



LAB2 : Actuator Report

Name

- | | | |
|---------------|---------------|-------------|
| • นายวรภัทร | ภัทรเปรมเจริญ | 66340500049 |
| • นายศุภวิชญ์ | แก้วปิ่น | 66340500055 |
| • นายภูริภัทร | โต๊ะใบเต้ | 66340500067 |

Objective

- เพื่อให้สามารถออกแบบการทดลองโดยใช้กระบวนการทางวิทยาศาสตร์ได้อย่างถูกต้องตลอดการทำการทดลอง
- เพื่อให้ได้ศึกษา และเข้าใจการทำงาน การใช้งานอุปกรณ์ต่างๆในการทดลองได้ดียิ่งขึ้น
- สามารถใช้หลักการทางคณิตศาสตร์ สถิติเพื่อเก็บผลการทดลอง วิเคราะห์ได้อย่างถูกต้องมีเหตุผลที่น่าเชื่อถือรองรับ และสามารถตรวจสอบกับทฤษฎีที่นำมาใช้อ้างอิงได้
- เพื่อให้เข้าใจถึงวิธีการรับรู้ปริมาณ การเปลี่ยนแปลง และการแปลงค่าของอุปกรณ์
- เพื่อให้สามารถกำหนดตัวแปรได้อย่างถูกต้อง การตั้งสมมุติฐานมีความสอดคล้องกับตัวแปรที่ได้ตั้งไว้
- เพื่อให้สามารถเขียนรายงานทางวิทยาศาสตร์ได้อย่างถูกต้อง และเหมาะสมตามหลักสากล

1 Brushless DC Motor

1.1 จุดประสงค์

- 1.1.1 เพื่อเข้าใจหลักการทำงาน โครงสร้าง และลักษณะสำคัญของ Brushless DC Motor
- 1.1.2 สามารถใช้โปรแกรมคอมพิวเตอร์ควบคุม และกำหนดค่าพารามิเตอร์ในการทำงานของมอเตอร์ได้
- 1.1.3 เพื่อให้เข้าใจความสัมพันธ์ของความถี่สัญญาณ PWM กับความเร็วของมอเตอร์ และวิธีการใช้ Speed sensing เพื่อวัดความเร็วการหมุนของมอเตอร์
- 1.1.4 เพื่อให้เข้าใจพารามิเตอร์ที่จำเป็นต่อการควบคุม Brushless DC Motor ทั้งในด้านแมคคานิค และอิเล็กทรอนิกส์
- 1.1.5 เพื่อเรียนรู้วิธีการการควบคุม Brushless DC Motor และเข้าใจความแตกต่างของวิธีการควบคุม
- 1.1.6 เพื่อให้เข้าใจวิธีการใช้งาน 4 Channels Oscilloscope ได้ รวมถึงสามารถอธิบายลักษณะสัญญาณ Pulse ทั้ง 3 เฟสของ Brushless DC Motor ที่อ่านจาก Oscilloscope ได้
- 1.1.7 สามารถประยุกต์ใช้งานใช้โปรแกรม MATLAB ร่วมกับ โปรแกรม STM32Cube ในการควบคุมมอเตอร์ได้
- 1.1.8 เพื่อให้สามารถคำนวณความเร็วของมอเตอร์โดยใช้ความถี่ของสัญญาณ Pulse ที่อ่านจาก Oscilloscope

1.2 สมมุติฐาน

ความเร็วของ BLDC motor ที่คำนวณได้จากสัญญาณ Oscilloscope จะสอดคล้องกับความเร็วที่ตั้งค่าไว้ในโปรแกรม Motor Workbench 6.3.2

1.3 ตัวแปร

- 1.3.1 ตัวแปรต้น : ความเร็วที่ตั้งค่าไว้ในโปรแกรม Motor Workbench 6.3.2
- 1.3.2 ตัวแปรตาม : ความถี่ของสัญญาณที่อ่านได้จาก Tablet oscilloscope
- 1.3.3 ตัวแปรควบคุม : แรงดันไฟฟ้าที่ใช้ขับเคลื่อน, ชุดบอร์ดการเรียนรู้สำเร็จรูป BLDC, บอร์ดควบคุม Nucleo-G474RE, โปรแกรมควบคุม และการตั้งค่าอื่นๆในโปรแกรม Motor Workbench 6.3.2

1.4 นิยามศัพท์เฉพาะ

1.4.1 Brushless DC Motor (BLDC) จากนั้นจะขอเรียกว่า BLDC คือ มอเตอร์แบบไร้แปรงถ่าน 3 เฟส มีประสิทธิภาพด้านความเร็วที่ดีว่ามอเตอร์แบบ Brushed DC Motor ทั่วไป หลักการทำงานเป็นการจ่ายกระแสไฟฟ้าเข้าขดลวดเหนี่ยวนำเพื่อให้เกิดสนามแม่เหล็กชั่วคราว ซึ่งเป็นหลักการเดียวกันกับ Brushed DC Motor

1.4.2 Trapezoidal Back EMF คือ รูปแบบของแรงดันไฟฟ้าที่เกิดจากมอเตอร์แบบไร้แปรงถ่าน (Brushless DC Motor หรือ BLDC) เมื่อทำงานเป็นเครื่องกำเนิดไฟฟ้า หรือแรงดันย้อนกลับ (Back Electromotive Force) ที่เกิดจากการหมุนของโรเตอร์ตัดผ่านสนามแม่เหล็กในสเตเตอร์ โดยลักษณะของแรงดันไฟฟ้าย้อนกลับนี้มีรูปร่างเป็นสัญญาณคล้าย "คลื่นขั้นบันได" (trapezoidal waveform) ซึ่งต่างจากมอเตอร์ชนิดอื่น เช่น มอเตอร์เชิงโครนัสแบบ AC ที่แรงดันไฟฟ้ามีรูปคลื่นเป็นไซน์เวฟ (sine wave)

1.4.3 Trapezoidal Control คือ หนึ่งในวิธีการควบคุมมอเตอร์กระแสตรงไร้แปรงถ่าน (Brushless DC Motor: BLDC) โดยมีลักษณะการควบคุมกระแสในขดลวดที่เรียกว่า Trapezoidal Commutation ซึ่งรูปคลื่นของแรงดันหรือกระแสที่ป้อนให้ขดลวดจะมีลักษณะเป็นรูปสามเหลี่ยมคว่ำหรือ trapezoidal waveforms แทนที่จะเป็นรูป sine wave (ในกรณีของ Sinusoidal Control)

1.4.4 Field-Oriented Control (FOC) หรือ Vector Control คือเทคนิคการควบคุมที่นิยมใช้ในการควบคุมมอเตอร์ไฟฟ้า โดยเฉพาะมอเตอร์ที่ใช้ในงานที่ต้องการความแม่นยำสูง เช่น Brushless DC Motor (BLDC) หรือ Permanent Magnet Synchronous Motor (PMSM) เทคนิคนี้ช่วยให้สามารถควบคุมแรงบิด (Torque) และความเร็ว (Speed) ของมอเตอร์ได้อย่างมีประสิทธิภาพและแม่นยำ

1.4.5 Pulse Width Modulation (PWM) คือเทคนิคในการปรับค่าของสัญญาณดิจิทัลเพื่อควบคุมกำลังไฟฟ้าที่ส่งไปยังโหลด เช่น มอเตอร์, หลอดไฟ, หรืออุปกรณ์อิเล็กทรอนิกส์อื่น ๆ โดยการปรับเปลี่ยนความกว้างของพัลส์ในสัญญาณความถี่คงที่ เพื่อให้ได้ค่ากำลังเฉลี่ยที่ต้องการ

1.4.6 Commutation Logic เป็นกระบวนการที่ใช้ในการควบคุมการทำงานของมอเตอร์กระแสตรงไร้แปรงถ่าน (Brushless DC Motor หรือ BLDC Motor) โดยการควบคุมนี้เกี่ยวข้องกับการจ่ายแรงดันหรือกระแสไฟฟ้าไปยังขดลวดสเตเตอร์ (Stator) ในลำดับที่ถูกต้องเพื่อสร้างสนามแม่เหล็กหมุน ซึ่งทำให้โรเตอร์ (Rotor) หมุนตามสนามแม่เหล็กนั้นได้อย่างมีประสิทธิภาพ

1.4.7 Six-Step Commutation (หรือที่เรียกว่า Trapezoidal Commutation) เป็นวิธีการควบคุมการจ่ายแรงดันไฟฟ้าไปยังขดลวดของมอเตอร์ Brushless DC Motor (BLDC) โดยแบ่งเป็น 6 ขั้นตอน (หรือ 6 steps) ต่อการหมุนครบหนึ่งรอบไฟฟ้า (Electrical Cycle) ของโรเตอร์

1.4.8 Space Vector Modulation (SVM) เป็นเทคนิคขั้นสูงที่ใช้ในการสร้างรูปคลื่นแรงดันไฟฟ้าแบบ Pulse Width Modulation (PWM) สำหรับควบคุมอินเวอร์เตอร์ 3 เฟส ซึ่งถูกออกแบบมาเพื่อให้มอเตอร์ AC หรือ Brushless DC (BLDC) ทำงานได้อย่างมีประสิทธิภาพสูงสุด โดยลดการสั่นสะเทือน (Torque Ripple) และเพิ่มความราบรื่นในการทำงาน โดย SVM ถูกนำไปใช้ในระบบควบคุมมอเตอร์ที่ต้องการแรงบิดที่แม่นยำ เช่น Field-Oriented Control (FOC) สำหรับมอเตอร์แบบ BLDC, PMSM หรือ Induction Motor

1.4.9 Sensorless Control คือการควบคุมมอเตอร์โดยไม่ใช้เซ็นเซอร์ตรวจจับตำแหน่งของโรเตอร์ เช่น Hall Sensors หรือ Encoders ในกรณีของมอเตอร์แบบ Brushless DC (BLDC) หรือ Permanent Magnet Synchronous Motor (PMSM) เทคนิคนี้ใช้การวิเคราะห์สัญญาณไฟฟ้าหรือสมการทางคณิตศาสตร์เพื่อประมาณตำแหน่งและความเร็วของโรเตอร์

1.4.10 Tablet Oscilloscope จากนั้นจะขอเรียกว่า Oscilloscope คือ อุปกรณ์วัดสัญญาณแรงดันไฟฟ้าและแสดงผลออกมาเป็นกราฟตามเวลา โดยใช้ scope จับที่ขั้วแรงดันไฟฟ้า บวก และ สายที่แยกออกมาจับที่ขั้วสัญญาณ ลบ หรือ กราวด์ (GND) ในการทดลองนี้ใช้เป็น Oscilloscope แบบ 4 channels เพื่อให้เพียงพอต่อการอ่านสัญญาณ BLDC 3 เฟส Adapter คือ อุปกรณ์ที่ใช้แปลงแรงดันไฟฟ้าจาก 220 V เป็นแรงดันไฟ 24 V และนำไปใช้ในส่วนต่างๆของวงจร บอร์ด หรือมอเตอร์ ในบางครั้งอาจจะเรียกว่า ไฟเลี้ยง

1.4.11 บอร์ดการเรียนรู้ BLDC คือ บอร์ดการเรียนรู้สำเร็จรูปที่ได้รับมาจาก TA ใช้ในการทดลอง BLDC เพื่อช่วยให้การจ่ายไฟเลี้ยงมอเตอร์ทำได้ง่ายขึ้น ลดขั้นตอนที่ไม่จำเป็นต่อการทดลอง และช่วยให้สามารถใช้ Oscilloscope จับสัญญาณได้ง่าย ภายในบอร์ดการเรียนรู้ BLDC มีอุปกรณ์ที่สำคัญ ดังนี้

- 1) Step Down Module คือ วงจร หรือบอร์ดที่ใช้ลดแรงดันไฟฟ้า หรือไฟเลี้ยงให้อยู่ในระดับที่ต้องการ
- 2) บอร์ด Nucleo-G474RE หลังจากนั้นจะขอเรียกว่า บอร์ดควบคุม คือ ส่วนที่รับคำสั่งการทำงานจากโปรแกรม คอมพิวเตอร์มาสั่งการไปยังบอร์ดอื่นๆ เพื่อควบคุมมอเตอร์ให้สามารถทำงานได้
- 3) สาย Scope หมายถึง สายรับสัญญาณของ Oscilloscope ที่มีลักษณะเฉพาะของลวดนำไฟฟ้าเพื่อให้อ่านสัญญาณ ได้แม่นยำนิยมเชิงปฏิบัติการ

4) BLDC (Brushless DC Motor) ที่ใช้ในการทดลอง คือ มอเตอร์ Brushless รุ่น A2212/13T 1000Kv เป็น BLDC แบบ Sensorless Controls 3 เฟส ควบคุมโดย บอร์ด X-Nucleo IHM08M1 ที่แรงดันไฟ 12 โวลต์



รูปที่ 1.1 รูปตัวอย่าง BLDC 3 เฟส รุ่น A2212/13T

ที่มา : <https://www.amazon.in/Brushless-Motor-1000kv-BLDC-Quadcopter/dp/B08DQDSZRD>

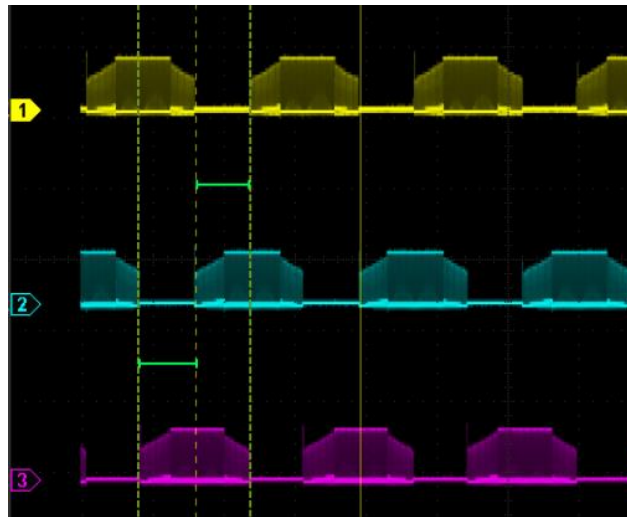
1.4.12 สัญญาณ Pulse คือ สัญญาณ Square wave ที่ใช้ควบคุมการเปิดปิดเฟสการทำงานมอเตอร์ ลักษณะของสัญญาณจะขึ้นอยู่กับ แรงดัน และความถี่ เช่น ความสูงของสัญญาณจะขึ้นอยู่กับแรงดันไฟฟ้าที่จ่ายเข้าไป และระยะห่างระหว่าง Pulse อาจมีความเกี่ยวข้องกับความเร็วตามสมมติฐานการทดลอง

1.4.13 ความเร็วของมอเตอร์ คือ ค่าความเร็วในการหมุนของมอเตอร์ วัดโดยใช้โปรแกรม MotorControl Workbench เวอร์ชัน 6.3.2 และ โปรแกรมMotor Pilot มีหน่วยเป็น RPM (Revolutions per minute)

1.4.14 Oscilloscope Series STO1004 คืออุปกรณ์วัดสัญญาณแรงดันไฟฟ้า ที่สามารถแสดงภาพออกมาเป็นกราฟได้ มีช่องรับสัญญาณทั้งหมด 4 ช่องสัญญาณ สามารถบันทึกหน้าจอ และเชื่อมต่อกับคอมพิวเตอร์ส่วนตัวเพื่อส่งออกภาพได้ ในการทดลองนี้ ใช้ในการวัด Pulse ที่จ่ายเข้าไปยัง BLDC

1.4.15 บอร์ดไดร์ฟเวอร์ คือ บอร์ด X-Nucleo IHM08M1 เป็นบอร์ดส่วนเสริมของซีรีส์ Nucleo ใช้สำหรับควบคุม BLDC (Brushless DC Motor) โดยเฉพาะ สามารถควบคุมได้ 2 ระบบ คือ 6-Step control และ FOC (Field oriented control) ใช้แรงดันไฟฟ้าทำงานในช่วง 10 ถึง 48 Vdc

1.4.16 Trapezoidal Control คือ รูปแบบหนึ่งของการควบคุมมอเตอร์แบบ BLDC (Brushless DC Motor) โดยเน้นการป้อนกระแสไฟฟ้าเข้าสู่ขดลวด (Stator Winding) ในรูปแบบของแรงดันหรือกระแสรูปคลื่น Trapezoidal เพื่อสร้างแรงบิดสำหรับหมุนโรเตอร์ (Rotor) อย่างต่อเนื่องตามลำดับตำแหน่งของโรเตอร์



รูปที่ 1.2 รูปตัวอย่างสัญญาณ Pulse จากทั้ง 3 เฟสของมอเตอร์

- 1) Peak หมายถึง ค่าสูงสุดของค่าใดค่าหนึ่ง คำนี้อาจถูกใส่ที่ท้ายหน่วยต่าง ๆ ที่สื่อถึงค่าสูงสุดที่เป็นไปได้ของค่านั้น
- 2) ระยะห่างของสัญญาณ Pulse หรือคาบของสัญญาณ คือ ระยะเวลาของสัญญาณ 1 ลูก มีหน่วยเป็น μs (ไมโครวินาที)

1.5 นิยายเชิงปฏิบัติการ

1.5.1 Trapezoidal Back EMF คือ รูปแบบของแรงดันไฟฟ้าที่เกิดจากมอเตอร์แบบไร้แปรงถ่าน (Brushless DC Motor หรือ BLDC) เมื่อทำงานเป็นเครื่องกำเนิดไฟฟ้า หรือแรงดันย้อนกลับ (Back Electromotive Force) ที่เกิดจากการหมุนของโรเตอร์ตัดผ่านสนามแม่เหล็กในสเตเตอร์ โดยลักษณะของแรงดันไฟฟ้าย้อนกลับนี้มีรูปร่างเป็นสัญญาณคล้าย "คลื่นขั้นบันได" (trapezoidal waveform) ซึ่งต่างจากมอเตอร์ชนิดอื่น เช่น มอเตอร์ซิงโครนัสแบบ AC ที่แรงดันไฟฟ้ามีรูปคลื่นเป็นไซน์เวฟ (sine wave)

1.5.2 Trapezoidal Control คือ หนึ่งในวิธีการควบคุมมอเตอร์กระแสตรงไร้แปรงถ่าน (Brushless DC Motor: BLDC) โดยมีลักษณะการควบคุมกระแสในขดลวดที่เรียกว่า Trapezoidal Commutation ซึ่งรูปคลื่นของแรงดันหรือกระแสที่ป้อนให้ขดลวดจะมีลักษณะเป็นรูปสามเหลี่ยมคว่ำหรือ trapezoidal waveforms แทนที่จะเป็นรูป sine wave (ในกรณีของ Sinusoidal Control)

1.5.3 Field-Oriented Control (FOC) หรือ Vector Control คือเทคนิคการควบคุมที่นิยมใช้ในการควบคุมมอเตอร์ไฟฟ้า โดยเฉพาะมอเตอร์ที่ใช้ในงานที่ต้องการความแม่นยำสูง เช่น Brushless DC Motor (BLDC) หรือ Permanent Magnet Synchronous Motor (PMSM) เทคนิคนี้ช่วยให้สามารถควบคุมแรงบิด (Torque) และความเร็ว (Speed) ของมอเตอร์ได้อย่างมีประสิทธิภาพและแม่นยำ

1.5.4 Commutation Logic เป็นกระบวนการที่ใช้ในการควบคุมการทำงานของมอเตอร์กระแสตรงไร้แปรงถ่าน (Brushless DC Motor หรือ BLDC Motor) โดยการควบคุมนี้เกี่ยวข้องกับการจ่ายแรงดันหรือกระแสไฟฟ้าไปยังขดลวดสเตเตอร์ (Stator) ในลำดับที่ถูกต้องเพื่อสร้างสนามแม่เหล็กหมุน ซึ่งทำให้โรเตอร์ (Rotor) หมุนตามสนามแม่เหล็กนั้นได้อย่างมีประสิทธิภาพ

1.5.5 Six-Step Commutation (หรือที่เรียกว่า Trapezoidal Commutation) เป็นวิธีการควบคุมการจ่ายแรงดันไฟฟ้าไปยังขดลวดของมอเตอร์ Brushless DC Motor (BLDC) โดยแบ่งเป็น 6 ขั้นตอน (หรือ 6 steps) ต่อการหมุนครบหนึ่งรอบไฟฟ้า (Electrical Cycle) ของโรเตอร์

1.5.6 Space Vector Modulation (SVM) เป็นเทคนิคขั้นสูงที่ใช้ในการสร้างรูปคลื่นแรงดันไฟฟ้าแบบ Pulse Width Modulation (PWM) สำหรับควบคุมอินเวอร์เตอร์ 3 เฟส ซึ่งถูกออกแบบมาเพื่อให้มอเตอร์ AC หรือ Brushless DC (BLDC) ทำงานได้อย่างมีประสิทธิภาพสูงสุด โดยลดการสั่นสะเทือน (Torque Ripple) และเพิ่มความราบรื่นในการทำงาน โดย SVM ถูกนำไปใช้ในระบบควบคุมมอเตอร์ที่ต้องการแรงบิดที่แม่นยำ เช่น Field-Oriented Control (FOC) สำหรับมอเตอร์แบบ BLDC, PMSM หรือ Induction Motor

1.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

1.6.1 Data Sheet ของ BLDC A2212/13T

1.6.2 Data Sheet ของ X-nucleo IHM08M1

1.6.3 Revolutions per minute (RPM)

คือหน่วยวัดความเร็วในการหมุนของวัตถุที่ใช้เพื่อบอกจำนวนรอบที่วัตถุนั้นหมุนในหนึ่งนาที หรือจำนวนการหมุนของเฟลา หรือส่วนหมุนในเวลา 1 นาที

สูตรการคำนวณคือ

$$RPM = \frac{Frequency \times 60}{Number\ of\ Pole\ Pairs}$$

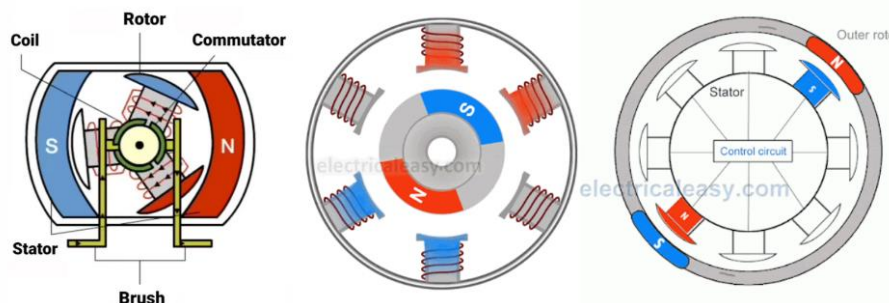
Frequency คือ ความถี่ของสัญญาณ Back EMF (หน่วย Hz)

Number of Pole Pairs คือ จำนวนคู่ขั้วแม่เหล็กในมอเตอร์ (จำนวนขั้วแม่เหล็กทั้งหมด ÷ 2)

1.6.4 Brushless DC Motor (BLDC) จากนั้นจะขอเรียกว่า BLDC คือ มอเตอร์แบบไร้แปรงถ่าน 3 เฟส มีประสิทธิภาพด้านความเร็วที่ดีกว่ามอเตอร์แบบ Brushed DC Motor ทัวไป หลักการทำงานเป็นการจ่ายกระแสไฟฟ้าเข้าขดลวดเหนี่ยวนำเพื่อให้เกิดสนามแม่เหล็กชั่วคราว ซึ่งเป็นหลักการเดียวกันกับ Brushed DC Motor แต่มีความแตกต่างกันดังนี้

1) Brushed จะมีก้านส่งกระแสไฟ (แปรงถ่าน) เพื่อใช้เปลี่ยนการจ่ายกระแสไฟฟ้าเข้าขดลวดแต่ละชุด ความเร็วในการสลับชุดขดลวดก็จะมีความสัมพันธ์กับความเร็วการหมุนของมอเตอร์ กลับกัน หากเป็น BLDC (Brushless DC Motor) จะไม่มีก้านส่งกระแสไฟ (แปรงถ่าน) เพื่อสลับการทำงานของชุดขดลวด แต่จะใช้สัญญาณ Pulse เพื่อสลับการทำงานของชุดขดลวดแทน ซึ่งวิธีการจ่ายสัญญาณ Pulse เพื่อให้มอเตอร์หมุนต่อไปได้อย่างราบรื่นจะถูกพูดถึงในหัวข้อถัดไป

2) โครงสร้างของ Brushed จะมีแม่เหล็กถาวรเป็นส่วน Stator และชุดขดลวดเป็น Rotor รูปด้านซ้าย แต่ BLDC (Brushless DC Motor) มีโครงสร้างภายในกลับกัน คือ แม่เหล็กถาวรเป็น Rotor ชุดขดลวดเป็น Stator หากแม่เหล็กถาวรหมุนอยู่ตรงกลางด้านในชุดขดลวดตามรูปตรงกลาง เราเรียกว่า Inrunner BLDC หากแม่เหล็กถาวรหมุนอยู่ด้านนอกชุดขดลวดตามรูปด้านขวา เราจะเรียกว่า Outrunner BLDC

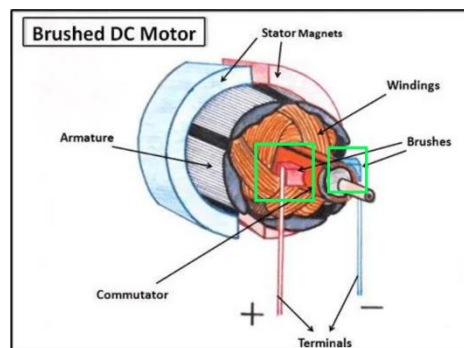


รูปที่ 1.3 เปรียบเทียบโครงสร้างของ Brushed และ BLDC และลักษณะของ Rotor

ที่มา : <https://www.electricaleasy.com/2015/05/brushless-dc-bldc-motor.html>

ที่มา : https://www.instagram.com/mechatronix_saran/reel/C8SR0xdScwR/

3) แปรงถ่าน หรือก้านส่งกระแสไฟ หมายถึง แผ่นเหล็กบาง หรือแท่งเหล็กที่ใช้สัมผัสกับ Rotor ของมอเตอร์ Brushed เพื่อสลับการทำงานของชุดขดลวด มีลักษณะดังรูปในกรอบสี่เหลี่ยม



รูปที่ 1.4 โครงสร้างของ Brushed Motor และลักษณะแปรงถ่าน (ในกรอบสี่เหลี่ยม)

ภาพจาก <https://th.mouser.com/applications/dont-ignore-the-brushed-dc-motor/>

1.7 วิธีดำเนินการทดลอง

- 1.7.1 เริ่มจากติดตั้งสายสโคป (Scope) ที่บอร์ดการเรียนรู้ และประกอบบอร์ดไดรฟ์เวอร์ บอร์ดการเรียนรู้ และบอร์ดควบคุมเข้าด้วยกัน
- 1.7.2 อัปโหลดโค้ดจากโปรแกรมควบคุมลงไปยังบอร์ดควบคุม
- 1.7.3 เริ่มสั่งการให้มอเตอร์หมุน โดยเริ่มที่ความเร็ว 1000 RPM ตามค่าความเร็วที่โปรแกรมคำนวณมาให้ เก็บภาพผลการทดลองจาก Oscilloscope เพื่อนำมาวิเคราะห์ และเพิ่มความเร็วครั้งละ 1000 RPM ไปจนถึง 10,000 RPM ซึ่งเป็นค่าสูงสุดของการทดลองนี้
- 1.7.4 นำภาพทั้งหมดที่ได้มาวิเคราะห์ความสัมพันธ์ของสัญญาณกับความเร็วที่กำหนด

1.8 วัสดุอุปกรณ์

- 1.8.1 บอร์ดควบคุม Nucleo G474RE
- 1.8.2 บอร์ดไดรฟ์เวอร์ X-Nucleo IHM08M1
- 1.8.3 บอร์ดการเรียนรู้สำเร็จรูป
- 1.8.4 Brushless DC Motor รุ่น A2212/13T
- 1.8.5 Tablet Oscilloscope Series STO1004
- 1.8.6 คอมพิวเตอร์ของผู้ทดลอง
- 1.8.7 โปรแกรม STM32CubeMX, STM32CubeIDE, Motorcontrol Workbench และ Motor Pilot
- 1.8.8 Adaptor 220 V AC to 24 V DC
- 1.8.9 สาย USB TypeC to TypeC เชื่อมต่อ Oscilloscope กับ คอมพิวเตอร์
- 1.8.10 สาย USB type A เชื่อมบอร์ดกับคอมพิวเตอร์

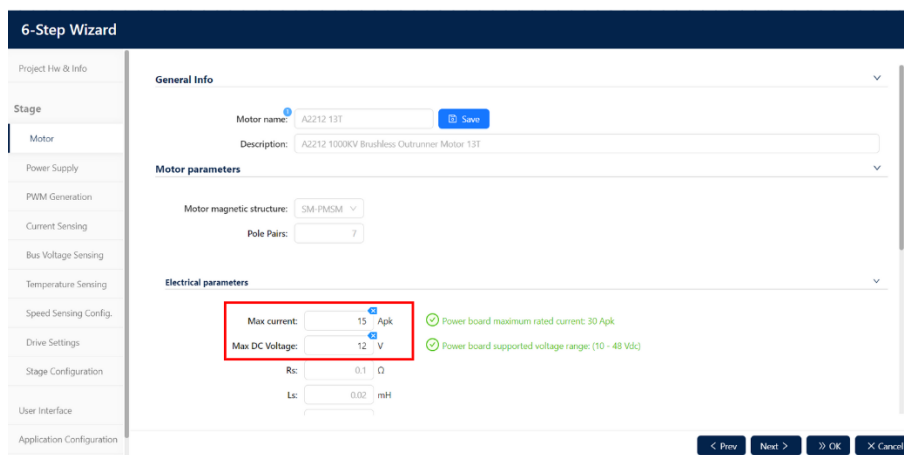
1.9 ขั้นตอนการดำเนินงาน

- 1.9.1 ขั้นตอนเตรียมอุปกรณ์ และเตรียมโปรแกรม
 - 1) ประกอบบอร์ดไดรฟ์เวอร์ บอร์ดการเรียนรู้ และบอร์ดควบคุมเข้าด้วยกัน จ่ายไฟเลี้ยงเข้าบอร์ดการเรียนรู้ และปรับโมดูลลดแรงดัน (step down module) ให้ได้ไฟเอาต์พุตที่ 12 V
 - 2) ดาวน์โหลดโปรแกรมควบคุมตามคู่มือที่ได้รับจาก TA รายการโปรแกรมที่ดาวน์โหลดมีดังนี้
 - 2.1) MotorControl Workbench เวอร์ชัน 6.3.2
 - 2.2) เมื่อติดตั้งเรียบร้อยแล้ว จะได้โปรแกรม Motor Pilot 6.3.2 ติดตั้งตามมาด้วย ซึ่งเป็นโปรแกรมหลักที่ใช้สั่งการทำงานของมอเตอร์



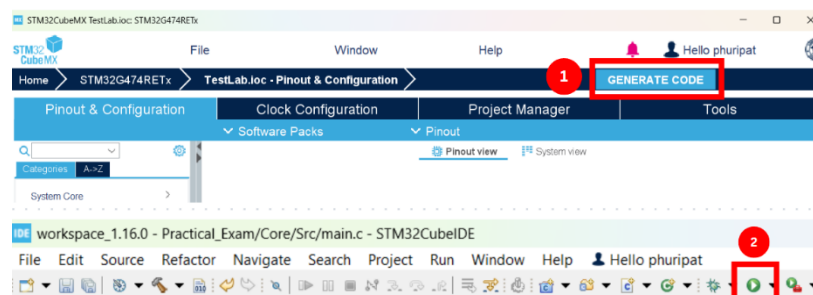
รูปที่ 1.5 โปรแกรมควบคุมมอเตอร์ที่ติดตั้งตามคู่มือของ TA

2.3) ตั้งค่า และสร้างโปรเจกตามคู่มือที่ได้รับจาก TA การตั้งค่านี้นี้เป็นการควบคุมแบบ 6-step control ตัวแปรสำคัญที่ต้องตั้งค่าใหม่คือ Max current ตั้งค่าใหม่เป็น 15 Apeak และ Max DC Voltage ตั้งค่าใหม่เป็น 12 V



รูปที่ 1.6 หน้าต่างการตั้งค่า ภายในโปรแกรม Motor Control Workbench

3) เมื่อตั้งค่าตัวแปรที่สำคัญในโปรแกรมเรียบร้อยแล้ว ให้กดที่ Generate the project เพื่อให้โปรแกรมสร้างโค้ดต้นแบบสำหรับการควบคุมมอเตอร์ ในขั้นตอนนี้จะเป็นการใช้งานโปรแกรม STM32CubeMX และกด Generate Code เพื่อช่วยตั้งค่าพารามิเตอร์ของบอร์ดควบคุม และขั้นตอนสุดท้าย ให้เปิดโปรแกรม STM32CubeIDE เพื่ออัปเดตการตั้งค่าทั้งหมดลงบอร์ดควบคุม

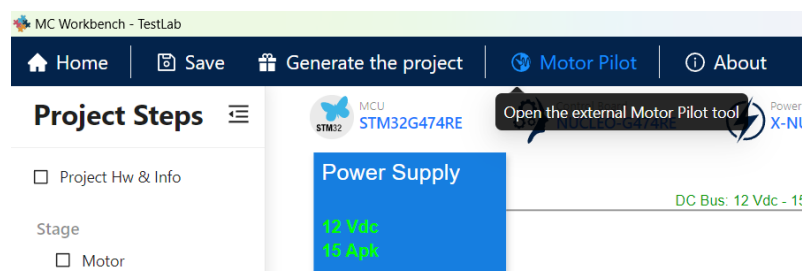


รูปที่ 1.7 การกด Generate code (1) และการกดอัปเดตโค้ด (2) ในโปรแกรม STM32CubeIDE

- 4) ใช้สาย scope จาก Tablet Oscilloscope จับที่ขา Output สัญญาณ Pulse ที่บอร์ดการเรียนรู้ โดยจับขา 1 เข้ากับ Oscilloscope ช่อง 1 ขาที่ 2 กับช่อง 2 และขาที่ 3 กับช่องที่ 3
- 5) ตั้งค่า Oscilloscope ให้เห็นสัญญาณทั้ง 3 สัญญาณ ในเวลาเดียวกัน โดยใช้วิธีกดปุ่ม Auto หรือใช้นิ้วมือแตะหน้าจอ ลากสัญญาณให้แยกออกจากกัน ตั้งค่าช่องเวลาที่ 500 μ s /div หรือปรับเพิ่มลดได้ตามความเหมาะสมของสัญญาณที่อ่านได้ ระวังการใช้นิ้วมือสองนิ้วแตะและลากเข้า-ออก จะเป็นการย่อขยายขนาดสัญญาณ อาจทำให้อ่านผิดพลาดได้
- 6) ใช้สายเชื่อมต่อแบบ typeC to TypeC เพื่อเชื่อมต่อ Oscilloscope และคอมพิวเตอร์ที่ใช้บันทึกภาพ

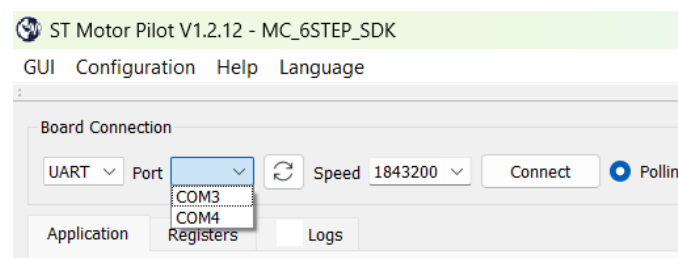
1.9.2 ขั้นตอนการเก็บค่าที่จำเป็น และสังเกตสัญญาณ

- 1) เมื่อติดตั้งตามขั้นตอนเตรียมอุปกรณ์เรียบร้อยแล้ว ให้เปิดโปรแกรม Motor Pilot จากโปรแกรม Motor Control Workbench ที่แถบด้านบนของโปรแกรมตามภาพที่ ***



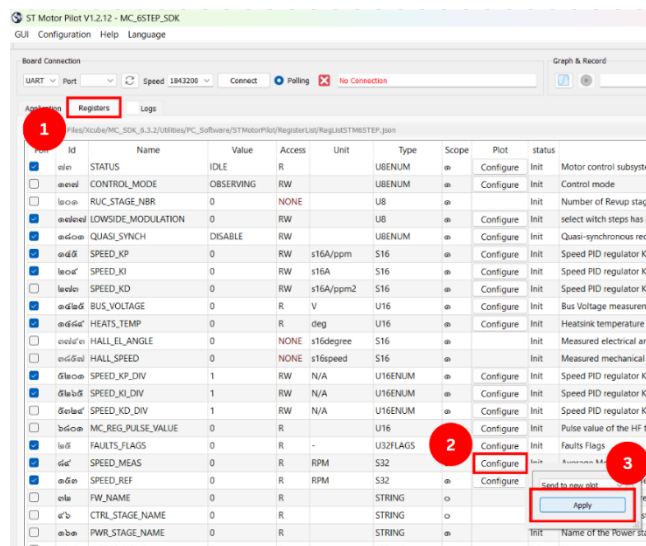
รูปที่ 1.8 แสดงวิธีเปิดใช้งานโปรแกรม Motor Pilot

- 2) เมื่อโปรแกรมถูกเปิดขึ้นมา จะเป็นสถานะที่พร้อมสำหรับการทำงานทันที หากเปิดโปรแกรมแล้วยังไม่สามารถใช้คำสั่งหมุนมอเตอร์ได้ ให้กดที่ port เลือก port ของคอมพิวเตอร์ที่เชื่อมต่อกับบอร์ดควบคุมอยู่ และกด Connect จะสามารถกลับมาใช้งานได้เหมือนเดิม



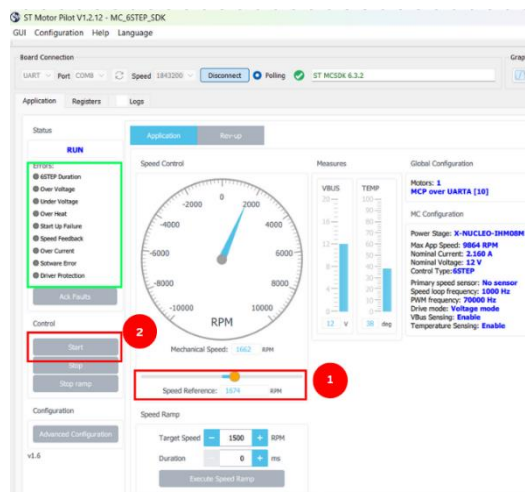
รูปที่ 1.9 การกดเชื่อมต่อ Port ใหม่ของโปรแกรม Motor Pilot

3) เปิดกราฟความเร็วจากโปรแกรม Motor Pilot โดยไปที่แถบชื่อว่า Register หาแถวที่เขียนกำกับว่า SPEED_MEAS (Speed measure) ทางขวาในคอลลัมน์ plot กดที่ Configure และ Apply จะปรากฏหน้าต่างใหม่ขึ้นและเริ่มพล็อตกราฟความเร็วจากการคำนวณของโปรแกรมในหน่วย RPM ในหน้า Register ยังสามารถดูกราฟอื่นที่โปรแกรมได้คำนวณไว้เพิ่มเติมได้อีกด้วย



รูปที่ 1.10 การเปิดใช้งานกราฟต่าง ๆ ใน โปรแกรม Motor Pilot

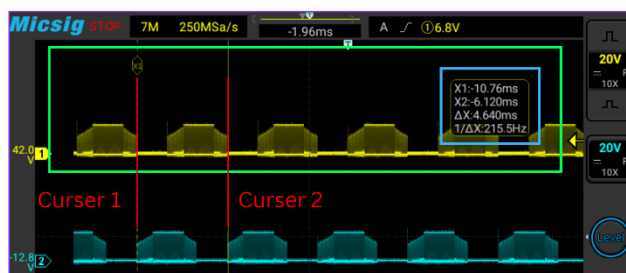
4) กำหนดความเร็วที่แถบ Slide Bar ด้านล่าง แล้วกดที่ปุ่ม Start ตามภาพที่ *** มอเตอร์จะเริ่มหมุน หากมอเตอร์หยุดหมุนหลังจากหมุนมาได้ช่วงเวลาหนึ่ง และขึ้น Error ในแถบด้านซ้ายในช่องสี่เหลี่ยม ให้ตรวจดูปัญหาที่เกิดขึ้น หากแก้ไขแล้ว ให้กดปุ่ม Ack faults เพื่อรีเซ็ต จะสามารถใช้งานใหม่ได้อีกครั้ง



รูปที่ 1.11 การกำหนดความเร็ว และการสั่งงานมอเตอร์ในโปรแกรม Motor Pilot

5) เมื่อมอเตอร์เริ่มหมุน ในหน้าจอของ Oscilloscope จะสังเกตุสัญญาณวิ่งไปมาอย่างรวดเร็ว ทำให้การสังเกตุเป็นไปได้ยาก ให้กดปุ่ม Start/Stop ที่มุมบนขวาของ Oscilloscope เพื่อหยุดการรับสัญญาณและทำให้สามารถสังเกตุสัญญาณที่จุดเวลานั้นๆได้ง่ายขึ้น

6) ในการเก็บค่าระยะห่างของสัญญาณ Pulse ทำโดยใช้ฟังก์ชัน Curser ของ Oscilloscope วิธีใช้คือ ใช้นิ้วมือแตะหน้าจอ และลากเส้น Curser ไปยังจุดที่สนใจ ในการทดลองนี้ จะลาก Curser1 ไปไว้ที่จุดเริ่มต้นสัญญาณ Pulse หรือจุดขอบข้างของสัญญาณก่อนหน้า และ Curser2 ไปยังจุดขอบข้างของสัญญาณ Pulse ดังภาพที่ *** จะสามารถวัดระยะเวลาของ 1 ลูกคลื่น หรือคาบ ได้จากข้อมูลในช่องสีฟ้า ที่ ΔX และความถี่จาก $1/\Delta X$



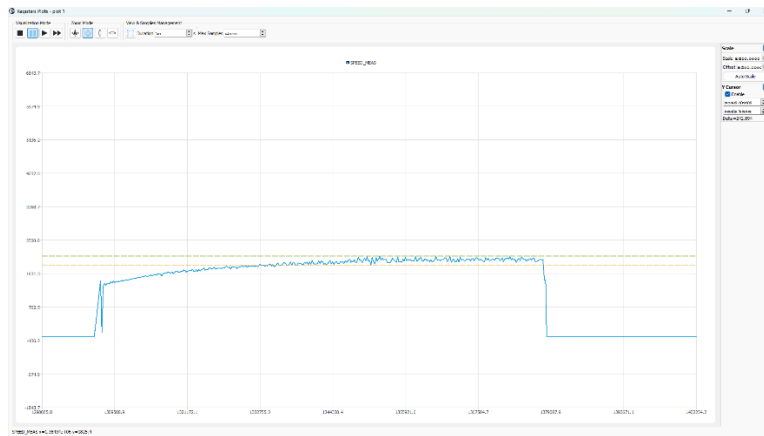
รูปที่ 1.12 รูปแสดงการวาง Curser ของ Oscilloscope ที่ใช้วัดระยะห่างระหว่างคลื่น

7) ข้อควรระวัง การเก็บภาพสัญญาณ Pulse ควรเก็บภาพที่กราฟความเร็วมีความเร็วถึงค่าที่กำหนด หรือใกล้เคียงค่าที่กำหนด

1.9.3 ทำการคำนวณค่า RPM ตามสูตรการคำนวณ

1) ยกตัวอย่างการคำนวณหาค่า RPM ของค่า 1000 RPM

1.1) นำค่า Frequency ของ BackEMF มาทำ Root Mean Square เนื่องจากค่า RPM มีความผันผวนตลอดเวลาดังรูปกราฟนี้



รูปที่ 1.13 รูปแสดงกราฟของค่า RPM จากโปรแกรม

1.2) ทำการคำนวณตามสูตรดังนี้

$$RPM = \frac{126.15 \times 60}{7}$$
$$RPM = 1081.36$$

1.10 ผลการทดลอง

จากการทดลองดังกล่าวเราได้ค่า RPM ของ BLDC Motor ที่ได้จากการคำนวณดังนี้

RPM ที่ตั้งค่าในโปรแกรม	RMS Frequency	RPM ที่ได้จากการคำนวณ
1000	126.15	1081.36
2000	224.06	1920.58
3000	338.21	2898.98
4000	456.61	3913.82
5000	584.82	5012.74
6000	697.40	5977.72
7000	809.78	6941.03
8000	909.97	7799.81
9000	987.36	8463.16
10000	1027.11	8803.82

1.11 สรุปผล

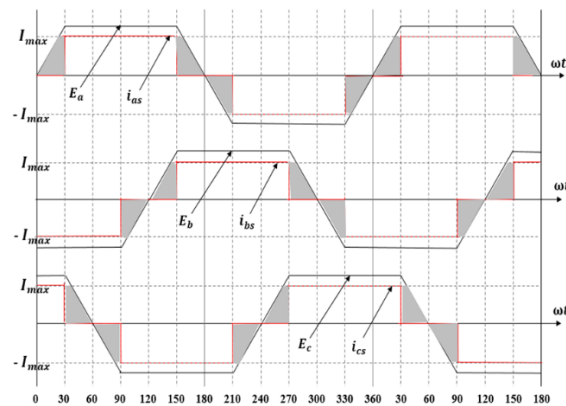
จากการทดลองที่ทำขึ้นพบว่าค่า RPM ของ BLDC motor ที่คำนวณจากสัญญาณ Oscilloscope จะสามารถสอดคล้องกับค่าความเร็วที่ตั้งค่าในโปรแกรม Motor Workbench 6.3.2 โดยทั่วไปแล้วค่าความเร็วที่คำนวณจะมีความคลาดเคลื่อนเล็กน้อย(คลาดเคลื่อน +- ไม่เกิน 10%)ซึ่งอาจเกิดจากปัจจัยต่างๆ เช่น ความแม่นยำของเซ็นเซอร์

1.12 อภิปรายผล

1.12.1 ความแตกต่างระหว่าง Trapezoidal Control แล Field Oriented Control (FOC)

1) Trapezoidal Control

ใช้การควบคุมกระแสแบบ Trapezoidal Waveform และใช้ Hall effect ในการวัดตำแหน่ง ใช้งานได้ง่ายกว่าแบบ FOC แต่ประสิทธิภาพไม่สูงเท่ากับ FOC เพราะการควบคุมกระแสในแบบ Trapezoidal อาจทำให้เกิดการสั่นสะเทือนและการสูญเสียพลังงานที่สูงกว่า



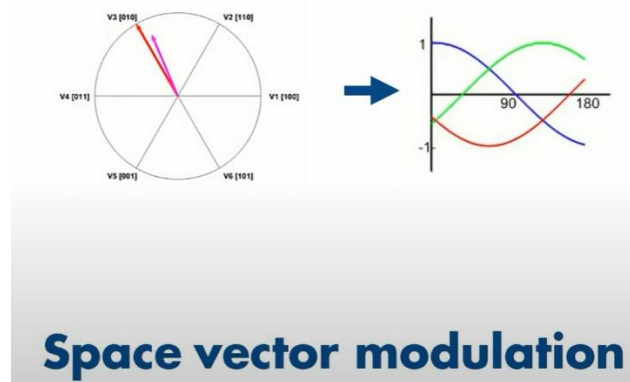
รูปที่ 1.14 รูปแสดง Trapezoidal BackEMF

2) Field Oriented Control (FOC)

ใช้การควบคุมกระแสที่ซับซ้อนกว่าและมุ่งเน้นไปที่การแยกการควบคุมกระแสในแกน D (direct axis) และ Q (quadrature axis) ซึ่งทำให้สามารถควบคุมมอเตอร์ได้เหมือนกับมอเตอร์กระแสสลับ (AC) โดยการคำนวณการมุดตำแหน่งของโรเตอร์ ซับซ้อนกว่า Trapezoidal Control เพราะต้องใช้ในการคำนวณเชิงคณิตศาสตร์และการประมวลผลที่มากขึ้น แต่มีประสิทธิภาพสูงกว่า Trapezoidal Control เพราะสามารถควบคุมกระแสในทิศทางที่เหมาะสมและลดการสูญเสียพลังงาน

1.12.2 ความแตกต่างของการควบคุมระหว่าง 6-Step Control และ Field Oriented Control (FOC)

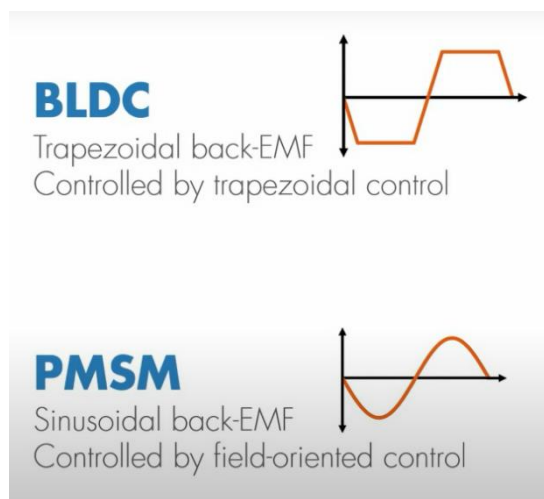
- 1) 6-Step Control ใช้การควบคุมกระแสแบบพัลส์ใน 6 สเต็ป ไม่มีการจัดการกับ Space Vector อย่างเต็มที่ เหมาะสำหรับการใช้งานที่ไม่ต้องการความแม่นยำสูง
- 2) Field Oriented Control (FOC) ใช้การควบคุมที่มีการจัดการกระแสในแกน D-Q และใช้ **Space Vector Modulation (SVM)** เพื่อให้ได้การควบคุมที่ราบรื่นและมีประสิทธิภาพสูงกว่า



รูปที่ 1.15 รูปแสดง การทำงานของ SVM

1.12.3 ความแตกต่างระหว่าง Trapezoidal และ Sinusoidal

- 1) รูปคลื่นกระแสโดย Trapezoidal จะใช้กระแสที่มีรูปคลื่นเป็นทรงสี่เหลี่ยมคางหมู แต่ Sinusoidal จะใช้กระแสที่มีรูปคลื่นเป็นคลื่นไซน์ทำให้มอเตอร์หมุนได้ราบรื่นและมีเสียงรบกวนน้อยกว่าแบบ Trapezoidal
- 2) การควบคุมแบบ Sinusoidal ให้ประสิทธิภาพสูงกว่าการควบคุมแบบ Trapezoidal โดยเฉพาะในงานที่ต้องการการควบคุมที่ละเอียดและมีความเสถียร แต่การควบคุมแบบ Trapezoidal ใช้งานได้ง่ายและมีต้นทุนถูก



รูปที่ 1.16 รูปแสดงความต่างระหว่าง Trapezoidal และ Sinusoidal

1.13 ข้อเสนอแนะ

- 1) การปรับ Sampling Rate ของ Oscilloscope ให้เพียงพอต่อการจับสัญญาณความถี่สูง เพื่อหลีกเลี่ยงการสูญเสียข้อมูลระหว่างการวัด
- 2) ใช้ Signal Filtering ช่วยในการลด noise ของสัญญาณที่เกิดขึ้น
- 3) หากใช้การควบคุมแบบ Trapezoidal หรือ Six-Step Control เพิ่มระบบ Feedback Loop เพื่อเพิ่มความเสถียรในการควบคุมความเร็ว

1.14 เอกสารอ้างอิง

BLDC Motor Control - MATLAB & Simulink

Field-Oriented Control - MATLAB & Simulink

Field-Oriented Control (FOC)

https://youtube.com/playlist?list=PLn8PRpmsu08qL-EG3DRMtRyokpXOJyhp7&si=0Uy1n4s2x_9nvV0o

Direct torque control of brushless DC motor with nonideal trapezoidal back EMF | IEEE Journals & Magazine | IEEE Xplore

<https://www.amazon.in/Brushless-Motor-1000kv-BLDC-Quadcopter/dp/B08DQDSZRD>

<https://www.electricaleasy.com/2015/05/brushless-dc-bldc-motor.html>

https://www.instagram.com/mechatronix_saran/reel/C8SR0xdScwR/

<https://th.mouser.com/applications/dont-ignore-the-brushed-dc-motor/>

2 BrushedDC Motor

2.1 จุดประสงค์

2.1.1 เพื่อให้เข้าใจหลักการทำงานของ Brushed DC Motor และ หา Characteristic ของ Brushed DC Motor เทียบกับแหล่งอ้างอิงที่น่าเชื่อถือได้

2.1.2 เพื่อให้สามารถใช้ Motor Dynamics หา Characteristics ของมอเตอร์ได้

2.1.3 สามารถอธิบาย Motor Constant, Torque Constant และ Back-EMF Constant ของ Brushed DC มอเตอร์ได้

2.1.4 เพื่อให้เข้าใจความสัมพันธ์ของ Speed, Current, Power, Torque และ Efficiency ของ Brushed Motor เมื่อมี Load มากระทำต่อ Motor ได้

2.1.5 สามารถอธิบายความสัมพันธ์ระหว่าง Speed และแรงดันไฟฟ้าได้

2.1.6 เพื่อให้เข้าใจความสัมพันธ์ของ PWM กับ ความถี่, Current , Torque และความเร็วมอเตอร์

2.1.7 เพื่อให้เข้าใจวิธีการควบคุม และใช้งานมอเตอร์โดยใช้ H-bridge Drive รูปแบบต่าง ๆ

2.1.8 เพื่อให้สามารถใช้ Signal Condition กับสัญญาณที่อ่านค่าได้ และนำมาใช้วิเคราะห์ผลการทดลองได้

2.1.9 เพื่อให้สามารถเขียนโปรแกรมโดยใช้ MatLab กับ Microcontroller Nucleo G474RE แสดงผลสิ่งที่สนใจที่เกี่ยวข้องกับมอเตอร์ในรูปแบบกราฟ โดยใช้สัญญาณจากเซนเซอร์ที่มีอยู่เป็น Input ได้

2.2 สมมุติฐาน

2.2.1 เราตั้งสมมุติฐานว่า หากให้ PWM กับ ความถี่ของ PWM ต่างกัน จะส่งผลต่อ Characteristics ของมอเตอร์ เช่น ความเร็วการหมุน แรงบิดที่มอเตอร์สามารถทำได้ และกระแสที่มอเตอร์ใช้

2.3 ตัวแปร

2.3.1 ตัวแปรต้น : ความถี่ของสัญญาณ PWM ที่จ่าย, Duty cycle ของ PWM

2.3.2 ตัวแปรตาม : ความเร็วเชิงมุมของมอเตอร์, กระแสไฟฟ้าที่มอเตอร์ใช้, แรงดันที่จ่ายให้มอเตอร์ และพลังงานที่มอเตอร์ใช้

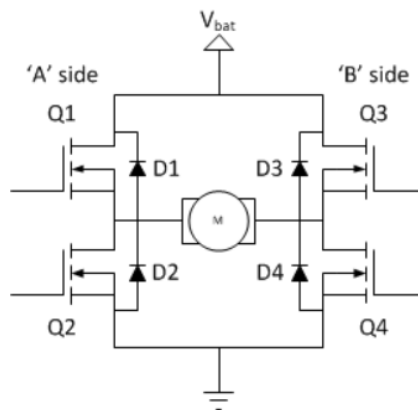
2.3.3 ตัวแปรควบคุม : แรงบิด, ระยะเวลาในการเก็บข้อมูล, โปรแกรมที่ใช้เก็บข้อมูล

2.4 นิยามศัพท์เฉพาะ

2.4.1 Brushed DC Motor ในรายงานต่อไปนี้จะขอเรียกว่า มอเตอร์ คืออุปกรณ์แปลงพลังงานไฟฟ้าเป็นพลังงานกลเชิงมุม ชนิดหนึ่ง ในการทดลองนี้มีอยู่ 2 ตัว คือมอเตอร์ที่เราสนใจ และมอเตอร์ที่ใช้จำลอง Load โดยจะขอเรียกแทนมอเตอร์แต่ละตัวว่า มอเตอร์ตัวที่ 1 และมอเตอร์ตัวที่ 2 ตามลำดับ

2.4.2 H-bridge Drive หลังจากนี้จะขอเรียกว่า การไคร์ฟ ตามด้วยชื่อรูปแบบ คือการต่อวงจรไฟฟ้ารูปแบบหนึ่งเพื่อควบคุมทิศทางการหมุนของมอเตอร์ มีลักษณะการต่อเป็นรูปตัว H และมีมอเตอร์ตรงกลาง จึงเรียกกันว่า H-bridge Drive มีรูปแบบการใช้งานอยู่ 3 รูปแบบ คือ

- 1) Sign Magnitude
- 2) Lock Anti Phase
- 3) Asynchronous Sign-Magnitude



ภาพที่ 2.1 ตัวอย่าง Schematics ของวงจร H-Bridge Circuit

ที่มา : https://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridge_drivers/

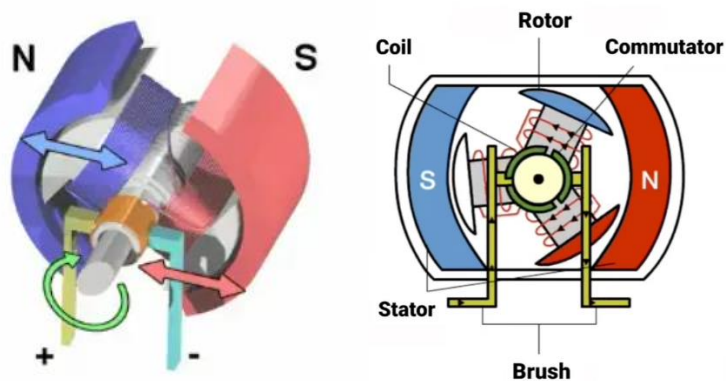
2.4.3 PWM หรือ Pulse width modulation คือ วิธีการจ่ายแรงดันไฟฟ้ารูปแบบ Square wave เพื่อควบคุมพลังงานหรือแรงดันที่จ่ายให้กับ Load ทางไฟฟ้า แทนการใช้ Resistor เพื่อลดการสูญเสียพลังงาน

2.5 นิยามเชิงปฏิบัติการ

- 1.

2.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.6.1 Brushed DC Motor คือมอเตอร์ที่ใช้แผ่นเหล็ก หรือแท่งเหล็กที่เรียกว่า Brushed เป็นตัวจ่ายกระแสเข้าไปยังขดลวด และยังเป็นตัวสลับเปลี่ยนขดลวดเพื่อให้มอเตอร์หมุนได้อย่างต่อเนื่อง หลักการหมุนก็คือ เมื่อต่อวงจรไฟฟ้าครบวงจรทั้ง 2 สายของมอเตอร์แล้ว จะเกิดสนามแม่เหล็กที่ขดลวด ซึ่งจะออกแรงผลัก และดูดเข้าหาแม่เหล็กถาวรที่ติดอยู่ด้านข้าง ทำให้มอเตอร์เริ่มขยับ ในระหว่างหมุนนั่นเอง ก็เป็นการเปลี่ยนหน้าสัมผัสของแปรงถ่านกับขดลวดไปในตัว ดังรูป และทำให้ขดลวดมีแรงดันไฟฟ้าเปลี่ยนแปลงไปเป็นลำดับกัน และหมุนได้ต่อเนื่อง



รูปที่ 2.2 รูปแสดงโครงสร้างภายในของ Brushed

2.6.2 Data Sheet ของมอเตอร์ทั้งสองตัวบนชุดการทดลอง

เนื่องจากมอเตอร์ที่ได้รับมานั้น ไม่ได้แนบ Datasheet มาให้ด้วย จึงต้องค้นหา Datasheet ของมอเตอร์ทั้งสองตัวที่ใช้งาน

1) ในส่วนของมอเตอร์ที่เราสนใจนั้น ไม่สามารถหา Datasheet ได้ ดังนั้นก่อนการทดลองจึงจำเป็นต้องหา Motor Characteristics ก่อนทำการทดลอง

2) Datasheet ของมอเตอร์ที่ใช้จำลอง Load ที่เกิดกับมอเตอร์ เราได้ดูที่บริเวณโดยรอบของมอเตอร์ พบว่ารุ่นของมอเตอร์ คือ ZGA60FM 1200 RPM แต่ก็ยังไม่สามารถหา Datasheet ได้ เราจึงหา Datasheet ของมอเตอร์ที่มีลักษณะคล้ายกันมาแทน จึงได้ Datasheet ของรุ่น ZGA37FH แทน

ZGA37FH(A6249)

Housing Material	Zinc-Sleeve
Bearing	Sleeve Bearing
Radial Force	≤ 1 kgf
Axial Force	≤ 0.7 kgf
Max Axial Pressing Force	≤ 5 kgf
Radial Play	≤ 0.1 mm
Axial Play	≤ 0.5 mm



รูปที่ 2.3 รูปแสดง Datasheet ของ ZGA37FH



รูปที่ 2.4 รูปแสดงรุ่นของมอเตอร์ที่ใช้ในการทดลอง

3) Motor constant คือ ตัวแปรที่บ่งบอกถึงประสิทธิภาพในการแปลงพลังงานไฟฟ้าเป็นพลังงานกล มักจะเขียนโดยย่อว่า K_m หรือ Motor constant ซึ่งมีความสัมพันธ์กับ Torque และ Power หรือกำลังทางไฟฟ้า โดยสามารถเขียนในรูปสมการได้

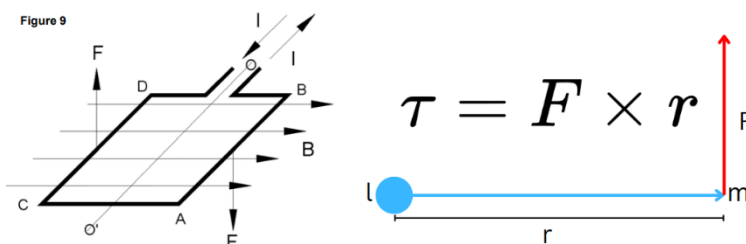
$$K_m = \frac{\tau}{\sqrt{P_{loss}}} \quad (1)$$

K_m คือ Motor constant หรือค่าคงที่มอเตอร์

τ คือ Torque สูงสุดที่มอเตอร์สามารถทำได้

P_{loss} คือ กำลังไฟฟ้าที่ใช้ไป

4) Torque หรือแรงบิด ในทางฟิสิกส์ แรงบิด คือหน่วยวัดของแรงที่ทำให้วัตถุหมุนรอบตัวเอง ในการทำงานของมอเตอร์หมายถึง แรงที่ Rotor ออกแรงผลักแม่เหล็กถาวร หรือออกแรงดูดเข้าหาแม่เหล็กถาวร เพื่อให้เกิดการหมุน โดยลักษณะของแรง และทิศทางจะเป็นไปตามภาพที่ *** จะแทนเส้นสีดำที่บ่งเป็นขดลวดที่เราจ่ายกระแส หรือแรงดันเข้าไป จะทำให้เกิดแรง F ที่ขดลวด และหมุนรอบแกน



รูปที่ 2.5 รูปแสดงทิศทางของแรงที่เกิดขึ้นในมอเตอร์(ซ้าย) และตัวอย่างการเขียน Freebody diagram เพื่อการคำนวณ(ขวา) สมการที่ใช้คำนวณแรงบิด จากภาพไดอะแกรมข้างต้นสามารถเขียนได้ว่า

$$\tau = F \times r \quad (2)$$

τ	คือ แรงบิด หรือแรงที่ทำให้วัตถุหมุนรอบตัวเอง มีหน่วยเป็น Nm (นิวตันเมตร)
F	คือ แรงที่กระทำที่วัตถุตั้งฉากกับระยะห่าง มีหน่วยเป็น N (นิวตัน)
r	คือ ระยะรัศมีจากจุดหมุน ถึงจุดที่มีแรงมากระทำ มีหน่วยเป็น m (เมตร)

นอกจากนี้ ในการทำงานของมอเตอร์ แรงที่ใช้คือแรงที่เกิดจากสนามแม่เหล็ก ซึ่งมีความสัมพันธ์กับกระแสไฟฟ้า จึงทำให้ได้สมการใหม่อีกสมการ คือ

$$\tau = K_t \cdot I \quad (1)$$

τ	คือ แรงบิด หรือแรงที่ทำให้วัตถุหมุนรอบตัวเอง มีหน่วยเป็น Nm (นิวตันเมตร)
--------	--

K_t	คือ ค่าคงที่แรงบิด หรือ Torque constant ที่จะบอกประสิทธิภาพในการแปลงกระแสไฟฟ้าเป็นแรงบิด มักใช้หน่วย Nm/A (นิวตันเมตร/แอมป์)
I	คือ กระแสไฟฟ้าที่จ่ายให้ขดลวดเพื่อแปลงให้เป็นแรงแม่เหล็กไฟฟ้า มีหน่วยเป็น A (แอมแปร์)

4.1) Torque Constant คือ ตัวแปรที่บ่งบอกถึง ประสิทธิภาพในการแปลงกระแสไฟฟ้าเป็นแรงบิดของมอเตอร์ โดยสามารถใช้ สมการในการคำนวณได้ ดังนี้

	$K_t = \frac{\tau}{I}$	(1)
K_t	คือ ค่าคงที่แรงบิด หรือ Torque constant ที่จะบอกประสิทธิภาพในการแปลงกระแสไฟฟ้าเป็นแรงบิด มักใช้หน่วย Nm/A (นิวตันเมตร/แอมป์)	
τ	คือ แรงบิด หรือแรงที่ทำให้วัตถุหมุนรอบตัวเอง มีหน่วยเป็น Nm (นิวตันเมตร)	
I	คือ กระแสไฟฟ้าที่จ่ายให้ขดลวดเพื่อแปลงให้เป็นแรงแม่เหล็กไฟฟ้า มีหน่วยเป็น A (แอมแปร์)	

5) Back EMF หรือ Back electromotive force คือ แรงดันย้อนกลับ จากหัวข้อ 2.1 เมื่อจ่ายกระแสไฟฟ้าให้มอเตอร์ จะทำให้ขดลวดมีสถานะเป็นแม่เหล็กไฟฟ้าชั่วคราว กลับกัน หากไม่ได้จ่ายกระแสไฟฟ้าเข้าไปยังขดลวด เมื่อมีแม่เหล็กเคลื่อนที่ผ่านขดลวดจะทำให้เกิดแรงดันไฟฟ้าย้อนกลับไปยังขั้วของมอเตอร์ ยิ่งผ่านไปเร็วเท่าไร ยิ่งมีแรงดันย้อนกลับมากเท่านั้น การคำนวณ Back EMF จึงมีความสัมพันธ์กับความเร็วการหมุนของมอเตอร์ตามสมการด้านล่าง

	$emf = K_e \cdot \omega$	(1)
emf	คือ ค่าของแรงดันไฟฟ้าย้อนกลับ มีหน่วยเป็น V (โวลต์)	
K_e	คือ ค่าคงที่แรงดันย้อนกลับ มีหน่วยเป็น V/(rad/s) (โวลต์ต่อเรเดียนต่อวินาที)	
ω	คือ ความเร็วในการหมุนของมอเตอร์ มีหน่วยเป็น rad/s (เรเดียนต่อวินาที)	

ในการคำนวณ Back EMF เราต้องใช้ค่า K_e หรือ Electromotive Force Constant

มาช่วยคำนวณดังสมการด้านล่าง

$$L \frac{dI}{dt} + RI + emf = V \quad (1)$$

- L** คือ ค่าของขดลวดเหนี่ยวนำ ซึ่งแทนขดลวดในมอเตอร์ มีหน่วยเป็น H (เฮนรี)
- R** คือ ค่าต้านทานของมอเตอร์ สามารถวัดได้โดยใช้ มัลติมิเตอร์วัดที่ขั้วของมอเตอร์ มีหน่วยเป็น Ω (โอห์ม)
- I** คือ กระแสที่ไหลผ่านขดลวดของมอเตอร์ มีหน่วยเป็น A (แอมป์)
- emf** คือ ค่าของแรงดันไฟฟ้าย้อนกลับ มีหน่วยเป็น V (โวลต์)
- V** คือ ค่าของแรงดันไฟฟ้าที่ตกคร่อมขั้วทั้งสองของมอเตอร์ มีหน่วยเป็น V (โวลต์)
- t** คือ เวลา มีหน่วยเป็น s (วินาที)

เมื่อนำ emf ในสมการ มาแทนในสมการและเมื่อกระแสไฟฟ้าไหลคงที่ จะไม่ทำให้เกิดแรงดันที่ขดลวดเหนี่ยวนำ

จึงทำให้ $L \frac{dI}{dt} = 0$ จัดรูปสมการเพื่อหา K_e จะได้สมการใหม่

$K_e = \frac{V - RI}{\omega}$ (1)	
K_e	คือ ค่าคงที่แรงดันย้อนกลับ มีหน่วยเป็น V/(rad/s) (โวลต์ต่อเรเดียนต่อวินาที)
R	คือ ค่าต้านทานของมอเตอร์ สามารถวัดได้โดยใช้ มัลติมิเตอร์วัดที่ขั้วของมอเตอร์ มีหน่วยเป็น Ω (โอห์ม)
I	คือ กระแสที่ไหลผ่านขดลวดของมอเตอร์ มีหน่วยเป็น A (แอมป์)
V	คือ ค่าของแรงดันไฟฟ้าที่ตกคร่อมขั้วทั้งสองของมอเตอร์ มีหน่วยเป็น V (โวลต์)
ω	คือ ความเร็วในการหมุนของมอเตอร์ มีหน่วยเป็น rad/s (เรเดียนต่อวินาที)

6) PWM หรือ Pulse Width Modulation คือ รูปแบบการจ่ายพลังงานที่ให้แก่โหลดทางไฟฟ้า โดยจ่ายแรงดันเป็นสัญญาณ Square wave ที่มีระยะเวลาการเปิดต่างกัน โดยใน 1 ลูกคลื่น จะมีทั้งช่วงปิด และช่วงเปิด เรานับช่วงเวลาที่สัญญาณเปิดใน 1 ลูกคลื่นนั้นว่า Duty cycle (%) ซึ่งจะบ่งบอกถึงเปอร์เซ็นต์การให้พลังงานแก่โหลดทางไฟฟ้า โดยวิธีนี้สามารถควบคุมพลังงานให้แก่โหลดได้โดยสูญเสียพลังงานน้อยที่สุด

7) Motor Characteristic คือ กราฟที่บ่งบอกถึง ค่าที่มีความสัมพันธ์กันในด้านต่าง ๆ ของมอเตอร์ เช่น ความเร็วกับแรงบิด ความเร็วกับกระแสไฟฟ้า ประสิทธิภาพกับแรงบิด โดยกราฟนี้สามารถใช้สมการคำนวณออกมาได้โดยใช้ทฤษฎีและกฎทางฟิสิกส์ เช่น Faraday's Law ที่กล่าวถึง แม่เหล็กไฟฟ้า และฟลักซ์แม่เหล็ก ซึ่งเป็นส่วนสำคัญของมอเตอร์, Lorentz Force ที่กล่าวถึง แรงที่เกิดจากสนามแม่เหล็ก รวมถึงแม่เหล็กไฟฟ้า Energy loss หรือการสูญเสียพลังงาน คือ

7.1) การสูญเสียพลังงานในระหว่างการทำงานของมอเตอร์ ไม่ว่าจะเป็นแรงเสียดทานของเพลารotor การสูญเสียพลังงานไฟฟ้าในรูปแบบความร้อน และเสียง โดยเราสามารถอธิบายได้ตามกฎการอนุรักษ์พลังงาน คือ พลังงานที่ถูกใส่เข้าไปในระบบใดระบบหนึ่ง จะเท่ากับพลังงานที่ส่งออกมา หรือเขียนเป็นสมการได้ว่า

$$E_{in} = E_{out} \text{ โดยที่ } E \text{ คือ พลังงานในรูปแบบต่าง ๆ}$$

8) Motor Drive ชนิดต่าง ๆ และ Datasheet ของมอเตอร์ไดรฟ์ ในการทดลองนี้เราได้รับ มอเตอร์ไดรฟ์เป็น Cytron MDD20A จำนวน 2 อัน เพื่อใช้ขับมอเตอร์ทั้ง 2 ตัว



รูปที่ 2.6 รูปแสดงมอเตอร์ไดรฟ์เวอร์ Cytron รุ่น MDD20A(ซ้าย) และภาพ Datasheet ของมอเตอร์ไดรฟ์(ขวา)

ภาพจาก <https://makermotor.com/pn00218-cyt7-cytron-20a-6-30v-dc-motor-driver-md20a/>

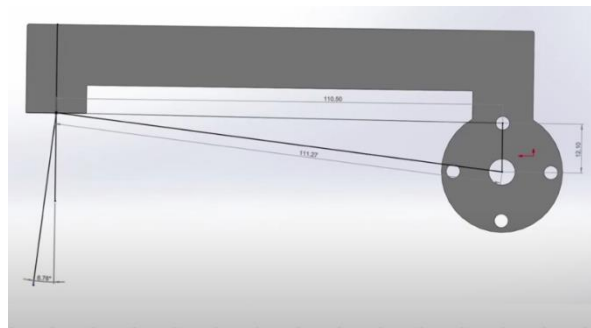
2.7 วิธีดำเนินการทดลอง

1. เริ่มจากดาวน์โหลดโปรแกรมที่สำคัญกับการทดลอง ติดตั้งและเชื่อมต่อสายสัญญาณต่าง ๆ ในบอร์ดการทดลองให้เรียบร้อย
2. เปิดโปรแกรมที่ได้รับจาก TA และทดสอบการใช้งานก่อนทดลองจริง
3. เขียนโปรแกรมที่สามารถเก็บค่าตัวแปรเมื่อเริ่มการทดลอง ให้เหมาะสมกับการทดลอง
4. เริ่มการทดลอง โดยอ่านค่าความเร็วจากโปรแกรม และเก็บค่าลงใน Matlab เพื่อนำไปวิเคราะห์

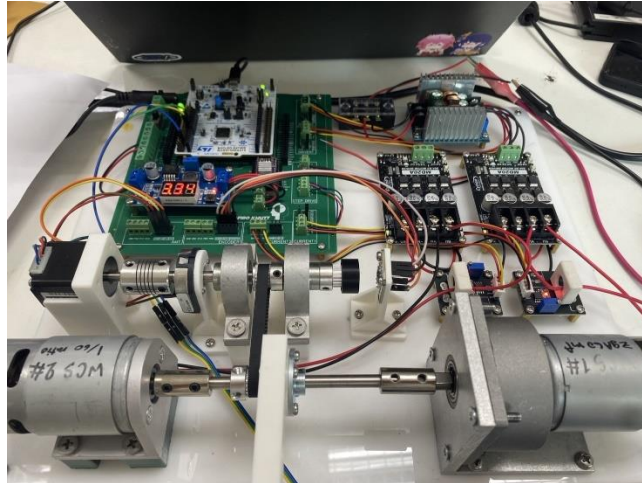
2.8 วัสดุอุปกรณ์

2.8.1 ชุดการทดลอง Motor Exploror 1 ชุด โดยภายในจะประกอบด้วยอุปกรณ์สำคัญดังนี้

- 1) มอเตอร์ DC 12 V 1 ตัว โดยจะขอเรียกว่า มอเตอร์ตัวที่ 2
- 2) มอเตอร์เกียร์ DC 12 V 1 ตัว โดยจะขอเรียกว่า มอเตอร์ตัวที่ 1
- 3) มอเตอร์ไทร์เวอร์ Cytron MDD20A 2 ตัว
- 4) HAL current sensor wcs1700 2 ตัว
- 5) Incremental Encoder AMT103-V 1 ตัว
- 6) ชิ้นส่วน 3D Print สำหรับช่วยวัดแรงบิด



รูปที่ 2.7 รูปแสดงขนาดของชิ้นส่วนสำหรับช่วยวัดแรงบิด



รูปที่ 2.8 ภาพชุดการทดลอง MotorXploer พร้อมบอร์ด Nucleo G474RE

- 7) ชุดการทดลอง Loadcell Explorer 1 ชุด
- 8) Power *Adaptor* 220 V to 24 V 1 ตัว
- 9) Power supply ที่สามารถจ่ายกระแสได้มากกว่า 10 A 1 ตัว
- 10) โปรแกรม Matlab และ Simulink
- 11) ถ่วงทราย 500 กรัม 2 ถัง

2.9 ขั้นตอนการดำเนินงาน

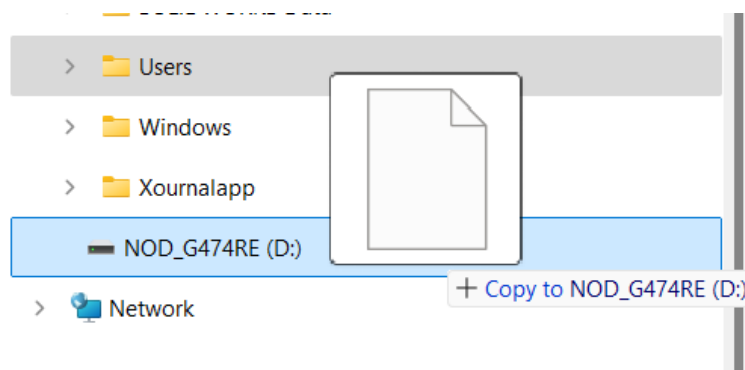
2.9.1 ดาวน์โหลดโปรแกรม Matlab และดาวน์โหลดไฟล์ควบคุมมอเตอร์ที่ได้รับจาก TA

- 1) เมื่อดาวน์โหลดแล้ว จะได้รับไฟล์ MotorXploer, RMX_Motor.bin, RMX_Motor.hex และ Waijang 1 Download Link
- 2) เปิดไฟล์ Waijang 1 Download Link จะขึ้นเป็นการโหลดไฟล์ Waijang จากนั้นแตกไฟล์ .zip ไว้ใน โฟลเดอร์เดียวกับไฟล์ MotorXploer เมื่อแตกไฟล์จะพบไฟล์ Install_waijang เป็นไฟล์ Matlab code
- 3) เปิดโปรแกรม Matlab และกด Run ไฟล์ Install_waijang รอดติดตั้งจนโปรแกรมขึ้นข้อความว่า “Finish Waijung Installation.” ถือว่าการติดตั้งเสร็จเรียบร้อยแล้ว
- 4) ปิดและเปิดโปรแกรมใหม่อีกครั้ง

Name	Date modified	Type	Size
MatlabDataDCMotor	18/12/2567 11:34	File folder	
slprj	18/12/2567 4:56	File folder	
waijung_18.11a	17/12/2567 11:58	File folder	
Fortest_Motorlab	18/12/2567 7:56	Simulink Model	150 KB
MotorXploer	18/12/2567 16:31	Simulink Model	230 KB
RMX_Motor.bin	17/12/2567 9:46	BIN File	44 KB
RMX_Motor.hex	17/12/2567 9:46	HEX File	121 KB
Waijang 1 Download Link	17/12/2567 9:46	Internet Shortcut	1 KB
MotorXploer.slx.autosave	18/12/2567 16:36	AUTOSAVE File	202 KB

รูปที่ 2.9 รูปแสดงไฟล์ทั้งหมดที่ได้รับมาจาก TA

2.9.2 ทดลองใช้งานก่อนทดลอง เมื่อเชื่อมต่อบอร์ดควบคุม กับต่อไฟเลี้ยง 24 V และเปิดสวิตช์เรียบร้อยแล้ว ให้นำไฟล์ RMX_Motor.bin ลากมาใส่ยังไดรฟ์ของ NOD_G474RE (D:) ที่ปรากฏด้านซ้ายมือของ Flie Exploror และกดปุ่ม Reset ที่บอร์ดควบคุมอีกครั้ง หรือกดทุกครั้งเมื่อสงสัยว่ามอเตอร์มีการทำงานผิดปกติ



รูปที่ 2.10 รูปแสดงการลากไฟล์ RMX_Motor.bin มาใส่ยัง NOD_G474RE (D:)

2.9.3 จากนั้นเปิดไฟล์ Simulink ชื่อว่า MotorExploer จะเจอโปรแกรมการควบคุมมอเตอร์พื้นฐานที่ TA ได้จัดทำไว้ โดยจะมีทั้ง ค่าที่อ่านได้จากเซนเซอร์วัดกระแส, Encoder วัดความเร็วเชิงมุม, พารามิเตอร์ที่ใช้ควบคุมมอเตอร์ โดยเราจะขออธิบายทีละส่วน เริ่มจากส่วนค่าที่อ่านได้จากเซนเซอร์ ในการทดลองนี้ เราจะสนใจแค่ตัวแปรที่อยู่ในกรอบสีแดงเท่านั้น

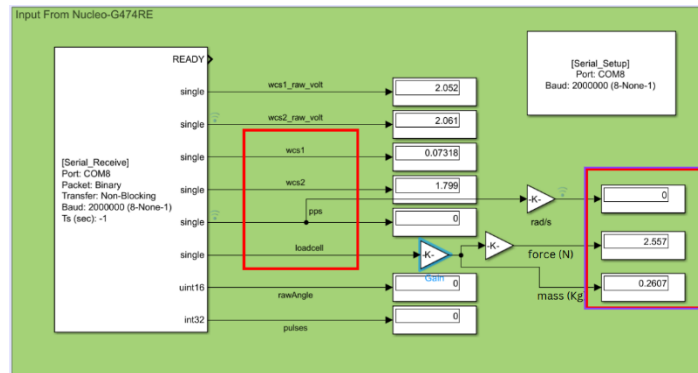
- 1) Wcs1 และ wcs2 คือ ค่าที่อ่านได้จาก Hal current sensor หรือก็คือค่ากระแสที่ไหลผ่านสายไฟไปยังมอเตอร์
- 2) pps หรือ Pulse per second คือ ค่าสัญญาณ Pulse ต่อวินาที จาก Encoder เพื่อนำมาคำนวณความเร็วที่ได้จากการหมุนของมอเตอร์ มีหน่วยเป็น rad/s
- 3) Loadcell คือ ค่าดิจิตอลที่ได้จาก ชุดการทดลอง Loadcell Exploror ในการใช้งานจริงต้องนำมาคำนวณหาค่าที่ถูกต้อง โดยการนำถ่วงหยาบมาวางไว้บน Loadcell และเอาค่าที่อ่านได้มาเปรียบเทียบกับน้ำหนักถ่วงหยาบ และใช้บัญญัติไตรยางศ์ในการคำนวณ โดยมีแนวคิดที่ว่า อัตราส่วนของเลขบิตที่อ่านได้ กับเลขบิตสูงสุด กับอัตราส่วนของ น้ำหนัก กับน้ำหนักสูงสุดที่สามารถอ่านได้ คือ 10 กิโลกรัม สามารถเขียนเป็นสมการได้ คือ

	$\frac{Bit}{4096} = \frac{m_{out}}{10}$	(1)
	$\frac{Bit}{4096} \cdot 10 = m_{out}$	(2)
Bit	คือ ค่าดิจิตอลที่ Loadcell อ่านได้ ซึ่งเป็น 12 บิต หรือตั้งแต่ 0 ถึง 4095	
m_{out}	คือ ค่าของน้ำหนักที่ผ่านการคำนวณแปลงจาก ดิจิตอลเป็น แอนาล็อกแล้ว มีหน่วยเป็น กิโลกรัม	

และหากต้องการให้อ่านค่าเป็น N (นิวตัน) ให้คูณ 9.81 เข้าไปในสมการข้างต้น

จะทำให้ได้แรงที่เป็นไปตามสมการ $F = ma$ ซึ่งมีหน่วยเป็น N (นิวตัน)

หากลองเปรียบเทียบแล้ว ค่า m_{out} ยังไม่ได้น้ำหนักที่ถูกต้องตามน้ำหนักถูกราย หรือค่ามีการ offset ขึ้นไป ให้ใช้บล็อก minus ใน Simulink ลบค่าออกเพื่อให้ค่าที่ได้ไม่ offset ขึ้น



รูปที่ 2.11 รูปแสดงบล็อกโปรแกรมที่ TA จัดทำให้ ในส่วนค่าที่อ่านได้จากเซนเซอร์

2.9.4 ในส่วนที่สอง คือพารามิเตอร์ที่ใช้ในการควบคุมมอเตอร์ โดยเราจะสนใจพารามิเตอร์ที่อยู่ในกรอบสีแดงเท่านั้น

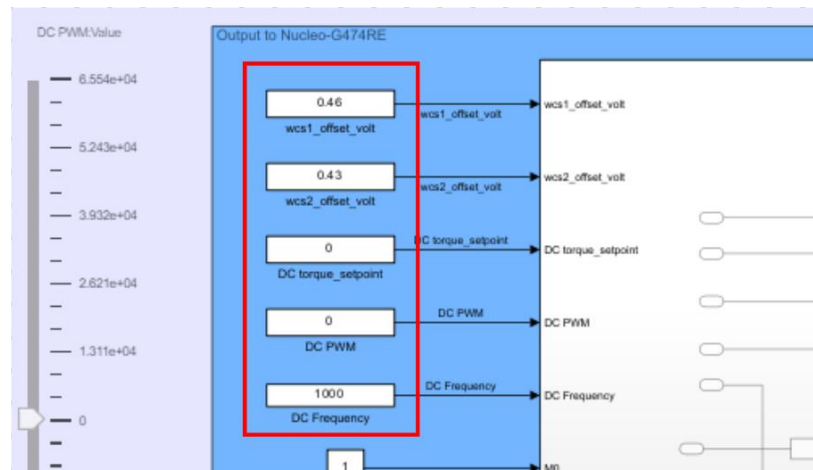
1) Wcs1_offset_volt และ wcs2_offset_volt คือ พารามิเตอร์ที่ช่วยปรับค่าที่อ่านได้จาก Hal current sensor ให้ไม่มีการ offset ขึ้นไป ทำได้โดย หยุดการทำงานของมอเตอร์ หรือมอเตอร์ได้รฟ์ เมื่อหยุด กระแสที่ควรอ่านได้ จะเท่ากับ 0 แต่ถ้าหากยังไม่เป็น 0 ปรับค่าในพารามิเตอร์นี้ เพื่อให้ค่าไม่ offset

2) DC torque_setpoint คือ การตั้งค่าทอร์คของมอเตอร์ตัวที่ 1 หรือมอเตอร์เกียร์ เพื่อใช้จำลอง Load ที่มากระทำกับมอเตอร์ตัวที่ 2 มีหน่วยเป็น Nm (นิวตันเมตร)

3) DC PWM คือ พารามิเตอร์ที่บ่งบอกแทนการใช้ Duty cycle ของสัญญาณ PWM โดยมีค่าตั้งแต่ -65536 ไปจนถึง 65535 ซึ่งเป็นค่า 16 บิต และ เครื่องหมาย - (ลบ) สื่อถึงการหมุนในทิศทางตรงกันข้าม หากจะนำมาคำนวณย้อนกลับ เป็น Duty cycle ให้ใช้การเทียบบัญญัติไตรยางค์ โดยไม่สนใจเครื่องหมาย - (ลบ) ด้านหน้า จะได้สมการคือ

	$Duty(\%) = \frac{bit}{65536} \cdot 100$	(1)
Duty(%)	คือ เปอร์เซนต์ของช่วงเวลาที่สัญญาณมีค่าเป็น 1 โดยคิดจาก 1 ลูกคลื่น	
bit	คือ จำนวนบิตที่เราใส่ไปในพารามิเตอร์ DC PWM	

4) DC Frequency คือ ความถี่ของสัญญาณ PWM หรือจำนวนสัญญาณ PWM ที่จ่ายเข้าไปยังมอเตอร์ภายใน 1 วินาที มีหน่วยเป็น Hz (เฮิรตซ์)



รูปที่ 2.12 รูปแสดงบล็อกโปรแกรมที่ TA จัดทำให้ ในส่วนพารามิเตอร์ปรับได้ที่เกี่ยวข้องกับมอเตอร์

2.9.5 เมื่อทำทุกอย่างตามขั้นตอนข้างต้นเรียบร้อยแล้ว สามารถทดสอบการหมุนของมอเตอร์ได้โดยกดที่ปุ่ม test Buttons เพื่อให้มอเตอร์หมุน หรือทดลองเปลี่ยนค่า DC PWM หรือ DC Frequency เพื่อให้มอเตอร์หมุนได้

2.9.6 ในการทดสอบการหมุนนี้ พบว่าในระบบของมอเตอร์เกิดการผิดจากการใช้ Pulley ที่เชื่อมต่อกับ Encoder และ Missalignment ที่อาจเกิดจากการขึ้นรูปชุดการทดลอง เราจึงกังวลว่าอาจส่งผลกระทบอย่างมากต่อผลการทดลองที่ได้ต่อจากนี้ เราจึงได้ออกแบบวิธีการใช้ชุดการทดลองใหม่เพื่อลดความผิดที่เกิดขึ้น โดยลดการเชื่อมต่อทางกลทั้งหมด มาต่อโดยตรงกับ เซนเซอร์ และใช้สเปรย์ช่วยลดความผิด เช่น WD-40

1) ในการทดสอบความเร็วการหมุนของมอเตอร์ เราได้ลดการใช้ Pulley โดยการย้ายมอเตอร์มาไว้ยังจุดที่ติด Stepper Motor เพราะสามารถเชื่อมต่อเพลามอเตอร์ กับ Encoder ได้โดยตรง และใช้ Flexible Coupling เพื่อลด Missalignment ระหว่างมอเตอร์กับเพลา Encoder



รูปที่ 2.13 รูปแสดงการแก้ไข ลดความผิดของชุดการทดลอง โดยใช้ลวดยึดมอเตอร์

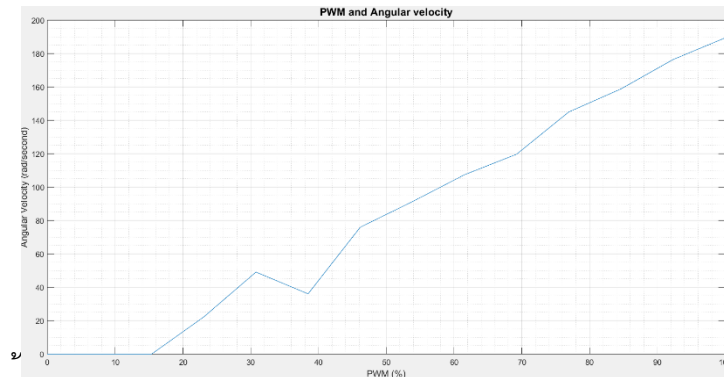
2.9.7 ในการเก็บค่าพารามิเตอร์ต่าง ๆ ที่เซนเซอร์อ่านได้ เราได้ใช้ Simulink ในการเก็บค่าโดยใช้บล็อก To workspace เพื่อส่งค่าออกไปยัง Workspace และบันทึกค่าเป็นไฟล์นามสกุล .mat เพื่อนำไปพิจารณาต่อไป ทั้งนี้ ในส่วนการเก็บค่าความเร็วที่เปลี่ยนไปตาม PWM และความถี่ เราอยากเห็นภาพรวมของการเปลี่ยนแปลง หรือก็คือ เราอยากให้ค่าพารามิเตอร์เปลี่ยนไปในขณะที่มอเตอร์กำลังทำงาน โดยให้ระยะเวลาในการเปลี่ยนค่าพารามิเตอร์มีค่าเท่ากัน เราจึงได้ใช้ Simulink ในการสร้างระบบที่สามารถเพิ่มค่าขึ้นแบบ Step ได้ และยังสามารถกำหนดระยะเวลาในการเพิ่มของ Step ได้อีกด้วย

ภาพที่ 2.14 ภาพ Simulink ระบบที่เราใช้เพิ่มค่า และเก็บค่าที่อ่านได้จากเซนเซอร์ในข้อที่ 3.1, 3.2, 3.3

เมื่อทดลองและเก็บผลสำเร็จแล้ว เราได้จัดทำกราฟความเร็วของ มอเตอร์เทียบกับ ค่าพารามิเตอร์ของ DC PWM เพื่อพิจารณาว่ามีการเปลี่ยนแปลงไปอย่างไร โดยให้ DC PWM เพิ่มขึ้นครั้งละ 5000 หน่วย ทำให้ได้กราฟตามภาพที่ ***

ภาพที่ 2.15 ภาพกราฟความเร็ว เทียบกับ DC PWM ที่เปลี่ยนแปลงไป

แต่ PWM นั้นมีหน่วยเป็น % เราจึงนำค่าทั้งหมดมาแปลง โดยใช้วิธีการคูณค่าพารามิเตอร์ DC PWM ที่บันทึกไว้ด้วย $\frac{100}{65535}$ ซึ่งเป็นวิธีการแปลงจากเลขบิต เป็นเปอร์เซ็นต์ของสัญญาณ PWM จัดทำกราฟใหม่จะได้



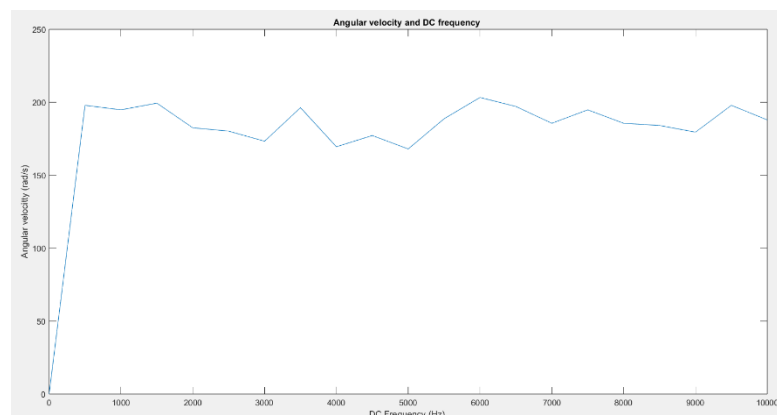
ภาพที่ 2.16 ภาพกราฟความเร็ว เทียบกับ DC PWM ที่เปลี่ยนแปลงไป ที่แปลงหน่วยของ PWM ให้เป็น % แล้ว

นอกจากนี้ เรายังได้สนใจ Current หรือกระแสที่ไหล ต่อความเร็วการหมุนของมอเตอร์อีกด้วย จึงได้นำค่าทั้งสองมาพล็อตเป็นกราฟ ได้กราฟดังรูปที่ ***



ภาพที่ 2.17 ภาพกราฟความสัมพันธ์ระหว่างความเร็วกับกระแสที่ใช้

ในขณะเดียวกัน เราได้ทดลองเปลี่ยน พารามิเตอร์ DC frequency ไปเรื่อยๆเป็น Step เช่นเดียวกับ DC PWM โดยเพิ่มครั้งละ 500 ไปจนถึง 10000 และเปรียบเทียบความเร็วที่ได้ ทำให้ได้ดังกราฟ



ภาพที่ 2.18 ภาพกราฟความสัมพันธ์ระหว่างความถี่ของสัญญาณ PWM กับ ความเร็วการหมุนของมอเตอร์

2.11 สรุปผล

จากการทดลองที่ได้ทำมาปรากฏว่า Duty cycle ของสัญญาณ PWM มีผลต่อความเร็วของมอเตอร์ แต่การปรับความถี่ของสัญญาณ PWM ไม่มีผลต่อความเร็วของมอเตอร์

2.12 อภิปรายผล

2.13 ข้อเสนอแนะ

2.14 เอกสารอ้างอิง

3 Stepper Motor

3.1 จุดประสงค์

3.1.1 เพื่อสามารถอธิบายหลักการทำงานของ Stepper Motor ได้

3.1.2 เพื่อสามารถอธิบายความสัมพันธ์ของ Speed เมื่อ Frequency ของสัญญาณที่จ่ายเข้า Stepper Motor เปลี่ยนแปลงไป

3.1.3 เพื่อสามารถอธิบายหลักการทำงานของรูปแบบการ Drive แบบ Full-Step, Half-Step และ Micro-Step ได้ ว่าส่งผลต่อการควบคุมความเร็วและตำแหน่งของ Stepper Motor อย่างไร

3.1.4 เพื่อสามารถอธิบายกระบวนการ Signal Conditioning, Signal Processing ทั้งหมดได้ตั้งแต่ต้นจนจบกระบวนการ ว่าค่าที่อ่านได้จาก Incremental Encoder, Hall Current Sensor มีที่มาอย่างไร อธิบายให้เห็นถึงวิธีคิดและขั้นตอนทั้งหมด ทั้งก่อนและ หลัง Calibrate Sensor หรือ วิธีจัดการข้อมูลที่ได้มา จัดการอย่างไร รวมถึงหาสมการความสัมพันธ์ระหว่างกระแสไฟฟ้าที่มอเตอร์ใช้จริง กับแรงดันไฟฟ้าที่ออกมาจาก Hall Current Sensor และอธิบายกระบวนการการ Unwrap ค่า

3.1.5 เพื่อสามารถเขียนโปรแกรม โดยประยุกต์ใช้ MATLAB และ Simulink ในการสั่งการควบคุมความเร็วของ Stepper Motor ในรูปแบบของความถี่ ความเร็วเชิงมุม และรับค่า ร่วมกับบอร์ด Nucleo STM32G474RE โดยใช้สัญญาณจาก Incremental Encoder, Hall Current Sensor เป็น Input และ แสดงสัญญาณ Output จากการ Log สัญญาณแสดงผลเป็นกราฟจาก Data Inspector ใน MATLAB Simulink แสดงให้เห็นว่าสัญญาณ Output แปรผันตามสัญญาณ Input แบบ Real Time โดยมี Output เป็นความเร็วเชิงมุม และกระแสไฟฟ้า ในหน่วย SI derived

3.2 สมมุติฐาน

การ Drive ของ Stepper Motor แบบ Full-Step, Half-Step, และ Micro-Step มีผลต่อความละเอียดและความเสถียรของการควบคุมตำแหน่งและความเร็ว

3.3 ตัวแปร

3.3.1 ตัวแปรต้น : รูปแบบการ Drive ของ Stepper Motor

3.3.2 ตัวแปรตาม : ตำแหน่งและความเร็ว

3.3.3 ตัวแปรควบคุม : แรงดันไฟฟ้า ความถี่ของสัญญาณ ประเภทของ Stepper Motor

3.4 นิยามศัพท์เฉพาะ

3.4.1 Stepper Motor คือ มอเตอร์แบบ Brushless ชนิดหนึ่งที่เคลื่อนที่แบบเป็นขั้น หรือ Step โดยตำแหน่งการหมุนและความเร็ว จะถูกควบคุมโดยสัญญาณไฟฟ้า รูปแบบการทำงานของ Stepper Motor ใช้หลักการสร้างสนามแม่เหล็กจากขดลวดที่ถูกกระตุ้นที่ละเฟส ซึ่งทำให้โรเตอร์จัดแนวตัวเองไปยังตำแหน่งที่กำหนด Stepper Motor นิยมใช้ในงานที่ต้องการควบคุมตำแหน่งและความเร็ว เช่น หุ่นยนต์, เครื่อง CNC, และเครื่องพิมพ์ 3 มิติ

3.4.2 Step Angle คือ มุมที่ Stepper Motor หมุนได้ต่อหนึ่งขั้น การคำนวณ Step Angle สามารถทำได้โดยใช้สมการ

	$Angle = \frac{360}{n}$	
Angle	คือ องศาที่ได้จากการหมุน 1 ขั้น (step)	
n	คือ จำนวนรอบที่ใช้หมุนจนครบ 1 รอบ	

ยิ่ง Step Angle มีค่าต่ำ ความละเอียดของการหมุนก็จะยิ่งสูง ตัวอย่างเช่น Stepper Motor ที่มี 200 ขั้นต่อรอบ จะมี Step Angle เท่ากับ 1.8 องศา Step Angle นี้เป็นปัจจัยสำคัญที่กำหนดความแม่นยำของมอเตอร์ในการควบคุมตำแหน่ง

3.4.3 สัญญาณ Pulse หมายถึง สัญญาณไฟฟ้ารูปแบบ Square wave ที่ส่งไปยัง Stepper Motor เพื่อควบคุมจำนวน Step และทิศทางการหมุน

3.4.4 Rotor หมายถึง ส่วนที่หมุนของ Stepper Motor หรือเพลลาของมอเตอร์

3.4.5 Stator หมายถึง ส่วนที่อยู่โดยรอบของ Rotor อาจจะเป็นได้ทั้ง แม่เหล็กถาวร หรือขดลวดเหนี่ยวนำ ใน Stepper motor Stator จะเป็นขดลวดเหนี่ยวนำ

3.4.6 โหมดการควบคุม หมายถึง รูปแบบการส่งสัญญาณ Pulse ที่แตกต่างกัน เพื่อให้สามารถควบคุม Stepper motor ได้มีประสิทธิภาพในด้านต่าง ๆ มากขึ้น

3.4.7 Stepper Driver หมายถึง วงจรสำเร็จรูปที่ใช้ทำงานร่วมกับบอร์ดคอนโทรลเลอร์ เพื่อให้สามารถควบคุม Stepper ได้ง่ายขึ้น และมีรูปแบบการควบที่หลากหลาย

3.4.8 Pulse Input หมายถึงสัญญาณไฟฟ้ารูปแบบพัลส์ที่ส่งไปยัง Stepper Motor เพื่อควบคุมจำนวนขั้นตอนและทิศทางการหมุน แต่ละพัลส์ที่ส่งไปจะทำให้มอเตอร์หมุนหนึ่งขั้นตอน

3.4.9 Rotor คือส่วนที่หมุนของ Stepper Motor ซึ่งอยู่ตรงกลางของมอเตอร์ โดยทั่วไปจะประกอบด้วยแม่เหล็กถาวรหรือวัสดุที่ตอบสนองต่อสนามแม่เหล็ก โรเตอร์มีหน้าที่ตอบสนองต่อสนามแม่เหล็กที่สร้างขึ้นจากขดลวดในสเตเตอร์ โดยการหมุนของโรเตอร์เกิดจากแรงดูดและแรงผลักระหว่างขั้วแม่เหล็กของโรเตอร์และสเตเตอร์

3.4.10 Stator คือส่วนที่อยู่หนึ่งของ Stepper Motor ซึ่งเป็นโครงสร้างรอบนอกที่ล้อมรอบโรเตอร์ สเตเตอร์มีขดลวดไฟฟ้าที่เมื่อมีกระแสไฟฟ้าไหลผ่าน จะสร้างสนามแม่เหล็กที่มีอิทธิพลต่อโรเตอร์ สนามแม่เหล็กนี้เป็นตัวกำหนดการเคลื่อนที่ของโรเตอร์ในแต่ละขั้นตอน

3.4.11 Holding Torque หมายถึงแรงบิดสูงสุดที่ Stepper Motor สามารถต้านทานได้ในขณะที่อยู่ในตำแหน่งหยุดนิ่ง และยังคงมีกระแสไฟฟ้าจ่ายให้ ขนาดของ Holding Torque เป็นตัวบ่งบอกถึงความสามารถของมอเตอร์ในการรักษาตำแหน่งโดยไม่ลื่นไถลเมื่อมีแรงภายนอกมากระทำ เช่น การล๊อคตำแหน่งของแกนเครื่องจักรขณะทำงาน

3.4.12 Microstepping เป็นเทคนิคการควบคุมที่ช่วยเพิ่มความละเอียดในการหมุนของ Stepper Motor โดยแบ่งแต่ละขั้นตอนออกเป็นขั้นย่อย เทคนิคนี้ใช้การปรับระดับความเข้มของกระแสในขดลวดแต่ละเฟส เพื่อให้โรเตอร์สามารถเคลื่อนไปยังตำแหน่งระหว่างขั้นตอนเต็ม (Full Step) Microstepping เหมาะสำหรับการใช้งานที่ต้องการความละเอียดสูง เช่น หุ่นยนต์และระบบนำทาง

3.4.13 Full-Step Mode เป็นโหมดการทำงานที่กระตุ้นสองเฟสของขดลวดพร้อมกันในแต่ละขั้นตอนของการหมุน ซึ่งช่วยให้เกิดสนามแม่เหล็กที่เข้มแข็งและสร้างแรงบิดสูงสุด Full-Step Mode เหมาะสำหรับการใช้งานที่ต้องการแรงบิดสูงและการควบคุมที่เรียบง่าย

3.4.14 Half-Step Mode เป็นโหมดการทำงานที่ผสมผสานระหว่างการกระตุ้นเฟสเดียวและสองเฟสในแต่ละขั้นตอน โดยโหมดนี้ช่วยเพิ่มความละเอียดของการหมุน เช่น ลด Step Angle ลงครึ่งหนึ่ง (จาก 90° เหลือ 45°) Half-Step Mode ช่วยปรับสมดุลระหว่างแรงบิดและความละเอียด

3.4.15 Wave Mode เป็นโหมดการทำงานที่กระตุ้นเพียงเฟสเดียวของขดลวดในแต่ละขั้นตอนของการหมุน แม้ว่าจะมีการควบคุมที่ง่าย แต่แรงบิดที่เกิดขึ้นจะต่ำกว่าโหมดอื่นๆ เช่น Full-Step Mode

3.4.16 Bipolar Stepper Motor มีขดลวดอิสระสองชุดต่อเฟส โดยต้องใช้การกลับทิศทางของกระแสไฟฟ้าในขดลวดเพื่อสร้างการหมุน แม้ว่าจะต้องการวงจรควบคุมที่ซับซ้อนกว่า แต่ Bipolar Motor ให้แรงบิดสูงและมีประสิทธิภาพมาก เหมาะสำหรับการใช้งานที่ต้องการความแม่นยำ เช่น เครื่อง CNC และเครื่องพิมพ์ 3 มิติ

3.4.17 Unipolar Stepper Motor มีขดลวดแบบ Center-Tapped ที่ทำให้กระแสไหลได้ในทิศทางเดียว ซึ่งช่วยให้ควบคุมได้ง่ายกว่า Bipolar Motor แต่แรงบิดต่ำกว่า เหมาะสำหรับการใช้งานที่ไม่ต้องการแรงบิดสูง เช่น งานอดิเรกและอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก

3.4.18 Driver Circuit คือวงจรที่ควบคุมการจ่ายกระแสและแรงดันไฟฟ้าให้กับ Stepper Motor วงจรนี้ทำหน้าที่สำคัญในการสร้างลำดับการกระตุ้นขดลวดในเฟสต่างๆ เพื่อให้มอเตอร์หมุนอย่างแม่นยำ

3.4.19 Detent Torque คือแรงบิดต้านที่เกิดขึ้นเองตามธรรมชาติของ Stepper Motor โดยไม่มีกระแสไฟฟ้าจ่ายให้ เกิดจากแม่เหล็กถาวรในมอเตอร์ Detent Torque มีบทบาทสำคัญในการช่วยให้มอเตอร์หยุดอยู่ในตำแหน่ง

3.4.20 Resolution คือความละเอียดในการหมุนของ Stepper Motor วัดจากจำนวนขั้นตอนต่อหนึ่งรอบ ยิ่งจำนวนขั้นตอนต่อรอบสูง ความละเอียดและความแม่นยำก็จะยิ่งดีขึ้น Resolution เป็นปัจจัยสำคัญที่ต้องพิจารณาในงานที่ต้องการการควบคุมตำแหน่งที่แม่นยำ

3.5 นิยามเชิงปฏิบัติการ

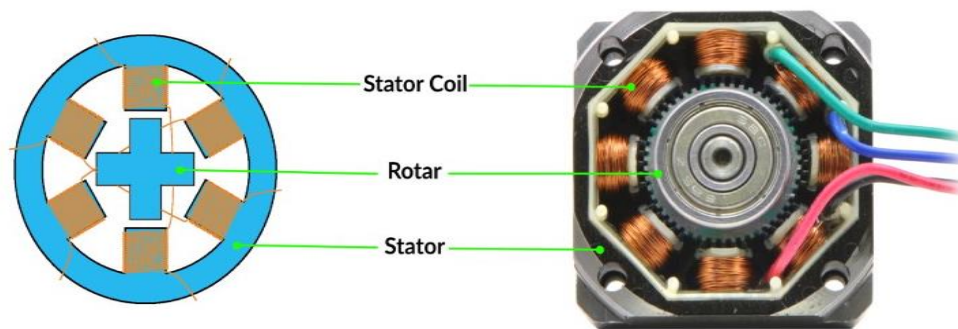
3.5.1 หลักการทำงานของ Stepper Motor : Stepper Motor คืออุปกรณ์ที่แปลงพลังงานไฟฟ้าเป็นการเคลื่อนที่เชิงกลแบบหมุนที่มีความละเอียดสูง โดยหมุนเป็นขั้นตอน (step) ตามสัญญาณพัลส์ที่ส่งมาจากตัวควบคุม การเคลื่อนที่ของมอเตอร์สามารถควบคุมตำแหน่ง ความเร็ว และความเร่งได้อย่างแม่นยำ

3.5.2 การสูญเสียขั้น (Loss Step) : ในการทำงานของ Stepper Motor หากแรงบิดที่ต้องการในการหมุนเกินกว่าที่มอเตอร์จะสามารถสร้างได้ จะเกิดการสูญเสียขั้น (step loss) ซึ่งส่งผลให้ตำแหน่งของมอเตอร์ไม่ตรงกับที่กำหนดไว้

3.5.3 การควบคุมความเร็วและความเร่งของ Stepper Motor : การเปลี่ยนความถี่ของสัญญาณพัลส์จะส่งผลโดยตรงต่อความเร็วของมอเตอร์ และการเปลี่ยนแปลงอัตราการส่งพัลส์สามารถปรับให้มอเตอร์มีความเร่ง (acceleration) หรือความหน่วง (deceleration) ได้ตามต้องการ

3.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

3.6.1 Stepper Motor คือ อุปกรณ์ไฟฟ้าเชิงกลที่ใช้ในการแปลงสัญญาณไฟฟ้าให้เป็นการขับเคลื่อนทางกล แบบเชิงมุมที่สามารถกำหนดองศาการหมุนที่แม่นยำได้ ซึ่งมีความแตกต่างจากมอเตอร์ Brushed DC ทั่วไปที่หมุนอย่างต่อเนื่องเมื่อได้รับพลังงานไฟฟ้าภายในของ Stepper motor จะมีส่วนที่ใช้ในการขับเคลื่อนหลักอยู่ 2 จุด คือ Rotor ที่เป็นแม่เหล็กถาวร และ Stator ที่เป็นขดลวดแม่เหล็กเหนี่ยวนำ สามารถสังเกตได้จากภาพที่ *** โดยการหมุน 1 step คือ มอเตอร์หมุนเคลื่อนที่ 1 ครั้ง แต่จะไม่ใช้การสั่งงานให้หมุน 1 รอบ



ภาพที่ 3.1 ภาพแสดงแผนผังของ Stepper Motor และโครงสร้างจริงภายในของมอเตอร์

3.6.2 Datasheet ของ Stepper Motor จากการสังเกตรอบ ๆ Stepper Motor พบว่า มอเตอร์ที่ใช้ในการทดลองนี้เป็นรุ่น Oukeda Motor OK28STH32-0674A หรือ Nema11 แต่ยังไม่สามารถหา Datasheet ที่ถูกต้องได้

3.6.3 Step Angle คือ องศาที่มอเตอร์สามารถหมุนได้ใน การทำงาน 1 step และหากต้องการให้มอเตอร์หมุนได้ครบ 1 รอบ อาจจะต้องใช้หลาย Step เพื่อให้หมุนครบ 1 รอบ โดยวิธีคำนวณ Step Angle สามารถทำได้ตามสมการดังนี้

	$Angle = \frac{360}{n}$	(1)
Angle	คือ องศาที่ได้จากการหมุน 1 step	
n	คือ จำนวนของ Step ที่ทำให้หมุนครบ 1 รอบ	

- 1) รูปแบบ Motor drive ของ Stepper Motor คือ วิธีการต่อวงจรเพื่อใช้ในการ Drive หรือขับเคลื่อนมอเตอร์
- 2) Datasheet ของ Motor Driver
- 3) สาย
- 4) วาล์ว
- 5) ส
- 6) สี่

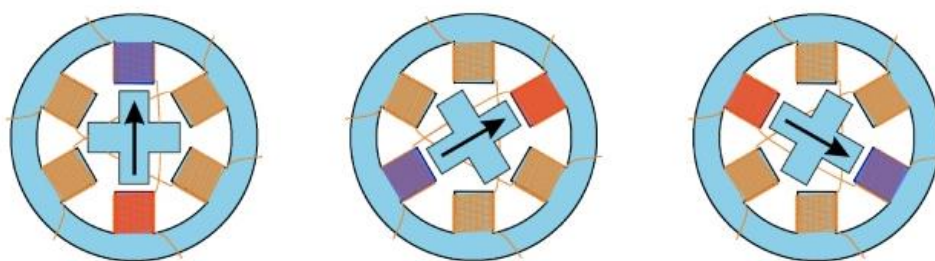
1.

ภาพที่ 3.2 ภาพแสดงแผนผังของ Stepper Motor และโครงสร้างจริงภายในของมอเตอร์

3.6.4 Stepper Motor ประกอบด้วยส่วนสำคัญ 2 ส่วน คือ โรเตอร์ (Rotor) และ สเตเตอร์ (Stator)

- 1) โรเตอร์ (Rotor): ส่วนที่หมุนอยู่ตรงกลาง โดยทั่วไปจะประกอบด้วยแม่เหล็กถาวร
- 2) สเตเตอร์ (Stator): โครงสร้างรอบนอกที่มีขดลวดไฟฟ้า (Coils of Wire)

3.6.5 การทำงานเกิดจากการกระตุ้นขดลวดในลำดับที่กำหนดไว้ล่วงหน้า ซึ่งจะสร้างสนามแม่เหล็กที่ทำให้โรเตอร์เคลื่อนที่เป็นขั้นตอนอย่างแม่นยำ จำนวนขั้น (Steps) ที่มอเตอร์สามารถหมุนได้ใน 1 รอบ เรียกว่า Step Angle ซึ่งขึ้นอยู่กับ การออกแบบและโครงสร้างของมอเตอร์



ภาพที่ 3.3 สีของฟันเฟืองในภาพแสดงถึงทิศทางของสนามแม่เหล็กที่เกิดจากการกระตุ้นขดลวดของสเตเตอร์ ช่วยให้เข้าใจการเคลื่อนที่ของโรเตอร์ได้ชัดเจนยิ่งขึ้น

หลักการทำงานพื้นฐานของ Stepper Motor มีดังนี้: เมื่อมีการกระตุ้นหนึ่งหรือมากกว่าหนึ่งเฟสของขดลวดในสเตเตอร์ (Stator) กระแสไฟฟ้าที่ไหลผ่านขดลวดจะก่อให้เกิดสนามแม่เหล็ก ซึ่งทำให้โรเตอร์ (Rotor) เคลื่อนที่ไปจัดแนวกับสนามแม่เหล็กที่เกิดขึ้น การกระตุ้นเฟสต่างๆ ต่อเนื่องกันเป็นลำดับ จะทำให้โรเตอร์หมุนที่ละมุมอย่างแม่นยำจนไปถึงตำแหน่งที่ต้องการ

- 1) เริ่มต้นที่ขดลวด A ถูกกระตุ้น โรเตอร์จะจัดแนวตัวเองให้ตรงกับสนามแม่เหล็กที่ขดลวด A สร้างขึ้น
- 2) เมื่อขดลวด B ถูกกระตุ้น สนามแม่เหล็กใหม่จะเกิดขึ้น ทำให้โรเตอร์หมุนตามเข็มนาฬิกาไปอีก 60° เพื่อจัดแนวกับสนามแม่เหล็กนี้กระบวนการเดียวกันนี้เกิดขึ้นเมื่อขดลวด C ถูกกระตุ้น

3.6.6 Stepper Motor มีหลากหลายประเภทและการออกแบบที่ปรับให้เหมาะสมกับความต้องการของการใช้งานแต่ละแบบ โดยประเภทที่พบได้บ่อย ได้แก่ Permanent Magnet Stepper Motors, Variable Reluctance Stepper Motors, และ Hybrid Stepper Motors ดังนี้:

- 1) Permanent Magnet Stepper Motor รวมโรเตอร์ที่เป็นแม่เหล็กถาวรและสเตเตอร์ที่มีขดลวดแม่เหล็กไฟฟ้าระหว่างการทำงาน ขดลวดที่ถูกกระตุ้นจะสร้างสนามแม่เหล็กซึ่งทำให้ฟันหรือขั้วของโรเตอร์จัดเรียงตัวกับสเตเตอร์มอเตอร์ชนิดนี้ได้รับการยอมรับว่ามีแรงบิดที่ดี จึงเหมาะกับการใช้งานที่ต้องการความแม่นยำในระดับปานกลาง
- 2) Variable Reluctance Stepper Motors มอเตอร์แบบนี้มีโรเตอร์ที่มีลักษณะเด่นชัดด้วยฟันและสเตเตอร์ที่มีขดลวดแม่เหล็กไฟฟ้า ประสิทธิภาพการทำงานเกิดจากการที่โรเตอร์พยายามเรียงตัวไปตามขั้วของสเตเตอร์ซึ่งมีความเหนี่ยวนำน้อยที่สุด ส่งผลให้เกิดการจัดเรียงตัวกับสนามแม่เหล็กของสเตเตอร์ได้อย่างแม่นยำ มอเตอร์ประเภทนี้สามารถทำงานด้วยความเร็วและความละเอียดที่สูงกว่า แต่แรงบิดมักจะต่ำกว่าและไม่มีแรงบิดขณะหยุด (detent torque)
- 3) Hybrid Stepper Motors มอเตอร์ชนิดนี้มีการออกแบบโรเตอร์แบบไฮบริดที่รวมคุณสมบัติของ Permanent Magnet และ Variable Reluctance เข้าด้วยกัน โรเตอร์มีฝาปิดสองด้านที่มีฟันเรียงตัวสลับกันและถูกแม่เหล็กในแนวแกน การออกแบบนี้ช่วยให้มอเตอร์มีข้อดีจากทั้งสองประเภท เช่น ความละเอียด ความเร็ว และแรงบิดที่สูง อย่างไรก็ตาม การเพิ่มประสิทธิภาพนี้ต้องแลกมาด้วยโครงสร้างที่ซับซ้อนและต้นทุนการผลิตที่สูงขึ้น มอเตอร์ชนิดนี้จึงเหมาะกับการใช้งานที่หลากหลาย เช่น เครื่อง CNC, เครื่องพิมพ์ 3D และระบบหุ่นยนต์

3.6.7 Stepper Motor Drive Type

1) Unipolar Stepper Motor มีลักษณะเด่นคือสเตเตอร์ที่มีขดลวดแบบ Center-Tapped ทำให้กระแสไฟฟ้าไหลได้ในทิศทางเดียวผ่านแต่ละขดลวด โดยทั่วไปแล้วมอเตอร์ประเภทนี้จะมีสายไฟ 4 หรือ 5 เส้น โดยแต่ละขดลวดจะถูกแบ่งออกเป็นสองส่วน มอเตอร์แบบ Unipolar มีการควบคุมที่ง่ายกว่า เนื่องจากต้องการเพียงการสลับขั้วของกระแสในขดลวดเดียวเพื่อสร้างการเคลื่อนที่ อย่างไรก็ตาม แรงบิดของมอเตอร์ชนิดนี้มักจะต่ำกว่า Bipolar Stepper Motors

2) Bipolar Stepper Motors มีลักษณะเด่นคือมีขดลวดอิสระสองชุดต่อเฟส และไม่มี Center Tap โดยทั่วไปจะมีสายไฟ 4 เส้น ซึ่งแต่ละเส้นเชื่อมต่อกับปลายของขดลวดทั้งสองชุด ต่างจาก Unipolar Motors, Bipolar Motors ต้องการการกลับทิศทางของกระแสไฟฟ้าในแต่ละขดลวดเพื่อสร้างการเคลื่อนที่ไวด์ แม้ว่าจะต้องใช้วงจรควบคุมที่ซับซ้อนกว่า แต่ก็ให้แรงบิดที่สูงกว่าและมีประสิทธิภาพมากกว่า

3.6.8 Stepper Motor Drive Mode สเต็ปเปอร์มอเตอร์ใช้เทคนิคการขับเคลื่อนที่หลากหลายเพื่อให้สามารถควบคุมการเคลื่อนที่ได้อย่างแม่นยำ เทคนิคเหล่านี้รวมถึงโหมดคลื่น โหมดเต็มขั้นตอน โหมดครึ่งขั้นตอน และไมโครสเต็ปปิง ในโหมดคลื่น มอเตอร์จะจ่ายไฟครึ่งเฟสเดียว ส่งผลให้โรเตอร์เคลื่อนที่ทีละขั้น ตัวอย่างเช่นในมอเตอร์สองเฟสกระแสจะไหลผ่านเฟส A ในทิศทางเดียวทำให้โรเตอร์อยู่ในแนวเดียวกับสนามแม่เหล็ก ต่อจากนั้นเฟส B จะได้รับพลังงานโดยหมุนโรเตอร์ 90° เพื่อให้สอดคล้องกับสนามแม่เหล็ก กระบวนการนี้ทำซ้ำสลับกันระหว่างเฟส A และ B โดยมีการเปลี่ยนแปลงในทิศทางปัจจุบันเป็นตัวกำหนดทิศทางการเคลื่อนที่ของโรเตอร์

1) โหมด Full-Step จะกระตุ้นสองเฟสพร้อมกันในแต่ละขั้นตอนของการหมุน ซึ่งช่วยเพิ่มแรงบิด (Torque) เมื่อเทียบกับโหมด Wave Mode รูปแบบการเคลื่อนที่ไวด์จะเป็นลำดับทีละขั้นเช่นเดียวกัน แต่ด้วยการกระตุ้นทั้งสองเฟสพร้อมกันในแต่ละขั้นตอน ทำให้เกิดสนามแม่เหล็กที่แรงขึ้นและเพิ่มแรงบิดของมอเตอร์

2) โหมด Half-Step เป็นการผสมผสานระหว่างโหมด Wave และ Full-Step โดยลดขนาดของแต่ละขั้นตอนลงครึ่งหนึ่ง (เช่น จาก 90° เหลือ 45°) แม้ว่าโหมดนี้จะให้ความละเอียดที่สูงขึ้น แต่แรงบิดจะมีความแปรผัน โดยจะสูงขึ้นเมื่อมีการกระตุ้นสองเฟสพร้อมกัน และลดลงเมื่อมีการกระตุ้นเพียงเฟสเดียว

3) Microstepping ช่วยเพิ่มความละเอียดของการเคลื่อนไหวด้วยการควบคุมความเข้มของกระแสในแต่ละเฟส ทำให้สามารถแบ่งขั้นตอนการหมุนออกเป็นขนาดเล็กลงได้ เทคนิคนี้ใช้การปรับระดับกระแสทีละขั้นเพื่อให้โรเตอร์อยู่ในตำแหน่งระหว่างขั้นตอนเต็ม (Full Steps) แม้ว่า Microstepping จะช่วยให้มอเตอร์มีความละเอียดสูงมากในการกำหนดตำแหน่ง แต่ก็ต้องใช้วงจรขับที่ซับซ้อนมากขึ้น และแรงบิดต่อขั้นจะลดลงเนื่องจากขนาดของขั้นที่เล็กลง

3.6.9 Loss Step ใน Stepper Motor หมายถึงสถานการณ์ที่โรเตอร์ของมอเตอร์ไม่สามารถหมุนไปยังตำแหน่งที่กำหนดตามคำสั่งได้อย่างแม่นยำ ซึ่งอาจเกิดจากปัจจัยดังนี้:

- 1) แรงบิดไม่เพียงพอ : มอเตอร์ไม่สามารถสร้างแรงบิดมากพอที่จะเอาชนะแรงต้าน เช่น แรงเฉื่อยของโหลด แรงเสียดทาน หรือแรงบิดขณะหยุด (Detent Torque)
- 2) แรงเสียดทานและน้ำหนักโหลดสูงเกินไป : หากแรงเสียดทานในระบบ เช่น ลูกปืน หรือบอลสกรู สูงเกินไป จะทำให้มอเตอร์ไม่สามารถหมุนได้จนกว่าจะแรงสะสมเพียงพอ
- 3) การตั้งค่าที่ไม่เหมาะสม : เช่น การใช้ Microstepping ที่มีขั้นน้อยมากเกินไป ส่งผลให้แรงบิดในแต่ละขั้นน้อยต่ำลง ไม่เพียงพอที่จะเคลื่อนที่โรเตอร์

3.6.10 วิธีป้องกัน Loss Step

- 1) ลดความถี่ของพัลส์ (Pulse Frequency) ที่ส่งให้มอเตอร์ เพื่อให้แรงบิดในแต่ละขั้นตอนเพียงพอต่อการเคลื่อนที่
- 2) ใช้แหล่งจ่ายไฟ (Power Supply) ที่มีแรงดันไฟฟ้าสูงขึ้น เพื่อให้มอเตอร์สร้างแรงบิดได้สูงขึ้น
- 3) ลดจำนวน Microsteps ต่อ Full Step เพื่อเพิ่มแรงบิดในแต่ละขั้นตอน ลดแรงเสียดทานในระบบ

3.6.11 วิธีตรวจสอบ Loss Step

- 1) ใช้ Encoder ตรวจสอบตำแหน่งจริงของโรเตอร์เปรียบเทียบกับตำแหน่งที่สั่งการ หากมีความแตกต่าง แสดงว่าเกิด Loss Step
- 2) ตรวจสอบกระแสที่จ่ายให้กับมอเตอร์ หากกระแสต่ำกว่าค่าที่มอเตอร์ต้องการ อาจเกิด Loss Step
- 3) ใช้ Oscilloscope ตรวจสอบสัญญาณพัลส์ที่ส่งไปยังมอเตอร์ เพื่อดูว่ามีความผิดปกติของสัญญาณหรือไม่ เช่น สัญญาณขาดหาย (Pulse Dropout) หรือรูปคลื่นผิดเพี้ยน

3.7 วิธีดำเนินการทดลอง

3.7.1 เริ่มจากดาวน์โหลดโปรแกรมที่สำคัญกับการทดลอง ติดตั้งและเชื่อมต่อสายสัญญาณต่าง ๆ ในบอร์ดการทดลองให้เรียบร้อย

3.7.2 เปิดโปรแกรมที่ได้รับจาก TA และทดสอบการใช้งานก่อนทดลองจริง

3.7.3 เขียนโปรแกรมที่สามารถเก็บค่าตัวแปรเมื่อเริ่มการทดลอง ให้เหมาะสมกับการทดลอง

3.7.4 เริ่มการทดลอง โดยอ่านค่าความเร็วจากโปรแกรม และเก็บค่าลงใน Matlab เพื่อนำไปวิเคราะห์

3.8 วัสดุอุปกรณ์

3.8.1 Nucleo STM32G474RE จำนวน 1

3.8.2 RS PRO Hybrid, Permanent Magnet Stepper Motor จำนวน 1 อัน

3.8.3 incremental Encoder AMT103-V จำนวน 1

3.8.4 driver DRV8825 จำนวน 1

3.8.5 MotorXplorer จำนวน 1

3.9 ขั้นตอนการดำเนินงาน

3.9.1 ดาวน์โหลดโปรแกรม Matlab และดาวน์โหลดไฟล์ควบคุมมอเตอร์ที่ได้รับจาก TA

1) เมื่อดาวน์โหลดแล้ว จะได้รับไฟล์ MotorXploer, RMX_Motor.bin, RMX_Motor.hex และ Waijang 1

Download Link

2) เปิดไฟล์ Waijang 1 Download Link จะขึ้นเป็นการโหลดไฟล์ Waijang จากนั้นแตกไฟล์ .zip ไว้ใน

โฟลเดอร์เดียวกับไฟล์ MotorXploer เมื่อแตกไฟล์จะพบไฟล์ Install_waijang เป็นไฟล์ Matlab code

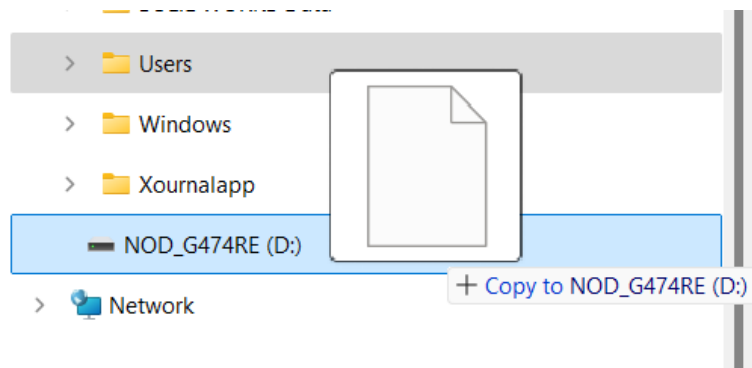
3) เปิดโปรแกรม Matlab และกด Run ไฟล์ Install_waijang รอดติดตั้งจนโปรแกรมขึ้นข้อความว่า “Finish Waijung Installation.” ถือว่าการติดตั้งเสร็จเรียบร้อยแล้ว

4) ปิดและเปิดโปรแกรมใหม่อีกครั้ง

Name	Date modified	Type	Size
MatlabDataDCMotor	18/12/2567 11:34	File folder	
slprj	18/12/2567 4:56	File folder	
waijung_18.11a	17/12/2567 11:58	File folder	
Fortest_Motorlab	18/12/2567 7:56	Simulink Model	150 KB
MotorXploer	18/12/2567 16:31	Simulink Model	230 KB
RMX_Motor.bin	17/12/2567 9:46	BIN File	44 KB
RMX_Motor.hex	17/12/2567 9:46	HEX File	121 KB
Waijang 1 Download Link	17/12/2567 9:46	Internet Shortcut	1 KB
MotorXploer.slx.autosave	18/12/2567 16:36	AUTOSAVE File	202 KB

ภาพที่ 3.4 ภาพแสดงไฟล์ทั้งหมดที่ได้รับมาจาก TA

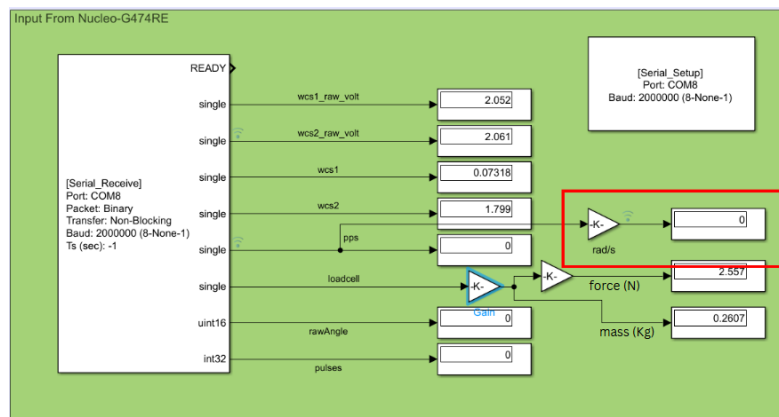
3.9.2 ทดลองใช้งานก่อนทดลอง เมื่อเชื่อมต่อบอร์ดควบคุม กับต่อไฟเลี้ยง 24 V และเปิดสวิตช์เรียบร้อยแล้ว ให้นำไฟล์ RMX_Motor.bin ลากมาใส่ยังไดรฟ์ของ NOD_G474RE (D:) ที่ปรากฏด้านซ้ายมือของ Flie Exploror และกดปุ่ม Reset ที่บอร์ดควบคุมอีกครั้ง หรือกดทุกครั้งเมื่อสงสัยว่ามอเตอร์มีการทำงานผิดปกติ



ภาพที่ 3.5 ภาพการลากไฟล์ RMX_Motor.bin มาใส่ยัง NOD_G474RE (D:)

3.9.3 จากนั้นเปิดไฟล์ Simulink ชื่อว่า MotorExploer จะเจอโปรแกรมการควบคุมมอเตอร์พื้นฐานที่ TA ได้จัดทำไว้ โดยจะมีทั้ง ค่าที่อ่านได้จากเซนเซอร์วัดกระแส, Encoder วัดความเร็วเชิงมุม, พารามิเตอร์ที่ใช้ควบคุมมอเตอร์ โดยเราจะขออธิบายทีละส่วน เริ่มจากส่วนค่าที่อ่านได้จากเซนเซอร์ ในการทดลองนี้ เราจะสนใจแค่ตัวแปรที่อยู่ในกรอบสีแดงเท่านั้น

- 1) pps หรือ Pulse per second และ rad/s คือ ค่าสัญญาณ Pulse ต่อวินาที จาก Encoder เพื่อนำมาคำนวณความเร็วที่ได้จากการหมุนของมอเตอร์ มีหน่วยเป็น rad/s



ภาพที่ 3.6 ภาพบล็อกโปรแกรมที่ TA จัดทำให้ ในส่วนค่าที่อ่านได้จากเซนเซอร์

3.9.4 ในส่วนที่สอง คือพารามิเตอร์ที่ใช้ในการควบคุมมอเตอร์ โดยเราจะสนใจพารามิเตอร์ที่อยู่ในกรอบสีแดงเท่านั้น

- 1) M0 M1 และ M2 คือ พารามิเตอร์ที่ใช้ปรับโหมดการไต่พื่นในรูปแบบ Digital Enable ซึ่งจะปรับได้ถึง 6 โหมด Step ด้วยกัน

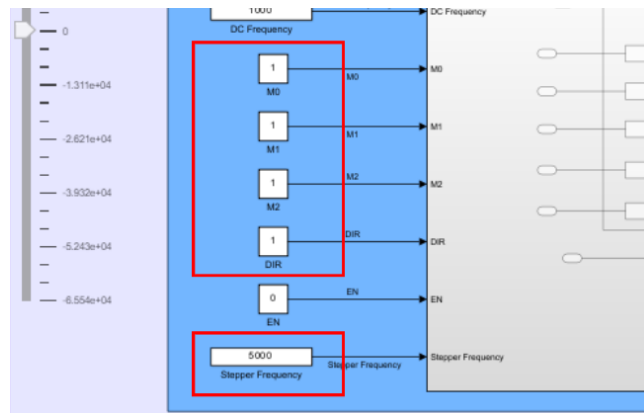
Table 1. Stepping Format

MODE2	MODE1	MODE0	STEP MODE
0	0	0	Full step (2-phase excitation) with 71% current
0	0	1	1/2 step (1-2 phase excitation)
0	1	0	1/4 step (W1-2 phase excitation)
0	1	1	8 microsteps/step
1	0	0	16 microsteps/step
1	0	1	32 microsteps/step
1	1	0	32 microsteps/step
1	1	1	32 microsteps/step

ภาพที่ 3.7 ภาพรูปแบบโหมด Step ที่สามารถทำได้ทั้ง 6 โหมด Step

2) DIR หรือ Direction คือ ที่ตั้งค่าทิศทางการหมุนของมอเตอร์

3) Stepper Frequency คือ ความถี่ของสัญญาณ Pluse ที่จ่ายเข้าไปยัง Stepper motor ภายใน 1 วิ มีหน่วยเป็น Hz (เฮิรตซ์)



ภาพที่ 3.8 ภาพลือคโปรแกรมที่ TA จัดทำให้ ในส่วนพารามิเตอร์ปรับได้ที่เกี่ยวข้องกับ Stepper Motor

3.9.5 ในการทดสอบนี้ พบว่าในระบบของมอเตอร์เกิดการฝืดจากการใช้ Bearing ที่ไม่ขนานกับเฟลมอเตอร์ และ Pulley ที่ใช้ร่วมกับ Brushed ก็มีแรงเสียดทานที่ก้านเฟลด้วยเช่นกัน

เราจึงกังวลว่าอาจส่งผลอย่างมากต่อผลการทดลองที่ได้ต่อจากนี้

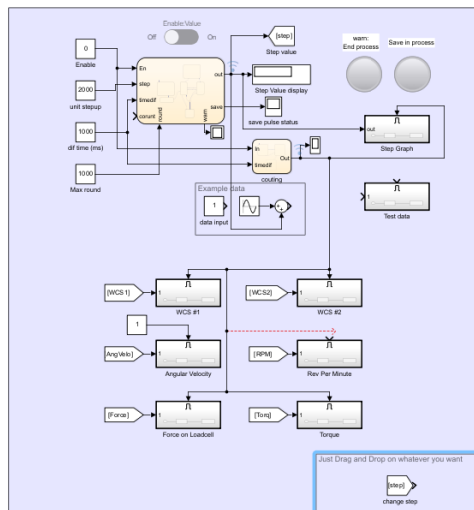
เราจึงได้ออกแบบวิธีการใช้ชุดการทดลองใหม่เพื่อลดความฝืดที่เกิดขึ้น โดยลดการเชื่อมต่อทางกลทั้งหมด

ถอดอุปกรณ์ที่ไม่มีความเกี่ยวข้องออก และใช้สเปรย์ช่วยลดความฝืด เช่น WD-40

3.9.6 ในการเก็บค่าพารามิเตอร์ต่าง ๆ ที่เซนเซอร์อ่านได้ เราได้ใช้ Simulink ในการเก็บค่าโดยใช้บล็อก To workspace เพื่อส่งค่าออกไปยัง Workspace และบันทึกค่าเป็นไฟล์นามสกุล .mat เพื่อนำไปพิจารณาต่อไป ทั้งนี้

ในส่วนการเก็บค่าความเร็วที่เปลี่ยนไปตามความถี่ เราอยากเห็นภาพรวมของการเปลี่ยนแปลง หรือก็คือ เราอยากให้ค่าพารามิเตอร์เปลี่ยนไปในขณะที่มอเตอร์กำลังทำงาน

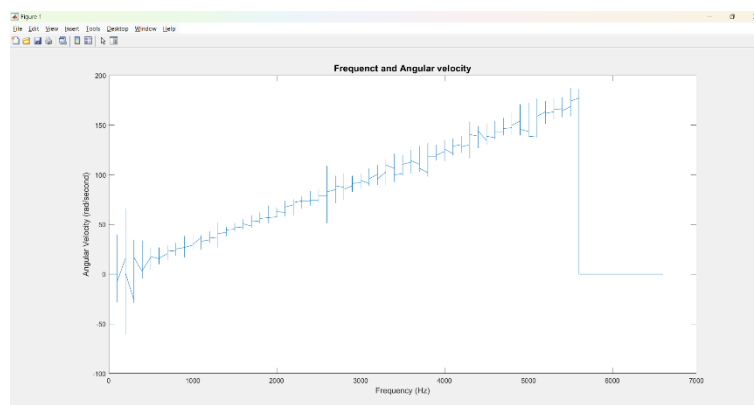
โดยให้ระยะเวลาในการเปลี่ยนค่าพารามิเตอร์มีค่าเท่ากัน เราจึงได้ใช้ Simulink ในการสร้างระบบที่สามารถเพิ่มค่าขึ้นแบบ Step ได้ และยังสามารถกำหนดระยะเวลาในการเพิ่มของ Step ได้อีกด้วย



ภาพที่ 3.9 ภาพ Simulink ระบบที่เราใช้เพิ่มค่าความถี่ของ Stepper frequency ในข้อที่ 3) และเก็บค่าที่อ่านได้จาก Encoder

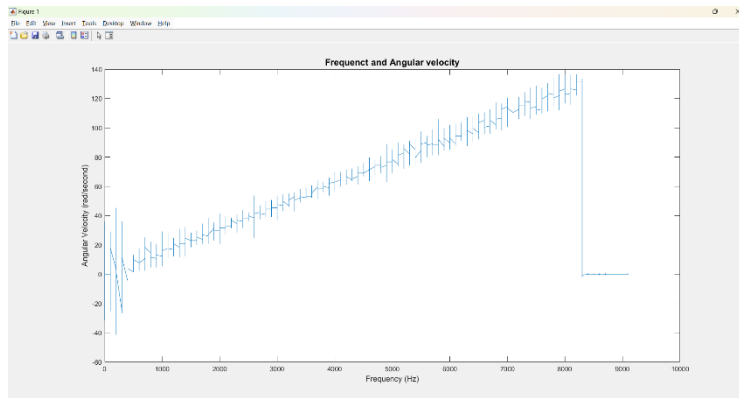
3.10 ผลการทดลอง

ผลการทดลองของการไต่แบบ **Full step Mode** โดยเรานำความเร็วการหมุนที่ได้มาทำกราฟเทียบความถี่ของสัญญาณ



ภาพที่ 3.10 ภาพกราฟการไต่แบบ **Full Step mode**

ผลการทดลองของการไต่แบบ **Half step Mode** โดยเรานำความเร็วการหมุนที่ได้มาทำกราฟเทียบความถี่ของสัญญาณ



3.11 สรุปผล

3.12 อภิปรายผล

3.13 ข้อเสนอแนะ

3.14 เอกสารอ้างอิง

Understanding Stepper Motors: Types, Principles, Applications

Stepper Motors and the Myth of Lost Steps - Motion Solutions