
- [2] Bryn Farnsworth. “What is EEG (Electroencephalography) and How Does it Work?”. [ออนไลน์]. เข้าถึงได้จาก <https://imotions.com/blog/what-is-eeeg/#eeg-measures>. (วันที่ค้นข้อมูล 13 ธันวาคม 2564)
- [3] The McGill Physiology Virtual Lab. “Biomedical Signals Acquisition.” [ออนไลน์]. เข้าถึงได้จาก [https://www.medicine.mcgill.ca/physio/vlab/biomed\\_signals/eeeg\\_n.htm](https://www.medicine.mcgill.ca/physio/vlab/biomed_signals/eeeg_n.htm). (วันที่ค้นข้อมูล 13 ธันวาคม 2564)
- [4] OpenBCI. “OpenBCI- An Open Source Brain-Computer Interface.” [ออนไลน์]. เข้าถึงได้จาก <https://www.pantechsolutions.net/blog/openbci-an-open-source-brain-computer-interface>. (วันที่ค้นข้อมูล 13 ธันวาคม 2564)
- [5] J. Satheesh Kumar, P. Bhuvaneshwari., “Analysis of Electroencephalography (EEG) Signals and Its Categorization–A Study,” Procedia Engineering, vol. 38, pp 2525-2536, 2012, doi: 10.1016/j.proeng.2012.06.298.
- [6] The McGill Physiology Virtual Lab. “ช่วงความถี่ของสัญญาณคลื่นไฟฟ้าจากสมอง.” [ออนไลน์]. เข้าถึงได้จาก [https://www.medicine.mcgill.ca/physio/vlab/biomed\\_signals/eeeg\\_n.htm](https://www.medicine.mcgill.ca/physio/vlab/biomed_signals/eeeg_n.htm). (วันที่ค้นข้อมูล 13 ธันวาคม 2564)
- [7] Ali Al-Saegh, Shefa A. Dawwd, Jassim M. Abdul-Jabbar., “Deep learning for motor imagery EEG-based classification: A review,” Biomedical Signal Processing and Control, vol 63, 2021, doi: 10.1371/journal.pone.0234178.
- [8] Lee, H.K. Choi, Y.-S., “Application of Continuous Wavelet Transform and Convolutional Neural Network in Decoding Motor Imagery Brain-Computer Interface,” Entropy 2019, 21, 1199, doi: 10.3390/e21121199.

## บรรณานุกรม (ต่อ)

- [9] Pfurtscheller, G., Lopes da Silva, F.H., “Event-related EEG/MEG synchronization and desynchronization: Basic principles,” *Clin. Neurophysiol.* vol 110, pp. 1842–1857, 1999, doi: 10.1016/S1388-2457(99)00141-8.
- [10] Jeon, Y.; Nam, C.S.; Kim, Y.-J.; Whang, M.C., “Event-related (De)synchronization (ERD/ERS) during motor imagery tasks: Implications for brain–computer interfaces,” *Int. J. Ind. Ergon.*, vol 41, pp. 428–436, 2011.
- [11] H. K. Lee and Y. Choi., "A convolution neural networks scheme for classification of motor imagery EEG based on wavelet time-frequecy image," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, pp. 906-909, 2018, doi: 10.1109/ICOIN.2018.8343254.
- [12] Jeon, Y.; Nam, C.S.; Kim, Y.-J.; Whang, M.C., “Event-related (De)synchronization (ERD/ERS) during motor imagery tasks: Implications for brain–computer interfaces,” *Int. J. Ind. Ergon.* vol 41, pp. 428–436, 2011.
- [13] O Ang, Kai & Chin, Zheng & Wang, Chuanchu & Guan, Cuntai & Zhang, Haihong., “Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b,” *Frontiers in neuroscience*, vol. 6, no. 39, 2012, doi: 10.3389/fnins.2012.00039.
- [14] Pfurtscheller, G., and Aranibar, A., “Evaluation of event-related desynchronization (ERD) preceding and following voluntary self-paced movement,” *Electroencephalogr.* Clin. Neurophysiol, vol. 46, pp. 138–146, 1979.
- [15] Pfurtscheller, G., and Lopes da Silva, F. H., “Event-related EEG/MEG synchronization and esynchronization: basic principles,” *Clin. Neurophysiol.* vol. 110, pp. 1842–1857, 1999.
- [16] J. Chen, Z. Yu, Z. Gu and Y. Li., “Deep Temporal-Spatial Feature Learning for Motor Imagery-Based Brain–Computer Interfaces,” in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 11, pp. 2356-2366, Nov. 2020, doi: 10.1109/TNSRE.2020.3023417.

## บรรณานุกรม (ต่อ)

- [17] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe and K. Muller., “Optimizing Spatial filters for Robust EEG Single-Trial Analysis,” in IEEE Signal Processing Magazine, vol. 25, no. 1, pp. 41-56, 2008, doi: 10.1109/MSP.2008.4408441.
- [18] J. Chen, Z. Yu, Z. Gu and Y. Li., “Deep Temporal-Spatial Feature Learning for Motor Imagery-Based Brain-Computer Interfaces,” in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 11, pp. 2356-2366, Nov. 2020, doi: 10.1109/TNSRE.2020.3023417.
- [19] H. K. Lee and Y. Choi., "A convolution neural networks scheme for classification of motor imagery EEG based on wavelet time-frequency image," 2018 International Conference on Information Networking (ICOIN), Chiang Mai, pp. 906-909, 2018, doi: 10.1109/ICOIN.2018.8343254.
- [20] E. Yavuz and O. Aydemir., “Olfaction recognition by EEG analysis using wavelet transform features,” in INnovations in Intelligent SysTems and Applications (INISTA), IEEE, pp. 1-4, 2016.
- [21] B. L. Koley and D. Dey., “Detection of Chracteristic Waves of Sleep EEG by Continuous Wavelet Transform,” in National Conference on Computing and Communication Systems (NCCCS) IEEE, pp. 1-5.
- [22] al-Qerem, A., Kharbat, F., Nashwan, S., Ashraf, S., & blaou, khairi., “General model for best feature extraction of EEG using discrete wavelet transform wavelet family and differential evolution,” International Journal of Distributed Sensor Networks, doi: 10.1177/1550147720911009.
- [23] Orhan, Umut & Hekim, Mahmut & Ozer, Mahmut., “EEG gnals classification using the K-means clustering and a multilayer perceptron neural network model. Expert Syst,” Appl. vol. 38, pp. 13475-13481, 2011, doi: 10.1016/j.eswa.2011.04.149.
- [24] Orthogonal and Biorthogonal Filter Banks. [ออนไลน์]. เข้าถึงได้จาก <https://www.mathworks.com/help/wavelet/orthogonal-and-biorthogonal-filter-banks.html> (วันที่ค้นข้อมูล 10 กันยายน 2564)

## บรรณานุกรม (ต่อ)

- [25] Orthogonal wavelet. [ออนไลน์]. เข้าถึงได้จาก  
<https://www.mathworks.com/help/wavelet/ref/orthfilt.html> (วันที่ค้นข้อมูล 10 กันยายน 2564)
- [26] Carrión-Ojeda, Dustin & Fonseca-Delgado, R. & Pineda, Israel., “Analysis of factors that influence the performance of biometric systems based on EEG signals,” *Expert Systems with Applications*, vol. 165, 2020, doi: 10.1016/j.eswa.2020.113967.
- [27] ณัฐวดี หงษ์บุญมี และ ณัฐพงศ์ จันทะวงค์. การประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกเพื่อวัดระดับความหวนของแตงโม ผ่านสมาร์ตโฟน. ภาควิชาวิทยาการคอมพิวเตอร์ และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์มหาวิทยาลัยนเรศวร.
- [28] X. Gu et al., "EEG-Based Brain-Computer Interfaces (BCIs): A Survey of Recent Studies on Signal Sensing Technologies and Computational Intelligence Approaches and Their Applications," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 5, pp. 1645-1666, 1 Sept.-Oct. 2021, doi: 10.1109/TCBB.2021.3052811.
- [29] การเรียนรู้เชิงลึกและMultilayer perceptron. [ออนไลน์]. เข้าถึงได้จาก  
[https://kb.psu.ac.th/psukb/bitstream/2553/2132/8/284819\\_ch3.pdf](https://kb.psu.ac.th/psukb/bitstream/2553/2132/8/284819_ch3.pdf). (วันที่ค้นข้อมูล : 13 ธันวาคม 2564)
- [30] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner., "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
- [31] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, “Deep Residual Learning for Image Recognition,” 7, 2015.
- [32] Ramzan, F., Khan, M.U.G., Rehmat, A. et al., “A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer’s Disease Stages Using Resting-State fMRI and Residual Neural Networks,” *J Med Syst* vol. 44, no. 37, 2020, doi: 10.1007/s10916-019-1475-2.

## บรรณานุกรม (ต่อ)

- [33] Review: ShuffleNet V1 — Light Weight Model (Image Classification). [ออนไลน์]. เข้าถึงได้จาก <https://towardsdatascience.com/review-shufflenet-v1-light-weight-model-image-classification-5b253dfe982f>. (วันที่ค้นข้อมูล : 13 ธันวาคม 2564)
- [34] Zhang, R., Zong, Q., Dou, L., Zhao, X., Tang, Y., & Li, Z., "Hybrid deep neural network using transfer learning for EEG motor imagery decoding," *Biomedical Signal Processing and Control*, vol. 63, 102144, 2021, doi: 10.1016/j.bspc.2020.102144.
- [35] TensorFlow vs Keras: A Comparison [ออนไลน์]. เข้าถึงได้จาก <https://towardsdatascience.com/tensorflow-vs-keras-d51f2d68fd6c>. (วันที่ค้นข้อมูล : 18 ตุลาคม 2564)
- [36] Unicorn Hybrid Black. [ออนไลน์]. เข้าถึงได้จาก <https://www.unicorn-bi.com/product/unicorn-hybrid-black/>. (วันที่ค้นข้อมูล: 22 มกราคม 2565)
- [37] Lab Streaming Layer (LSL). [ออนไลน์]. เข้าถึงได้จาก <http://openvibe.inria.fr/how-to-use-labstreamingLayer-in-openvibe/>. (วันที่ค้นข้อมูล: 23 มกราคม 2565)
- [38] OpenVIBE Software. [ออนไลน์]. เข้าถึงได้จาก <http://openvibe.inria.fr/discover/>. (วันที่ค้นข้อมูล: 23 มกราคม 2565)
- [39] OpenVIBE and Lab Streaming Layer. [ออนไลน์]. เข้าถึงได้จาก <http://openvibe.inria.fr/how-to-use-labstreamingLayer-in-openvibe/>. (วันที่ค้นข้อมูล: 23 มกราคม 2565)
- [40] CoppeliaSim from the creator of V-REP. [ออนไลน์]. เข้าถึงได้จาก <https://www.coppeliarobotics.com/>. (วันที่สืบค้นข้อมูล: 23 มกราคม 2565)
- [41] Matlab and CoppeliaSim(V-rep). [ออนไลน์] เข้าถึงได้จาก <https://document.wlkata.com/?doc=wlkata-mirobot-resources-for-education/matlab-and-coppelasimv-rep/> (วันที่สืบค้นข้อมูล: 23 มกราคม 2565)
- [42] T. Pachidis *et al.*, "Forward Kinematic Analysis of JACO<sup>2</sup> Robotic Arm Towards Implementing a Grapes Harvesting Robot," *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2020, pp. 1-6, doi: 10.23919/SoftCOM50211.2020.9238297.

## บรรณานุกรม (ต่อ)

- [43] Palacios, Ronald H., “Robotic arm manipulation laboratory with a six degree of freedom JACO arm,” Monterey, California: Naval Postgraduate School, 2015, Dec.
- [44] BCI-based control of a JACO robotic arm with OpenViBE. [ออนไลน์]. เข้าถึงได้จาก <http://openvibe.inria.fr/openvibe/wp-content/uploads/2014/10/OpenViBEWorkshop2014-Bougrain-RoboticArm.pdf> (วันที่ค้นข้อมูล: 19 กุมภาพันธ์ 2565)
- [45] LAI Robotics., “Jaco2 7 DOF with spherical wrist ver 2,” University of Cassino. 2020.
- [46] [EEGLAB. [ออนไลน์]. เข้าถึงได้จาก <https://eeglab.org>. (วันที่ค้นข้อมูล : 10 กันยายน 2564)
- [47] Sebastián-Romagosa, Marc et al., “Brain Computer Interface Treatment for Motor Rehabilitation of Upper Extremity of Stroke Patients-A Feasibility Study,” *Frontiers in neuroscience*, vol. 14, 21 Oct. 2020, doi: 10.3389/fnins.2020.591435.
- [48] Tangermann, Michael et al., “Review of the BCI Competition IV,” *Frontiers in neuroscience*, vol. 6, 13 Jul. 2012, doi: 10.3389/fnins.2012.00055.
- [49] Dong, li & Zhao, Lingling & Zhang, Yufan & Yu, Xue & Li, Fali & Li, Jianfu & Lai, Yongxiu & Liu, Tiejun & Yao, Dezhong., “Reference Electrode Standardization Interpolation Technique (RESIT): A Novel Interpolation Method for Scalp EEG,” *Brain Topography*, vol. 34, pp. 1-12, 2021, doi: 10.1007/s10548-021-00844-2.
- [50] spline. [ออนไลน์]. เข้าถึงได้จาก <https://www.mathworks.com/help/matlab/ref/spline.html>. (วันที่ค้นข้อมูล : 18 ตุลาคม 2564)
- [51] Numerical Interpolation: Natural Cubic Spline. [ออนไลน์]. เข้าถึงได้จาก <https://towardsdatascience.com/numerical-interpolation-natural-cubic-spline-52c1157b98ac>. (วันที่ค้นข้อมูล : 13 ธันวาคม 2564)
- [52] Common Spatial Pattern (CSP). [ออนไลน์]. เข้าถึงได้จาก <https://www.mathworks.com/matlabcentral/fileexchange/72204-common-spatial-patterns-csp>. (วันที่ค้นข้อมูล : 18 ตุลาคม 2564)

## บรรณานุกรม (ต่อ)

- [53] Layer weight regularizers. [ออนไลน์]. เข้าถึงได้จาก  
<https://keras.io/api/layers/regularizers/>. (วันที่ค้นข้อมูล : 22 พฤศจิกายน 2564)
- [54] Lua Stimulator Document. [ออนไลน์]. เข้าถึงได้จาก  
[http://openvibe.inria.fr/documentation/2.0.0/Doc\\_BoxAlgorithm\\_LuaStimulator.html](http://openvibe.inria.fr/documentation/2.0.0/Doc_BoxAlgorithm_LuaStimulator.html). (วันที่ค้นข้อมูล: 19 กุมภาพันธ์ 2565)
- [55] Graz Visualization Document.[ออนไลน์]. เข้าถึงได้จาก  
[http://openvibe.inria.fr/documentation/1.2.1/Doc\\_BoxAlgorithm\\_GrazVisualization.html](http://openvibe.inria.fr/documentation/1.2.1/Doc_BoxAlgorithm_GrazVisualization.html). (วันที่ค้นข้อมูล: 19 กุมภาพันธ์ 2565)
- [56] CSV File Writer Document. [ออนไลน์]. เข้าถึงได้จาก  
[http://openvibe.inria.fr/documentation/2.1.0/Doc\\_BoxAlgorithm\\_CSVFileWriter.html](http://openvibe.inria.fr/documentation/2.1.0/Doc_BoxAlgorithm_CSVFileWriter.html). (วันที่ค้นข้อมูล: 19 กุมภาพันธ์ 2565)
- [57] PLayer Controller Document. [ออนไลน์]. เข้าถึงได้จาก  
[http://openvibe.inria.fr/documentation/1.2.0/Doc\\_BoxAlgorithm\\_PLayerController.html](http://openvibe.inria.fr/documentation/1.2.0/Doc_BoxAlgorithm_PLayerController.html). (วันที่ค้นข้อมูล: 19 กุมภาพันธ์ 2565)
- [58] Tangermann, Michael et al., "Review of the BCI Competition IV.," *Frontiers in neuroscience*, vol. 6, 55., 13 Jul. 2012. doi:10.3389/fnins.2012.00055
- [59] OpenVIBE Stimulation Codes. [ออนไลน์]. เข้าถึงได้จาก  
<http://openvibe.inria.fr/stimulation-codes/>. (วันที่ค้นข้อมูล: 19 กุมภาพันธ์ 2565)
- [60] G. S. Sagee and S. Hema. "EEG feature extraction and classification in multiclass multiuser motor imagery brain computer interface using Bayesian Network and ANN," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), pp. 938-943, 2017, doi: 10.1109/ICICT1.2017.83426