

การเปรียบเทียบความสามารถในการควบคุมกระบวนการที่มีสัญญาณเข้า  
ออกเชิงเดี่ยวและไม่เป็นเชิงเส้นระหว่าง  
ระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงและระบบควบคุมแบบอัจฉริยะ

A COMPARISON OF THE CONTROLLABILITY OF MODEL REFERENCE  
ADAPTIVE CONTROL AND INTELLIGENT CONTROL SYSTEMS FOR A  
NONLINEAR SISO PLANT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัจฉริยะเชิงคำนวณ  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ.2566

KMITL-2024-EN-M-407-235

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A COMPARISON OF THE CONTROLLABILITY OF MODEL REFERENCE  
ADAPTIVE CONTROL AND INTELLIGENT CONTROL SYSTEMS FOR A  
NONLINEAR SISO PLANT



A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF ENGINEERING IN ROBOTICS AND COMPUTATIONAL  
INTELLIGENCE SYSTEMS  
SCHOOL OF ENGINEERING  
KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
2024  
KMITL-2024-EN-M-407-235

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2024

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|                   |  |
|-------------------|--|
| หัวข้อวิทยานิพนธ์ | การเปรียบเทียบความสามารถในการควบคุมกระบวนการที่มีสัญญาณเข้าออกเชิงเดี่ยวและไม่เป็นเชิงเส้นระหว่างระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงและระบบควบคุมแบบอัจฉริยะ |
| ชื่อนักศึกษา      | วรวัฒน์ ดวนใหญ่  |
| รหัสประจำตัว      | 65016082   |
| ปริญญา            | วิศวกรรมศาสตรมหาบัณฑิต   |
| สาขาวิชา          | วิศวกรรมหุ่นยนต์และระบบอัจฉริยะเชิงคำนวณ   |
| พ.ศ.              | 2566   |
| อาจารย์ที่ปรึกษา  | Asst. Prof. Dr. Weon Keun Song   |

## บทคัดย่อ

วิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อแก้ไขปัญหาความท้าทายในการออกแบบระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง (Model Reference Adaptive Control: MRAC) โดยเฉพาะอย่างยิ่งในการค้นหาฟังก์ชัน Lyapunov ลำดับสูง ซึ่งเป็นอุปสรรคสำคัญในการสร้างระบบควบคุมสำหรับกระบวนการที่ไม่เป็นเชิงเส้น การศึกษานี้นำเสนอการเปรียบเทียบเชิงประสิทธิผลของการควบคุมกระบวนการที่มีสัญญาณเข้าออกเชิงเดี่ยวและไม่เป็นเชิงเส้นระหว่าง MRAC แบบดั้งเดิมที่มีพื้นฐานตามเสถียรภาพของ Lyapunov กับระบบควบคุมแบบอัจฉริยะ (Intelligent Control System: ICS) ที่พัฒนาขึ้นใหม่โดยใช้เครือข่ายประสาทเทียม ทั้งในตัวควบคุมที่สามารถปรับแต่งตัวเองได้และตัวระบุคุณลักษณะกระบวนการ โดยระบบควบคุมทั้งสองระบบนั้นอ้างอิงตามหลักการการควบคุมแบบปรับตัว (Adaptive Control) ที่สามารถปรับแต่งพารามิเตอร์ภายในตอบสนองตามสัญญาณควบคุม

การศึกษานี้ใช้การปรับแต่งของเครือข่ายประสาทเทียมในทุกช่วงเวลาการควบคุมสำหรับตัวควบคุมที่สามารถปรับแต่งตัวเองได้ (Self-tuning Controller) โดยไม่เพียงแต่เป็นการปรับแต่งพารามิเตอร์ปกติของเครือข่ายประสาทแต่ยังรวมถึงการปรับแต่งไฮเปอร์พารามิเตอร์ผ่านการปรับแต่งแบบเบย์เซียน (Bayesian Optimization) อีกทั้งในส่วนของตัวระบุคุณลักษณะกระบวนการได้ถูกออกแบบโดยใช้เครือข่ายประสาทเทียมที่ระบุข้อมูลเชิงกายภาพที่มีการเรียนรู้แบบเมตา (Meta-Learning-based Physics-Informed Neural Network: MLPINN) สำหรับใช้ในการระบุคุณลักษณะแบบออนไลน์โดยไม่ต้องมีการระบุข้อมูล รวมเข้ากับกลไกการเรียนรู้แบบเมตาเชิงเหตุการณ์ (Meta-learning-triggered mechanism) เพื่อควบคุมจัดการทรัพยากรในการประมวลผลให้มีประสิทธิภาพ

นอกจากนี้ในการออกแบบ ICS วิธีการการเรียนรู้แบบถ่ายทอด (Transfer Learning) ได้ถูกนำมาใช้ทั้งในตัวควบคุมและใน MLPINN เพื่อให้ทั้งสองสามารถปรับตัวให้เข้ากับข้อมูลใหม่ได้อย่างรวดเร็วยิ่งขึ้น

งานวิจัยนี้ได้แสดงให้เห็นว่า ICS ไม่เพียงแต่มีประสิทธิภาพที่เหนือกว่า MRAC แต่ยังมีกรออกแบบที่เรียบง่ายกว่า อีกทั้งยังสามารถรวมเอาเทคนิคทั้งหมดเข้าด้วยกันและทำงานร่วมกันโดยไม่ต้องอาศัยความรู้ทางคณิตศาสตร์ที่ซับซ้อน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|                       |  |
|-----------------------|--|
| <b>Thesis</b>         | A Comparison of the Controllability of Model Reference Adaptive Control and Intelligent Control Systems for a Nonlinear SISO Plant |
| <b>Student</b>        | Mr. Worrawat Duanyai   |
| <b>Student ID.</b>    | 65016082   |
| <b>Degree</b>         | Master of Engineering  |
| <b>Program</b>        | Robotic and Computational Intelligence Systems Engineering   |
| <b>Year</b>           | 2024   |
| <b>Thesis Advisor</b> | Asst. Prof. Dr. Weon Keun Song   |

## ABSTRACT

The study aims to address challenges in the design of the MRAC system, especially such as the discovery of a high-order Lyapunov function candidate, becoming an obstacle in building nonlinear control schemes. It presents a comparison of control performance between a conventional model reference adaptive control (MRAC) system based on Lyapunov stability and a newly developed intelligent control system (ICS) with a neural network-based self-tuning controller and a plant identifier for a single-input single-output nonlinear plant. Both control systems are based on adaptive control principles, which adjust internal parameters in response to control signals.

This study utilizes a neural network updated at each time step, optimizing not only model parameters but also hyperparameters through Bayesian optimization for the self-tuning controller. The plant identifier is devised using a meta-learning-based physics-informed neural network (MLPINN) for online identification without labeled data and a meta-learning-triggered mechanism to manage computational resources efficiently. In the design of the ICS, furthermore, transfer learning is applied to both the controller and the MLPINN for quick adaptation to new inputs.

This research demonstrates that the ICS not only outperforms the MRAC but is also simpler in design, incorporating all the techniques without requiring extensive mathematical knowledge.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา Asst. Prof. Dr Weon Keun Song ที่ให้ความช่วยเหลือ ให้คำชี้แนะตลอดจนให้ความรู้และประกาณที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณผู้พัฒนาไลบรารี Tensorflow และ Keras และผู้เกี่ยวเนื่องทุกท่านที่ได้พัฒนาไลบรารีที่สามารถก่อประโยชน์และสามารถทำให้ผู้วิจัยได้ดำเนินการวิจัยต่อได้อย่างราบรื่น

ขอขอบพระคุณ ดร.วีระพล โมนยะกุล และ รศ.ดร.วรรณดี เพชรมณีล้ำค่า ที่ได้ให้เกียรติเป็นคณะกรรมการสอบหัวข้อและโครงร่างวิทยานิพนธ์ ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี

ขอขอบพระคุณอาจารย์ ศ.ดร.อุมา สิบญูเรื่อง ที่ให้ความอนุเคราะห์ในการดำเนินการอนุมัติวารสารสำหรับวิทยานิพนธ์

ขอขอบพระคุณเจ้าหน้าที่สำนักทะเบียน เจ้าหน้าที่บัณฑิตศึกษาคณะวิศวกรรมศาสตร์ รวมไปถึงเจ้าหน้าที่ทุกท่านที่เกี่ยวข้องที่มีความอนุเคราะห์ในการดำเนินการต่างๆ ทำให้การสอบวิทยานิพนธ์เป็นไปได้อย่างราบรื่น

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่านทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

วรวัฒน์ ดวนใหญ่

# สารบัญ

|  | หน้า |
|--|------|
| บทคัดย่อ.....  | I    |
| ABSTRACT.....  | III  |
| กิตติกรรมประกาศ.....   | V    |
| สารบัญ.....  | VI   |
| สารบัญรูป.....   | VIII |
| สารบัญตาราง.....   | IX   |
| บทที่1 บทนำ.....   | 1    |
| 1.1 ความเป็นมาและความสำคัญ.....  | 1    |
| 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....  | 1    |
| 1.3 สมมติฐานของการศึกษา.....   | 2    |
| 1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....   | 3    |
| 1.5 ขอบเขตการวิจัย.....  | 4    |
| 1.6 ขั้นตอนของการศึกษา.....  | 4    |
| 1.7 ข้อจำกัดของการศึกษา.....   | 5    |
| 1.8 คำจำกัดความที่ใช้ในการศึกษา.....   | 5    |
| บทที่2 งานวิจัยที่เกี่ยวข้อง.....  | 8    |
| 2.1 Lyapunov-based MRAC การวิเคราะห์เสถียรภาพของกระบวนการที่มีสัญญาณเข้าและ<br>ออกเชิงเดียว..... | 8    |
| 2.2 เครือข่ายประสาทเทียม (Artificial Neural Network).....  | 13   |
| 2.3 MLPINN.....  | 18   |
| 2.4 Hyper Parameter Optimization ด้วยวิธีการปรับแต่งแบบ Bayesian.....                            | 22   |
| บทที่3 ความสำคัญของการออกแบบระบบควบคุมแบบอัจฉริยะ.....   | 30   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

|  | หน้า |
|--|------|
| 3.1 ปัญหาของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง .....  | 30   |
| 3.2 การสร้าง MRAC-RK6 เพื่อใช้ในการเปรียบเทียบ .....   | 31   |
| 3.3 ความสำคัญของ ICS.....  | 32   |
| บทที่4 การออกแบบระบบควบคุมแบบอัจฉริยะโดยใช้ MLPINN ร่วมกับกลไกการเรียนรู้แบบเมต้าเชิง<br>เหตุการณ์ และตัวควบคุมเครือข่ายประสาทที่สามารถปรับปรุงตัวเองได้ ..... | 33   |
| 4.1 การออกแบบแผนภาพการควบคุมของ ICS.....   | 33   |
| 4.2 การสร้าง MLPINN เพื่อใช้ในการหาคุนสมบัติของกระบวนการ .....   | 35   |
| 4.3 การสร้าง Intelligent Controller.....   | 40   |
| 4.4 การปรับปรุง Intelligent Controller ด้วย HPO .....  | 43   |
| 4.5 ผลการทดลอง .....   | 46   |
| บทที่5 สรุปผลการวิจัยและข้อเสนอแนะ .....   | 49   |
| 5.1 สรุปผลการวิจัย .....   | 49   |
| 5.2 ข้อเสนอแนะ .....   | 51   |
| บรรณานุกรม.....  | 52   |
| ภาคผนวก.....   | 54   |
| ประวัติผู้เขียน .....  | 55   |

# สารบัญรูป

| รูปที่  | หน้า |
|---|------|
| 2.1 แผนภาพรูปแบบการควบคุมของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง.....                    | 9    |
| 2.2 ไดอะแกรมการทำงานของเซลล์ประสาทเทียม.....  | 14   |
| 2.3 Activation Function.....  | 15   |
| 2.4 ประเภทของ Neural Network.....   | 16   |
| 2.5 แผนภาพกลไกการทำงานของ MLPINN (Duanyai, 2024).....                                     | 21   |
| 2.6 การส่งต่อการเรียนรู้ของตัวเรียนรู้ภายใน (Duanyai, 2024).....                          | 21   |
| 2.7 การเปรียบเทียบระหว่าง PINN ชนิด IVP และ MLPINN ของการจำลองการสั่นพ้องแบบ Duffing..... | 22   |
| 2.8 Gaussian Process.....   | 25   |
| 2.9 ความสัมพันธ์ระหว่าง Gaussian Process และ Expected Improvement.....                    | 27   |
| 3.1 ผลตอบสนองของ RK6-MRAC .....   | 31   |
| 4.1 แผนภาพการควบคุมของระบบควบคุมแบบอัจฉริยะ.....  | 34   |
| 4.2 Physical Inform Neural Network รูปแบบเรียนรู้.....                                    | 37   |
| 4.3 การทำงานกลไกการเรียนรู้แบบเมตาเชิงเหตุการณ์ใน MLPINN (Duanyai et al., 2024).....      | 40   |
| 4.4 สถาปัตยกรรมของตัวควบคุมอัจฉริยะ.....  | 41   |
| 4.5 สัญญาณอ้างอิงที่ใช้ในการเปรียบเทียบ .....   | 42   |
| 4.6 ผลลัพธ์การควบคุมของระบบควบคุมแบบอัจฉริยะ.....   | 43   |
| 4.7 กระบวนการ HPO สำหรับตัวควบคุมอัจฉริยะ.....  | 44   |
| 4.8 ผลลัพธ์การควบคุมของระบบควบคุมแบบอัจฉริยะที่มีการปรับปรุงไฮเพอร์พารามิเตอร์.....       | 46   |
| 4.9 เปรียบเทียบผลตอบสนองระหว่าง MRAC และ Self-Tuning ICS.....                             | 47   |
| 4.10 เปรียบเทียบความคลาดเคลื่อนระหว่าง MRAC และ Self-Tuning ICS.....                      | 47   |

# สารบัญตาราง

|   |      |
|---|------|
| ตารางที่  | หน้า |
| 4.1 ผลการเปรียบเทียบระหว่าง MRAC และ ICS โดยใช้ L2 Normalization..... | 48   |



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญ

ในการออกแบบระบบควบคุมมีการศึกษาเป็นจำนวนมาก และใช้หลากหลายเทคนิควิธีการเช่น ระบบควบคุมแบบคงทน (Robust Control) ระบบควบคุมแบบความไวได้ และระบบควบคุมแบบปรับตัวได้ (Adaptive Control) ซึ่งทฤษฎีการควบคุมแบบปรับตัวนั้นมีข้อแตกต่างเป็นอย่างมากจากวิธีการควบคุมแบบอื่นๆ ในการที่พารามิเตอร์ของตัวควบคุมนั้นเป็นตัวแปรที่สามารถปรับตัวได้ และมีอัลกอริทึมที่สามารถปรับปรุงพารามิเตอร์ของตัวเองได้โดยอ้างอิงจากสัญญาณเข้าและออกของกระบวนการแบบปิดได้

ระบบที่มีการเปลี่ยนแปลงแบบไดนามิกต่างๆนั้นจะมีลักษณะของสัญญาณและคุณสมบัติของกระบวนการที่ไม่เป็นเชิงเส้น หากแต่ว่าการวิจัยการควบคุมแบบปรับตัวนั้นกลับมุ่งเน้นไปที่การวิจัยของกระบวนการที่เป็นเชิงเส้น (หรือถูกทำให้เป็นเชิงเส้นในช่วงใดช่วงหนึ่ง) ซึ่งเป็นเหตุผลสืบเนื่องมาจากความซับซ้อนของทฤษฎีการควบคุมกระบวนการที่ไม่เป็นเชิงเส้น อย่างไรก็ตามในความเป็นจริงนั้นกระบวนการที่ใช้งานอย่างแพร่หลายนั้นเมื่อพิจารณาในภาพที่กว้างขึ้นจะมีลักษณะที่ไม่เป็นเชิงเส้น ซึ่งเป็นสิ่งที่น่ากังวลในด้านเสถียรภาพของระบบ หากพิจารณาถึงความยากง่ายในการควบคุมนั้นระบบควบคุมเชิงเส้นที่นำไปควบคุมระบบที่ไม่เป็นเชิงเส้นในช่วงใดช่วงหนึ่งนั้นสามารถพัฒนาได้โดยง่าย แต่หากต้องการควบคุมระบบที่ไม่เป็นเชิงเส้นโดยสมบูรณ์และมีเสถียรภาพจำเป็นที่จะต้องใช้วิธีการที่พิเศษกว่า ซึ่งระบบควบคุมแบบปรับตัวชนิดไม่เป็นเชิงเส้นนั้นถูกนำไปใช้ในแอปพลิเคชันที่หลากหลายเช่น วิศวกรรมหุ่นยนต์ วิศวกรรมชีวเคมี Hydraulic servomechanism การสื่อสารในระบบโทรคมนาคม ระบบควบคุมปีกเครื่องบินในสถานะที่ของไหลไม่สามารถบีบตัวได้ หุ่นยนต์ปฏิบัติการในอวกาศ และเตาปฏิกรณ์ชีวภาพ ซึ่งในการวิจัยนี้จะให้ความสำคัญไปที่ระบบควบคุมแบบปรับตัวสำหรับกระบวนการที่ไม่เป็นเชิงเส้นและมีสัญญาณเข้าและออกเชิงเดียว

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

Model Reference Adaptive Control (MRAC) นั้นมีความโดดเด่นในด้านของความแข็งแกร่งของระบบ (Robustness) และสามารถนำไปใช้กับเป้าหมายหลากหลายขนาด (Scalability) จึงได้รับความนิยมน้อยแพร่หลายในงานวิจัยที่เกี่ยวข้อง (Avila et al., 2013) หลักการของ MRAC ถูกนำเสนอเป็นครั้งแรกโดย Whitaker (Shekhar & Sharma, 2018) ในปีค.ศ. 1950 ซึ่งต่อมาในปี ค.ศ. 1970 มีนักวิทยาศาสตร์ได้นำทฤษฎีการวิเคราะห์ความเสถียรของ Aleksandr Lyapunov นักคณิตศาสตร์และ

ฟิลิปปินส์ชาวรัสเซีย นำมาผนวกรวมเข้ากับการออกแบบ MRAC นับแต่นั้นมา MRAC บนพื้นฐานของ Lyapunov (Lyapunov-based MRAC) ก็ได้เป็นหนึ่งในระบบควบคุมที่ได้รับความนิยมเป็นอย่างมาก ดังที่ได้กล่าวไป ในขณะเดียวกันตั้งแต่ปีค.ศ. 1990 เป็นต้นมาจนถึงปัจจุบันก็ได้มีการนำเสนอแนวทางใหม่ที่นำเครือข่ายประสาทเทียม (Neural Network) มาประยุกต์ใช้ในการออกแบบระบบควบคุมด้วย เช่นเดียวกัน โดยระบบควบคุมที่ใช้เครือข่ายประสาทรุ่นนั้นสามารถแบ่งจำแนกได้ออกเป็น 7 ชนิด

1. NN-based adaptive linear control (Hunt et al., 1992)
2. NN-based direct inverse control (Calvo-Rolle et al., 2014; Hunt et al., 1992)
3. NN-based internal model control (Hunt et al., 1992; Nahas et al., 1992)
4. NN-based predictive control (Hunt et al., 1992)
5. NN-based model reference control (NNMRC) (Duanyai et al., 2024; Narendra, 1990; Hunt et al., 1992)
6. NN-based adaptive critic control (Balakrishnan & Biega, 1996)
7. Reinforcement learning-based control (Hunt et al., 1992)

อย่างไรก็ตามการออกแบบระบบควบคุมทั้ง Lyapunov-based MRAC และแบบที่ใช้เครือข่ายประสาทเทียม ต่างก็เป็นเป็นสาขาวิชาที่มีการวิจัยมาอย่างยาวนาน แต่ทว่ายังคงมีความท้าทายและข้อจำกัดอยู่บางประการอันสืบเนื่องมาจากปัจจัยบางประการเช่น การรบกวน ความไม่แน่นอนของรูปแบบที่อ้างอิง ความไม่แน่นอนของตัวแปรพารามิเตอร์ต่างๆ ความละเอียดของเวลาในการคำนวณ ความล่าช้า และความซับซ้อนของการออกแบบระบบ ดังนั้นจึงได้มีการศึกษาระบบควบคุมแบบอัจฉริยะขึ้นที่เป็นการนำ Neural Network มาใช้ทั้งภาคควบคุม และภาคของการหาคุณสมบัติของกระบวนการ ดังจะได้อธิบายโดยละเอียดในบทที่ 3

### 1.3 สมมติฐานของการศึกษา

ในงานวิจัยฉบับนี้ผู้วิจัยคาดหวังว่า Intelligent Control System ที่พัฒนาขึ้นจะมีความสามารถที่สามารถควบคุมกระบวนการที่ไม่เป็นเชิงเส้นได้อย่างมีประสิทธิภาพกว่า MRAC อันเป็นเหตุผลสืบเนื่องมาจากภาคการควบคุมนั้นใช้ Neural Network ที่มีความสามารถในการเรียนรู้เพื่อปรับปรุงพารามิเตอร์ของตนเอง ซึ่งเป็นลักษณะที่คล้ายคลึงกับ Lyapunov-based MRAC แต่ Neural Network นั้นมีจำนวนพารามิเตอร์ และความยืดหยุ่นในการออกแบบที่สูงกว่าอย่างยิ่งยวดขึ้นอยู่กัสภาพปฏิกิริยาที่สร้าง ดังนั้นจึงมีแนวโน้มที่จะสามารถเรียนรู้และตอบสนองได้ดีกว่า ยิ่งไปกว่านั้นวิทยานิพนธ์ฉบับนี้ยังแสดงให้เห็นถึงการพัฒนาประสิทธิภาพของการควบคุมอัจฉริยะอย่างยิ่งยวด โดยใช้ตั้งแต่การออกแบบแผนผังการควบคุม อัลกอริทึมการปรับพารามิเตอร์แบบออนไลน์ (Zayegh & Al Bassam, 2018; Abdel-Nasser, 2020) กระบวนการ Hyper Parameter Optimization (Yang & Shami, 2020;

Frazier, 2018; Rasmussen & Williams, 2006) ด้วย Bayesian Optimization และ อัลกอริทึมในการเรียนรู้แบบถ่ายโอน อีกทั้งระบบควบคุมแบบอัจฉริยะในภาคของการหาค่าคุณสมบัติของกระบวนการนั้นได้มีการนำ Neural Network แบบพิเศษที่เรียกว่า Meta-learning-based physics-informed Neural Network (MLPINN) (Duanyai, 2024; Finn et al., 2017) มาใช้งาน ซึ่งมีความสามารถในการวิเคราะห์สมการเชิงอนุพันธ์ได้เป็นอย่างดีและมีความถูกต้องมากกว่าวิธีทั่วไป ด้วยเหตุผลดังที่ได้กล่าวมาทั้งหมด Intelligent Control System จึงมีแนวโน้มที่จะมีประสิทธิภาพที่สูงกว่า Lyapunov-based MRAC

#### 1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

ในงานวิจัยฉบับนี้ได้มีการนำหลายทฤษฎีมาพิจารณา ในด้านของ MRAC มีงานวิจัยที่ได้รับการตีพิมพ์และรองรับอยู่มากมายตั้งแต่อดีตจนถึงปัจจุบัน ทั้งแบบดั้งเดิมไปจนถึงแบบที่มีการนำ Neural Network MRAC (NNMRC) (Duanyai et al., 2024) มาใช้ ซึ่งในงานวิจัยฉบับนี้จะนำ Lyapunov-based MRAC ที่เป็นชนิดย่อยของ MRAC มาใช้ในการเปรียบเทียบ

ในด้านของ Intelligent Control System มีการนำหลากหลายทฤษฎีมาพัฒนา ทั้งความสามารถในการปรับปรุง Hyperparameter ของตัวควบคุมด้วยวิธีการ Bayesian Optimization ความสามารถในการปรับตัวแบบออนไลน์ และความสามารถในการหาค่าคุณสมบัติของกระบวนการด้วย MLPINN ที่มีรากฐานการวิจัยมาจาก Physical Informed Neural Network (PINN)

(Raissi et al., 2019) ได้ริเริ่มนำเสนอการในการนำ PINN ที่เป็นการใช้ Neural Network โดยมีการระบุข้อมูลเพียงเล็กน้อย มาใช้ในการคำนวณหาสมการเชิงอนุพันธ์แบบแยกส่วน (Partial Differential Equation: PDE) ความสำเร็จของ Raissi ได้ถูกนำไปใช้ในหลากหลายลักษณะงาน เช่น กลศาสตร์ของไหล แมคคานิคควอนตัม ปฏิกิริยาการแพร่ (Reaction-Diffusion) และการเคลื่อนที่ของคลื่นน้ำ ทำให้นักวิจัยหลายท่านได้เริ่มทำการวิจัย PINN และนำไปสู่การวิจัยสาขาอื่นๆ

ต่อมาเทคนิค Meta-Learning ได้ถูกนำเสนอโดย (Peng et al., 2002) ซึ่งเป็นการเปลี่ยนแปลงการออกแบบอัลกอริทึมการเรียนรู้ของเครือข่ายประสาท โดยโมเดลที่มีการใช้งาน Meta-Learning ที่ถูกเรียกว่า Meta-model นั้นมีหลักการจากการหาเกรเดียนในรูปแบบของ Model-Agnostic Meta-Learning (MAML) ที่มีการปรับแต่งสองระดับ (bi-level optimization) (Finn et al., 2017) ซึ่งมีตัวเรียนรู้ภายในหลายตัวอยู่ในระดับแรกเพื่อทำการปรับพารามิเตอร์อย่างละเอียดของโมเดลสนับสนุน และมีตัวเรียนรู้ภายนอกเพียงตัวเดียวในระดับที่สองเพื่อทำการเรียนรู้จากผลลัพธ์ในระดับที่หนึ่ง อย่างไรก็ตามมีหลากหลายความคิดในการออกแบบอัลกอริทึมของ Meta-learning ซึ่งต่างก็เป็นการนำไปใช้เพื่อปรับปรุงประสิทธิภาพของ PINN ในงานวิจัยฉบับนี้ได้มีการนำแนวความคิด MAML มาประกอบเข้ากับ PINN เรียกว่า MLPINN มาใช้เพื่อหาค่าคุณสมบัติของกระบวนการโดยปราศจากการระบุข้อมูลให้กับชุดข้อมูลเรียนรู้ (Duanyai et al., 2024; Duanyai, 2024)

ความสามารถในการปรับแต่งไฮเปอร์พารามิเตอร์ของตัวควบคุมของตัวเองใน Intelligent Control System นั้นเป็นความสามารถที่มีความสำคัญอย่างยิ่ง ซึ่งก่อให้เกิดประสิทธิภาพของตัวควบคุมนั้นเพิ่มขึ้นเป็นอย่างมาก เนื่องจากปกติทั่วไปแล้วการปรับแต่ง Hyperparameter จำเป็นที่จะต้องใช้ประสบการณ์และความเชี่ยวชาญของผู้วิจัยในการปรับแต่ง เป็นกระบวนการที่ต้องใช้เวลาเป็นอย่างมาก อย่างไรก็ตามได้มีวิธีการมากมายในการปรับแต่งหนึ่งในนั้นก็คือวิธีการปรับแต่งที่เรียกว่า Bayesian Optimization โดยใช้กระบวนการเกาส์เซียนในการหาการฟังก์ชันเป้าหมาย และทำการสุ่มตัวอย่างไฮเปอร์พารามิเตอร์ขึ้นมาในกระบวนการ หาผลลัพธ์จากฟังก์ชันเป้าหมาย ทำการอัปเดตข้อมูลและคาดคะเนจุดตัวอย่างต่อไปจากผลลัพธ์ที่ได้โดยใช้ทฤษฎีความน่าจะเป็น Bayesian ด้วยวิธีการนี้ในท้ายที่สุดนั้นจะได้ผลลัพธ์ของตัวอย่างที่ได้ผลสำเร็จที่เข้าสู่เป้าหมายได้มากที่สุด (Yang & Shami, 2020; Rasmussen & Williams, 2006)

## 1.5 ขอบเขตการวิจัย

ในงานวิจัยฉบับนี้จะเป็นการวิจัยข้อเปรียบเทียบระหว่าง Lyapunov-based MRAC และ Intelligent Control System (ICS) ดังนั้นงานวิจัยจึงแบ่งออกเป็น 2 ส่วน คือ MRAC และ ICS คือบทที่ 3 และ บทที่ 4 โดยนำไปควบคุมกระบวนการที่ไม่เป็นเชิงเส้น ซึ่งในงานทดลองกระบวนการที่ใช้ในการเปรียบเทียบจะมีลักษณะเพิ่มขึ้นแบบเลขยกกำลัง (Exponential) ลักษณะนี้เป็นลักษณะที่ไม่เสถียรอย่างยิ่งเมื่อเวลาผ่านไป และมีรูปแบบเป้าหมายอ้างอิงแบบคลื่นไซน์ที่มีการเปลี่ยนแปลงขึ้นกับช่วงคาบเวลา ด้วยเงื่อนไขทั้งสองนี้จะเป็นการจำลองกระบวนการที่ไม่เป็นเชิงเส้นเพื่อใช้ในการควบคุม

ในงานวิจัยนี้เป็นงานวิจัยเพื่อเปรียบเทียบความสามารถในการควบคุมแต่ทว่า ICS อยู่ในขั้นตอนของการวิจัย งานวิจัยนี้จึงไม่รวมไปถึงการปรับแต่งการใช้ทรัพยากรการประมวลผลและการพิจารณาการนำไปใช้งานแบบเวลาจริง

## 1.6 ขั้นตอนของการศึกษา

งานวิจัยฉบับนี้จะแบ่งออกเป็น 9 ขั้นตอน

1. ตั้งปัญหาที่เกิดขึ้น
2. ทำการวิจัยหาสาเหตุและทฤษฎีที่เกี่ยวข้องเพื่อใช้เป็นหลักการในการวิจัย
3. เริ่มทำการหาเงื่อนไขที่เหมาะสมเพื่อใช้ในการเปรียบเทียบ
4. สร้างและทดสอบตัวควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงบนพื้นฐานของ Lyapunov โดยใช้วิธีการ Runge-Kutta ลำดับที่ 6 ในการหาคุณสมบัติของกระบวนการ เพื่อใช้ในการเปรียบเทียบ
5. สร้างและทดสอบตัวอย่างทดสอบ MLPINN เพื่อพิสูจน์ทฤษฎี

6. สร้างและทดสอบตัวควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงบนพื้นฐานของ Lyapunov โดยใช้ MLPINN ในการหาคุณสมบัติของกระบวนการ เพื่อพิสูจน์ทฤษฎี
7. สร้างและทดสอบตัวควบคุมอัจฉริยะ โดยใช้ MLPINN ในการหาคุณสมบัติของกระบวนการ เรียกว่า Intelligent Control System เพื่อใช้ในการเปรียบเทียบ
8. สร้างและทดสอบตัวควบคุมอัจฉริยะที่สามารถปรับแต่งตัวเองได้ โดยใช้ MLPINN ในการหาคุณสมบัติของกระบวนการ เรียกว่า Self-Tuning Intelligent Control System เพื่อใช้ในการเปรียบเทียบ
9. สรุปผลและอภิปรายผลการเปรียบเทียบ

### 1.7 ข้อจำกัดของการศึกษา

ในการทดลองนั้นมีข้อจำกัดในการวิจัยอยู่สองประการ ทำให้ต้องลดขอบเขตในการวิจัยลง ซึ่งปัจจัยดังกล่าวมีดังต่อไปนี้

1. ระบบควบคุมแบบอัจฉริยะใช้ทรัพยากรการประมวลผลที่สูงมาก ด้วยข้อจำกัดทางด้านทรัพยากร ทำให้ใช้ระยะเวลาในการประมวลผลเป็นระยะเวลาที่ยาวนาน
2. ด้วยข้อจำกัดของเทคโนโลยีในปัจจุบัน Library software ของตัวปรับแต่งนั้นมีปัญหาทำให้หน่วยความจำเข้าถึงแบบสุ่มโดยไม่คืนให้กับระบบ ทำให้หน่วยความจำลดน้อยลงอย่างรวดเร็ว ทำให้ ICS ไม่สามารถที่จะใช้งานเป็นระยะเวลานานได้จึงจำเป็นต้องจำกัดช่วงเวลาที่น่าสนใจของการวิจัย

### 1.8 คำจำกัดความที่ใช้ในการศึกษา

**ระบบควบคุมแบบอัจฉริยะ (Intelligent Control System : ICS)** หมายถึงระบบควบคุมที่ใช้เครือข่ายประสาทเป็นตัวควบคุมตามรูปแบบอ้างอิง และใช้เครือข่ายประสาทเทียมในการหาคุณสมบัติของกระบวนการ

**ระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง (Model Reference Adaptive Control)** หมายถึงระบบควบคุมที่มีความสามารถปรับตัวได้โดยใช้รูปแบบอ้างอิง ใช้คณิตศาสตร์ขั้นสูงเป็นการคำนวณในการปรับตัวและหาคุณสมบัติของกระบวนการ

**ไลบรารี (Library)** หมายถึง ตัวซอฟต์แวร์ที่ผู้อื่นได้พัฒนาขึ้นมาให้ผู้อื่นได้พัฒนาวิจัยต่อเนื่องจากที่ได้พัฒนาเอาไว้แล้ว

**ระบบควบคุมแบบคงทน (Robust Control)** หมายถึงระบบควบคุมที่คงที่ใช้พารามิเตอร์ที่กำหนดเอาไว้ในการควบคุมกระบวนการ พารามิเตอร์ไม่สามารถปรับตัวได้ เช่น PID Controller เป็นต้น

**ระบบควบคุมแบบปรับตัว (Adaptive Control)** หมายถึงระบบควบคุมที่พารามิเตอร์ในการควบคุมเป็นตัวแปรที่สามารถปรับได้ จึงมีความยืดหยุ่นและซับซ้อนกว่าระบบควบคุมแบบคงทน

**เครือข่ายประสาท (Neural Network)** หมายถึงเครือข่ายประสาทเทียมที่มีความสามารถในการเรียนรู้ และคาดเดา โดยใช้วิธีการทางคณิตศาสตร์ มีสถาปัตยกรรมที่แตกต่างกันออกไปขึ้นอยู่กับลักษณะของงานที่นำไปใช้ และผู้พัฒนา

**พารามิเตอร์ (Parameter)** หมายถึงค่าที่ใช้ในการกำหนดคุณสมบัติภายในของสิ่งต่างๆ เป็นชุดข้อมูลที่ใช้ในการคำนวณจากสัญญาณเข้าสู่สัญญาณออก ในกรณีของเครือข่ายประสาทเทียมหมายถึง Weight ในการกำหนดน้ำหนักเพื่อกำหนดผลลัพธ์ของการคาดการณ์

**ไฮเปอร์พารามิเตอร์ (Hyperparameter)** หมายถึงพารามิเตอร์การตั้งค่าของการปรับแต่งเครือข่ายประสาท ลักษณะและประสิทธิภาพของการเรียนรู้ของเครือข่ายประสาทขึ้นอยู่กับไฮเปอร์พารามิเตอร์โดยตรง โดยทั่วไปพารามิเตอร์นี้ผู้พัฒนาจะเป็นผู้กำหนดโดยอาศัยประสบการณ์และความเชี่ยวชาญในการกำหนด

**เกรเดียน (Gradient)** หมายถึง การเปลี่ยนแปลงการสูญเสียเทียบกับพารามิเตอร์ หรือการเปลี่ยนแปลงเชิงอนุพันธ์ใดๆ

**การเรียนรู้ (Learning)** หมายถึงกระบวนการในการเรียนรู้ของเครือข่ายประสาท

**การคาดการณ์ (Prediction)** หมายถึงกระบวนการในการคาดคะเนผลลัพธ์จากข้อมูลที่ได้รับมาของเครือข่ายประสาท

**Forward Propagation** หมายถึงกระบวนการในการเรียนรู้ที่เครือข่ายประสาทนำข้อมูลหรือชุดของข้อมูลประมวลผลร่วมกับพารามิเตอร์ ลักษณะของการประมวลผลเป็นไปตามสถาปัตยกรรมของเครือข่ายประสาท และได้ผลลัพธ์ออกมาในท้ายที่สุด

**Back Propagation** หมายถึงกระบวนการในการเรียนรู้ที่เครือข่ายประสาทนำผลลัพธ์ที่ได้จากกระบวนการ Forward Propagation แล้วนำมาคำนวณเปรียบเทียบกับผลลัพธ์ที่ถูกต้องใน Cost function เพื่อหาค่าการสูญเสีย แล้วนำไปคำนวณหาเกรเดียนเพื่อนำไปอัปเดตพารามิเตอร์ตามไฮเปอร์พารามิเตอร์ที่ตั้งค่าไว้ของเครือข่ายประสาท

**Cost function** หมายถึงฟังก์ชันทางคณิตศาสตร์ใช้ในการคำนวณเปรียบเทียบระหว่างผลลัพธ์ที่ได้จาก Forward Propagation กับผลลัพธ์ที่ถูกต้องเพื่อหาค่าการสูญเสียหรือค่าความแตกต่างจากกระบวนการ Cost Function อาจสามารถเรียกอีกอย่างว่า Loss Function

**ค่าการสูญเสีย (Loss)** หมายถึงตัวเลขที่แสดงถึงความแตกต่างระหว่างตัวแปรสองตัวแปรที่ได้จาก Cost function โดยทั่วไปจะมีค่าเป็นบวกเสมอ

**เครือข่ายประสาทชนิดระบุข้อมูลกายภาพ (Physical-Informed Neural Network : PINN)** หมายถึงเครือข่ายประสาทที่ใช้การระบุข้อมูลทางฟิสิกส์ หรือข้อมูลทางกายภาพ โดยมีการระบุข้อมูล เพื่อใช้ในการหาคุณลักษณะทางฟิสิกส์ โดยทั่วไปจะใช้ในการคำนวณหาผลลัพธ์ของสมการเชิงอนุพันธ์ทั้งแบบแยกส่วนและแบบสามัญ

**เครือข่ายประสาทชนิดระบุข้อมูลกายภาพที่มีการเรียนรู้แบบเมตา (Meta-Learning Physical-Informed Neural Network : MLPINN)** หมายถึงเครือข่ายประสาทที่ใช้การระบุข้อมูลทางฟิสิกส์ โดยมีการเรียนรู้แบบเมตา สามารถทำการหาคุณสมบัติทางกายภาพโดยปราศจากการระบุข้อมูล

**รูปแบบ (Model)** หมายถึงรูปแบบพฤติกรรมของสัญญาณหรือกระบวนการใดๆที่สนใจ หรือหากเป็นในกรณีของเครือข่ายประสาทเทียมจะหมายถึงตัวเครือข่ายข่ายประสาทเทียมเอง

**รูปแบบอ้างอิง (Reference Model)** หมายถึงรูปแบบพฤติกรรมของสัญญาณหรือกระบวนการใดๆที่ใช้อ้างอิงให้แก่รูปแบบอื่นๆ



## บทที่ 2

### งานวิจัยที่เกี่ยวข้อง

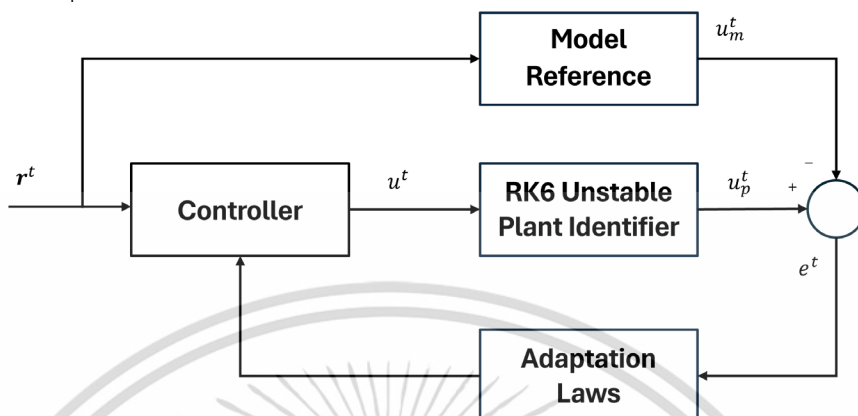
#### 2.1 Lyapunov-based MRAC การวิเคราะห์เสถียรภาพของกระบวนการที่มีสัญญาณเข้าและออกเชิงเดียว

โดยอ้างอิงจากงานวิจัย Event-triggered model reference adaptive control system design for SISO plants using meta-learning-based physics-informed neural networks without labeled data and transfer learning (Duanyai et al., 2024) MRAC คือวิธีการในการออกแบบระบบที่สามารถปรับตัวได้ โดยประสิทธิภาพของระบบจะถูกรัดจากความคลาดเคลื่อนหรือความแตกต่างระหว่างผลลัพธ์ที่ได้จากกระบวนการที่ไม่เสถียร และผลลัพธ์จากรูปแบบอ้างอิง หรือสามารถกล่าวได้ว่ารูปแบบอ้างอิงนั้นคือรูปแบบของผลลัพธ์ที่ต้องการจากกระบวนการ ตัวควบคุมแบบปรับตัวนั้นจะทำหน้าที่บังคับให้ผลลัพธ์ของกระบวนการทั้งกระบวนการที่เป็นเชิงเส้น กระบวนการที่ไม่เป็นเชิงเส้น และกระบวนการที่ไม่เสถียรเหล่านั้นให้เป็นไปตามรูปแบบอ้างอิงมากที่สุดเท่าที่จะเป็นไปได้ แต่การควบคุมดังกล่าวก็ยังจำเป็นต้องอยู่ภายใต้การพิจารณาเสถียรภาพของการควบคุม รวมไปถึงอาจประกอบร่วมกับข้อจำกัดเชิงประสิทธิภาพอื่นๆด้วย ดังรูปที่ 2.1 แผนภาพรูปแบบการควบคุมของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง จะแสดงให้เห็นถึงแผนภาพรูปแบบของการควบคุมในระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงรูปแบบทั่วไปโดยสามารถแยกองค์ประกอบออกมาด้วยกันเป็น 4 ส่วนสำคัญก็คือ

1. ตัวควบคุม (Controller)
2. ตัวหาคุณสมบัติของกระบวนการ (Plant Identifier)
3. รูปแบบอ้างอิง (Model Reference)
4. กฎการปรับแต่ง (Adaptation Law)

ตัวควบคุมจะทำหน้าที่ในการควบคุมกระบวนการโดยรับสัญญาณขาเข้าและแปลงเป็นสัญญาณควบคุม โดยในความเป็นจริงแล้วนั้น Plant Identifier จะถูกแทนที่ด้วยกระบวนการที่ต้องการควบคุมเช่น ความเร็วมอเตอร์ ระดับน้ำ และอื่นๆอีกมากมาย แต่ในกรณีการศึกษาเชิงวิจัยแล้วนั้นจะใช้วิธีการต่างๆในการจำลองกระบวนการขึ้นมาโดยกำหนดแทนด้วยสมการทางคณิตศาสตร์ในรูปแบบของสมการเชิงอนุพันธ์ ซึ่งด้วยวิธีการนี้จะสามารถทำการทดลองและทดสอบการควบคุมได้อย่างหลากหลายและมีประสิทธิภาพ โดยในจุดนี้เอง 1 ในวิธีที่หลากหลายและเป็นที่ยอมรับนั้นจะใช้ Runge-Kutta ลำดับที่ 6 ในการหาคุณสมบัติของกระบวนการ เมื่อสามารถหาผลลัพธ์ของกระบวนการที่ถูกควบคุมโดยตัวควบคุมแล้ว

นั้น จะนำผลลัพธ์มาเปรียบเทียบกับรูปแบบอ้างอิง ค่าความแตกต่างนี้เองจะเอาไปใช้ในการปรับตัวแปรภายในของตัวควบคุมโดยพิจารณาจาก Adaptation Laws ต่อไป



รูปที่ 2.1 แผนภาพรูปแบบการควบคุมของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง

จากรูปจะสามารถพิจารณาโดย  $t$  คือเวลา  $r^t$  คือสัญญาณอ้างอิงภายนอกหรือคำสั่งเป้าหมาย  $u^t$  คือสัญญาณควบคุมกระบวนการ  $u_p^t$  คือผลลัพธ์จากกระบวนการที่สนใจ  $u_m^t$  คือผลลัพธ์ของรูปแบบอ้างอิง และ  $e^t$  คือความคลาดเคลื่อนของตัวควบคุม โดยในท้ายที่สุดแล้วความคลาดเคลื่อน  $e^t$  นั้นจะมีลักษณะลู่เข้าหาค่าความคลาดเคลื่อนที่ยอมรับได้ด้วยตัวของระบบเอง

### 2.1.1 คุณสมบัติของกระบวนการ

ในกรณีของ Lyapunov-based MRAC นั้น จะใช้ทฤษฎีความเสถียรของ Lyapunov ในการออกแบบระบบควบคุม MRAC โดยสามารถออกแบบโดยให้เริ่มต้นจากการพิจารณาคุณสมบัติของกระบวนการพื้นฐานให้อยู่ในรูปแบบของสมการเชิงอนุพันธ์ลำดับที่หนึ่ง (First-order plant) ดังแสดงในสมการที่ 2.1

$$\dot{u}_p^t = -a_p \cdot u_p^t - c_p \cdot f_p^t + b_p \cdot u^t \quad (2.1)$$

พารามิเตอร์  $a_p, b_p, c_p$ , และ  $f_p^t$  จะถูกใช้ในการกำหนดคุณสมบัติของกระบวนการรวมถึงพฤติกรรมของกระบวนการด้วย อาจสามารถยกตัวอย่างการออกแบบคุณสมบัติของกระบวนการได้ออกเป็นสองกรณีคือกระบวนการที่เป็นเชิงเส้นและกระบวนการที่ไม่เป็นเชิงเส้นดังนี้

กระบวนการที่เป็นเชิงเส้น:

$$a_p = 10, \quad b_p = 3, \quad c_p = -20, \quad f_p^t = 1$$

นำพารามิเตอร์ข้างต้นไปแทนที่ในสมการที่ 2.1 จะได้

$$\dot{u}_p^t = -10 \cdot u_p^t + 20 + 3 \cdot u^t; u_p^0 = 0 \quad (2.2)$$

เมื่อพิจารณาสมการที่ 2.2 จะเห็นได้ว่าการเปลี่ยนแปลงของกระบวนการ  $\dot{u}_p^t$  จะมีลักษณะที่เป็นเชิงเส้น โดยมี  $u^t$  ที่เป็นค่าคงที่ในแต่ละช่วงเวลาย่อยเป็นสัญญาณที่ใช้ในการควบคุมโดยอ้างอิงจากค่าของกระบวนการ  $u_p^t$  เป็นจุดเริ่มต้น  
กระบวนการที่ไม่เป็นเชิงเส้น:

$$a_p = -1, b_p = 3, c_p = -1, f_p^t = (u_p^t)^2; u_p^0 = 0$$

นำพารามิเตอร์ข้างต้นไปแทนที่ในสมการที่ 2.1 จะได้

$$\dot{u}_p^t = u_p^t + (u_p^t)^2 + 3 \cdot u^t \quad (2.3)$$

เมื่อพิจารณาสมการที่ 2.3 จะเห็นได้ว่าการเปลี่ยนแปลงของกระบวนการ  $\dot{u}_p^t$  จะมีคุณสมบัติที่เป็นเลขยกกำลังทำให้มีลักษณะที่ไม่เป็นเชิงเส้นและมีความไม่เสถียร ซึ่งมี  $u^t$  ที่เป็นค่าคงที่ในแต่ละช่วงเวลาย่อยเป็นสัญญาณที่ใช้ในการควบคุมโดยอ้างอิงจากค่าของกระบวนการ  $u_p^t$  เป็นจุดเริ่มต้น

ในการทำงานเดียวกันสำหรับรูปแบบอ้างอิงเองก็สามารถพิจารณารูปแบบอ้างอิงลำดับที่หนึ่ง (First-order reference model) โดยอ้างอิงจากสมการที่ 2.1 ได้เป็นสมการที่ 2.4

$$\dot{u}_m^t = -a_m \cdot u_m^t - c_m \cdot f_m^t + b_m \cdot r^t \quad (2.4)$$

และสามารถพิจารณารูปแบบอ้างอิงที่เป็นเชิงเส้นและไม่เป็นเชิงเส้นได้ว่า  
รูปแบบอ้างอิงที่เป็นเชิงเส้น:

$$a_m = b_m = c_m = f_m^t = 0, r^t = 1; u_m^0 = 4$$

นำพารามิเตอร์ข้างต้นไปแทนที่ในสมการที่ 2.4 จะได้

$$\dot{u}_m^t = 0 \quad (2.5)$$

เมื่อพิจารณาสมการที่ 2.5 จะเห็นได้ว่าการเปลี่ยนแปลงของผลลัพธ์ของรูปแบบอ้างอิง หรือรูปแบบเป้าหมาย  $\dot{u}_m^t$  จะไม่มีการเปลี่ยนแปลงใดๆ โดยมีค่าเริ่มต้นที่  $u_m^0 = 4$  และจะคงที่ไปตลอดทุกช่วงเวลาในการควบคุมหรือสามารถกล่าวได้ว่า  $u_m^t = 4$

รูปแบบที่ไม่เป็นเชิงเส้น:

$$a_m = b_m = 4, c_m = f_m^t = 0, r^t = 4 \cdot \sin(3t); u_m^0 = 0$$

นำพารามิเตอร์ข้างต้นไปแทนที่ในสมการที่ 2.4 จะได้

$$\dot{u}_m^t = -4 \cdot u_m^t + (16)\sin(3t) \quad (2.6)$$

เมื่อพิจารณาสมการที่ 2.6 จะเห็นได้ว่าการเปลี่ยนแปลงของกระบวนการของรูปแบบอ้างอิง หรือรูปแบบเป้าหมาย  $\dot{u}_m^t$  จะมีการเปลี่ยนแปลงตลอดเวลาที่ควบคุม โดยเริ่มจุดเริ่มต้นที่  $u_m^0 = 0$  ซึ่งเมื่อถอดสมการเชิงอนุพันธ์ออกมาแล้วจะได้ว่า  $u_m^t = \frac{16}{25}(3e^{-4t} + 4\sin(3t) - 3\cos(3t))$  ซึ่งจะมีลักษณะเป็นคลื่นดังรูปที่รูปที่ 4.5 สัญญาณอ้างอิงที่ใช้ในการเปรียบเทียบ

### 2.1.2 การออกแบบตัวควบคุม

ตามทฤษฎีการออกแบบตัวควบคุมโดย Lyapunov-Theory สามารถเริ่มต้นในการออกแบบตัวควบคุมแบบปรับตัวโดยเริ่มต้นจากการหาหลักการควบคุมให้อยู่ในรูปแบบของ

$$u^t = a_u^t \cdot u_p^t + a_r^t \cdot r^t + a_f^t \cdot f_p^t \quad (2.7)$$

เมื่อ  $a_u^t, a_r^t$ , และ  $a_f^t$  คือพารามิเตอร์ตัวแปรย้อนกลับที่ปรับตัวได้ แทนสมการที่ 2.7 ในสมการที่ 2.1 จะทำให้ได้สมการการเปลี่ยนแปลงในกระบวนการระบบปิดได้ว่า

$$\dot{u}_p^t = -(a_p - b_p \cdot a_u^t) \cdot u_p^t - (c_p - b_p \cdot a_f^t) \cdot f_p^t + b_p \cdot a_r^t \cdot r^t \quad (2.8)$$

เราสามารถสร้างสมการใหม่ได้ว่า

$$a_p - a_m = b_p \cdot a_u^* \quad (2.9.1)$$

$$b_m = b_p \cdot a_r^* \quad (2.9.2)$$

$$c_p - c_m = b_p \cdot a_f^* \quad (2.9.3)$$

จากสมมติฐานที่ตั้งไว้ว่าตัวควบคุมจะทำหน้าที่ในการควบคุมกระบวนการให้มีลักษณะตรงตามรูปแบบอ้างอิง นั่นหมายความว่าหากต้องการควบคุมกระบวนการให้ไม่เกิดความคลาดเคลื่อนผลลัพธ์แล้วนั้นผลลัพธ์จากสมการที่ 2.8 และ 2.4 นั้นจะต้องเท่ากันตลอดในทุกช่วงเวลา สามารถเขียนในรูปแบบของสมการได้ว่า

$$e^t = u_p^t - u_m^t \quad (2.10)$$

ให้การปรับปรุงตัวแปรป้อนกลับเป็นดังสมการที่ 2.11

$$\tilde{a}_u^t = a_u^t - a_u^* \quad (2.11.1)$$

$$\tilde{a}_r^t = a_r^t - a_r^* \quad (2.11.2)$$

$$\tilde{a}_f^t = a_f^t - a_f^* \quad (2.11.3)$$

ซึ่งจะใช้ตัวแปรใหม่นี้ในการคาดการณ์ความคลาดเคลื่อนของพารามิเตอร์ที่ใช้ในการควบคุม โดยกำหนดให้  $a_u^*$ ,  $a_r^*$  และ  $a_f^*$  เป็นค่าการปรับเปลี่ยนตัวแปรป้อนกลับและเป็นค่าคงที่ ดังนั้นแล้วจากสมการที่ 2.10 จะสามารถพิจารณาได้อีกนัยหนึ่งว่าการเปลี่ยนแปลงของความคลาดเคลื่อนของกระบวนการ ( $e^t$ ) นั้นจะเท่ากับกับความแตกต่างระหว่างการเปลี่ยนแปลงของกระบวนการ และการเปลี่ยนแปลงของสัญญาณจากรูปแบบอ้างอิง ซึ่งสามารถหาได้จากการนำสมการที่ 2.1 ลบสมการที่ 2.8 และพิจารณาด้วยสมการที่ 2.9.1 – 2.9.3 จะได้สมการที่ 2.12 ดังนี้

$$\dot{e}^t = -a_m e^t + b_p (\tilde{a}_u^t \cdot u_p^t + \tilde{a}_r^t \cdot r^t + \tilde{a}_f^t \cdot f_p^t) \quad (2.12)$$

จะสามารถทำการวิเคราะห์เสถียรภาพของระบบและหาการลู่เข้าของความคลาดเคลื่อนของกระบวนการได้โดยใช้ทฤษฎีของ Lyapunov โดยพิจารณาฟังก์ชัน Lyapunov เข้ากับกระบวนการลำดับที่ 1 ได้ว่า

$$V(e^t, \tilde{a}_u^t, \tilde{a}_r^t, \tilde{a}_f^t) = \frac{1}{2} (e^t)^2 + \frac{1}{2\gamma} |b_p| (\tilde{a}_u^t{}^2 + \tilde{a}_r^t{}^2 + \tilde{a}_f^t{}^2) \quad (2.13)$$

หรือหากทำการหาอนุพันธ์ลำดับที่ 1 ของสมการที่ 2.13 จะได้ว่า

$$\begin{aligned} \dot{V}(e^t, \tilde{a}_u^t, \tilde{a}_r^t, \tilde{a}_f^t) = & -a_m (e^t)^2 + \frac{1}{\gamma} |b_p| [ \tilde{a}_u^t (\dot{\tilde{a}}_u^t + \text{sgn}(b_p) \gamma \cdot e^t \cdot u_p^t) \\ & + \tilde{a}_r^t (\dot{\tilde{a}}_r^t + \text{sgn}(b_p) \gamma \cdot e^t \cdot r^t) + \tilde{a}_f^t (\dot{\tilde{a}}_f^t \\ & + \text{sgn}(b_p) \gamma \cdot e^t \cdot f_p^t) ] \end{aligned} \quad (2.14)$$

เมื่อนำบทแทรกของ Barbalat (สมการที่ 2.17.1 - 2.17.3) มาวิเคราะห์ความเสถียรของสมการที่ 2.13 – 2.14 และใช้สมการที่ 2.11.1 – 2.11.3 จะสามารถออกแบบ Adaptation Laws ได้ว่า

$$\dot{\tilde{a}}_u^t = -\gamma \cdot e^t \cdot u_p^t \quad (2.15.1)$$

$$\dot{\tilde{a}}_r^t = -\gamma \cdot e^t \cdot r^t \quad (2.15.2)$$

$$\dot{\tilde{a}}_f^t = -\gamma \cdot e^t \cdot f_p^t \quad (2.15.3)$$

โดยทำการปรับปรุงตัวแปรป้อนกลับได้จาก

$$(2.16.1)$$

$$(2.16.2)$$

$$(2.16.3)$$

$$a_{u_{\pm 1 \Delta t}}^{t+\Delta t} = a_{u_{\pm 1 \Delta t}}^t + \dot{a}_{u_{\pm 1 \Delta t}}^t \cdot \Delta t$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_f^{t+\Delta t} = a_f^t + \dot{a}_f^t \cdot \Delta t$$

ซึ่งกำหนดให้  $\gamma = 1.5$  เป็นส่วนประกอบค่าคงที่ที่เป็นบวกโดยเป็นค่าที่แสดงถึงเกน (Gain) ของการปรับแต่ง และ  $\Delta t = 0.01$  เป็นเวลาที่เพิ่มขึ้นในแต่ละรอบของการควบคุม จากบทแทรกของ Barbalat ได้สนับสนุนสมการที่ 2.13 – 2.14 ไว้ว่า เมื่อ

$$V(e^t, \tilde{a}_u^t, \tilde{a}_r^t, \tilde{a}_f^t) \geq 0 \quad (2.17.1)$$

$$\dot{V}(e^t, \tilde{a}_u^t, \tilde{a}_r^t, \tilde{a}_f^t) \leq 0 \quad (2.17.2)$$

ประกอบกับ  $\dot{V}(e^t, \tilde{a}_u^t, \tilde{a}_r^t, \tilde{a}_f^t)$  มีความต่อเนื่องอย่างสม่ำเสมอตลอดช่วงเวลาดังสมการที่ 2.17.3 แล้วแสดงว่าตัวควบคุมมีเสถียรภาพเพียงพอ

$$\lim_{t \rightarrow \infty} [\dot{V}(e^t, \tilde{a}_u^t, \tilde{a}_r^t, \tilde{a}_f^t)] = 0 \quad (2.17.3)$$

เมื่อพิจารณาตามสมการที่ 2.17.2 ที่เป็นเงื่อนไขในการสร้าง Adaptation Laws อยู่แล้ว ร่วมกับพิจารณาสมการที่ 2.17.1 นั้นเป็นเงื่อนไขในการออกแบบที่ต้องมีความเกี่ยวเนื่องของทุกตัวแปรภายในเช่นเดียวกัน ซึ่งความเกี่ยวเนื่องนั้นยังรวมไปถึงความเกี่ยวเนื่องกันของตัวแปรและความคลาดเคลื่อนของกระบวนการ  $e^t$  และสุดท้ายคือการพิจารณาความต่อเนื่องของสมการที่ 2.13 ซึ่งสามารถพิสูจน์ได้จากการหาสมการเชิงอนุพันธ์ลำดับที่สองเป็นสมการที่ 2.18 เป็นหลักฐานว่ามีความต่อเนื่องอย่างสม่ำเสมอในฟังก์ชัน

$$\dot{V}(e^t, \tilde{a}_u^t, \tilde{a}_r^t, \tilde{a}_f^t) = -2a_m \cdot e^t \cdot \dot{e}^t \quad (2.18)$$

ดังนั้นแล้วจากการพิสูจน์โดยบทแทรกของ Barbalat ข้างต้นจะสามารถสรุปได้ว่าระบบ MRAC นั้นมีความเสถียรและสามารถที่จะทำให้เกิดการลู่เข้าของ  $e^t$  ได้โดยสมบูรณ์ในทุกช่วงเวลา

สมการที่ 2.1 – 2.18 พิสูจน์ใน Event-triggered model reference adaptive control system design for SISO plants using meta-learning-based physics-informed neural networks without labeled data and transfer learning (Duanyai et al., 2024)

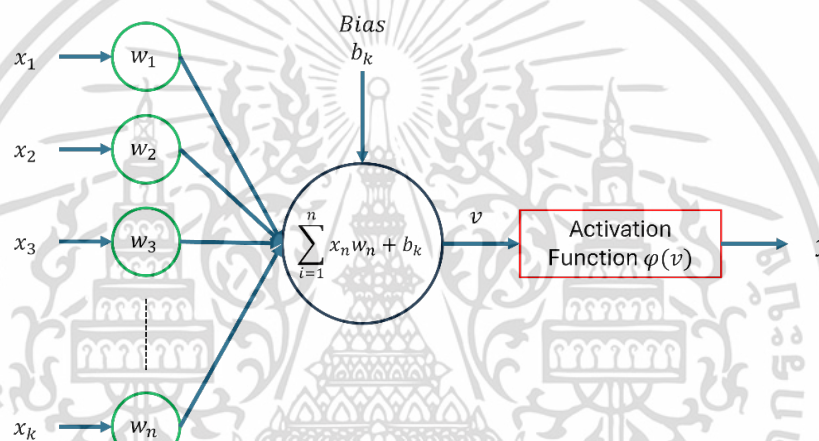
## 2.2 เครือข่ายประสาทเทียม (Artificial Neural Network)

Neural Network Principles and Applications (Zayegh & Al Bassam, 2018) ได้อธิบายเครือข่ายประสาทเทียมคือเทคนิควิธีการในการคำนวณที่ถูกรวบรวมมาเพื่อให้สามารถจำลองการคิดวิเคราะห์ของสมองมนุษย์ได้ โดยริเริ่มในปี ค.ศ. 1943 โดยนักประสาทชีววิทยา McCulloch และนักสถิติ Pitts ได้ตีพิมพ์ผลงานวิชาการที่ชื่อว่า A logical calculus of ideas immanent in nervous activity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นครั้งแรกในการอธิบายการทำงานของสมองและวิธีการพื้นฐานที่หน่วยประมวลผลของสมองหรือเซลล์ประสาททำงานร่วมกันแบบขนานเพื่อตัดสินใจบางสิ่งบางอย่างโดยอ้างอิงจากสัญญาณที่รับเข้ามา

ดังนั้นแล้วลักษณะโครงสร้างของของเครือข่ายประสาทเทียมนั้นก็ได้ออกแบบมาจากหลักการการทำงานของสมองมนุษย์ ซึ่งประกอบไปด้วยเซลล์ประสาทเทียมจำนวนมาก (Node) และเชื่อมต่อเข้าด้วยกันด้วย Weight หรือหากเป็นในเชิงของคณิตศาสตร์ก็คือพารามิเตอร์ของโมเดลการคำนวณ จากนั้นแล้วจะสามารถคำนวณหาผลลัพธ์ (Output Signal) ได้จาก Activation Function ที่ใช้ผลรวมของ Weight ทุกจุดร่วมกับสัญญาณที่รับเข้ามา (Input Signal) ดังแสดงในรูปที่ 2.2 ไดอะแกรมการทำงานของเซลล์ประสาทเทียม



รูปที่ 2.2 ไดอะแกรมการทำงานของเซลล์ประสาทเทียม

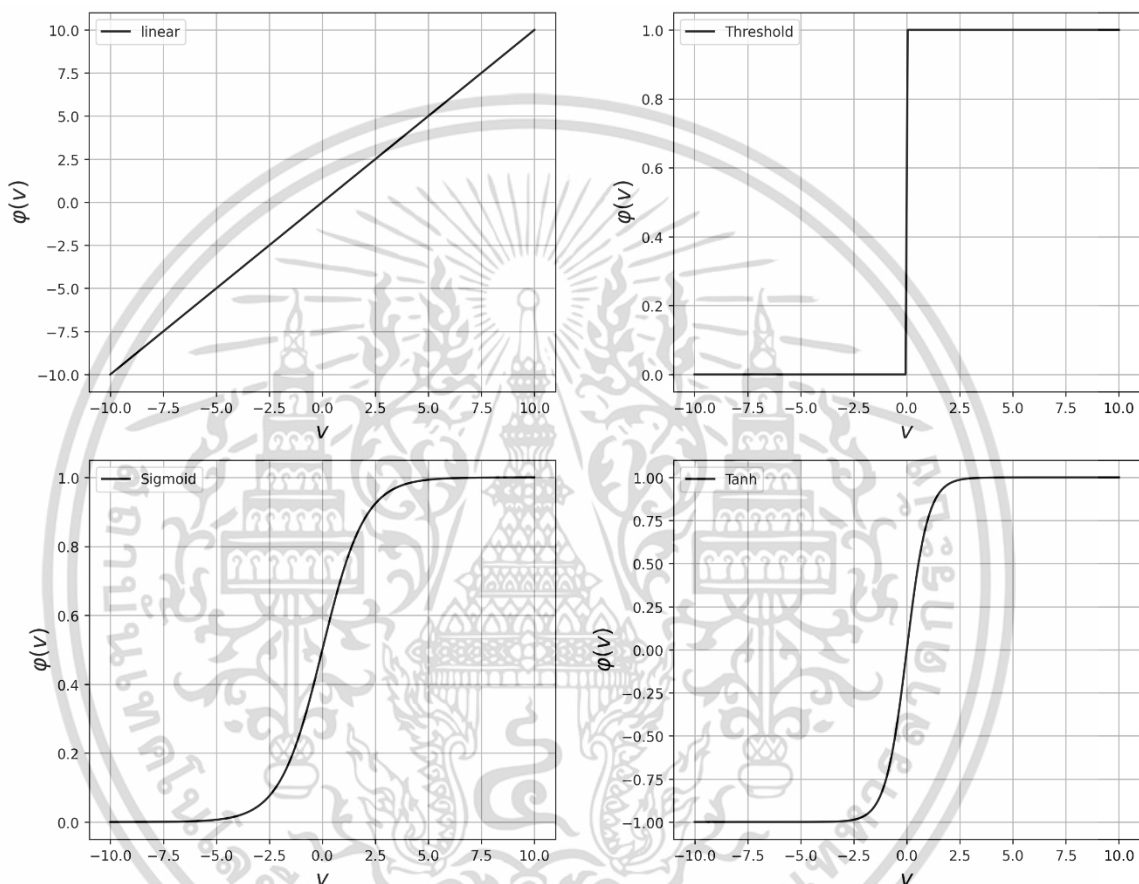
จะเห็นได้ว่าการเชื่อมต่อระหว่างเซลล์ประสาทเทียมและสัญญาณสัญญาณขาเข้านั้นจะใช้ Weight ( $w_n$ ) เป็นพารามิเตอร์ในการกำหนดน้ำหนักและความเข้มข้นของสัญญาณ จากนั้นเซลล์ประสาทจะรวบรวมทุกผลลัพธ์นั้นเข้าไว้ด้วยกัน โดยมี Bias เป็นตัวแปรรองเพื่อใช้ในการปรับแต่งสัญญาณที่สนใจ ( $v$ ) ก่อนที่จะกรองผลลัพธ์ที่ว่านั้นด้วย Activation function เพื่อกำหนดแอมพลิจูดและคุณสมบัติของผลลัพธ์ ( $y$ ) โดยสามารถอธิบายได้ในทางคณิตศาสตร์ได้เป็นดังสมการที่ 2.19

$$y = \varphi(\sum_{i=1}^n x_i w_i + b_k) \quad (2.19)$$

ต่อมาจะกล่าวถึง Activation Function หมายถึงฟังก์ชันทางคณิตศาสตร์ที่นำมาเพื่อใช้ในการกรองผลลัพธ์และกำหนดคุณสมบัติของผลลัพธ์เพื่อให้เหมาะสมกับแต่ละลักษณะงาน และทำให้การคาดเดาผลลัพธ์ของเครือข่ายประสาทเทียมมีความเที่ยงตรงและตรงต่อความต้องการ ฟังก์ชัน Activation แต่ละชนิดนั้นจะมีคุณสมบัติที่แตกต่างกันออกไปที่เป็นที่รู้จักและเป็นที่ยอมรับในปัจจุบันจะมีดังนี้

1. Linear Function:  $y = v$
2. Threshold Function:  $y = \begin{cases} 1 & \text{for } v \geq 0 \\ 0 & \text{for } v < 0 \end{cases}$
3. Sigmoid Function:  $y = \frac{1}{1+e^{-v}}$
4. Tanh Function:  $y = \frac{2}{1+e^{-2v}} - 1$

โดยแสดงลักษณะของฟังก์ชัน Activation แต่ละชนิดได้ดังรูปที่ 2.3 Activation Function



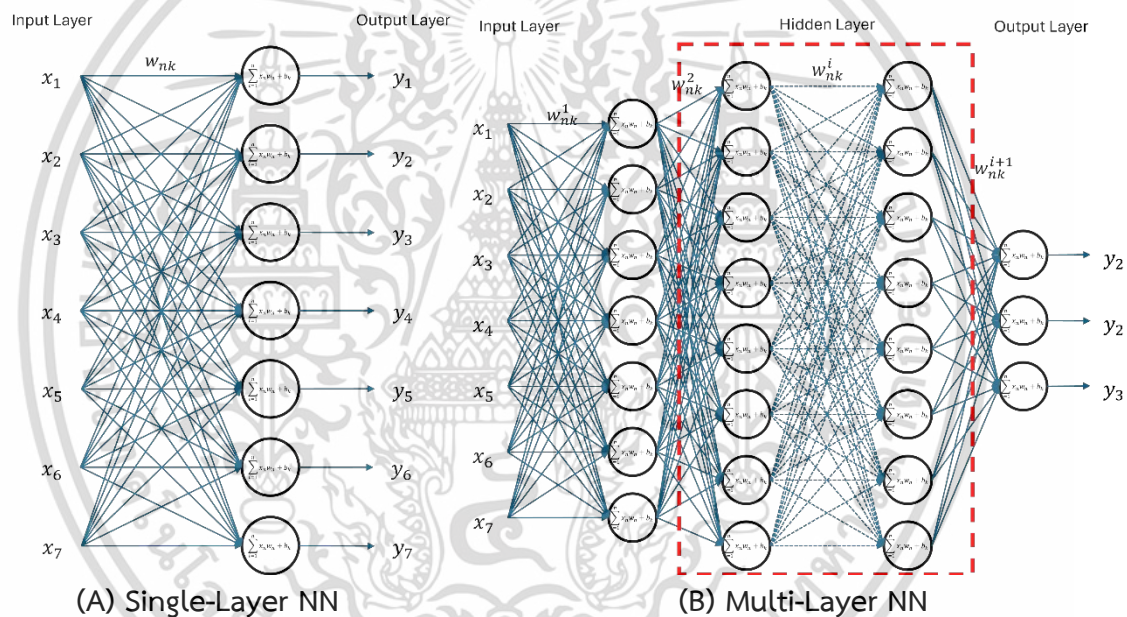
รูปที่ 2.3 Activation Function

สถาปัตยกรรมของเครือข่ายประสาทนั้นสามารถแบ่งออกได้เป็น 3 ชั้นหลักก็คือ Input Layer Hidden Layer และ Output Layer โดยในแต่ละชั้นนั้นจะมีจำนวนเซลล์ประสาทเทียมที่อาจจะเท่ากันหรือแตกต่างกันออกไปก็ได้ และในแต่ละชั้นสามารถมีฟังก์ชัน Activation เป็นของตัวเองเป็นส่วนย่อยเสริม ซึ่งรวมไปถึงสามารถมีฟังก์ชันชนิดอื่นๆร่วมด้วยขึ้นอยู่กับการออกแบบของผู้พัฒนาให้เหมาะสมต่อลักษณะงานของเครือข่ายประสาทเทียมนั้นๆเช่น ฟังก์ชันส่วนเบี่ยงเบนมาตรฐาน หรือ ฟังก์ชันการแจกแจงปกติ เป็นต้น

ในปัจจุบันนี้เราสามารถจำแนกประเภทสถาปัตยกรรมของเครือข่ายประสาทได้จากรูปแบบและจำนวนของ Layer โดยแบ่งเป็น 2 ประเภทได้คือ Single-Layer Feed Forward Neural Network และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multi-Layer Feed Forward Neural Network ซึ่งมีลักษณะตรงตามชื่อก็คือ Single-Layer จะปราศจาก Hidden Layer และ Multi-Layer นั้นจะมีจำนวน Hidden Layer 1 ชั้นหรือมากกว่าดังแสดงใน Error! Reference source not found. ซึ่งเป็นที่แน่นอนว่า Multi-Layer Feed Forward Neural Network นั้นจะมีความซับซ้อนที่มากกว่า แต่สิ่งที่โดดเด่นและทำให้เป็นที่นิยมในปัจจุบันนั้นก็คือความยืดหยุ่นและสามารถรองรับลักษณะงานที่มีความซับซ้อนและมีความผันผวนสูงได้ รวมไปถึงมีความแม่นยำที่มากกว่า เนื่องจาก Multi-Layer NN มีจำนวนพารามิเตอร์ที่มากกว่าทำให้มีความเป็นไปได้ในการสร้างรูปแบบของผลลัพธ์ใหม่ได้นั่นเอง โดยรูปแบบที่เป็นที่นิยมมากที่สุดในปัจจุบันก็คือ Multi-Layer Perceptron Neural Network ซึ่งเป็นชนิดย่อยหนึ่งของ Multi-Layer Neural Network ที่ทุกเซลล์ประสาทเทียมนั้นจะเชื่อมต่อกันโดยสมบูรณ์



รูปที่ 2.4 ประเภทของ Neural Network

จะสังเกตจากรูปที่ 2.4 ประเภทของ Neural Network ได้ว่าจำนวนของเซลล์ประสาทเทียมในแต่ละชั้น และจำนวน Hidden Layer นั้นอาจมีจำนวนไม่เท่ากันได้ ทั้งนี้ขึ้นอยู่กับการออกแบบและลักษณะของงานที่นำไปใช้เฉกเช่นเดียวกับการเลือกใช้ฟังก์ชัน Activation ในแต่ละชั้นซึ่งจะใช้ฟังก์ชันเดียวกันหรือแตกต่างกันก็ได้

หลักการของเครือข่ายประสาทนั้นจะมีอยู่ด้วยกันสองกระบวนการ คือกระบวนการที่นำไปใช้และกระบวนการในการเรียนรู้ ซึ่งเป็นสองเอกลักษณ์สำคัญที่ทำให้เครือข่ายประสาทมีคุณสมบัติดังสมองของสิ่งมีชีวิตที่สามารถทำการจดจำเรียนรู้พร้อมทั้งสามารถนำไปใช้งานได้ด้วยตัวของตนเอง โดยกระบวนการที่นำไปใช้งานนั้นจะมีเพียงการคาดเดาผลลัพธ์จากสถาปัตยกรรมข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการในการเรียนรู้ของเครือข่ายประสาทนั้นจะประกอบไปด้วยสองกระบวนการสำคัญคือ Forward Propagation และ Backward Propagation เป็นดั่งสิ่งมีชีวิต กระบวนการเหล่านี้เป็นวิธีการที่สิ่งมีชีวิตใช้ในการเรียนรู้กล่าวคือสิ่งมีชีวิตเมื่อจะเรียนรู้สิ่งใดแล้วมันจะต้องผ่านการลองผิดลองถูก เมื่อสิ่งมีชีวิตได้กระทำการใดๆแล้ว สิ่งมีชีวิตจะสามารถรับรู้ความผิดพลาดโดยอาศัยปัจจัยต่างๆเป็นตัวบ่งชี้ จากนั้นสิ่งมีชีวิตจะนำข้อผิดพลาดนั้นมาปรับปรุงแก้ไขเพื่อไม่ให้เกิดข้อผิดพลาดเดิม ซึ่งอาจจะต้องผ่านการลองผิดลองถูกเพียงไม่กี่ครั้งจนถึงนับไม่ถ้วนจึงจะสามารถเรียนรู้ได้โดยสมบูรณ์ ซึ่งกระบวนการในการเรียนรู้ของเครือข่ายประสาทเทียมนั้นใช้หลักการเดียวกันกับสิ่งมีชีวิต โดยกระบวนการ Forward Propagation คือกระบวนการที่จะทำการประมวลผลข้อมูลที่ได้รับโดยใช้พารามิเตอร์ที่มีอยู่ในการคำนวณหาผลลัพธ์ ซึ่งกระบวนการนี้เราอาจสามารถเรียกอีกอย่างว่าการคาดการณ์ (Prediction) เรียกผลลัพธ์ที่ได้จากกระบวนการนี้ว่าค่าคาดการณ์ (Prediction Value) จากนั้นจะทำการหาความผิดพลาดที่เกิดขึ้นโดยเปรียบเทียบกับค่าจริง (True Value) ซึ่งได้มาจากการบ่งชี้ต่างๆ โดยทั่วไปแล้วจะได้มาจากการที่มนุษย์เป็นผู้บอกค่าจริงให้แก่เครือข่ายประสาทเทียม (Labelling Data) โดยการเปรียบเทียบค่าทั้งสองนี้นั้นอาจมีตัวแปรรบกวนต่างๆหรือปัจจัยอื่นๆทำให้การเปรียบเทียบไม่แน่นอน หรือการเปรียบเทียบมีค่าน้อยมากเกินไปจนไม่อาจทำให้เกิดการเรียนรู้ได้ เพื่อกำจัดตัวแปรรบกวนรวมถึงขยายสัญญาณต่างๆ จะมีการใช้กระบวนการทางคณิตศาสตร์มาใช้งานโดยเรียกฟังก์ชันนี้ว่าฟังก์ชันสูญเสีย (Loss Function) ซึ่งตลอดทั้งกระบวนการนั้นเครือข่ายประสาทจะสามารถหาการเปลี่ยนแปลงของค่าการสูญเสียเทียบกับการเปลี่ยนแปลงของพารามิเตอร์ภายใน เรียกการเปลี่ยนแปลงนี้ว่าเกรเดียน (Gradient) ของการสูญเสีย  $\left(\frac{\partial L}{\partial \theta}\right)$  เป็นอันเสร็จสิ้นกระบวนการ Forward Propagation หลังจากนั้นกระบวนการ Back Propagation จะนำเกรเดียนที่เกิดขึ้นนำไปปรับปรุงพารามิเตอร์โดยใช้กระบวนการทางคณิตศาสตร์ โดยฟังก์ชันที่ใช้งานจะแตกต่างกันไปตามแต่อัลกอริทึมที่เลือกใช้เช่น Stochastic Gradient Descent (SGD) ดังแสดงในสมการที่ 2.20 ซึ่งเป็นอัลกอริทึมสมการที่ใช้ในการเปลี่ยนแปลงพารามิเตอร์ของเครือข่ายประสาทเทียม โดยการใช้การเปลี่ยนแปลงของค่าการสูญเสียเทียบกับพารามิเตอร์โดยมี  $\alpha$  เป็นค่าที่ใช้ในการปรับอัตราการเปลี่ยนแปลงซึ่งอาจสามารถเรียกพารามิเตอร์นี้ได้ว่าเป็นค่าของอัตราการเรียนรู้ (Learning Rate)

$$\theta_{ij} = \theta_{ij} - \alpha \frac{\partial L}{\partial \theta_{ij}} \quad (2.20)$$

จะสังเกตเห็นว่าเครือข่ายประสาทเทียมนั้นถูกสร้างโดยอาศัยพารามิเตอร์มากมายรวมไปถึงพารามิเตอร์เหล่านี้สามารถปรับปรุงได้ สามารถจำแนกพารามิเตอร์ออกได้เป็นสองประเภทคือประเภทที่ใช้ในการคาดเดาโดยตรงซึ่งก็คือ Weight โดยจะเรียกประเภทนี้ว่าพารามิเตอร์ปกติ และประเภทที่ไม่ส่งผลต่อการคาดเดาโดยตรงแต่ส่งผลต่อประสิทธิภาพของกระบวนการเรียนรู้และลักษณะพฤติกรรมของเครือข่ายประสาท

เทียบเช่น จำนวนเซลล์ประสาทในแต่ละเลเยอร์ จำนวนของ Hidden Layer ฟังก์ชัน Activation และ อัตราการเรียนรู้ เป็นต้น โดยจะเรียกพารามิเตอร์ประเภทนี้ว่าไฮเพอร์พารามิเตอร์ (Hyperparameter)

### 2.3 MLPINN

ในบทย่อๆนี้จะกล่าวถึงทฤษฎีของ Meta-Learned Physical Inform Neural Network (Duanyai, 2024) โดยต่อไปนี้จะเรียกโดยย่อว่า MLPINN และเรียกว่า Physical inform Neural Network ว่า PINN

MLPINN นั้นคือชนิดย่อยของ PINN ที่มีการเรียนรู้แบบเมต้า ซึ่งมีความสามารถในการหาผลลัพธ์ของสมการเชิงอนุพันธ์ได้โดยปราศจากการระบุข้อมูล (Labelling Data) มีทั้งความแม่นยำ และง่ายต่อการใช้งานเนื่องจากไม่มีความจำเป็นที่จะต้องให้มนุษย์หรืออุปกรณ์อื่นในการระบุข้อมูลที่ต้องอาศัยคณิตศาสตร์ในการหาผลลัพธ์ โดยงานวิจัยฉบับนี้จะนำ MLPINN มาเป็นส่วนหนึ่งของการสร้างระบบควบคุมแบบอัจฉริยะขึ้นมาในส่วนของ การหาคุณลักษณะของกระบวนการ หากยกตัวอย่างโดยง่ายคือการนำ MLPINN มาใช้ในการถอดสมการที่ 2.1 เพื่อใช้ในการหาผลลัพธ์ของกระบวนการ จากนั้นนำไปเปรียบเทียบกับรูปแบบอ้างอิงเพื่อทำการควบคุมต่อไป

PINN เป็นเครือข่ายประสาทชนิดหนึ่งที่ใช้ในการประมาณการผลลัพธ์ของฟังก์ชันทางคณิตศาสตร์โดยเรียนรู้จากความสัมพันธ์ภายในของฟังก์ชัน ซึ่ง PINN ใช้ตัวเลขเป็นสัญญาณรับเข้าและผ่านการประมวลผลของเครือข่ายประสาทเทียมหลายชั้นแบบเพอร์เซพตรอนก่อนจะทำการคาดการณ์ผลลัพธ์ออกมา ซึ่งสามารถอธิบายเป็นสมการที่ 2.21 ได้ว่า

$$u_p^t = f_{\theta}(x) \quad (2.21)$$

โดย  $x$  คือเวกเตอร์สัญญาณรับเข้า และ  $\theta$  คือพารามิเตอร์ที่สามารถเรียนรู้ได้ของเครือข่ายประสาทเทียม โดย PINN แบบทั่วไปนั้นสามารถแบ่งออกเป็น 2 ประเภทได้คือ Initial Value Problem และ Initial-Boundary Value problem ซึ่งต่างก็เป็นวิธีการเรียนรู้ที่อาศัยการระบุข้อมูลทั้งสิ้น ซึ่งจะแตกต่างกับวิธีการเรียนรู้แบบเมต้าที่ไม่ต้องใช้การระบุข้อมูล

โดยในงานวิจัยนี้สำหรับการหาคุณสมบัติกระบวนการจะกำหนดให้  $x$  หรือสัญญาณรับเข้าในชั้นแรกคือ  $t$  หรือเวลา และผลลัพธ์ของแต่ละชั้นคือกระบวนการ Forward Propagation ซึ่งสามารถอธิบายได้สมการที่ 2.22 และ รูปที่ 4.2 Physical Inform Neural Network รูปแบบเรียนรู้

$$\begin{aligned}
\mathbf{x} &= \mathbf{t} \\
\mathbf{z}^{[1]} &= \sigma^{[1]}(\boldsymbol{\theta}^{[0]} \cdot \mathbf{x} + \mathbf{b}^{[1]}) \\
&\vdots \\
\mathbf{z}^{[l]} &= \sigma^{[l]}(\boldsymbol{\theta}^{[l-1]} \cdot \mathbf{z}^{[l-1]} + \mathbf{b}^{[l]}) \\
u_p^t &= \boldsymbol{\theta}^{[l]} \cdot \mathbf{z}^{[l]} + \mathbf{b}^{[l+1]}
\end{aligned} \tag{2.22}$$

จากรูปที่ 4.2 Physical Inform Neural Network รูปแบบเรียนรู้จะเห็นได้ว่าในส่วนช่วงการทำงานในส่วนของเครือข่ายประสาทนั้นจะมีลักษณะที่เหมือนกับเครือข่ายประสาททั่วไป แต่เมื่อเข้าสู่ช่วงของการเรียนรู้ Forward Propagation และ Backward Propagation แล้วนั้นจะมีความแตกต่างออกไป ในกรณีของเครือข่ายประสาทโดยปกตินั้นจะใช้การระบุข้อมูลแบ่งออกเป็นประเภทของชุดข้อมูลคือ ชุดข้อมูลสำหรับการฝึกสอน (Training Dataset) และ ชุดข้อมูลสำหรับการตรวจสอบ (Test Dataset) แต่ในกรณีของ PINN นั้นจะใช้สมการทางคณิตศาสตร์หรือคุณสมบัติของกระบวนการในการเป็นตัวแทนของความสัมพันธ์ภายในแทนการระบุข้อมูลส่วนใหญ่ โดยอาศัยความสามารถในการติดตามการเปลี่ยนแปลงของเครือข่ายประสาทในการหาค่าการสูญเสียของการเรียนรู้ ซึ่งPINNจะทำการปรับปรุงพารามิเตอร์ของตนเองโดยมีเป้าหมายในการลดค่าการสูญเสียให้น้อยที่สุดในกระบวนการ Backward propagation และจะทำซ้ำกระบวนการนี้แต่ต้นอีกครั้งไปจนกว่าจะครบเงื่อนไขบางประการเช่นว่าจำนวนรอบในการเรียนรู้ถึงขีดจำกัดที่ตั้งไว้, ค่าการสูญเสียของการเรียนรู้อยู่ในค่าที่สามารถยอมรับได้แล้ว หรือ ค่าการสูญเสียที่เกิดขึ้นมีลักษณะที่ลู่ออกค่าใดค่าหนึ่งจนเกิดการเปลี่ยนแปลงน้อยกว่าค่าที่ยอมรับได้ โดยสามารถพิจารณาLossที่เกิดขึ้นได้จากสมการที่ 2.23 – 2.24

$$L(\boldsymbol{\theta}, \mathbf{D}_{tr}) = \lambda_f MSE_f + \lambda_I MSE_I + \lambda_B MSE_B + \lambda_d MSE_d \tag{2.23}$$

เมื่อ  $\lambda_f = \lambda_I = \lambda_B = \lambda_d = 1$  ค่าการสูญเสียที่เกิดขึ้นจะหาได้จากผลรวมของ Mean Square Error (MSE) ของแต่ละชุดอินพุตของเครือข่ายประสาท และสามารถหา MSE ได้จากสมการที่ 2.24

$$MSE_f = \frac{1}{N_f} \sum_{p=1}^{N_f} [f(u_{p,N}^t)]^2 \tag{2.24.1}$$

$$MSE_I = \frac{1}{N_I} \sum_{N=1}^{N_I} (u_{p,N}^t - \bar{u}_N^t)^2 \tag{2.24.2}$$

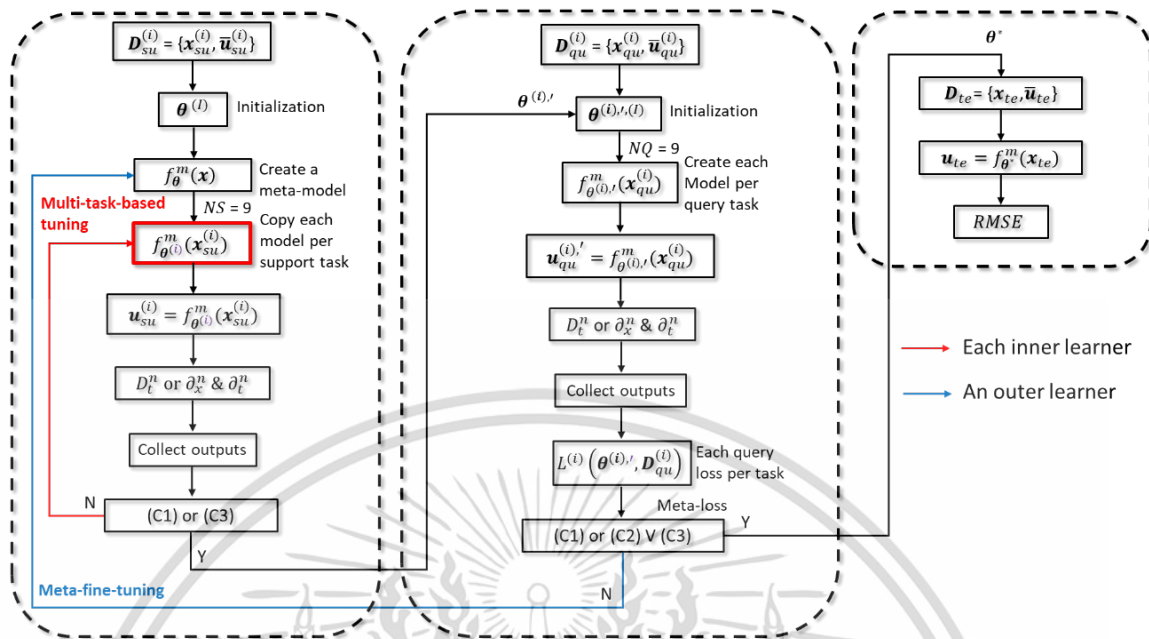
$$MSE_B = \frac{1}{N_B} \sum_{N=1}^{N_B} (u_{p,N}^t - \bar{u}_N^t)^2 \tag{2.24.3}$$

$$MSE_d = \frac{1}{N_d} \sum_{N=1}^{N_d} (u_{p,N}^t - \bar{u}_N^t)^2 \tag{2.24.4}$$

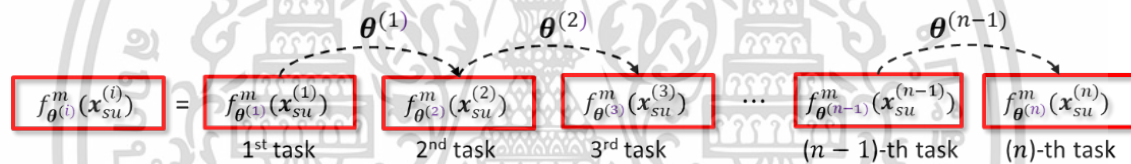
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PINN เป็นเครือข่ายประสาทที่มีความน่าเชื่อถือในหลักการการทำงานเป็นอย่างมาก เป็นเครือข่ายประสาทชนิดหนึ่งที่สามารถแก้ปัญหาทางคณิตศาสตร์ต่างๆได้โดยที่สามารถตรวจสอบความถูกต้องได้ด้วยตัวของเครือข่ายประสาทเอง โดยใช้การระบุข้อมูลที่แท้จริงเพียงเล็กน้อยเท่านั้นทั้งในชนิด IVP และ IBVP โดยปกติทั่วไปแล้ว IVP จะใช้การระบุข้อมูลที่ถูกต้องที่ตำแหน่งเริ่มต้นเพียงเท่านั้น ซึ่งวิธีการนี้เองทำให้เกิดปัญหาเมื่อสัญญาณที่รับเข้ามาเริ่มออกห่างจากอินพุตเริ่มต้นมากเท่าไร ค่า Loss ที่เกิดขึ้นก็จะสะสมมากขึ้นจนในท้ายที่สุดค่า Loss ก็จะมีเกินกว่าค่าที่ยอมรับได้ อย่างไรก็ตาม IBVP PINN จะลดการเกิดปัญหาดังที่ว่านี้เนื่องจาก IBVP มีลักษณะการระบุข้อมูลในหลายๆช่วงอินพุตที่สม่ำเสมอทำให้ PINN สามารถเรียนรู้ค่าที่ถูกต้องได้จากตำแหน่งเหล่านั้นและใช้ในการเรียนรู้ แต่ทว่าวิธีการ IBVP ต้องแลกมากับการระบุข้อมูลที่มากขึ้นตามมา จำเป็นที่ต้องใช้อุปกรณ์อื่นๆช่วยในการระบุข้อมูลที่ถูกต้อง

การเรียนรู้แบบเมตาเป็นเทคนิควิธีการที่มีงานวิจัยที่นำมาแก้ปัญหา IVP ของ PINN ซึ่งมีส่วนช่วยเป็นอย่างมากในการปรับปรุงคุณภาพของเครือข่ายประสาทอย่างมีนัยสำคัญ โดยการเรียนรู้แบบเมตาถูกออกแบบขึ้นมาโดยมีหัวใจหลักในการปรับปรุงความสามารถในการปรับตัวของค่าเริ่มต้นของพารามิเตอร์ของเครือข่ายประสาท  $\theta$  ซึ่งทำให้เกิดวิธีการในการปรับปรุงพารามิเตอร์ให้ลู่เข้าไปในทางที่ดีกว่าเดิม โดยหลักการการพัฒนาการเรียนรู้แบบเมตาขึ้นมานั้นต้องใช้การปรับตัวของเครือข่ายประสาทสองระดับ คือ ตัวเรียนรู้ภายนอกและตัวเรียนรู้ภายใน ตัวเรียนรู้ภายนอกในทุกๆรอบของการเรียนรู้จะทำหน้าที่ในการเป็นแม่แบบและกำหนดค่าเริ่มต้นให้แก่ตัวเรียนรู้ภายในแต่ละตัว จากนั้นตัวเรียนรู้ภายในแต่ละตัวจะทำการเรียนรู้โดยสุ่มข้อมูลที่แตกต่างกันและรวบรวม Loss ที่เกิดทั้งหมดนำไปปรับแต่งละเอียดที่ตัวเรียนรู้ภายนอกอีกครั้งหนึ่งเพื่อเป็นการเปิดเส้นทางในการเรียนรู้ที่ดีกว่าการมีตัวเรียนรู้เพียงตัวเดียวดังแสดงในรูปที่ 2.5 แผนภาพกลไกการทำงานของ MLPINN (Duanyai, 2024) และรูปที่ 2.6 การส่งต่อการเรียนรู้ของตัวเรียนรู้ภายใน (Duanyai, 2024) ที่แสดงถึงหลักการการทำงานของ MLPINN



รูปที่ 2.5 แผนภาพกลไกการทำงานของ MLPINN (Duanyai, 2024)



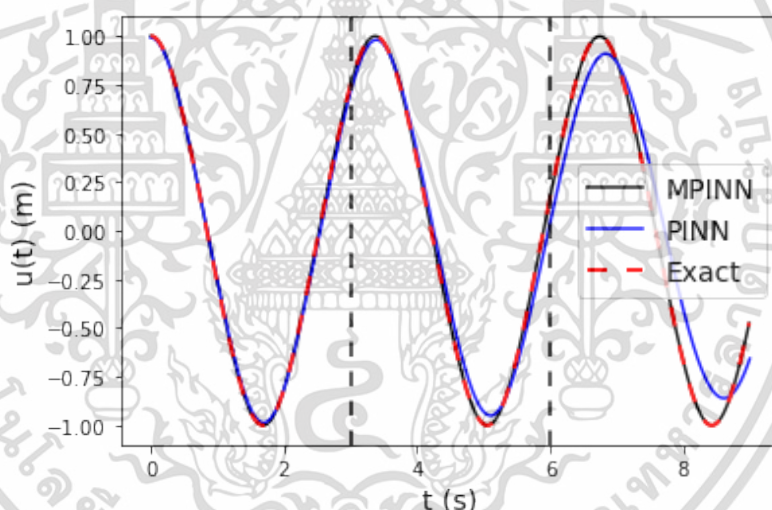
รูปที่ 2.6 การส่งต่อการเรียนรู้ของตัวเรียนรู้อยู่ภายใน (Duanyai, 2024)

จากรูปที่ 2.5 แผนภาพกลไกการทำงานของ MLPINN (Duanyai, 2024) ในขั้นตอนแรกนั้น PINN จะเริ่มต้นการเรียนรู้จากการสร้างชุดข้อมูลเริ่มต้น  $D_{su}^{(i)}$  ขึ้นมาแบบสุ่มโดยประกอบไปด้วยอินพุตแบบสุ่มคือ  $x_{su}^{(i)}$  หรือหากเป็นการหาค่าคุณสมบัติของกระบวนการก็คือชุดข้อมูลของเวลาที่สนใจ และค่าของกระบวนการที่จุดเริ่มต้น  $\bar{u}_{su}^{(i)}$  เพียงหนึ่งจุด จากนั้นทำการกำหนดค่าเริ่มต้นให้แก่พารามิเตอร์ของเครือข่ายแม่แบบและทำการสร้างโมเดลเครือข่ายประสาทขึ้นมาตามสถาปัตยกรรมที่ได้ตั้งค่าเอาไว้ ซึ่งจากกระบวนการจนถึงขั้นตอนนี้คือกระบวนการในการสร้าง PINN ในรูปแบบปกตินั่นเอง จากรูปที่ 2.5 แผนภาพกลไกการทำงานของ MLPINN จะแสดงให้เห็นว่าการเรียนรู้แบบเมตานั้นจะทำการสร้างตัวเรียนรู้อยู่ภายในขึ้นมาตามจำนวน  $NS$  โดยในลักษณะเดียวกัน  $NQ$  จะมีจำนวนที่เท่ากัน ตัวเรียนรู้อยู่ภายในแต่ละตัวจะทำการเรียนรู้และปรับตัวโดยเมื่อเข้าสู่เงื่อนไขของการหยุดการเรียนรู้แล้วก็จะเริ่มการเรียนรู้ของตัวเรียนรู้อยู่ภายในในลำดับถัดไป ซึ่งพารามิเตอร์ที่ตัวเรียนรู้อยู่ภายในที่ได้ทำการเรียนรู้และปรับแต่งแล้วนั้นจะถูกกำหนดเป็นพารามิเตอร์เริ่มต้นให้กับตัวเรียนรู้อยู่ภายในตัวถัดไป เรียกกระบวนการนี้ว่า Transfer Learning ดังแสดงให้เห็นในรูปที่ 2.6 การส่งต่อการเรียนรู้ของตัวเรียนรู้อยู่ภายในกระบวนการนี้เองช่วยให้ระยะเวลาที่ใช้ในการเรียนรู้สั้นลงเป็นอย่างมาก เมื่อตัวเรียนรู้อยู่ภายในทุกตัวได้ทำการเรียนรู้และปรับแต่งพารามิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสร็จสิ้นแล้ว พารามิเตอร์จะนำไปปรับปรุงให้แก่ตัวเรียนรู้ภายนอกเพื่อใช้ในการหา Loss ของตัวเรียนรู้ภายนอกโดยใช้ชุดข้อมูลจำนวน  $NQ$  จากนั้นทำการรวบรวม Loss ที่เกิดขึ้นทั้งหมดและตรวจสอบตามเงื่อนไขซึ่งประกอบไปด้วยการตรวจสอบ Loss ถึงค่าเป้าหมายหรือไม่ มีจำนวนรอบการเรียนรู้ภายนอกถึงค่าที่กำหนดหรือไม่ และสุดท้ายความแตกต่างของ Loss ที่เกิดขึ้นนั้นมีลักษณะลู่เข้าค่าใดค่าหนึ่งและมีการเปลี่ยนแปลงน้อยกว่าค่าที่ยอมรับได้หรือไม่ หากไม่เป็นไปตามเงื่อนไขดังกล่าว ผลรวมของ Loss ดังสมการที่ 2.22 จะถูกนำไปหาค่าของเกรเดียนต์และนำไปปรับละเอียดตัวเรียนรู้ภายนอก และวนกระบวนการใหม่อีกครั้งหนึ่ง แต่หากเป็นไปตามเงื่อนไขกระบวนการการเรียนรู้เมื่อดำเนินการสิ้นสุดลงประสิทธิภาพของการเรียนรู้แบบเมต้าแสดงในรูปที่ 2.7 การเปรียบเทียบระหว่าง PINN ชนิด IVP และ MLPINN ของการจำลองการสั่นพ้องแบบ Duffing (Duanyai, 2024)

สมการที่ 2.21 – 2.24 พิสูจน์ใน Meta-learning-based Physics-informed Neural Network: Numerical Simulations of Initial Value Problems of Nonlinear Dynamical Systems without Labeled Data and Correlation Analyses (Duanyai, 2024)



รูปที่ 2.7 การเปรียบเทียบระหว่าง PINN ชนิด IVP และ MLPINN ของการจำลองการสั่นพ้องแบบ Duffing (Duanyai, 2024)

## 2.4 Hyper Parameter Optimization ด้วยวิธีการปรับแต่งแบบ Bayesian

จากงานวิจัย On hyperparameter optimization of machine learning algorithms: Theory and practice (Yang & Shami, 2020) ไฮเปอร์พารามิเตอร์คือพารามิเตอร์ที่เครือข่ายประสาทใช้ในการกำหนดรูปแบบในการเรียนรู้ เช่นจำนวนหน่วยประสาท (Neural Node), จำนวนชั้นซ่อน (Hidden Layer), วิธีการตั้งพารามิเตอร์เริ่มต้น (Initializer), อัตราการเรียนรู้ (Learning Rate) และอื่นๆ โดยปกติแล้วไฮเปอร์พารามิเตอร์เหล่านี้จะถูกกำหนดโดยผู้พัฒนาหรือผู้วิจัย ซึ่งการเปลี่ยนแปลงต่าง ๆ นั้น

ขึ้นอยู่กับความเชี่ยวชาญและประสบการณ์ของผู้วิจัยนั่นเอง อย่างไรก็ตามกระบวนการในการปรับแต่งเหล่านี้อาจใช้เวลาไม่น้อยแตกต่างกันออกไป ซึ่งไม่มีความแน่นอนขึ้นอยู่กับหลากหลายปัจจัย

Hyper Parameter Optimization โดยต่อไปนี้จะเรียกโดยย่อว่า HPO คือกระบวนการในการปรับแต่งไฮเพอร์พารามิเตอร์ของเครือข่ายประสาท มีจุดประสงค์หลักในการทำการปรับแต่งไฮเพอร์พารามิเตอร์แบบอัตโนมัติ และนำไปใช้ในการตั้งค่าโมเดลเครือข่ายประสาทในแต่ละลักษณะงานต่อไป ซึ่งจะเป็นการลดหน้าที่ในการปรับแต่งของมนุษย์ลง, สร้างความสะดวกสบายให้มากขึ้น, ปรับปรุงประสิทธิภาพของโมเดลเครือข่ายประสาท และ ทำให้งานวิจัยต่างๆนั้นสามารถทำซ้ำได้อย่างมีประสิทธิภาพมากขึ้น

ในปัจจุบัน HPO นั้นมีอยู่ด้วยกันหลากหลายเทคนิคที่ใช้ตั้งแต่วิธีการดั้งเดิมอย่าง Gradient Descent ซึ่งเป็นวิธีการที่ล้าสมัยและอาจไม่เหมาะสมต่อลักษณะงานในปัจจุบันไปจนถึงวิธีการที่ใหม่กว่าและเหมาะสมกว่าเช่น วิธีการ Decision-Theoretic การปรับแต่งแบบBayesian การปรับแต่งแบบMulti-fidelity และอัลกอริทึมMetaheuristics ซึ่งวิธีการสมัยใหม่เหล่านี้จะใช้หลักการในการหาค่าของไฮเพอร์พารามิเตอร์ที่เหมาะสมที่สุดที่อยู่ภายในย่านของไฮเพอร์พารามิเตอร์ที่กำหนดเอาไว้ แต่จะมีความแตกต่างการในวิธีการหาไฮเพอร์พารามิเตอร์เหล่านั้น โดยในงานวิจัยฉบับนี้จะนำวิธีการปรับแต่งแบบ Bayesian มาใช้งาน

กระบวนการ HPO ที่ใช้การปรับแต่งแบบ Bayesian นั้นจะมีหลักการในการปรับแต่งที่แตกต่างจากวิธีการอื่นๆ ในกระบวนการการหาค่าไฮเพอร์พารามิเตอร์ที่เหมาะสมนั้นจะทำการทดสอบไฮเพอร์พารามิเตอร์เพื่อหาผลลัพธ์ของฟังก์ชันเป้าหมายจากนั้น จะทำการหาค่าไฮเพอร์พารามิเตอร์ทดสอบลำดับถัดไปโดยอ้างอิงจากผลลัพธ์ของการทดสอบไฮเพอร์พารามิเตอร์ก่อนหน้านี้โดยหลีกเลี่ยงค่าไฮเพอร์พารามิเตอร์ที่ไม่จำเป็นออกไป นั่นหมายความว่าจะสามารถหาไฮเพอร์พารามิเตอร์ที่เหมาะสมที่สุดได้โดยมีแนวโน้มในการใช้รอบการเรียนรู้ที่น้อยกว่าวิธีการอื่นๆ โดยการปรับแต่งแบบ Bayesian นั้นจะมีขั้นตอนพื้นฐานอยู่ 5 ขั้นตอนดังนี้

1. สร้างโมเดลจำลองที่แสดงถึงความน่าจะเป็นฟังก์ชันเป้าหมาย
2. ทำการหาไฮเปอร์พารามิเตอร์ที่น่าจะเหมาะสมที่สุดจากโมเดลความน่าจะเป็นในขั้นตอนที่ 1
3. ทำการทดสอบไฮเปอร์พารามิเตอร์จากผลลัพธ์ในขั้นตอนที่ 2 ด้วยฟังก์ชันเป้าหมาย
4. ปรับแต่งโมเดลจำลองด้วยผลลัพธ์การทดสอบในขั้นตอนที่ 3
5. ทำซ้ำขั้นตอนที่ 2-4 จนกว่าจะถึงขอบเขตของจำนวนรอบในการเรียนรู้หรือเงื่อนไขบางประการ

โมเดลจำลองความน่าจะเป็นของฟังก์ชันเป้าหมายในขั้นตอนที่ 1 นั้นมีอยู่ด้วยกันหลายโมเดลที่ใช้ในปัจจุบันตัวอย่างเช่น Gaussian Process (GP) Random Forest (RF) และ Tree Parzen Estimator (TPE) โดยในงานวิจัยนี้จะกล่าวเฉพาะโมเดลพื้นฐานอย่าง Gaussian Process เท่านั้น

### 2.4.1 การสร้างโมเดลจำลองด้วย Gaussian Process

อ้างอิงจากงานวิจัย A Tutorial Bayesian Optimization (Frazier, 2018) สามารถเริ่มพัฒนา Bayesian Optimization ด้วยการพิจารณาฟังก์ชันเป้าหมายที่มีไฮเปอร์พารามิเตอร์เป็นตัวแปรภายใน เรียกว่า  $f(\lambda)$  โดยไฮเปอร์พารามิเตอร์นั้นมีความน่าจะเป็นที่เป็นไปได้อยู่หลายตำแหน่ง  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$  ตามทฤษฎีเมื่อใดก็ตามที่มีปริมาณจำนวนที่ไม่รู้จักในเชิงสถิติของ Bayesian แล้ว เราสามารถคาดเดาได้ว่าปริมาณเหล่านั้นเป็นการสุ่มโดยธรรมชาติด้วยการแจกแจงความน่าจะเป็นบางอย่าง ซึ่ง Gaussian process จะทำหน้าที่ในการทำให้รูปแบบการแจกแจงนี้เป็นหลายตัวแปรแบบปกติของสมการทั่วไป โดยใช้ค่าเฉลี่ยของไฮเปอร์พารามิเตอร์เหล่านั้นร่วมกับเมทริกซ์ความแปรปรวนร่วม (Covariance Matrix) เราจะสามารถสร้างฟังก์ชันค่าเฉลี่ยได้จากค่าเฉลี่ยของไฮเปอร์พารามิเตอร์ในแต่ละไฮเปอร์พารามิเตอร์ และสร้างเมทริกซ์ความสัมพันธ์ได้จากไฮเปอร์พารามิเตอร์ที่อยู่ในเมทริกซ์ จากนั้นแล้วฟังก์ชันเคอร์เนลจะมีแนวโน้มในการเลือกไฮเปอร์พารามิเตอร์ชุดใหม่ที่อยู่ที่ใกล้กับไฮเปอร์พารามิเตอร์ทดสอบเนื่องจากมีความสัมพันธ์ที่มากกว่าตำแหน่งที่ไกลออกไป

$$f(\lambda) \sim GP(m(\lambda), K(\lambda, \lambda)) \quad (2.25)$$

พิจารณาสมการที่ 2.25  $f(\lambda)$  นั้นคือฟังก์ชันเป้าหมายของ Gaussian Process (GP) โดยจะกำหนดคุณสมบัติกระบวนการเกาส์เซียน โดยสองคุณสมบัติคือ  $m(\lambda)$  เป็นฟังก์ชันเส้นค่าเฉลี่ยของผลลัพธ์ของฟังก์ชันเป้าหมายที่มีชุดไฮเปอร์พารามิเตอร์ที่อยู่บนย่านที่สนใจ และ  $K(\lambda, \lambda)$  คือเมทริกซ์ความแปรปรวนร่วมหรืออีกชื่อคือเมทริกซ์เคอร์เนลที่อยู่ในรูปแบบของ  $k(\lambda_i, \lambda_j)$  โดยมีชุดค่าเริ่มต้นของไฮเปอร์พารามิเตอร์เป็น  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$  ในที่นี้  $n$  หมายถึงจำนวนของไฮเปอร์พารามิเตอร์ทั้งหมดในย่านไฮเปอร์พารามิเตอร์นั้นๆ ซึ่งฟังก์ชันเป้าหมายของไฮเปอร์พารามิเตอร์นั้นถูกกำหนดด้วยสมการที่ 2.26

$$f(\lambda) = f(\lambda_1, \lambda_2, \dots, \lambda_n) \quad (2.26)$$

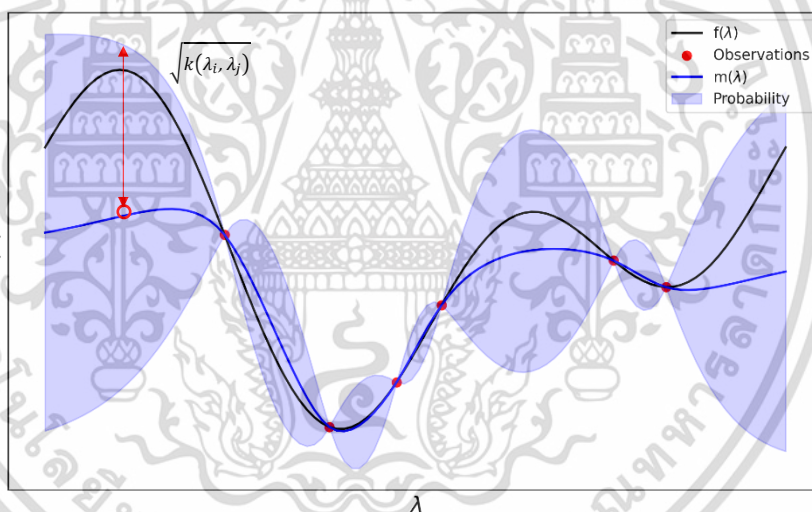
$$k(\lambda_i, \lambda_j) = c \cdot \exp\left(-\frac{|\lambda_i - \lambda_j|^2}{2l^2}\right) + \sigma^2 \cdot \delta_{ij} \quad (2.27)$$

ผลลัพธ์ความสัมพันธ์ของความแปรปรวนในกระบวนการเกาส์เซียนสามารถหาได้ด้วยฟังก์ชันเคอร์เนล หรือฟังก์ชันค่าเฉลี่ยสามารถหาได้จากฟังก์ชันในรูปแบบรูปแบบของ Radial Basis Function (RBF) ดังแสดงในสมการที่ 2.27 โดยสามารถพิจารณาให้  $c$  คืออัตราเปลี่ยนแปลงโดยรวม  $l$  คืออัตราการเปลี่ยนแปลงแนวยาว  $\sigma^2$  คือตัวชี้วัดความแปรปรวนของชุดข้อมูล และ  $\delta_{ij}$  คือความแตกต่าง Kronecker ที่  $\delta_{ij} = 1$  เมื่อ  $i = j$  และ  $\delta_{ij} = 0$  เมื่อ  $i \neq j$  โดยฟังก์ชันเคอร์เนลที่อยู่ในรูปแบบของ

สมการที่ 2.27 นั้นมีอยู่หลายฟังก์ชันให้สามารถเลือกใช้งานได้ โดยหนึ่งในนั้นที่สามารถวิเคราะห์และพัฒนาได้โดยง่ายที่งานวิจัยฉบับนี้นำมาใช้ก็คือ Matérn kernel ดังแสดงในสมการที่ 2.28

$$k(\lambda_i, \lambda_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left( \frac{\sqrt{2\nu}}{l} d(\lambda_i, \lambda_j) \right)^\nu K_\nu \left( \frac{\sqrt{2\nu}}{l} d(\lambda_i, \lambda_j) \right) \quad (2.28)$$

ฟังก์ชัน Matérn kernel เป็นที่ใช้กันอย่างแพร่หลายในกระบวนการเกาส์เซียนเนื่องจากมีพารามิเตอร์  $\nu$  ที่ใช้ในการควบคุมความสั่นไหวของฟังก์ชัน และมีรูปแบบของฟังก์ชันที่คล้ายคลึงกับ RBF ทำให้ง่ายต่อการเข้าใจ โดยหากทำการปรับลดพารามิเตอร์  $\nu$  จะทำให้ความสั่นไหวของฟังก์ชันน้อยลง และหาก  $\nu \rightarrow \infty$  จะทำให้ฟังก์ชันเคอร์เนลนั้นจะมีลักษณะที่ตรงตาม RBF และเมื่อ  $\nu = 0.5$  ฟังก์ชันเคอร์เนลจะอยู่ในรูปแบบอุดมคติคือเป็นเคอร์เนลเอกซ์โพเนนเชียลสมบูรณ์ โดย  $\Gamma(\cdot)$  คือฟังก์ชันแกมมา,  $d(\lambda_i, \lambda_j)$  คือระยะทาง Euclidean ระหว่างสองไฮเปอร์พารามิเตอร์, และ  $K_\nu(\cdot)$  คือฟังก์ชัน Bessel รูปแบบประยุกต์ โดยมีความน่าจะเป็นเป็นรากที่สองของความแปรปรวนดังแสดงในรูปที่ 2.8 Gaussian Process



รูปที่ 2.8 Gaussian Process

#### 2.4.2 การหาไฮเปอร์พารามิเตอร์ทดสอบที่ดีที่สุดที่เหมาะสมและการทดสอบ

สามารถหาไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุดได้จากไฮเปอร์พารามิเตอร์จุดเริ่มต้นได้โดยใช้ฟังก์ชันทางคณิตศาสตร์เรียกว่าฟังก์ชัน Expected Improvement (EI) เป็นฟังก์ชันที่จะทำการค้นหาไฮเปอร์พารามิเตอร์ทดสอบลำดับที่ดีที่สุดที่เหมาะสมโดยอ้างอิงจากฟังก์ชันเป้าหมาย และอยู่บนย่านไฮเปอร์พารามิเตอร์ที่สนใจ

$$EI(\lambda) \equiv \mathbb{E}[\max(f(\lambda_{best}), -f(\lambda), 0)]$$

$$= \begin{cases} (f(\lambda_{best}) - m(\lambda)) \Phi\left(\frac{f(\lambda_{best}) - m(\lambda)}{\sigma}\right) + \sigma \cdot \phi\left(\frac{f(\lambda_{best}) - m(\lambda)}{\sigma}\right) & \text{for } \sigma > 0 \\ 0 & \text{for } \sigma = 0 \end{cases} \quad (2.29)$$

พิจารณาสมการที่ 2.29 ฟังก์ชัน  $EI(\lambda)$  นั้นจะทำการหาไฮเปอร์พารามิเตอร์ใหม่  $\lambda^{NEW}$  เพื่อนำไปทดสอบ ในจุดที่ไฮเปอร์พารามิเตอร์จุดนั้นมีค่าคาดการณ์การพัฒนาสูงที่สุด (EI) ในแต่ละครั้งของการทดสอบ โดย  $f(\lambda_{best})$  คือค่าของฟังก์ชันเป้าหมายที่ดีที่สุด (มีไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด) ของทุกการทดสอบ ก่อนหน้า,  $\Phi(\cdot)$  คือ Cumulative Distribution Function (CDF),  $\phi(\cdot)$  คือ Probability Density Function (PDF), และ  $\sigma$  ค่าเบี่ยงเบนมาตรฐานของการแปรปรวนที่คาดเดาในข้อมูลที่สนใจ

เพื่อให้สามารถควบคุมคุณสมบัติของ ฟังก์ชัน EI ระหว่างคุณสมบัติการมุ่งเน้นในการหาไฮเปอร์พารามิเตอร์ กับคุณสมบัติในการเลือกใช้ไฮเปอร์พารามิเตอร์ที่ดีที่สุดได้ สามารถทำได้โดยการเพิ่มพารามิเตอร์  $\beta$  เข้าไปในสมการที่ 2.29 จะได้ว่า

$$EI(\lambda) \equiv \mathbb{E}[\max(f(\lambda_{best}) + \beta - f(\lambda), 0)]$$

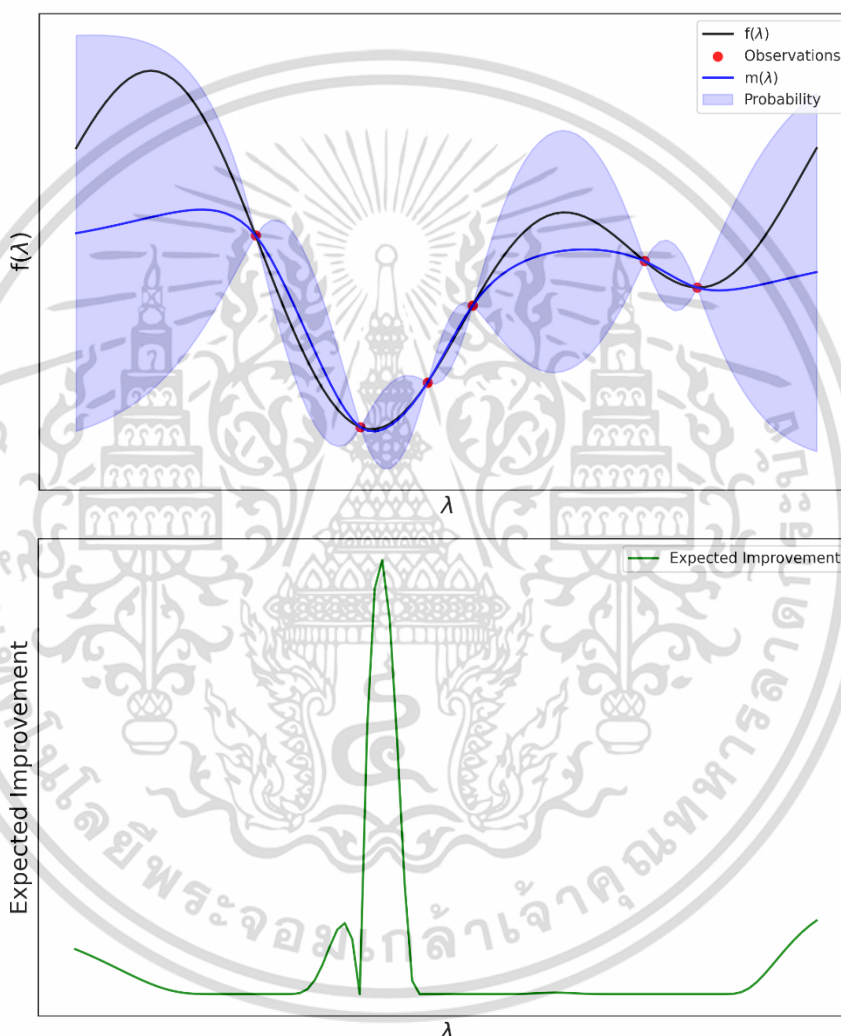
$$= \begin{cases} (f(\lambda_{best}) + \beta - m(\lambda)) \Phi\left(\frac{f(\lambda_{best}) + \beta - m(\lambda)}{\sigma}\right) + \sigma \cdot \phi\left(\frac{f(\lambda_{best}) + \beta - m(\lambda)}{\sigma}\right) & \text{for } \sigma > 0 \\ 0 & \text{for } \sigma = 0 \end{cases} \quad (2.30)$$

จากสมการที่ 2.30 จะเห็นได้ว่าเมื่อพารามิเตอร์  $\beta$  เพิ่มขึ้นจะทำให้อัลกอริทึมของการปรับแต่งแบบ Bayesian นั้นจะมุ่งเน้นในการหาไฮเปอร์พารามิเตอร์ใหม่ซึ่งหมายความว่า จะให้ความสำคัญกับการหาไฮเปอร์พารามิเตอร์ในย่านไฮเปอร์พารามิเตอร์ที่กำหนดเพื่อหา  $\lambda^{NEW}$  มากกว่าการใช้ค่าไฮเปอร์พารามิเตอร์ที่ดีที่สุดที่เคยทำการทดสอบมา

เมื่อพิจารณาลึกลงไป  $\mathbb{E}[\max(f(\lambda_{best}) + \beta - f(\lambda), 0)]$  ในสมการที่ 2.30 นั้นเป็นการทำให้แน่ใจว่ามีเพียงการพัฒนาเชิงบวกเท่านั้นหรือก็คือเมื่อ  $f(\lambda)$  มีค่าน้อยกว่า  $f(\lambda_{best}) + \beta$  ซึ่งในกรณีนี้เองที่กระบวนการเกาส์เซียน จะได้ปรับปรุงชุดไฮเปอร์พารามิเตอร์ทดสอบที่ฟังก์ชัน EI แนะนำ ( $\lambda^{NEW}$ ) ในขณะที่หาก  $f(\lambda)$  มีค่ามากกว่า  $f(\lambda_{best}) + \beta$  นั้นหมายความว่าไฮเปอร์พารามิเตอร์ที่ทดสอบในปัจจุบันนั้นไม่มีการพัฒนาในทางที่ดีจากไฮเปอร์พารามิเตอร์ที่ดีที่สุดในการทดสอบก่อนหน้านี้หรืออาจหมายความว่าค่า  $\beta$  นั้นให้ความสำคัญแก่การหาไฮเปอร์พารามิเตอร์ใหม่ๆมากกว่านั่นเอง อย่างไรก็ตามการปรับค่า  $\beta$  นั้นจะต้องพิจารณาผลลัพธ์ของฟังก์ชันเป้าหมายในรอบการทดสอบปัจจุบันนั้นต้องมีแนวโน้มที่ดีขึ้นซึ่งผู้พัฒนาอาจวิเคราะห์ได้ว่าการพัฒนานั้นอาจไม่เป็นความจริงจะต้องทำการปรับพารามิเตอร์นี้เพิ่มเพื่อลดปัจจัยนี้ลงนั่นเอง

ผลลัพธ์ของฟังก์ชัน EI นั้นเป็นค่าที่วัดความมั่นใจของ Gaussian Process นั้นหมายความว่ายิ่ง EI มีค่าที่สูงมากเท่าใดจะยังมีแนวโน้มที่จะมีการพัฒนาที่ดีมากยิ่งขึ้นของฟังก์ชันเป้าหมายมากกว่านั้น หรือ

อีกนัยหนึ่งคือ EI จะทำการแนะนำ Gaussian Process ให้มีการค้นหาและทดสอบเฉพาะชุดไฮเปอร์พารามิเตอร์เท่าที่จำเป็น ซึ่งโดยในสถานการณ์ปกติแล้ว EI จะมีแนวโน้มในการนำเสนอไฮเปอร์พารามิเตอร์ในจุดที่ EI มีค่าที่สูงซึ่งตำแหน่งนั้นจะตรงกับจุดที่มีความไม่แน่นอนสูงใน Gaussian Process สาเหตุที่เป็นเช่นนั้นเนื่องจากว่าตำแหน่งนั้นมีความเป็นไปได้ที่จะทำให้เกิดผลลัพธ์ของฟังก์ชันเป้าหมายที่ดีที่สุดได้นั่นเองดังแสดงความสัมพันธ์ระหว่าง Gaussian Process และ Expected Improvement ในรูปที่ 2.9 ความสัมพันธ์ระหว่าง Gaussian Process และ Expected Improvement



รูปที่ 2.9 ความสัมพันธ์ระหว่าง Gaussian Process และ Expected Improvement

$\beta$  เป็นพารามิเตอร์ที่ใช้ในการควบคุมลำดับความสำคัญของ EI ระหว่างการ Exploration ที่เป็นการหาไฮเปอร์พารามิเตอร์บนย่านของความไม่แน่นอน และ Exploitation ที่เป็นการหาและใช้ไฮเปอร์พารามิเตอร์บนจุดที่มีความแน่นอน โดย EI จะทำการ Exploration เมื่อพบเจอความไม่แน่นอนในผลลัพธ์ของฟังก์ชันเป้าหมายที่คาดเดาขึ้นมาและตัดสินใจให้ตำแหน่งนั้นมีความเป็นไปได้ที่จะทำให้เกิดการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันเป้าหมายไปในทางที่ดีขึ้นได้ ดังนั้นแล้วกระบวนการ Exploration ในย่านที่สนใจนั้นจะนำไปสู่การค้นพบไฮเปอร์พารามิเตอร์ที่ดีที่สุดได้

เมื่อ EI ให้ลำดับความสำคัญแก่การ Exploration ในแต่ละการทดสอบแล้ว EI จะแนะนำชุดของไฮเปอร์พารามิเตอร์ชุดใหม่เพื่อใช้ในการทดสอบ จากนั้นทำการทดสอบไฮเปอร์พารามิเตอร์นั้นด้วยฟังก์ชันการสูญเสีย เพื่อทำการปรับปรุงฟังก์ชันค่าเฉลี่ยและฟังก์ชันเคอร์เนลของฟังก์ชันเป้าหมายโดยใช้ไฮเปอร์พารามิเตอร์ที่ทดสอบดังกล่าว

หาก EI ให้ลำดับความสำคัญแก่การ Exploitation ในแต่ละรอบการทดสอบแล้ว โมเดล Gaussian Process จะใช้ชุดไฮเปอร์พารามิเตอร์ที่ดีที่สุดจากชุดไฮเปอร์พารามิเตอร์ที่เคยทำการทดสอบมาแล้ว ซึ่งในกรณีนี้เองฟังก์ชันค่าเฉลี่ยของฟังก์ชันเป้าหมายจะไม่ได้รับการปรับปรุงเนื่องมาจาก EI ให้ความสำคัญกับการหาไฮเปอร์พารามิเตอร์ในชุดที่ได้จากการทดสอบก่อนหน้านี้มากกว่าการให้ความสำคัญในการค้นหาไฮเปอร์พารามิเตอร์ใหม่ๆในย่านอื่นๆ

อย่างไรก็ตามในท้ายที่สุดแล้ว EI จะค้นพบไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุดในขั้นตอนสุดท้ายของการหาค่าของฟังก์ชันเป้าหมายที่ดีที่สุดของ Gaussian Process จากนั้นพารามิเตอร์ที่ดีที่สุดและไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุดจะถูกนำไปใช้งานต่อไป โดย EI นั้นจะอยู่ใน Exploitation เมื่อขั้นตอนการทดสอบจบลงเนื่องจากความเป็นไปได้ของชุดไฮเปอร์พารามิเตอร์ที่ดีกว่านั้นจะลดลงจนกระทั่งช่วงของการทดสอบจบลง

### 2.4.3 การปรับปรุงพารามิเตอร์ Gaussian Process

ในขณะที่อยู่ในช่วงของการทดสอบไฮเปอร์พารามิเตอร์นั้น EI จะอยู่ใน Exploration mode ซึ่งมีการเปลี่ยนแปลงค่าเฉลี่ยผลลัพธ์ของฟังก์ชันเป้าหมายและมีการปรับปรุง Gaussian Process ด้วย ซึ่งในขั้นตอนนี้จะมีการใช้ทฤษฎีบทของ Bayes ในการสังเคราะห์ปรับปรุงฟังก์ชันจากข้อมูลที่ได้รับ  $\mathbf{y}$  โดยฟังก์ชันเป้าหมาย  $f(\boldsymbol{\lambda})$  จะได้รับการปรับปรุงโดยไฮเปอร์พารามิเตอร์ใหม่ที่ได้จาก EI ในขั้นตอนนี้ก่อนหน้าและเปลี่ยนเป็น  $f(\boldsymbol{\lambda}^{new})$  โดยอ้างอิงจากฟังก์ชันค่าเฉลี่ยใหม่  $m^*(\boldsymbol{\lambda}^{new})$  และฟังก์ชันเคอร์เนลใหม่  $K^*(\boldsymbol{\lambda}^{new}, \boldsymbol{\lambda}^{new})$  ดังนั้นแล้วการคาดเดาการแจกแจง Gaussian Process จะถูกคำนวณเพื่อคาดเดาผลลัพธ์ของฟังก์ชันเป้าหมายที่แท้จริง ซึ่งสามารถคำนวณได้จากสมการดังต่อไปนี้

$$f(\boldsymbol{\lambda}^{new})|\mathbf{y} \sim N(m^*(\boldsymbol{\lambda}^{new}), K^*(\boldsymbol{\lambda}^{new}, \boldsymbol{\lambda}^{new})) \quad (2.31)$$

และสามารถคาดเดาผลลัพธ์ของฟังก์ชันค่าเฉลี่ยได้จากการแทนค่า  $\boldsymbol{\lambda}^{new}$  เป็นสมการที่ 2.32

$$m^*(\boldsymbol{\lambda}^{new}) = K(\boldsymbol{\lambda}^{new}, \boldsymbol{\lambda})(K(\boldsymbol{\lambda}, \boldsymbol{\lambda}) + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \quad (2.32)$$

โดย  $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$  แสดงถึงชุดข้อมูลที่สนใจ,  $\mathbf{I}$  คือ Identity Matrix และ  $\sigma^2$  แสดงถึงพารามิเตอร์ที่ถูกเพิ่มเข้าไปในเมทริกซ์เคอร์เนลในระหว่างการปรับแต่งโมเดลตามจำนวนของการแปรปรวนที่คาดการณ์ซึ่งอยู่ในช่วงประสิทธิภาพที่สนใจ ในแต่ละ  $y_i$  แสดงถึงปริมาณสเกลาร์ที่สัมพันธ์กับข้อมูลที่สนใจซึ่งโดยปกติคือ Loss ใช้ในการปรับปรุงพารามิเตอร์ของการแจกแจง Gaussian Process โดยสามารถคำนวณ  $y_i$  จากชุดไฮเปอร์พารามิเตอร์  $\lambda_i^{new} = [\lambda_1, \lambda_2, \dots, \lambda_i^{new}, \dots, \lambda_n]^T$  ของ  $\lambda^{new} = [\lambda_1^{new}, \lambda_2^{new}, \dots, \lambda_i^{new}, \dots, \lambda_n^{new}]^T$  เมื่อกำหนดให้ไฮเปอร์พารามิเตอร์อื่นนอกเหนือจาก  $\lambda_i^{new}$  นั้นมาจากการทดสอบครั้งก่อนหน้า กระบวนการหาค่า Loss นี้จะซ้ำไปจนกว่าจะครบทุกตัว ฟังก์ชันการคาดเดาเคอร์เนลที่  $\lambda^{new}$  จะเป็น

$$K^*(\lambda^{new}, \lambda^{new}) = K(\lambda^{new}, \lambda^{new}) - K(\lambda^{new}, \lambda)(K(\lambda, \lambda) + \sigma^2 \mathbf{I})^{-1} K(\lambda, \lambda^*) \quad (2.33)$$

โดยกำหนดให้  $\lambda$  คือชุดไฮเปอร์พารามิเตอร์ที่ใช้ในการเรียนรู้ และ  $\lambda^{new}$  คือชุดไฮเปอร์พารามิเตอร์ที่ทดสอบ ชุดไฮเปอร์พารามิเตอร์ที่ใช้ในการเรียนรู้คือไฮเปอร์พารามิเตอร์ที่ใช้ในการหาค่า Loss ในช่วงของการทดสอบก่อนหน้า และชุดไฮเปอร์พารามิเตอร์ทดสอบประกอบไปด้วยไฮเปอร์พารามิเตอร์ที่ถูกแนะนำโดย EI เพื่อใช้ในการคำนวณ Loss ในช่วงการทดสอบปัจจุบัน

## บทที่ 3

# ความสำคัญของการออกแบบระบบควบคุมแบบอัจฉริยะ

ในบทนี้จะกล่าวถึงความสำคัญและแรงบันดาลใจในการออกแบบระบบควบคุมแบบอัจฉริยะ ICS ขึ้นมาซึ่งจะมีการกล่าวอ้างอิงเปรียบเทียบกับระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง

### 3.1 ปัญหาของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง

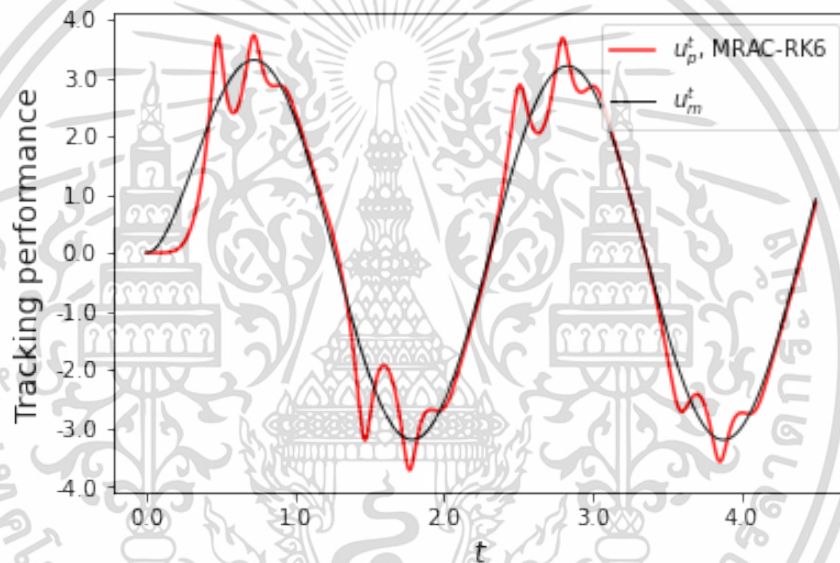
ปัญหาของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงในหลายๆกรณีนั้นมาจากความลำบากในการออกแบบที่ต้องใช้ความรู้ทางคณิตศาสตร์ในระดับที่สูงขึ้น MRAC ที่ถูกออกแบบมาเพื่อควบคุมกระบวนการที่มีลำดับอนุพันธ์ลำดับใดลำดับหนึ่งแล้วนั้นจะสามารถควบคุมกระบวนการที่มีลำดับอนุพันธ์นั้นได้อย่างมีประสิทธิภาพ แต่ทว่า MRAC นั้นจะไม่สามารถควบคุมกระบวนการที่มีลำดับอนุพันธ์ที่สูงกว่าการออกแบบได้ ดังเช่น MRAC ที่ออกแบบขึ้นโดยใช้ฟังก์ชัน Lyapunov ลำดับที่สอง (Duanyai et al., 2024) นั้นไม่สามารถที่จะควบคุมกระบวนการที่ไม่เป็นเชิงเส้นและมีการเปลี่ยนแปลงของสัญญาณอ้างอิงที่ไม่คงที่ได้ ซึ่งการจะทำให้ MRAC ดังกล่าวนั้นสามารถที่จะควบคุมกระบวนการที่ไม่เป็นเชิงเส้นได้นั้น จำเป็นที่จะต้องใช้ฟังก์ชัน Lyapunov ที่มีลำดับอนุพันธ์ที่สูงขึ้นตามไปด้วย ดังที่ได้แสดงให้เห็นในบทที่ 2.1.2 การออกแบบตัวควบคุมที่เป็นเพียงการออกแบบสำหรับกระบวนการลำดับที่หนึ่งแล้วนั้น กระบวนการออกแบบสำหรับกระบวนการที่มีลำดับอนุพันธ์ที่สูงขึ้นนั้นจำเป็นต้องใช้ความรู้ความเข้าใจทางคณิตศาสตร์ที่สูงขึ้น และมีความซับซ้อนและระยะเวลาในการออกแบบและพิสูจน์สูงขึ้นตามลำดับอนุพันธ์ดังกล่าว

โดย MRAC ในรูปแบบที่งานวิจัยนี้ได้นำมาเปรียบเทียบกับนั้นคือ MRAC รูปแบบพื้นฐานโดยใช้ฟังก์ชัน Lyapunov อนุพันธ์ลำดับที่สอง และใช้ Runge-Kuta ลำดับที่ 6 ในการหาคุนสมบัติของกระบวนการที่ไม่เป็นเชิงเส้น โดยแทนระบบควบคุมนี้ว่า RK6-MRAC สามารถออกแบบแผนผังการควบคุมได้เป็นดังรูปที่ 2.1 แผนภาพรูปแบบการควบคุมของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง

โดยวิธีการ Runge-Kuta order 6<sup>th</sup> หรือ RK6 คือวิธีการที่ใช้ในการประมาณค่าผลลัพธ์ของสมการเชิงอนุพันธ์แบบ IVP โดยตัวเลขลำดับของชื่อวิธีการนั้นเป็นตัวเลขบ่งชี้ถึงระดับความละเอียดของการประมาณการซึ่งลำดับที่ 6 นั้นมีความละเอียดในระดับที่สูงกว่าระดับที่ 4 และเป็นที่ยอมรับว่ายิ่งระดับของวิธีการนั้นมีค่าสูงขึ้นเท่าใด ยิ่งจะต้องใช้ทรัพยากรในการประมวลผลมากขึ้นเท่านั้น วิธีการ RK นั้นได้ถูกนำไปใช้อย่างกว้างขวางเพื่อแก้ปัญหาดังกล่าว โดยในงานวิจัยนี้ได้นำ RK6 มาใช้งานดังที่ได้กล่าวไป

### 3.2 การสร้าง MRAC-RK6 เพื่อใช้ในการเปรียบเทียบ

ในเบื้องต้นนั้นสามารถพิจารณาการออกแบบการควบคุมตาม รูปที่ 2.1 แผนภาพรูปแบบการควบคุมของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง ร่วมกับการออกแบบโดยใช้ฟังก์ชัน Lyapunov อนุพันธ์ลำดับที่สองดังที่กล่าวถึงในบทย่อที่ 2.1 Lyapunov-based MRAC การวิเคราะห์เสถียรภาพของกระบวนการที่มีสัญญาณเข้าและออกเชิงเดี่ยว โดยใช้กระบวนการที่ไม่คงที่ดังสมการที่ 2.3  $\dot{u}_p^t = u_p^t + (u_p^t)^2 + 3 \cdot u^t$  และสัญญาณรูปแบบอ้างอิงตามสมการที่ 2.6  $\dot{u}_m^t = -4 \cdot u_m^t + (16)\sin(3t)$  ตั้งแต่เวลา  $t=0$  ถึง  $t=4.5$  วินาทีสามารถพิจารณาผลลัพธ์ของการควบคุมได้ดัง รูปที่ 3.1 ผลตอบสนองของ RK6-MRAC



รูปที่ 3.1 ผลตอบสนองของ RK6-MRAC

$u_p^t$  เป็นผลลัพธ์ของกระบวนการ และ  $u_m^t$  เป็นสัญญาณอ้างอิงหรือเป้าหมายในการควบคุม ซึ่ง MRAC นั้นมีความพยายามควบคุมกระบวนการที่ไม่เป็นเชิงเส้นดังกล่าว แต่ทว่าไม่อาจที่จะสามารถกำจัดความผิดพลาดที่เกิดขึ้นได้อย่างมีประสิทธิภาพดังแสดงให้เห็นว่าการเปลี่ยนแปลงของกระบวนการนั้นไม่ทับซ้อนเข้ากับสัญญาณอ้างอิงโดยเฉพาะในช่วงแรกของการควบคุม ซึ่งปัญหาที่เกิดขึ้นนี้เกิดขึ้นตั้งแต่ขั้นตอนการออกแบบได้โดยการพิจารณา Adaptation Laws ที่ได้จากการออกแบบดังสมการที่ 2.15 ในบทย่อที่ 2.1 จะสังเกตเห็นได้ว่าตัวควบคุมนั้นมีพารามิเตอร์เพียงตัวเดียวที่ใช้ในการควบคุมการปรับเปลี่ยนพารามิเตอร์ป้อนกลับภายในคือ  $\gamma$  ซึ่งตามลักษณะเชิงคณิตศาสตร์แล้วการเปลี่ยนแปลงของพารามิเตอร์ภายในที่มีเพียงพารามิเตอร์เดียวแล้วนั้นจะมีลักษณะเป็นเชิงเส้นซึ่งไม่อาจทำให้เกิดการเปลี่ยนแปลงพารามิเตอร์ได้อย่างยืดหยุ่น ซึ่งปัญหานี้สามารถแก้ได้จากขั้นตอนการออกแบบโดยการ

เพิ่มตัวแปรให้กับ Adaptation Law เพื่อให้สามารถรองรับการเปลี่ยนของกระบวนการที่ไม่คงที่ได้ แต่ด้วยเหตุผลเชิงคณิตศาสตร์ที่ได้กล่าวข้างต้นนั่นเอง การจะออกแบบตัวควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงสำหรับกระบวนการที่ไม่เป็นเชิงเส้นนั้นมีความซับซ้อนเป็นอย่างมาก

### 3.3 ความสำคัญของ ICS

ตัวควบคุมอัจฉริยะ (Intelligent Controller) ที่ใช้หลักการของ Multi-Layer Perceptron Neural Network ในการควบคุมกลับมีวิธีการออกแบบและวิธีการในการแก้ปัญหาที่ง่ายกว่า MRAC และมีแนวโน้มที่จะมีประสิทธิภาพที่ดีกว่า อันเป็นเหตุผลเนื่องมาจากการพัฒนาในวิทยาการเครือข่ายประสาทเทียมอย่างก้าวกระโดดในปัจจุบัน ทั้งในแง่ของอัลกอริทึมที่มีวิธีการที่ใหม่กว่าในการพัฒนาการปรับตัวของตัวควบคุมให้ดีขึ้น และแม่นยำมากยิ่งขึ้น ทั้งในแง่ของอุปกรณ์ที่ใช้งานที่มีทั้งพลังและขีดความสามารถที่มากขึ้น และในแง่ของคุณสมบัติของตัวเครือข่ายประสาทเองที่มีจำนวนพารามิเตอร์หรือจำนวนเซลล์ประสาทจำนวนมากกว่า MRAC เป็นอย่างมาก ทั้งยังกำหนดจำนวนพารามิเตอร์ได้ทั้งแนวกว้างและแนวลึก ส่งผลให้สามารถออกแบบประสิทธิภาพในการควบคุมได้อย่างอิสระ สามารถรองรับการปรับตัวพารามิเตอร์ได้อย่างยืดหยุ่นและมีประสิทธิภาพ ประกอบกับความสามารถของเครือข่ายประสาทที่สามารถปรับปรุงพารามิเตอร์ตัวเองได้ในทันทีจากการรับรู้การเปลี่ยนแปลงความคลาดเคลื่อนของกระบวนการเพียงเล็กน้อย ด้วยเหตุผลเหล่านี้จึงเป็นเหตุผลสำคัญที่ทำให้เกิดการพัฒนาระบบควบคุมอัจฉริยะนั่นเอง

## บทที่ 4

# การออกแบบระบบควบคุมแบบอัจฉริยะโดยใช้ MLPINN ร่วมกับ กลไกการเรียนรู้แบบเมต้าเชิงเหตุการณ์ และตัวควบคุมเครือข่าย ประสาทที่สามารถปรับปรุงตัวเองได้

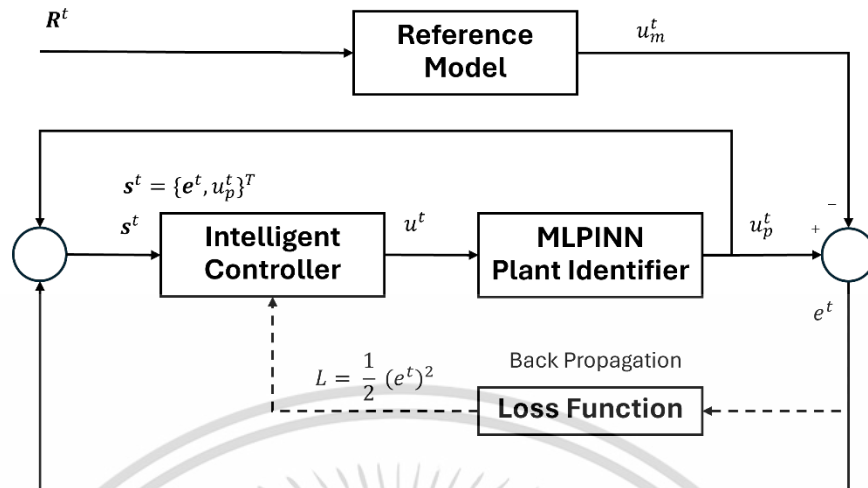
ในบทนี้จะกล่าวถึงขั้นตอนและทฤษฎีในการออกแบบ ระบบควบคุมแบบอัจฉริยะ (Intelligent Control System: ICS) โดยจะเริ่มตั้งแต่การพัฒนา MLPINN เพื่อนำมาใช้ในการหาคุณสมบัติของกระบวนการ ไปจนถึงการพัฒนาขีดความสามารถของ ICS ให้ดียิ่งขึ้นด้วยวิธีการปรับแต่งไฮเพอร์พารามิเตอร์

การออกแบบนั้นจะแบ่งออกเป็น 2 ส่วนหลักคือ

1. การออกแบบ MLPINN
  2. การออกแบบ Intelligent Controller
- โดยมีขั้นตอนในการดำเนินการดังนี้
1. ศึกษาทฤษฎีที่เกี่ยวข้องดังแสดงในบทที่ 2
  2. ออกแบบแผนภาพการควบคุมของ ICS
  3. สร้าง MLPINN เพื่อใช้ในการวิจัย
  4. สร้าง Intelligent Controller
  5. ปรับปรุง Intelligent Controller ด้วย HPO

### 4.1 การออกแบบแผนภาพการควบคุมของ ICS

ระบบควบคุมอัจฉริยะนั้นใช้หลักการออกแบบโดยอ้างอิงจากระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง และตัวควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิงโดยใช้เครือข่ายประสาทร่วม ดังที่กล่าวในบทที่ 2.1 และบทที่ 3 ดังนั้นแล้วเป้าหมายและหลักการของ ICS จึงมีลักษณะที่คล้ายคลึงกับสองระบบควบคุมดังกล่าว กล่าวคือระบบควบคุมแบบอัจฉริยะมีเป้าหมายในการกำจัดความคลาดเคลื่อน  $e^t$  ของกระบวนการที่ไม่เป็นเชิงเส้นโดยเปรียบเทียบกับสัญญาณอ้างอิงจากรูปแบบอ้างอิงและนำความคลาดเคลื่อนดังกล่าวมาปรับปรุงตัวแปรภายในของตัวควบคุม ซึ่งต้องอยู่บนเงื่อนไขของความเสถียรและความสามารถในการควบคุมกระบวนการที่ไม่เป็นเชิงเส้นได้อย่างสมบูรณ์ โดยสามารถออกแบบแผนภาพการควบคุมได้ตามแนวคิดดังกล่าวได้ดังรูปที่ 4.1 แผนภาพการควบคุมของระบบควบคุมแบบอัจฉริยะ



รูปที่ 4.1 แผนภาพการควบคุมของระบบควบคุมแบบอัจฉริยะ

จากรูปจะเห็นได้ว่าตัวควบคุมอัจฉริยะจะมีข้อแตกต่างจาก RK6-MRAC (รูปที่ 2.1 แผนภาพรูปแบบการควบคุมของระบบควบคุมแบบปรับ) ในการที่ตัวควบคุมจะไม่ได้รับสัญญาณขาเข้าโดยตรง ซึ่งข้อดีอีกอย่างหนึ่งของตัวควบคุมที่ใช้เครือข่ายประสาทเทียมในการออกแบบ ที่สถาปัตยกรรมของเครือข่ายประสาทเทียมสามารถออกแบบสัญญาณที่รับเข้ามาได้ โดยตัวควบคุมอัจฉริยะจะใช้สัญญาณรับเข้าในรูปแบบของเมทริกซ์  $s^t$  ประกอบด้วยผลลัพธ์ของกระบวนการ  $u_p^t$  และความคลาดเคลื่อนของการควบคุมเทียบกับสัญญาณอ้างอิง  $e^t$  โดยมีการปรับพารามิเตอร์  $\theta$  จากกระบวนการ Back Propagation ของเครือข่ายประสาทเทียมตามการเปลี่ยนแปลงของการสูญเสียเทียบกับพารามิเตอร์  $\frac{\partial L}{\partial \theta}$  ซึ่งหาได้จากฟังก์ชันการสูญเสียดังสมการที่ 4.1

$$L = \frac{1}{2} (e^t)^2 \quad (4.1)$$

ซึ่งหมายความว่า การควบคุมกระบวนการจะไม่ได้อ้างอิงจากสัญญาณรับเข้าโดยตรงแต่จะใช้วิธีการในการตรวจสอบความคลาดเคลื่อนของการควบคุมและผลลัพธ์ของกระบวนการที่ไม่เป็นเชิงเส้นในการคาดเดาสัญญาณควบคุม ( $u^t$ ) เพื่อให้ผลลัพธ์เป็นไปตามลักษณะของสัญญาณอ้างอิง

การออกแบบการควบคุมโดยใช้เครือข่ายประสาทเทียมนั้นสามารถเริ่มการออกแบบโดยอ้างอิงจากสมการที่ 2.17 และสมการที่ 2.19 อาจพิจารณาได้ว่าสัญญาณควบคุมกระบวนการ  $u^t$  ที่ได้จากการคาดการณ์ของเครือข่ายประสาทเทียมนั้นคือผลลัพธ์ของผลรวมของพารามิเตอร์ภายใน  $\theta$  ที่ขึ้นอยู่กับสัญญาณที่รับเข้ามาซึ่งในกรณีนี้คือ  $s^t$  ดังแสดงในสมการที่ 4.2

$$f_{\theta}(s^t) = u^t \quad (4.2)$$

จากสมการที่ 3.2 จะแสดงถึงวิธีที่เครือข่ายประสาทใช้ในการคาดเดาผลลัพธ์ของกระบวนการ ซึ่งเป็นหนึ่งในคุณสมบัติของเครือข่ายประสาทเทียมที่สามารถทำได้ โดยในการออกแบบนี้จะใช้อีกหนึ่งคุณสมบัติของเครือข่ายประสาทเทียมด้วยเช่นเดียวกันนั่นก็คือความสามารถในการเรียนรู้โดยการปรับพารามิเตอร์ภายในจากการรับรู้การสูญเสียที่เกิดขึ้นหรือกระบวนการ Forward propagation และ Backward propagation โดยพิจารณาร่วมกับทฤษฎีการควบคุมแบบปรับตัวจากบทที่ 2.1 พิจารณาสมการที่ 2.1 และ 2.4 ที่เป็นสมการเชิงอนุพันธ์ตัวแทนของกระบวนการที่ไม่เป็นเชิงเส้นและรูปแบบอ้างอิงตามลำดับร่วมกับการปรับพารามิเตอร์ด้วยอัลกอริทึมแบบ SGD ดังสมการที่ 2.18 จะได้เป็นสมการที่ 4.3

$$\begin{aligned}\theta_{ij} &= \theta_{ij} - \alpha \frac{\partial \frac{1}{2}(u_m^t - u_p^t)^2}{\partial \theta_{ij}} \\ \theta_{ij} &= \theta_{ij} - \alpha \frac{\partial \frac{1}{2}((-a_m \cdot u_m^t - c_m \cdot f_m^t + b_m \cdot r^t) - (-a_p \cdot u_p^t - c_p \cdot f_p^t + b_p \cdot u^t))^2}{\partial \theta_{ij}} \\ \theta_{ij} &= \theta_{ij} - \alpha \frac{\partial \frac{1}{2}((-a_m \cdot u_m^t - c_m \cdot f_m^t + b_m \cdot r^t) - (-a_p \cdot u_p^t - c_p \cdot f_p^t + b_p \cdot f(s^t, \theta_{ij})))^2}{\partial \theta_{ij}}\end{aligned}\quad (4.3)$$

ซึ่งแสดงการปรับเปลี่ยนพารามิเตอร์แต่ละตัวสัมพันธ์กับการเปลี่ยนแปลงค่าการสูญเสียเทียบกับพารามิเตอร์แต่ละตัวที่เปลี่ยนแปลงไป

## 4.2 การสร้าง MLPINN เพื่อใช้ในการหาคูณสมบัติของกระบวนการ

ในบทย่อนี้ จะกล่าวถึงกระบวนการสร้าง MLPINN ที่เป็นหนึ่งในฟังก์ชันที่สำคัญของระบบควบคุมแบบอัจฉริยะ สามารถหาคูณสมบัติของกระบวนการที่มีทั้งความไม่เป็นเชิงเส้นและถูกควบคุมด้วยสัญญาณควบคุมที่ได้จากตัวควบคุมอัจฉริยะอีกทีหนึ่งโดยอ้างอิงจากงานวิจัย Event-triggered MRAC System Design for SISO Plants Using Meta-tuning-based Physics-informed Neural Networks and Transfer Learning without Labeled Data (Duanyai et al., 2024) ซึ่งได้อธิบายถึงทฤษฎีไปแล้วในบทที่ 2 งานวิจัยที่เกี่ยวข้อง

การสร้าง MLPINN และนำมาใช้งานในระบบควบคุมแบบอัจฉริยะนั้นมีอยู่ด้วยกัน 4 ขั้นตอน ซึ่งประกอบไปด้วย

1. กำหนดกระบวนการที่ไม่เป็นเชิงเส้นในรูปแบบของสมการเชิงอนุพันธ์
2. การสร้าง Physical Informed Neural Network (PINN)
3. สร้างการเรียนรู้แบบเมตาเข้ากับ PINN ในส่วนที่ 2
4. สร้างกลไกการเรียนรู้แบบเมตาเชิงเหตุการณ์สำหรับ MLPINN

#### 4.2.1 กำหนดกระบวนการที่ไม่เป็นเชิงเส้นในรูปแบบของสมการเชิงอนุพันธ์

ในกระบวนการขั้นตอนี้มีความสำคัญเป็นอย่างมากในแง่ของการกำหนดแนวทางของการวิจัยและพัฒนา เนื่องมาจากกระบวนการนี้ผู้วิจัยจะต้องทำการเลือกลักษณะของกระบวนการที่มีลักษณะไม่เป็นเชิงเส้นและมีความไม่เสถียร หากกระบวนการที่เลือกนั้นมีความไม่เป็นเชิงเส้นน้อยเกินไปจะทำให้ไม่สามารถแสดงประสิทธิภาพของตัวควบคุมได้เป็นที่ประจักษ์ และหากมีความซับซ้อนสูงเกินไปจะทำให้ยากต่อการเข้าใจได้

การสร้างสมการเชิงอนุพันธ์ของกระบวนการนั้นสามารถสร้างได้จากสมการที่ 2.1 อันเป็นสมการพื้นฐานของสมการเชิงอนุพันธ์ ซึ่งสามารถแทนค่าคงที่  $a_p, b_p, c_p$  และฟังก์ชันความแปรปรวนของกระบวนการ  $f_p^t$  ได้อย่างอิสระ และมี  $u^t$  เป็นสัญญาณควบคุมกระบวนการโดยกำหนดให้

$$a_p = -1, b_p = 3, c_p = -1, f_p^t = (u_p^t)^2; u_p^0 = 0$$

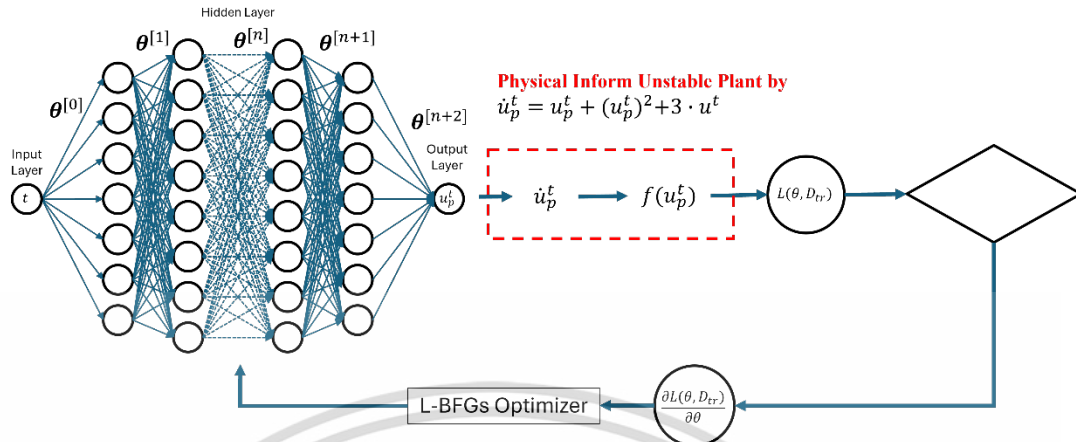
นำพารามิเตอร์ข้างต้นไปแทนที่ในสมการที่ 2.1 จะได้สมการที่ 2.3 ซึ่งมีลักษณะของกระบวนการที่เมื่อ  $u_p^t \neq 0$ ; จะทำให้  $u_p^t \rightarrow \pm\infty$  หากไม่ทำการควบคุมกระบวนการ โดยจะใช้คุณสมบัติกระบวนการเดียวกันนี้ในการเปรียบเทียบ MRAC และ Intelligent Control System

#### 4.2.2 การสร้าง Physical Informed Neural Network

ในขั้นตอนนี้จะเป็นการสร้าง PINN แบบ IVP ในรูปแบบพื้นฐานโดยใช้สมการเชิงอนุพันธ์ที่ได้จากกระบวนการที่ 1

โดยปกติแล้วในกระบวนการเรียนรู้ของ Neural Network นั้นจะใช้ชุดข้อมูลในการเรียนรู้โดยเปรียบเทียบผลลัพธ์ที่ได้จากการคาดเดาและข้อมูลที่แท้จริงในการเปรียบเทียบความคลาดเคลื่อน ซึ่งข้อมูลจริงที่ใช้ในการเปรียบเทียบนั้นมนุษย์จะเป็นผู้กำหนดโดยเครือข่ายประสาทจะมีแนวโน้มที่ความแม่นยำมากขึ้นเมื่อข้อมูลชุดนี้มีจำนวนมาก แต่ในกรณีของ PINN จะใช้การบอกลักษณะเชิงคณิตศาสตร์ในการหาค่าจริงเหล่านั้นซึ่งรองรับตั้งแต่สมการเชิงอนุพันธ์ลำดับที่หนึ่ง ไปจนถึงสมการเชิงอนุพันธ์ลำดับที่สูงกว่า โดยที่มีความซับซ้อนในการออกแบบที่อยู่ในระดับที่สามารถทำความเข้าใจได้โดยง่าย โดยไม่จำเป็นที่จะต้องมีการระบุข้อมูลเป็นจำนวนมากเหมือนรูปแบบเครือข่ายประสาทเทียมแบบทั่วไป

ซึ่ง PINN ในระบบควบคุมแบบอัจฉริยะนั้นสามารถแบ่งรูปแบบการทำงานออกเป็นสองรูปแบบคือรูปแบบการหาคุณสมบัติกระบวนการ และรูปแบบเรียนรู้ ดังแสดงใน รูปที่ 4.2 Physical Inform Neural Network รูปแบบเรียนรู้



รูปที่ 4.2 Physical Inform Neural Network รูปแบบเรียนรู้

จะเห็นได้ว่ารูปแบบที่ใช้ในการหาคูณสมบัติการเรียนรู้นั้นมีสถาปัตยกรรมแบบเครือข่ายประสาทเทียมเพอร์เซพตรอนหลายชั้นแบบทั่วไป แต่สิ่งที่ทำให้ PINN มีความพิเศษคือรูปแบบการเรียนรู้ที่ได้นำสมการที่ 2.3 ผสมเข้ากับการเรียนรู้ และปรับปรุงพารามิเตอร์ด้วย L-BFGs Optimizer และใช้การสุ่มเวลาตลอดช่วงเวลาที่สนใจของกระบวนการเป็นสัญญาณรับเข้าของเครือข่ายประสาท และเมื่อการเรียนรู้เสร็จสิ้น เครือข่ายประสาทเทียมจะสามารถหาคูณสมบัติของกระบวนการได้ตลอดช่วงเวลาที่สนใจ

โดยต่อไปนี้จะกล่าวถึงการกำหนดพารามิเตอร์ต่างๆ และวิธีการพัฒนาที่ใช้ในการทดลอง ซึ่งงานวิจัยนี้ได้ใช้เทคนิคการโปรแกรมมภาษา Python และ Library ของ Tensorflow และ Keras เป็นหลักในการพัฒนา ผลของการทดลองอาจเปลี่ยนแปลงเล็กน้อยไปตามแต่ละเวอร์ชันของไลบรารีดังกล่าว รวมไปถึงฮาร์ดแวร์การประมวลผลที่ใช้

1. กำหนดขนาดของสัญญาณรับเข้าและขนาดของสัญญาณผลลัพธ์ คือ 1 เนื่องจากเป็นกระบวนการแบบ SIS0
2. กำหนดจำนวนของ Hidden Layer ซึ่งส่งผลต่อความสามารถในการปรับตัวเข้ากับกระบวนการที่มีความซับซ้อน
3. กำหนดจำนวนของเซลล์ประสาทเทียมในแต่ละ Hidden Layer
4. กำหนดอัลกอริทึมที่ใช้ในการสุ่มพารามิเตอร์ตั้งต้น
5. กำหนดฟังก์ชัน Activation เป็น Hyperbolic Tangent ซึ่งเป็นฟังก์ชันที่ PINN ใช้งานได้อย่างมีประสิทธิภาพ
6. ทำการสุ่มเวลา (สัญญาณรับเข้า) ตั้งแต่  $t=0$  จนถึง  $t_{end}$  ซึ่งขึ้นอยู่กับช่วงเวลาที่สนใจ จำนวนในการสุ่มนั้นส่งผลต่อความแม่นยำของ PINN โดยตรง
7. เริ่มการทำ Forward Propagation ผ่านเครือข่ายประสาทเทียม โดยใช้คูณสมบัติการหาการเปลี่ยนแปลงของกระบวนการเทียบกับหน่วยเวลาด้วยความสามารถของเครือข่ายประสาท ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบกับสมการการเปลี่ยนแปลงของกระบวนการที่ 2.3 โดยระบุข้อมูลเริ่มต้นของกระบวนการเป็น 0 ตามเงื่อนไขของกระบวนการที่ไม่คงที่

8. เปรียบเทียบค่าหาเกรเดียนต์ที่เกิดขึ้นจากการสูญเสียที่เกิดขึ้นในกระบวนการ Forward propagation เพื่อนำไปปรับพารามิเตอร์โดยใช้ L-BFGs Optimizer ในกระบวนการ Backward propagation
9. ทำซ้ำตั้งแต่กระบวนการที่ 7-8 จนกระทั่งเกรเดียนต์ที่เกิดขึ้นมีค่าน้อยเกินกว่าค่าที่ยอมรับได้ หรือมีจำนวนการทำซ้ำที่มากกว่าค่าที่กำหนดเอาไว้

ในแต่ละรอบการเรียนรู้ในตัวเครือข่ายประสาทจะทำการปรับตัวเพื่อให้ผลลัพธ์ที่ได้มีการเปลี่ยนแปลงในรูปแบบเดียวกับสมการที่ 2.3 แต่ด้วยข้อจำกัดของ PINN ที่อัลกอริทึมในการหาค่าตั้งต้นของพารามิเตอร์นั้นเกิดความคลาดเคลื่อนจากความเป็นจริงเล็กน้อยในช่วงเริ่มต้น และเมื่อเกิดความผิดพลาดในช่วงเริ่มต้นนั้นทำให้การหาสมการเชิงอนุพันธ์นั้นจะเกิดขนาดความคลาดเคลื่อนสูงขึ้นสะสมตามระยะเวลาที่ผ่านไป ทำให้เมื่อเวลาผ่านไปผลลัพธ์ที่ได้จะมีความคลาดเคลื่อนจากความเป็นจริงสูง โดยจะใช้การเรียนรู้แบบเมต้าเข้าช่วยแก้ปัญหาในหัวข้อที่ 4.2.3

#### 4.2.3 การพัฒนา MLPINN

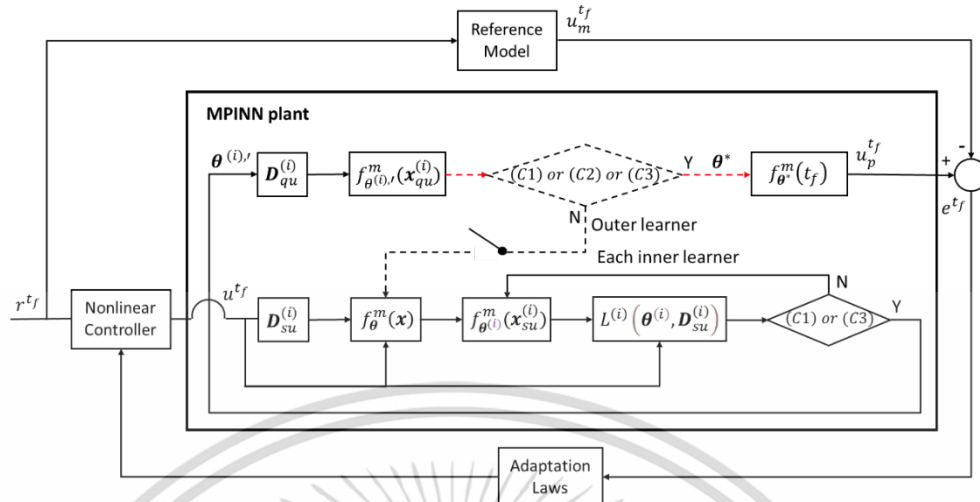
ดังอธิบายในบทที่ 2 การเรียนรู้แบบเมต้าเป็นวิธีการเรียนรู้ที่ใช้ในการพัฒนาเครือข่ายประสาทเทียมแบบต่างๆ โดยใช้หลักการของตัวเรียนรู้หลายตัวในการขยายความเป็นไปได้ในการเรียนรู้บนช่วงที่ไม่เคยปรากฏของเครือข่ายประสาทเทียม ซึ่งการวิจัยนี้ก็นำเอาวิธีการที่ว่ามีมาใช้เพื่อปรับปรุงการเรียนรู้ของ PINN ทำให้เกิดพารามิเตอร์ตั้งต้นที่เหมาะสมกับลักษณะงานของ PINN มากยิ่งขึ้น โดยสามารถพิจารณาอัลกอริทึมที่ใช้ในการสร้างการเรียนรู้แบบเมต้าสำหรับ PINN โดยอ้างอิงจากทฤษฎีในบทที่ 2 รูปที่ 2.5 แผนภาพกลไกการทำงานของ MLPINN (Duanyai, 2024) ได้ดังนี้

1. กำหนดจำนวน Task หรือจำนวนของตัวเรียนรู้ภายใน
2. สร้างชุดข้อมูลสำหรับ PINN ตามจำนวนของตัวเรียนรู้ภายใน โดยการสุ่มที่แตกต่างกันออกไปแต่ละชุด โดยมีจำนวนและช่วงของสัญญาณเหมือนกัน
3. ทำการตั้งค่าเริ่มต้นสำหรับพารามิเตอร์
4. สร้าง PINN ต้นแบบซึ่งแทนตัวเรียนรู้ภายนอกเรียกว่า MLPINN
5. ทำการสร้าง PINN ตัวเรียนรู้ภายในตามจำนวนของ Task โดยอ้างอิงพารามิเตอร์จากตัวเรียนรู้ภายนอก
6. เริ่มกระบวนการเรียนรู้ของ PINN ตัวเรียนรู้ภายในแต่ละตัวโดยใช้ชุดข้อมูลที่แตกต่างกันในขั้นตอนที่ 2

7. เมื่อตัวเรียนรู้ภายในลำดับที่  $n$  เสร็จสิ้นตามเงื่อนไขที่กำหนดไว้ จะทำการส่งพารามิเตอร์ที่ทำการเรียนรู้แล้วไปยังตัวเรียนรู้ลำดับ  $n+1$
  8. ทำซ้ำขั้นตอนที่ 6-7 จนกระทั่งถึงตัวเรียนรู้ภายในตัวสุดท้าย
  9. สร้างชุดข้อมูลใหม่เป็นจำนวนเท่ากับจำนวน Task เพื่อทำการหาค่าการสูญเสียที่เกิดขึ้น
  10. ปรับปรุงพารามิเตอร์ของตัวเรียนรู้ภายนอกโดยใช้พารามิเตอร์ที่ได้จากตัวเรียนรู้ภายในล่าสุด
  11. ใช้ชุดข้อมูลทุกชุดที่สร้างจากขั้นตอนที่ 9 ในการทดสอบหาค่าการสูญเสียที่เกิดขึ้น
  12. รวบรวมค่าการสูญเสียที่เกิดขึ้นเรียกค่าการสูญเสียใหม่นี้ว่า Meta-Loss
  13. ทำการหาเกรเดียนต์ของการเปลี่ยนของ Meta-Loss เทียบกับพารามิเตอร์ของตัวเรียนรู้ภายนอก
  14. นำเกรเดียนต์ที่ได้จากขั้นตอนที่ 13 นำไปปรับปรุงพารามิเตอร์ของตัวเรียนรู้ภายนอกโดยใช้ L-BFGs Optimizer
  15. ทำซ้ำตั้งแต่ขั้นตอนที่ 5-14 จนกว่าจะครบเงื่อนไขของกลไกการเรียนรู้แบบเมต้าเชิงเหตุการณ์
- เมื่อดำเนินการมาถึงขั้นตอนสุดท้ายพารามิเตอร์ของเครือข่ายประสาทเทียม MLPINN นั้นจะได้รับการปรับปรุงเป็นอย่างมากจนทำให้ประสิทธิภาพนั้นเพิ่มขึ้นอย่างก้าวกระโดด ขั้นตอนเหล่านี้เป็นขั้นตอนโดยสรุปของบทที่ 2 หากต้องการทราบในรายละเอียดของที่มาและทฤษฎีโปรดพิจารณาบทที่ 2.3 อย่างไม่รู้ดังที่แสดงให้เห็นตามขั้นตอนการทำงานของ MLPINN ดังกล่าวแล้ว จะเห็นได้ว่า MLPINN นั้นมีจำนวนรอบในการเรียนรู้ที่ใช้ระยะเวลานานมาก ดังนั้นแล้วจึงจำเป็นต้องมีการจำกัดระยะเวลาในการประมวลผลโดยการพัฒนากลไกการเรียนรู้แบบเมต้าเชิงเหตุการณ์เพื่อจำกัดจำนวนรอบในการเรียนรู้ดังกล่าว

#### 4.2.4 การพัฒนากลไกการเรียนรู้แบบเมต้าเชิงเหตุการณ์

กลไกการเรียนรู้แบบเมต้าเชิงเหตุการณ์คือกลไกหรือเงื่อนไขบางประการที่เมื่อเกิดเหตุการณ์หรือเงื่อนไขแล้วนั้นจะมีการสั่งการบางอย่างหรือปิดการทำงานของบางสิ่งบางอย่างลง ซึ่งเป็นวิธีการที่คิดค้นเพื่อจำกัดรอบการเรียนรู้ของ MLPINN โดยเฉพาะเพื่อไม่ให้กระบวนการการเรียนรู้ที่สูงมากจนเกินไป



รูปที่ 4.3 การทำงานกลไกการเรียนรู้แบบเมตาเชิงเหตุการณ์ใน MLPINN (Duanyai et al., 2024)

ดังแสดงทฤษฎีและรูปแบบในรูปที่ 4.3 การทำงานกลไกการเรียนรู้แบบเมตาเชิงเหตุการณ์ใน MLPINN (Duanyai et al., 2024) มาใช้งานอีกด้วย โดยในกระบวนการจะใช้กลไกทั้ง 3 C1 C2 และ C3 ในขั้นตอนที่ 15 ของหัวข้อย่อยที่ 4.2.3 เมื่อ (C1) คือการกำหนดรอบการเรียนรู้เมตานั้นถึงรอบการเรียนรู้ภายนอกสูงที่สุดแล้วหรือไม่  $iter = iter^{max}$ ; (C2) คือการกำหนดเงื่อนไขค่าการสูญเสียเมตานั้นมีค่าต่ำกว่าค่าที่ยอมรับได้หรือไม่  $L(\theta, D_{tr}) < \epsilon$  และ (C3) เงื่อนไขค่าต่ำที่สุดของการเปลี่ยนแปลงการสูญเสียที่เกิดขึ้นนั้นน้อยกว่าค่าคงที่ที่กำหนดหรือไม่  $\|\nabla L(\theta, D_{tr})\| < \epsilon^H$

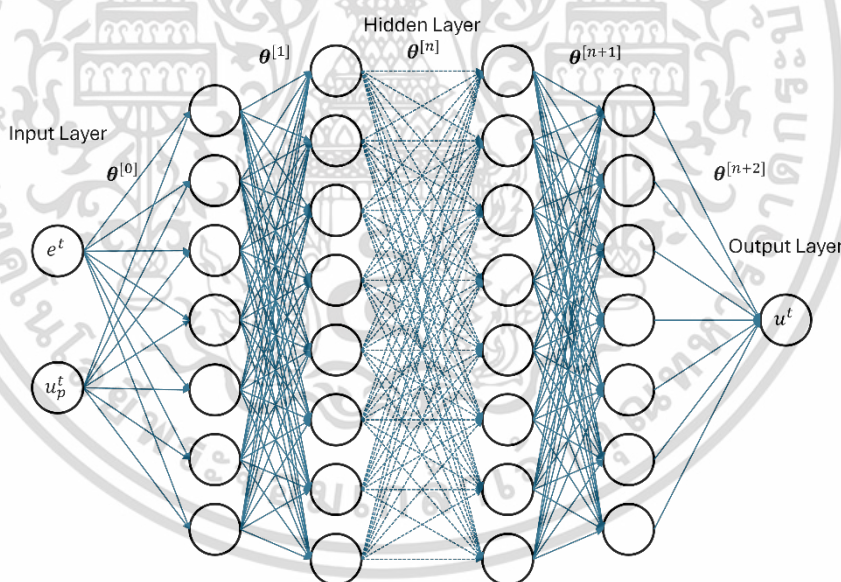
โดยกำหนดให้  $\epsilon = 8 \times 10^{-4}$  ซึ่งเป็นค่าคงที่ในเงื่อนไข C2 เนื่องมาจากค่าคงที่นี้ในงานวิจัยแสดงให้เห็นถึงการเริ่มมีความไม่เสถียรของการควบคุมแบบ MRAC ซึ่งเป็นตัวอย่างที่ดีในการเปรียบเทียบความสามารถการควบคุมกระบวนการที่ไม่เป็นเชิงเส้นระหว่าง ระบบควบคุมแบบอัจฉริยะ ICS และ Lyapunov-based MRAC ได้เป็นอย่างดีรวมถึงเป็นจุดสมดุลระหว่างระยะเวลาในการประมวลผลและประสิทธิภาพของ MLPINN

### 4.3 การสร้าง Intelligent Controller

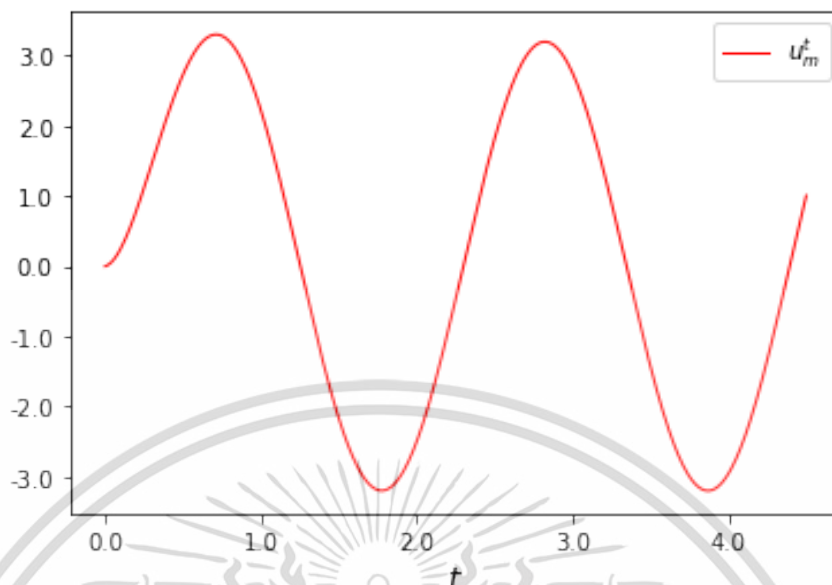
ตัวควบคุมอัจฉริยะนั้นใช้ความสามารถในการคาดเดาผลลัพธ์ของเครือข่ายประสาทเทียมในการควบคุมกระบวนการ โดยมีการปรับตัวด้วยกระบวนการ Back Propagation เพื่อให้สามารถปรับพารามิเตอร์ของตัวเองได้ด้วยวิธีการที่กล่าวไปในสมการที่ 4.3 ซึ่งเป็นส่วนทฤษฎีย่อยและหลักการโดยพื้นฐาน ในความเป็นจริงแล้วกระบวนการในการคาดเดาทั้ง Forward Propagation และกระบวนการในการปรับตัวอย่าง Backward Propagation นั้นสามารถพัฒนาได้โดยง่ายและไม่มีความซับซ้อนโดยใช้การเขียนโปรแกรมในภาษา Python และไลบรารี Tensorflow และ Keras เป็นหลักโดยสามารถพัฒนาตามแผนภาพการควบคุมที่ได้ออกแบบไว้ในรูปที่ 4.1 แผนภาพการควบคุมของระบบควบคุมแบบอัจฉริยะดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ทำการสร้างเมทริกซ์  $s^t$  ซึ่งสมาชิกเมทริกซ์ประกอบไปด้วยผลลัพธ์ของกระบวนการที่ได้จาก MLPINN  $u_p^t$  และ ค่าความคลาดเคลื่อน  $e^t$  ที่ได้จากการเปรียบเทียบ  $u_p^t$  กับสัญญาณอ้างอิง  $u_m^t$
2. สร้างเครือข่ายประสาทเทียมรูปแบบปกติที่มี Input Layer ในรูปแบบของ 2 สัญญาณรับเข้า และ Output Layer 1 สัญญาณขาออกเป็นสัญญาณผลลัพธ์ในการควบคุมกระบวนการ ดังแสดงในรูปที่ 4.4 สถาปัตยกรรมของตัวควบคุมอัจฉริยะ
3. ทำการกำหนดไฮเพอร์พารามิเตอร์ตั้งต้นได้แก่ จำนวนเซลล์ประสาทในแต่ละ Hidden Layer และอัตราการเรียนรู้ของ Optimizer
4. ทำการเลือกOptimizerที่ใช้ในการเรียนรู้
5. ทำการตั้งค่าเริ่มต้นให้พารามิเตอร์
6. ทำการควบคุมกระบวนการโดยใช้รูปแบบอ้างอิงจากสมการที่ 2.6  $\dot{u}_m^t = -4 \cdot u_m^t + (16)\sin(3t)$  ซึ่งมีลักษณะสัญญาณอ้างอิงดังรูปที่ 4.5 และกระบวนการที่ไม่เป็นเชิงเส้นจากสมการที่ 2.3 ตั้งแต่เวลา  $t=0s$  ถึง  $t=4.5s$  โดยกำหนดให้ช่วงเวลาย่อยห่างกัน  $0.1s$
7. เปรียบเทียบกับการควบคุมแบบ MRAC-RK6



รูปที่ 4.4 สถาปัตยกรรมของตัวควบคุมอัจฉริยะ



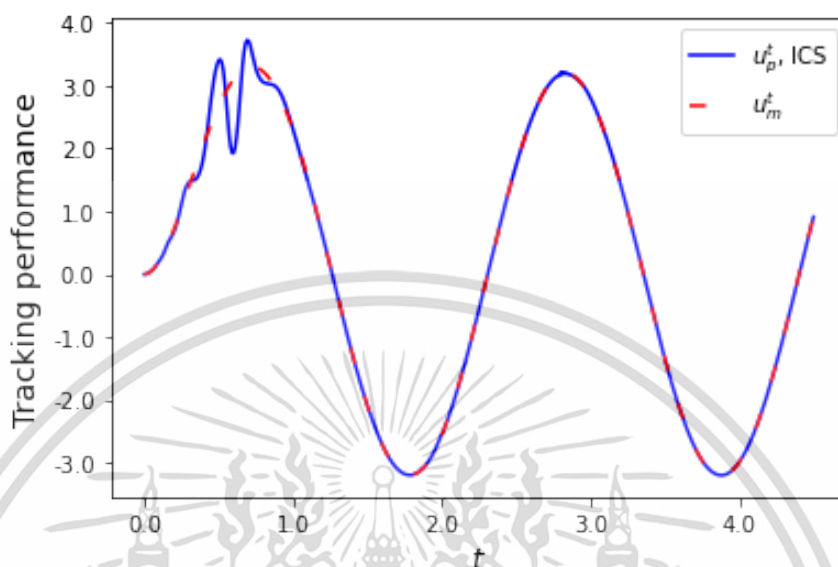
รูปที่ 4.5 สัญญาณอ้างอิงที่ใช้ในการเปรียบเทียบ

เครือข่ายประสาทแบบปกตินั้นจะเป็นประเภทที่ต้องมีการเรียนรู้แบบออฟไลน์ก่อนที่จะนำไปใช้งาน ซึ่งลักษณะเช่นนี้ไม่เหมาะต่อการใช้งานในเชิงการควบคุมอันเนื่องมาจากเครือข่ายรูปแบบปกติดำเนินต้องมีการเรียนรู้จากข้อมูลที่เคยมีมาก่อน ซึ่งทำให้ไม่สามารถตอบสนองต่อการเปลี่ยนแปลงสัญญาณรับเข้าที่ไม่เคยมีอยู่ได้ จึงทำให้การควบคุมนั้นไม่อาจควบคุมกระบวนการได้อย่างสมบูรณ์

จากวิธีการสร้างตัวควบคุมอัจฉริยะข้างต้นจะแสดงให้เห็นว่าเครือข่ายประสาทที่ใช้ในระบบควบคุมแบบอัจฉริยะจะเป็นการควบคุมและเรียนรู้แบบออนไลน์กล่าวคือใช้กระบวนการการคาดการณ์ของเครือข่ายประสาทเพื่อคาดการณ์สัญญาณควบคุม หลังจากนั้นนำสัญญาณไปควบคุมกระบวนการ และได้ผลลัพธ์ของกระบวนการจาก MLPINN แล้วนั้น ตัวควบคุมจะทำการติดตามความคลาดเคลื่อนที่เกิดขึ้นและทำการปรับพารามิเตอร์ตัวเองผ่านกระบวนการ Back Propagation ซึ่งในกระบวนการนี้ในกรณีของเครือข่ายประสาทที่ออฟไลน์นั้นอาจจะใช้ระยะเวลานานเนื่องจากมีจำนวนข้อมูลมากในการเรียนรู้ แต่กระบวนการในการเรียนรู้ของตัวควบคุมอัจฉริยะนั้นต่างออกไปเนื่องจากโครงสร้างที่ออกแบบนั้นไม่มีความจำเป็นที่จะต้องใช้ปริมาณข้อมูลจำนวนมาก ทำให้ตัวควบคุมอัจฉริยะนั้นมีการใช้จำนวนข้อมูลที่น้อยกว่าและมีประสิทธิภาพที่สูงกว่า

โดยในการทดลองในขั้นตอนที่ 7 จะเป็นการเปรียบเทียบประสิทธิภาพของ ICS vs MRAC-RK6 ซึ่งผลของการทดลองนั้นได้แสดงให้เห็นประสิทธิภาพของระบบควบคุมแบบอัจฉริยะเปรียบเทียบกับ MRAC-RK6 อย่างน่าประทับใจ ที่ระบบควบคุมแบบอัจฉริยะนอกจากจะสามารถควบคุมกระบวนการที่ไม่เป็นเชิงเส้นให้เป็นไปตามสัญญาณของรูปแบบอ้างอิงที่ไม่คงที่และมีความผันผวนแล้วนั้น ตัวควบคุมยังสามารถกำจัดค่าความคลาดเคลื่อนในการควบคุมได้ดีกว่า MRAC-RK6 อย่างเห็นได้ชัดอีกทั้งยังรองรับ

ความแปรปรวนที่เกิดจาก MLPINN ที่มีเงื่อนไข C2 ที่ใช้ค่าคงที่  $\varepsilon = 8 \times 10^{-4}$  ได้อย่างมีประสิทธิภาพ สามารถแสดงผลลัพธ์ของการควบคุมของการควบคุมอัจฉริยะได้ดังรูปที่ 4.6



รูปที่ 4.6 ผลลัพธ์การควบคุมของระบบควบคุมแบบอัจฉริยะ

ผลลัพธ์ที่ได้นั้นเป็นที่น่าพอใจ แต่ทว่า ICS แบบปกติมีช่วงการควบคุมที่ไม่เสถียรในช่วงเริ่มแรกของการควบคุม ซึ่งสามารถอธิบายได้จากพฤติกรรมของเครือข่ายประสาทนั้นจะมีลักษณะที่ต้องทำการเรียนรู้จากความคลาดเคลื่อนที่เกิดขึ้นซึ่งอาศัยการลองผิดลองถูกเพื่อทำการปรับแต่งพารามิเตอร์ให้เหมาะสมกับกระบวนการที่จะเกิดขึ้นโดยมีลักษณะและความเร็วในการเรียนรู้ที่ขึ้นอยู่กับไฮเพอร์พารามิเตอร์ โดยปกติแล้วไฮเพอร์พารามิเตอร์นั้นถูกปรับตั้งโดยผู้วิจัย ทำให้ความสามารถในการเรียนรู้ของเครือข่ายประสาท(หรือในที่นี้คือการปรับตัวของตัวควบคุม)นั้นแตกต่างกันออกไปตามความเชี่ยวชาญและผ่านการลองผิดลองถูกของผู้วิจัยมานับครั้งไม่ถ้วน ซึ่งมีความไม่แน่นอนและมีโอกาสที่ต้องทำการตั้งค่าไฮเพอร์พารามิเตอร์ทุกครั้งที่มีการเปลี่ยนแปลงกระบวนการที่ต้องการควบคุม ไม่อาจที่สามารถทำซ้ำได้ ดังนั้นแล้วจึงมีการพัฒนาความสามารถในการปรับปรุงการตั้งค่าด้วยตนเองของเครือข่ายประสาทมาใช้งาน โดยเรียกกระบวนการนี้ว่า Hyper Parameter Optimization ด้วยวิธีการ Bayesian Optimization ที่ได้มีการกล่าวถึงทฤษฎีและหลักการไปในบทที่ 2

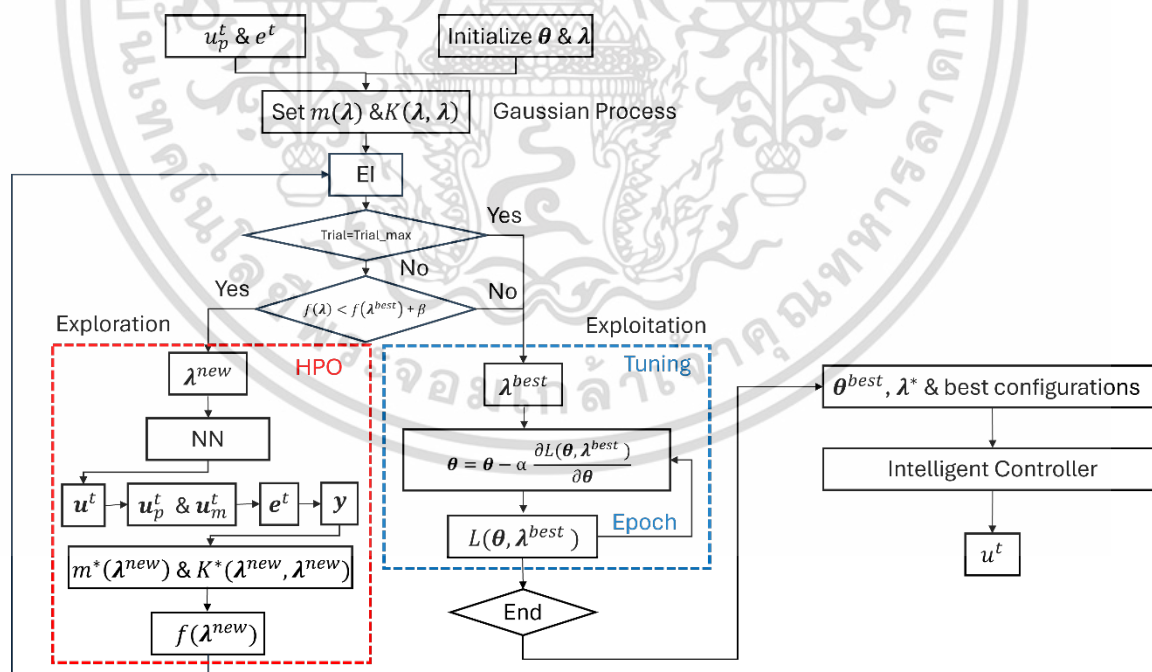
#### 4.4 การปรับปรุง Intelligent Controller ด้วย HPO

โดยทฤษฎีแล้วกระบวนการ HPO ด้วยวิธีการ Bayesian Optimization นั้นมีความซับซ้อนในระดับที่สูงและอาจจะยากต่อการทำความเข้าใจ แต่วิธีการในการพัฒนาเข้ากับเครือข่ายประสาทโดยใช้ Keras library นั้นกลับง่ายกว่ามาก โดยสิ่งที่ต้องทำความเข้าใจก่อนจะพัฒนานั้นมีเพียงการเข้าใจในหลักการและวิธีการนำไปใช้งานเพียงเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HPO คือกระบวนการที่ใช้ในการปรับแต่งไฮเพอร์พารามิเตอร์โดยตัวของเครือข่ายประสาทเทียมเอง เพื่อเป็นการทำให้เครือข่ายประสาทเทียมสามารถหาไฮเพอร์พารามิเตอร์เหมาะสมของตัวเองได้โดยลดความช่วยเหลือของมนุษย์ลง ซึ่งก่อให้เกิดความสามารถในการทำซ้ำใหม่ได้ และขยายความเป็นไปได้ในการพัฒนาให้กับเครือข่ายประสาท โดยวิธีการที่ก่อให้เกิดกระบวนการ HPO นั้นเองมีอยู่หลายวิธีการหนึ่งในนั้นคือ Bayesian Optimization ที่จะนำมาปรับใช้กับตัวควบคุมอัจฉริยะเพื่อแก้ปัญหาที่ได้กล่าวนำไปในบทย่อก่อนหน้า

โดยหลักการของ Bayesian Optimization จะใช้การหาความเป็นไปได้ของไฮเพอร์พารามิเตอร์ที่เหมาะสมจากกระบวนการ Gaussian Process ซึ่งทำงานโดยสัมพันธ์กับ ฟังก์ชัน Expected Improvement หรือฟังก์ชันคาดการณ์การพัฒนาที่เหมาะสม โดยฟังก์ชันคาดการณ์นั้นจะทำการแนะนำค่าของไฮเพอร์พารามิเตอร์บนย่านไฮเพอร์พารามิเตอร์ที่กำหนดจากความเป็นไปได้ใน Gaussian process โดยฟังก์ชันคาดการณ์จะทำการหาไฮเพอร์พารามิเตอร์บนย่านที่ว่านั้นเพื่อนำมาใช้ในการทดสอบบนฟังก์ชันชั้นเป้าหมายเพื่อหาไฮเพอร์พารามิเตอร์ที่ดีที่สุดในการทำผลลัพธ์ของฟังก์ชันเป้าหมายที่ดีที่สุดนั่นเอง ซึ่งจำนวนรอบของการทดสอบนั้นผู้พัฒนาสามารถกำหนดได้ ดังนั้นแล้วเงื่อนไขในการพัฒนา Bayesian Optimization นั้นมีสองเงื่อนไขคือการระบุฟังก์ชันเป้าหมายและทิศทางให้ถูกต้อง และ ต้องเลือกชนิดของไฮเพอร์พารามิเตอร์ที่จะก่อให้เกิดความเป็นไปได้การพัฒนาสูงที่สุด



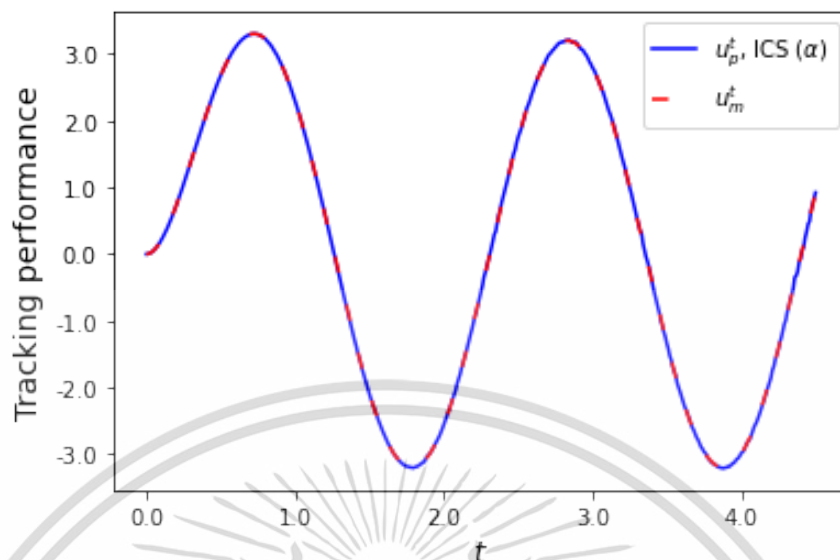
รูปที่ 4.7 กระบวนการ HPO สำหรับตัวควบคุมอัจฉริยะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปนี้จะกล่าวถึงวิธีการพัฒนา HPO ในตัวควบคุมอัจฉริยะโดยอ้างอิงจากรูปที่ 4.7 กระบวนการ HPO สำหรับตัวควบคุมอัจฉริยะ

1. กำหนดประเภทของไฮเพอร์พารามิเตอร์ที่ต้องการปรับตั้ง โดยในงานวิจัยนี้จะปรับเพียงอัตราการเรียนรู้ของตัวควบคุมอัจฉริยะเท่านั้น เนื่องจากเป็นไฮเพอร์พารามิเตอร์ที่ส่งผลต่อการเรียนรู้โดยตรงและสามารถเข้าใจทิศทางของการปรับตัวได้โดยง่าย
2. กำหนดจำนวนรอบที่ใช้ในการทดสอบของ Bayesian Optimization ในการทดลองนั้นมีการทดลองปรับตั้งแค่ 2 รอบไปจนถึง 20 รอบและพบว่าการปรับตัวเพียง 5 รอบนั้นเป็นจุดสมดุลระหว่างระยะเวลาการประมวลผลและประสิทธิภาพของตัวควบคุมอัจฉริยะที่ได้รับ
3. กำหนดสถาปัตยกรรมของตัวควบคุมอัจฉริยะ โดยใช้พารามิเตอร์จากรอบการควบคุมก่อนหน้าเพื่อให้เกิดการส่งต่อการเรียนรู้
4. สร้างตัวควบคุมจำลองโดยใช้ไฮเพอร์พารามิเตอร์และสถาปัตยกรรมในข้อที่ 1 และ 2
5. กำหนดฟังก์ชันเป้าหมายและการสูญเสียเป็นความคลาดเคลื่อนของกระบวนการโดยใช้การพัฒนาแบบเดียวกับหัวข้อย่อยที่ 4.3
6. เริ่มกระบวนการ HPO โดยใช้ Bayesian Optimization และมีฟังก์ชันเป้าหมายเป็นกระบวนการที่ใช้ในการควบคุม โดยมีค่าการสูญเสียของกระบวนการเป็น Loss ให้กับ Optimizer และมีเป้าหมายโดยการทำให้ค่า Loss น้อยที่สุด (กรอบสีแดง)
7. เมื่อกระบวนการ HPO ถึงรอบการทดสอบสุดท้ายหรือ Bayesian Optimization ไม่สามารถปรับปรุงไฮเพอร์พารามิเตอร์ได้อีก นำไฮเพอร์พารามิเตอร์ที่ได้ มาปรับปรุงให้กับตัวควบคุมอัจฉริยะ
8. ทำการหาผลลัพธ์และปรับปรุงพารามิเตอร์ของตัวควบคุมจำลองตามรูปแบบของตัวควบคุมอัจฉริยะในหัวข้อย่อยที่ 4.3
9. ทำซ้ำขั้นตอนที่ 2-8 จนกระทั่งถึงรอบการควบคุมสุดท้าย

ผลลัพธ์ให้เห็นในรูปที่ 4.8 ผลลัพธ์การควบคุมของระบบควบคุมแบบอัจฉริยะที่มีการปรับปรุงไฮเพอร์พารามิเตอร์ ลักษณะของกระบวนการที่ถูกควบคุมนั้นมีลักษณะตรงกันกับสัญญาณอ้างอิงใน รูปที่ 4.5 สัญญาณอ้างอิงที่ใช้ในการเปรียบเทียบเป็น ซึ่งเป็นการแสดงถึงความสามารถอันยอดเยี่ยมของ ICS ที่มีการนำ HPO มาใช้งาน เป็นการพัฒนาตัวควบคุมอัจฉริยะครั้งสำคัญ โดยตัวควบคุมนั้นสามารถที่จะกำจัดความคลาดเคลื่อนของการควบคุมจนเข้าใกล้ศูนย์ได้อย่างเกือบสมบูรณ์แบบ แสดงให้เห็นว่ากระบวนการ HPO นั้นสามารถแก้ปัญหาที่ได้กล่าวไว้ในบทย่อยที่ 4.3 ได้เป็นอย่างดี โดยจะอภิปรายและสรุปผลได้ในบทย่อยที่ 4.5 ต่อไป

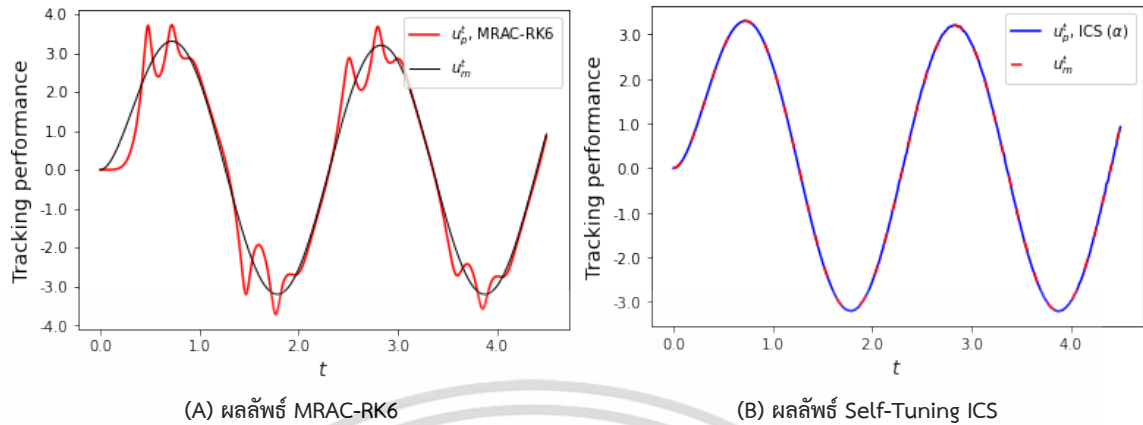


รูปที่ 4.8 ผลลัพธ์การควบคุมของระบบควบคุมแบบอัจฉริยะที่มีการปรับปรุงไฮเปอร์พารามิเตอร์

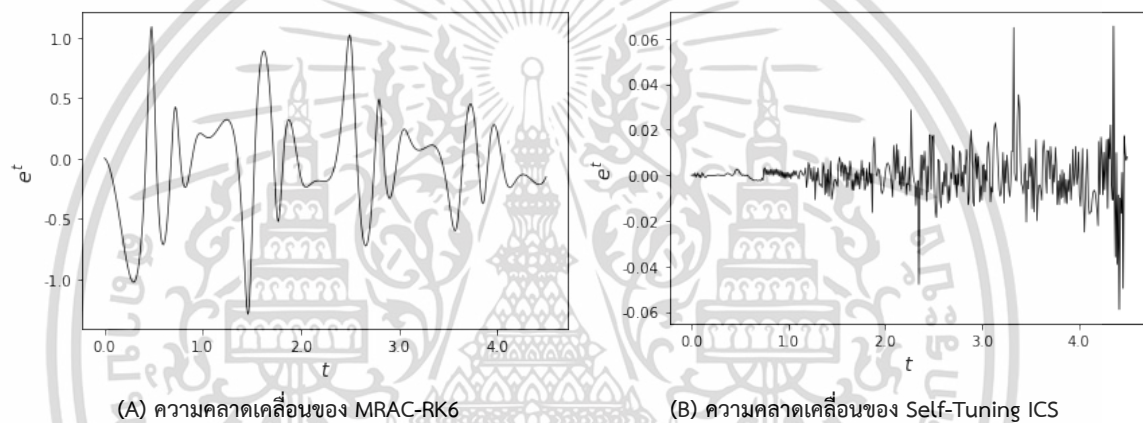
#### 4.5 ผลการทดลอง

ในการเริ่มต้นการวิจัยนี้จะมีสองตัวแปรควบคุมที่ต้องควบคุมเพื่อให้การเปรียบเทียบมีความเท่าเทียมกันและปราศจากการรบกวนต่างๆคือ (1) กระบวนการที่ไม่เป็นเชิงเส้นที่ใช้ในการเปรียบเทียบ (2) รูปแบบอ้างอิง (3) ช่วงเวลาที่ทำการควบคุม โดยมีตัวแปรต้นเป็นวิธีการที่ใช้ในการควบคุมคือ MRAC และ ICS และมีตัวแปรตามเป็นความสามารถในการทำให้ความคลาดเคลื่อนจากการควบคุม  $e^t$  เข้าสู่ศูนย์ พิจารณาตัวแปรต้น (1)  $\dot{u}_p^t = u_p^t + (u_p^t)^2 + 3 \cdot u^t$  (2)  $\dot{u}_m^t = -4 \cdot u_m^t + (16)\sin(3t)$  และ (3)  $t = [0, 4.5]$  และทำการพิจารณาตัวแปรตาม  $e^t = u_m^t - u_p^t$  จากการดำเนินการวิจัยและทดลองตั้งแต่บทที่ 2 ถึงบทที่ 4 จะแสดงให้เห็นว่าขั้นตอนการออกแบบและสร้างนั้นอ้างอิงอยู่บนตัวแปรเหล่านี้ทั้งหมดสมบัติของกระบวนการ และคุณสมบัติของรูปแบบอ้างอิง

จากนี้ไปจะเป็นการสรุปและอภิปรายผลการทดลองที่เกิดขึ้นโดยจะทำการทำการเปรียบเทียบระหว่าง ICS Self-Tuning และ MRAC โดยพิจารณารูปที่ 4.9 เปรียบเทียบผลตอบสนองระหว่าง MRAC และ Self-Tuning ICS



รูปที่ 4.9 เปรียบเทียบผลตอบสนองระหว่าง MRAC และ Self-Tuning ICS



รูปที่ 4.10 เปรียบเทียบความคลาดเคลื่อนระหว่าง MRAC และ Self-Tuning ICS

หากพิจารณาจากประสิทธิภาพของตัวควบคุมทั้งสองแล้วจะเห็นได้ว่า ICS นั้นมีประสิทธิภาพในการติดตามการเปลี่ยนแปลงของสัญญาณอ้างอิงได้ดีกว่า MRAC ได้อย่างชัดเจนแม้ความละเอียดการแสดงผลของผลตอบสนองนั้นไม่ได้แสดงโดยละเอียดมากนัก แต่หากวิเคราะห์ให้ถี่ถ้วนลงไปอีกระดับในการพิจารณาความคลาดเคลื่อนของการควบคุมดังแสดงให้เห็นในรูปที่ 4.10 เปรียบเทียบความคลาดเคลื่อนระหว่าง MRAC และ Self-Tuning ICS จะพบว่าความคลาดเคลื่อนสูงสุดของ MRAC นั้นอยู่ที่ประมาณ 1.0, -1.3 ในขณะที่ ICS อยู่ที่ประมาณ 0.06, -0.06 เพียงเท่านั้น และหากพิจารณาโดยใช้การคำนวณหาการสูญเสียโดยวิธีการ L2 Normalization มาช่วยในการพิจารณาผลการทดลองโดยมีฟังก์ชันทางคณิตศาสตร์ดังสมการที่ 4.1 ใช้ในการคำนวณเปรียบเทียบระหว่างสัญญาณอ้างอิงและผลลัพธ์ของกระบวนการตั้งแต่  $t=0$  ถึง  $t=4.5$  และแสดงผลดังแสดงในตารางที่ 4.1

$$L2 \text{ Normalization} = \sqrt{\sum_{t=0}^{4.5} (u_m^t - u_p^t)^2} \quad (4.1)$$

ตารางที่ 4.1 ผลการเปรียบเทียบระหว่าง MRAC และ ICS โดยใช้ L2 Normalization

| Control System | L2 Normalization    | Performance Improvement (%) |
|----------------|---------------------|-----------------------------|
| MRAC           | 8.710778469602346   | -                           |
| ICS            | 3.562938973799301   | 144.483%                    |
| ICS, HPO       | 0.21940959149570785 | 3870.099%                   |

จากผลการเปรียบเทียบแสดงให้เห็นว่าระบบควบคุมแบบ ICS ทั้งแบบที่ไม่ใช้กระบวนการ HPO และแบบที่ใช้กระบวนการ HPO ในทุกรอบของการควบคุมนั้นมีประสิทธิภาพที่ดีกว่า MRAC-RK6 อย่างเห็นได้ชัด โดย ICS และ ICS, HPO มีค่า L2 Normalization ที่ดีกว่า MRAC ถึง 144.483%, 3870.099% ตามลำดับ

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

### 5.1 สรุปผลการวิจัย

จากผลของการวิจัยการเปรียบเทียบความสามารถในการควบคุมเป้าหมายที่มีสัญญาณเข้า และสัญญาณออกของเดียวโดยมีอัตราการเปลี่ยนแปลงไม่คงที่ (Nonlinear Single-Input Single-Output Plant) ซึ่งเป็นการเปรียบเทียบความสามารถของการควบคุมระหว่าง การควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง (Model Reference Adaptive Control) บนหลักการของ Lyapunov และ การควบคุมอัจฉริยะ (Intelligent Control) โดยกำหนดตัวแปรควบคุมเป็น กระบวนการที่ไม่เป็นเชิงเส้นในลักษณะของเลขยกกำลังที่เมื่อเวลาผ่านกระบวนการจะมีลักษณะเข้าสู่อินันต์ และกำหนดรูปแบบอ้างอิงให้มีการเปลี่ยนแปลงของสัญญาณอ้างอิงที่ไม่คงที่ตลอดช่วงเวลาที่สนใจให้มีลักษณะของความเป็นคลื่น ซึ่งด้วยตัวแปรควบคุมนี้เองทำให้การเปรียบเทียบในงานวิจัยฉบับนี้นั้นปราศจากตัวแปรรบกวนต่างๆ

ตัวควบคุมอัจฉริยะในงานวิจัยออกแบบบนพื้นฐานของเครือข่ายประสาทเทียม โดยมีการประยุกต์รูปแบบการควบคุมมาจาก MRAC โดยทำการปรับแต่งในบางส่วนให้เหมาะสมกับเครือข่ายประสาทเทียมและสามารถแสดงศักยภาพของ ICS ที่ใช้เครือข่ายประสาทเทียมได้อย่างมีประสิทธิภาพสูงที่สุด โดยสามารถเปรียบเทียบแผนผังการควบคุมได้จากรูปที่ 2.1 แผนภาพรูปแบบการควบคุมของระบบควบคุมแบบปรับตัวชนิดรูปแบบอ้างอิง และ รูปที่ 4.1 แผนภาพการควบคุมของระบบควบคุมแบบอัจฉริยะ

โดยในงานวิจัยการออกแบบ MRAC-RK6 จะใช้การออกแบบบนพื้นฐานของ Lyapunov และหาคุณสมบัติกระบวนการด้วยวิธี RK6 ซึ่งเป็นวิธีการในการหาผลลัพธ์สมการเชิงอนุพันธ์ที่เป็นที่รู้จักเป็นอย่างดี และการออกแบบ ICS ดังที่ได้กล่าวไปแล้วจะใช้เครือข่ายประสาทเทียมแบบ Multi-Layer Perceptron Neural Network เป็นโครงสร้างหลักของตัวควบคุม รวมไปถึงใช้ MLPINN หรือ PINN ที่มีการเรียนรู้แบบเมต้าในการหาคุณสมบัติของกระบวนการที่อยู่ในรูปแบบของสมการเชิงอนุพันธ์

ทั้งสองระบบควบคุมมีคุณสมบัติที่สามารถปรับปรุงพารามิเตอร์ภายในตัวควบคุมได้จากการตรวจสอบความคลาดเคลื่อนของการควบคุมได้ โดย MRAC ใช้ Adaptation Laws ในการปรับปรุง ในขณะที่ ICS ใช้กระบวนการ Back Propagation ในการปรับปรุงพารามิเตอร์ภายใน ยิ่งไปกว่านั้น ICS ยังผสมเข้ากับกระบวนการ Hyper Parameter Optimization โดยใช้หลักการ Bayesian Optimization ทำให้สามารถปรับปรุงไฮเพอร์พารามิเตอร์ของตัวเองในทุกรอบการควบคุมอีกด้วย

ในการนี้โดยปกติแล้วเครือข่ายประสาทเทียมจำเป็นต้องอาศัยมนุษย์ในการระบุชุดข้อมูลจริงให้ ในกระบวนการเรียนรู้ของเครือข่ายประสาทเทียม แต่เครือข่ายประสาทที่ใช้ใน ICS นั้นสามารถระบุข้อมูล และทำการปรับข้อมูลนั้นได้ด้วยตัวของเครือข่ายประสาทเอง อีกทั้งการพัฒนา HPO ให้แก่ตัวควบคุมอัจฉริยะนั้นยังลดภาระงานของมนุษย์ลงไปอีกชั้นหนึ่ง อาจสามารถกล่าวได้ว่าในกระบวนการการออกแบบและสร้าง ICS นั้นตัวผู้วิจัยไม่จำเป็นต้องมีการระบุข้อมูลแต่อย่างใด ตัวเครือข่ายประสาทใน ICS นั้นสามารถสร้างรูปแบบ, ปรับแต่งพารามิเตอร์และไฮเพอร์พารามิเตอร์ของตัวเอง และรวมไปถึงสามารถตรวจสอบความผิดพลาดของตัวเองได้อีกด้วย

ในด้านความสามารถในการควบคุม ดังแสดงให้เห็นในรูปที่ 4.9 เปรียบเทียบผลตอบสนองระหว่าง MRAC และ Self-Tuning ICS และ รูปที่ 4.10 เปรียบเทียบความคลาดเคลื่อนระหว่าง MRAC และ Self-Tuning ICS แสดงให้เห็นว่า ICS นั้นมีความสามารถในการควบคุมกระบวนการ  $n^{\text{th}}$  ตามสัญญาณอ้างอิง  $n^{\text{th}}$  ที่สูงกว่า MRAC เป็นอย่างมาก ในขณะที่ MRAC นั้นไม่สามารถกำหนดรูปแบบการควบคุมได้ แต่ ICS นั้นสามารถกระทำได้อย่างดี เนื่องจาก ICS มีจำนวนพารามิเตอร์ภายในที่สูงกว่าทำให้สามารถรองรับการเปลี่ยนแปลงได้ดีกว่า อีกทั้ง MRAC นั้นโดยลักษณะในการออกแบบแล้วนั้น MRAC จะไม่สามารถควบคุมกระบวนการที่มีลำดับอนุพันธ์ที่สูงกว่าลำดับอนุพันธ์ที่ใช้ในการออกแบบได้ ซึ่งสามารถสังเกตได้จากผลลัพธ์ของงานวิจัย “Event-triggered MRAC System Design for SISO Plants Using Meta-tuning-based Physics-informed Neural Networks and Transfer Learning without Labeled Data” ในกรณีที่เงื่อนไขเหตุการณ์ C2 ของ MLPINN มีค่า  $\varepsilon = 8 \times 10^{-4}$  ทำให้ลำดับอนุพันธ์ของกระบวนการมีลำดับที่สูงกว่าที่ออกแบบไว้ ทำให้ MRAC ไม่สามารถควบคุมความแปรปรวนดังกล่าวได้ และดังแสดงให้เห็นในการเปรียบเทียบที่ MRAC-RK6 ไม่สามารถที่จะควบคุมกระบวนการที่ไม่เป็นเชิงเส้นพร้อมกันกับสัญญาณอ้างอิงที่ไม่เป็นเชิงเส้นเช่นเดียวกันได้ โดยเฉพาะในช่วงเริ่มต้นของการควบคุม

ดังที่ได้สรุปผลการทดลองไว้ในบทที่ 4 เราสามารถที่จะประเมินผลการเปรียบเทียบได้จากวิธีการทางคณิตศาสตร์ L2 Normalization โดยผลดังกล่าวของ MRAC, ICS และ ICS( $\alpha$ ) นั้นเป็น 8.71, 3.56, และ 0.22 ตามลำดับซึ่งแสดงให้เห็นถึงประสิทธิภาพที่สูงกว่าอย่างน่าประทับใจของ ICS ทั้งแบบธรรมดาและแบบที่มีการปรับแต่งไฮเพอร์พารามิเตอร์ด้วยกระบวนการ HPO นั้นมีประสิทธิภาพที่สูงกว่า MRAC-RK6 เป็นอย่างมาก และยังมีแนวโน้มที่จะสามารถควบคุมกระบวนการที่มีลำดับอนุพันธ์ที่สูงยิ่งกว่าได้อีกด้วย

ซึ่งอาจสรุปได้ว่า MRAC นั้นต้องพึ่งพาการออกแบบจากกระบวนการทางคณิตศาสตร์ที่ซับซ้อน โดยเฉพาะการออกแบบการควบคุมสำหรับกระบวนการที่ไม่เป็นเชิงเส้นและมีลำดับอนุพันธ์ระดับสูง ความซับซ้อนของการออกแบบนี้ทำให้ MRAC การออกแบบ MRAC สำหรับกระบวนการดังกล่าวนี้มีอุปสรรคอย่างใหญ่หลวง และไม่สามารถปรับตัวเข้ากับกระบวนการที่มีลำดับอนุพันธ์ที่สูง

กว่าได้ ในทางตรงกันข้ามสถาปัตยกรรมของการควบคุมอัจฉริยะนั้นมีความโดดเด่นอย่างมากทั้งในด้านของการออกแบบนั้นไม่ซับซ้อนและมีประสิทธิภาพ อีกทั้งยังผนวกด้วยเครือข่ายประสาทเทียม Meta-Learning Physical-Informed Neural Network (MLPINN) ที่มีประสิทธิภาพสูงในการหาคุณสมบัติของกระบวนการ ยิ่งไปกว่านั้นผลการทดลองยังแสดงให้เห็นถึงการพัฒนาประสิทธิภาพของการควบคุมอัจฉริยะที่เหนือกว่า MRAC อย่างยิ่งยวด โดยมีการพัฒนาตั้งแต่การออกแบบแผนผังการควบคุม, อัลกอริทึมการปรับพารามิเตอร์แบบออนไลน์, กระบวนการ Hyper Parameter Optimization ด้วย Bayesian Optimization ไปจนถึงการพัฒนาอัลกอริทึมในการส่งต่อพารามิเตอร์ โดยพิสูจน์ประสิทธิภาพที่สูงกว่าได้จากผลการทดลองที่  $ICS(\alpha)$  สามารถติดตามความคลาดเคลื่อนที่เกิดขึ้นได้ดีกว่า MRAC ถึง 3870%

## 5.2 ข้อเสนอแนะ

จากผลการทดลองนั้นความสามารถในการควบคุมกระบวนการของ  $ICS$  นั้นสูงกว่า MRAC อย่างเห็นได้ชัด แต่ทว่า  $ICS$  นั้นเป็นเครือข่ายประสาทเทียมที่มีลักษณะการเรียนรู้แบบออนไลน์ทั้งระบบ รวมไปถึงมีลักษณะการเรียนรู้ที่เป็นลำดับขั้นทำให้ระยะเวลาประมวลผลในแต่ละรอบของการควบคุมนั้นต้องใช้ระยะเวลาเป็นอย่างมาก อีกทั้งด้วยข้อบกพร่องของไลบรารีที่ใช้งานทำให้เกิดปัญหาการรั่วไหลของหน่วยความจำแบบสุ่มในทุกรอบของการเรียนรู้ ทำให้ช่วงเวลาที่ใช้ในการจำลองการควบคุมนั้นต้องลดลงอย่างหลีกเลี่ยงไม่ได้ จึงเป็นเหตุผลว่าผลของการทดลองมีช่วงเวลาที่สนใจเพียงตั้งแต่  $t=0$  ถึง  $t=4.5$  วินาทีเท่านั้น ด้วยเหตุผลเหล่านี้เอง  $ICS$  จึงยังไม่สามารถนำไปใช้งานในลักษณะการทำงานแบบ Real-time ได้ และจำเป็นต้องใช้ทรัพยากรในการคำนวณที่สูงอีกด้วย ในงานวิจัยในอนาคตอาจจำเป็นที่จะต้องปรับปรุงข้อบกพร่องดังกล่าวทั้งในแง่เชิงเทคนิควิธีการ และการออกแบบอัลกอริทึมที่สามารถลดระยะเวลาการประมวลผลลงได้

## บรรณานุกรม

- [1] Balakrishnan, S. N., and Biega, V. (1996). "Adaptive-critic-based neural networks for aircraft optimal control ". **Journal of Guidance, Control, and Dynamics**, 19(4), 893-898.
- [2] Calvo-Rolle, J. L., Fontenla-Romero, O., Pérez-Sánchez, B., and Guijarro-Berdiñas, B. (2014). "Adaptive Inverse Control Using an Online Learning Algorithm for Neural Networks ". **Informatica**, 25(3), 401-414.
- [3] Duanyai, W., Song, W. K., Konghuayrob, P., and Parnichkun, M. (2024). "Event-triggered model reference adaptive control system design for SISO plants using meta-learning-based physics-informed neural networks without labeled data and transfer learning ". **International Journal of Adaptive Control and Signal Processing**, 38(4), 1442-1456.
- [4] Narendra, K. S. P., K. (1990). "Identification and control of dynamical systems using neural networks ". **IEEE Transactions on Neural Networks**, 1(1), 4-27.
- [5] Peng, Y., Flach, P. A., Soares, C., and Brazdil, P. (2002). "Improved Dataset Characterisation for Meta-learning". In **Discovery Science** (pp. 141-152).
- [6] Duanyai, W. S., Weon Keun ; Chitthamlerd, Thanadol ; Kumar, Girish. (2024). "Meta-learning-based Physics-informed Neural Network: Numerical Simulations of Initial Value Problems of Nonlinear Dynamical Systems without Labeled Data and Correlation Analyses ". **Journal of Nonlinear Modeling and Analysis**.
- [7] Finn, C., Abbeel, P., and Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. International Conference on Machine Learning,
- [8] Avila, L. I. M., Martínez, A. V., Zhang, Y., Castañón, L. E. G., Ortiz, E. R. C., and Viola, J. C. (2013). "Model-based Control Approaches for Optimal Integration of a Hybrid Wind-diesel Power System in a Microgrid **Proceedings of the 2nd International Conference on Smart Grids and Green IT Systems**,
- [9] Zayegh, A., and Al Bassam, N. (2018). "Neural Network Principles and Applications". In **Digital Systems**.

- [10] Hunt, K. J., Sbarbaro, D., Zbikowski, R., and Gawthrop, P. J. (1992). "Neural networks for control systems—A survey ". **Automatica**, 28(6), 1083-1112.
- [11] Nahas, E. P., Henson, M. A., and Seborg, D. E. (1992). "Nonlinear internal model control strategy for neural network models ". **Computers & Chemical Engineering**, 16(12), 1039-1057.
- [12] Yang, L., and Shami, A. (2020). "On hyperparameter optimization of machine learning algorithms: Theory and practice ". **Neurocomputing**, 415, 295-316.
- [13] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations ". **Journal of Computational Physics**, 378, 686-707.
- [14] Abdel-Nasser, S. (2020). "Principle of Neural Network and Its Main Types: Review ". **Journal of Advances in Applied & Computational Mathematics**, 7, 8-19.
- [15] Shekhar, A., and Sharma, A. (2018). Review of Model Reference Adaptive Control. 2018 International Conference on Information , Communication, Engineering and Technology (ICICET), Pune, India.
- [16] Frazier, P. (2018). "A Tutorial on Bayesian Optimization ". **ArXiv**, abs/1807.02811.
- [17] Rasmussen, C. E., and Williams, K. I. (2006). "Varying the Hyperparameters". In **Gaussian Processes for Machine Learning** (pp. 19-21). Massachusetts Institute of Technology.

## ภาคผนวก

วิทยานิพนธ์ฉบับนี้ประกอบด้วยผลงานวิจัยที่ได้รับการตีพิมพ์ทั้งหมด 2 ฉบับ

1. Duanyai, W., Song, W. K., Chitthamlerd, T., and Kumar, G. (2024). "Meta-learning-based Physics-informed Neural Network: Numerical Simulations of Initial Value Problems of Nonlinear Dynamical Systems without Labeled Data and Correlation Analyses ". Journal of Nonlinear Modeling and Analysis.
2. Duanyai, W., Song, W. K., Konghuayrob, P., and Parnichkun, M. (2024). "Event-triggered model reference adaptive control system design for SISO plants using meta-learning-based physics-informed neural networks without labeled data and transfer learning ". International Journal of Adaptive Control and Signal Processing, 38(4), 1442-1456.

## ประวัติผู้เขียน

|                                 |   |
|---------------------------------|---|
| ชื่อ-นามสกุล                    | นายวรวัฒน์ ดวนใหญ่  |
| วัน เดือน ปีเกิด                | 4 มิถุนายน 2540 ที่กรุงเทพมหานคร  |
| ที่อยู่                         | 78/790 Niche Mono Mega Space Bangna Condo ถ.บางนา-ตราด ตำบล<br>บางแก้ว อำเภอบางพลี สมุทรปราการ 10540  |
| ประวัติการศึกษา                 | 2562 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอัตโนมัติ<br>สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  |
| ความชำนาญเฉพาะด้าน              | 1.) ระบบอัตโนมัติอุตสาหกรรม<br>2.) ระบบควบคุมอัตโนมัติ<br>3.) เซอไวโมเตอร์<br>4.) เครือข่ายประสาทเทียม  |
| ประสบการณ์การทำงานและผลงานวิจัย |   |
| พ.ศ. 2562-2563                  | ตำแหน่งวิศวกรอัตโนมัติบริษัทเฟิร์สเมคจำกัด<br>- ผลงานการสร้างและพัฒนาหุ่นยนต์ใส่พินให้กับแผงพีซีบอร์ด<br>- ผลงานการสร้างและพัฒนาเครื่องจักรตรวจสอบความถูกต้องของแหวนล้อค<br>ข้อต่อเพลารถยนต์โดยใช้กล้องความละเอียดสูง   |
| พ.ศ.2564 - 2565                 | ตำแหน่งวิศวกรกระบวนการบริษัทนาสไปเซอร์จำกัด<br>- ดูแลเครื่องจักรในกระบวนการผลิตให้เป็นไปตามแผนที่วางไว้<br>- ผลงานการวิจัยและพัฒนาการพ่นสีและอบแห้งเพลารถยนต์ด้วยด้วยรังสี<br>Ultraviolet<br>- ผลงานการพัฒนาเครื่องจักรในโรงงานให้เป็นระบบอัตโนมัติ   |
| พ.ศ.2565 - 2567                 | ตำแหน่งวิศวกรโซลูชันและแอปพลิเคชันอาวุโส<br>Beckhoff Automation Pte Ltd.<br>- ออกแบบระบบอัตโนมัติสำหรับอุตสาหกรรมเล็ก-หนัก<br>- ผลงานการพัฒนาเครื่องจักรพับผ้าสำหรับอุตสาหกรรมหนัก<br>- ผลงานการพัฒนาเครื่องป้อนผ้าสำหรับอุตสาหกรรมหนัก<br>- ผลงานการพัฒนาเครื่องตัดเหล็กสำหรับอุตสาหกรรมหนัก |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบัน

ตำแหน่งที่ปรึกษาการพัฒนาเทคโนโลยี

Beckhoff Automation Pte Ltd.

- วางแผนการขายเพื่อพัฒนาธุรกิจ
- ให้คำปรึกษาการพัฒนาเครื่องจักร และระบบควบคุม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้