



การพัฒนากระบวนการประมาณค่าแรงบิดสำหรับมอเตอร์ไร้แปรงถ่าน  
ด้วยเทคนิคไร้เซนเซอร์

A Development of a Torque Estimation Process for Brushless Motors  
with Sensorless Techniques

วันนาเดีย นาแว

Wannadeear Nawae

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา  
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the  
Degree of Master of Engineering in Electrical Engineering

Prince of Songkla University

2563

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

**ชื่อวิทยานิพนธ์**      การพัฒนากระบวนการประมาณค่าแรงบิดสำหรับมอเตอร์ไร้แปรงถ่านด้วยเทคนิคไร้เซนเซอร์

**ผู้เขียน**              นายวันนาเดีย นาแว

**สาขาวิชา**            วิศวกรรมไฟฟ้า

---

**อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก**

**คณะกรรมการสอบ**

.....

.....ประธานกรรมการ

(ดร.กิตติคุณ ทองพูล)

(รองศาสตราจารย์ ดร. ญัฐธา จินดาเพ็ชร)

.....กรรมการ

(ดร.พลสิทธิ์ ศานติประพันธ์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.พิทักษ์ บุญนุ่น)

.....กรรมการ

(ดร.กิตติคุณ ทองพูล)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยาลัยฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....

(ศาสตราจารย์ ดร.ดำรงศักดิ์ ฟ้ารุ่งแสง)

คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้เป็นผลมาจากการศึกษาวิจัยของนักศึกษาเอง และขอแสดงความขอบคุณ  
บุคคลที่มีส่วนเกี่ยวข้อง

ลงชื่อ.....

(ดร.กิตติคุณ ทองพูล)

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ลงชื่อ.....

(นายวันนาเดียง นาวะ)

นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อนและ  
ไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....

(นายวันนาเดีย นาแวง)

นักศึกษา

|                 |   |
|-----------------|---|
| ชื่อวิทยานิพนธ์ | การพัฒนากระบวนการประมาณค่าแรงบิดสำหรับมอเตอร์ไร้แปรงถ่านด้วยเทคนิคไร้เซนเซอร์ |
| ผู้เขียน        | นายวันนาเดีย นาแว   |
| สาขาวิชา        | วิศวกรรมไฟฟ้า   |
| ปีการศึกษา      | 2563  |

### บทคัดย่อ

ในงานวิจัยนี้ได้นำเสนอการประมาณค่าแรงบิดของมอเตอร์โดยไม่ใช้เซนเซอร์วัดแรงบิดสำหรับประยุกต์ใช้กับการทำงานของแขนหุ่นยนต์เพื่อต้องการลดต้นทุนในการผลิตและความซับซ้อนในการออกแบบของแขนหุ่นยนต์ โดยมีวัตถุประสงค์ของงานวิจัยคือการเปรียบเทียบประสิทธิภาพการประมาณค่าแรงบิดด้วย Machine learning โดยให้ความสำคัญกับการทำงานและการประมวลผลแบบทันเวลารวมไปถึงการออกแบบและผลิตอุปกรณ์ในทางด้านฮาร์ดแวร์ที่ใช้สำหรับทำการทดลอง โดยมีการออกแบบอ้างอิงจากการทำงานของแขนหุ่นยนต์ ในงานวิจัยได้เลือกใช้มอเตอร์ไร้แปรงถ่านในการเคลื่อนไหวของแขนหุ่นยนต์โดยมอเตอร์ชนิดนี้เป็นมอเตอร์ 3 เฟสที่มีสัญญาณไฟฟ้าย้อนกลับ ในขณะที่มอเตอร์ทำงานจึงสามารถนำสัญญาณดังกล่าวมาวิเคราะห์เพื่อหาข้อมูลและนำไปสู่กระบวนการประมาณค่าของแรงบิดได้ โดยในการควบคุมมอเตอร์ไร้แปรงถ่านจะใช้วิธีการควบคุม Field orientation control เป็นการควบคุมมอเตอร์ 3 เฟสที่ให้ประสิทธิภาพของแรงบิดและความเร็วได้ดีกว่ามอเตอร์กระแสตรงแบบมีแปรงถ่าน ในการทดลองของวิจัยได้เลือกใช้ตัวแปรที่เป็นส่วนหนึ่งของวิธีการควบคุมมอเตอร์ดังกล่าว เนื่องจากตัวแปรมีความสัมพันธ์กันโดยตรงกับแรงบิดที่เกิดขึ้นของมอเตอร์ และใช้กระบวนการเรียนรู้โมเดลของ Regression, Support vector machines และ Neural network เพื่อนำไปสู่การประมาณค่า ในการทดลองจะมีทั้งหมด 3 การทดลองประกอบไปด้วยการทดลองการประมาณค่าแรงบิดแบบของออฟไลน์, การทดลองการประมาณค่าแรงบิดแบบของออนไลน์และการทดลองการจำลองสถานการณ์การใช้งานร่วมกับแขนหุ่นยนต์เพื่อดูประสิทธิภาพ, ระยะเวลาและความล่าช้าในการประมาณค่าแรงบิดโดยเปรียบเทียบกับการวัดแรงบิดด้วยเซนเซอร์ โดยในการทดลองแบบออนไลน์พบว่าโมเดลของ Neural network สามารถให้คำตอบได้ดีที่สุดเมื่อเทียบกับเซนเซอร์วัดแรงบิด แต่เมื่อประเมินด้วยเวลาในการประมาณค่าจะใช้เวลามากกว่าโมเดลของ Regression

**คำสำคัญ** ประมาณค่าแรงบิด, มอเตอร์ไร้แปรงถ่าน, เทคนิคการควบคุมด้วยวิธีไร้เซนเซอร์

|                      |  |
|----------------------|--|
| <b>Thesis Title</b>  | A Development of a Torque Estimation Process for Brushless Motors with Sensorless Techniques |
| <b>Author</b>        | Mr. Wannadeear Nawae   |
| <b>Major Program</b> | Electrical Engineering   |
| <b>Academic Year</b> | 2020   |

### ABSTRACT

In this research, we propose a torque estimation without using a torque sensor for applying to robot operation to reduce production costs and the complexity of robot design. The objective of the research is to compare the efficiency of the torque estimation with machine learning, which focuses on real-time operation. This research has designed and built hardware devices used for experiments based on the operation of the robot arm. The operation of the robot arm uses a brushless dc motor to operate. This type of motor is a 3-phase motor that can take back electromotive force to analyse to find information which will lead to the torque estimation. Controlling a brushless dc motor using a vector control method, also known as Field orientation control, is a 3-phase motor control that makes the most of torque performance. The experiments in this research have chosen to use variables that are a part of motor control method because that variable has a direct change to the torque of the motor. The regression model, Support vector machine and Neural network can be used for torque estimation. The experiment will include both offline and online experiments, including application to the robot arm to see the performance. In an online experiment, the Neural network model provides the best solution when compared to a torque sensor. However, it takes much more estimation time than the Regression model.

**Keyword** Torque Estimation, Brushless DC motor, Sensorless control techniques

## กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จสมบูรณ์ได้ด้วยดีเพราะได้รับความกรุณาชี้แนะและช่วยเหลืออย่างดียิ่งจาก ดร.กิตติคุณ ทองพูล อาจารย์ที่ปรึกษาวิทยานิพนธ์ และรองศาสตราจารย์ ดร.ณัฐชา จินดาเพ็ชร์ ประธานกรรมการสอบวิทยานิพนธ์ และดร.พลสิทธิ์ ศานติประพันธ์ และผู้ช่วยศาสตราจารย์ ดร.พิทักษ์ บุญนุ่น กรรมการผู้ทรงคุณวุฒิสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร.รักรกฤตว์ ดวงสร้อยทอง ประธานคณะกรรมการบริหารหลักสูตรที่ให้คำแนะนำและตรวจแก้ไขข้อบกพร่องมาโดยตลอด ตั้งแต่เริ่มต้นจนสำเร็จเรียบร้อย ผู้วิจัยขอกราบขอบพระคุณด้วยความเคารพอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณบัณฑิตวิทยาลัย และคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ที่สนับสนุนเงินอุดหนุนวิจัย

ขอขอบพระคุณบิดา มารดา ที่สนับสนุนและให้กำลังใจจนงานวิจัยสำเร็จด้วยดี คุณค่าและประโยชน์อันพึงมีจากการศึกษาวิจัยนี้ ผู้วิจัยขอน้อมบูชาพระคุณบิดามารดาและบูรพาจารย์ทุกท่าน ที่ได้อบรมสั่งสอนวิชาความรู้ และให้ความเมตตาแก่ผู้วิจัยมาโดยตลอด และเป็นกำลังใจสำคัญ ที่ทำให้การศึกษาระดับนี้สำเร็จลุล่วงได้ด้วยดี

ลงชื่อ.....

(นายวันนาเดีย นาแว)

นักศึกษา

## สารบัญ

|  |      |
|--|------|
| บทคัดย่อ .....   | (5)  |
| ABSTRACT .....   | (6)  |
| กิตติกรรมประกาศ.....   | (7)  |
| สารบัญ.....  | (8)  |
| สารบัญรูปภาพ.....  | (11) |
| สารบัญตาราง.....   | (14) |
| บทที่ 1 บทนำ .....   | 1    |
| 1.1 ความสำคัญและที่มาของหัวข้อวิจัย.....   | 1    |
| 1.2 การตรวจสอบเอกสาร บทความและงานวิจัยที่เกี่ยวข้อง .....  | 2    |
| 1.2.1 Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application Trends[1].....                     | 3    |
| 1.2.2 A Motor controller using field oriented control and hall effect rotor position sensor: simulation and implementation[2] .....    | 3    |
| 1.2.3 Instantaneous Torque Estimation in Sensorless Direct-Torque-Controlled Brushless DC Motors[3].....                               | 4    |
| 1.2.4 Electromagnetic torque and speed estimators for permanent magnet synchronous motor drive system[4].....                          | 4    |
| 1.2.5 A Nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation[5] ..... | 5    |
| 1.2.6 Monitoring of induction motor load by neural network techniques[6] .....   | 5    |
| 1.3 วัตถุประสงค์ของโครงการวิจัย .....  | 5    |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ.....   | 6    |
| 1.5 ขอบเขตการวิจัย .....   | 6    |
| 1.6 แผนการดำเนินการ.....   | 7    |
| 1.7 อุปกรณ์และสถานที่ทำวิจัย.....  | 8    |
| บทที่ 2 ทฤษฎีและหลักการ.....   | 9    |
| 2.1 ฮาร์ดแวร์สำหรับทำงานวิจัย.....   | 9    |



|   |    |
|---|----|
| 2.1.1 Brushless direct current motor (BLDC motor)[7].....                                   | 9  |
| 2.1.2 Brushless direct current motor control[2] .....                                       | 10 |
| 2.1.3 การประมาณค่าแรงบิดด้วยสัญญาณไฟฟ้าย้อนกลับด้วยสมการ Electromagnetic Torque[8]<br>..... | 11 |
| 2.2 ซอฟต์แวร์สำหรับทำงานวิจัย .....   | 12 |
| 2.2.1 Regression model .....  | 12 |
| 2.2.2 Support vector machine (SVM).....   | 12 |
| 2.2.3 Neural network[5] .....   | 13 |
| 2.2.4 Root mean square error (RMSE).....  | 14 |
| 2.2.5 Regression value (R value).....   | 14 |
| บทที่ 3 การออกแบบงานวิจัยและขั้นตอนการทดลอง .....   | 15 |
| 3.1 ออกแบบระบบฮาร์ดแวร์สำหรับการทดลองเพื่อวัดแรงบิดของมอเตอร์ .....                         | 15 |
| 3.2 ออกแบบการทดลองสำหรับการประมาณค่าแรงบิด.....   | 17 |
| 3.3 การดำเนินการและการเก็บข้อมูลบนบอร์ดไมโครคอนโทรลเลอร์ .....                              | 24 |
| 3.4 การใช้ Neural network ของโปรแกรม MATLAB.....  | 26 |
| 3.5 การใช้งาน Regression learner ของ MATLAB .....   | 29 |
| บทที่ 4 ผลการทดลองและการวิเคราะห์.....  | 33 |
| 4.1 ผลการออกแบบอุปกรณ์สำหรับการทดลอง.....   | 33 |
| 4.2 ผลการทดลองในการประมาณค่าแรงบิด.....   | 37 |
| 4.2.1 ผลการทดลองการประมาณค่าแรงบิดแบบออฟไลน์ .....  | 37 |
| 4.2.2 ผลการทดลองการประมาณค่าแรงบิดแบบออนไลน์.....   | 43 |
| 4.3 ผลการทดลองของการประมาณค่าแรงบิดด้วยการจำลองในแขนหุ่นยนต์.....                           | 53 |
| การทดลองที่ 4.3.1 แรงบิดเท่ากับ 3 Nm .....  | 54 |
| การทดลองที่ 4.3.2 แรงบิดเท่ากับ 5 Nm .....  | 55 |
| การทดลองที่ 4.3.3 แรงบิดเท่ากับ 8 Nm .....  | 56 |
| การทดลองที่ 4.3.4 แรงบิดเท่ากับ 10 Nm .....   | 57 |
| บทที่ 5 บทสรุป.....   | 58 |
| 5.1 ผลการออกแบบอุปกรณ์สำหรับการทดลอง.....   | 58 |
| 5.2 ผลการทดลองการประมาณค่าแรงบิดแบบออฟไลน์.....   | 58 |

|  |    |
|--|----|
| 5.3 ผลการทดลองการประมาณค่าแรงบิดแบบออนไลน์.....                        | 59 |
| 5.4 ผลการทดลองการประมาณค่าแรงบิดจากการจำลองการทำงานของแขนหุ่นยนต์..... | 60 |
| 5.5 ข้อเสนอแนะ.....  | 61 |
| 5.6 แนวทางการพัฒนาต่อไป .....  | 62 |
| บรรณานุกรม.....  | 63 |
| ภาคผนวก 1.....   | 65 |
| ประวัติผู้เขียน.....   | 68 |

## สารบัญรูปรภาพ

|             |   |    |
|-------------|---|----|
| ภาพที่ 1. 1 | โครงสร้างการควบคุมมอเตอร์ไร้แปรงถ่าน.....                               | 3  |
| ภาพที่ 1. 2 | รูปแบบเวกเตอร์จากการควบคุมด้วย Field orientation control.....           | 4  |
| ภาพที่ 1. 3 | โครงสร้างการควบคุมมอเตอร์โดยการใช้ Sliding mode observer.....           | 4  |
| ภาพที่ 1. 4 | โครงสร้างการควบคุมมอเตอร์ด้วย Field orientation control.....            | 5  |
| ภาพที่ 2. 1 | บล็อกไดอะแกรมของ Field orientation control.....                         | 11 |
| ภาพที่ 2. 2 | สัญญาณในกระบวนการของ Field orientation control.....                     | 11 |
| ภาพที่ 2. 3 | โครงสร้างของ NARX neural network [5].....                               | 13 |
| ภาพที่ 3.1  | มอเตอร์ไร้แปรงถ่าน.....   | 16 |
| ภาพที่ 3.2  | เกียร์ Harmonic ที่มีอัตราทด 100:1.....                                 | 16 |
| ภาพที่ 3.3  | บอร์ด เลือกใช้ LAUNCHXL-F28069M และ DRV8305.....                        | 17 |
| ภาพที่ 3.4  | บล็อกไดอะแกรมสำหรับการควบคุมมอเตอร์และแสดงผลต่าง ๆ.....                 | 18 |
| ภาพที่ 3.5  | การควบคุมมอเตอร์ FOC.....   | 19 |
| ภาพที่ 3.6  | กราฟแรงดันและกระแสจากมอเตอร์ขณะที่ไม่มีโหลด.....                        | 20 |
| ภาพที่ 3.7  | กราฟแรงดันและกระแสจากมอเตอร์ขณะที่มีโหลด.....                           | 20 |
| ภาพที่ 3.8  | กราฟกระแสของ $I_{\alpha}$ และ $I_{\beta}$ ของ Clarke tranformation..... | 21 |
| ภาพที่ 3.9  | กราฟกระแสของ $I_q$ และ $I_d$ ของ Clarke transformation.....             | 21 |
| ภาพที่ 3.10 | บล็อกไดอะแกรมของ FAST estimator.....                                    | 22 |
| ภาพที่ 3.11 | ไดอะแกรมของการประมาณค่าแรงบิด.....                                      | 23 |
| ภาพที่ 3.12 | บล็อกไดอะแกรมการสื่อสารระหว่างบอร์ดกับคอม.....                          | 24 |
| ภาพที่ 3.13 | ขั้นตอนการอัปเดตโหลดลงบอร์ดไมโครคอนโทรลเลอร์.....                       | 25 |
| ภาพที่ 3.14 | ขั้นตอนการบันทึกข้อมูล.....   | 26 |
| ภาพที่ 3.15 | ขั้นตอนการทำ Time series neural network 1.....                          | 27 |
| ภาพที่ 3.16 | ขั้นตอนการทำ Time series neural network 2.....                          | 27 |
| ภาพที่ 3.17 | ขั้นตอนการทำ Time series neural network 3.....                          | 28 |
| ภาพที่ 3.18 | ขั้นตอนการทำ Time series neural network 4.....                          | 28 |
| ภาพที่ 3.19 | ขั้นตอนการทำ Time series neural network 5.....                          | 29 |
| ภาพที่ 3.20 | ขั้นตอนการทำ Time series neural network 6.....                          | 29 |

|             |  |    |
|-------------|--|----|
| ภาพที่ 3.21 | ขั้นตอนการทำ Regression model 1.....                         | 30 |
| ภาพที่ 3.22 | ขั้นตอนการทำ Regression model 2.....                         | 31 |
| ภาพที่ 3.23 | ขั้นตอนการทำ Regression model 3.....                         | 31 |
| ภาพที่ 3.24 | ขั้นตอนการทำ Regression model 4.....                         | 32 |
| ภาพที่ 4.1  | อุปกรณ์สำหรับการทดลองโดยใช้เซนเซอร์แรงบิด .....              | 34 |
| ภาพที่ 4.2  | แรงบิดที่เกิดขึ้นจากการหมุนมอเตอร์.....                      | 35 |
| ภาพที่ 4.3  | อุปกรณ์สำหรับการทดลองโดยใช้ load cell.....                   | 35 |
| ภาพที่ 4.4  | แรงที่เกิดขึ้นในอุปกรณ์ที่ใช้ในการทดลอง .....                | 36 |
| ภาพที่ 4.5  | กราฟแรงบิดที่ได้จากอุปกรณ์ .....                             | 36 |
| ภาพที่ 4.6  | กระบวนการประมาณค่าแรงบิด .....                               | 37 |
| ภาพที่ 4.7  | ผลการทดลองของ neural network ที่มี TDL เท่ากับ 1 .....       | 38 |
| ภาพที่ 4.8  | ผลการทดลองของ neural network ที่มี TDL เท่ากับ 10.....       | 38 |
| ภาพที่ 4.9  | ผลการทดลองของ neural network ที่มี TDL เท่ากับ 20.....       | 39 |
| ภาพที่ 4.10 | ผลการทดลองของ linear regression .....                        | 40 |
| ภาพที่ 4.11 | ผลการทดลองของ stepwise regression .....                      | 40 |
| ภาพที่ 4.12 | ผลการทดลองของ linear SVM.....                                | 41 |
| ภาพที่ 4.13 | ผลการทดลองของ Quadratic SVM.....                             | 42 |
| ภาพที่ 4.14 | ผลการทดลองของ Medium SVM .....                               | 42 |
| ภาพที่ 4.15 | ผลของแรงบิดที่ได้จากเซนเซอร์.....                            | 44 |
| ภาพที่ 4.16 | ผลการทดลองของ neural network.....                            | 45 |
| ภาพที่ 4.17 | ผลการทดลองของ stepwise regression .....                      | 45 |
| ภาพที่ 4.18 | ผลการทดลองของ Linear regression .....                        | 46 |
| ภาพที่ 4.19 | ผลการทดลองของ Electromagnetic torque.....                    | 46 |
| ภาพที่ 4.20 | เป็นผลการเรียงข้อมูลระหว่างผลที่วัดได้กับผลจากเซนเซอร์ ..... | 46 |
| ภาพที่ 4.21 | ผลการทดลองของ neural network.....                            | 47 |
| ภาพที่ 4.22 | ผลการทดลองของ stepwise regression .....                      | 47 |
| ภาพที่ 4.23 | ผลการทดลองของ Linear regression .....                        | 48 |
| ภาพที่ 4.24 | ผลการทดลองของ Electromagnetic torque.....                    | 48 |
| ภาพที่ 4.25 | ผลการเรียงข้อมูลระหว่างผลที่วัดได้กับผลจากเซนเซอร์ .....     | 48 |

|  |    |
|--|----|
| ภาพที่ 4.26 ผลการทดลองของ neural network.....                        | 49 |
| ภาพที่ 4.27 ผลการทดลองของ Stepwise regression .....                  | 49 |
| ภาพที่ 4.28 ผลการทดลองของ Linear regression .....                    | 50 |
| ภาพที่ 4.29 ผลการทดลองของ Electromagnetic torque.....                | 50 |
| ภาพที่ 4.30 ผลการเรียงข้อมูลระหว่างผลที่วัดได้กับผลจากเซนเซอร์ ..... | 51 |

## สารบัญตาราง

|  |    |
|--|----|
| ตารางที่ 4.1 พารามิเตอร์มอเตอร์.....                                 | 34 |
| ตารางที่ 4.2 ผลการทดลองของ Time series neural network.....           | 39 |
| ตารางที่ 4.3 ผลการทดลองของ Linear regression model .....             | 41 |
| ตารางที่ 4.4 ผลการทดลองของ Support vector machines .....             | 42 |
| ตารางที่ 4.5 ผลการเปรียบเทียบของ RMSE และ R value ในแต่ละโมเดล ..... | 43 |
| ตารางที่ 4.6 ผลการเปรียบเทียบของ RSME และค่า R ของแต่ละโมเดล.....    | 52 |
| ตารางที่ 4.7 แรงบิดเท่ากับ 3 Nm .....                                | 54 |
| ตารางที่ 4.8 แรงบิดเท่ากับ 5 Nm .....                                | 55 |
| ตารางที่ 4.9 แรงบิดเท่ากับ 8 Nm .....                                | 56 |
| ตารางที่ 4.10 แรงบิดเท่ากับ 10 Nm.....                               | 57 |

## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญและที่มาของหัวข้อวิจัย

ปัจจุบันมีหุ่นยนต์มากมายหลายชนิดที่เข้ามาช่วยเหลือการทำงานของมนุษย์รวมไปถึงแขนหุ่นยนต์ซึ่งถึงเป็นหุ่นยนต์ชนิดหนึ่งที่มีบทบาทในงานทางด้านอุตสาหกรรมมากขึ้นเรื่อยๆ เพื่อใช้ในการอำนวยความสะดวกต่อการทำงานในด้านต่าง ๆ ไม่ว่าจะเป็นงานแทนแรงงานมนุษย์ที่ต้องทำงานตลอด 24 ชั่วโมง, งานที่ต้องทำซ้ำตลอดเวลา, งานที่ต้องการความละเอียดสูง, และยังสามารถทำงานได้ในที่มีสารพิษหรือในที่ที่อาจจะก่อให้เกิดอันตรายต่อมนุษย์ได้ ซึ่งข้อดีของแขนหุ่นยนต์คือมีความแน่นอน, แม่นยำ และยังมีประสิทธิภาพในการทำงานที่เหมือนเดิมตลอดการทำงาน โดยในปัจจุบันแขนหุ่นยนต์ได้มีการนำเข้ามาใช้ในด้านต่าง ๆ มากขึ้นอย่างเช่น ทางด้านการทหารหรือทางด้านการแพทย์ เป็นต้น เพื่อลดความเสี่ยงที่จะเกิดขึ้นในงานนั้น ๆ

แขนหุ่นยนต์ในปัจจุบันเริ่มมีการนำมาประยุกต์ใช้ให้เข้ากับงานที่ต้องการมากขึ้น ซึ่งแขนหุ่นยนต์ก็จะมีหลากหลายขนาดหรือขอบเขตในการทำงานที่แตกต่างกันเพื่อให้เหมาะกับงานนั้น ๆ ดังนั้นขอบเขตในการทำงาน, ขนาดของแขนหุ่นยนต์และพื้นที่สำหรับการทำงานจึงเป็นสิ่งสำคัญที่ต้องคำนึงถึง ส่วนประกอบหลักของแขนหุ่นยนต์จะประกอบไปด้วย มอเตอร์ที่ใช้สำหรับในการหมุน (ปรับเปลี่ยนลักษณะท่าทางของแขนหุ่นยนต์), ลักษณะแขนหุ่นยนต์ทางด้านเชิงกล และเซนเซอร์หลากหลายชนิดที่ใช้สำหรับกระบวนการป้อนกลับและรับรู้ เป็นต้น โดยในการออกแบบแขนหุ่นยนต์ต้องคำนึงถึงปัจจัยเพื่อให้สามารถทำงานใกล้ชิดกับมนุษย์ได้ เช่น ความสะดวกและประสิทธิภาพในการทำงาน และยิ่งไปกว่านั้นต้องคำนึงถึงความปลอดภัยในระหว่างการทำงานด้วย เพื่อไม่ให้เกิดอุบัติเหตุที่จะเกิดขึ้นในระหว่างการทำงาน ดังนั้นเซนเซอร์ต่าง ๆ ในแขนหุ่นยนต์จึงเป็นปัจจัยสำคัญต่อการควบคุมแขนหุ่นยนต์ คือ ความเร็วในการหมุน, ตำแหน่ง, และแรงบิดของแขนหุ่นยนต์เป็นอย่างน้อย โดยในปกติเซนเซอร์ต่าง ๆ จะต้องอยู่ในทุก ๆ จุดที่มีการเคลื่อนไหวของแขนหุ่นยนต์ แต่การที่มีเซนเซอร์จำนวนมากในแขนหุ่นยนต์จะทำให้การออกแบบทางด้านเชิงกลยังมีความซับซ้อนมากขึ้น และมีขนาดใหญ่ขึ้นไปด้วย อาจจะทำให้ไม่เหมาะกับการทำงานที่มีพื้นที่การทำงานน้อย เนื่องการที่มีจำนวนของเซนเซอร์ที่มากจะส่งผลต่อการอ่านค่าและอาจจะส่งผลถึงประมวลผลทำให้เกิดความล่าช้าในการตอบสนองของแขนหุ่นยนต์และยิ่งไปกว่านั้นจะทำให้มีราคาที่สูงขึ้นไปด้วย ในงานวิจัยนี้มีความต้องการที่จะลดความซับซ้อนของแขนหุ่นยนต์และต้นทุนในการผลิตและพัฒนาแขนหุ่นยนต์ โดยหนึ่งในวิธีการลดความซับซ้อนในการออกแบบแขนหุ่นยนต์คือการลดจำนวนของเซนเซอร์ จึงมี

ความมุ่งเน้นไปที่เซนเซอร์วัดแรงบิด เนื่องจากในการออกแบบทางกลในแขนหุ่นยนต์ค่อนข้างมีข้อจำกัดและรวมไปถึงค่าอุปกรณ์ที่มีราคาสูง แต่ในการวัดแรงบิดในแขนหุ่นยนต์ก็เป็นสิ่งที่จำเป็นในเรื่องของความปลอดภัยในการทำงานร่วมกับมนุษย์ จึงเป็นที่มาของงานวิจัยที่ต้องการวัดแรงบิดโดยไม่ใช้เซนเซอร์และเป็นเหตุผลที่จะต้องใช้มอเตอร์ชนิดไร้แปรงถ่าน ( Brushless DC motor) ด้วยกระบวนการควบคุมมอเตอร์แบบไร้เซนเซอร์ (Sensorless control) แต่จะทำการอ่านค่าจากสัญญาณไฟฟ้าย้อนกลับจากมอเตอร์ โดยนำสัญญาณมาวิเคราะห์และหาความสัมพันธ์ของข้อมูลเพื่อใช้ในการควบคุมและวัดค่าแรงบิดของมอเตอร์ในการเคลื่อนที่ของแขนหุ่นยนต์ ในงานวิจัยจะให้ความสำคัญในเรื่องของการพัฒนาการประมาณค่าแรงบิด โดยใช้กระบวนการเรียนรู้โมเดลของ Machine learning ในการวิเคราะห์และศึกษาพฤติกรรมของสัญญาณที่ได้จากการทำงานของมอเตอร์เพื่อนำไปสู่การประมาณค่าแรงบิด โมเดลที่เลือกใช้จะมุ่งเน้นถึงความสะดวกต่อการใช้งานและสามารถนำมาประมวลผลบนไมโครคอนโทรลเลอร์แบบทันเวลาและที่สำคัญจะต้องพัฒนางานวิจัยบนโปรแกรม MATLAB Simulink ได้

งานวิจัยนี้มุ่งเน้นและนำเสนอการพัฒนากระบวนการประมาณค่าแรงบิดสำหรับมอเตอร์ไร้แปรงถ่านด้วยเทคนิคไร้เซนเซอร์ที่เป็นปฏิกิริยาย้อนกลับของสัญญาณไฟฟ้าจากการทำงานของมอเตอร์เพื่อนำไปสู่กระบวนการประมาณค่าแรงบิด, เปรียบเทียบการประมาณค่าแรงบิดในแต่ละโมเดลของ Machine learning กับแรงบิดที่วัดได้จริงและเวลาที่ใช้ในการประมาณค่าแรงบิด สามารถนำมาใช้งานในการวัดแรงบิดในแขนหุ่นยนต์ได้แบบทันเวลา และเวลาที่ใช้ในการประมาณค่าในแต่ละโมเดลของ Machine learning โดยการจำลองการเกิดเหตุการณ์ขึ้นเมื่อมีสิ่งกีดขวางในระหว่างการทำงานของแขนหุ่นยนต์ โดยแขนหุ่นยนต์จะสามารถรับรู้ได้เมื่อเกิดการกระทบกระแทกขึ้นและจะต้องหยุดการทำงานทันที เพื่อลดความเสี่ยงที่จะเกิดอุบัติเหตุต่าง ๆ กับมนุษย์ได้

## 1.2 การตรวจสอบเอกสาร บทความและงานวิจัยที่เกี่ยวข้อง

จากการทบทวนวรรณกรรมในงานวิจัยจะเน้นทางด้านกระบวนการควบคุมมอเตอร์ด้วยวิธีที่ไม่ใช้เซนเซอร์และจะทำการประมาณค่าแรงบิดจากสัญญาณไฟฟ้าย้อนกลับโดยใช้ Machine learning เข้ามาช่วยในการประมวลผลการประมาณค่าแรงบิดเพื่อให้ได้ประสิทธิภาพที่ดีที่สุด

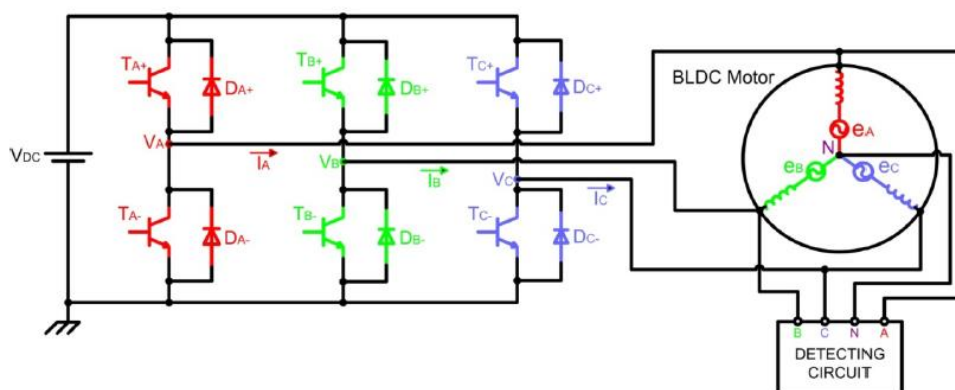
การควบคุมมอเตอร์ไร้แปรงถ่านมีหลายวิธีแต่ในงานวิจัยจะมุ่งเน้นการควบคุมด้วย Field orientation control ซึ่งจะเป็นการควบคุมแบบเวกเตอร์ โดยจะมีทั้งแบบการใช้เซนเซอร์และไม่ใช้เซนเซอร์ในการควบคุม เพื่อให้ได้ประสิทธิภาพของแรงบิดมากที่สุดโดยจะอยู่ในบทความที่ [1]-[4]



การวิเคราะห์ข้อมูลและประมาณค่าด้วย Machine learning โดยจะมีวิธี Neural network , Linear regression, Stepwise regression เพื่อจะช่วยให้ประสิทธิภาพในการประมาณค่าแรงบิดให้ดีขึ้นจะอยู่ในบทความที่ [5]-[6]

### 1.2.1 Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application Trends [1]

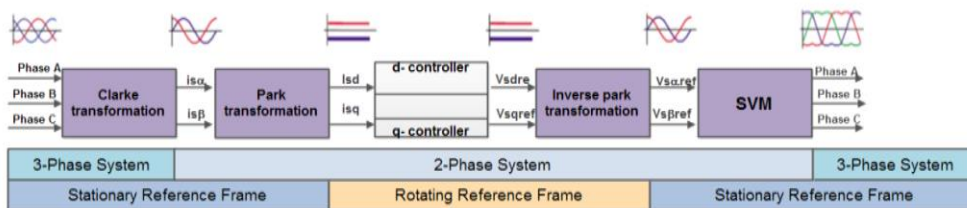
งานวิจัยนี้เป็นการรวบรวมผลงานและข้อมูลงานวิจัยต่าง ๆ ที่เกี่ยวข้องกับมอเตอร์ไร้แปรงถ่านทั้งข้อดีและข้อเสียของมอเตอร์ชนิดนี้รวมไปถึงการควบคุมมอเตอร์ไร้แปรงถ่านด้วยวิธีต่าง ๆ ทั้งที่ใช้เซนเซอร์ในการควบคุมเช่น Hall sensor, Accelerometer, Encoder และที่ไม่ใช้เซนเซอร์ในการควบคุมโดยจะใช้สัญญาณไฟฟ้าย้อนกลับจากมอเตอร์ เช่น Back-EMF Zero Crossing Detection method, Sliding-Mode Observer, Artificial Neural Networks เป็นต้น เพื่อใช้เป็นข้อมูลมาทดแทนการใช้เซนเซอร์ อย่างเช่น ตำแหน่งของโรเตอร์มอเตอร์และความเร็วในการหมุนมอเตอร์ เป็นต้น



ภาพที่ 1.1 โครงสร้างการควบคุมมอเตอร์ไร้แปรงถ่าน

### 1.2.2 A Motor controller using field oriented control and hall effect rotor position sensor: simulation and implementation [2]

งานวิจัยนี้พูดถึงการควบคุมมอเตอร์ด้วย Field orientation control ที่ใช้กับมอเตอร์ที่เป็นชนิดของซิงโครนัส (Synchronous) การควบคุมด้วยวิธีนี้จะต้องใช้ตำแหน่งของโรเตอร์มอเตอร์เข้ามาทำให้สนามแม่เหล็กของโรเตอร์และสเตเตอร์ทำตัวตั้งฉากกันเพื่อให้ได้แรงบิดที่ดีขึ้น และอธิบายถึงรูปแบบของเวกเตอร์ที่เกิดขึ้นในแต่ละโมดูลของการควบคุมด้วยวิธี Field orientation control ซึ่งจะส่งผลทำให้การควบคุมมอเตอร์มีประสิทธิภาพมากขึ้น

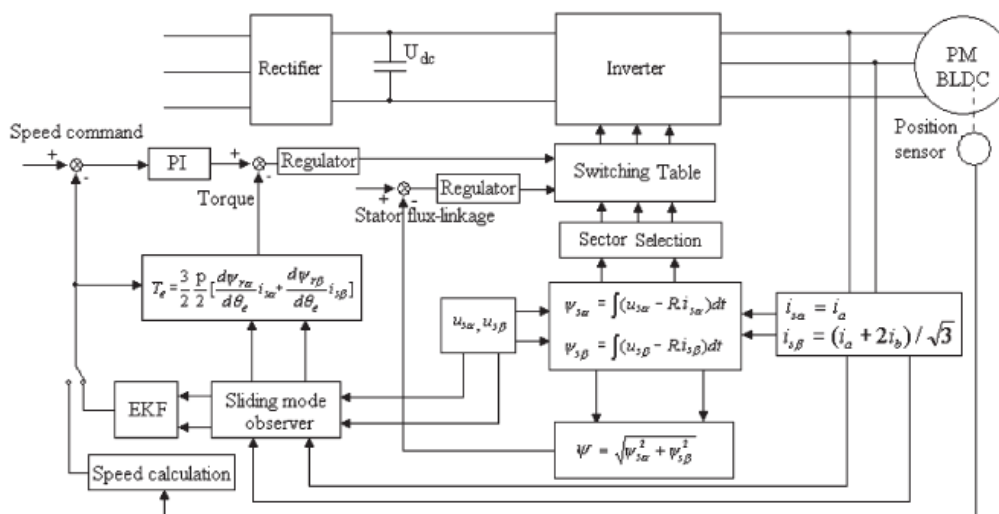


ภาพที่ 1.2 รูปแบบเวกเตอร์จากการควบคุมด้วย Field orientation control

### 1.2.3 Instantaneous Torque Estimation in Sensorless Direct-Torque-Controlled Brushless DC Motors [3]

#### Instantaneous Torque Estimation in Sensorless Direct-Torque-Controlled Brushless DC Motors [3]

งานวิจัยนี้ได้อธิบายถึงการควบคุมมอเตอร์ไร้แปรงถ่านด้วยวิธีไร้เซนเซอร์ เพื่อให้ได้ความเร็วมอเตอร์ โดยได้พูดถึงวิธี Sliding mode observer[4] เพื่อที่ต้องการรูปแบบของสัญญาณไฟฟ้าย้อนกลับ และใช้วิธี Extended Kalman filter (EKF) เพื่อให้ได้ข้อมูลของความเร็วของมอเตอร์ และในงานวิจัยนี้ยังได้พูดถึงการหาค่าแรงบิดของมอเตอร์ด้วยสมการ Electromagnetic Torque Estimation ที่ได้มาจากสัญญาณไฟฟ้าย้อนกลับของมอเตอร์



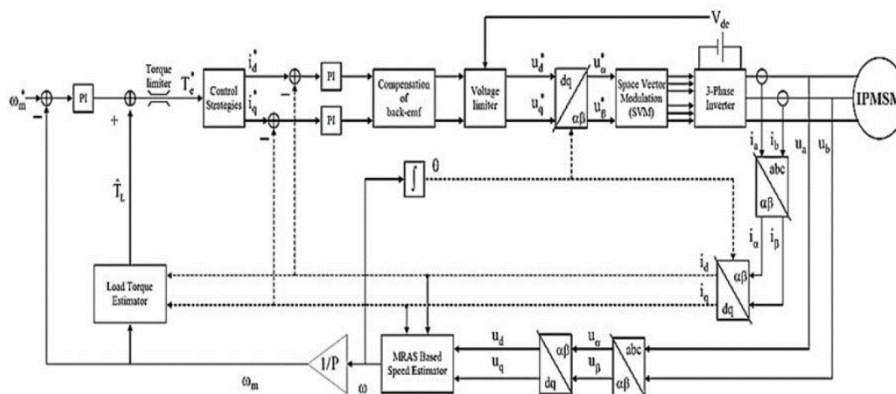
ภาพที่ 1.3 โครงสร้างการควบคุมมอเตอร์โดยการใช้ Sliding mode observer

### 1.2.4 Electromagnetic torque and speed estimators for permanent magnet synchronous motor drive system [5]

#### Electromagnetic torque and speed estimators for permanent magnet synchronous motor drive system [5]

งานวิจัยนี้ได้นำเสนอการใช้ Field orientation control ในการควบคุมมอเตอร์ไร้แปรงถ่านด้วยวิธีที่ไม่ใช้เซนเซอร์ แต่จะใช้ข้อมูลจากสัญญาณไฟฟ้าย้อนกลับและข้อมูลบาง

โมดูลจาก Field orientation control ในการประมาณค่าตำแหน่งของโรเตอร์และความเร็วของมอเตอร์ และยังอธิบายถึงการประมาณค่าแรงบิดด้วยสมการ Electromagnetic Torque



ภาพที่ 1.4 โครงสร้างการควบคุมมอเตอร์ด้วย Field orientation control

### 1.2.5 A Nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation [6]

งานวิจัยนี้ได้พูดถึงการใช้กระบวนการ Machine learning ในการประมาณค่าที่จะเกิดขึ้นในอนาคตโดยเป็นข้อมูลในรูปแบบของ Time series ที่เป็นข้อมูลที่ไม่เป็นเชิงเส้นแต่สามารถทำนายค่าได้โดยการใช้โมเดลของ Nonlinear autoregressive exogenous (NARX) ที่กระบวนการประมวลผลของ Neural network เป็นการใช้ข้อมูลในปัจจุบันร่วมกับข้อมูลในอดีตในการประมาณค่าของข้อมูลที่ต้องการในปัจจุบัน

### 1.2.6 Monitoring of induction motor load by neural network techniques [7]

งานวิจัยนี้ได้นำเสนอการใช้โมเดล Neural network ในการวิเคราะห์ข้อมูลและแสดงผลเพื่อใช้ในการประมาณค่ารูปแบบของแรงบิดที่เกิดขึ้น และได้เปรียบเทียบผลที่ได้กับสมการ Electromagnetic-Torque และหาค่าความถี่ของสเปกตรัมเพื่อตรวจจับความผิดปกติของมอเตอร์

## 1.3 วัตถุประสงค์ของโครงการวิจัย

1.3.1 เพื่อศึกษาและพัฒนาการประมาณค่าแรงบิดจากมอเตอร์โดยไม่ใช้เซนเซอร์แรงบิด

1.3.2 เพื่อเปรียบเทียบประสิทธิภาพการประมาณค่าแรงบิดด้วย Machine learning จากข้อมูลในการควบคุมมอเตอร์และประยุกต์ใช้เข้ากับแขนหุ่นยนต์

#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถประมาณค่าแรงบิดได้จากตัวแปรของมอเตอร์แทนการใช้เซ็นเซอร์แรงบิด

1.4.2 ได้ต้นแบบของแขนหุ่นยนต์ที่สามารถประมาณค่าแรงบิดโดยไม่ใช้เซ็นเซอร์แรงบิด

1.4.3 สามารถประมาณค่าแรงบิดจากมอเตอร์เมื่อแขนหุ่นยนต์เคลื่อนที่กระทบสิ่งกีดขวางระหว่างการทำงาน

1.4.4 สามารถลดต้นทุนในการผลิตแขนหุ่นยนต์โดยใช้ระบบการประมาณค่าแรงบิดแทนการใช้เซ็นเซอร์แรงบิด

#### 1.5 ขอบเขตการวิจัย

1.5.1 ศึกษาและการควบคุมมอเตอร์โดยไม่ใช้เซ็นเซอร์โดยใช้ Field orientation control

1.5.2 ศึกษาและพัฒนาการประมาณค่าแรงบิดที่เกิดขึ้นด้วยสัญญาณไฟฟ้าย้อนกลับจากการหมุนมอเตอร์

1.5.3 ประมาณค่าค่าแรงบิดในช่วงที่มีความเร็วคงที่เท่านั้น

1.5.4 พัฒนาการควบคุมมอเตอร์ไร้แปรงถ่าน (Brushless DC motor) ในการประมาณค่าแรงบิดในแขนหุ่นยนต์และแขนหุ่นยนต์จะหยุดทำงานทันทีเมื่อมีการกระทบกับวัตถุในระหว่างการทำงาน

1.5.5 ใช้โปรแกรม MATLAB และ Simulink ในการควบคุมมอเตอร์และการประมาณค่าแรงบิด

1.5.6 ใช้ Regression model และ Neural network ที่เป็นเครื่องมือของ MATLAB ในการเปรียบเทียบประสิทธิภาพการประมาณค่าแรงบิดและสามารถทำงานบนบอร์ดไมโครคอนโทรลเลอร์ได้



## 1.7 อุปกรณ์และสถานที่ทำวิจัย

1.7.1 บอร์ดไมโครคอนโทรลเลอร์ LAUNCHXL-F28069M TI

1.7.2 บอร์ดควบคุมมอเตอร์ DRV8305 TI

1.7.3 มอเตอร์ไร้แปรงถ่าน Nanotec และเกียร์ Harmonic

1.7.4 แหล่งจ่ายไฟ 24 Vdc

1.7.5 เซนเซอร์แรงบิด

1.7.6 โปรแกรม MATLAB และ Simulink

1.7.7 โปรแกรม Solidwork

1.7.8 ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

วิทยาเขตหาดใหญ่

## บทที่ 2

### ทฤษฎีและหลักการ

ในบทนี้นำเสนอทฤษฎีและหลักการที่ใช้ในงานวิจัยโดยมีวัตถุประสงค์เพื่อการออกแบบอุปกรณ์สำหรับใช้ในการทดลองและประเมินผลการประมาณค่าแรงบิดโดยใช้ข้อมูลจากสัญญาณที่ได้มาจากการทำงานของมอเตอร์เพื่อมาเปรียบเทียบกับเซนเซอร์แรงบิด ในงานวิจัยจะประกอบด้วยส่วนของฮาร์ดแวร์ที่ใช้สำหรับในการทดลองและส่วนของซอฟต์แวร์ที่เป็นการวิเคราะห์สัญญาณจากการทำงานของมอเตอร์เพื่อใช้สำหรับทำกระบวนการประมาณค่าแรงบิดโดยใช้เทคนิคของ Machine learning โดยในบทนี้จะมาอธิบายถึงองค์ประกอบในการออกแบบ, การควบคุมมอเตอร์และเทคนิคในการประมาณค่าแรงบิด

#### 2.1 ฮาร์ดแวร์สำหรับทำงานวิจัย

ในงานวิจัยมุ่งเน้นในการประมาณค่าแรงบิดสำหรับประยุกต์ใช้ในงานของแขนหุ่นยนต์ ทำให้ในการออกแบบอุปกรณ์สำหรับทำการทดลองจึงอ้างอิงการทำงานของแขนหุ่นยนต์ จึงต้องมีองค์ประกอบหลักดังนี้ มอเตอร์ที่ใช้ในการเคลื่อนที่ของแขนหุ่นยนต์, เกียร์ทดกำลังเพื่อเพิ่มอัตราทดให้กับแขนหุ่นยนต์ และเซนเซอร์ต่าง ๆ เพื่อใช้สำหรับการควบคุมมอเตอร์ของแขนหุ่นยนต์

##### 2.1.1 มอเตอร์ไร้แปรงถ่าน (BLDC motor) [8]

มอเตอร์ในปัจจุบันมีหลายชนิดที่สามารถใช้ในการสร้างแขนหุ่นยนต์โดยจะขึ้นอยู่กับขนาดของการทำงาน, พื้นที่ในการทำงานและประเภทของงาน ในงานวิจัยจะอ้างอิงกับแขนหุ่นยนต์ที่มีขนาดเล็กและสามารถทำงานได้ในพื้นที่จำกัด ดังนั้นมอเตอร์ที่ใช้จะต้องเป็นมอเตอร์ที่มีขนาดเล็กไปด้วย โดยในวิจัยจะเลือกใช้มอเตอร์ชนิดกระแสตรงและเป็นมอเตอร์ไร้แปรงถ่าน (Brushless Direct Current Motor) เนื่องจากเป็นมอเตอร์ที่เป็น 3 เฟสโดยจะมีข้อดีคือทำให้ประสิทธิภาพทางด้านแรงบิดและความเร็วได้ดีกว่ามอเตอร์ที่ใช้แปรงถ่าน, ไม่มีประกายไฟ และมีอายุการใช้งานที่นาน แต่ในการควบคุมมอเตอร์จะค่อนข้างยากเนื่องจากเป็นมอเตอร์ชนิดซิงโครนัสทำให้การควบคุมมอเตอร์ชนิดดังกล่าวจะมีเรื่องของเวลาในการควบคุมเป็นสิ่งสำคัญ โดยในการควบคุมจะต้องให้เฟสต่างกัน 120 องศา และทั้ง 3 เฟสต้องมีการจ่ายไฟที่สัมพันธ์กันโดยในการควบคุมมอเตอร์จะจ่ายไฟให้กับมอเตอร์ที่ละ 2 เฟส และสลับเฟสไปเรื่อย ๆ อย่างเช่น มอเตอร์มี 3 เฟสคือ A B และ C การควบคุมมอเตอร์จะจ่ายไฟให้เฟส A และเฟส B ส่วนเฟส C จะไม่มีการจ่ายไฟ ต่อมาจะสลับการจ่ายไฟให้กับเฟส B และเฟส C ส่วนเฟส A จะไม่มีการจ่ายไฟ จากนั้นจะจ่ายไฟให้กับเฟส C และเฟส A ส่วนเฟส B จะไม่มีการจ่ายไฟ โดยจะทำสลับกันแบบนี้ไปเรื่อย ๆ ทำให้เกิดการหมุนของ

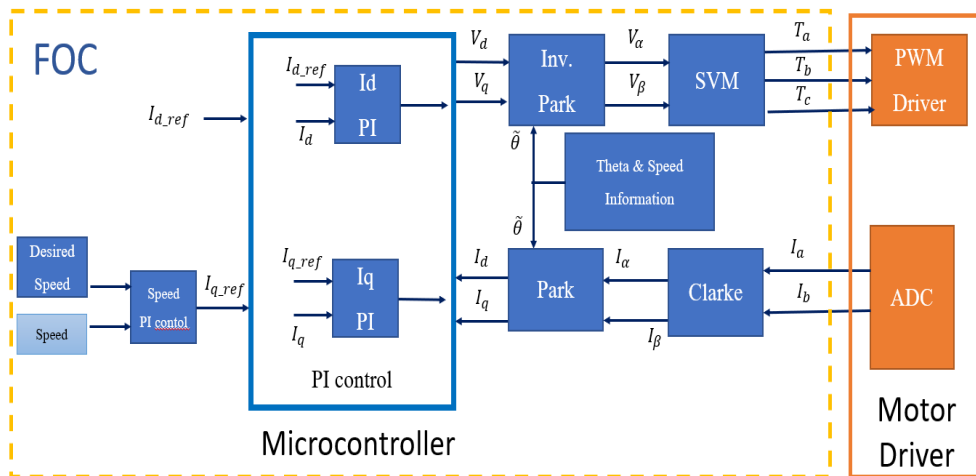
มอเตอร์ขึ้น ดังนั้นการควบคุมมอเตอร์จะใช้ 2 ใน 3 เฟสในการหมุนมอเตอร์ ทำให้ในระหว่างที่มอเตอร์หมุนเกิดสัญญาณไฟฟ้าไหลออกมาจากมอเตอร์ที่เป็น 1 เฟสที่ไม่ได้จ่ายไฟ ณ เวลานั้นจะคล้ายกับหลักการทำงานของไดนาโม โดยสัญญาณนี้จะเรียกว่าสัญญาณไฟฟ้าย้อนกลับ (Back electromotive force หรือ Back-EMF) โดยประโยชน์ของสัญญาณดังกล่าวคือสามารถนำมาวิเคราะห์เพื่อหาข้อมูลต่าง ๆ ได้ อย่างเช่น การคำนวณหาค่าแรงบิดได้

### 2.1.2 การควบคุมมอเตอร์ไร้แปรงถ่าน [2]

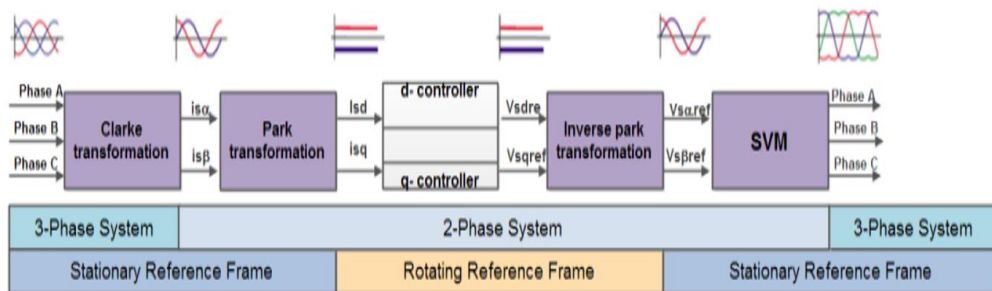
การควบคุมมอเตอร์ไร้แปรงถ่านสามารถควบคุมได้หลายวิธี อย่างเช่น Scalar control , Direct torque control (DTC) และ Field orientation control (FOC) เป็นต้น โดยในงานวิจัยจะเลือกใช้วิธีการ Field orientation control เนื่องจากเป็นการควบคุมมอเตอร์ไร้แปรงถ่านแบบเวกเตอร์โดยข้อดีของหลักการงานนี้คือสามารถให้ประสิทธิภาพของแรงบิดและความเร็วของมอเตอร์ได้ดี แต่สิ่งที่สำคัญของวิธีการนี้คือจำเป็นจะต้องใช้ตำแหน่งของโรเตอร์มอเตอร์มาช่วยในการควบคุมการหมุนมอเตอร์ ดังนั้นวิธีการนี้จะต้องมีข้อมูลป้อนกลับในการควบคุมมอเตอร์

หลักการทำงานของ Field orientation control หรือ FOC เป็นการสร้างเวกเตอร์ของสัญญาณไฟฟ้าในการควบคุมมอเตอร์ โดยใช้ข้อมูลป้อนกลับจากตำแหน่งของโรเตอร์และความเร็วของมอเตอร์ซึ่งจะเหมาะกับการควบคุมมอเตอร์โดยใช้เซ็นเซอร์ แต่ในงานวิจัยนี้มุ่งเน้นไปในการควบคุมมอเตอร์แบบไร้เซ็นเซอร์และมีข้อจำกัดในการออกแบบทางด้านฮาร์ดแวร์จึงจะใช้สัญญาณไฟฟ้าย้อนกลับจากมอเตอร์แทนการใช้เซ็นเซอร์โดยเลือกใช้อัลกอริทึมของ Sliding mode observer ในการบอกตำแหน่งและความเร็วของโรเตอร์มอเตอร์ซึ่งจะควบคุมด้วยกระบวนการของ PID ในการสร้างสัญญาณในการควบคุมมอเตอร์ จากภาพที่ 2.1 เป็นบล็อกไดอะแกรมการทำงานของ Field orientation control โดยจะรับสัญญาณไฟฟ้าย้อนกลับจากมอเตอร์ไปเข้าสู่ Clark transformation เพื่อแปลงจากสัญญาณ 3 เฟสให้อยู่ในรูปของ 2 เฟส ที่ตั้งฉากกันใน Stationary reference frame หลังจากนั้นจะเข้าสู่ Park transformation ซึ่งจะแปลงสัญญาณให้อยู่ในรูปของ Rotating reference frame โดยในขั้นตอนนี้จะต้องใช้ข้อมูลของตำแหน่งของโรเตอร์มอเตอร์สำหรับการคำนวณโดยจะได้สัญญาณออกเป็น 2 สัญญาณคือกระแสของ Direct และ Quadrature ของมอเตอร์ โดยที่สัญญาณของ Quadrature จะใช้เปรียบเทียบกับความเร็วที่ต้องการและ Direct ใช้ในแก้ปัญหาเรื่องของค่าฟลักซ์ในการควบคุมมอเตอร์ด้วยกระบวนการในการควบคุมด้วย PID จากนั้นจะเข้าสู่ Invert park transformation เพื่อแปลงสัญญาณกลับมาให้อยู่ในรูปของ Stationary reference frame และแปลงสัญญาณไปเป็น PWM ด้วย Space Vector Modulation (SVM) เพื่อใช้ในการควบคุมสัญญาณ 3 เฟสของมอเตอร์ ตามภาพที่ 2.2





ภาพที่ 2.1 บล็อกไดอะแกรมของ Field orientation control



ภาพที่ 2.2 สัญญาณในกระบวนการของ Field orientation control

### 2.1.3 การประมาณค่าแรงบิดด้วยสัญญาณไฟฟ้าย้อนกลับด้วยสมการ

#### Electromagnetic Torque [9]

จากการใช้มอเตอร์ 3 เฟสในการควบคุมมอเตอร์ด้วย Field orientation control สามารถใช้ค่าพารามิเตอร์และสัญญาณบางชนิดมาวิเคราะห์เพื่อหาค่าแรงบิดที่เกิดขึ้นของมอเตอร์ได้โดยใช้ตามสมการที่ 2.1 และ 2.2

$$\varphi_d = L_d i_d + \varphi_m \tag{2.1}$$

$$\varphi_q = L_q i_q \tag{2.2}$$

เมื่อ  $\varphi_d, \varphi_q$  คือค่าฟลักซ์ของสเตเตอร์ในแกน Direct และ Quadrature ตามลำดับ,  $\varphi_m$  คือค่าฟลักซ์ของโรเตอร์,  $i_q$  และ  $i_d$  คือสัญญาณไฟฟ้าที่ได้จาก Park transformation ในแกน Direct และ Quadrature ตามลำดับ และ  $L_d$  และ  $L_q$  เป็นค่าเหนี่ยวนำทางไฟฟ้าในแกน Direct และ Quadrature ตามลำดับ

$$T_e = \frac{3p}{2} (\varphi_d i_q - \varphi_q i_d) \quad (2.3)$$

$$T_e = \frac{3p}{2} (\varphi_m i_q + (L_d - L_q) i_q i_d) \quad (2.4)$$

$$T_e = \frac{3p}{2} (\varphi_m i_q) \quad (2.5)$$

จากสมการที่ 2.1 และ 2.2 สามารถนำไปแทนค่าในสมการของการประมาณค่าแรงบิดในสมการที่ 2.3 โดยจะได้สมการออกมาอยู่ในรูปสมการที่ 2.4 เนื่องจากมอเตอร์ชนิดไร้แปรงถ่านมีค่าเหนี่ยวนำทางไฟฟ้าของ Direct และ Quadrature ที่เท่ากัน ทำให้สามารถลดสมการลงได้ ทำให้ได้สมการสุดท้ายตามสมการที่ 2.5 โดยที่ให้  $p$  คือจำนวนโพลของมอเตอร์ และ  $T_e$  คือ แรงบิดทางไฟฟ้า

## 2.2 ซอฟต์แวร์สำหรับการทำงานวิจัย

ในงานวิจัยต้องการประมาณค่าแรงบิดจากมอเตอร์โดยใช้กระบวนการของ Machine learning โดยจะเลือกใช้โมเดลที่ติดตั้งมาพร้อมกับโปรแกรมและสามารถนำมาประยุกต์ใช้กับงานสมองฝังกล (Embedded system) ที่จะให้ความสำคัญในการทำงานลักษณะของการประมวลผลแบบทันทีโดยจะประกอบไปด้วยโมเดลของ Regression, Support vector machines และ Neural network ที่เป็นเครื่องมือในโปรแกรม MATLAB Simulink โดยโมเดลทั้งสามโมเดลจะใช้สมการของ RMSE และค่า R ในการวิเคราะห์ประสิทธิภาพในการประมาณค่าแรงบิด

### 2.2.1 Regression model

หลักการของโมเดล Regression ที่ใช้จะเป็น Linear regression [10] และ Stepwise regression [11] ซึ่งเป็นการวิเคราะห์หาความสัมพันธ์ระหว่างตัวแปร โดยจะเป็นข้อมูลทางด้านสถิติเพื่อที่จะประมาณค่าข้อมูลในอนาคต โดยจะมีเป็นความสัมพันธ์เชิงเส้นตรงระหว่างข้อมูลที่ต้องการประมาณค่าและข้อมูลที่ใช้สำหรับในการประมาณค่า

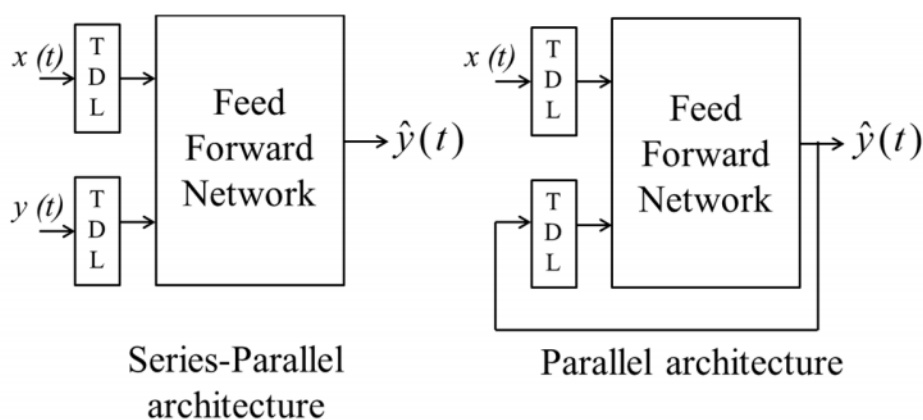
### 2.2.2 Support vector machine (SVM)

หลักการของโมเดล Support vector machine จะเป็นการวิเคราะห์ข้อมูลในเชิงของการแบ่งกลุ่มข้อมูล โดยเป็นการหาค่าสัมประสิทธิ์ของสมการเพื่อสร้างเส้นแบ่งกลุ่มข้อมูล ในการประมาณค่าจะใช้หลักของ epsilon-insensitive SVM ( $\epsilon$ -SVM) โดยเป็นสมการ

Regression ที่จะใช้ชุดข้อมูลที่มีข้อมูลที่ต้องการประมาณค่าและข้อมูลที่ใช้สำหรับการประมาณค่า โดยมีเป้าหมายในการหาฟังก์ชันที่มีค่าเบี่ยงเบนไม่เกินค่า  $\epsilon$  ในแต่ละการเรียงของข้อมูล

### 2.2.3 Neural network [6]

หลักการของโมเดล Neural network จะเป็นการนำข้อมูลระหว่างข้อมูลที่ต้องการประมาณค่าหรือข้อมูลที่ต้องการและข้อมูลที่จะใช้สำหรับการประมาณค่า โดยทั้งสองข้อมูลนี้จะได้มาจากการเก็บข้อมูลในเบื้องต้นเพื่อใช้ในการหาค่าความสัมพันธ์และปรับค่าน้ำหนักให้เหมาะสมกันจนกว่าค่าที่ได้จากการประมาณค่าจะมีค่าที่ใกล้เคียงกันกับข้อมูลที่ต้องการประมาณค่า กระบวนการ Neural network มีหลายรูปแบบ แต่ในงานวิจัยจะเลือกใช้โมเดลของ Nonlinear autoregressive with exogenous (NARX) ซึ่งจะเป็นโมเดลในรูปแบบของพลวัตหรือคล้ายกับโมเดลของ Recurrent neural network ที่จะใช้ข้อมูลที่ได้จากการประมาณค่าในอดีตมาช่วยในการประมาณค่าข้อมูลในปัจจุบันคล้ายกับการป้อนข้อมูลย้อนกลับ โดยโมเดลนี้จะแบ่งออกเป็นสองรูปแบบในการประมาณค่าคือในรูปแรกจะเป็น Series-parallel architecture คือการใช้ข้อมูลในปัจจุบันในการประมาณค่าเพียงอย่างเดียวและจะทำเพียงครั้งเดียวในตอนเริ่มต้นใช้โมเดลเท่านั้น หลังจากนั้นจะเข้าสู่รูปที่สองคือ Parallel architecture คือการนำข้อมูลที่ได้จากการประมาณค่าในรูปแรก(ผลของการประมาณค่าจะกลายเป็นข้อมูลในอดีต)มาช่วยในการประมาณค่าในปัจจุบัน ตามภาพที่ 2.3 โดยให้ TDL (Tape delay line) คือการหน่วงเวลาของข้อมูลเพื่อสะสมข้อมูลก่อนจะเข้าสู่กระบวนการเรียนรู้ (Feed forward network) ของ Neural network สำหรับใช้ในการประมาณค่าข้อมูลใน Feed Forward Network ตัวอย่าง TDL เท่ากับ 5 หมายความว่า จะเก็บข้อมูลของตัวแปรนั้นให้ครบ 5 ตัวก่อนที่จะเข้าสู่กระบวนการเรียนรู้



ภาพที่ 2.3 โครงสร้างของ NARX neural network [5]

### 2.2.4 Root mean square error (RMSE)

Root mean square error หรือ RMSE เป็นผลของลักษณะการคำนวณหา ค่าเฉลี่ยของความผิดพลาดระหว่างค่าจากข้อมูลที่ได้จากการวัดและข้อมูลที่ได้จากการประมาณค่า ตามสมการที่ 2.2 ผลการประมาณค่ายิ่งน้อยหรือเข้าใกล้ศูนย์จะยิ่งดีนั่นหมายความว่าข้อมูลจากการ ประมาณค่าจะยังมีข้อมูลที่เหมือนกับข้อมูลจริงหรือข้อมูลที่วัดได้

$$RMSE = \sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n} \quad (2.3)$$

โดยที่  $n$  เป็นจำนวนของข้อมูล ,  $\hat{y}_i$  เป็นข้อมูลจากการประมาณค่า และ  $y_i$  เป็นข้อมูลจริงหรือ ข้อมูลที่ได้จากการวัด

### 2.2.5 Regression value (R value)

Regression value หรือ R value เป็นผลของ Linear correlation coefficient ระหว่าง 2 ตัวแปรคือค่าจากข้อมูลที่ได้จากการวัดและข้อมูลที่ได้จากการประมาณค่า ตามสมการที่ 2.4 จะเป็นตัวบ่งบอกถึงค่าความแปรปรวนของข้อมูลสองข้อมูล ถ้าค่า R เข้าใกล้ 1 นั้น หมายความว่าค่าความแปรปรวนของข้อมูลที่ได้จากการประมาณค่าและข้อมูลที่ได้จากการวัดจะมีความสัมพันธ์ในทิศทางเดียวกัน แต่ถ้าเข้าใกล้ 0 ความสัมพันธ์ระหว่าง 2 ข้อมูลจะยังไม่มี ความสัมพันธ์กัน หรือในทางตรงกันข้าม R เข้าใกล้ -1 จะหมายถึงข้อมูลสองข้อมูลมีความสัมพันธ์ใน ทิศทางตรงกันข้าม

$$R = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{[n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2][n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2]}} \quad (2.4)$$

โดยที่  $n$  เป็นจำนวนข้อมูล ,  $x_i$  เป็นข้อมูลจากการวัด และ  $y_i$  เป็นข้อมูลจากการประมาณค่า

### บทที่ 3

#### การออกแบบงานวิจัยและขั้นตอนการทดลอง

ในบทนี้จะนำเสนอการออกแบบโครงสร้างของอุปกรณ์สำหรับวัดแรงบิดมอเตอร์ (Motor test bench) และการออกแบบการทดลองเพื่อวัดประสิทธิภาพของแรงบิดจากการประมาณค่าด้วยระเบียบวิธี Machine learning ในงานวิจัยจะแบ่งออกเป็น 3 ส่วน โดยจะประกอบไปด้วยการออกแบบโครงสร้างฮาร์ดแวร์สำหรับการทดลองเพื่อวัดแรงบิดของมอเตอร์, การออกแบบการทดลองสำหรับการประมาณค่าแรงบิดเพื่อเปรียบเทียบผลที่ได้ระหว่างแรงบิดจากการประมาณค่าและแรงบิดที่ได้จากการวัดด้วยเซนเซอร์ และการออกแบบการทดลองในการจำลองสถานการณ์การทำงานของแขนหุ่นยนต์ โดยในงานวิจัยจะเลือกใช้กระบวนการของ Regression model, Support vector machines และ Neural network โดยในทุกโมเดลจะให้ความสำคัญกับการประมวลผลที่ทันเวลา

#### 3.1 ออกแบบระบบฮาร์ดแวร์สำหรับการทดลองเพื่อวัดแรงบิดของมอเตอร์

ในงานวิจัยต้องการนำเสนอการประมาณค่าแรงบิดของมอเตอร์โดยไม่ใช้เซนเซอร์ในการวัด โดยใช้กระบวนการของ Machine learning เข้ามาช่วยในการประมาณค่าแรงบิด การศึกษารูปแบบของแรงบิดเพื่อใช้ในการประมาณค่าแรงบิดจำเป็นต้องมีเซนเซอร์แรงบิดเพื่อใช้ในการวัดแรงบิดของมอเตอร์เพื่อใช้เป็นข้อมูลอ้างอิง จึงจำเป็นต้องมีการออกแบบโครงสร้างเพื่อจับยึดระหว่างเซนเซอร์แรงบิดและมอเตอร์เข้าด้วยกัน ทำให้การออกแบบของโครงสร้างต้องมีการคัดเลือกอุปกรณ์ต่าง ๆ ให้เหมาะสมและไม่ซับซ้อนในการใช้งาน ซึ่งอุปกรณ์ที่เลือกใช้มีดังนี้

##### 3.1.1 มอเตอร์

ในงานวิจัยนี้เลือกใช้มอเตอร์ไร้แปรงถ่าน (BLDC motor) เนื่องจากมอเตอร์ชนิดนี้เป็นมอเตอร์ 3 เฟส ในการควบคุมมอเตอร์จะใช้เพียง 2 ใน 3 เฟสเท่านั้น ทำให้มี 1 เฟสที่ไม่ได้ใช้ในการควบคุม ณ เวลานั้น ๆ ปล่อยสัญญาณไฟฟ้าออกมา สัญญาณนี้จะเรียกว่าสัญญาณไฟฟ้าย้อนกลับหรือ Back-electromotive force (Back-emf) ที่สามารถใช้กระบวนการวิเคราะห์สัญญาณและนำไปสู่กระบวนการประมาณค่าแรงบิดได้ และให้ประสิทธิภาพของแรงบิดได้ดีกว่ามอเตอร์กระแสตรงทั่วไป



ภาพที่ 3.1 มอเตอร์ไร้แปรงถ่าน

### 3.1.2 เกียร์ทดกำลัง

เกียร์ทดกำลังเป็นอุปกรณ์เพื่อเพิ่มแรงบิดให้กับแขนหุ่นยนต์ เนื่องจากงานวิจัยได้ออกแบบการควบคุมมอเตอร์โดยการอ้างอิงและจำลองการทำงานของแขนหุ่นยนต์ ซึ่งต้องการแรงบิดที่สูงกว่าแรงบิดของตัวมอเตอร์ ในงานวิจัยได้เลือกใช้เกียร์ Harmonic [12] ที่มีอัตราทด 100:1 ซึ่งเป็นเกียร์ชนิดพิเศษที่แตกต่างจากเกียร์ชนิดอื่นคือการไม่มีช่องว่างระหว่างเกียร์หรือที่เรียกว่า Backlash ทำให้เกียร์ชนิดนี้มีความสามารถในการหมุนกลับไปกลับมาได้อย่างมีประสิทธิภาพ



ภาพที่ 3.2 เกียร์ Harmonic ที่มีอัตราทด 100:1

### 3.1.3 บอร์ดไมโครคอนโทรลเลอร์และบอร์ดควบคุมมอเตอร์

ในงานวิจัยได้เลือกใช้ LAUNCHXL-F28069M [13] และ DRV8305 [14] ตามลำดับ ซึ่งเป็นของบริษัท Texas Instrument โดยบอร์ดชนิดนี้มีอัลกอริทึมสำหรับใช้ในการศึกษาและพัฒนาในการควบคุมมอเตอร์ชนิดไร้แปรงถ่าน และที่สำคัญสามารถพัฒนาการควบคุมมอเตอร์บนโปรแกรม MATLAB Simulink ได้ทำให้เหมาะต่อการศึกษาและค้นคว้าวิจัย



ภาพที่ 3.3 บอร์ด LAUNCHXL-F28069M (ด้านล่าง) และ DRV8305 (ด้านบน)

### 3.1.4 เซนเซอร์แรงบิด

เซนเซอร์แรงบิดเป็นเครื่องมือที่ใช้สำหรับการวัดแรงบิดของมอเตอร์เพื่อใช้ในการเปรียบเทียบประสิทธิภาพของแรงบิดที่ได้จากการประมาณค่าและค่าที่วัดได้จากเซนเซอร์ ในการวัดแรงบิดจะสามารถวัดได้สองรูปแบบคือ แรงบิดแบบคงที่ (Static torque) [15] และแรงบิดแบบพลวัต (Dynamic torque) [15] โดยแรงบิดแบบคงที่จะวัดแรงบิดที่ให้เวลาเป็นค่าคงที่ ส่วนแรงบิดแบบพลวัต จะเป็นการวัดแรงบิดโดยใช้เวลาและความเร่งเข้ามาเกี่ยวข้องด้วย แต่วิธีการวัดแบบพลวัตจะมีราคาที่สูง ดังนั้นในงานวิจัยจะมุ่งเน้นไปที่เซนเซอร์วัดแรงบิดแบบคงที่เพื่อลดความซับซ้อนของงานวิจัยและลดค่าใช้จ่ายในการจัดซื้ออุปกรณ์

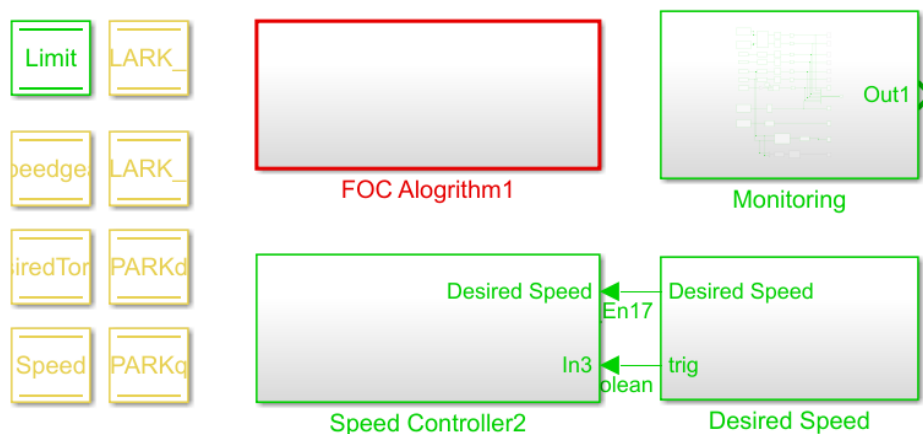
ในการออกแบบการทดลองจะอ้างอิงและจำลองการทำงานของแขนหุ่นยนต์ โดยจะเลือกใช้วัสดุที่เป็นแทนอลูมิเนียม (Aluminium profile) , พลาสติกที่เป็น PLA (Polylactic Acid) และ ABS (Acrylonitrile Butadiene Styrene) ขึ้นอยู่กับส่วนนั้น ๆ เพื่อให้ได้ความแข็งแรงและการใช้งานที่เหมาะสม เนื่องจาก 2 วัสดุที่เป็นพลาสติกจะมีคุณสมบัติที่ต่างกันคือ PLA มีความแข็งแรงเหนียวและสามารถผลิตชิ้นงานได้ง่าย ส่วน ABS จะมีความแข็งแรงแต่การผลิตชิ้นรูปจะยากกว่า ในการผลิตชิ้นรูปขึ้นงานจะผลิตด้วยการพิมพ์สามมิติด้วยเครื่อง 3D printer สำหรับแทนอลูมิเนียมใช้สำหรับโครงสร้างจับอุปกรณ์วัดแรงบิดมอเตอร์และแขนหุ่นยนต์จำลอง ในการออกแบบอุปกรณ์สำหรับวัดแรงบิดมอเตอร์จะใช้โปรแกรม Solidwork [16] เพื่อให้การขึ้นรูปโครงสร้างของอุปกรณ์มีความถูกต้องตามที่ถูกออกแบบไว้ ส่วนของการควบคุมมอเตอร์และอัลกอริทึมจะใช้โปรแกรม MATLAB & Simulink [17] เนื่องจากเป็นโปรแกรมสำหรับการคำนวณทางด้านคณิตศาสตร์และสะดวกต่อการประยุกต์ใช้งานด้านการประมวลผลสัญญาณ

### 3.2 ออกแบบการทดลองสำหรับการประมาณค่าแรงบิด

การควบคุมมอเตอร์ไร้แปรงถ่านสามารถควบคุมได้หลายวิธี ในวิจัยจะเลือกใช้การควบคุมด้วย Field orientation control จะเป็นหนึ่งในวิธีการควบคุมมอเตอร์ไร้แปรงถ่านที่สามารถให้ประสิทธิภาพของแรงบิดมากที่สุด จากภาพที่ 3.4 เป็นบล็อกไดอะแกรมที่ออกแบบบน Simulink

สำหรับการควบคุมมอเตอร์ไร้แปรงถ่าน โดยจะมีส่วนประกอบไปด้วยส่วนของการควบคุมมอเตอร์ไร้แปรงถ่านที่อยู่ในบล็อก FOC Alorithm1, ส่วนของการกำหนดความเร็วและทิศทางของการหมุนมอเตอร์อยู่ในบล็อก Desired Speed และ Speed controller2 เป็นส่วนที่รับ input จาก Desired speed และความเร็วของมอเตอร์เพื่อกำหนดค่า  $i_q$  Desired ของ FOC Alorithm1 ตามภาพที่ 3.5 และส่วนของการแสดงสัญญาณต่าง ๆ จะอยู่ในบล็อกของ Monitoring เพื่อดูพฤติกรรมของสัญญาณการประมวลผลแบบทีละเวลา

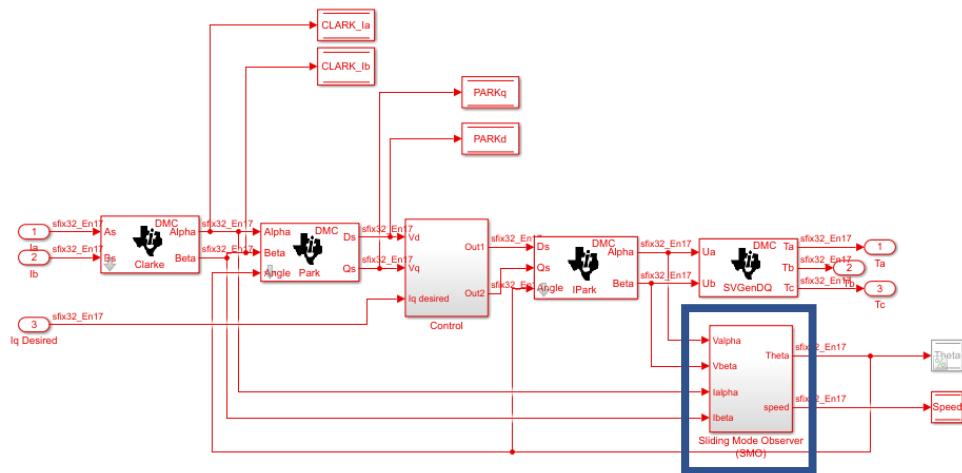
## Torque estimation in sensorless BLDC drive using machine learning



ภาพที่ 3.4 บล็อกไดอะแกรมสำหรับการควบคุมมอเตอร์และแสดงผลต่าง ๆ

จากภาพที่ 3.5 เป็นบล็อกไดอะแกรมในการควบคุมมอเตอร์ในบล็อกของ FOC Alorithm1 โดยจะเห็นได้ว่าการควบคุมมอเตอร์นี้จะใช้ตำแหน่งและความเร็วของโรเตอร์มอเตอร์สำหรับการใช้ในการป้อนกลับ (Feedback) ที่ได้มาจากสัญญาณไฟฟ้าย้อนกลับจากโมเดลของ Sliding mode observer ในกรอบสีน้ำเงินเพื่อลดความซับซ้อนในการออกแบบทางด้านฮาร์ดแวร์สำหรับการใช้ในการทดลอง





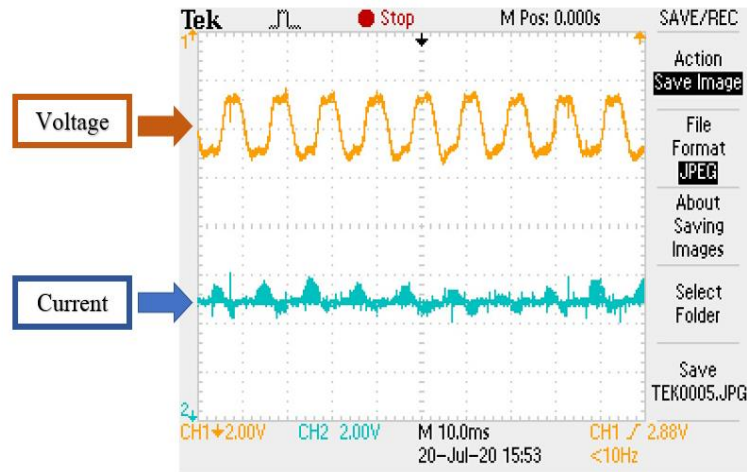
ภาพที่ 3.5 การควบคุมมอเตอร์ FOC

การประมาณค่าแรงบิดจากมอเตอร์โดยใช้ Machine Learning จะต้องให้ความสำคัญในการเลือกลักษณะเด่นเพื่อให้เหมาะสมและสามารถทำการประมวลผลแบบทันเวลาได้ แต่ถ้าใช้ลักษณะเด่นที่มากเกินไปในการทำการเรียนรู้โมเดลจะส่งผลต่อการประมวลผลแบบทันเวลา ทำให้เกิดความล่าช้าของข้อมูลในการประมาณค่า

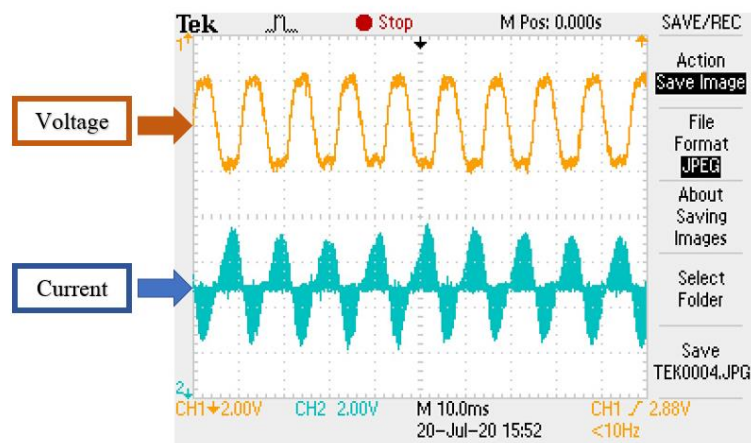
จากการศึกษาสัญญาณกระแสไฟฟ้าที่ได้มาจากสัญญาณไฟฟ้าย้อนกลับ จะเห็นว่าการเปลี่ยนแปลงรูปแบบของสัญญาณเมื่อมีการเปลี่ยนแปลงของโหลดในขณะที่มอเตอร์หมุนตามภาพที่ 3.6 และภาพที่ 3.7 โดยในภาพนี้จะเป็นสัญญาณของมอเตอร์เพียงหนึ่งเฟส แต่ในการเก็บข้อมูลจริงจะใช้ทั้งสามเฟสเพื่อให้ได้ผลประสิทธิภาพที่ดีที่สุด แต่จะมีข้อจำกัดในเก็บข้อมูลระหว่างไมโครคอนโทรลเลอร์และคอมพิวเตอร์ที่ไม่สามารถเก็บข้อมูลที่มากเกินไป เนื่องจากกระบวนการเก็บข้อมูลในโปรแกรม MATLAB Simulink จะไปรบกวนการทำงานของมอเตอร์ด้วย เนื่องจากการควบคุมมอเตอร์ชนิดนี้จะเป็นแบบซิงโครนัส

ข้อมูลที่เลือกใช้เป็นลักษณะเด่นของการเรียนรู้โมเดลจะเลือกมาจากสัญญาณที่ได้มาจากระบบการ Field orientation control ซึ่งจะเป็นสัญญาณของ Clarke transformation และ Park transformation ตามภาพที่ 3.8 และ ภาพที่ 3.9 โดยจะแสดงให้เห็นว่า Clarke transformation และ Park transformation มีความสัมพันธ์กันกับแรงบิดที่เกิดขึ้น ยกเว้น Park transformation ในมิติของ Direct หรือ  $i_d$  ที่ไม่เห็นถึงความสัมพันธ์กับแรงบิดที่เกิดขึ้น โดยเมื่อนำสัญญาณของ Clarke transformation และ Park transformation มาเปรียบเทียบกับสัญญาณทั้ง

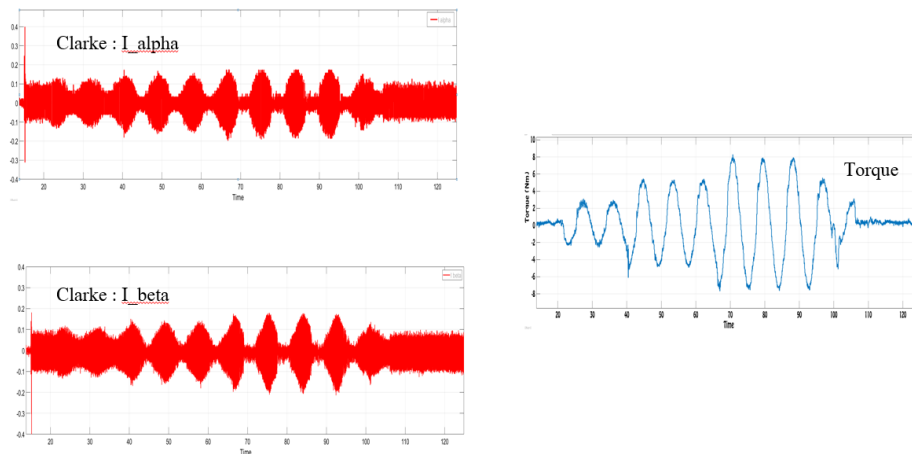
สามเฟส ( $i_a, i_b, i_c$ ) ที่ได้จากสัญญาณไฟฟ้าย้อนกลับโดยตรง สัญญาณที่ได้จาก Clarke transformation และ Park transformation จะให้สัญญาณที่ชัดเจนกว่า



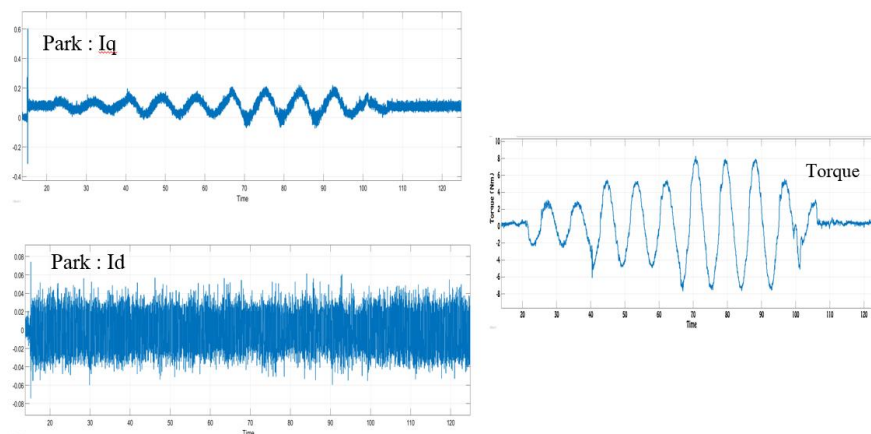
ภาพที่ 3.6 กราฟแรงดันและกระแสจากมอเตอร์ขณะที่ไม่มีโหลด



ภาพที่ 3.7 กราฟแรงดันและกระแสจากมอเตอร์ขณะที่มีโหลด



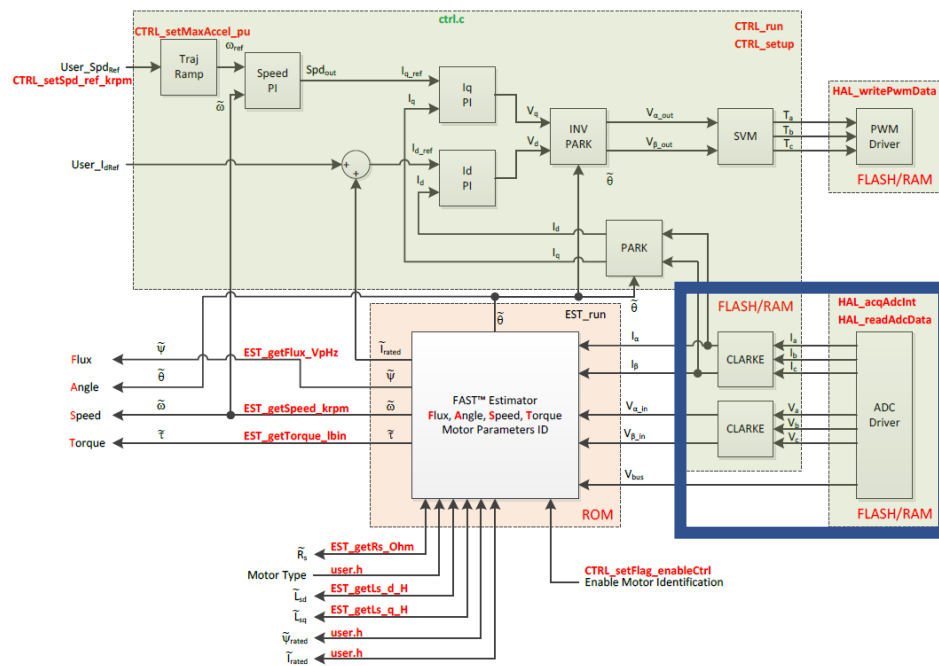
ภาพที่ 3.8 กราฟกระแสของ  $I_{\alpha}$  และ  $I_{\beta}$  ของ Clarke transformation



ภาพที่ 3.9 กราฟกระแสของ  $I_q$  และ  $I_d$  ของ Clarke transformation

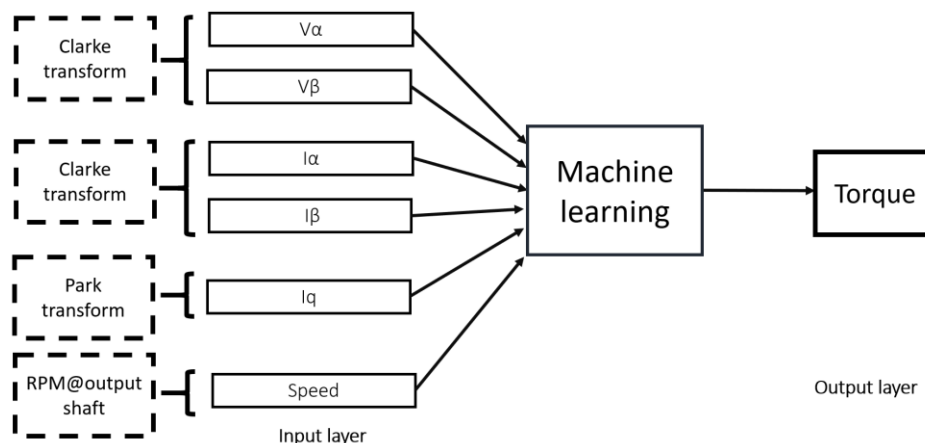
ดังนั้นข้อมูลสำหรับใช้เป็นลักษณะเด่นตอนนี้จะมีสามข้อมูลประกอบไปด้วย  $i_{\alpha}$  ,  $i_{\beta}$  และ  $i_q$  ซึ่งจะสอดคล้องกับสมการของ Electromagnetic torque ตามสมการที่ 2.5 และเพื่อเพิ่มประสิทธิภาพในการประมาณค่าให้มากขึ้นจึงได้ศึกษาสัญญาณของแรงดันด้วยโมดูลของ Clarke transformation โดยอ้างอิงมาจากกระบวนการการทำงานของ FAST estimator [18] ตามภาพที่ 3.10 ซึ่งเป็นอัลกอริทึมของบริษัท Texas Instrument ที่พัฒนาขึ้นเพื่อใช้ในการควบคุมมอเตอร์สามเฟสแบบใช้เซนเซอร์และไม่ใช้เซนเซอร์เพื่อให้เกิดการควบคุมมอเตอร์ที่ประสิทธิภาพสูง โดยมีฟังก์ชันพิเศษคือสามารถประมาณค่าฟลักซ์, ตำแหน่งของโรเตอร์ของมอเตอร์, ความเร็วของมอเตอร์ และแรงบิดของมอเตอร์ได้ ซึ่งได้ถูกพัฒนามาบนบอร์ด LAUNCHXL-F28069M แต่ฟังก์ชันนี้ไม่สามารถนำมาวิเคราะห์ทางด้านสมการได้เนื่องจากเป็นอัลกอริทึมที่ถูกปิดบังไว้ สามารถเรียกฟังก์ชันมาใช้งาน

ได้เพียงอย่างเดียว โดยในการเรียกใช้ฟังก์ชันนี้จะต้องเรียกถึงรีจิสเตอร์ที่อยู่ภายในของบอร์ด ไมโครคอนโทรลเลอร์เท่านั้น ทำให้การพัฒนาการประมาณค่าแรงบนในโปรแกรม MATLAB Simulink จะทำได้ยาก ดังนั้นในงานวิจัยไม่สามารถเรียกใช้ FAST estimator ได้ แต่ในงานวิจัยได้เลือกลักษณะเด่นจากกระบวนการนี้เพื่อใช้ในการศึกษาเพื่อพัฒนาในการประมาณค่าแรงบิดคือ  $V_\alpha$  และ  $V_\beta$  ที่ใช้ใน FAST estimator ในกรอบสีน้ำเงินของภาพที่ 3.10 และลักษณะเด่นสุดท้ายที่เลือกใช้เป็นความเร็วของเอาต์พุตเกียร์



ภาพที่ 3.10 บล็อกไดอะแกรมของ FAST estimator

ดังนั้นลักษณะเด่นสำหรับการเรียนรู้โมเดลที่เลือกใช้ในการประมาณค่าแรงบิดมีทั้งหมด 6 ลักษณะเด่นประกอบไปด้วย  $i_\alpha$ ,  $i_\beta$ ,  $V_\alpha$  และ  $V_\beta$  ที่ได้มาจาก Clarke transformation,  $i_q$  ที่ได้มาจาก Park transformation และสุดท้ายเป็นความเร็วของระบบหรือเอาต์พุตเกียร์ ตามภาพที่ 3.11 โดยทั้งหมดนี้จะนำมาวิเคราะห์เพื่อใช้ในการประมาณค่าแรงบิด



ภาพที่ 3.11 ไดอะแกรมของการประมาณค่าแรงบิด

เมื่อเลือกลักษณะเด่นครบแล้วต่อไปจะเป็นกระบวนการนำไปวิเคราะห์เพื่อใช้ในการเรียนรู้โมเดลรูปแบบสัญญาณแรงบิด จากการศึกษาพบว่าโมเดลต่าง ๆ ของ Machine learning ที่อยู่บนโปรแกรม MATLAB บางโมเดลสามารถวิเคราะห์สัญญาณและประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ได้ แต่บางโมเดลไม่สามารถประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ได้แต่สามารถวิเคราะห์สัญญาณได้เพียงอย่างเดียว ทำให้งานวิจัยได้แบ่งการทดลองออกเป็น 3 การทดลองประกอบไปด้วย 1. การประมาณค่าแรงบิดแบบออฟไลน์ 2. การประมาณค่าแรงบิดแบบออนไลน์ และ 3. การประมาณค่าแรงบิดด้วยการจำลองในเซนหุ่นยนต์

### 3.2.1 การประมาณค่าแรงบิดแบบออฟไลน์

การประมาณค่าแรงบิดแบบออฟไลน์เป็นการหาค่าประมาณแรงบิดโดยไม่ประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์แต่เป็นการประมวลผลบนคอมพิวเตอร์ส่วนบุคคลเท่านั้น โดยการเก็บข้อมูลจากการทำงานของมอเตอร์แล้วนำไปหาค่าแรงบิดด้วยกระบวนการ Machine learning จะประกอบด้วยโมเดลของ Neural network, Regression model และ Support vector machines โดยทั้งสามโมเดลเป็นเครื่องมือที่ติดตั้งมาพร้อมกับ MATLAB Simulink แต่ในบางโมเดลที่เลือกใช้ไม่สามารถประมวลผลบนบอร์ด LAUNCHXL-F28069M ได้อย่างเช่น Support vector machines และ โมเดลบางตัวของ Regression

### 3.2.2 การประมาณค่าแรงบิดแบบออนไลน์

การประมาณค่าแรงบิดแบบออนไลน์เป็นการหาค่าประมาณแรงบิดที่ใช้การประมวลผลบนบอร์ด LAUNCHXL-F28069M และจะทำงานแบบประมวลผลทันเวลา โมเดลที่สามารถประมวลผลได้นั้นมี Neural network และ Regression model โดยในการทดลองต้องการดู

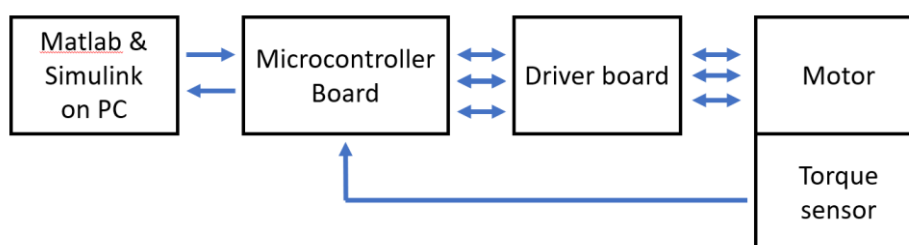
ประสิทธิภาพในการประมาณค่าแรงบิดโดยการประเมินผลการเปรียบเทียบกับแรงบิดที่วัดได้จาก เซนเซอร์ ซึ่งจะแบ่งการทดลองออกเป็น 2 การทดลองประกอบด้วย การทดลองที่ 1 เป็นการประมาณค่าแรงบิดที่มีการเปลี่ยนแปลงในรูปแบบที่คงตัวและโหลดคงที่เพื่อดูผลของการประมาณค่าและรูปแบบของแรงบิดที่เกิดขึ้น ส่วนการทดลองที่ 2 เป็นการประมาณค่าแรงบิดที่เกิดการเปลี่ยนแปลงแบบทันทีทันใดเพื่อจำลองการทำงานของแขนหุ่นยนต์เมื่อเกิดการสัมผัสเข้ากับวัตถุที่ไม่ต้องการในระหว่างการทำงาน เพื่อที่จะวัดความสามารถในการประมาณค่าแรงบิดที่เกิดขึ้นได้แบบทันทีทันใด โดยในการทดลองนี้จะเพิ่มโหลดแบบทันทีและไม่คงที่ในแต่ละครั้งที่ทำการทดลอง

### 3.2.3 การประมาณค่าแรงบิดด้วยแขนหุ่นยนต์จำลอง

เป็นการลอกเลียนแบบการเคลื่อนที่ของแขนหุ่นยนต์ เพื่อดูพฤติกรรมของการประมาณค่าแรงบิดจากการใช้งานจริง โดยเมื่อแขนหุ่นยนต์สัมผัสเข้ากับวัตถุในระหว่างที่กำลังเคลื่อนที่ แขนหุ่นยนต์จะต้องหยุดการทำงานทันทีเพื่อความปลอดภัยและลดความเสียหายที่จะเกิดขึ้น จุดประสงค์ของการทดลองนี้เพื่อวัดประสิทธิภาพของการประมาณค่าในแต่ละช่วงของแรงบิดและเพื่อวัดการตอบสนองของการทำงานในการวัดการประมาณค่าแรงบิด

### 3.3 การประมวลผลและการเก็บข้อมูลบนบอร์ดไมโครคอนโทรลเลอร์

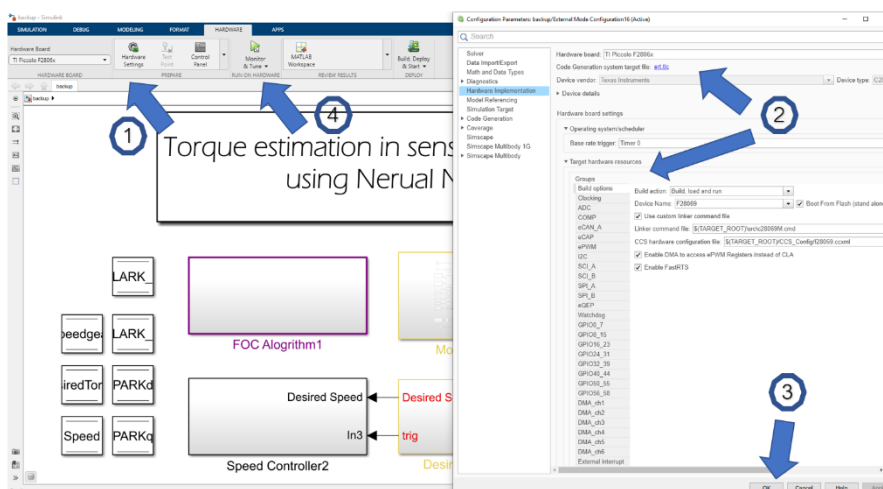
การประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ในโปรแกรม MATLAB และ Simulink โดยในการทดลองจะใช้โปรแกรม MATLAB สำหรับกำหนดค่าพารามิเตอร์ต่าง ๆ ของการควบคุมมอเตอร์และใช้สำหรับการจัดการข้อมูลต่าง ๆ ที่ได้จากการควบคุมมอเตอร์ ส่วนโปรแกรม Simulink จะใช้ในการออกแบบอัลกอริทึมในควบคุมมอเตอร์และการประมาณค่าแรงบิด จากนั้นจะประมวลผลลงบนบอร์ดไมโครคอนโทรลเลอร์ในการควบคุมและจะแสดงผลผ่านหน้าจอคอมพิวเตอร์ผ่านการสื่อสารแบบ Serial ระหว่างคอมพิวเตอร์และบอร์ดไมโครคอนโทรลเลอร์ตามภาพที่ 3.12



ภาพที่ 3.12 บล็อกไดอะแกรมการสื่อสารระหว่างบอร์ดกับคอมพิวเตอร์

ในขั้นตอนแรกต้องติดตั้งเครื่องมือของ Embedded Coder Support Package for Texas Instruments C2000 Processors สำหรับบอร์ดไมโครคอนโทรลเลอร์ในโปรแกรม

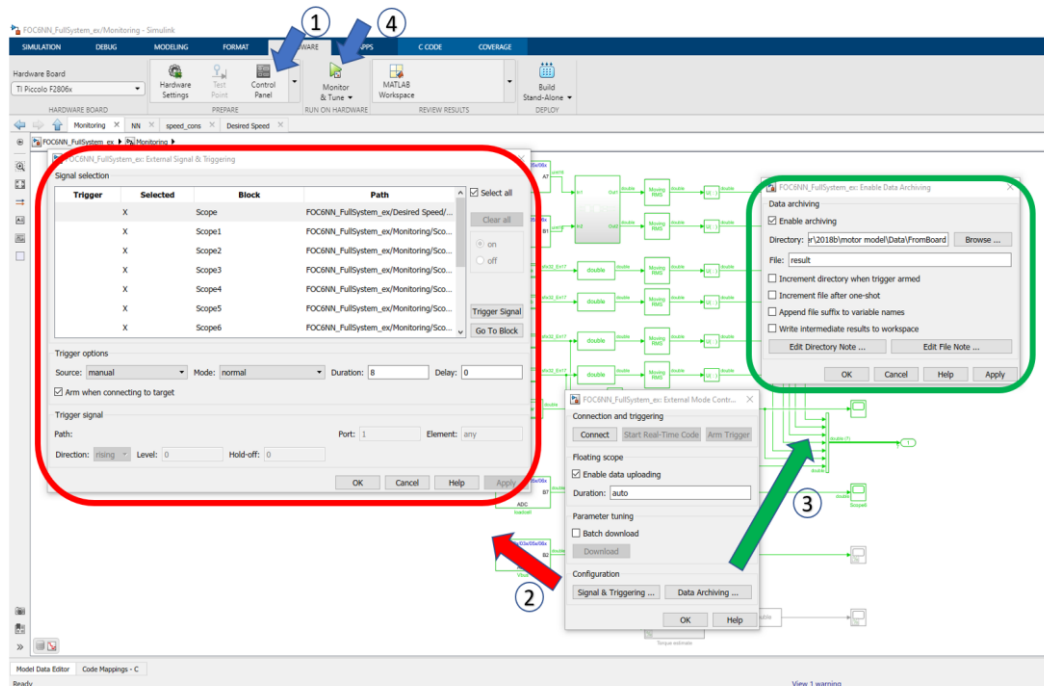
MATLAB Simulink จากนั้นพัฒนาอัลกอริทึมในการควบคุมมอเตอร์และการประมาณค่าแรงบิด และเมื่อต้องการอัปเดตอัลกอริทึมลงบนบอร์ดไมโครคอนโทรลเลอร์ให้เลือกไปที่ Hardware จากนั้นเลือก Hardware settings เพื่อตั้งค่าพารามิเตอร์ต่าง ๆ ให้ตรงกับบอร์ดไมโครคอนโทรลเลอร์ที่ใช้ จากนั้นกดไปที่ Monitor & Tune ตามภาพที่ 3.13 ตัวโปรแกรมจะทำการอัปเดตอัลกอริทึมลงบนบอร์ดและพร้อมใช้งาน เมื่อบอร์ดเริ่มทำงานจะมีการสื่อสารระหว่างบอร์ดกับโปรแกรมและแสดงข้อมูลที่ต้องการศึกษา



ภาพที่ 3.13 ขั้นตอนการอัปเดตลงบอร์ดไมโครคอนโทรลเลอร์

แต่ข้อมูลที่ได้จากการแสดงผลนั้นจะมาเป็นรูปแบบของข้อมูลสัญญาณที่เป็นลักษณะรูปภาพ ทำให้ข้อมูลนี้ไม่ใช่ข้อมูลที่เป็นตัวเลขที่สามารถบันทึกค่าได้แต่สามารถบันทึกเป็นรูปภาพได้เท่านั้น แต่ในการบันทึกเป็นรูปภาพแล้วนำมาแปลงเป็นข้อมูลตัวเลขจะทำให้ข้อมูลผิดพลาดไปรวมถึงจำนวนของข้อมูลที่ได้รับมาด้วย ทำให้ไม่สามารถนำข้อมูลที่มีการเก็บในลักษณะดังกล่าวมาใช้ในการหาค่าประมาณค่าแรงบิดได้ โดยในการเก็บข้อมูลที่เป็นตัวเลขจึงมีขั้นตอนในการเก็บข้อมูลที่แตกต่างจากอัลกอริทึมที่ใช้ทั่วไปโดยก่อนที่จะอัปเดตอัลกอริทึมลงบนบอร์ดต้องไปตั้งค่าที่ Control panel ของโปรแกรม MATLAB ก่อน โดยการเลือกที่ Signal & Triggering จากนั้นเลือกข้อมูลที่ต้องการบันทึกและตั้งค่า Duration ให้เหมาะสม และเลือกโพลเดอร์ที่ต้องการเก็บไฟล์ข้อมูลโดยการไปเลือกที่ Data archiving ของ Control panel แล้วกด Enable และเลือกที่อยู่ของโพลเดอร์ในการเก็บข้อมูล จากนั้นทำการอัปเดตอัลกอริทึมลงบนบอร์ดอีกครั้ง การตั้งค่าลักษณะดังกล่าวจะสามารถบันทึกข้อมูลสัญญาณที่ต้องการไปยังโพลเดอร์ที่ถูกตั้งค่าไว้ตามภาพที่ 3.14 ในขณะที่มอเตอร์กำลังทำงานได้ การเก็บข้อมูลลักษณะนี้จะต้องตั้งค่า Duration ในการเก็บข้อมูลให้เหมาะสม โดยการเก็บข้อมูลจะมี

ความเกี่ยวข้องกับเวลาในการทำงานของบอร์ดด้วย ซึ่งถ้าใช้ Duration ที่มากเกินไปจะทำให้เกิดปัญหาในการควบคุมมอเตอร์ด้วย เนื่องจากการเก็บข้อมูลลักษณะนี้ จะทำให้บอร์ดไมโครคอนโทรลเลอร์ต้องแบ่งเวลาในการทำงานจากการควบคุมมอเตอร์ไปเก็บข้อมูลด้วย



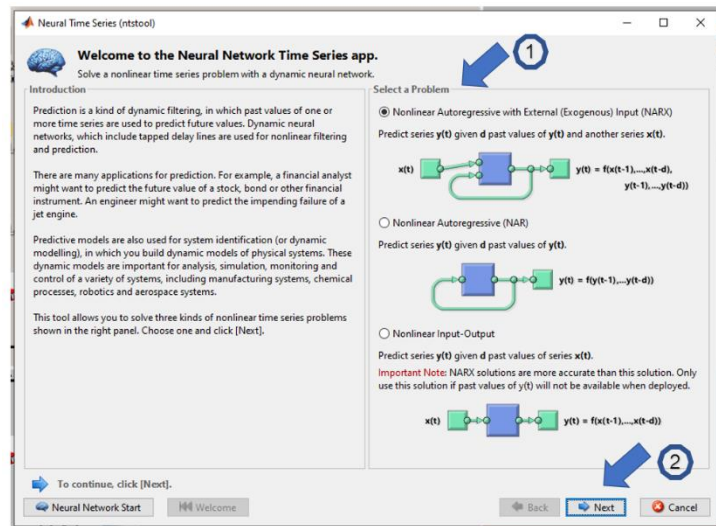
ภาพที่ 3.14 ขั้นตอนการบันทึกข้อมูล

### 3.4 การใช้ Neural network ของโปรแกรม MATLAB

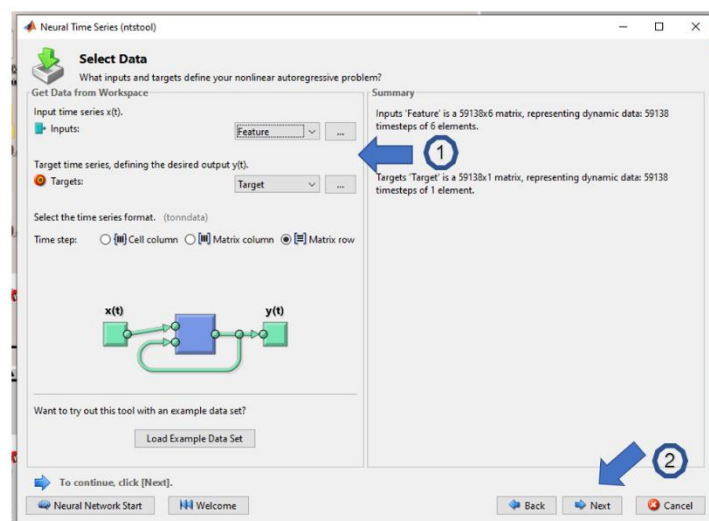
หลังจากการเก็บข้อมูลที่ต้องการและมีการจัดการข้อมูลสำหรับการทำ Machine learning เรียบร้อยแล้ว ต่อไปจะเป็นการเรียนรู้โมเดลของ Neural network เพื่อใช้ในการประมาณค่าแรงบิดของมอเตอร์โดยในขั้นตอนแรกจะต้องเตรียมข้อมูลแบ่งออกเป็นสองชุดสำหรับข้อมูลลักษณะเด่นและข้อมูลที่ต้องการประมาณค่า จากนั้นพิมพ์คำสั่ง ntstool ลงไปใน Command windows ในโปรแกรม MATLAB โดยจะมีหน้าต่างของ Time series neural network ขึ้นมาซึ่งจะมีโมเดลให้เลือก 3 โมเดลประกอบไปด้วย Nonlinear autoregressive with external (exogenous) input (NARX) , Nonlinear autoregressive (NAR) และ Nonlinear input-output ตามภาพที่ 3.15 ในงานวิจัยเลือกใช้โมเดล NARX เนื่องจากเป็นโมเดลการใช้ข้อมูลจากการประมาณค่าในอดีตมาช่วยประมาณค่าในปัจจุบันเพื่อเพิ่มประสิทธิภาพในการประมาณค่าแรงบิด เมื่อเลือกโมเดลที่ต้องการแล้วจากนั้นโปรแกรมจะให้ระบุข้อมูลของลักษณะเด่นและข้อมูลที่ต้องการประมาณค่าตามภาพที่ 3.16 และแบ่งข้อมูลสำหรับ Train, Validation และ Test ตามภาพที่ 3.17 จากนั้นโปรแกรมจะให้



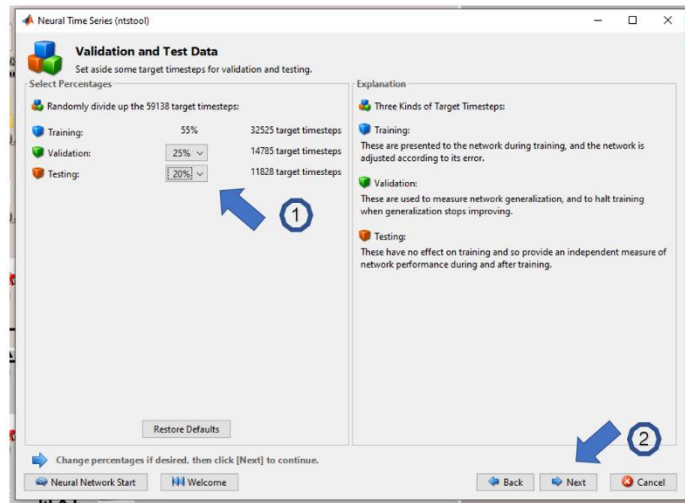
เลือก Hidden layer และ Delay ของข้อมูลสำหรับการเรียนรู้โมเดลของ Neural network ตามภาพที่ 3.18 จากนั้นเลือกอัลกอริทึมสำหรับการเรียนรู้ข้อมูลโดยจะมีให้เลือก Levenberg-Marquardt, Bayesian Regularization และ Scaled Conjugate Gradient ในแต่ละอัลกอริทึมจะเป็นการเรียนรู้โมเดลที่แตกต่างกันขึ้นอยู่กับฮาร์ดแวร์หรือคอมพิวเตอร์ที่ใช้ในกระบวนการ Machine learning จากนั้นเลือกที่ Train ตามภาพที่ 3.19 ตัวโปรแกรมจะเริ่มทำการเรียนรู้ของข้อมูลที่ได้เตรียมไว้เพื่อใช้ในการประมาณค่าแรงบิดจนสำเร็จและจะให้ผลออกมาตามภาพที่ 3.20 และเมื่อได้ผลจากการเรียนรู้ของโมเดลที่ต้องการ สามารถนำไปประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ได้โดยที่โปรแกรมจะสร้างโมเดลให้สำหรับโปรแกรม Simulink เพื่อไปใช้ในงานวิจัยต่อไปได้



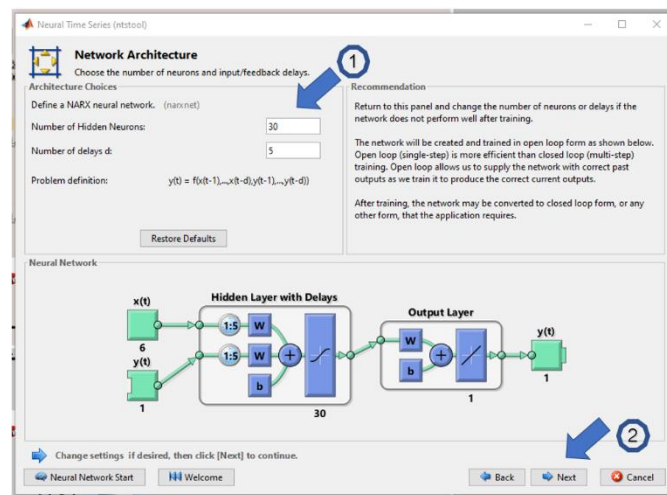
ภาพที่ 3.15 ขั้นตอนการทำ Time series neural network 1



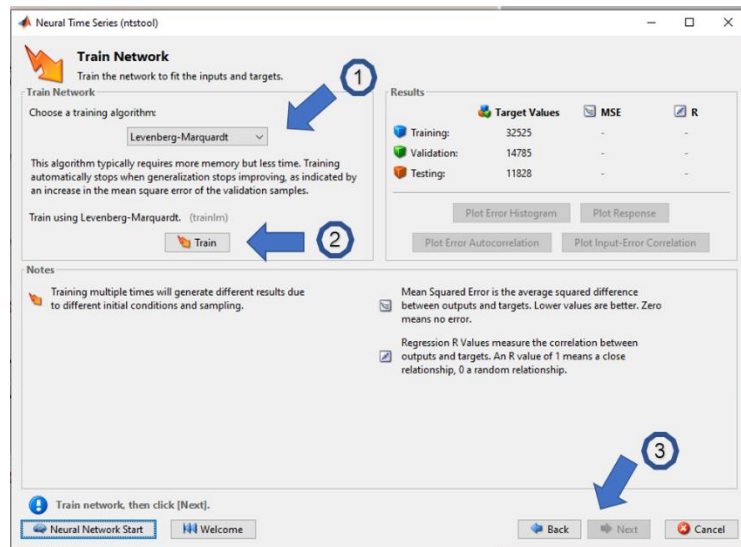
ภาพที่ 3.16 ขั้นตอนการทำ Time series neural network 2



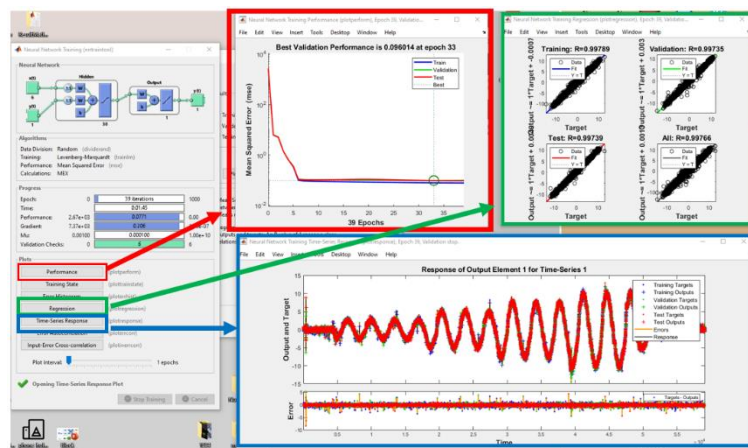
ภาพที่ 3.17 ขั้นตอนการทำ Time series neural network 3



ภาพที่ 3.18 ขั้นตอนการทำ Time series neural network 4



ภาพที่ 3.19 ขั้นตอนการทำ Time series neural network 5

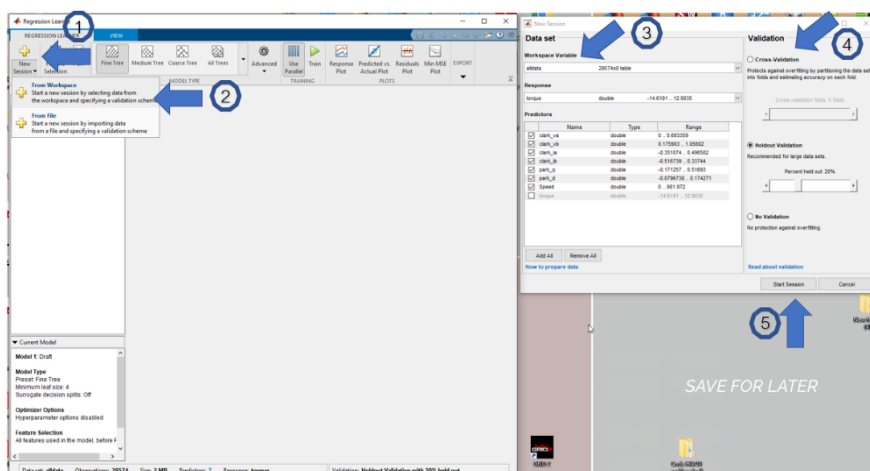


ภาพที่ 3.20 ขั้นตอนการทำ Time series neural network 6

### 3.5 การใช้งาน Regression learner ของโปรแกรม MATLAB

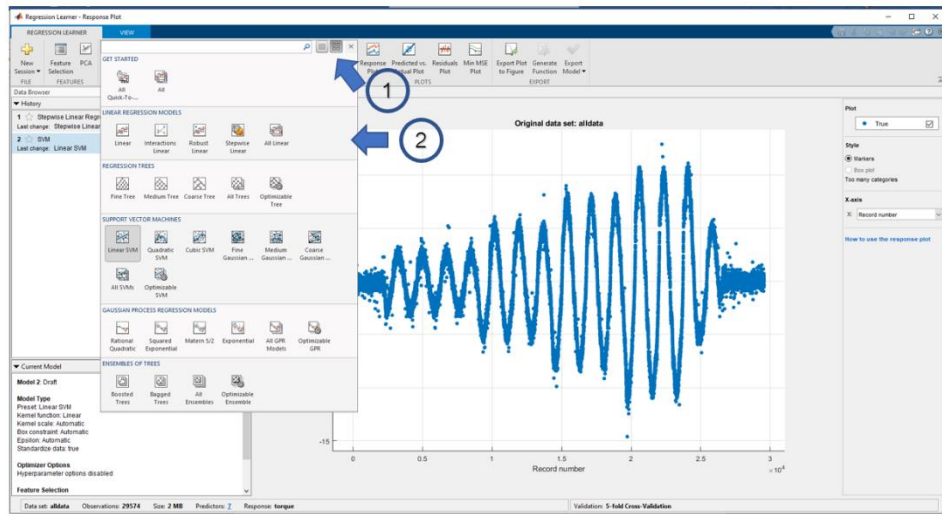
จากการเก็บข้อมูลที่แบ่งออกเป็นลักษณะเด่นและข้อมูลที่ต้องการทำการประมาณค่าเพื่อทำการเรียนรู้โมเดลด้วย Regression โดยใช้โปรแกรม MATLAB ที่มีชื่อว่า Regression Learner app ที่มีโมเดลหลากหลายโมเดลให้เลือกใช้ แต่ในงานวิจัยจะมุ่งเน้นที่โมเดลที่สามารถทำงานแบบประมวลผลทันเวลา จากการทดลองจะมี 3 โมเดลจาก Regression Learner app ที่สามารถประมวลผลทันเวลาได้ประกอบไปด้วยโมเดล Regressions , Support vector machines และ Gaussian process regression แต่มีเพียงโมเดลเดียวที่สามารถประมวลผลบนไมโครคอนโทรลเลอร์ได้คือโมเดลของ Regression โดยในขั้นตอนแรกจะกดไปยัง Regression Learner ตรง App ของ MATLAB จากนั้นเลือก New Session -> From Workspace ซึ่งข้อมูลตรง

นี้จะแตกต่างจาก Neural network คือข้อมูลของลักษณะเด่นและข้อมูลที่ต้องการประมาณค่าจะต้องอยู่รวมเป็นไฟล์เดียวกัน จากนั้นจะมีกระบวนการตรวจสอบการเรียนรู้ (Validation) ให้เลือก 3 แบบ ประกอบไปด้วย Cross-Validation, Holdout validation และ No validation เมื่อเลือกเรียบร้อยแล้วให้กดที่ Start Session ตามภาพที่ 3.21

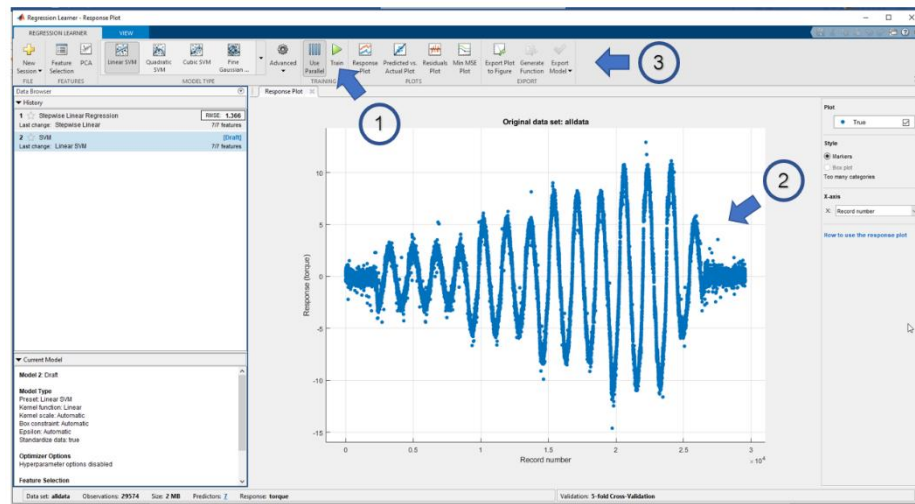


ภาพที่ 3.21 ขั้นตอนการทำ Regression model 1

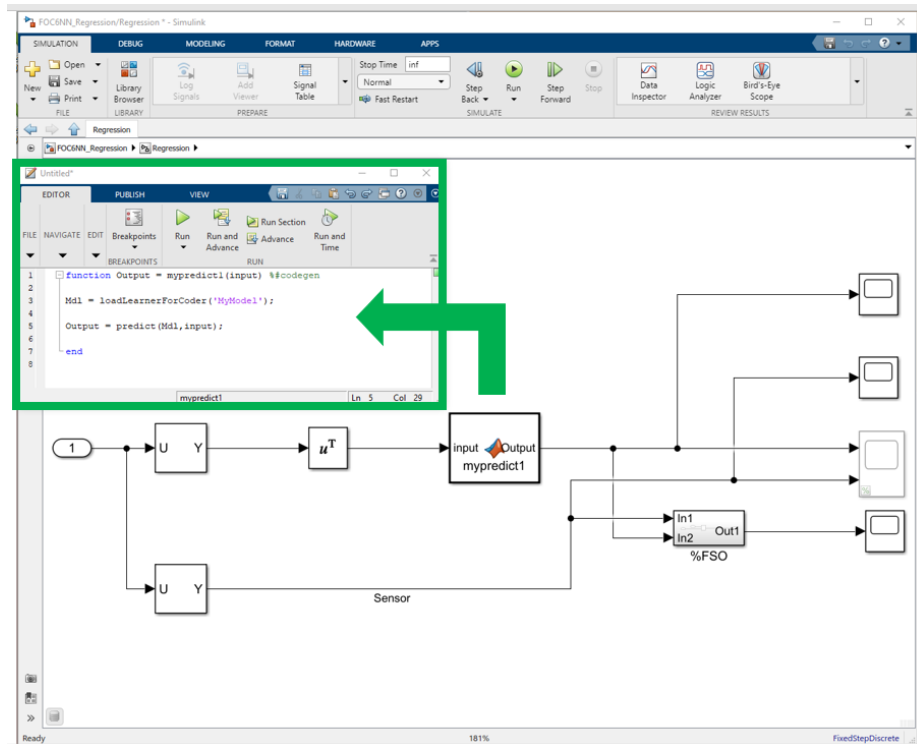
หลังจากนั้นทำการเลือกโมเดลของ Regression หรือโมเดลอื่น ๆ ที่ต้องการทำ Machine learning ตามภาพที่ 3.22 เมื่อเลือกโมเดลแล้วให้กด Train เพื่อทำการเรียนรู้ตามโมเดลนั้น ๆ ที่เลือกไว้ เมื่อเรียนรู้โมเดลนั้น ๆ เรียบร้อยแล้วจะให้ผลออกมาตามภาพที่ 3.23 และสามารถนำผลของโมเดลนี้ได้จากการเรียนรู้ที่นั้น ๆ มาประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ได้ (เฉพาะโมเดลที่สามารถนำประมวลผลลงบนบอร์ดไมโครคอนโทรลเลอร์ได้) โดยเลือกไปที่ Export model ซึ่งจะได้โมเดลที่อยู่บน Workspace จากนั้นทำการบันทึกข้อมูลเพื่อที่จะใช้สำหรับการประมาณค่าแรงบิดต่อไป โดยการใช้คำสั่ง `saveLearnerForCoder(MyModel, 'MyModel')` เมื่อต้องการทำการประมาณค่าแรงบิดบน Simulink แบบประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ จะต้องสร้าง MATLAB Function โดยเรียก `Mymodel` ที่ถูกบันทึกไว้โดยจะใช้คำสั่ง `loadLearnerForCoder('MyModel')`; ตามภาพที่ 3.24



ภาพที่ 3.22 ขั้นตอนการทำ Regression model 2



ภาพที่ 3.23 ขั้นตอนการทำ Regression model 3



ภาพที่ 3.24 ขั้นตอนการทำ Regression model 4

## บทที่ 4

### ผลการทดลองและการวิเคราะห์

บทนี้นำเสนอผลการทดลองพร้อมกับการวิเคราะห์ผลการทดลองในการประมาณค่าแรงบิด ด้วยวิธีการ Machine learning ในงานวิจัยนี้ได้แบ่งออกเป็น 3 ส่วน ประกอบไปด้วย ผลการออกแบบอุปกรณ์สำหรับการทดลอง, การทดลองในการประมาณค่าแรงบิดแบบต่าง ๆ เป็นกรณีศึกษา และการจำลองการประมาณค่าในแขนหุ่นยนต์ โดยทั้งหมดในการทดลองจะใช้โปรแกรม MATLAB Simulink ในการควบคุมมอเตอร์และพัฒนางานวิจัย ในส่วนการออกแบบอุปกรณ์สำหรับการทดลองจะใช้โปรแกรม Solidwork ในบทนี้จะนำเสนอตามหัวข้อดังนี้

#### 4.1 ผลการออกแบบอุปกรณ์สำหรับการทดลอง

#### 4.2 ผลการทดลองการประมาณค่าแรงบิด

##### 4.2.1 ผลการทดลองการประมาณค่าแรงบิดแบบออฟไลน์

- โหลดคงที่

##### 4.2.2 ผลการทดลองการประมาณค่าแรงบิดแบบออนไลน์

- โหลดคงที่
- โหลดแบบทันทีทันใด

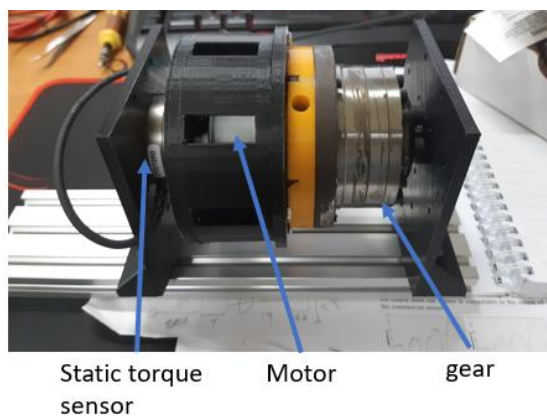
#### 4.3 ผลการทดลองของการประมาณค่าแรงบิดด้วยการจำลองในรูปแบบของแขนหุ่นยนต์

##### 4.1 ผลการออกแบบอุปกรณ์สำหรับการทดลอง

ผลการออกแบบอุปกรณ์สำหรับการทดลอง (Motor test bench) ที่มุ่งเน้นในการศึกษาการประมาณค่าแรงบิดเพื่อใช้สำหรับศึกษาพฤติกรรมของข้อมูลจากมอเตอร์และใช้สำหรับการเก็บข้อมูลเพื่อทำการเรียนรู้โมเดลต่อไป โดยในการออกแบบจะอ้างอิงการทำงานของแขนหุ่นยนต์ที่มีองค์ประกอบของมอเตอร์และเกียร์ทดกำลัง โดยพารามิเตอร์ของมอเตอร์ที่ใช้เป็นไปตามตารางที่ 4.1 ทำให้เหมาะต่อการนำมาใช้งานในแขนหุ่นยนต์ ประโยชน์ของเกียร์ทดกำลังช่วยเพิ่มแรงบิดและทำให้แขนหุ่นยนต์เคลื่อนที่ในความเร็วที่ต่ำ ในขณะที่เดียวกันจะทำให้ค่าสัญญาณไฟฟ้าย้อนกลับได้สัญญาณที่ชัดเจนมากขึ้น

ตารางที่ 4.1 พารามิเตอร์มอเตอร์

| Motor parameter            | Value   | Unit      |
|----------------------------|---------|-----------|
| Pole                       | 8       | Pole pair |
| Rated speed                | 4840    | RPM       |
| Stator resistance          | 0.43    | $\Omega$  |
| Stator inductance          | 0.00014 | H         |
| Rated voltage              | 24      | V         |
| Rated torque               | 0.13    | Nm        |
| Gear ratio (Harmonic gear) | 100:1   | -         |

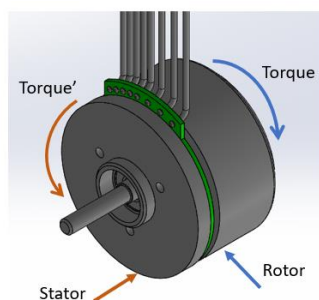


ภาพที่ 4.1 อุปกรณ์สำหรับการทดลองโดยใช้เซนเซอร์แรงบิด

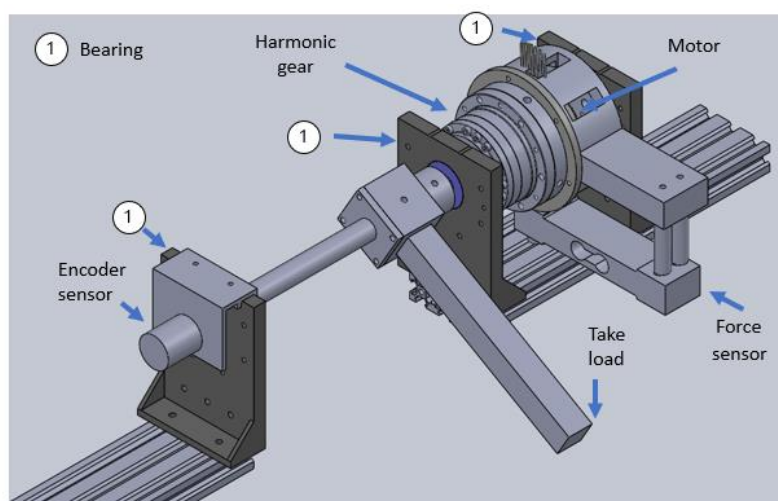
จากภาพที่ 4.1 เป็นการออกแบบอุปกรณ์สำหรับการทดลองโดยใช้เซนเซอร์แรงบิดโดยเฉพาะของ ATI mini40[19] ของบริษัท ATI Industrial Automaton โดยในการออกแบบจะใช้แรงบิดที่เกิดขึ้นที่ตัวสเตเตอร์ของมอเตอร์ที่ติดเข้ากับตัวเซนเซอร์แรงบิด ซึ่งเป็นผลที่เกิดจากผลกระทบของตัวโรเตอร์ที่ต่ออยู่กับโหลดตามภาพที่ 4.2 โดยในการออกแบบจะใช้แผ่นพลาสติก ABS (สีเหลือง) ได้จากการออกแบบด้วยโปรแกรม Solidwork และขึ้นรูปด้วยกระบวนการพิมพ์สามมิติมาประกบเข้ากับเกียร์ Harmonic ที่ใช้นี้อัตยัดผ่านแผ่นพลาสติกสีเหลืองเข้ากับมอเตอร์ และออกแบบ



กล่องบรรจุของมอเตอร์ที่เป็นพลาสติก ABS (สีดำ) เพื่อใช้จับยึดระหว่างเกียร์กับเซนเซอร์แรงบิดเข้าด้วยกัน (พลาสติก ABS เป็นพลาสติกที่มีความแข็งแรงสูงและทนความร้อน เมื่อเทียบกับพลาสติกชนิดอื่น) แต่เมื่อเริ่มทำการทดลองพบว่าค่าที่ใช้ในการวัดแรงบิดของเซนเซอร์ไม่เพียงพอต่อการใช้งาน และตัวอุปกรณ์มีราคาที่สูง จึงจำเป็นต้องออกแบบใหม่เพื่อเพิ่มประสิทธิภาพในการวัดแรงบิด



ภาพที่ 4.2 แรงบิดที่เกิดขึ้นจากการหมุนมอเตอร์

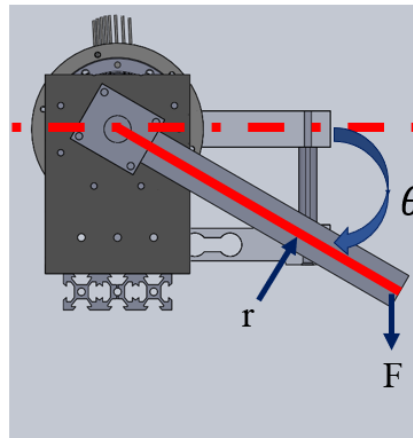


ภาพที่ 4.3 อุปกรณ์สำหรับการทดลองโดยใช้ Load cell

จากการออกแบบอุปกรณ์วัดแรงบิดด้วยเซนเซอร์แรงบิดโดยเฉพาะไม่สามารถใช้งานได้จึงเปลี่ยนอุปกรณ์เป็น strain gauge หรือ Load cell โดยวัดแรงที่เกิดขึ้นแล้วแปลงไปเป็นแรงบิดตาม ภาพที่ 4.3 ในการออกแบบยังคงใช้หลักการเดียวกันกับการใช้เซนเซอร์แรงบิดของ ATI mini40 แต่เปลี่ยนวิธียึดใหม่เพื่อให้เหมาะสมกับ Load cell โดยเพิ่มขึ้นส่วนพลาสติกจากตัวเคสของมอเตอร์ยึดออกมาจับเข้ากับ Load cell และเพิ่มเซนเซอร์ความเร็วที่เพลาของเกียร์ Harmonic เพื่อใช้วัดความเร็วของเกียร์หรือการหมุนของแกนหุ่นยนต์ ซึ่งในการวัดแรงบิดจะเป็นไปตามสมการ 4.1

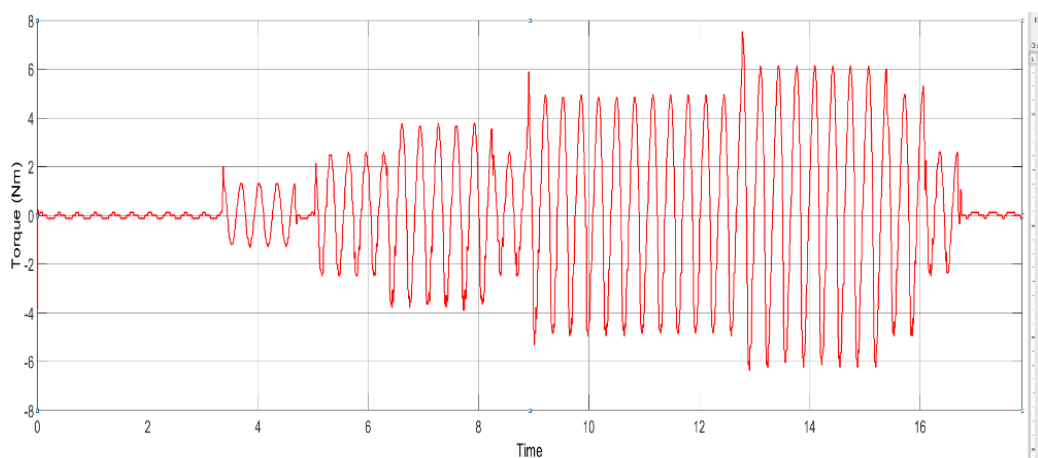
และภาพที่ 4.4 และในทุกจุดที่มีการหมุนจะใช้ตัวล็อกปืนเพื่อลดแรงเสียดทานในการหมุนเพื่อให้ได้ประสิทธิภาพของแรงบิดมากที่สุด

$$\tau = Fr\sin\theta \quad \text{----- (4.1)}$$



ภาพที่ 4.4 แรงที่เกิดขึ้นในอุปกรณ์ที่ใช้ในการทดลอง

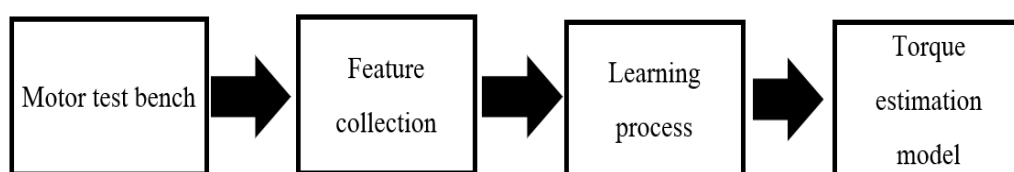
จากการทดลองของอุปกรณ์สำหรับอ่านค่าแรงบิดโดยใช้ Load cell จะได้ผลของแรงบิดตามภาพที่ 4.5 ซึ่งจะมีลักษณะคล้ายกับสัญญาณ Sinusoidal ซึ่งเป็นกระแทกจากแรงโน้มถ่วงของโลก เนื่องจากเกิดการหมุนของโหนดที่มีจุดหมุนขนานกับพื้นโลก โดยให้แกน y คือแรงบิดจากมอเตอร์ขึ้นอยู่กับโหนดที่ใช้ เมื่อมีโหนดมากขึ้นจะมียอดคลื่นที่สูงขึ้นไปด้วย ส่วนแกน x เป็นข้อมูลของเวลา โดยในงานวิจัยนี้จะอ้างอิงความเร็วของมอเตอร์ที่เหมาะสมสำหรับใช้ในงานแขนหุ่นยนต์ ในงานวิจัยนี้ได้กำหนดความเร็วในหมุนมอเตอร์อยู่ที่ 7 RPM ถึง 15 RPM ที่จุดหมุนของเกียร์



ภาพที่ 4.5 กราฟแรงบิดที่ได้จากอุปกรณ์

## 4.2 ผลการทดลองในการประมาณค่าแรงบิด

ผลการทดลองการประมาณค่าแรงบิดจากการควบคุมมอเตอร์โดยใช้ Machine learning ในการประมาณค่าแรงบิดในแขนหุ่นยนต์ โดยในการทดลองจะใช้ความเร็วคงที่ ในกระบวนการเรียนรู้เพื่อสร้างโมเดลสำหรับการประมาณค่าแรงบิดจะเริ่มจากการเก็บข้อมูล Motor test bench และทำการเลือกลักษณะเด่น เพื่อทำการเรียนรู้โมเดลจนนำไปสู่โมเดลของการประมาณค่าแรงบิดตามภาพที่ 4.6 จากนั้นนำโมเดลที่สำเร็จแล้วมาทำกระบวนการประมาณค่าแรงบิดแบบออฟไลน์และออนไลน์



ภาพที่ 4.6 กระบวนการประมาณค่าแรงบิด

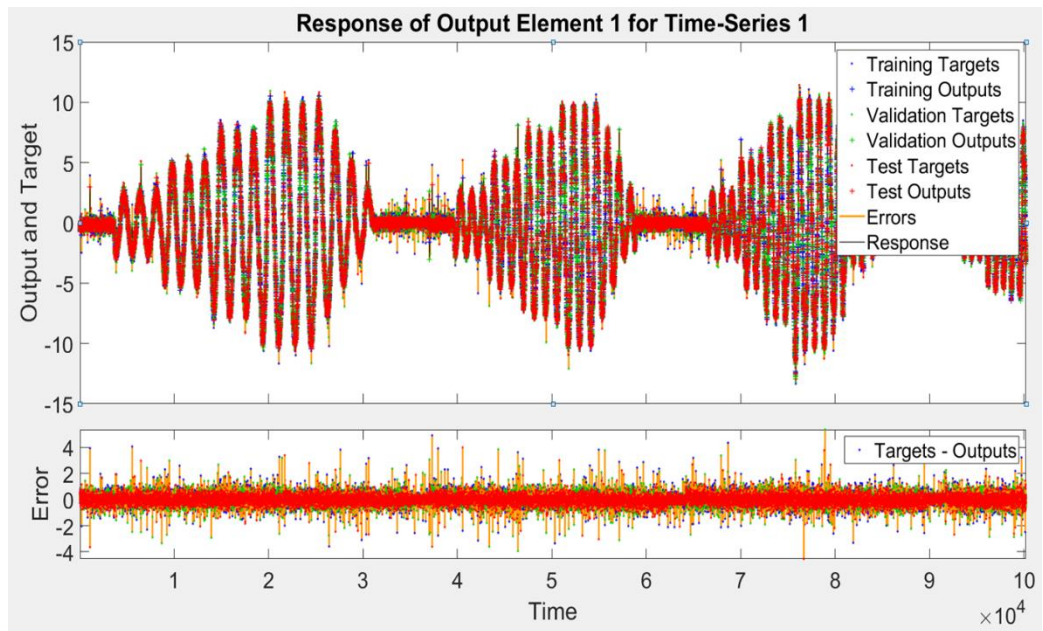
### 4.2.1 ผลการทดลองการประมาณค่าแรงบิดแบบออฟไลน์

ในการทดลองจะเลือกใช้โมเดลของ Time series neural network, Linear regression และ Support vector machines เพื่อดูผลความผิดพลาดระหว่างผลการประมาณค่าแรงบิดกับแรงบิดที่ได้จากเซนเซอร์ โดยในการประมาณค่าแรงบิดจะใช้โหลดที่ 1, 2, 3 และ 4 kg โดยเพิ่มโหลดขึ้นเมื่อหมุนครบรอบและจะทดลองในช่วงความเร็วที่ 700 , 1000 , 1200 และ 1500 RPM ของมอเตอร์

#### 4.2.1.1 Time series neural network

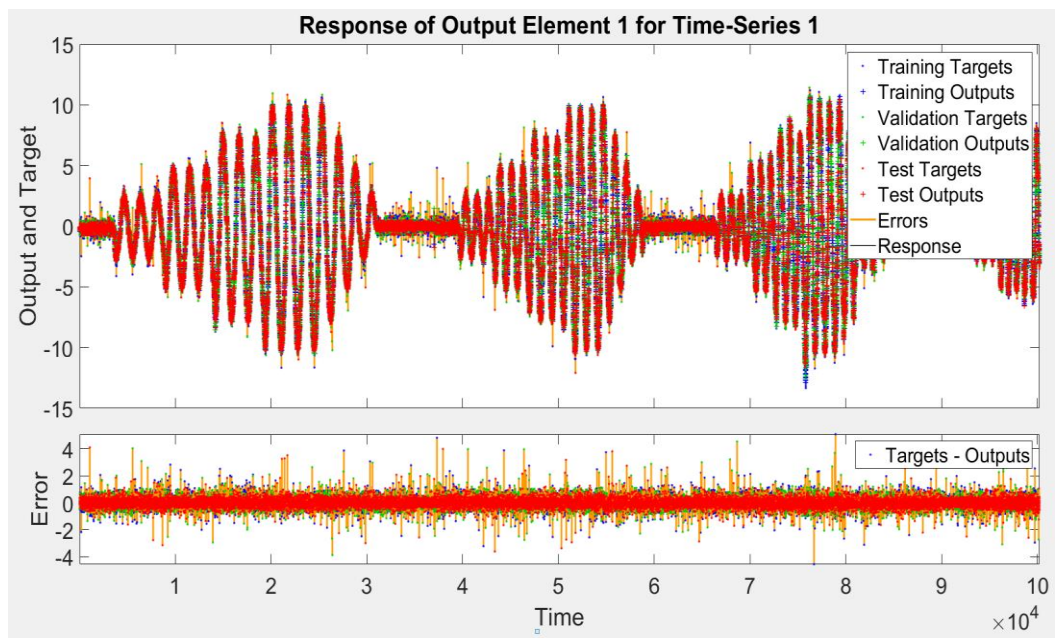
ในงานวิจัยนี้ได้เลือกใช้โมเดลของ Nonlinear autoregressive exogenous model หรือ NARX model[6] สำหรับกระบวนการประมาณค่าแบบ Neural network กำหนดให้มีจำนวน Hidden layer เท่ากับ 30 layer และใช้ฟังก์ชัน Hyperbolic sigmoid tangent เป็นแอคทีฟฟังก์ชันในงานวิจัยได้แบ่งการทดลองเป็น 4 การทดลองที่มีค่าของ TDL ที่ต่างกันคือ 1 , 5 , 10 และ 20 โดยข้อมูลของ Train , Validation และ Test ได้ถูกแบ่งสัดส่วนเป็น 55:25:20 ตามลำดับ

TDL = 1



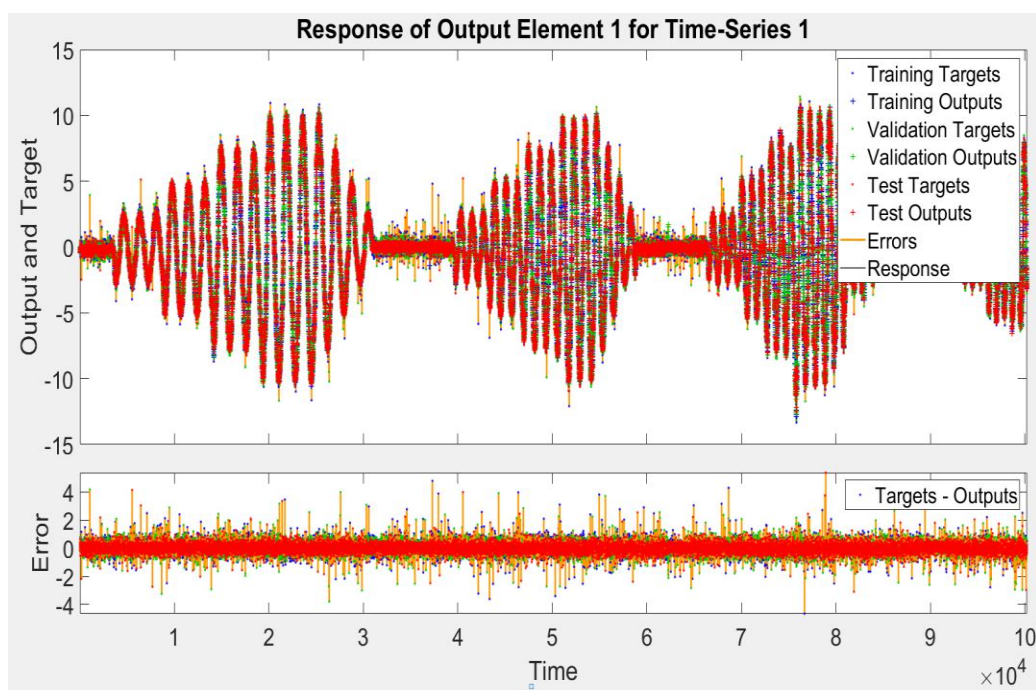
ภาพที่ 4.7 ผลการทดลองของ neural network ที่มี TDL เท่ากับ 1

TDL = 10



ภาพที่ 4.8 ผลการทดลองของ neural network ที่มี TDL เท่ากับ 10

TDL = 20



ภาพที่ 4.9 ผลการทดลองของ neural network ที่มี TDL เท่ากับ 20

ตารางที่ 4.2 ผลการทดลองของ Time series neural network

|          | RMSE  | R value | Training time (sec) |
|----------|-------|---------|---------------------|
| TDL = 1  | 0.138 | 0.995   | 22                  |
| TDL = 5  | 0.110 | 0.996   | 66                  |
| TDL = 10 | 0.106 | 0.996   | 265                 |
| TDL = 20 | 0.101 | 0.966   | 891                 |

จากตารางที่ 4.2 จะเห็นได้ว่าเมื่อมีค่า TDL ที่มากขึ้นจะทำให้ค่า RMSE ดีขึ้นด้วย แต่จะใช้เวลาในการเรียนรู้มากขึ้นด้วยเช่นกัน เมื่อคิดเป็นผลต่างของ RMSE เท่ากับ 0.038 แต่ใช้เวลาต่างกันถึง 40 เท่าเมื่อเทียบกับ TDL เท่ากับ 1 และ 20

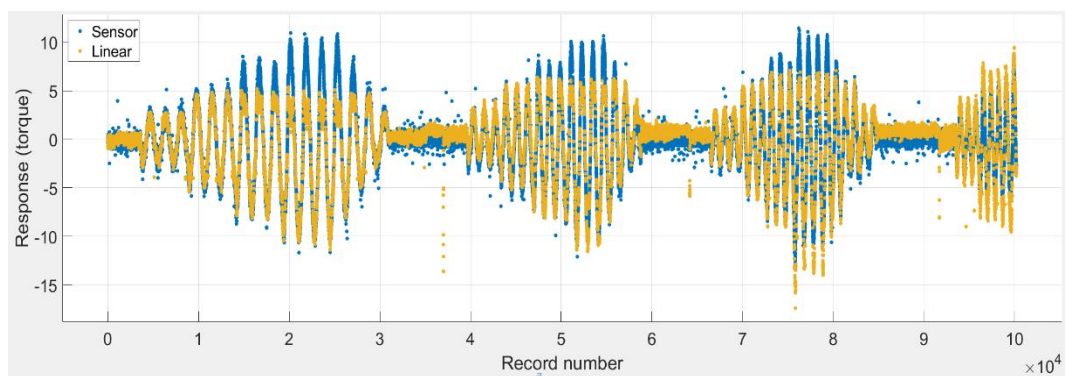
จากการทดลองที่ TDL เท่ากับ 5 ขึ้นไปจะเห็นได้ว่าค่าของ R value เท่ากันด้วย แต่ TDL ที่เท่ากับ 5 ใช้เวลาน้อยกว่า TDL ที่เท่ากับ 10 อยู่ 199 วินาที ซึ่งใช้เวลา

ในการเรียนรู้โมเดลมากกว่าเท่าตัวในการประมาณค่าแรงบิด ทำให้ TDL เท่ากับ 5 ให้ผลที่ดีที่สุดเมื่อประเมินด้วยค่า RMSE, R value และเวลาในการเรียนรู้โมเดล

#### 4.2.1.2 Regression model

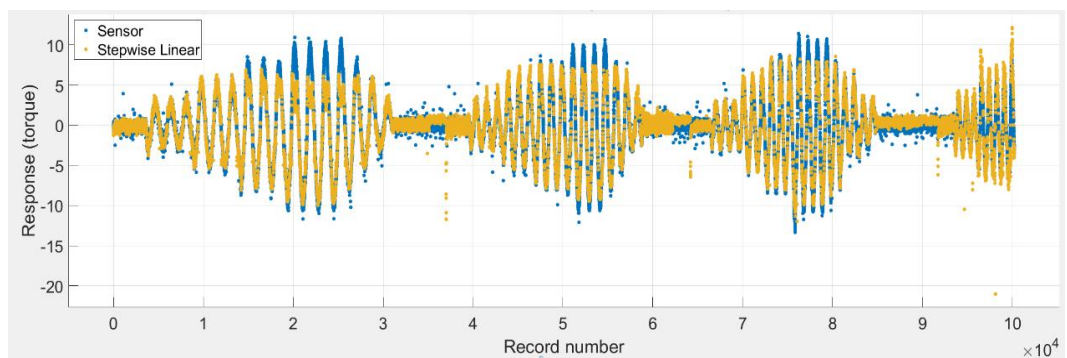
การทดลองนี้ได้เลือกโมเดลของ Linear regression และ Stepwise regression ซึ่งได้ใช้โครงสร้างในการแบ่งข้อมูลของ Train และ Test ได้ถูกแบ่งสัดส่วนเป็น 80:20 ตามลำดับ

##### Linear regression



ภาพที่ 4.10 ผลการทดลองของ linear regression

##### Stepwise regression



ภาพที่ 4.11 ผลการทดลองของ stepwise regression



ตารางที่ 4.3 ผลการทดลองของ Linear regression model

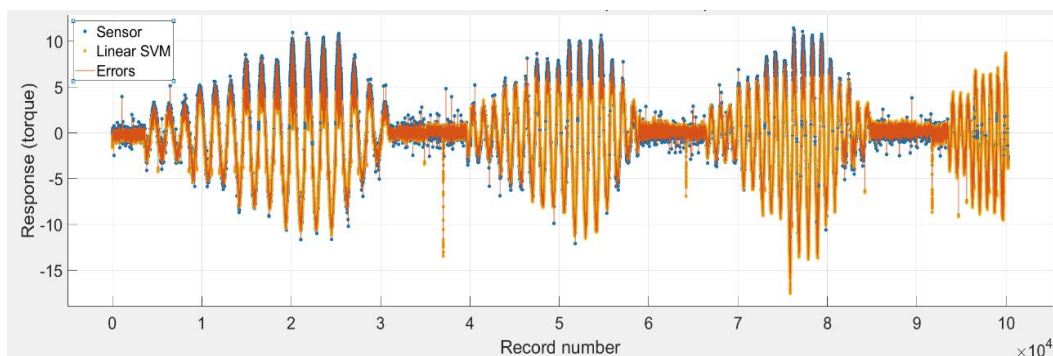
|                     | RMSE  | R value | Training time (sec) |
|---------------------|-------|---------|---------------------|
| Linear regression   | 1.385 | 0.933   | 6.134               |
| Stepwise regression | 1.177 | 0.954   | 34.435              |

จากตารางที่ 4.3 จะเห็นว่า Stepwise regression สามารถให้ผลการประเมินด้วยค่าของ RMSE และ R value ที่ดีกว่า Linear regression และเมื่อนำมาวิเคราะห์ผลแล้วจะเห็นว่า Linear regression ให้ผลที่แย่กว่า Stepwise regression อยู่ 0.208 ในขณะที่เวลาสำหรับเรียนรู้โมเดลน้อยกว่า 28.301 วินาที

#### 4.2.1.3 Support vector machines

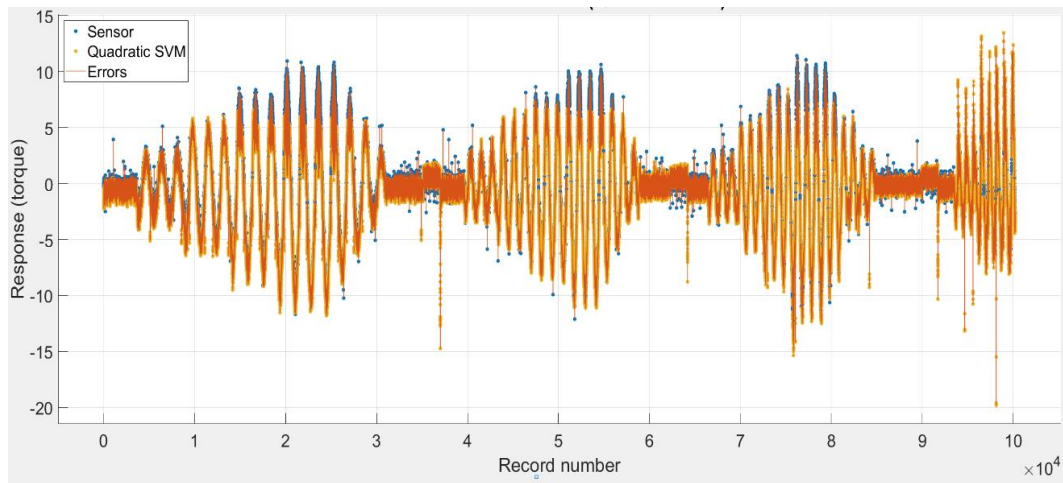
การทดลองนี้ได้เลือกโมเดลมา 3 โมเดลประกอบไปด้วย Linear SVM , Quadratic SVM และ Medium gaussian SVM โดยใช้โครงสร้างในการแบ่งข้อมูลเหมือนกับกรณีของ Regression model

##### Linear SVM



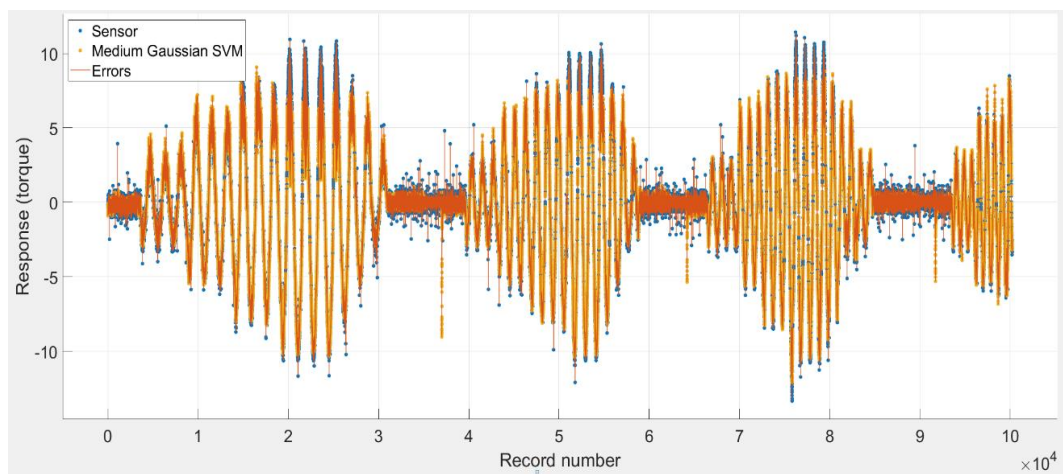
ภาพที่ 4. 12 ผลการทดลองของ Linear SVM

## Quadratic SVM



ภาพที่ 4. 13 ผลการทดลองของ Quadratic SVM

## Medium gaussian SVM



ภาพที่ 4. 14 ผลการทดลองของ Medium SVM

ตารางที่ 4.4 ผลการทดลองของ Support vector machines

|                     | RMSE  | R value | Training time (sec) |
|---------------------|-------|---------|---------------------|
| Linear SVM          | 1.452 | 0.927   | 19418               |
| Quadratic SVM       | 1.4   | 0.933   | 25076               |
| Medium gaussian SVM | 0.965 | 0.97    | 7055                |



จากตารางที่ 4.4 จะเห็นได้ว่า Medium gaussian SVM สามารถให้คำตอบได้ดีที่สุดเมื่อประเมินด้วยค่า RMSE และ R value และใช้เวลาน้อยที่สุดด้วย รองลงมาจะเป็น Quadratic SVM เมื่อเทียบกับผลของ RMSE และ R value ที่ต่างกันอยู่ที่ 0.052 และ 0.006 ตามลำดับ แต่ถ้าเทียบด้วยระยะเวลาในการประมาณค่าแรงบิดจะเป็น Linear SVM ที่ทำได้ดีกว่า วิธี Quadratic SVM ใช้เวลามากกว่า Linear SVM อยู่ถึง 5658 วินาที หรือ 1.34 ชั่วโมงในการประมาณค่า และเมื่อเทียบกับผลที่ได้ จึงสรุปได้ว่า Linear SVM ให้คำตอบและเวลาในการประมาณค่าได้ดีกว่า Quadratic SVM

ตารางที่ 4.5 ผลการเปรียบเทียบของ RMSE และ R value ในแต่ละโมเดล

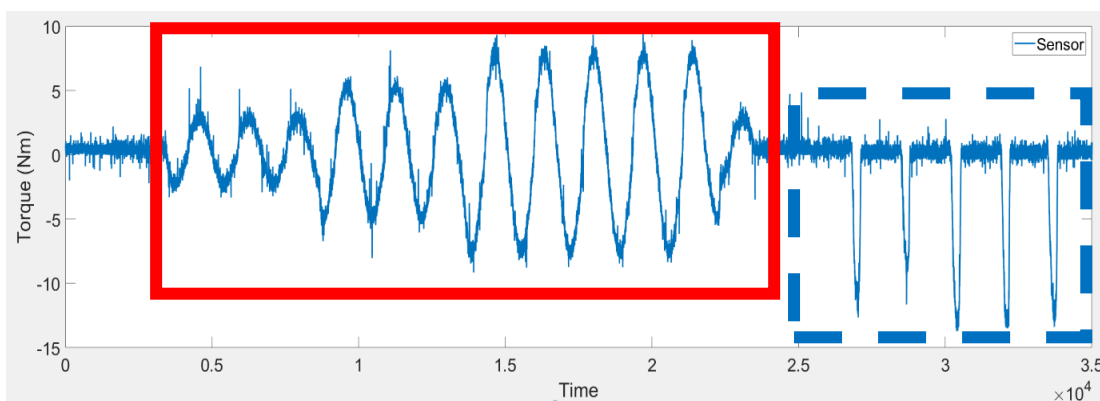
|                     |          | RMSE  | R value | Training time (sec) |
|---------------------|----------|-------|---------|---------------------|
| Neural network      | TDL = 1  | 0.138 | 0.995   | 22                  |
|                     | TDL = 5  | 0.110 | 0.996   | 66                  |
|                     | TDL = 10 | 0.106 | 0.996   | 265                 |
|                     | TDL = 20 | 0.101 | 0.966   | 891                 |
| Linear regression   |          | 1.385 | 0.933   | 6.134               |
| Stepwise regression |          | 1.177 | 0.954   | 34.435              |
| Linear SVM          |          | 1.452 | 0.927   | 19418               |
| Quadratic SVM       |          | 1.4   | 0.933   | 25076               |
| Medium gaussian SVM |          | 0.965 | 0.97    | 7055                |

จากตารางที่ 4.5 จะเห็นได้ว่าโมเดลของ Neural network ให้คำตอบได้ดีที่สุดคือมีค่าของ R value ที่ใกล้เคียง 1 ส่วนค่าของ RMSE จะดีขึ้นเมื่อมีค่าของ TDL ที่มากขึ้น แต่เมื่อเทียบเรื่องระยะเวลาในการเรียนรู้ที่ TDL เท่ากับ 5 จะให้คำตอบได้ดีที่สุด รองลงมาเป็นโมเดลของ SVM คือ Medium gaussian ที่ให้ค่า RMSE และ R value แต่ใช้เวลาในการเรียนมาก เมื่อเทียบกับโมเดลของ Regression และต่อมาจะเป็นโมเดล Stepwise regression ที่ให้คำตอบได้ดี และยังใช้เวลาในการเรียนรู้ที่น้อยที่สุด

#### 4.2.2 ผลการทดลองการประมาณค่าแรงบิดแบบออนไลน์

สำหรับผลการทดลองนี้จะเลือกโมเดลที่สามารถประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ได้เพื่อใช้ในการทดลองกับอุปกรณ์วัดแรงบิดจริง ซึ่งจะใช้วิธี Time series neural network และ Linear regression model เนื่องจากมีความซับซ้อนที่ไม่สูงเกินไปและสามารถนำไปประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ได้ สัดส่วนการแบ่งข้อมูลใช้รูปแบบเดียวกับ

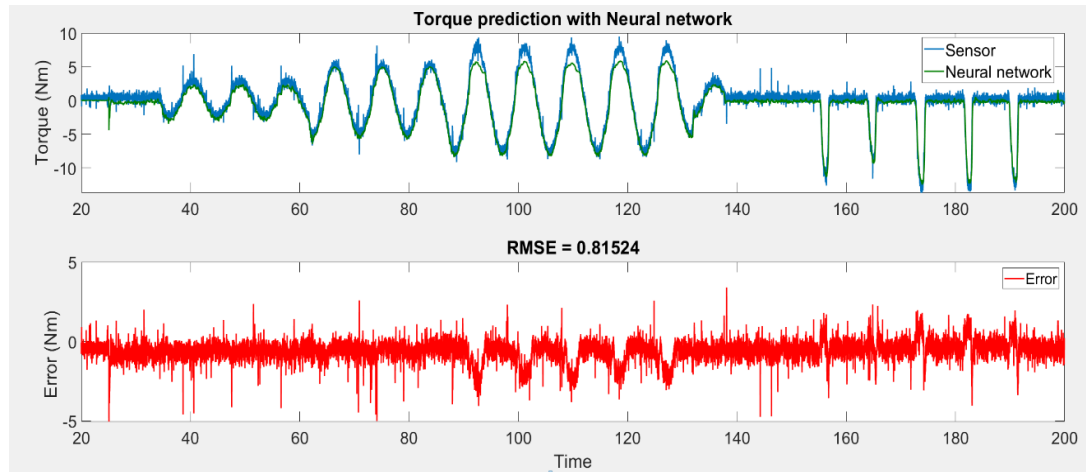
การทดลองแบบออฟไลน์ โดยใช้ Time series neural network จะแบ่งข้อมูลออกเป็น 55:25:20 และใช้ TDL เท่ากับ 5 เนื่องจากผลการทดลองที่ผ่านมาสามารถใช้เวลาในเรียนรู้โมเดลของการประมาณค่าแรงบิดน้อยแต่ยังคงให้ผลที่ดีที่สุดเมื่อประเมินด้วยค่าของ RMSE และ R value และในส่วน Linear regression model จะแบ่งข้อมูลออกเป็น 80:20 ซึ่งผลการเรียนรู้จาก Machine learning ที่จะนำไปประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ในการทดลองครั้งนี้จะได้รับการทดลองก่อนหน้านี้ทั้งหมด โดยในการทดลองได้แบ่งออกเป็น 2 กรณี คือ โหลดคงที่ (Load constant) และ โหลดแบบทันทีทันใด (Instantaneous torque) และใช้ความเร็วที่ 700 rpm , 1000 rpm และ 1200 rpm ส่วน 1500 RPM จะไม่ใช้ในการทดลองนี้เนื่องจากการทดลองก่อนหน้านี้พบว่าเมื่อมอเตอร์มีความเร็วมากจะทำให้เกิดการแกว่งของโหลดมากเกินไปทำให้ค่าแรงบิดที่วัดได้จากเซนเซอร์มีการแกว่งมากไปด้วย



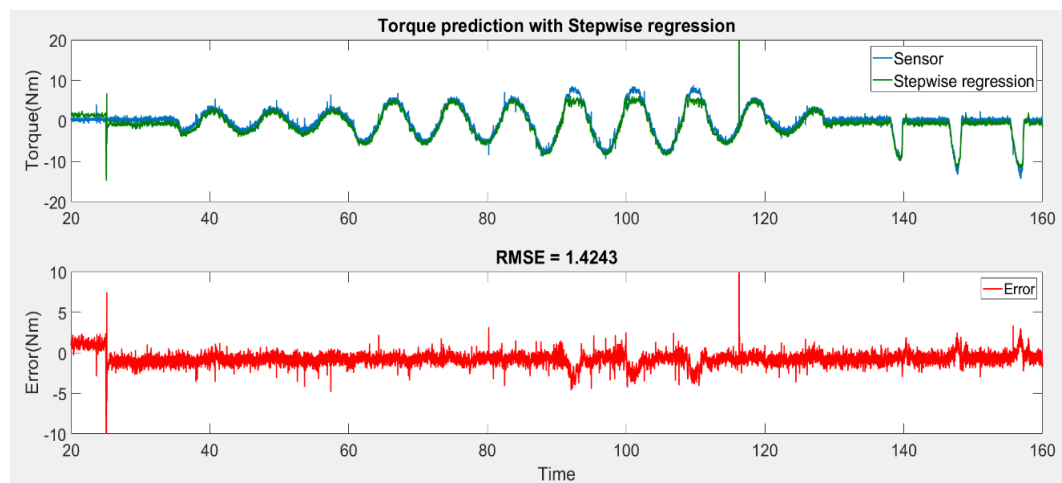
ภาพที่ 4.15 ผลของแรงบิดที่ได้จากเซนเซอร์

ภาพที่ 4.15 เป็นผลที่ได้จากเซนเซอร์ ซึ่งในกรอบสีแดงทึบเป็นการทดลองที่โหลดคงที่ ซึ่งจะเพิ่มโหลดไปที่ละรอบ ส่วนกรอบสีน้ำเงินเส้นประเป็นการทดลองที่โหลดไม่คงที่ จะมีลักษณะเป็นแบบทันทีทันใด

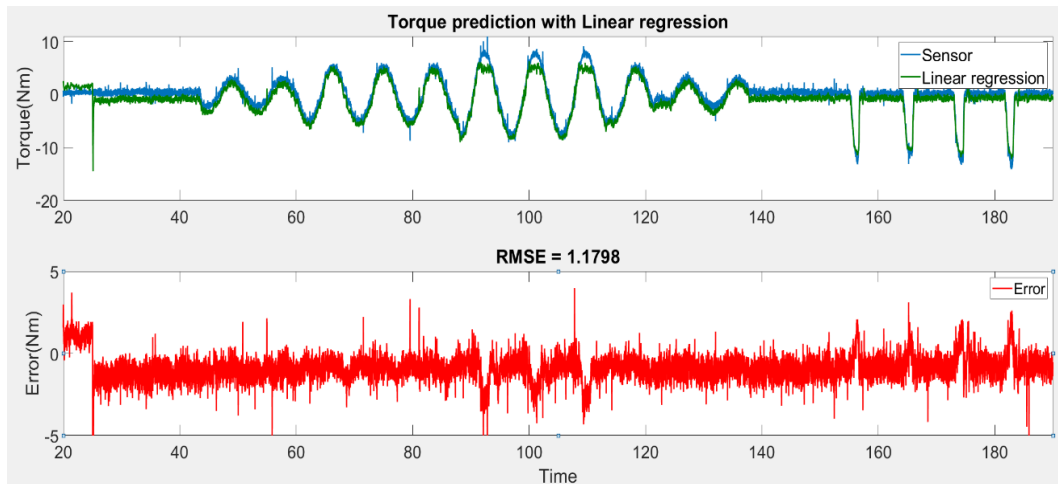
## 4.2.2.1 ความเร็วที่ 700 RPM



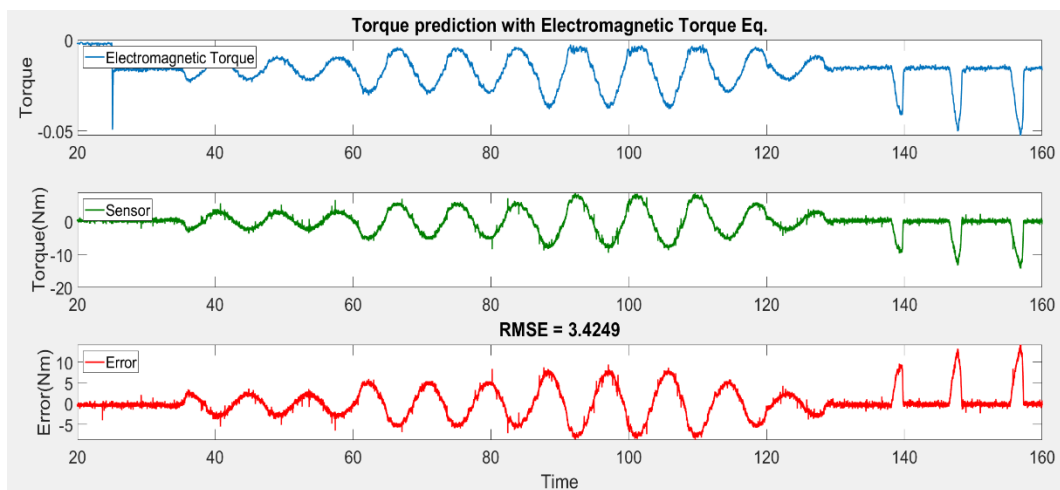
ภาพที่ 4.16 ผลการทดลองของ Neural network



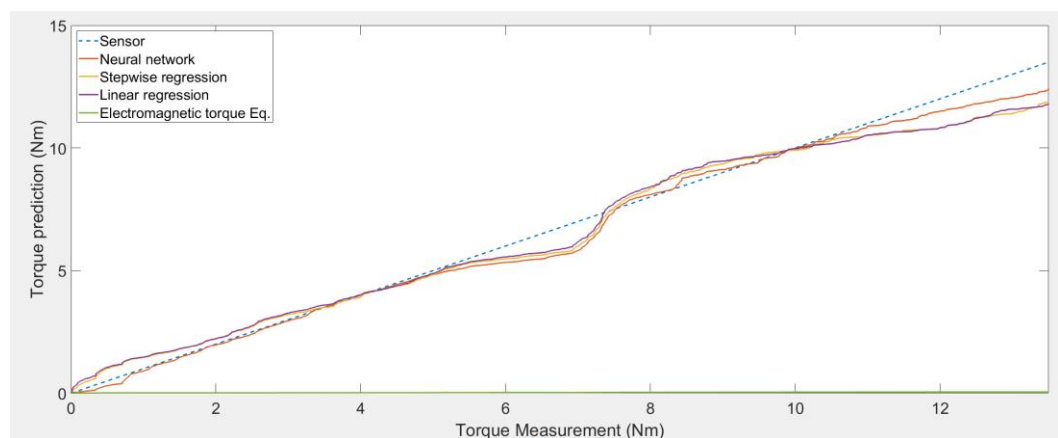
ภาพที่ 4.17 ผลการทดลองของ Stepwise regression



ภาพที่ 4.18 ผลการทดลองของ Linear regression



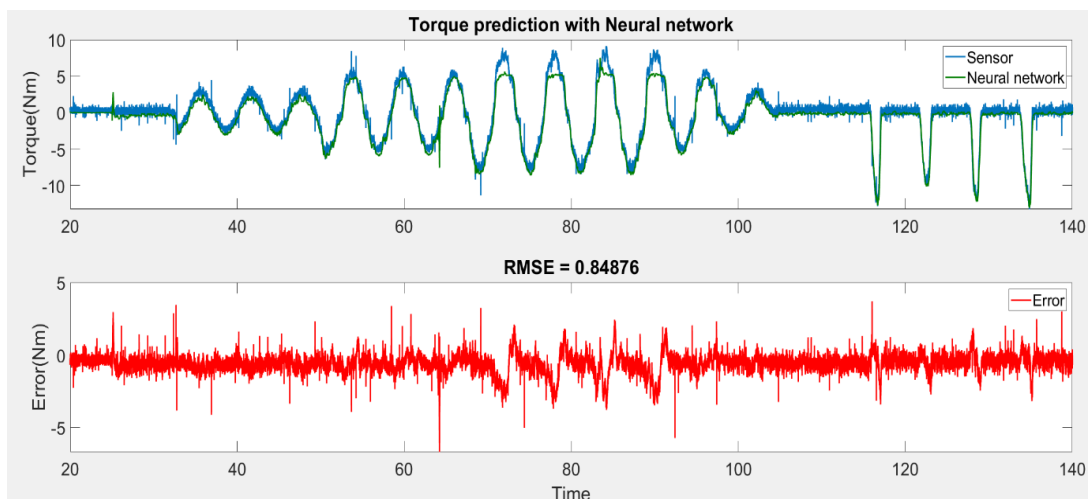
ภาพที่ 4.19 ผลการทดลองของ Electromagnetic torque eq.



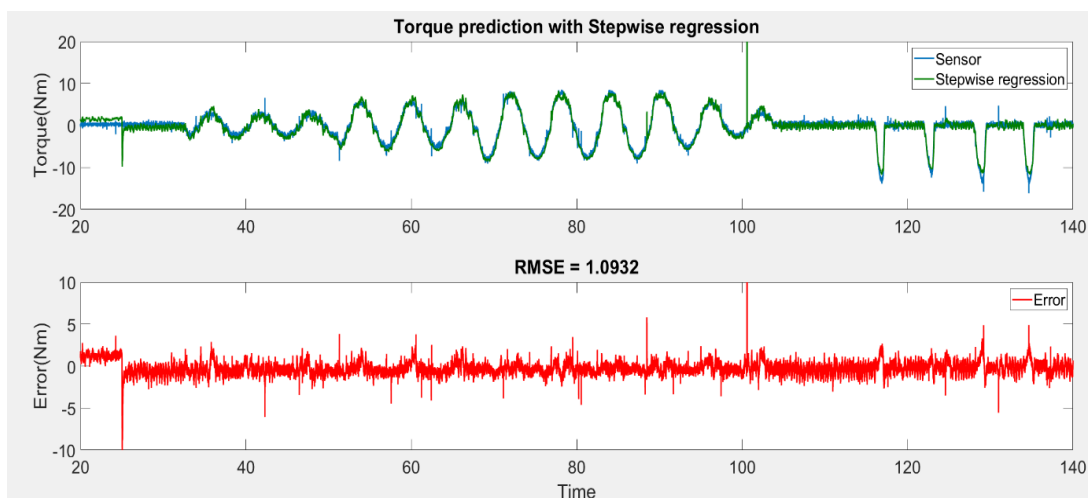
ภาพที่ 4.20 ผลการเรียงข้อมูลระหว่างผลที่วัดได้กับผลจากเซนเซอร์

จากผลการทดลองที่ความเร็วที่ 700 rpm สามารถเรียงลำดับประสิทธิภาพตามค่า RMSE ได้ตามนี้ Neural network, Linear regression, Stepwise regression, Electromagnetic torque eq. โดย Neural network , Linear regression และ Stepwise regression มีค่า RMSE เท่ากับ 0.815 และ 1.179 และ 1.424 ตามลำดับ ซึ่งมีค่าความผิดพลาดทั้งโหลดคงที่และโหลดไม่คงที่ในช่วงที่มีโหลดมาก จากภาพที่ 4.20 จะเห็นว่าทั้ง 3 โมเดลสามารถประมาณค่าแรงบิดที่ใกล้เคียงกับค่าอ้างอิงของเซนเซอร์(เส้นประสีน้ำเงิน)ได้ ส่วน Electromagnetic torque eq. ไม่สามารถประมาณค่าขนาดของแรงบิดได้แต่สามารถบอกถึงรูปแบบที่เกิดขึ้นของแรงบิดได้

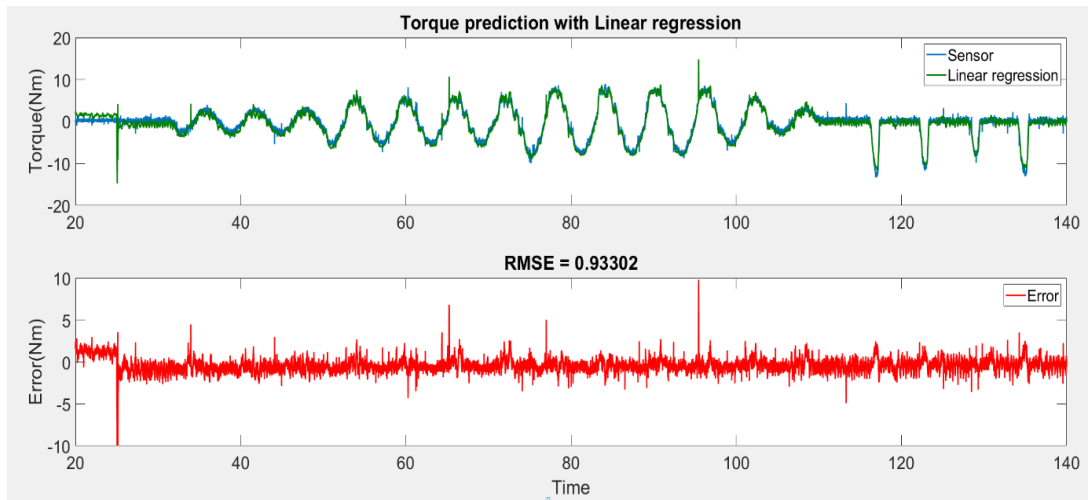
#### 4.2.2.2 ความเร็วที่ 1000 RPM



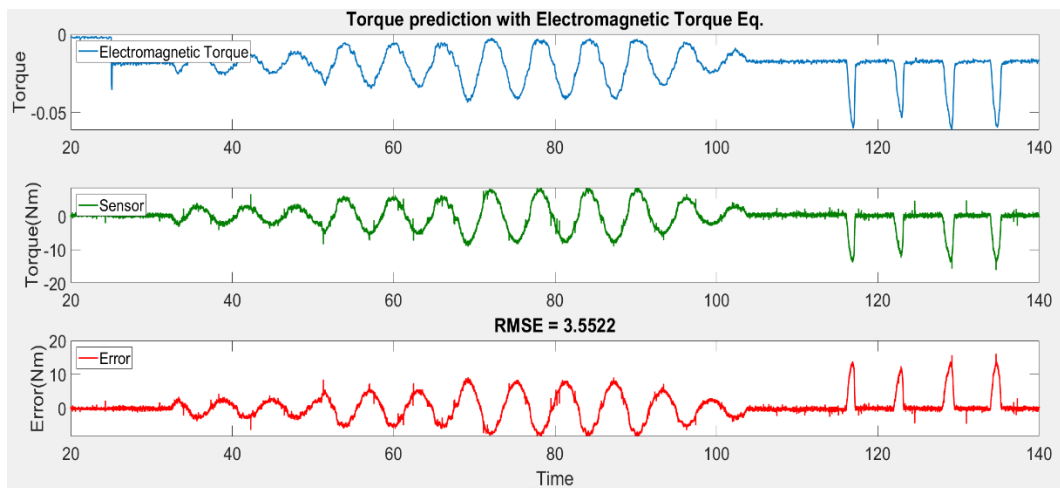
ภาพที่ 4.21 ผลการทดลองของ Neural network



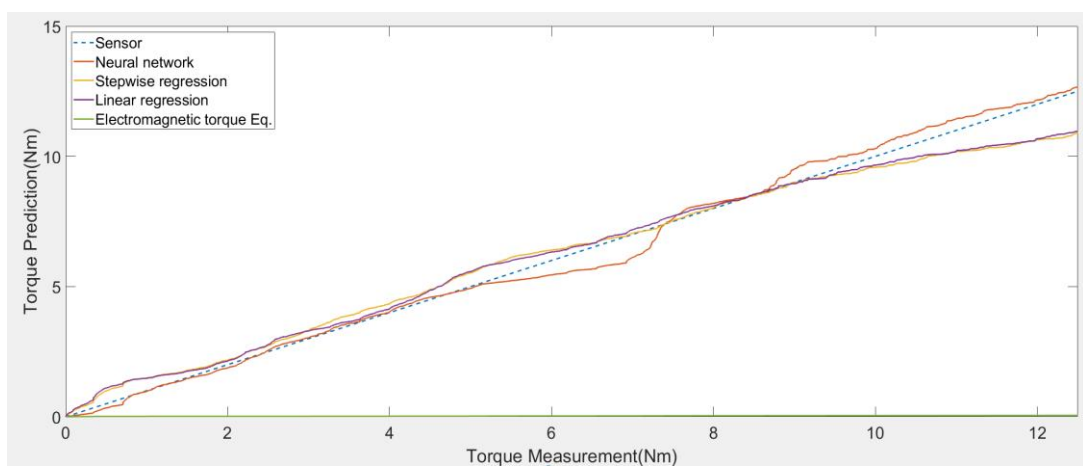
ภาพที่ 4.22 ผลการทดลองของ Stepwise regression



ภาพที่ 4.23 ผลการทดลองของ Linear regression



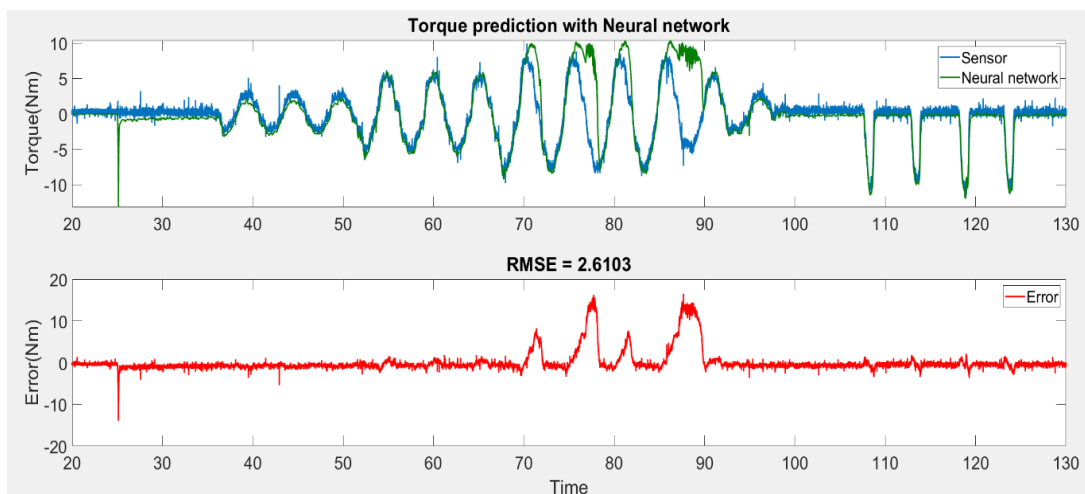
ภาพที่ 4.24 ผลการทดลองของ Electromagnetic torque eq.



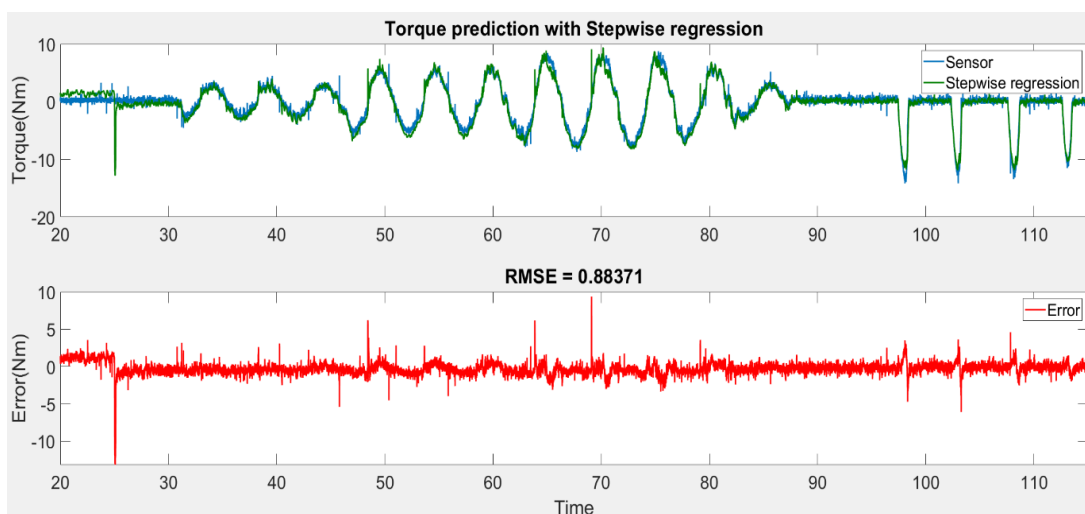
ภาพที่ 4.25 ผลการเรียงข้อมูลระหว่างผลที่วัดได้กับผลจากเซนเซอร์

จากผลการทดลองที่ความเร็วที่ 1000 rpm สามารถเรียงลำดับประสิทธิภาพตามค่า RMSE ได้ตามนี้ Neural network, Linear regression, Stepwise regression, Electromagnetic torque eq. โดย Neural network , Linear regression และ Stepwise regression มีค่า RMSE เท่ากับ 0.848 , 0.933 และ 1.093 ตามลำดับ ซึ่ง Neural network มีค่าความผิดพลาดที่ไหลดคงที่ในช่วงที่มีไหลดมาก จากภาพที่ 4.25 จะเห็นว่าทั้ง 3 โมเดลสามารถประมาณค่าแรงบิดที่ใกล้เคียงกับค่าอ้างอิงของเซนเซอร์(เส้นประสีน้ำเงิน)ได้ ส่วน Electromagnetic torque eq. ไม่สามารถประมาณค่าขนาดของแรงบิดได้แต่สามารถบอกถึงรูปแบบที่เกิดขึ้นของแรงบิดได้

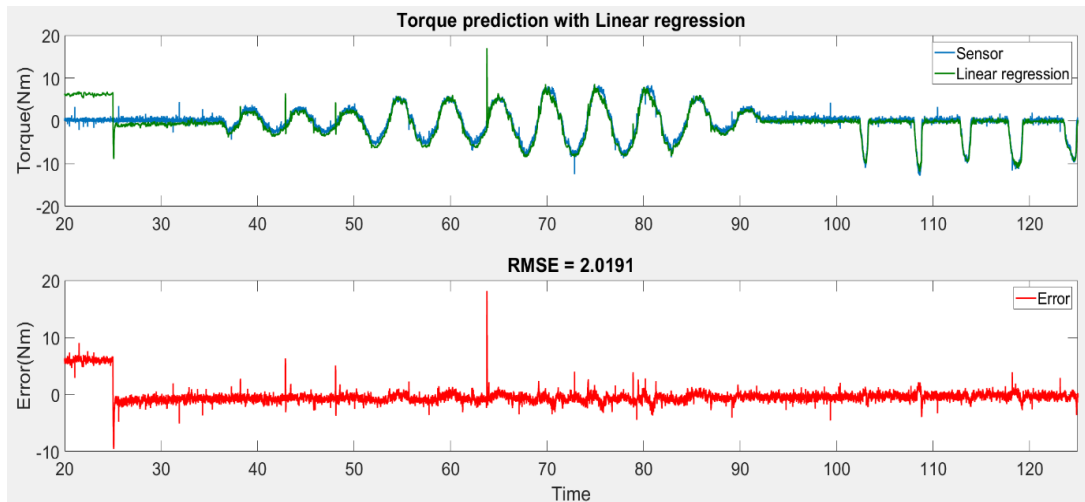
#### 4.2.2.3 ความเร็วที่ 1200 RPM



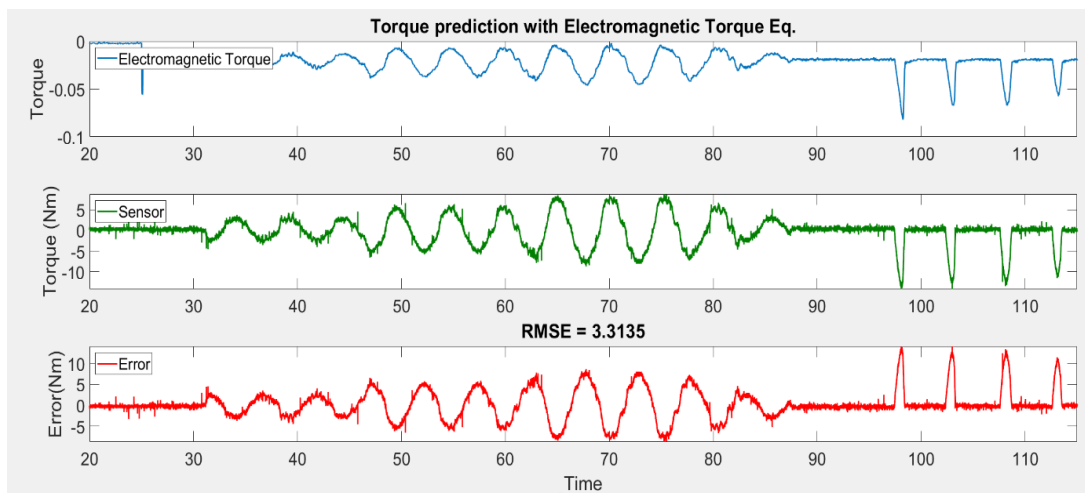
ภาพที่ 4.26 ผลการทดลองของ Neural network



ภาพที่ 4.27 ผลการทดลองของ Stepwise regression

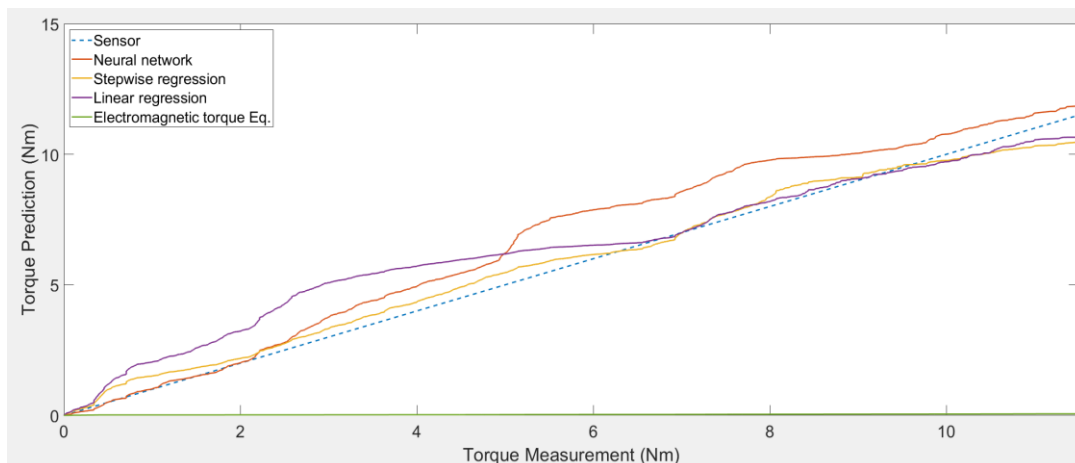


ภาพที่ 4.28 ผลการทดลองของ Linear regression



ภาพที่ 4.29 ผลการทดลองของ Electromagnetic torque eq.





ภาพที่ 4.30 ผลการเรียงข้อมูลระหว่างผลที่วัดได้กับผลจากเซนเซอร์

จากผลการทดลองที่ความเร็วที่ 1200 rpm สามารถเรียงลำดับประสิทธิภาพตามค่า RMSE ได้ตามนี้ Stepwise regression, Linear regression, Neural network, Electromagnetic torque eq. โดย Stepwise regression และ Linear regression มีค่า RMSE เท่ากับ 0.884 และ 2.02 ตามลำดับ ส่วน Neural network มีค่าความผิดพลาดมากที่สุด โหลดคงที่ในช่วงที่มีโหลดมาก ซึ่งมีค่า RMSE เท่ากับ 2.61 จากภาพที่ 4.30 จะเห็นว่าทั้ง 3 โมเดลสามารถประมาณค่าแรงบิดที่ใกล้เคียงกับค่าอ้างอิงของเซนเซอร์(เส้นประสีน้ำเงิน)ได้ ส่วน Electromagnetic torque eq. ไม่สามารถประมาณค่าขนาดของแรงบิดได้แต่สามารถบอกถึงรูปแบบที่เกิดขึ้นของแรงบิดได้

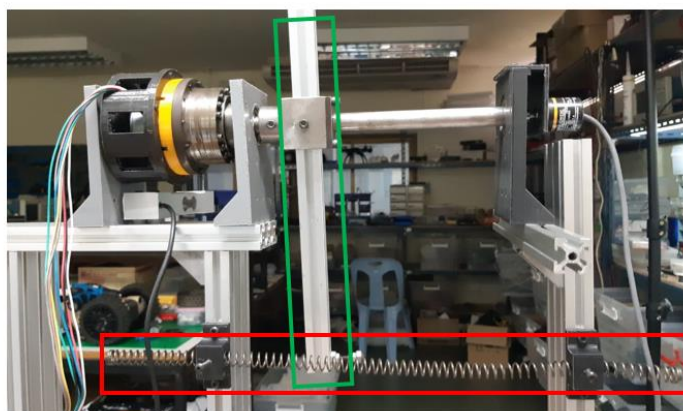
ตารางที่ 4.6 ผลการเปรียบเทียบของ RSME และค่า R ของแต่ละโมเดล

|                        | 700 RPM |         | 1000 RPM |         | 1200 RPM |         | Time prediction (ms) |
|------------------------|---------|---------|----------|---------|----------|---------|----------------------|
|                        | RMSE    | R value | RMSE     | R value | RMSE     | R value |                      |
| Neural network         | 0.815   | 0.989   | 0.849    | 0.984   | 2.610    | 0.757   | 3.5                  |
| Linear regression      | 1.18    | 0.967   | 0.933    | 0.970   | 2.019    | 0.851   | 1.35                 |
| Stepwise regression    | 1.424   | 0.932   | 1.093    | 0.957   | 0.884    | 0.968   | 1.35                 |
| Electromagnetic torque | 3.425   | 0.894   | 3.552    | 0.893   | 3.314    | 0.837   | 0.2                  |

จากตารางที่ 4.5 จะเห็นได้ว่า Neural network สามารถประมาณค่าแรงบิดได้ดีที่สุดที่ 700 และ 1000 RPM เมื่อประเมินจากค่า RMSE และค่า R value ตามด้วย Linear regression , Stepwise regression และสมการของ Electromagnetic torque ส่วนความเร็วที่ 1200 RPM มีเพียง Stepwise regression ที่ได้คำตอบที่มีค่า RMSE ต่ำกว่า 1 แต่เวลาที่ใช้ในประมาณค่าแรงบิด Neural network จะใช้เวลามากที่สุดอยู่ที่ 3.5 ms ส่วน Linear regression และ Stepwise regression จะใช้เวลาเท่ากันอยู่ที่ 1.35 ms และสมการของ Electromagnetic torque ใช้เวลาในการประมาณค่าน้อยที่สุดอยู่ที่ 0.2 ms แต่สมการนี้ไม่สามารถบอกถึงค่าของแรงบิดได้ แต่สามารถบอกได้เพียงรูปแบบของแรงบิดเท่านั้น เนื่องจากค่า RMSE จะมีค่าความผิดพลาดที่สูงมาก แต่กลับมีค่าของ R value ที่ใกล้เคียง 1

#### 4.3 ผลการทดลองของการประมาณค่าแรงบิดด้วยการจำลองในรูปแบบของแขนหุ่นยนต์

ในการทดลองจะแบ่งออกเป็น 4 การทดลองตามค่าแรงบิดต่าง ๆ ตั้งแต่ช่วงแรงบิดต่ำและเพิ่มขึ้นเรื่อย ๆ ในแต่ละช่วงความเร็วประกอบไปด้วย 3 , 5 , 8 และ 10 Nm โดยในการทดลองจะใช้โมเดลที่ได้จากผลการทดลองที่ 4.2 คือผลการทดลองการประมาณค่าแรงบิดแบบออนไลน์หรือที่เป็นการประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ซึ่งจะมีโมเดลของ Time series neural network ที่ TDL เท่ากับ 5, Linear regression และ Stepwise regression ซึ่งอุปกรณ์ที่ใช้สำหรับการทดลองยังคงใช้อุปกรณ์ชุดเดียวกัน แต่ในการควบคุมแรงบิดเพื่อใช้ในการจำลองวัตถุที่เกิดการกระทบหรือกระแทกจะใช้สปริงยาว (กรอบสี่แดง) ที่สามารถยืดหยุ่นได้มายึดไว้ตามภาพที่ 5.1 เมื่อแขนหุ่นยนต์จำลอง (กรอบสีเขียว) มีการกระทบเข้ากับสปริงที่ขวางอยู่จะทำให้เกิดแรงบิดขึ้นและแขนหุ่นยนต์จะต้องหยุดการทำงานทันทีเมื่อค่าแรงบิดเกินค่าที่กำหนดไว้ โดยในการทดลองจะกำหนดแรงบิดในการควบคุมการหยุดของมอเตอร์และทำการเปรียบเทียบแรงบิดจากการประมาณค่าและแรงบิดที่วัดได้เพื่อดูค่าความผิดพลาดในแต่ละช่วงแรงบิดและความเร็วของมอเตอร์ และในการทดลองนี้จะไม่ใช่สมการ Electromagnetic torque ในการทดลอง เนื่องจากสมการนี้ไม่สามารถให้คำตอบของแรงบิดที่ใกล้เคียงค่าจริงได้ แต่สามารถบอกถึงพฤติกรรมของแรงบิดที่เกิดขึ้นได้เท่านั้น



ภาพที่ 5.1 อุปกรณ์การทดลองจำลองการทำงานของแขนหุ่นยนต์

โดยกำหนดให้ Latency ในหน่วย ms คือการวัดระยะเวลาห่างกันระหว่างเซนเซอร์แรงบิดและการประมาณค่าแรงบิดเมื่อเกินค่าแรงบิดที่กำหนด , Method คือ ค่าที่วัดได้เมื่อแรงบิดเกินกว่าค่าที่กำหนดเพื่อต้องการหยุดการทำงานของหุ่นยนต์ในแต่ละโมเดล และ Sensor คือค่าที่วัดจากเซนเซอร์ ณ เวลาที่การประมาณค่าแรงบิดแรงบิดเกินค่าที่กำหนด

## การทดลองที่ 4.3.1 แรงบิดเท่ากับ 3 Nm

ตารางที่ 4.7 แรงบิดเท่ากับ 3 Nm

|          | Model               | Latency<br>(ms) | Method<br>(Nm) | Sensor<br>(Nm) | Error <br>(Nm) |
|----------|---------------------|-----------------|----------------|----------------|----------------|
| 700 rpm  | Linear regression   | 10              | 3.37           | 2.95           | 0.42           |
|          | Stepwise regression | 10              | 3.11           | 2.90           | 0.21           |
|          | Neural network      | 10              | 3.07           | 3.32           | 0.25           |
| 1000 rpm | Linear regression   | 15              | 3.17           | 3.20           | 0.03           |
|          | Stepwise regression | 20              | 3.40           | 3.21           | 0.19           |
|          | Neural network      | 10              | 3.10           | 3.33           | 0.23           |
| 1200 rpm | Linear regression   | 20              | 3.32           | 3.28           | 0.04           |
|          | Stepwise regression | 15              | 3.17           | 2.45           | 0.72           |
|          | Neural network      | 10              | 3.23           | 3.17           | 0.06           |

จากการทดลองที่ 4.3.1 พบว่าเมื่อมีความเร็วที่มากขึ้นจะใช้เวลาในประมาณค่าแรงบิดเพิ่มขึ้นสำหรับ Linear regression และ Stepwise regression ส่วน Neural network ใช้เวลาในการประมาณค่าที่เท่ากันทั้ง 3 ความเร็ว ซึ่งทั้ง 3 โมเดลมีค่าความผิดพลาดของแรงบิดในการประมาณค่าเทียบกับแรงบิดที่วัดได้ไม่เกิน 1 Nm

## การทดลองที่ 4.3.2 แรงบิดเท่ากับ 5 Nm

ตารางที่ 4.8 แรงบิดเท่ากับ 5 Nm

|          | Model               | Latency (ms) | Method (Nm) | Sensor (Nm) | Error  (Nm) |
|----------|---------------------|--------------|-------------|-------------|-------------|
| 700 rpm  | Linear regression   | 15           | 5.18        | 5.34        | 0.16        |
|          | Stepwise regression | 5            | 5.23        | 5.05        | 0.18        |
|          | Neural network      | 20           | 5.03        | 5.45        | 0.42        |
| 1000 rpm | Linear regression   | 10           | 5.13        | 5.76        | 0.63        |
|          | Stepwise regression | 15           | 5.17        | 5.72        | 0.55        |
|          | Neural network      | 15           | 5.12        | 5.60        | 0.48        |
| 1200 rpm | Linear regression   | 15           | 5.04        | 5.32        | 0.28        |
|          | Stepwise regression | 15           | 5.03        | 6.24        | 1.21        |
|          | Neural network      | 15           | 5.03        | 6.48        | 1.45        |

จากการทดลองที่ 4.3.2 พบว่าสามารถประมาณค่าแรงบิดได้และใช้เวลาในประมาณค่าแรงบิดอยู่ในช่วง 12.5 ms โดยเฉลี่ย แต่สำหรับช่วงความเร็วสูงของ Stepwise regression และ Neural network มีค่าความผิดพลาดของการประมาณค่าแรงบิดเกิน 1 Nm

## การทดลองที่ 4.3.3 แรงบิดเท่ากับ 8 Nm

ตารางที่ 4.9 แรงบิดเท่ากับ 8 Nm

|          | Model               | Latency<br>(ms) | Method<br>(Nm) | Sensor<br>(Nm) | Error <br>(Nm) |
|----------|---------------------|-----------------|----------------|----------------|----------------|
| 700 rpm  | Linear regression   | 20              | 8.4            | 9.50           | 1.1            |
|          | Stepwise regression | 40              | 8.05           | 9.00           | 0.95           |
|          | Neural network      | 40              | 8.09           | 9.88           | 1.79           |
| 1000 rpm | Linear regression   | 10              | 8.08           | 8.97           | 0.89           |
|          | Stepwise regression | 15              | 8.07           | 9.37           | 1.3            |
|          | Neural network      | 35              | 8.19           | 10.5           | 2.31           |
| 1200 rpm | Linear regression   | 10              | 8.08           | 8.39           | 0.31           |
|          | Stepwise regression | 20              | 8.18           | 9.45           | 1.27           |
|          | Neural network      | 45              | 8.55           | 10.2           | 1.65           |

จากการทดลองที่ 4.3.3 พบว่ายิ่งค่าแรงบิดที่สูงขึ้นจะเริ่มค่าความผิดพลาดในการประมาณค่าแรงบิดสูงขึ้นด้วยเมื่อเปรียบเทียบกับ การทดลองก่อนหน้านี้ ซึ่งจะเห็นได้ว่าค่าความผิดพลาดมีค่าเกิน 1 Nm เป็นส่วนใหญ่ในทั้ง 3 โมเดล และเวลาที่ใช้ในการประมาณค่าที่สูงขึ้นด้วย โดยเฉพาะ Neural network ที่ใช้เวลาเฉลี่ยเท่ากับ 40 ms ในการประมาณค่า ส่วน Stepwise regression ที่ 700 rpm ก็ใช้เวลาในการประมาณค่าสูงเช่นกันอยู่ที่ 40 ms แต่ค่าความผิดพลาดในการประมาณค่าไม่เกิน 1 Nm

## การทดลองที่ 4.3.4 แรงบิดเท่ากับ 10 Nm

ตารางที่ 4.10 แรงบิดเท่ากับ 10 Nm

|          | Model               | Latency<br>(ms) | Method<br>(Nm) | Sensor<br>(Nm) | Error <br>(Nm) |
|----------|---------------------|-----------------|----------------|----------------|----------------|
| 700 rpm  | Linear regression   | 25              | 10.22          | 10.98          | 0.76           |
|          | Stepwise regression | 25              | 10.31          | 10.92          | 0.61           |
|          | Neural network      | 65              | 10.01          | 12.05          | 2.04           |
| 1000 rpm | Linear regression   | 45              | 10.37          | 12.62          | 2.25           |
|          | Stepwise regression | 55              | 10.32          | 13.35          | 3.03           |
|          | Neural network      | 55              | 10.27          | 13.88          | 3.61           |
| 1200 rpm | Linear regression   | 35              | 10.11          | 12.82          | 2.71           |
|          | Stepwise regression | 40              | 10.47          | 13.00          | 2.53           |
|          | Neural network      | 40              | 10.06          | 15.36          | 5.3            |

จากการทดลองที่ 4.3.4 เป็นการทดลองที่เป็นข้อมูลที่ไม่ได้อยู่ในช่วงที่ทำ Machine learning พบว่ามีค่าความผิดพลาดที่สูงมาก จากการทดลองจะเห็นว่า Linear regression สามารถประมาณค่าได้ดีที่สุดทั้งเรื่องเวลาในการประมาณค่าและค่าความผิดพลาดโดยเฉลี่ยอยู่ที่ 35 ms และ 1.91 Nm และ Neural network ใช้ในเวลาในการประมาณค่าแรงบิดนานที่สุดเฉลี่ยอยู่ที่ 53 ms

## บทที่ 5

### บทสรุป

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาการประมาณค่าแรงบิดจากมอเตอร์ไร้แปรงถ่านแทนการใช้เซนเซอร์แรงบิดโดยใช้วิธี Machine learning ซึ่งในงานวิจัยได้แบ่งออกเป็น 4 ส่วน ประกอบด้วย การออกแบบอุปกรณ์เพื่อใช้สำหรับการทดลองโดยอ้างอิงและจำลองการทำงานของแขนหุ่นยนต์, ทดสอบประสิทธิภาพการประมาณค่าแรงบิดจากข้อมูลการทำงานของมอเตอร์โดยไม่มี การประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์หรือการประมาณค่าแบบออฟไลน์, ทดสอบ ประสิทธิภาพการประมาณค่าแรงบิดโดยการประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์หรือการ ประมาณค่าแบบออนไลน์ และทดสอบการประมาณค่าแรงบิดจากการจำลองการทำงานของแขน หุ่นยนต์ โดยสามารถสรุปผลการทดลองได้ดังนี้

#### 5.1 ผลการออกแบบอุปกรณ์สำหรับการทดลอง

ในหัวข้อนี้มีวัตถุประสงค์การออกแบบอุปกรณ์สำหรับใช้ในการทดลองและ ประเมินผลการประมาณค่าแรงบิดจากมอเตอร์ด้วยระเบียบที่นำเสนอโดยมีแนวทางในการออกแบบ อ้างอิงการทำงานของแขนหุ่นยนต์ตามวัตถุประสงค์ของงานวิจัยนี้ที่มีเป้าหมายในการประยุกต์ใช้กับ แขนหุ่นยนต์ โดยในการออกแบบจะใช้มอเตอร์ติดตั้งเข้ากับเกียร์ที่มีอัตราทดเพื่อเพิ่มแรงบิดให้กับ แขนหุ่นยนต์ และใช้ Strain gauge หรือ Load cell ในการวัดแรงบิดแทนการใช้เซนเซอร์วัดแรงบิด โดยเฉพาะ เนื่องจากเซนเซอร์ชนิดนี้มีราคาที่สูงและมีข้อจำกัดทางด้านพิสัยการวัดแรงบิด วัสดุที่ใช้ใน การจับยึดระหว่างมอเตอร์กับเกียร์จะใช้พลาสติกชนิด ABS ที่มีความแข็งแรงสูงเพื่อลดการเกิดการสับแรง ในตัวพลาสติก

ในการวัดแรงบิดที่ส่วนสเตเตอร์ของมอเตอร์จะเป็นผลกระทบที่เกิดจากการหมุน ของโรเตอร์มอเตอร์ ซึ่งแรงบิดที่เกิดขึ้นจะขึ้นอยู่กับโหลดที่เพิ่มเข้าไปในระบบเท่านั้น เนื่องจากทุกจุด ที่มีการหมุนในระบบจะใช้ตัลบลูกปืนเพื่อลดแรงเสียดทานที่เกิดขึ้น และในระบบจะมีเซนเซอร์วัด ความเร็วของเพล่าที่เข้ากับเกียร์โดยตรง ซึ่งผลที่ได้จากการวัดแรงบิดด้วยอุปกรณ์การทดลองนี้จะทำ ให้ได้ข้อมูลแรงบิดที่มีลักษณะคล้ายกับสัญญาณ Sine wave โดยยอดคลื่นจะขึ้นอยู่กับโหลดที่เพิ่มเข้า ไป

#### 5.2 ผลการทดลองการประมาณค่าแรงบิดแบบออฟไลน์

เป็นการทดลองที่ไม่มีการประมวลผลการประมาณค่าแรงบิดบนบอร์ด ไมโครคอนโทรลเลอร์หรือประมาณค่าแรงบิดจากการเก็บข้อมูลของมอเตอร์และทำการประมวลผล



บนคอมพิวเตอร์ส่วนบุคคล ในการทดลองจะใช้โมเดล Machine learning ในโปรแกรม MATLAB ทั้งหมด 3 โมเดล คือ Time series neural network , Linear regression model และ Support vector machines โดยในการทดลองจะมีโหลดเข้าไปในระบบ 4 ค่า คือ 1, 2, 3 และ 4 kg และมีความเร็วคงที่ที่ 700, 1000, 1200 และ 1500 RPM ที่โรเตอร์ของมอเตอร์

### 5.2.1 การทดลองโดยการใช้โมเดลของ Neural network

จะเห็นได้ว่ายิ่งมีการเพิ่มของ TDL มากขึ้น จะให้ผลของ RMSE ดีขึ้นไปด้วย แต่ก็เพิ่มระยะเวลาในการประมาณค่าแรงบิดด้วยเช่นกัน ในขณะที่ค่าของ R value กลับใกล้เคียงกันโดยไม่ขึ้นกับ TDL ซึ่งที่ TDL มีค่าเท่ากับ 1 จะใช้เวลาในการเรียนรู้โมเดลที่ 22 วินาที แต่ที่ TDL เท่ากับ 20 จะใช้เวลาในการเรียนรู้โมเดลที่ 891 วินาที ซึ่งความแตกต่างของ RMSE อยู่ที่ 0.038

### 5.2.2 การทดลองโดยการใช้โมเดลของ Regression

การทดลองนี้ใช้โมเดลของ Linear regression และ Stepwise regression เพื่อเปรียบเทียบประสิทธิภาพในการประมาณค่าแรงบิด จะเห็นได้ว่า Stepwise regression สามารถให้คำตอบทั้ง RMSE และ R value ได้ดีกว่า Linear regression แต่จะใช้เวลาในการประมาณค่าแรงบิดเพิ่มขึ้น เมื่อเทียบทั้งสองโมเดลนี้จะใช้เวลาต่างกันถึง 28.301 วินาที แต่ Linear regression ให้ผลของ RMSE เพียง 0.208

### 5.2.3 การทดลองโดยการใช้โมเดลของ Support vector machines

การทดลองจะเลือกใช้โมเดล Linear SVM , Quadratic SVM และ Medium gaussian SVM ในการเปรียบเทียบประสิทธิภาพในการประมาณค่าแรงบิด จะเห็นได้ว่า Medium gaussian SVM สามารถให้คำตอบที่ดีที่สุดและระยะเวลาในการประมาณค่าสั้นที่สุดในขณะที่ Linear SVM ถึงแม้ว่าจะให้คำตอบได้เร็วกว่า Quadratic SVM แต่เมื่อเทียบกับระยะเวลาที่ใช้ในการเรียนรู้โมเดลกลับดีกว่า ซึ่งสั้นกว่า 1.34 ชั่วโมง

การทดลองทั้ง 3 โมเดล คือ Time series neural network , Linear regression และ Support vector machines จะเห็นได้ว่า Neural network สามารถให้ค่า RMSE และ R value ได้ดีกว่าทุก ๆ โมเดล ซึ่งมีผลค่าความผิดพลาดที่น้อย และมีค่า RMSE ไม่เกิน 1 ซึ่ง Medium gaussian SVM ก็ให้ผล RMSE ที่ต่ำกว่า 1 ด้วย แต่เมื่อเทียบด้วยเวลาในการเรียนรู้โมเดลประมาณค่าแรงบิดจะแตกต่างกันและรองลงมาเป็นโมเดลของ Regression และ Support vector machines

## 5.3 ผลการทดลองการประมาณค่าแรงบิดแบบออนไลน์

การทดลองการประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์โดยจะแบ่งการทดลองออกตามช่วงความเร็วประกอบไปด้วย 700, 1000 และ 1200 RPM ที่โรเตอร์ของมอเตอร์ โดยจะ

เลือกใช้โมเดลของ Linear regression, Stepwise regression และ Time series neural network เพื่อใช้ในการเปรียบเทียบประสิทธิภาพในการประมาณค่าแรงบิดด้วยข้อมูลจากมอเตอร์แบบทันเวลา รวมถึงสมการของ Electromagnetic torque ด้วย โดยการทดลองพบว่า Neural network สามารถให้คำตอบที่มี RMSE และ R value ได้ดีที่สุดที่ช่วงความเร็ว 700 และ 1000 RPM รองลงมาเป็น Linear regression และ Stepwise regression แต่ในช่วงความเร็วที่ 1200 RPM โมเดลของ Stepwise regression กลับให้คำตอบได้ดีกว่าโมเดลอื่น ๆ โดยค่า RMSE น้อยกว่า 1 ต่างจากโมเดลอื่นที่มากกว่า 2 ของ RMSE และค่า R value ก็ใกล้เคียง 1 มากกว่าโมเดลอื่น ส่วนสมการของ Electromagnetic torque ไม่สามารถให้ผลคำตอบของแรงบิดได้เลย แต่สามารถบอกได้เพียงรูปแบบของแรงบิดที่เกิดขึ้นเท่านั้นในทุกช่วงความเร็วและโหลดในระบบ ซึ่งในกรณีการประยุกต์ที่ให้ความสำคัญเรื่องการประมวลผลที่ทันเวลาหรือระยะเวลาในการประมวลผลค่าแรงบิดสมการ Electromagnetic torque จะให้คำตอบที่สั้นที่สุด รองลงมาเป็นโมเดลของ Regression และ Neural network

#### 5.4 ผลการทดลองการประมาณค่าแรงบิดจากการจำลองการทำงานของแขนหุ่นยนต์

ในการทดลองวัดแรงบิดจากการจำลองการทำงานในแขนหุ่นยนต์จะเป็นการจำลองการเคลื่อนที่ของแขนหุ่นยนต์จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง โดยในระหว่างการเคลื่อนที่นั้น เมื่อมีสิ่งกีดขวางระหว่างการเคลื่อนที่ของแขนหุ่นยนต์ทำให้เกิดค่าของแรงบิดเกินกว่าค่าที่กำหนดไว้ แขนหุ่นยนต์จะต้องหยุดการทำงานทันที จากนั้นจะวัดประสิทธิภาพของการประมาณค่าแรงบิดเทียบกับเซนเซอร์แรงบิดและวัดระยะเวลาระหว่างการประมาณค่าเทียบกับเซนเซอร์แรงบิด โดยในการทดลองจะแบ่งการทดลองออกเป็น 4 การทดลองประกอบไปด้วย 3, 5, 8 และ 10 Nm เพื่อใช้เป็นตัวกำหนดในการหยุดการทำงานของแขนหุ่นยนต์และในแต่ละการทดลองจะใช้โมเดลที่สามารถประมวลผลบนบอร์ดไมโครคอนโทรลเลอร์ได้ โดยในการทดลองจะเลือกใช้ความเร็วที่ 700, 1000 และ 1200 RPM จากการทดลองพบว่า

##### 5.4.1 แรงบิดที่กำหนดเท่ากับ 3 Nm

ในการทดลองพบว่ามีค่าความผิดพลาดน้อยต่ำกว่า 1 และมีความล่าช้าในการตอบสนองต่อแรงบิดในประมาณค่าแรงบิดอยู่ในช่วง 10 ถึง 20 ms โดยเฉลี่ยอยู่ที่ 13.3 ms และที่ 700 RPM โมเดลของ Stepwise regression สามารถคำตอบได้ดีที่สุดเมื่อเทียบกับค่าความผิดพลาดในการประมาณค่าแรงบิด ส่วนที่ 1000 RPM เป็น Linear regression ที่ให้คำตอบโดยรวมได้ดี ถึงแม้จะมีความล่าช้าในการตอบสนองต่อแรงบิดมากกว่า Neural network แต่มากกว่าเพียง 5

ms เท่านั้น แต่ค่าความผิดพลาดต่างกันถึง 0.2 Nm และที่ความเร็ว 1200 RPM โมเดลของ Neural network ที่ให้คำตอบโดยรวมได้ดีกว่าโมเดลอื่น ๆ

#### 5.4.2 แรงบิดที่กำหนดเท่ากับ 5 Nm

จะเห็นได้ว่ายิ่งมีความเร็วในการหมุนมากขึ้นจะให้คำตอบที่ผิดพลาดมากขึ้นไปด้วย แต่มีความล่าช้าในการตอบสนองต่อแรงบิดอยู่ในช่วง 15 ถึง 20 ms มีเพียง Stepwise regression ที่ความเร็ว 700 RPM ที่มีความล่าช้าในการตอบสนองต่อแรงบิดที่ 5 ms ในการประมาณค่าแรงบิด และยังให้ค่าความผิดพลาดน้อยเมื่อกับโมเดลอื่น ๆ

#### 5.4.3 แรงบิดที่กำหนดเท่ากับ 8 Nm

จะเห็นได้ว่าภาพรวมของ Linear regression สามารถทำได้ดีที่สุดในทุกความเร็วทั้งเรื่องของเวลาในการประมาณค่าแรงบิดและค่าความผิดพลาด ส่วน Neural network จะมีค่าความผิดพลาดที่มากกว่า 1 Nm และใช้เวลาในการประมาณค่าแรงบิดโดยเฉลี่ยที่ 40 ms ถือว่ามากที่สุดในทุก ๆ โมเดล

#### 5.4.4 แรงบิดที่กำหนดเท่ากับ 10 Nm

เป็นช่วงแรงบิดที่ไม่ได้ทำการเก็บข้อมูลเพื่อกระบวนการเรียนรู้ของโมเดล จะเห็นได้ว่ามีความล่าช้าในการตอบสนองต่อแรงบิดที่เพิ่มขึ้นจากการทดลองอื่น ๆ และมีค่าความผิดพลาดที่มากกว่า 2 Nm เป็นส่วนใหญ่

การทดลองทั้งสี่การทดลองพบว่าในแรงบิดที่กำหนด 3 และ 5 Nm ในทุก ๆ โมเดลสามารถให้คำตอบในการประมาณค่าแรงบิดได้ค่าที่ใกล้เคียงเมื่อเปรียบเทียบกับเซนเซอร์แรงบิด ซึ่งโดยรวมแล้วมีค่าความผิดพลาดไม่เกิน 1 Nm และมีความล่าช้าในการตอบสนองต่อแรงบิดในการประมาณค่าไม่เกิน 20 ms ส่วนแรงบิดที่กำหนดเท่ากับ 8 และ 10 Nm จะมีความล่าช้าในการตอบสนองต่อแรงบิดในการประมาณค่าที่มากขึ้นและยังมีความเร็วที่มากขึ้นจะให้ค่าความผิดพลาดที่มากขึ้นไปด้วย

### 5.5 ข้อเสนอแนะ

งานวิจัยนี้มุ่งเน้นในศึกษาและทดลองการประมาณค่าแรงบิดจากมอเตอร์เพื่อใช้ในการทำงานของแขนหุ่นยนต์เพื่อมาทดแทนการใช้เซนเซอร์แรงบิดในแขนหุ่นยนต์ Motor test bench ได้ถูกออกแบบโดยอ้างอิงการทำงานของแขนหุ่นยนต์ แต่ในการออกแบบในการเพิ่มโหลดเข้าไปในระบบจะเกิดแกว่งขึ้นเมื่อเพิ่มโหลดในระหว่างที่มอเตอร์กำลังทำงานทำให้ในการวัดแรงบิดจาก

เซนเซอร์แรงบิดจะมีการแกว่งไปด้วยและสัญญาณที่ออกมาจากมอเตอร์จะมีความเสถียรน้อยลงไปด้วย ซึ่งจะส่งผลประสิทธิภาพของกระบวนการประมาณค่าแรงบิด ข้อเสนอแนะควรออกแบบลดการแกว่งของโหลดที่จะเพิ่มเข้าไปเพื่อเพิ่มประสิทธิภาพในการวัดแรงบิดมากขึ้น

## 5.6 แนวทางการพัฒนาต่อไป

5.6.1 เพิ่มช่วงการประมาณค่าแรงบิดให้มากขึ้น เพื่อให้เหมาะสมกับการทำงานของเซนเซอร์มากขึ้น

5.6.2 ศึกษาและเปรียบเทียบโมเดลอื่น ๆ เพื่อเพิ่มประสิทธิภาพในการประมาณค่าแรงบิดของเซนเซอร์

5.6.3 แก้ไข Motor test bench ในการทดลองในการวัดแรงบิดเพื่อลดการแกว่งของโหลดในระหว่างการหมุนมอเตอร์

### บรรณานุกรม

- [1] J. C. Gamazo-Real, E. Vázquez-Sánchez, and J. Gómez-Gil, “Position and speed control of brushless dc motors using sensorless techniques and application trends,” *Sensors*, vol. 10, no. 7, pp. 6901–6947, 2010, doi: 10.3390/s100706901.
- [2] D. A. Morales, K. Findlater, and V. Chandran, “A motor controller using field oriented control and Hall effect rotor position sensors: Simulation and implementation,” *EDERC 2014 - Proc. 6th Eur. Embed. Des. Educ. Res. Conf.*, pp. 235–239, 2014, doi: 10.1109/EDERC.2014.6924395.
- [3] Y. Liu, Z. Q. Zhu, and D. Howe, “Instantaneous torque estimation in sensorless direct-torque-controlled brushless DC motors,” *IEEE Trans. Ind. Appl.*, vol. 42, no. 5, pp. 1275–1283, 2006, doi: 10.1109/TIA.2006.880854.
- [4] M. Okte and Sathans, “Sliding-mode observer for estimating position and speed and minimizing ripples in rotor parameters of PMSM,” *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 506–511, 2018, doi: 10.1109/ICISC.2018.8399124.
- [5] M. Babaei, R. Asgharei, and A. Ahmarinejad, “Electromagnetic Torque and Speed Estimators for Permanent Magnet Synchronous Motor Drive Systems,” *Energy Procedia*, vol. 100, pp. 291–296, 2016, doi: 10.1016/j.egypro.2016.10.180.
- [6] Z. Boussaada, O. Curea, A. Remaci, H. Camblong, and N. Mrabet Bellaaj, “A Nonlinear Autoregressive Exogenous (NARX) Neural Network Model for the Prediction of the Daily Direct Solar Radiation,” *Energies*, vol. 11, no. 3, p. 620, Mar. 2018, doi: 10.3390/en11030620.
- [7] G. Salles, F. Filippetti, C. Tassoni, G. Grellet, and G. Franceschini, “Monitoring of induction motor load by neural network techniques,” *IEEE Trans. Power Electron.*, vol. 15, no. 4, pp. 762–768, Jul. 2000, doi: 10.1109/63.849047.
- [8] “What are Brushless DC Motors | Renesas Electronics.” <https://www.renesas.com/us/en/support/technical-resources/engineer-school/brushless-dc-motor-01-overview.html> (accessed May 18, 2020).
- [9] D. Janiszewski, “Load torque estimation in sensorless PMSM drive using unscented kalmana filter,” in *Proceedings - ISIE 2011: 2011 IEEE International Symposium on Industrial Electronics*, 2011, pp. 643–648, doi:

10.1109/ISIE.2011.5984233.

- [10] X. Gang Su, *Linear regression analysis: Theory and computing*. World Scientific Publishing Co., 2009.
- [11] “Stepwise regression - Wikipedia.” [https://en.wikipedia.org/wiki/Stepwise\\_regression](https://en.wikipedia.org/wiki/Stepwise_regression) (accessed May 19, 2020).
- [12] “Harmonic Drive LLC | Harmonic Drive | Harmonic Drive.” <https://www.harmonicdrive.net/technology/harmonicdrive> (accessed Jun. 03, 2020).
- [13] “LAUNCHXL-F28069M C2000 Piccolo MCU F28069M LaunchPad™ development kit | TI.com.” [https://www.ti.com/tool/LAUNCHXL-F28069M?jkttype=tools\\_software](https://www.ti.com/tool/LAUNCHXL-F28069M?jkttype=tools_software) (accessed Feb. 24, 2020).
- [14] “BOOSTXL-DRV8305EVM DRV8305N 3-Phase Motor Drive BoosterPack Evaluation Module | TI.com.” [https://www.ti.com/tool/BOOSTXL-DRV8305EVM?jkttype=tools\\_software](https://www.ti.com/tool/BOOSTXL-DRV8305EVM?jkttype=tools_software) (accessed Feb. 24, 2020).
- [15] “Torque (article) | Khan Academy.” <https://www.khanacademy.org/science/physics/torque-angular-momentum/torque-tutorial/a/torque> (accessed Jun. 03, 2020).
- [16] “HOMEPAGE | SOLIDWORKS.” <https://www.solidworks.com/> (accessed Jun. 03, 2020).
- [17] “MATLAB - MathWorks - MATLAB & Simulink.” <https://www.mathworks.com/products/matlab.html> (accessed May 19, 2020).
- [18] L. Number and R. February, “User Guide. InstaSPIN-FOC™ and InstaSPIN-MOTION™,” *Texas Instruments*, no. January 2013, 2017.
- [19] “ATI Industrial Automation: F/T Sensor Mini40.” [https://www.atia.com/products/ft/ft\\_models.aspx?id=Mini40](https://www.atia.com/products/ft/ft_models.aspx?id=Mini40) (accessed May 18, 2020).

ภาคผนวก 1

Asia Pacific Conference on Robot IoT System Development and Platform 2018

## Design and development of robotic arm using BLDC motor with sensorless control

Wannadear Nawae<sup>†1</sup> Dr.Kittikhun Thongpull<sup>†1</sup>

**Abstract:** In this paper, a sensorless BLDC motor control technique is investigated and experimented preliminarily. Field oriented control was employed as an algorithm to control winding activity in order to ensure the optimal rotor angle for each winding switch pattern. The experiment was performed with a developed motor test bench. The results show that the estimated signals are consist to their original. The estimated signal show their potential to be used as a vital information for sensorless robot safety system.

**Keywords:** Sensorless, Field oriented control, Back-EMF, Torque estimation

### 1. Introduction

Currently robotic arms are being used as a common equipment in various application areas such as manufacturing, medical or military. Each application requires a number of specific operational parameters for example operating space, degree of freedom, size, weight, speed and torque. In addition, in safety aspect, it is responsible for every robot to maintain accident-free operations especially robots or robotic arms that interact with human or even operate together with human. Thus to realize safety requirements, complex sensors, i.e., magnetic absolute encoder are employed to provide information that are needed in algorithms for safety condition awareness of the robot. However, delicate sensors are in exchange for cost, size, power consumption and etc. These gain the interest in research for developing sensorless techniques to obtain information of robot without using actual physical sensor, for example, as focused in this paper, torque estimation in Brushless DC Motor (BLDC). By employing sensorless technique, better speed versus torque characteristics, high dynamic response and zero electrical wear [1] have been achieved. Back-EMF signal generated from motor windings during rotation is commonly used in sensorless motor control techniques due to its relationship to rotor position, speed and torque. When apply in a robotic arm, conventional sensors can be omitted, e.g., incremental and absolute encoder, force or torque transducer, and resolver, thus, resulted in optimized cost and size while maintain performance of the operation. This work presents the investigation and preliminary development of BLDC motor control using sensorless field orientation control technique for sensing torque without a sensor to apply to robotic arm.

### 2. Methods

#### 2.1 BLDC motor control

BLDC motors are direct current motor with no brush to deliver energy to motor windings as the rotor is installed with permanent magnets thus relaying on synchronous electric machine principle. BLDC motors have a linear relationship between current and torque, voltage and speed [4]. This paper investigates on the field orientation control (FOC) for BLDC motor control due to its

mechanism that aim to produce orthogonal magnetic field between the stator and the rotor to achieve the high efficient and maximum torque.

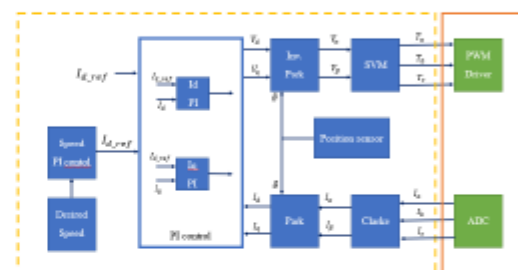


Figure 1. Block diagram of Field orientation control

#### 2.2 Torque estimation

Torque estimation acquire information from a free phase winding a motor drive control uses only two of three phases to conducting motor at a time and one phase of motor is not conducted. The free winding generates back-EMF signal at this moment. The back-EMF signal can be used to estimate the parameters such as speed and torque. In this research work, we focus on torque estimation using back-EMF information by the equation below.

$$T_e = \frac{3p}{2} (\varphi i_q + (L_d - L_q) i_d i_q) \quad (1)$$

#### 2.3 Concept of Robotic arm using sensorless

Robotic arm is a technology that imitate specific part of human that is useful to apply to various type of applications. They are being required to be operated together with human, Therefore the safety must be the main consideration. The feedback information must be in the structure of robot control system, thus, conventional sensors are required. In this research we would like to reduce cost and size of the robotic arm by replacing conventional sensors with the investigated speed and torque estimation system using back-EMF information.

<sup>†1</sup> Department of Electrical Engineering, Prince of Songkla University



### 3. Result

#### 3.1 Experimental setup

To investigate the behaviors and operation of sensorless BLDC motor control technique, we setup an experiment for estimation torque and speed by using TMS320F28069 microcontroller from Texas Instrument TI and DRV8035 a gate drive IC with back-EMF frontend for microcontroller interface. An incremental encoder was used for comparing between actual speed and estimated ones.

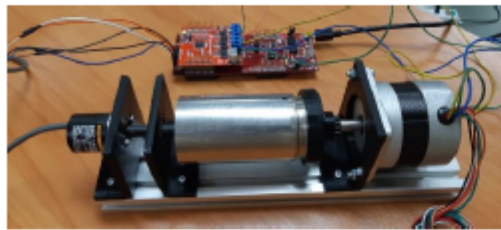


Figure 2. Experiment setup

#### 3.2 Experiment results

From Figure 3 we found that the speed from estimation and sensor similar to each other but when the speed is changed the error of estimation was increased. Figure 4 is the estimation results of torque and speed. When load has changed while motor is running, we found that torque is increased but the speed is decreased.

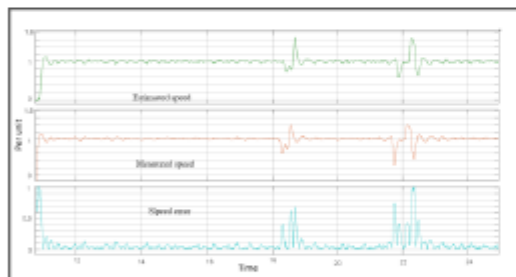


Figure 3. Result between actual speed and estimate speed

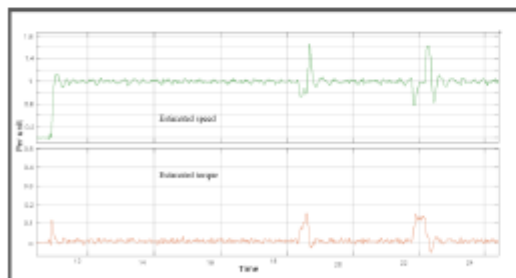


Figure 4. Result between torque and speed estimation

### 4. Conclusion

This work presents an investigation of speed and torque estimate in BLDC motor as a sensorless field orientation control technique. The experiment results show that sensorless technique can estimate value from back-EMF to analyze speed and torque correctly and motor control procedure still be able to spin motor without using sensor. The next step is to use torque estimates based on back-EMF for apply to robotic arm in safety procedure.

### Appendix

Table I. Motor parameters

|                      |         |
|----------------------|---------|
| Number of pole (P)   | 6       |
| DC bus voltage (Vdc) | 24      |
| Resistance (R)       | 0.4357  |
| Inductances (Ls)     | 0.00088 |

### Reference

- [1] "Absolute position sensing: the key to better brushless DC motor control-Electronic-Products." [Online]. Available: [https://www.electronicproducts.com/Analog\\_Mixed\\_Signal\\_ICs/Sensors/Absolute\\_position\\_sensing\\_the\\_key\\_to\\_better\\_brushless\\_DC\\_motor\\_control.aspx](https://www.electronicproducts.com/Analog_Mixed_Signal_ICs/Sensors/Absolute_position_sensing_the_key_to_better_brushless_DC_motor_control.aspx). [Accessed: 28-Nov-2017].
- [2] "Sensorless Field Oriented Control of 3-Phase Permanent Magnet Synchronous Motors" [Online]. Available: <http://www.ti.com.cn/lit/an/sprabq3/sprabq3.pdf>. [Accessed: 20-Apr-2018].
- [3] Y. Liu, Z. Q. Zhu, and D. Howe, "Instantaneous Torque Estimation in Sensorless Direct-Torque-Controlled Brushless DC Motors," *IEEE Transactions on Industry Applications*, vol. 42, no. 5, pp. 1275–1283, Sep. 2006.
- [4] J. C. Gamazo-Real, E. Vázquez-Sánchez, and J. Gómez-Gil, "Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application Trends," *Sensors (Basel)*, vol. 10, no. 7, pp. 6901–6947, Jul. 2010.

## ประวัติผู้เขียน

ชื่อ ชื่อสกุล นายวันนาเดีย นาแว  
รหัสประจำตัวนักศึกษา 6010120068  
วุฒิการศึกษา  
วุฒิ ชื่อสถาบัน ปีที่สำเร็จการศึกษา  
วิศวกรรมศาสตรบัณฑิต มหาวิทยาลัยสงขลานครินทร์ 2559  
(วิศวกรรมเมคาทรอนิกส์)

## ทุนการศึกษาที่ได้รับ (ที่ได้รับระหว่างการการศึกษา)

1. ทุนศิษย์ก้นกุฏิ คณะวิศวกรรมศาสตร์
2. ทุนอุดหนุนการวิจัยเพื่อวิทยานิพนธ์ มหาวิทยาลัยสงขลานครินทร์