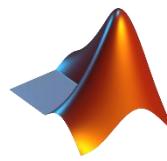


RMXplorer



LAB2: Actuator

สมาชิก

- นายภูนุวัฒน์ บุญเกิด 66340500042
- นายภูมิญ ประเสริฐสม 66340500045
- นายรัชฎพงศ์ สายบุญรอด 66340500047

วัตถุประสงค์

- คณผู้จัดทำต้องสามารถออกแบบการทดลองโดยใช้ทักษะกระบวนการทางวิทยาศาสตร์ เพื่อสืบเสาะ พฤติกรรมและปรากฏการณ์ บันทึกผลการทดลองสรุปผล และอภิปรายผลได้อย่างถูกต้องรวมถึงเข้าใจ หลักการทำงานของเซ็นเซอร์และอุปกรณ์ทั้งหมดที่เกี่ยวข้อง ตลอดจนใช้ หลักการ ทางวิทยาศาสตร์ คณิตศาสตร์ สลักติ และศาสตร์อื่นๆ ที่เกี่ยวข้องเพื่อประยุกต์ใช้ร่วมกับโปรแกรม MATLAB ใน การเก็บ และวิเคราะห์ผลการทดลอง พร้อมตรวจสอบความเที่ยงตรง และความแม่นยำของข้อมูลได้อย่างถูกต้อง พร้อมมีเหตุผลรองรับ และเปรียบเทียบความถูกต้อง กับทฤษฎีที่น่าเชื่อถือ
- คณผู้จัดทำต้องสามารถอธิบายความสามารถในการตรวจวัดปริมาณทางฟิสิกส์ของเซ็นเซอร์ทั้งหมดตั้งแต่ ต้นจนจบกระบวนการ เช่น การวัดกระแสไฟฟ้าของ Current Sensor (จากแรงดันไฟฟ้าแปลงเป็นกระแสไฟฟ้าได้อย่างไร) หรือการวัดความเร็วของ Brushless DC Motor จาก Hall Sensor (จากคลื่นสัญญาณไฟฟ้าแปลงเป็นความเร็วได้อย่างไร)
- คณผู้จัดทำต้องสามารถกำหนดตัวแปรในการทดลองได้อย่างถูกต้องและสมเหตุสมผล เช่น ตัวแปรต้น ตัวแปรตาม และตัวแปรควบคุม พร้อมทั้งอธิบายจุดประสงค์การทดลอง สมมติฐานที่สอดคล้องกับตัวแปร นิยามเชิงปฏิบัติการ และทฤษฎีที่น่าเชื่อถือรองรับ เช่น ทฤษฎีทางฟิสิกส์หรือข้อมูลจาก Datasheet
- คณผู้จัดทำต้องสามารถออกแบบวิธีการทดลองเพื่อหาคำตอบ โดยสอดคล้องกับผลการเรียนรู้ย่อทั้งหมดได้อย่างถูกต้องตามหลักวิทยาศาสตร์ รวมถึงการบันทึกผล สรุปผล และอภิปรายผลจากข้อมูลที่บันทึกได้จริง พร้อมมีการทำซ้ำกระบวนการ อธิบายแหล่งที่มาของผลการทดลอง และใช้อุปกรณ์ เครื่องมือ เช่น ไฟล์ Simulink, mlx หรือชุดการทดลองพร้อมบอร์ด Microcontroller ที่ได้รับการจัดเตรียมไว้
- คณผู้จัดทำต้องสามารถเขียนรายงานที่ถูกต้องตามหลักสากล เช่น การเขียนสัญลักษณ์และสมการทาง วิทยาศาสตร์ การตั้งขนาดและรูปแบบตัวอักษร การเว้นช่องไฟ การเว้นขอบกระดาษ และการจัดระเบียบ พิมพ์ให้ได้ระยะที่เหมาะสมตามหลักการที่ยอมรับในระดับสากล

1. Brushless DC Motor

1.1. จุดประสงค์

- คณะผู้จัดทำต้องทำการวิเคราะห์ผลการเรียนรู้หลัก
- คณะผู้จัดทำต้องสามารถอธิบายหลักการทำงานของ Brushless DC (BLDC) Motor ได้อย่างละเอียด
- คณะผู้จัดทำต้องเข้าใจวิธีการตั้งค่ามอเตอร์ในโปรแกรม Motor Workbench 6.3.2
- คณะผู้จัดทำต้องเข้าใจวิธีการตั้งค่า PWM Generation Frequency และ Speed Sensing
- คณะผู้จัดทำต้องเข้าใจและสามารถอธิบาย Electrical Parameters, Mechanical Parameters, และ Motor Magnetic Structure ของ BLDC Motor
- คณะผู้จัดทำต้องเข้าใจประเภทการควบคุมของ BLDC Motor ทั้งในรูปแบบ Sensorless Control และ Sensor-based Control
- คณะผู้จัดทำต้องสามารถวิเคราะห์และอธิบายกราฟสัญญาณ 3 เฟสของ BLDC Motor พร้อมอภิราย พฤติกรรมที่พบรจากกราฟตามที่ข้อย่ออยู่ต่อไปนี้:
 - หลักการทำงานของ BLDC Motor
 - ลักษณะของ Trapezoidal Back EMF
 - โครงสร้างของ Permanent Magnet Synchronous Machine (PMSM)
 - การควบคุมแบบ Trapezoidal Control
 - การควบคุมแบบ Field-Oriented Control (FOC)
 - เทคนิค Pulse Width Modulation (PWM)
 - การทำงานของ Commutation Logic
 - การควบคุมแบบ Six-Step Commutation
 - เทคนิค Space Vector Modulation (SVM)
 - การควบคุมแบบ Sensorless และการใช้ Hall Effect Sensor
- คณะผู้จัดทำต้องนำผลการทดลองมาเปรียบเทียบกับพฤติกรรมของ PMSM ที่ควบคุมแบบ FOC โดยใช้แหล่งข้อมูลภายนอก เช่น MATLAB
- คณะผู้จัดทำต้องเข้าใจวิธีการคำนวณความเร็วของ BLDC Motor จาก Frequency ของสัญญาณ ที่อ่านได้จาก Oscilloscope
- คณะผู้จัดทำต้องเข้าใจการใช้งาน Oscilloscope 4 Channels ในการจับสัญญาณของ BLDC Motor ทั้ง 3 เฟสอย่างถูกต้อง

1.2. สมมติฐาน

- 1.2.1. การควบคุม BLDC Motor แบบ Sensorless จะมีประสิทธิภาพที่แตกต่างจากแบบ Sensor-based โดยคาดว่าการควบคุมด้วย Sensor จะให้การตอบสนองที่รวดเร็วและแม่นยำกว่าเนื่องจากสามารถตรวจสอบจับตำแหน่งโรเตอร์ได้โดยตรง
- 1.2.2. ความถี่ของ PWM ที่ตั้งค่าในโปรแกรม Motor Workbench จะส่งผลต่อความرابรื่นของสัญญาณ 3 เฟสและพฤติกรรมการทำงานของ BLDC Motor โดยคาดว่าความถี่ที่สูงกว่าจะให้การควบคุมที่นุ่มนวลมากขึ้น
- 1.2.3. สัญญาณ 3 เฟสที่วัดได้จาก BLDC Motor จะมีลักษณะตามรูปแบบการควบคุม เช่น การควบคุมแบบ Trapezoidal จะให้สัญญาณที่มีลักษณะเชิงเหลี่ยม ในขณะที่การควบคุมแบบ FOC จะให้สัญญาณที่มีลักษณะคลื่นไส้
- 1.2.4. การเปรียบเทียบผลการทดลองของ BLDC Motor ที่ควบคุมแบบ PMSM (FOC) กับแบบ Trapezoidal จะช่วยแสดงความแตกต่างในด้านความเสถียรและประสิทธิภาพ
- 1.2.5. การใช้ Oscilloscope 4 Channels จะช่วยให้สามารถวิเคราะห์พฤติกรรมสัญญาณ 3 เฟสของ BLDC Motor ได้อย่างแม่นยำและช่วยยืนยันสมรรถนะของระบบควบคุม
- 1.2.6. การเปลี่ยนแปลงของความเร็ว BLDC Motor จะสัมพันธ์กับความถี่ของสัญญาณที่จับได้จาก Oscilloscope โดยตรง และสามารถคำนวณความเร็วได้ด้วยสมการทางไฟฟ้า
- 1.2.7. Electrical Parameters และ Mechanical Parameters ของ BLDC Motor จะส่งผลต่อกราฟสัญญาณ และสามารถนำมารวิเคราะห์เพื่อปรับปรุงการตั้งค่าการควบคุมให้เหมาะสมที่สุด

1.3. ตัวแปร

- 1.3.1. ตัวแปรต้น (Independent Variables)

1.3.1.1. Electrical Parameters

- ค่าความต้านทานของดาวลากของมอเตอร์ (Phase Resistance, R)
- ค่าความเหนี่ยวนำของดาวลากมอเตอร์ (Phase Inductance, L)
- แรงดันไฟฟ้าขาเข้า (Input Voltage)

1.3.1.2. Mechanical Parameters

- แรงบิด (Torque)
- ความเฉื่อยของโรเตอร์ (Rotor Inertia)
- ความเร็ว rob ที่ตั้งค่า (Setpoint Speed)

1.3.1.3. Mechanical Parameters

- จำนวนขั้วแม่เหล็ก (Number of Pole Pairs)
- รูปทรงสนามแม่เหล็ก (Magnetic Field Shape: Trapezoidal หรือ Sinusoidal)

1.3.2. ตัวแปรตาม (Dependent Variables)

1.3.2.1. Electrical Parameters

- แรงดันไฟฟ้าขาออก (Output Voltage)
- กระแสไฟฟ้าขาออก (Output Current)
- Back EMF (Electromotive Force)
- รูปแบบสัญญาณ PWM (Pulse Width Modulation Signal)

1.3.2.2. Mechanical Parameters

- ความเร็ว (Speed)
- แรงบิดที่เกิดขึ้น (Generated Torque)
- แรงเฉื่อย (Inertia)

1.3.2.3. Mechanical Parameters

- พฤติกรรมของสนามแม่เหล็ก (Magnetic Field Behavior)
- จำนวนรอบต่อการหมุน (Commutation Steps per Rotation)
- การเปลี่ยนเฟสของสัญญาณ (Phase Commutation)

1.3.3. ตัวแปรควบคุม (Control Variables)

1.3.3.1. Electrical Parameters

- การตั้งค่ากระแสขาออกสูงสุด (Max Output Current)
- การตั้งค่า PWM Frequency

1.3.3.2. Mechanical Parameters

- สภาพโหลด (Load Condition)
- ความเสียดทานในระบบ

1.3.3.3. Mechanical Parameters

- โครงสร้างแม่เหล็กของมอเตอร์ (Permanent Magnet Type: Ferrite หรือ Neodymium)
- ขนาดของมอเตอร์และโรเตอร์

1.4. นิยามศัพท์เฉพาะ

- Trapezoidal Back EMF คือ ลักษณะเฉพาะของแรงเดือนไฟฟ้าย้อนกลับ (Back Electromotive Force หรือ Back EMF) ในมอเตอร์กระแสตรงไร้แปรงถ่าน (Brushless DC Motor หรือ BLDC Motor) ซึ่งเกิดขึ้นในชุดลวดเมื่อโรเตอร์หมุนในสนามแม่เหล็ก固定 โดยลักษณะของแรงดันไฟฟ้านี้จะมีรูปคลื่นที่มีลักษณะใกล้เคียง "สี่เหลี่ยมคงที่" (Trapezoidal waveform) ไม่ใช่ลักษณะไซน์ (Sinusoidal waveform) เช่นในมอเตอร์同步 Permanent Magnet Synchronous Motor (PMSM)

- Permanent Magnet Synchronous Machine (PMSM) คือ มอเตอร์ไฟฟ้ากระแสสลับที่ใช้แม่เหล็กถาวรในโรเตอร์แทนการใช้ชุดลวดหรือแหล่งจ่ายกระแสไฟฟ้าเพื่อสร้างสนามแม่เหล็ก ลักษณะการทำงานของ PMSM คือโรเตอร์จะหมุนด้วยความเร็วที่สัมพันธ์กับความถี่ของกระแสไฟฟ้าที่ป้อนเข้าสู่ชุดลวดสเตเตอร์ (Synchronous Speed)
- Trapezoidal Control คือ วิธีการควบคุมมอเตอร์แบบ Brushless DC Motor (BLDC) ที่อาศัยหลักการควบคุมแรงดัน (หรือกระแส) ให้เข้ากับลักษณะของแรงดันกลับ (Back EMF) ซึ่งมีรูปแบบเป็นคลื่นทรงสี่เหลี่ยมคงที่ (Trapezoidal).
- Field-Oriented Control (FOC) หรือบางครั้งเรียกว่า Vector Control คือ วิธีการควบคุมมอเตอร์แบบไฟฟ้ากระแสสลับ (AC Motor) เช่น Permanent Magnet Synchronous Machine (PMSM) หรือ Brushless DC Motor (BLDC) ที่ออกแบบมาเพื่อให้ควบคุมแรงบิด (Torque) และความเร็ว (Speed) ได้อย่างแม่นยำและราบรื่น.
- Pulse Width Modulation (PWM) คือ เทคนิคที่ใช้ในการปรับเปลี่ยนสัญญาณดิจิทัลเพื่อควบคุมกำลังงานที่ส่งไปยังโหลดหรืออุปกรณ์ไฟฟ้า เช่น มอเตอร์, หลอดไฟ, หรืออุปกรณ์อิเล็กทรอนิกส์ โดยการเปลี่ยนความกว้างของพัลส์ในช่วงเวลาหนึ่ง (Duty Cycle) เพื่อควบคุมค่าเฉลี่ยของกำลังไฟฟ้าที่จ่ายออกมานา
- การทำงานของ Commutation Logic ในมอเตอร์ BLDC (Brushless DC Motor) คือ กระบวนการในการควบคุมการจ่ายกระแสไฟฟ้าให้กับชุดลวดในมอเตอร์ โดยให้ทำงานเป็นลำดับขั้นตอนที่เฉพาะเจาะจงเพื่อสร้างสนามแม่เหล็กที่ทำให้โรเตอร์หมุนไปในทิศทางที่ต้องการ
- การควบคุมแบบ Six-Step Commutation (หรือที่เรียกว่า 6-step commutation) เป็นวิธีการควบคุมมอเตอร์ BLDC (Brushless DC Motor) โดยการเปลี่ยนแปลงกระแสไฟฟ้าในชุดลวดมอเตอร์อย่างเป็นขั้นตอนตามตำแหน่งของโรเตอร์ วิธีนี้เป็นที่นิยมในมอเตอร์ที่ใช้สำหรับการควบคุมที่ไม่ต้องการความซับซ้อนมากและให้ประสิทธิภาพในระดับปานกลาง
- Space Vector Modulation (SVM) เป็นเทคนิคที่ใช้ในการควบคุมมอเตอร์ไฟฟ้า โดยเฉพาะในมอเตอร์แบบ Permanent Magnet Synchronous Motor (PMSM) และ Induction Motor (IM) เทคนิคนี้ใช้ในการสร้างสัญญาณ PWM (Pulse Width Modulation) ที่มีประสิทธิภาพสูงเพื่อควบคุมแรงดันที่ไปยังมอเตอร์ในระบบสามเฟส
- การควบคุมแบบ Sensorless คือการควบคุมมอเตอร์ BLDC โดยไม่ใช้เซนเซอร์เพื่อวัดตำแหน่งของโรเตอร์ แต่จะอาศัยการวัดค่าแรงดันหรือกระแสที่เกิดขึ้นในมอเตอร์เพื่อคำนวณตำแหน่งของโรเตอร์ การควบคุมนี้ใช้หลักการของการประมาณค่าตำแหน่งและความเร็วของโรเตอร์จากสัญญาณแรงดันหรือกระแสที่เกิดขึ้นในแต่ละเฟสของมอเตอร์

- การควบคุมแบบใช้ Hall Effect Sensors เป็นเซนเซอร์ที่ใช้หลักการของสนามแม่เหล็กในการตรวจจับตำแหน่งของโรเตอร์ในมอเตอร์ BLDC โดยการตรวจจับสนามแม่เหล็กที่เกิดขึ้นจากแม่เหล็กภายในโรเตอร์ มักใช้ในการควบคุมการหมุนและการสลับเฟสของมอเตอร์ BLDC
- Driving Algorithm หรือ อัลกอริธึมการขับเคลื่อนมอเตอร์ ใน การควบคุมมอเตอร์ หมายถึง วิธีการทางคณิตศาสตร์หรือลอจิกที่ใช้ในการควบคุมการทำงานของมอเตอร์ โดยเฉพาะในระบบที่ต้องการควบคุมความเร็ว ตำแหน่ง และแรงบิดของมอเตอร์อย่างแม่นยำ ซึ่งใช้ในงานที่ต้องการความแม่นยำสูง เช่น มอเตอร์ BLDC (Brushless DC) มอเตอร์สเต็ปเปอร์ และอุปกรณ์อื่นๆ ที่ใช้ในการขับเคลื่อนที่ต้องการกลยุทธ์การควบคุมที่ซับซ้อน
- Zero Crossing หมายถึง จุดที่สัญญาณ (โดยเฉพาะสัญญาณที่มีลักษณะเป็นคลื่นไส้) ข้ามผ่านระดับศูนย์ ซึ่งหมายความว่าสัญญาณจะเปลี่ยนจากค่าบวกไปเป็นค่าลบ หรือจากค่าลบไปเป็นค่าบวกในช่วงเวลาหนึ่ง จุดนี้มีความสำคัญในหลายๆ ด้านในวิศวกรรมไฟฟ้า และการประมวลผลสัญญาณ
- Magnetic Structure ในมอเตอร์ไฟฟ้า เป็นการออกแบบและการจัดเรียงของแม่เหล็กที่ใช้ภายในมอเตอร์เพื่อสร้างสนามแม่เหล็ก ซึ่งสนามแม่เหล็กนี้มีบทบาทสำคัญในการทำให้มอเตอร์ทำงานได้อย่างมีประสิทธิภาพ
- Pole Pairs ในมอเตอร์ไฟฟ้า หมายถึง จำนวนของคู่แม่เหล็กในโรเตอร์ที่สร้างสนามแม่เหล็ก เพื่อให้มอเตอร์สามารถหมุนได้ จำนวนของ pole pairs มีผลต่อการคำนวณความเร็ว การหมุนของมอเตอร์และคุณสมบัติการควบคุม
- Nucleo STM32G474RE: บอร์ดพัฒนาที่ไมโครคอนโทรลเลอร์ STM32 เพื่อช่วยในการประมวลผล และวิเคราะห์ข้อมูลที่ได้จากการทดลอง
- X-NUCLEO-IHM08M1 เป็นบอร์ดขยายสำหรับควบคุมมอเตอร์ที่ออกแบบมาเพื่อใช้ในการควบคุมมอเตอร์ BLDC (Brushless DC Motor) และ PMSM (Permanent Magnet Synchronous Motor) โดยเฉพาะ โดยบอร์ดนี้ใช้ STM32 microcontroller จาก STMicroelectronics ซึ่งช่วยให้สามารถพัฒนาระบบควบคุมมอเตอร์ได้อย่างง่ายดายและรวดเร็ว
- Clock Frequency หมายถึง ความเร็วที่สัญญาณนาฬิกา (Clock signal) ทำงานภายในระบบดิจิทัล โดยส่วนใหญ่จะวัดเป็นหน่วย Hertz (Hz) ซึ่งบ่งบอกจำนวนรอบหรือการสั่นของสัญญาณ ในหนึ่งวินาที ในระบบคอมพิวเตอร์และไมโครคอนโทรลเลอร์ (เช่น STM32) ความถี่ของนาฬิกามีผลต่อความเร็วในการประมวลผลคำสั่ง และการทำงานของอุปกรณ์ต่างๆ ภายในระบบ
- Clock Source หมายถึงแหล่งที่มาของสัญญาณนาฬิกาหรือสัญญาณที่ใช้กำหนดความถี่ของการทำงานในระบบดิจิทัล เช่น ในไมโครคอนโทรลเลอร์ STM32 ความถี่ของนาฬิกาจะถูกกำหนดจากแหล่งสัญญาณที่เรียกว่า "Clock Source" ซึ่งสามารถมีหลายประเภทตามประเภทของการใช้งานและความต้องการของระบบ

- Bus Voltage Sensing คือกระบวนการวัดแรงดันไฟฟ้าในระบบบัส (Bus) เช่นในระบบพลังงานไฟฟ้า หรือระบบควบคุมการจ่ายไฟภายในวงจร ซึ่งมีบทบาทสำคัญในการตรวจสอบและรักษาเสถียรภาพของการทำงานของระบบไฟฟ้า
- Clarke Transformation เป็นเทคนิคสำคัญที่ใช้ในการควบคุมมอเตอร์แบบ Brushless DC Motor (BLDC) และ Permanent Magnet Synchronous Motor (PMSM) เพื่อแปลงเฟสสาม เฟส (a , b , c) ให้กลายเป็นระบบพิกัดสองแกนแบบสติ๊ต (α , β) ซึ่งมีการใช้งานอย่างแพร่หลายใน Field-Oriented Control (FOC) เพื่อปรับปรุงประสิทธิภาพของการควบคุมมอเตอร์
- Park Transformation เป็นกระบวนการทางคณิตศาสตร์ที่ใช้ในการแปลงสัญญาณในระบบพิกัดสองแกนแบบสติ๊ต (α , β) ไปเป็นระบบพิกัดสองแกนแบบหมุน (d - q) โดยใช้กรอบอ้างอิงที่หมุนตามการหมุนของสนามแม่เหล็กในมอเตอร์ BLDC (Brushless DC Motor) หรือ PMSM (Permanent Magnet Synchronous Motor) เทคนิคนี้มีบทบาทสำคัญในการควบคุมแบบ Field-Oriented Control (FOC)
- Duty Cycle ของ BLDC Motor คือสัดส่วนของเวลาที่สัญญาณ PWM (Pulse Width Modulation) อยู่ในสถานะ High (เปิด) ต่อรอบเวลาทั้งหมดของสัญญาณ PWM นั้น โดยปกติจะระบุเป็นเปอร์เซ็นต์ (%)
- Sliding Mode Observer (SMO) เป็นวิธีการสังเกตค่าของสถานะ (state estimation) ในระบบควบคุม โดยเฉพาะอย่างยิ่งใน BLDC Motor สำหรับการควบคุมแบบ Sensorless Control โดยที่ไม่ต้องใช้เซนเซอร์วัดตำแหน่งหรือความเร็วรอบโดยตรง แต่ใช้การคำนวณจากสัญญาณไฟฟ้าแทน
- Phase-Locked Loop (PLL) เป็นเทคนิคในระบบควบคุมที่ใช้สำหรับซิงโครไนซ์สัญญาณเฟสของมอเตอร์กับการอ้างอิง โดยเฉพาะอย่างยิ่งใน BLDC Motor เพื่อการควบคุมตำแหน่งความเร็ว หรือเฟสของโรเตอร์ โดย PLL ทำงานโดยการจับคู่เฟสระหว่างสัญญาณที่วัดได้และสัญญาณอ้างอิงเพื่อรักษาความสมดุลของมอเตอร์
- Extended Kalman Filter (EKF) ในบริบทของ BLDC Motor คืออัลกอริธึมสำหรับการประมาณค่าตำแหน่งของโรเตอร์ ความเร็วรอบ และพารามิเตอร์อื่น ๆ ของมอเตอร์แบบไม่ใช้เซนเซอร์ (Sensorless Control) โดยอาศัยการวัดกระแสและแรงดันร่วมกับการคำนวณผ่านแบบจำลองทางคณิตศาสตร์ของมอเตอร์
- Surface-Mounted Permanent Magnet Synchronous Machine (SPMSM) ของมอเตอร์ BLDC เป็นประเภทของมอเตอร์ซึ่งโครงสร้างแม่เหล็กถาวร (PMSM) ที่แม่เหล็กถาวรถูกติดตั้งอยู่บนพื้นผิวด้านนอกของโรเตอร์ ซึ่งมีผลต่อคุณสมบัติการทำงานและพฤติกรรมของมอเตอร์

- Interior Permanent Magnet (IPM) Rotor ของ BLDC Motor เป็นการออกแบบโรเตอร์ที่แม่เหล็กถาวรถูกฝังอยู่ภายในเนื้อโรเตอร์แทนที่จะติดตั้งบนพื้นผิว ซึ่งมีคุณสมบัติที่โดดเด่นในการเพิ่มประสิทธิภาพและความสามารถของมอเตอร์ในด้านต่าง ๆ
- Air Gap ของ BLDC Motor หมายถึงช่องว่างระหว่างโรเตอร์ (Rotor) และสเตเตอร์ (Stator) ซึ่งเป็นพื้นที่ที่ฟลักซ์แม่เหล็ก (Magnetic Flux) ส่งผ่านจากแม่เหล็กถาวรของโรเตอร์ไปยังชุดลวดของสเตเตอร์ หรือในทิศทางกลับกัน
- Zero Voltage Vector ของ BLDC Motor เป็นสถานะในวงจรควบคุมมอเตอร์ที่อินเวอร์เตอร์ (Inverter) ไม่สร้างแรงดันไฟฟ้าขึ้นได้ ๆ ให้กับมอเตอร์ ทำให้แรงดันไฟฟ้าของเฟสทั้งหมดในสเตเตอร์มีค่าเท่ากับศูนย์ในเชิงสัมพัทธ์ การใช้ Zero Voltage Vector มักพบในเทคนิคการควบคุมแบบ Space Vector Modulation (SVM) เพื่อปรับปรุงประสิทธิภาพการทำงานของมอเตอร์

1.5. นิยามเชิงปฏิบัติการ

- การวิเคราะห์ผลการเรียนรู้หลัก:

หมายถึงการสรุปความเข้าใจของคนผู้จัดทำเกี่ยวกับหลักการทำงานของมอเตอร์ BLDC โดยใช้การทดลอง การเก็บข้อมูลจากเซ็นเซอร์ และการวิเคราะห์ผลผ่านโปรแกรม MATLAB Simulink เพื่อสร้างความเข้าใจในระดับลึก

- การอธิบายหลักการทำงานของ BLDC Motor:

หมายถึงการอธิบายเชิงเทคนิคถึงวิธีที่ BLDC Motor สร้างการเคลื่อนไหว โดยรวมถึงกระบวนการทำงานของ Back EMF, Commutation Logic, และการควบคุมสัญญาณ PWM ซึ่งคนผู้จัดทำต้องสามารถแสดงรายละเอียดในเชิงลึกผ่านการทดลองและการนำเสนอ

- การตั้งค่าใน Motor Workbench 6.3.2:

หมายถึงการกำหนดค่าพารามิเตอร์ทางไฟฟ้า เช่น PWM Frequency, Speed Sensing และพฤติกรรมของมอเตอร์ ผ่านซอฟต์แวร์ที่ใช้ควบคุมมอเตอร์ BLDC เพื่อให้ได้ผลลัพธ์ตามที่ต้องการ

- การตั้งค่า PWM Generation Frequency และ Speed Sensing:

หมายถึงการปรับเปลี่ยนค่าความถี่ของสัญญาณ PWM เพื่อควบคุมพฤติกรรมของมอเตอร์ รวมถึงการตั้งค่าการตรวจสอบความเร็วของมอเตอร์หรือวิธี Sensorless Control เพื่อใช้เป็นข้อมูลสำหรับการวิเคราะห์พฤติกรรมมอเตอร์

- การอธิบาย Electrical และ Mechanical Parameters:

หมายถึงการวิเคราะห์ค่าพารามิเตอร์สำคัญ เช่น ความต้านทาน ความหน่วง แรงบิด ความเรือย และโครงสร้างสนามแม่เหล็ก เพื่อแสดงความเข้าใจถึงผลกระทบที่มีต่อประสิทธิภาพการทำงานของมอเตอร์

- การวิเคราะห์กราฟสัญญาณ 3 เฟส:

หมายถึงการใช้ Oscilloscope และ MATLAB เพื่อจับและวิเคราะห์รูปแบบคลื่นไฟฟ้า 3 เฟส เช่น

Trapezoidal Back EMF หรือ Sinusoidal เพื่อปรับเปลี่ยนพารามิเตอร์ในการควบคุมมอเตอร์ในรูปแบบต่าง ๆ

- การควบคุม Sensorless และ Sensor-based:

หมายถึงการปรับเปลี่ยนการควบคุมแบบมีเซ็นเซอร์ (Hall Effect) และแบบไม่มีเซ็นเซอร์ (Sensorless) เพื่อวิเคราะห์ข้อดีข้อเสียในด้านความแม่นยำและประสิทธิภาพ

- การคำนวณความเร็วจาก Oscilloscope:

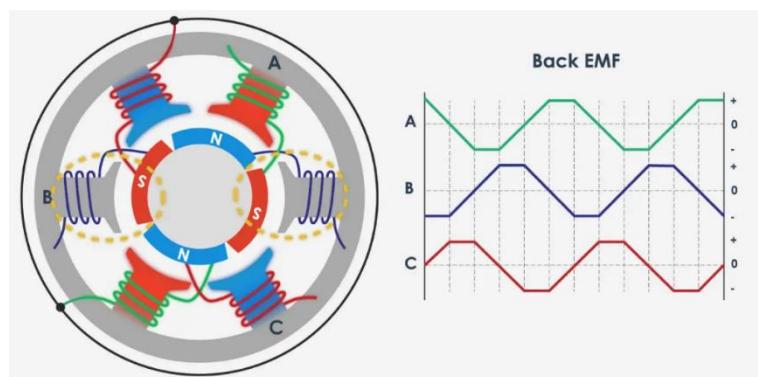
หมายถึงการอ่านค่าความถี่จากการฟลัตต์ที่จับได้จาก Oscilloscope และใช้สมการทางไฟฟ้าในการคำนวณความเร็วเชิงมุ่งของมอเตอร์ BLDC เพื่อใช้ในการวิเคราะห์ผลการทดลอง

1.6. เอกสารและงานวิจัยที่เกี่ยวข้อง

1.6.1. การวิเคราะห์พฤติกรรมและเทคนิคการควบคุมมอเตอร์ BLDC และ PMSM จากสัญญาณ 3 เฟส

1.6.1.1. หลักการทำงานของ BLDC Motor

มอเตอร์กระแสตรงไร้แปรงถ่าน (BLDC Motor) มอเตอร์ BLDC ทำงานบนหลักการสร้างสนามแม่เหล็กหมุน โดยขดลวดในสเตเตอร์จะถูกจ่ายกระแสไฟฟ้าสลับเพื่อทำให้เกิดสนามแม่เหล็กหมุนขึ้นสนามแม่เหล็กหมุนนี้จะดึงดูดและผลักดันแม่เหล็กตัว rotor ให้หมุนตามการสับเปลี่ยนเฟส (Commutation) จะถูกควบคุมด้วยวงจรอิเล็กทรอนิกส์ ทำให้มอเตอร์ทำงานได้อย่างราบรื่นและมีประสิทธิภาพสูง

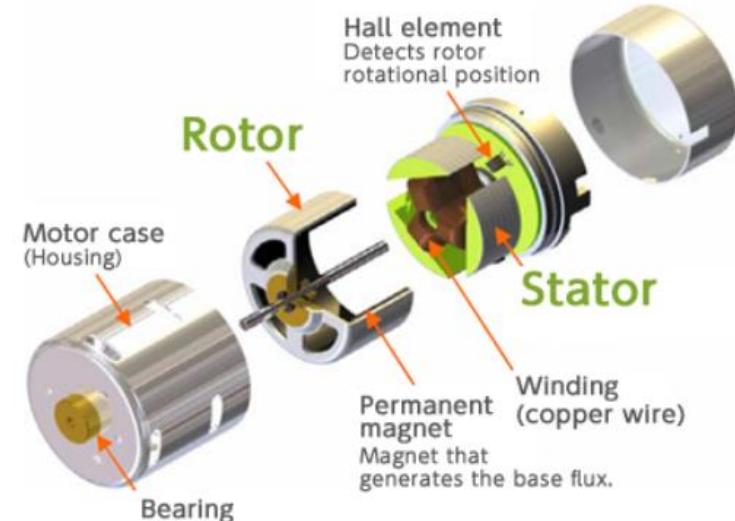


รูปแสดง การทำงานของ BLDC Motor (A B C สามารถแทนกันกับ u v w ได้)

<https://th.twmotor.net/info/how-brushless-dc-motors-work-97376907.html>

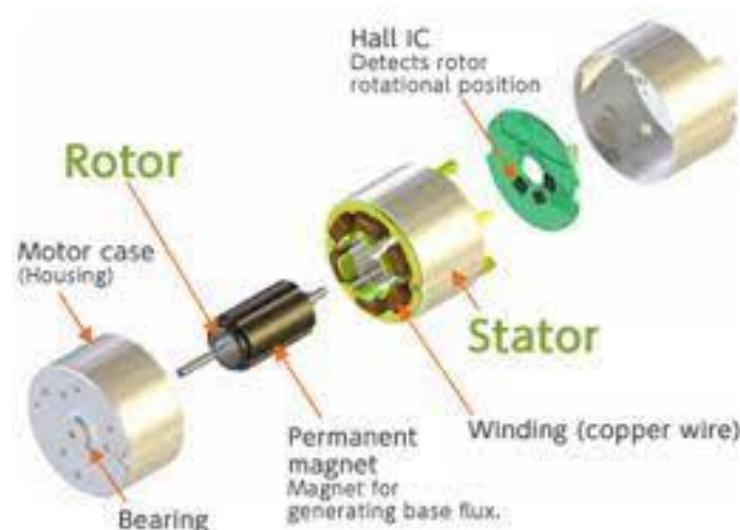
โครงสร้างของมอเตอร์ BLDC มอเตอร์ BLDC ประกอบด้วยส่วนสำคัญดังนี้

- Stator: ประกอบด้วยขดลวดที่พันรอบแกนเหล็ก
- Rotor: ประกอบด้วยแม่เหล็กตัว rotor
- Sensor: เช่น Hall Effect Sensor ใช้ตรวจจับตำแหน่งของ Rotor เพื่อควบคุมการสับเปลี่ยน phase



รูปแสดง โครงสร้างภายในของ BLDC Motor แบบ Outrunner

https://www.9engineer.com/index.php?m=article&a=show&article_id=2606

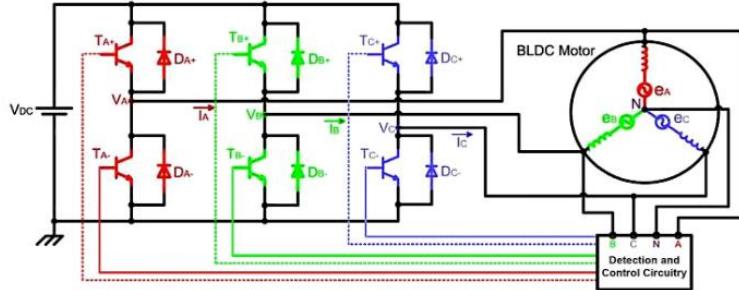


รูปแสดง โครงสร้างภายในของ BLDC Motor แบบ Inrunner

https://www.9engineer.com/index.php?m=article&a=show&article_id=2606

1.6.1.1.1. Sensorless Control (การควบคุมแบบไร้เซ็นเซอร์)

การควบคุมแบบไร้เซ็นเซอร์เป็นวิธีการควบคุมมอเตอร์ BLDC โดยไม่ต้องอาศัยเซ็นเซอร์ในการตรวจจับตำแหน่งของโรเตอร์ วิธีนี้มีข้อดีคือลดต้นทุนและความซับซ้อนของระบบ แต่ต้องอาศัยการประมวลผลสัญญาณที่ซับซ้อนกว่า



รูปแสดง วงจรการควบคุม BLDC Motor แบบ Sensorless

<https://www.allaboutcircuits.com/technical-articles/sensorless-brushless-dc-bldc-motor-control/>

หลักการทำงาน:

- การประมาณตำแหน่ง: ระบบจะทำการประมาณตำแหน่งของโรเตอร์โดยอาศัยการวิเคราะห์สัญญาณ Back EMF ซึ่งเป็นแรงดันไฟฟ้าที่เกิดขึ้นในชล漉ดของมอเตอร์ขณะหมุน
- การสังเกตพารามิเตอร์: นอกจากสัญญาณ Back EMF แล้ว ระบบยังอาจใช้พารามิเตอร์อื่นๆ เช่น กระแสไฟฟ้า หรือแรงดันไฟฟ้าในการประมาณตำแหน่ง
- อัลกอริทึมการควบคุม: ระบบจะใช้ข้อมูลที่ได้จากการประมาณตำแหน่งเพื่อสร้างสัญญาณควบคุมมอเตอร์ให้ทำงานตามต้องการ

1.6.1.1.2. Sensor-based Control (การควบคุมแบบมีเซ็นเซอร์)

การควบคุมแบบมีเซ็นเซอร์เป็นวิธีการควบคุมมอเตอร์ BLDC โดยใช้เซ็นเซอร์ในการตรวจจับตำแหน่งของโรเตอร์โดยตรง วิธีนี้มีความแม่นยำสูง แต่มีต้นทุนสูงกว่า

หลักการทำงาน:

- เซ็นเซอร์: เซ็นเซอร์ที่นิยมใช้คือ Hall Effect Sensor ซึ่งจะตรวจจับสนามแม่เหล็กที่เกิดจากแม่เหล็กถาวรในโรเตอร์
- การสับเปลี่ยนเฟส: สัญญาณจากเซ็นเซอร์จะถูกนำไปใช้ในการควบคุมการสับเปลี่ยนเฟสของมอเตอร์

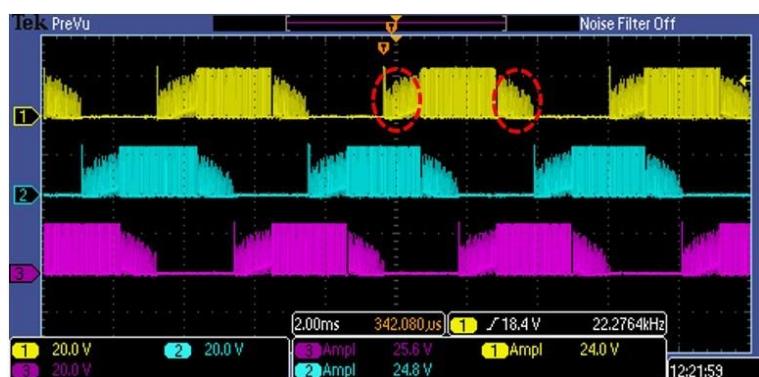
เทคนิคการควบคุมมอเตอร์ BLDC

- Trapezoidal Control: เป็นวิธีการควบคุมที่ง่าย โดยอาศัยสัญญาณ Back EMF รูปสี่เหลี่ยมคงที่ในการสับเปลี่ยนเฟส เหมาะสมสำหรับมอเตอร์ขนาดเล็ก และการใช้งานทั่วไป
- Field-Oriented Control (FOC): เป็นวิธีการควบคุมที่ซับซ้อนกว่า แต่ให้ประสิทธิภาพสูง โดยควบคุมเวกเตอร์ของกระแสไฟฟ้าให้ตั้งฉากกัน เหมาะสมสำหรับมอเตอร์ที่มีสมรรถนะสูง

- Pulse Width Modulation (PWM): เป็นเทคนิคการปรับความกว้างของ พลั๊สเพื่อควบคุมแรงดันและกระแสไฟฟ้าที่จ่ายให้กับมอเตอร์ ช่วยให้การควบคุมมอเตอร์มีความละเอียดและแม่นยำ

1.6.1.2. ลักษณะของ Trapezoidal Back EMF

Trapezoidal Back EMF เป็นแรงดันไฟฟ้าที่เกิดจากการเหนี่ยวนำแม่เหล็กไฟฟ้าภายในมอเตอร์ BLDC (Brushless DC Motor) ซึ่งมีลักษณะเป็นคลื่นรูป สี่เหลี่ยมคงที่ (Trapezoidal Waveform) ลักษณะนี้มีบทบาทสำคัญต่อการควบคุมและการทำงานของมอเตอร์ในระบบขับเคลื่อนที่ต้องการความแม่นยำและประสิทธิภาพสูง



รูปแสดง Trapezoidal Back EMF

<https://e2e.ti.com/support/motor-drivers-group/motor-drivers/f/motor-drivers-forum/827237/drv8308evm-3-phase-bldc-motor--back-emf-measurement>

หลักการเกิด Trapezoidal Back EMF (Electromotive Force) เป็นแรงดันไฟฟ้าที่เกิดขึ้นเมื่อโรเตอร์หมุนผ่านสนามแม่เหล็กของขดลวดในสเตเตอร์ ตามหลักการของการเหนี่ยวนำแม่เหล็กไฟฟ้า (Faraday's Law of Electromagnetic Induction) ลักษณะคลื่นของ Back EMF ใน BLDC Motor ถูกออกแบบให้เป็นสี่เหลี่ยมคงที่ เพื่อให้เหมาะสมกับการควบคุมมอเตอร์แบบ Trapezoidal Control ดูสมบูรณ์ของ Trapezoidal Back EMF

1. รูปคลื่นแรงดัน:

- สัญญาณ Back EMF มีช่วง Flat (แรงดันคงที่) ซึ่งเกิดเมื่อโรเตอร์อยู่ในช่วงที่ตำแหน่งขั้วแม่เหล็ก固定 ตรงกับขดลวด
- ช่วง Rising และ Falling เป็นช่วงที่แรงดันเพิ่มขึ้นและลดลงตามการเปลี่ยนตำแหน่งของโรเตอร์

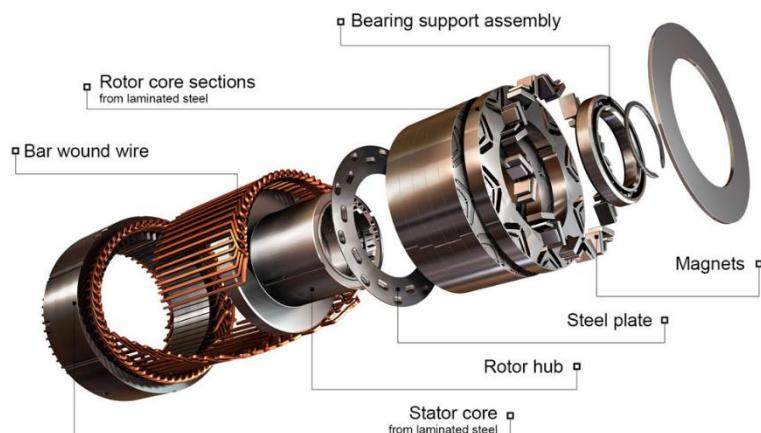
2. ลำดับการทำงานของเฟส:

- คลื่น Back EMF ของเฟส A, B และ C จะทำงานสลับกันตามตำแหน่งการหมุนของโรเตอร์

- การทำงานของแต่ละเฟสครอบคลุมช่วง 120 องศาไฟฟ้า โดยมีการซ้อนทับ (Overlap) ระหว่างเฟส
- ผลต่อแรงบิด:
 - คลื่น Back EMF แบบ Trapezoidal ช่วยให้เกิดแรงบิดที่ต่อเนื่องในช่วง Flat
 - ในช่วง Transition ระหว่างเฟส อาจเกิดแรงบิดกระเพื่อม (Torque Ripple)
 - ความสัมพันธ์กับความเร็วรอบ:
 - แรงดัน Back EMF แปรผันตรงกับความเร็วรอบของโรเตอร์ เมื่อความเร็วรอบเพิ่มขึ้น ค่าแรงดันสูงสุดของ Back EMF จะเพิ่มขึ้นตาม

1.6.1.3. โครงสร้างของ Permanent Magnet Synchronous Machine (PMSM)

Permanent Magnet Synchronous Machine (PMSM) เป็นมอเตอร์ชนิดหนึ่งที่ใช้แม่เหล็กถาวร (Permanent Magnet) ในโรเตอร์และการกระตุนไฟฟ้าจากสเตเตอร์เพื่อสร้างแรงบิด (Torque) ที่มีความต่อเนื่อง โครงสร้างของ PMSM ถูกออกแบบมาเพื่อให้มีประสิทธิภาพสูง การทำงานเงียบ และเหมาะสมกับงานที่ต้องการแรงบิดและความเร็วที่แม่นยำ



รูปแสดง โครงสร้าง Permanent Magnet Synchronous Machine (PMSM)

<https://about-motors.com/motorcontrol/pmsm/>

โครงสร้างหลักของ PMSM

1. สเตเตอร์ (Stator):

- ประกอบด้วยแกนเหล็ก (Laminated Core) และชุดลวดทองแดง (Copper Windings)
- ชุดลวดจัดเรียงเป็นเฟสสาม (Three-Phase Windings) เพื่อสร้างสนามแม่เหล็กหมุน (Rotating Magnetic Field)
- ใช้วัสดุที่ลดการสูญเสียพลังงาน เช่น เหล็กซิลิโคน (Silicon Steel)

2. โรเตอร์ (Rotor):

- ใช้แม่เหล็กถาวรที่ติดตั้งบนหรือฝังอยู่ในโรเตอร์
- Surface-Mounted Rotor: แม่เหล็กถาวรติดตั้งบนพื้นผิวของโรเตอร์
- Interior Permanent Magnet (IPM) Rotor: แม่เหล็กถาวรถูกฝังอยู่ภายในโครงสร้างของโรเตอร์
- โรเตอร์หมุนด้วยความเร็วที่สัมพันธ์กับความถี่ของสนามแม่เหล็กในสเตเตอร์ (Synchronous Speed)

3. ช่องว่างอากาศ (Air Gap):

- พื้นที่ระหว่างสเตเตอร์และโรเตอร์ ออกแบบให้เล็กที่สุดเพื่อเพิ่มประสิทธิภาพของสนามแม่เหล็ก

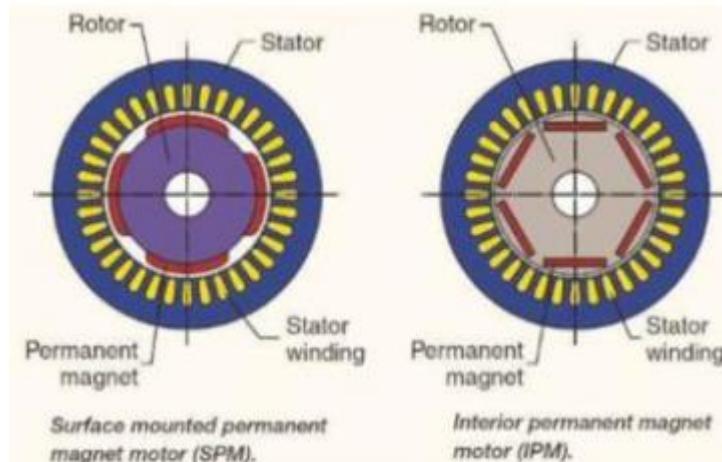
4. แกนหมุน (Shaft):

- เป็นส่วนที่ถ่ายทอดกำลังไปยังโหลด (Load) หรือเครื่องจักรที่ต้องการแรงบิด

หลักการทำงานของ PMSM

PMSM ทำงานโดยสร้างสนามแม่เหล็กหมุนจากขดลวดในสเตเตอร์ สนามแม่เหล็กนี้จะดึงดูดหรือผลักแม่เหล็กถาวรในโรเตอร์ ทำให้เกิดการหมุน ความเร็วของโรเตอร์เท่ากับความเร็วของสนามแม่เหล็กหมุน (Synchronous Speed)

ประเภทของ PMSM



รูปแสดง ประเภทของ PMSM

<https://linktechthailand.com/portfolio/synchronous-motor-pmsm-ev-conversion/>

1. Surface-Mounted PMSM (SPMSM):

- แม่เหล็กถาวรติดตั้งบนพื้นผิวของโรเตอร์
- โครงสร้างง่าย ราคาถูก แต่มีความสามารถในการต้านแรงเหวี่ยงต่ำกว่า

2. Interior PMSM (IPMSM):

- แม่เหล็กฝังอยู่ในโรเตอร์
- มีความทนทานสูงกว่า รองรับแรงเหวี่ยงได้ดี และมีประสิทธิภาพในการสร้างแรงบิดที่ดีกว่า SPMSM

คุณสมบัติและข้อดีของ PMSM

1. ประสิทธิภาพสูง:

- ไม่มีการสูญเสียพลังงานจากการสร้างสนามแม่เหล็กในโรเตอร์

2. แรงบิดสูง:

- สนามแม่เหล็กถาวรในโรเตอร์ให้แรงบิดที่ต่อเนื่องและมีเสถียรภาพ

3. ขนาดกะทัดรัด:

- โครงสร้างที่เล็กและน้ำหนักเบาเหมาะสมกับการใช้งานในพื้นที่จำกัด

4. เสียงรบกวนต่ำ:

- การหมุนของโรเตอร์ที่สมพันธ์กับสนามแม่เหล็กช่วยลดการสั่นสะเทือน

1.6.1.4. การควบคุมแบบ Trapezoidal Control

1.6.1.4.1. หลักการของ Trapezoidal Control

การควบคุมแบบ Trapezoidal ใช้เทคนิค Six-Step Commutation ในการควบคุมกระแสไฟฟ้าผ่านชุดลวดสามเฟส (Phase A, B, C) ของมอเตอร์ โดยหลักการทำงานคือ

- ใช้ Back EMF ในการกำหนดตำแหน่งของโรเตอร์
- กระแสจะถูกจ่ายให้กับสองเฟสที่กำลังทำงานในขณะนั้น และอีกเฟสหนึ่งจะถูกปล่อยให้เป็นสถานะว่าง (Floating)
- ในหนึ่งรอบการหมุนของโรเตอร์ (360 องศาไฟฟ้า) จะมี 6 ช่วงการสลับเฟส (6 Steps) ซึ่งทำให้เกิดแรงบิดอย่างต่อเนื่อง

คุณสมบัติของ Trapezoidal Control

1. การทำงานง่าย:

- ใช้เทคนิคการควบคุมที่ไม่ซับซ้อน
- สามารถใช้งานได้โดยไม่ต้องใช้คอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ที่ซับซ้อน

2. แรงบิดต่อเนื่อง:

- แม้ว่าแรงบิดจะมีการกระเพื่อม (Torque Ripple) ในช่วงเปลี่ยนเฟส แต่ยังคงให้แรงบิดที่เพียงพอสำหรับงานทั่วไป

3. ความแม่นยำต่ำกว่าวิธี FOC:

- วิธีนี้เหมาะสมกับงานที่ไม่ต้องการความแม่นยำสูง เช่น พัดลมหรือปั๊มน้ำ องค์ประกอบในระบบ Trapezoidal Control

1. Hall Effect Sensor:

- ใช้ตรวจจับตำแหน่งของโรเตอร์ เพื่อกำหนดการสลับเฟส

2. วงจรสลับเปลี่ยน (Commutation Circuit):

- ทำหน้าที่ควบคุมการจ่ายกระแสไฟฟ้าไปยังเฟสที่ถูกต้อง

3. ตัวขับเคลื่อน (Driver Circuit):

- จัดการการเปิด-ปิดทรานซิสเตอร์เพื่อควบคุมกระแสไฟฟ้า

4. ไมโครคอนโทรลเลอร์ (ในบางกรณี):

- ใช้ควบคุมลำดับการทำงานของวงจร

1.6.1.5. การควบคุมแบบ Field-Oriented Control (FOC)

Field-Oriented Control (FOC) หรือที่เรียกว่า Vector Control เป็นเทคนิคการควบคุมมอเตอร์ไฟฟ้าที่ซับซ้อนและมีประสิทธิภาพสูง ใช้ในการควบคุมมอเตอร์ AC เช่น PMSM (Permanent Magnet Synchronous Motor) และ BLDC Motor (Brushless DC Motor) วิธีนี้ช่วยให้สามารถควบคุมแรงบิด (Torque) และฟลักซ์แม่เหล็ก (Flux) ได้อย่างอิสระ ส่งผลให้แรงบิดเรียบ (Smooth Torque) และการทำงานเงียบกว่าเทคนิคการควบคุมอื่นๆ

หลักการของ Field-Oriented Control (FOC)

FOC ใช้หลักการแปลงระบบสามเฟสของมอเตอร์ให้เป็นระบบแกนสองมิติเพื่อให้ควบคุมแรงบิดและฟลักซ์ได้เหมือนมอเตอร์กระแสตรง (DC Motor) วิธีการทำงานหลักคือ:

1. การแปลง Clarke และ Park:

สัญญาณสามเฟส (u, v, w) ถูกแปลงเป็นระบบสองแกน (a, β) ด้วย Clarke Transformation จากนั้น แปลงระบบ (a, β) เป็นระบบ (d, q) ซึ่งอิงตามการหมุนของฟลักซ์แม่เหล็กโรเตอร์ ด้วย Park Transformation

2. การควบคุมกระแสในแกน $d - q$:

- แกน d : ควบคุมฟลักซ์แม่เหล็กในมอเตอร์
- แกน q : ควบคุมแรงบิดที่สร้างจากมอเตอร์
- ควบคุมกระแสในแต่ละแกนผ่านตัวควบคุม PI (Proportional-Integral Controller)

3. การแปลงกลับ:

- สัญญาณควบคุมจากระบบ $d - q$ ถูกแปลงกลับไปเป็นระบบสามเฟส (u, v, w) ด้วย Inverse Park Transformation และ Space Vector Modulation (SVM) เพื่อสร้างสัญญาณ PWM สำหรับขั้มนอเตอร์

ขั้นตอนการทำงานของ FOC

1. อ่านค่า กระแสสามเฟส (i_a, i_b, i_c) จากมอเตอร์
2. ใช้ Clarke Transformation เพื่อแปลงเป็นระบบสองแกน (i_α, i_β)
3. ใช้ Park Transformation เพื่อแปลงเป็นระบบ $d - q$ โดยอิงตามตำแหน่งโรเตอร์
4. ใช้ตัวควบคุม PI เพื่อควบคุมกระแสในแกน $d - q$:
 - กระแส d : ควบคุมฟลักซ์แม่เหล็ก
 - กระแส q : ควบคุมแรงบิด
5. ใช้ Inverse Park Transformation และ Space Vector Modulation (SVM) เพื่อสร้างสัญญาณ PWM สำหรับขั้มนอเตอร์

1.6.1.6. เทคนิค Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) หรือการมอดูเลตความกว้างของพัลส์ เป็นเทคนิคที่ใช้ควบคุมแรงดันไฟฟ้าหรือกระแสไฟฟ้าในระบบอิเล็กทรอนิกส์ โดยการสร้างสัญญาณพัลส์ที่มีความถี่คงที่แต่ปรับความกว้างของพัลส์ (Duty Cycle) ให้เหมาะสมกับความต้องการ เทคนิคนี้ถูกนำไปใช้อย่างแพร่หลายในงานควบคุมมอเตอร์ไฟฟ้า เช่น BLDC Motor และ PMSM เพื่อปรับความเร็ว แรงบิด และประสิทธิภาพของมอเตอร์

หลักการทำงานของ PWM

1. สัญญาณ PWM เป็นสัญญาณดิจิทัลที่มีรูปแบบพัลส์ (Pulse) ซึ่งสลับกันระหว่างสถานะเปิด (On) และปิด (Off)
2. Duty Cycle คือสัดส่วนของช่วงเวลาที่สัญญาณอยู่ในสถานะเปิด เทียบกับเวลา (Period) ทั้งหมด คำนวณได้จากสมการ:

$$\text{Duty Cycle}(\%) = \frac{\text{เวลาที่ On}}{\text{เวลาที่รวมใน 1 รอบ(Period)}} \times 100$$

3. การปรับ Duty Cycle จะเปลี่ยนค่าเฉลี่ยของแรงดันไฟฟ้าที่จ่ายให้กับโหลด เช่น มอเตอร์

คุณสมบัติสำคัญของ PWM

1. ความถี่ของสัญญาณ:
 - ความถี่ของ PWM ต้องเหมาะสมกับการทำงานของอุปกรณ์ เช่น มอเตอร์ BLDC นักใช้ PWM ที่ความถี่ตั้งแต่ 10 kHz ถึง 20 kHz
2. Duty Cycle:
 - Duty Cycle 0% หมายถึงไม่มีกระแสในหนึ่งรอบเวลา
 - Duty Cycle 100% หมายถึงจ่ายกระแสในหนึ่งรอบเวลา
 - การปรับ Duty Cycle ในช่วง 0-100% ช่วยควบคุมกระแสในอุปกรณ์ที่ส่งไปยังโหลดได้อย่างยืดหยุ่น

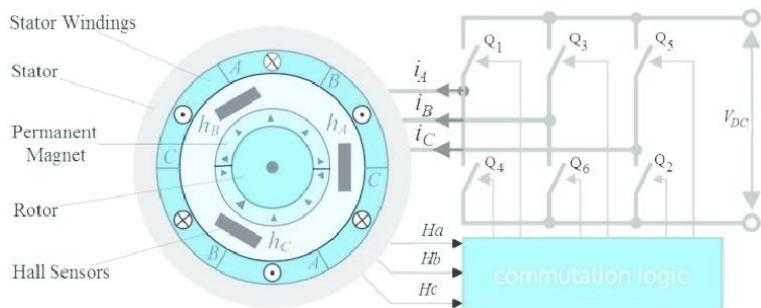
ประเภทของ PWM

1. Single-Pulse PWM:
 - ใช้พัลส์เพียงพัลส์เดียวในหนึ่งรอบเวลา
2. Multiple-Pulse PWM:
 - ใช้พัลส์หลายพัลส์ในหนึ่งรอบเวลาเพื่อปรับปรุงประสิทธิภาพ
3. Sinusoidal PWM (SPWM):
 - รูปแบบ PWM ที่สร้างพัลส์ตามรูปคลื่นไอน์ เพื่อปรับกระแสในไฟฟ้าให้เรียบและลด Harmonics
4. Space Vector PWM (SVPWM):
 - เทคนิค PWM ขั้นสูงที่เหมาะสมสำหรับมอเตอร์สามเฟส เช่น PMSM และ BLDC Motor

1.6.1.7. การทำงานของ Commutation Logic

Commutation Logic คือกระบวนการควบคุมการสลับกระแสไฟฟ้าในขดลวดของมอเตอร์ BLDC (Brushless DC Motor) หรือ PMSM (Permanent Magnet Synchronous Motor) เพื่อให้เกิดแรงบิดที่เหมาะสมและต่อเนื่องในระหว่างการหมุนของโรเตอร์ กระบวนการนี้ถือเป็นหัวใจสำคัญของการทำงานของมอเตอร์ไฟฟ้าไร้แปรงถ่าน เนื่องจากไม่มีแปรงถ่านในการสลับกระแสไฟฟ้าแบบมอเตอร์ DC ทั่วไป Commutation Logic จะทำหน้าที่นี้แทนด้วยอุปกรณ์อิเล็กทรอนิกส์

หลักการทำงานของ Commutation Logic



รูปแสดง วงจร Commutation Logic

https://www.researchgate.net/figure/BLDC-Motor-with-Commutation-Logic_fig1_308571505

1. ตรวจจับตำแหน่งของโรเตอร์:

- ตำแหน่งของโรเตอร์ถูกตรวจจับโดยใช้เซ็นเซอร์ เช่น Hall Effect Sensor หรือวิธีการตรวจจับแบบไร้เซ็นเซอร์ (Sensorless Detection) เช่น การวิเคราะห์ Back EMF
- ตำแหน่งโรเตอร์ช่วยกำหนดว่าควรจ่ายกระแสไฟฟ้าไปยังเฟสใดของมอเตอร์

2. สลับเฟส (Phase Switching):

- กระแสไฟฟ้าจะถูกจ่ายให้ขดลวดสองเฟสที่เหมาสมในแต่ละช่วงของการหมุนโรเตอร์
- ในมอเตอร์ BLDC หนึ่งรอบของการหมุนโรเตอร์ (360 องศาไฟฟ้า) จะถูกแบ่งออกเป็น 6 ช่วงการทำงาน (Six-Step Commutation)

3. ควบคุมลำดับการจ่ายกระแส:

- ลำดับการจ่ายกระแสถูกควบคุมโดยวงจรหรือซอฟต์แวร์ที่เรียกว่า Commutation Logic Circuit
- กระแสในแต่ละเฟสจะทำงานตามลำดับ เพื่อสร้างแรงบิดที่สอดคล้องกับตำแหน่งของโรเตอร์

ประเภทของ Commutation Logic

1. การทำงานด้วยเซ็นเซอร์ (Sensor-Based Commutation):

- ใช้ Hall Effect Sensor ติดตั้งในมอเตอร์เพื่อบอกตำแหน่งของโรเตอร์
- ตำแหน่งที่ได้จะถูกใช้ในการกำหนดลำดับการจ่ายกระแส
- วิธีนี้มีความแม่นยำและเหมาะสมสำหรับงานที่ต้องการแรงบิดคงที่

2. การทำงานแบบไร้เซ็นเซอร์ (Sensorless Commutation):

- ใช้การวิเคราะห์ Back EMF หรือพารามิเตอร์อื่นๆ ของมอเตอร์ เช่น กระแสไฟฟ้า เพื่อกำหนดตำแหน่งของโรเตอร์
 - วิธีนี้ช่วยลดต้นทุนและเหมาะสมสำหรับงานที่ไม่ต้องการความแม่นยำสูง
- กระบวนการทำงานในแต่ละขั้นตอน
- การอ่านค่าตำแหน่ง:เซ็นเซอร์หรือระบบวิเคราะห์ตำแหน่งจะส่งข้อมูลตำแหน่งของโรเตอร์ไปยัง Commutation Logic Circuit
 - การประมวลผล:Commutation Logic Circuit วิเคราะห์ตำแหน่งที่ได้รับและกำหนดเฟสที่ต้องจ่ายกระแส
 - การจ่ายกระแส:กระแสไฟฟ้าถูกส่งไปยังขดลวดสองเฟสที่เหมาะสมในช่วงเวลาใดเวลาหนึ่ง และอิกหนึ่งเฟสจะถูกปล่อยเป็นสถานะว่าง (Floating)
 - การสลับกระแส:เมื่อโรเตอร์เปลี่ยนตำแหน่ง ลำดับการจ่ายกระแสจะเปลี่ยนไปตามตำแหน่งใหม่ เพื่อให้แรงบิดต่อเนื่อง

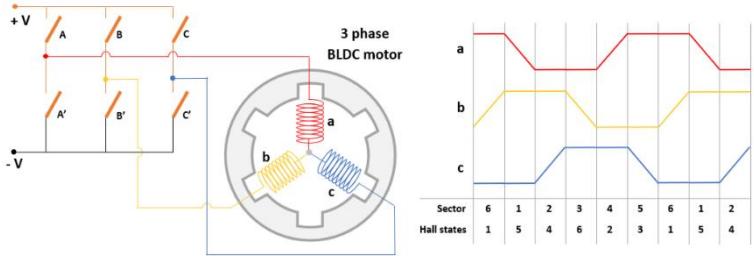
องค์ประกอบสำคัญของ Commutation Logic

- เซ็นเซอร์ตรวจจับตำแหน่ง:เช่น Hall Effect Sensor หรือ Back EMF Detection
- วงจรสลับเฟส:ใช้ทรานซิสเตอร์ (MOSFET หรือ IGBT) เป็นสวิตช์ในการควบคุมการจ่ายกระแส
- ไมโครคอนโทรลเลอร์ (Microcontroller):ใช้สำหรับประมวลผลข้อมูลตำแหน่งและควบคุมวงจรสลับเฟส
- อัลกอริทึมการควบคุม:กำหนดลำดับการจ่ายกระแสในแต่ละช่วงของตำแหน่งโรเตอร์

1.6.1.8. การควบคุมแบบ Six-Step Commutation

การควบคุมแบบ Six-Step Commutation เป็นเทคนิคการควบคุมที่ใช้กับ BLDC Motor (Brushless DC Motor) โดยการจ่ายกระแสไฟฟ้าสลับกันในขดลวดสามเฟสของมอเตอร์ตามตำแหน่งของโรเตอร์ เทคนิคนี้เรียกว่า Trapezoidal Commutation เนื่องจากสัญญาณ Back EMF ของมอเตอร์มีลักษณะเป็นคลื่นรูปสี่เหลี่ยมคง猛 วิธีนี้มีความเรียบง่ายและเหมาะสมสำหรับการควบคุมมอเตอร์ในระบบที่ไม่ต้องการความแม่นยำสูง

หลักการของ Six-Step Commutation



รูปแสดง วงจร Six-Step Commutation (a b c สามารถแทนกันกับ u v w ได้)

<https://www.mathworks.com/help/mcb/ref/sixstepcommutation.html>

Six-Step Commutation ทำงานโดยสลับกระแสไฟฟ้าในขดลวดสามเฟส (u, v, w) ในลำดับที่แน่นอน โดยมีหลักการดังนี้:

1. การแบ่งรอบการหมุน:

- หนึ่งรอบของการหมุนโรเตอร์ (360 องศาไฟฟ้า) ถูกแบ่งออกเป็น 6 ขั้นตอน (Steps)
- ในแต่ละขั้นตอน มีสองเฟสที่จ่ายกระแส และอีกหนึ่งเฟสที่ถูกปล่อยให้ว่าง (Floating)

2. การกำหนดลำดับการจ่ายกระแส:

- กระแสไฟฟ้าถูกสลับตามลำดับที่สัมพันธ์กับตำแหน่งของโรเตอร์
- ตำแหน่งของโรเตอร์สามารถตรวจจับได้โดยใช้ Hall Effect Sensor หรือวิธีการตรวจจับแบบไรเซ็นเซอร์ (Sensorless)

3. การสร้างแรงบิด:

- การสลับกระแสในลำดับที่เหมาะสมทำให้แรงบิดที่เกิดขึ้นในมอเตอร์มีความต่อเนื่อง และเพียงพอสำหรับการทำงาน

กระบวนการทำงานของ Six-Step Commutation

1. อ่านค่าตำแหน่งของโรเตอร์:

- ตำแหน่งของโรเตอร์ถูกตรวจจับผ่าน Hall Effect Sensor หรือ Back EMF
- ข้อมูลตำแหน่งถูกส่งไปยังวงจรควบคุม

2. กำหนดลำดับการจ่ายกระแส:

- ระบบใช้ข้อมูลตำแหน่งของโรเตอร์เพื่อกำหนดว่าขดลวดเฟสใดควรจ่ายกระแส
- มีการกำหนดลำดับการจ่ายกระแสที่แน่นอนใน 6 ขั้นตอน

3. จ่ายกระแสและสลับเฟส:

- กระแสไฟฟ้าถูกจ่ายไปยังเฟสที่กำหนดในแต่ละขั้นตอน โดยอีกเฟสจะถูกปล่อยเป็นสถานะว่าง

4. วนซ้ำกระบวนการ:

- เมื่อโตร์มหุน ครบ 6 ขั้นตอน จะกลับไปเริ่มต้นใหม่ และกระบวนการทำซ้ำ ต่อเนื่องไปเรื่อยๆ

1.6.1.9. เทคนิค Space Vector Modulation (SVM)

Space Vector Modulation (SVM) เป็นเทคนิคการmodulateขั้นสูงที่ใช้ในระบบควบคุมมอเตอร์ไฟฟ้า เช่น PMSM (Permanent Magnet Synchronous Motor) และ BLDC Motor เทคนิคนี้ช่วยเพิ่มประสิทธิภาพการควบคุมมอเตอร์ โดยเฉพาะในระบบสามเฟส (Three-Phase System) ด้วยการสร้างสัญญาณที่สอดคล้องกับตำแหน่งของเวกเตอร์แรงดันไฟฟ้า หลักการของ Space Vector Modulation

Space Vector Modulation ใช้การคำนวณทางคณิตศาสตร์เพื่อกำหนดตำแหน่งของเวกเตอร์แรงดันไฟฟ้าในระบบสามเฟส โดยมีขั้นตอนหลักดังนี้:

1. การสร้างเวกเตอร์แรงดัน (Voltage Space Vector):

- สัญญาณสามเฟส (V_u, V_v, V_w) ถูกแปลงเป็นระบบสองแกน (V_α, V_β) ด้วย Clarke Transformation
- ระบบสองแกนนี้แสดงตำแหน่งของแรงดันไฟฟ้าในรูปแบบเวกเตอร์ในระนาบสองมิติ

2. การแบ่งพื้นที่ (Space Vector Sectors):

- วงจรอินเวอร์เตอร์สามเฟสสามารถสร้างสถานะการสลับ (Switching States) ได้ทั้งหมด 8 สถานะ:
 - 6 สถานะสร้างแรงดันเวกเตอร์ที่ไม่เป็นศูนย์
 - 2 สถานะเป็นศูนย์ (Zero Voltage Vector)
- พื้นที่ของแรงดันไฟฟ้าถูกแบ่งออกเป็น 6 โซน (Sectors) ตามตำแหน่งของเวกเตอร์แรงดันในระบบแกน $\alpha - \beta$

3. การสร้างสัญญาณ PWM:

- ในแต่ละช่วงเวลา เวกเตอร์แรงดันเป้าหมายจะถูกประมาณด้วยการเปิด-ปิดสวิตช์อินเวอร์เตอร์ในลำดับที่เหมาะสม
- สัญญาณ PWM ถูกสร้างขึ้นโดยการรวมสถานะที่ไม่เป็นศูนย์และสถานะศูนย์เพื่อให้ได้ค่าแรงดันเฉลี่ยที่ใกล้เคียงกับแรงดันเป้าหมาย

ขั้นตอนการทำงานของ SVM

1. การวิเคราะห์แรงดันสามเฟส:

- อ่านค่ากำลังแรงดัน (V_u, V_v, V_w) ที่ต้องการจ่ายไปยังมอเตอร์

2. การแปลงแรงดัน:

- ใช้ Clarke Transformation เพื่อแปลงแรงดันสามเฟสเป็น (V_α, V_β)

3. การรับปุ่ม:

- กำหนดตำแหน่งของเวกเตอร์แรงดันใน 6 โซน

4. การคำนวณเวลาสวิตช์:

- คำนวณเวลาที่แต่ละสถานะสวิตช์จะทำงาน (Duty Cycle) เพื่อสร้างแรงดันเฉลี่ยตามเป้าหมาย

5. การสร้างสัญญาณ PWM:

- ใช้ข้อมูลเวลาสวิตช์ในการสร้างสัญญาณ PWM สำหรับควบคุมสวิตช์ของอินเวอร์เตอร์

1.6.1.10. การควบคุมแบบ Sensorless และการใช้ Hall Effect Sensor

การควบคุมมอเตอร์ BLDC (Brushless DC Motor) อาศัยข้อมูลตำแหน่งของโรเตอร์เพื่อกำหนดการสลับกระแสในชุดลวดสามเฟส ซึ่งสามารถทำได้ทั้งแบบมีเซ็นเซอร์ (Hall Effect Sensor) และแบบไร้เซ็นเซอร์ (Sensorless Control) การเลือกใช้ขึ้นอยู่กับข้อกำหนดของระบบ เช่น ความแม่นยำ ต้นทุน และสภาพแวดล้อมการใช้งาน

การควบคุมแบบ Sensorless

การควบคุมแบบ Sensorless หมายถึงการควบคุมมอเตอร์ BLDC โดยไม่ใช้เซ็นเซอร์ตรวจจับตำแหน่งของโรเตอร์ วิธีนี้ช่วยลดต้นทุนและความซับซ้อนของฮาร์ดแวร์ แต่ต้องอาศัยการประมวลผลที่ซับซ้อนกว่าเพื่อประมาณตำแหน่งของโรเตอร์

หลักการทำงานของ Sensorless Control

1. การวิเคราะห์ Back EMF:

- ใช้สัญญาณ Back EMF (Electromotive Force) ที่เกิดขึ้นในชุดลวดเมื่อโรเตอร์หมุน เพื่อตรวจจับตำแหน่งของโรเตอร์
- Back EMF จะสัมพันธ์กับตำแหน่งของโรเตอร์และเฟสที่กำลังทำงาน

2. การคำนวณและประมวลผล:

- ระบบคำนวณตำแหน่งของโรเตอร์โดยวิเคราะห์ค่ากระแสไฟฟ้าและแรงดันในชุดลวด
- อัลกอริทึมที่นิยมใช้ เช่น Sliding Mode Observer (SMO), Phase-Locked Loop (PLL), หรือ Extended Kalman Filter (EKF)

3. การควบคุมแบบไร้เซ็นเซอร์ที่ความเร็วต่ำ:

- ที่ความเร็วต่ำ Back EMF มีค่าน้อย จึงใช้พารามิเตอร์อื่น เช่น กระแสไฟฟ้าหรือแรงดันไฟฟ้า แทนในการประมาณตำแหน่ง

การควบคุมโดยใช้ Hall Effect Sensor

Hall Effect Sensor เป็นเซ็นเซอร์ที่ติดตั้งในมอเตอร์ BLDC เพื่อตรวจจับตำแหน่งของโรเตอร์ เซ็นเซอร์เหล่านี้ทำงานโดยการตรวจจับการเปลี่ยนแปลงของสนามแม่เหล็กในโรเตอร์

หลักการทำงานของ Hall Effect Sensor

1. การตรวจจับสนามแม่เหล็ก:

- Hall Effect Sensor จะตรวจจับสนามแม่เหล็กที่เปลี่ยนแปลงตามการหมุนของโรเตอร์
- สนามแม่เหล็กนี้ถูกแปลงเป็นสัญญาณไฟฟ้า (High หรือ Low) เพื่อแสดงตำแหน่งของโรเตอร์

2. การกำหนดลำดับการทำงานของเฟส:

- สัญญาณจาก Hall Sensor ใช้กำหนดลำดับการสลับเฟส (Commutation) ของขดลวด
- ตัวอย่าง: หาก Hall Sensor ส่งสัญญาณ 100100100, 110110110, 010010010, 011011011, 001001001, 101101101 ระบบจะสลับเฟสตามลำดับที่กำหนด

3. การประมาณสัญญาณ:

- ไมโครคอนโทรลเลอร์จะใช้สัญญาณจาก Hall Sensor เพื่อกำหนดว่าเฟสใดควรจ่ายกระแสในแต่ละช่วงการหมุน

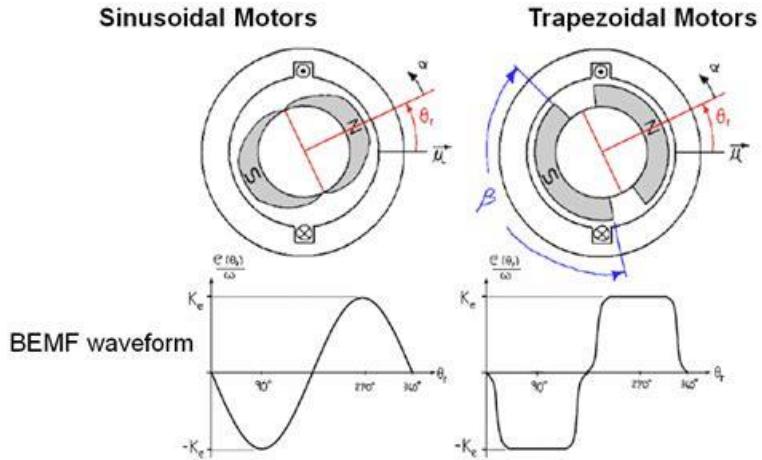
•

1.6.1.11. ความแตกต่างของการควบคุม 6-Step Control และ FOC

การควบคุมมอเตอร์ BLDC มีสองวิธิหลัก ได้แก่ 6-Step Control ซึ่งเป็นวิธีการพื้นฐาน และ FOC ซึ่งเป็นวิธีการที่ซับซ้อนกว่า 6-Step Control เป็นวิธีการที่ง่าย โดยอาศัยการสับเปลี่ยนเฟสของมอเตอร์ตามลำดับที่กำหนดไว้ ทำให้เกิดแรงบิดในการขับเคลื่อนมอเตอร์ เวกเตอร์กระแสในขนาดและทิศทางคงที่ แต่มีข้อจำกัดในเรื่องของความแม่นยำในการควบคุมแรงบิดและความเร็วรวมถึงประสิทธิภาพที่ค่อนข้างต่ำ เนื่องจากกระบวนการใช้งานที่ไม่ต้องการความแม่นยำสูง ในทางกลับกัน FOC เป็นวิธีการควบคุมที่ใช้เทคนิค Space Vector Modulation (SVM) ในการจัดการเวกเตอร์กระแสไฟฟ้า ทำให้สามารถควบคุมแรงบิดและความเร็วได้อย่างแม่นยำ มีประสิทธิภาพสูง และแรงสั่นสะเทือนต่ำ FOC สามารถทำงานได้ทั้งแบบ Sensorless (ไม่ใช้เซ็นเซอร์) หรือใช้ Hall Effect Sensor ในการตรวจจับตำแหน่งของโรเตอร์ เมื่อ FOC จะมีข้อดีหลายประการ แต่ก็มี

ความซับซ้อนและมีต้นทุนสูงกว่า 6-Step Control เนื่องจากกระบวนการที่ต้องการความแม่นยำสูง และประสิทธิภาพที่ดี เช่น ในอุตสาหกรรมหุ่นยนต์หรือเครื่องจักรที่มีความซับซ้อน

1.6.1.12. ความแตกต่างระหว่าง Trapezoidal และ Sinusoidal



รูปภาพนี้แสดง ความแตกต่างระหว่างมอเตอร์ที่ควบคุมด้วย Sinusoidal และ Trapezoidal โดยเน้นที่รูปคลื่นของ Back EMF และโครงสร้างการทำงานของมอเตอร์ BLDC

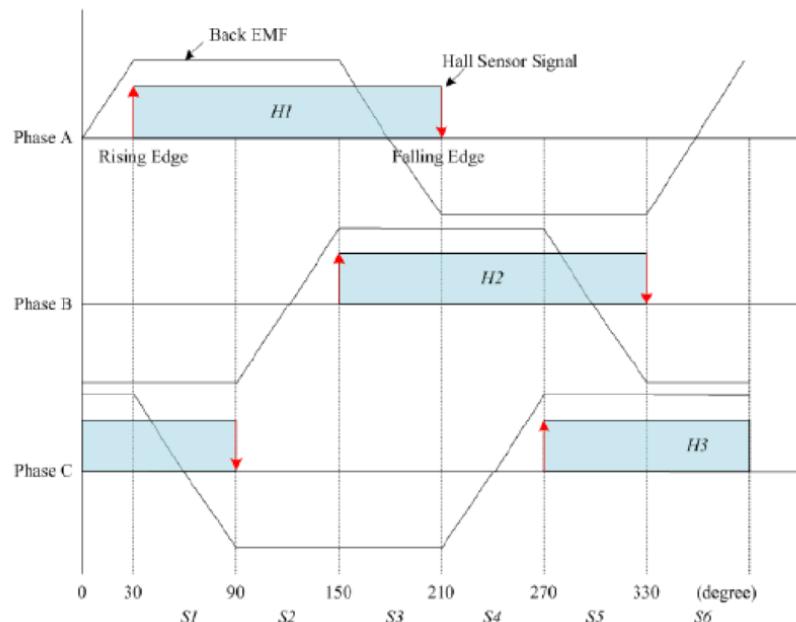
<https://electronics.stackexchange.com/questions/347683/how-can-one-achieve-sinusoidal-back-emf-in-pmsm-and-trapezoidal-back-emf-in-bldc>

สัญญาณ Trapezoidal และ Sinusoidal ในมอเตอร์ BLDC แตกต่างกันที่รูปคลื่นของกระแสไฟฟ้าที่จ่ายให้มอเตอร์ โดย Trapezoidal มีลักษณะคลื่นเป็นสี่เหลี่ยมคงหมุน ควบคุมง่าย แต่มีประสิทธิภาพต่ำกว่าและเกิด Torque Ripple รวมถึงเสียงรบกวนมากกว่า ส่วน Sinusoidal มีคลื่นกระแสเรียบเนียน ให้ประสิทธิภาพสูง แรงบิดเรียบ และเสียงรบกวนน้อย การเลือกใช้ขึ้นอยู่ กับแอปพลิเคชัน เช่น Sinusoidal เหมาะสมกับงานที่ต้องการความเรียบและประสิทธิภาพสูง ส่วน Trapezoidal เหมาะสมกับงานที่เน้นความเรียบง่ายและไม่ซับซ้อน

1.6.1.13. ความสัมพันธ์ระหว่าง 6-Step Control และ PWM

6-Step Control เป็นวิธีการควบคุมมอเตอร์ BLDC แบบพื้นฐานที่อาศัยการสลับกระแสไฟฟ้าในชุดลวดสามเฟสตามลำดับที่กำหนดไว้ โดย PWM (Pulse Width Modulation) เข้ามาช่วยปรับความกว้างของ脉冲ไฟฟ้าที่จ่ายให้กับชุดลวดเพื่อควบคุมแรงดันเฉลี่ยและทำให้สามารถปรับแรงบิดและความเร็วของมอเตอร์ได้อย่างละเอียด ทั้ง PWM และ 6-Step Control ทำงานร่วมกันเพื่อสร้างสนามแม่เหล็กหมุนที่ผลักดันให้มอเตอร์หมุนตามที่ต้องการ โดย PWM จะทำหน้าที่ปรับแรงดันไฟฟ้าในแต่ละขั้นตอนของ 6-Step Control ซึ่งเป็นกลไกหลักในการควบคุมการทำงานของมอเตอร์ BLDC

1.6.1.14. ความสัมพันธ์ระหว่าง PWM และ Frequency Output (Back EMF และ Hall Sensor)



รูปภาพแสดงความสัมพันธ์ระหว่างแรงดันย้อนกลับ (Back EMF), สัญญาณจาก Hall Sensor (H1, H2, H3), และการเปลี่ยนแปลงตำแหน่งเชิงมุมของมอเตอร์ BLDC

https://www.researchgate.net/figure/Figure-3-4-Back-EMF-and-hall-effect-sensor-signals-11_fig13_262933380

PWM (Pulse Width Modulation) ใช้ในการควบคุมมอเตอร์ BLDC โดยการปรับความกว้างของพัลส์ไฟฟ้าที่จ่ายให้ขาด漉ด เพื่อควบคุมแรงดันไฟฟ้าเฉลี่ยที่ส่งไปยังมอเตอร์ ส่งผลโดยตรงต่อแรงบิดและความเร็วรอบของมอเตอร์ ความเร็วรอบที่เปลี่ยนแปลงจะส่งผลต่อกำลังที่ของ Back EMF (แรงดันไฟฟ้าที่เกิดจากการตัดสนามแม่เหล็ก) และสัญญาณจาก Hall Sensor ซึ่งใช้ตรวจจับตำแหน่งมอเตอร์ ทั้งสองสัญญาณมีความถี่สัมพันธ์โดยตรงกับความเร็วรอบ ดังนั้น การปรับ PWM ไม่เพียงควบคุมความเร็วรอบของมอเตอร์ แต่ยังส่งผลต่อกำลังที่ของ Back EMF และสัญญาณ Hall Sensor โดยตรงด้วย

1.6.1.15. ความแตกต่างระหว่าง Feedback จาก Back EMF และ Hall Sensor

Feedback จาก Back EMF ใช้ในการควบคุมแบบไร้เซ็นเซอร์ (Sensorless Control) โดยวิเคราะห์แรงดันไฟฟ้าที่เกิดจากการเหนี่ยวนำแม่เหล็กไฟฟ้าในขาด漉ดเมื่อ)motoร์หมุน หมายความรับความเร็วรอบสูงที่ Back EMF มีค่าสูงพอสำหรับการตรวจจับ แต่มีความซับซ้อนในกระบวนการประมวลผลและไม่เหมาะสมกับความเร็วต่ำ ในขณะที่ Hall Sensor ให้สัญญาณดิจิทัลโดยตรงจากตำแหน่งของมอเตอร์ หมายความรับการควบคุมที่ต้องการความแม่นยำและสามารถใช้

งานได้ในทุกช่วงความเร็ว แต่ต้องการ shaft driver เพิ่มเติม ทำให้มีต้นทุนสูงกว่า Back EMF Feedback

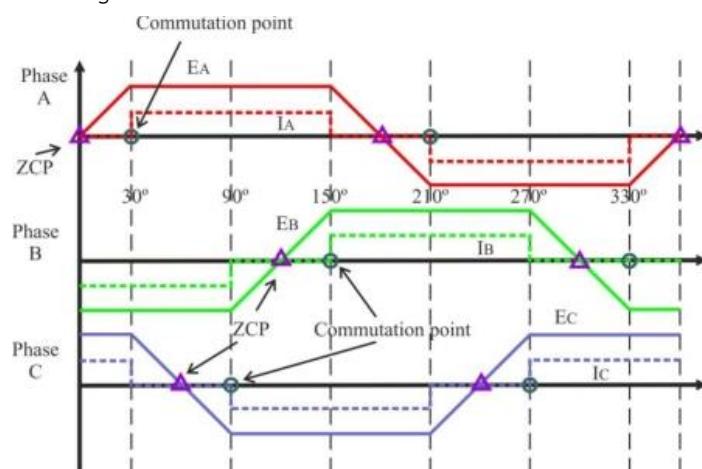
1.6.1.16. การจัดการกับ Feedback ของ Sensorless Control เพื่อหาตำแหน่งเทียบกับ Feedback ของ Sensor-based Control บน ESC และ MCU

ระบบ Sensor-based Control จะใช้เซ็นเซอร์เช่น Hall Effect sensors หรือ Encoder เพื่อตรวจสอบตำแหน่งของโรเตอร์และให้ข้อมูล Feedback กับ ESC หรือ MCU ในการปรับการจ่ายพลังงานไปยังมอเตอร์เพื่อควบคุมความเร็วและทิศทางได้อย่างแม่นยำ ในขณะที่ Sensorless Control จะไม่ใช้เซ็นเซอร์ใดๆ แต่จะใช้เทคนิค เช่น Back EMF (Electromotive Force) หรือการประมาณตำแหน่งโรเตอร์จากการตรวจจับความต้านทานในการหมุนของมอเตอร์ หรือการคำนวณจากข้อมูลการหมุนรอบเพื่อให้ Feedback ที่จำเป็นสำหรับการควบคุมมอเตอร์ การเลือกใช้ระหว่างทั้งสองวิธีขึ้นอยู่กับข้อกำหนดด้านความแม่นยำ, ประสิทธิภาพ, โดย Sensor-based Control จะเหมาะสมสำหรับการควบคุมที่ต้องการความแม่นยำสูง

1.6.1.17. Op-Amp

Op-Amp ในระบบมอเตอร์ BLDC ถูกใช้ในการประมวลผลและจัดการสัญญาณเพื่อการควบคุมที่แม่นยำ โดยหน้าที่หลักคือการขยายสัญญาณจากเซ็นเซอร์ เช่น Hall Effect Sensors เพื่อให้มีแรงดันเพียงพอสำหรับการประมวลผล การกรองสัญญาณรบกวนที่เกิดจากการสวิตช์ของ ESC และการสร้างวงจร Comparator เพื่อตรวจสอบแรงดันหรือสัญญาณตำแหน่งของโรเตอร์ Op-Amp ยังมีบทบาทในวงจรควบคุมกระแสและแรงดันไฟฟ้าสำหรับขั้บเคลื่อน MOSFET ใน ESC ทำให้มอเตอร์ทำงานได้อย่างมีเสถียรภาพและประสิทธิภาพสูงสุด

1.6.1.18. Zero Crossing



รูปแสดง Zero Crossing point

ที่มา: https://www.researchgate.net/figure/Zero-crossing-points-of-the-back-EMF-and-phase-current-commutation-points-25_fig9_45267009

จังหวะที่สัญญาณของแรงดันหรือกระแสไฟฟ้าในวงจรมอเตอร์ผ่านค่าศูนย์ (0V) ซึ่งมักเกิดขึ้นในช่วงที่โรเตอร์ของมอเตอร์อยู่ในตำแหน่งที่แน่นอน Zero Crossing เป็นจุดสำคัญในการควบคุมมอเตอร์ BLDC เพราะช่วยให้สามารถจับตำแหน่งของโรเตอร์ได้โดยไม่ต้องใช้เซ็นเซอร์ตำแหน่ง เช่น Hall Effect Sensors หรือ Encoder โดยในการควบคุมมอเตอร์แบบ Sensorless, การตรวจจับ Zero Crossing สามารถใช้ในการคำนวณการหมุนของโรเตอร์และกำหนดช่วงเวลาการเปลี่ยนแปลงสัญญาณขั้บเคลื่อนเพื่อให้มอเตอร์หมุนได้อย่างราบรื่นและมีประสิทธิภาพ โดยปกติ Zero Crossing จะเกิดขึ้นที่ตำแหน่งที่มีการเปลี่ยนแปลงจากการกระแสหรือแรงดันบวกไปเป็นลบ และสามารถใช้ข้อมูลนี้ในการควบคุมการจ่ายไฟและการหมุนของมอเตอร์ได้อย่างแม่นยำ

1.6.1.19. ความสัมพันธ์ระหว่าง 6-Step Control และ PWM

ในระบบมอเตอร์ BLDC ที่ใช้ 6-Step Control, การใช้ PWM จะถูกนำมาใช้ร่วมกับการสลับสถานะของการจ่ายพลังงานในแต่ละขั้นตอนของการควบคุมมอเตอร์ ในแต่ละขั้นตอนของ 6-Step Control, กระแสที่เหลือไปยังชุดลวดจะถูกควบคุมโดยการใช้ PWM เพื่อปรับความแรงของกระแสที่จ่ายไปยังมอเตอร์ ซึ่งช่วยให้สามารถควบคุมความเร็วและการหมุนของมอเตอร์ได้อย่างแม่นยำและมีประสิทธิภาพสูงสุด ดังนั้น, 6-Step Control และ PWM จึงทำงานร่วมกันในการควบคุมการหมุนของมอเตอร์ BLDC โดยที่ 6-Step Control ควบคุมลำดับการสลับสถานะของการจ่ายพลังงาน ส่วน PWM ควบคุมความแรงของพลังงานที่จ่ายไปยังมอเตอร์ในแต่ละช่วงเวลา

1.6.2. RPM (Revolutions Per Minute)

จำนวนรอบที่โรเตอร์ของมอเตอร์หมุนได้ในหนึ่งนาที ซึ่งเป็นตัวชี้วัดสำคัญในระบบที่ใช้มอเตอร์ไฟฟ้า เช่น มอเตอร์ BLDC, DC motor หรือมอเตอร์ AC โดยค่าของ RPM มีความสัมพันธ์โดยตรงกับความเร็วเชิงมุมของมอเตอร์และแรงดันไฟฟ้าที่ป้อนให้กับชุดลวด

การคำนวณ RPM

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

โดย

- Frequency = ความถี่ของสัญญาณไฟฟ้าในหนึ่งเฟส(วัดจาก Oscilloscope) มีหน่วยเป็น เฮิรตซ์(Hz)

- Number of Pole Pairs= จำนวนคู่ขั้วแม่เหล็ก (Pole Pairs) ของมอเตอร์ BLDC
- คูณด้วย 60 เพื่อแปลงความถี่ต่อวินาทีเป็นรอบต่อนาที (RPM)

สำหรับมอเตอร์ BLDC: การคำนวณ RPM อาจใช้อัตราส่วนระหว่าง Back EMF และแรงดันไฟฟ้าที่ป้อนเข้า

1.7. วิธีดำเนินการทดลอง

การดำเนินการทดลองเริ่มต้นจากการตั้งค่าพื้นฐานในโปรแกรม STM32 Motor Control Workbench 6.3.2 ซึ่งออกแบบมาเพื่อควบคุมมอเตอร์ BLDC (Brushless DC Motor) ในโหมด 6-Step (Trapezoidal Control) โดยเลือกใช้มอเตอร์ A2212 13T พร้อมกำหนดพารามิเตอร์ที่สำคัญ เช่น กระแสไฟฟ้าสูงสุด (Maximum Current), แรงดันไฟฟ้าสูงสุด (Maximum Voltage) และความถี่ PWM เพื่อให้การควบคุมมอเตอร์ มีความเสถียร หลังจากการตั้งค่าโปรแกรมเสร็จสมบูรณ์ ขั้นตอนถัดไปคือการเชื่อมต่อมอเตอร์เข้ากับระบบควบคุม โดยใช้บอร์ด NUCLEO-G474RE ซึ่งทำหน้าที่เป็น Control Board

ตั้งค่า RPM (Revolutions Per Minute) เพื่อควบคุมความเร็วรอบของมอเตอร์ คณิตผู้จัดทำได้กำหนดค่า Max App Speed ของมอเตอร์ที่ 9,864 RPM ซึ่งเป็นความเร็วสูงสุดที่มอเตอร์สามารถทำงานได้ จากนั้นได้เก็บข้อมูลในช่วงความเร็วจาก 0% ถึง 100% ของ RPM สูงสุด โดยบันทึกค่าที่ 0%, 20%, 40%, 60%, 80%, และ 100% RPM ซึ่งช่วยในการวิเคราะห์การเปลี่ยนแปลงของ RPM เมื่อปรับค่าความเร็วของมอเตอร์ในแต่ละช่วงการทำงานและประเมินประสิทธิภาพการทำงานของมอเตอร์ในสภาพต่าง ๆ

การทดสอบเริ่มต้นที่ 20% RPM (1,973 RPM) เนื่องจากพบว่าในการทดสอบที่ค่าความเร็วต่ำกว่า 20% RPM มอเตอร์สามารถหมุนได้เพียงช่วงเวลาสั้น ๆ หรือเพียงแค่ช่วงที่ขาดๆ หายๆ เนื่องจากต้องการแรงบิดเริ่มต้นที่สูง เพียงพอเพื่อให้มอเตอร์หมุนได้อย่างต่อเนื่อง จากนั้นได้ทดสอบที่ความเร็ว RPM 50% (4,932 RPM) ในทั้งสองทิศทาง คือ ตามเข็มนาฬิกาและวนเข็มนาฬิกา โดยพบว่าค่าตอบแทนในการหมุนมีค่าเท่ากัน แต่มีความแตกต่างในมุมเฟสของสัญญาณไฟฟ้าที่เกิดขึ้น จึงเลือกเก็บข้อมูลเฉพาะในทิศทางทวนเข็มนาฬิกาเท่านั้น

ในการตรวจสอบสัญญาณไฟฟ้า 3 เฟสจากมอเตอร์ BLDC คณิตผู้จัดทำได้ใช้ Oscilloscope เชื่อมต่อกับขั้วของมอเตอร์ที่ขา 1, 2, และ 3 ของบอร์ดควบคุม ซึ่งตรงกับขั้ว U, V, และ W ของมอเตอร์ การสังเกตสัญญาณ 3 เฟสช่วยให้สามารถติดตามการทำงานของมอเตอร์ในแต่ละเฟสและทำความเข้าใจพฤติกรรมไฟฟ้าในระบบ การตั้งค่าของ Oscilloscope ถูกปรับในแกน X (เวลา) เพื่อให้สามารถวิเคราะห์ลักษณะของสัญญาณในแต่ละช่วงเวลาของ 1 Cycle ของมอเตอร์ได้อย่างละเอียด และในแกน Y (แรงดันไฟฟ้า) ตั้งค่าความละเอียดที่ 10 V ต่อช่องเพื่อแสดงแรงดันของสัญญาณในแต่ละเฟสอย่างชัดเจน

ข้อมูลที่เก็บจาก Oscilloscope จำนวน 10 ครั้งในแต่ละ Duty Cycle ใช้ในการศึกษาและเปรียบเทียบความสัมพันธ์ระหว่างสัญญาณ 3 เฟสกับ RPM ที่วัดได้จากมอเตอร์ ผลการทดลองพบว่า ความถี่ของสัญญาณไฟฟ้า 3 เฟสมีความสัมพันธ์เชิงเส้นกับความเร็วรอบของมอเตอร์ และเมื่อคำนวณค่าความเร็วรอบ RPM จากการวัดค่าตอบแทนใน 1 Cycle ของสัญญาณพบว่า RPM ที่คำนวณจากสัญญาณมีค่าที่ใกล้เคียงกับค่าที่ตั้งไว้ในโปรแกรม

1.8. วัสดุอุปกรณ์

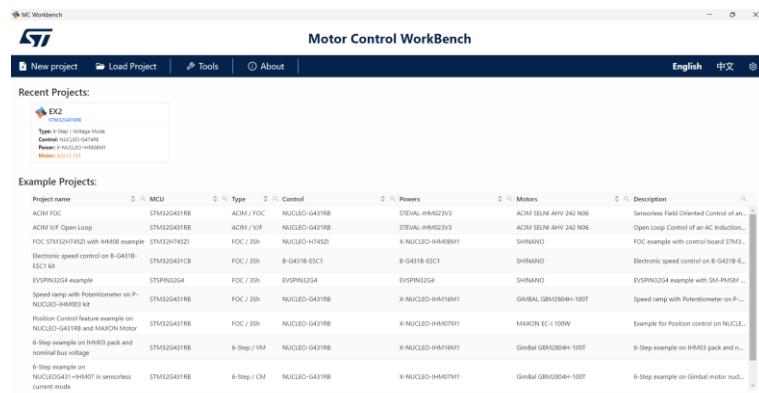
- BLDC Motor จำนวน 1 อัน
- STMICROELECTRONICS X-NUCLEO-IHM08M1 จำนวน 1 อัน
- Nucleo STM32G474RE พร้อมสายอัปโหลด จำนวน 1 ชุด
- BLDCXplorer จำนวน 1 ชุด - ฐานสามารถบรรจุบอร์ดควบคุม, Breadboard, 3D-Print ใช้สำหรับการประกอบกับ BLDC Motor

1.9. ขั้นตอนการดำเนินงาน

1.9.1. ตั้งค่ามอเตอร์ในโปรแกรม STM32 Motor Control Workbench 6.3.2 เพื่อควบคุมการทำงาน

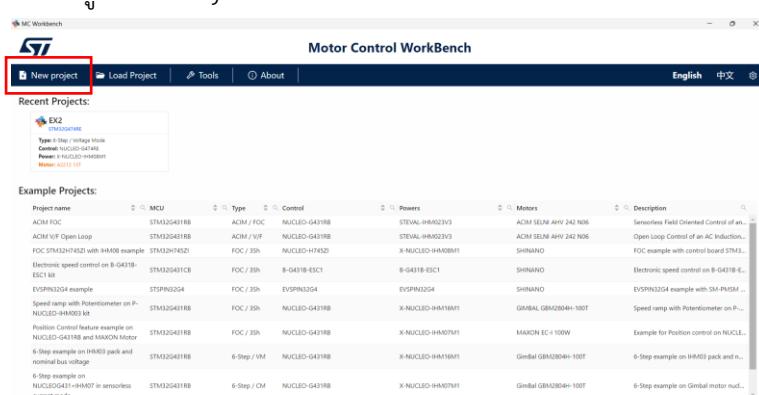
1.9.1.1. เปิดโปรแกรม STM32 Motor Control Workbench 6.3.2 และสร้างโปรเจกต์

1.9.1.1.1. เปิดโปรแกรม STM32 Motor Control Workbench 6.3.2



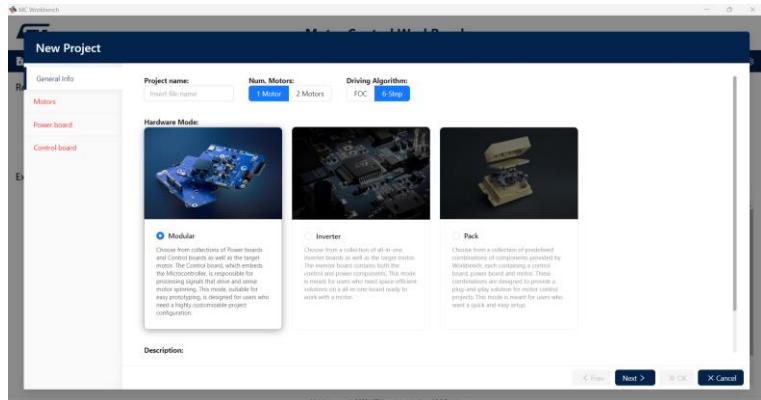
รูปแสดง โปรแกรม Motor Workbench 6.3.2

1.9.1.1.2. เลือกเมนู New Project เพื่อเริ่มสร้างโปรเจกต์ใหม่



รูปแสดง เลือกเมนู New Project

1.9.1.1.3. กรอกข้อมูลชื่อโปรเจกต์และเลือกประเภทมอเตอร์ที่ใช้งาน

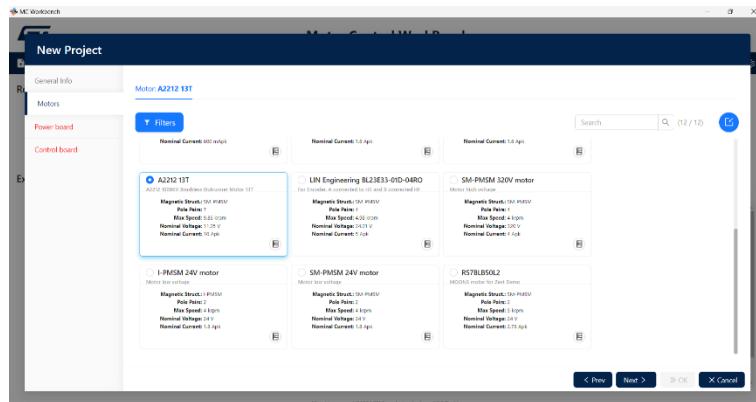


รูปแสดง การกรอกข้อมูลชื่อโปรเจกต์และเลือกประเภทมอเตอร์ที่ใช้งาน

- เลือกประเภท Num. Motors เป็น 1 Motor เพื่อกำหนดว่าการทดลองหรือการควบคุมจะใช้มอเตอร์เพียงตัวเดียว
- เลือกประเภท Driving Algorithm เป็น 6-Step ซึ่งเป็นเทคนิคการควบคุมมอเตอร์ BLDC แบบ Trapezoidal ที่ง่ายต่อการใช้งานและเหมาะสมสำหรับมอเตอร์
- เลือกประเภท Hardware Mode เป็น Modular ซึ่งหมายถึงการใช้บอร์ดควบคุมแบบแยกส่วน
- ทำการตั้งชื่อโปรเจกต์

1.9.1.2. เลือกมอเตอร์ที่ใช้งาน (Motor Selection) ในโปรแกรม STM32 Motor Control

Workbench 6.3.2

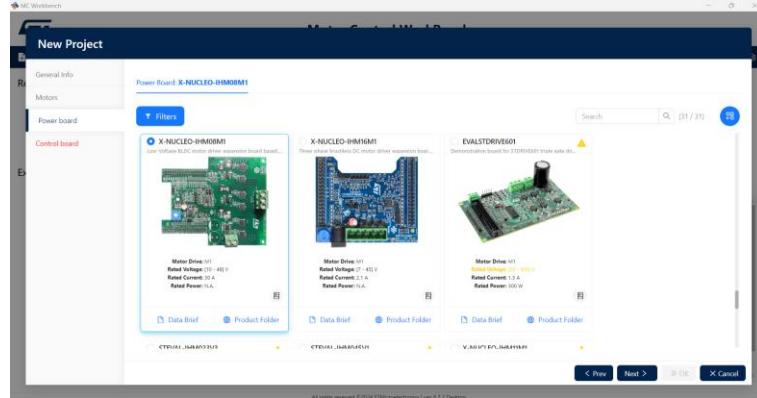


รูปแสดง เลือกมอเตอร์ที่ใช้งาน (Motor Selection) ในโปรแกรม

- ในเมนู "Motors" ให้ทำการเลือกมอเตอร์ที่ต้องการใช้งาน เป็น A2212_13T
- โดยรายละเอียดของมอเตอร์ A2212_13T มีข้อมูลดังนี้
 - Magnetic Structure: โครงสร้างแม่เหล็ก (SM-PMSM)
 - Pole Pairs: จำนวนคู่ขั้วแม่เหล็ก (7)

- Max Speed: ความเร็วสูงสุด (9.86 KRPM)
- Nominal Voltage: แรงดันไฟฟ้าที่กำหนด (11.05 V)
- Nominal Current: กระแสไฟฟ้าที่กำหนด (10 Apk)

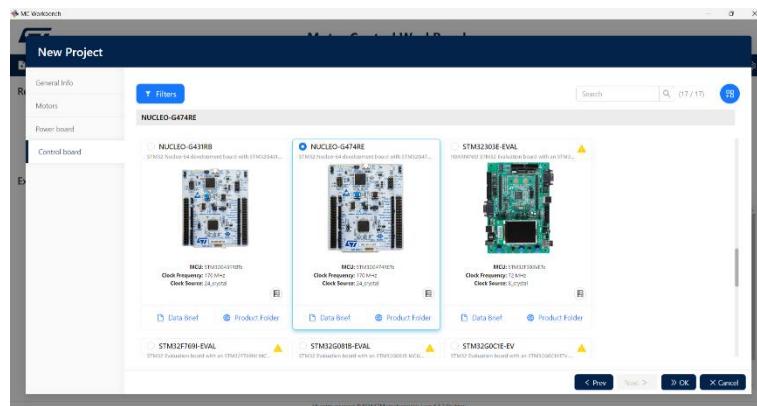
1.9.1.3. เลือก Power Board ในโปรแกรม STM32 Motor Control Workbench 6.3.2



รูปแสดง Power Board ในโปรแกรม

- ในเมนู "Power Board" ให้ทำการเลือก X-NUCLEO-IHM08M1
- โดยรายละเอียดของ X-NUCLEO-IHM08M1 มีข้อมูลดังนี้
 - Motor Drive: M1
 - Rated Voltage: รองรับแรงดันไฟฟ้าระหว่าง 10 - 48V
 - Rated Current: รองรับกระแสสูงสุดที่ 30A
 - Rated Power: ไม่มีข้อมูลกำลังไฟสูงสุดที่ระบุ
 - ประเภท: บอร์ดขัมomatic BLDC สำหรับแรงดันต่ำ

1.9.1.4. เลือก Control Board ในโปรแกรม STM32 Motor Control Workbench 6.3.2

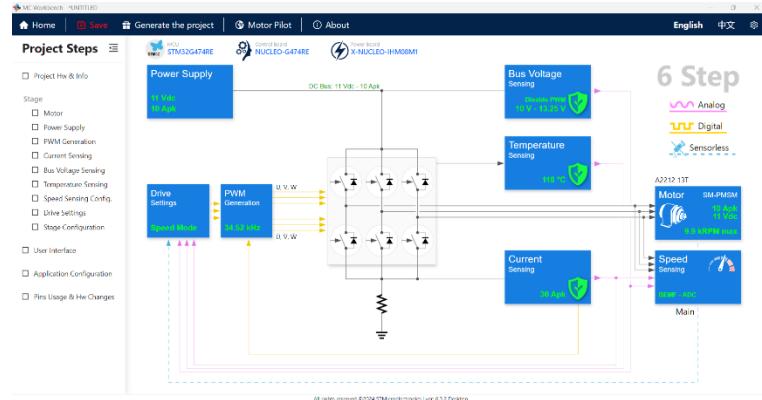


รูปแสดง Control Board ในโปรแกรม

- ในเมนู "Control Board" ให้ทำการเลือก NUCLEO-G474RE
- โดยรายละเอียดของ NUCLEO-G474RE มีข้อมูลดังนี้
 - MCU: STM32G474RETx
 - Clock Frequency: 170 MHz

- Clock Source: 24 MHz crystal ประเภท: บอร์ดซิบมอเตอร์ BLDC สำหรับแรงดันต่ำ

หลังจากตั้งค่าระบบตามที่กำหนดไว้ข้างต้น ระบบจะเปิดหน้า Project Steps ขึ้นมาสำหรับการควบคุมมอเตอร์ในโหมด 6 Step (Trapezoidal Control) โดยมีข้อมูลดังนี้



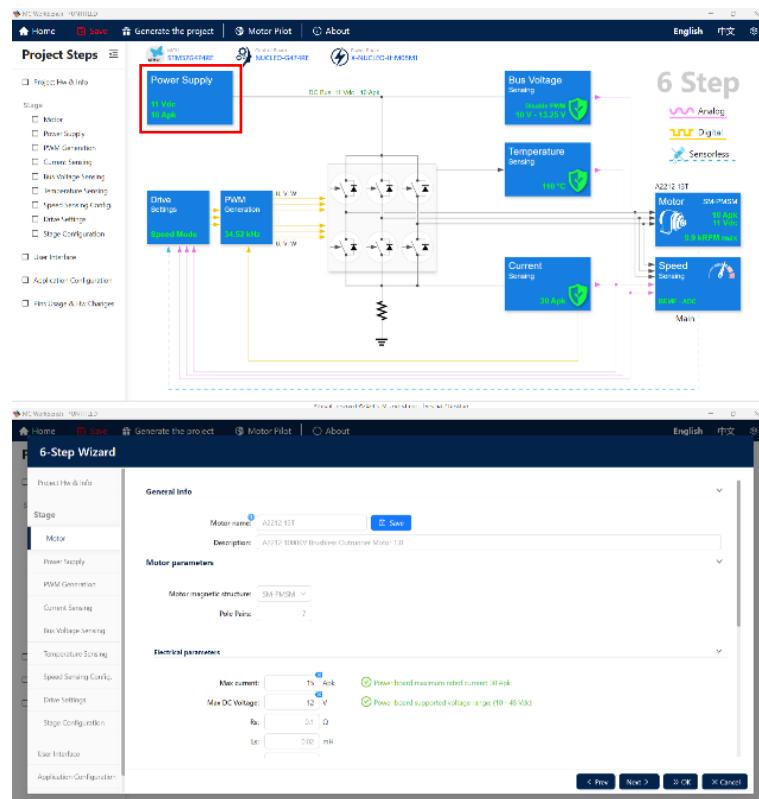
รูปแสดง หน้า Project Steps

- MCU และ Control Board
 - STM32G474RE: ไมโครคอนโทรลเลอร์ที่ใช้ในระบบควบคุมมอเตอร์
 - X-NUCLEO-IHM08M1: Power Board (Driver Board) สำหรับควบคุมมอเตอร์
- Power Supply
 - 11 VDC, 10 Apk: ค่าของแหล่งจ่ายไฟที่ใช้ในระบบ
- PWM Generation
 - 34.52 kHz: ความถี่ของสัญญาณ PWM ที่ใช้ควบคุมมอเตอร์
- Drive Settings
 - Speed Mode: ระบบถูกตั้งค่าให้ควบคุมความเร็วของมอเตอร์
- Bus Voltage Sensing
 - ใช้เซนเซอร์วัดแรงดันของบัสไฟฟ้าภายในระบบ
 - 10 V - 13.25 V: ช่วงแรงดันที่กำหนด และมีฟังก์ชันสำหรับปิด PWM หากแรงดันอยู่นอกช่วง
- Temperature Sensing
 - ระบบวัดอุณหภูมิที่กำหนดค่าไว้ที่ 110 °C พร้อมระบบป้องกัน
- Current Sensing
 - ตรวจจับกระแสที่แหล่งผ่านอินเวอร์เตอร์
 - รองรับค่ากระแสสูงสุด 30 Apk
- Motor
 - A2212 13T: รุ่นของมอเตอร์ที่กำหนดไว้

- SM-PMSM (Synchronous Permanent Magnet Motor): ประเภทของมอเตอร์
- 10 A_pk, 11 VDC, 9.9 kRPM max: ข้อมูลスペกของมอเตอร์
- Speed Sensing
 - ใช้วิวัฒนาการความเร็วแบบ Sensorless (ไม่ใช้เซนเซอร์ออลล์หรือออปติก)
 - BEMF-ADC: วัดแรงดันย้อนกลับ (Back EMF) ผ่าน ADC เพื่อคำนวณความเร็ว

1.9.1.5. ตั้งค่า Max current และ Max DC Voltage ในโปรแกรม STM32 Motor Control

Workbench 6.3.2



รูปแสดง หน้าการตั้งค่า Max current และ Max DC Voltage ในโปรแกรม
โดยทำการตั้งค่า Max current และ Max DC Voltage โปรแกรม STM32 Motor Control
Workbench 6.3.2 ดังนี้

● Max current

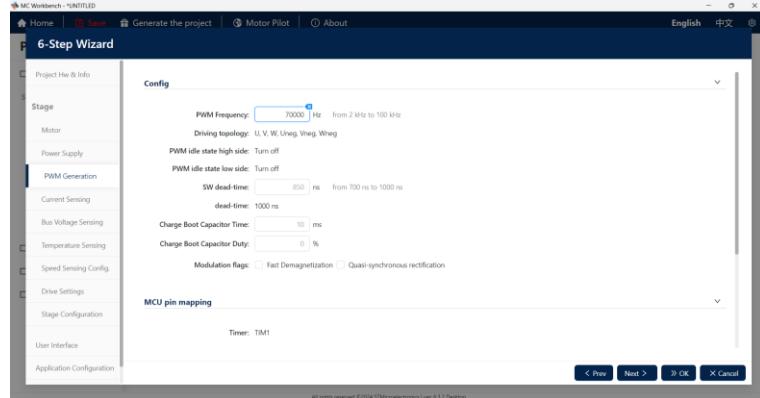
- ค่าที่ตั้งไว้คือ 15 A_pk หมายถึงค่ากระแสไฟฟ้าสูงสุดที่มอเตอร์สามารถรองรับได้
ระหว่างการทำงาน โดยการกำหนดค่านี้ช่วยป้องกันไม่ให้มอเตอร์เกิดความเสียหาย
หรือความร้อนเกินพิกัด ซึ่งอาจเกิดขึ้นในกรณีที่มอเตอร์โหลดมากหรือกระแสไฟฟ้าเกิน
กำหนด

● Max DC Voltage

- ค่าที่ตั้งไว้คือ 12 V หมายถึงแรงดันไฟฟ้ากระแสตรงสูงสุดที่สามารถจ่ายให้กับ
มอเตอร์ได้ในระบบนี้ โดยค่าที่ตั้งไว้นี้สอดคล้องกับแรงดันที่ Power Board (X-

NUCLEO-IHM08M1) รองรับได้ และช่วยให้ระบบสามารถทำงานได้อย่างมีเสถียรภาพและมีประสิทธิภาพ โดยไม่เสี่ยงต่อการทำงานผิดพลาดหรือเกิดความเสียหายต่อตัวมอเตอร์และระบบควบคุม

1.9.1.6. ตั้งค่า PWM Frequency ในโปรแกรม STM32 Motor Control Workbench 6.3.2



รูปแสดง หน้าการตั้งค่า PWM Frequency ในโปรแกรม
โดยทำการตั้งค่า PWM Frequency โปรแกรม STM32 Motor Control Workbench 6.3.2

ดังนี้

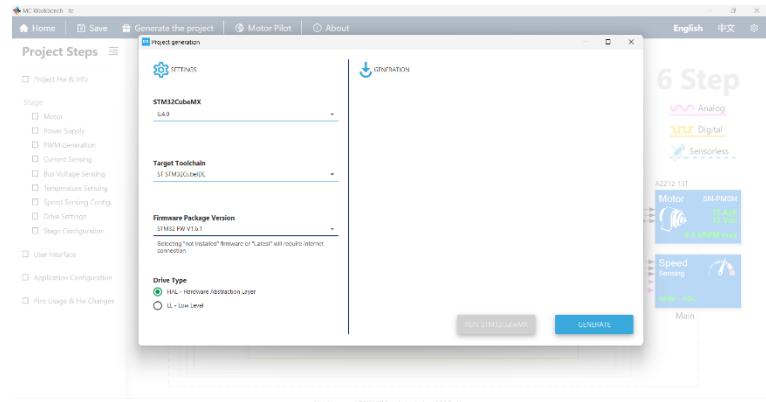
- PWM Frequency

- ค่าที่ตั้งไว้คือ 70,000 Hz (70 kHz) ช่วยลดเสียงรบกวนที่เกิดจากการทำงานของมอเตอร์ (Acoustic Noise)
- ช่วงค่าที่สามารถกำหนดได้: 2 kHz ถึง 100 kHz
- ค่าความถี่ที่สูงขึ้นช่วยลดเสียงรบกวนและทำให้การควบคุมราบรื่นขึ้น แต่จะเพิ่มภาระการสลับของ MOSFET (Switching Loss)

จากนั้นตรวจสอบการตั้งค่าทั้งหมด เช่น Max current, Max DC Voltage, และ PWM Frequency ให้แน่ใจว่าเหมาะสมกับมอเตอร์และฮาร์ดแวร์ที่ใช้งาน รวมถึงอยู่ในช่วงค่าที่ระบบรองรับ จากนั้นคลิกที่ปุ่ม “Generate the project” ที่อยู่ด้านบนของหน้าจอ เพื่อให้โปรแกรม Motor Workbench 6.3.2 ทำการสร้างโค้ดสำหรับใช้งานร่วมกับ STM32CubeIDE

1.9.1.7. สร้างโปรเจกต์ (Project Generation) ในโปรแกรม STM32 Motor Control Workbench

6.3.2



รูปแสดง หน้าการตั้งค่า Project Generation ในโปรแกรม

1.9.1.8. โดยทำการตั้ง Project Generation ในโปรแกรม STM32 Motor Control Workbench

6.3.2 ดังนี้

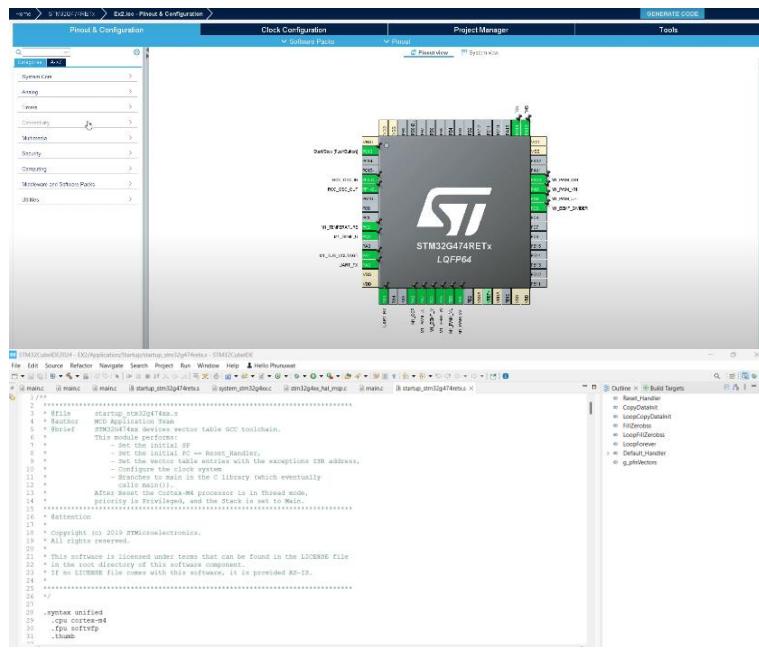
• SETTINGS

- STM32CubeMX: 6.4.0 ระบุว่าเวอร์ชันของโปรแกรม STM32CubeMX ที่จะใช้ในการสร้างโปรเจกต์คือเวอร์ชัน 6.4.0
- Target Toolchain: ST STM32CubeIDE กำหนดให้ใช้ STM32CubeIDE เป็นเครื่องมือสำหรับพัฒนาและปรับแก้โค้ด
- Firmware Package Version: STM32 FW V1.6.1 ใช้เวอร์ชันเฟิร์มแวร์ STM32 FW V1.6.1 เพื่อสร้างโปรเจกต์ หรือขึ้นกับการติดตั้งเวอร์ชันในระบบ หรือเวอร์ชันล่าสุดที่ต้องการ

• Drive Type

- ตั้งค่าเป็น HAL (Hardware Abstraction Layer) ซึ่งช่วยให้การพัฒนามีความง่ายและรวดเร็ว เหมาะสำหรับผู้เริ่มต้นหรือการใช้งานที่ต้องการความยืดหยุ่น จากนั้นคลิกที่ปุ่ม “GENERATE” เพื่อให้โปรแกรมสร้างโค้ดตามค่าที่ตั้งไว้ใน 6-Step Wizard เมื่อการสร้างโปรเจกต์เสร็จสิ้น ระบบจะแสดงตัวเลือกให้ดาวน์โหลดโค้ดหรือเปิดใน STM32CubeIDE เพื่อปรับแก้ไขและแฟลชโค้ดลงในไมโครคอนโทรลเลอร์เพื่อใช้งาน

1.9.1.9. เริ่มต้นการใช้งานผ่าน STM32CubeMX, STM32CubeIDE และ STM32 Motor Control Workbench 6.3.2

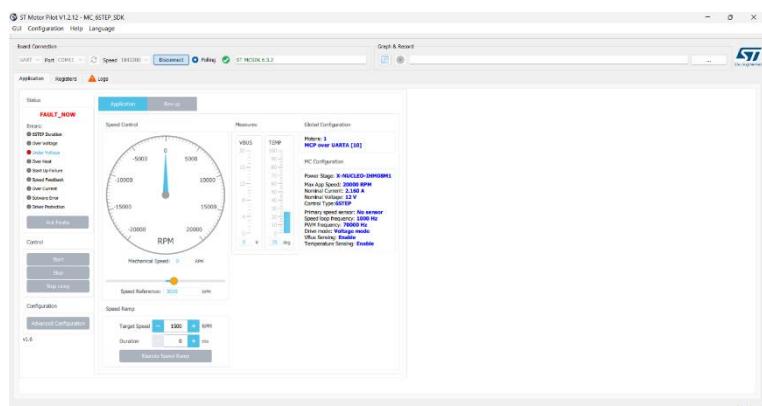


รูปแสดง หน้าการใช้งานผ่าน STM32CubeMX, STM32CubeIDE และ STM32 Motor Control

Workbench 6.3.2

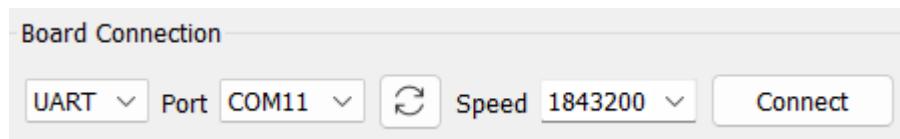
- กด Generate Code และ Open Project ใน STM32CubeMX หลังจากตั้งค่าพารามิเตอร์และกำหนดค่าต่าง ๆ เสร็จเรียบร้อย ให้กดปุ่ม Generate Code เพื่อสร้างプロジェクト และเปิดไฟล์ใน STM32CubeIDE เพื่อเริ่มต้นพัฒนาเพิ่มเติม
- กด Run ใน STM32CubeIDE เชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์ STM32 กับคอมพิวเตอร์ผ่านสาย USB จากนั้นกด Run เพื่อแฟลชโค้ดลงบอร์ด โดยโปรแกรมจะอัปโหลดโค้ดที่สร้างไว้ไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงานของมอเตอร์ตามค่าที่กำหนด
- เปิด Motor Pilot ภายใน โปรแกรม STM32 Motor Control Workbench 6.3.2

1.9.2. ตั้งค่าโปรแกรม Motor Pilot



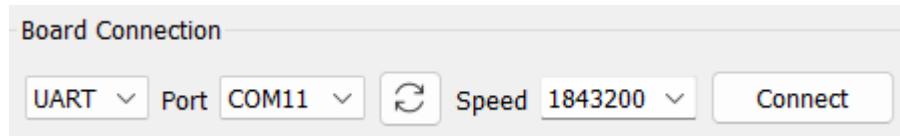
รูปแสดง หน้าโปรแกรม Motor Pilot

1.9.2.1. ตั้งค่า Board Connection ให้เป็นแบบ UART ทำการเลือก Port และกด Connect



รูปแสดง หน้าการตั้งค่า Board Connection ในโปรแกรม

1.9.2.2. ตั้งค่า Motor Speed ที่บริเวณ Speed Ramp ให้ตรงกับความต้องการจากนั้นกด Execute Speed Ramp และกด Run



รูปแสดง หน้าการตั้งค่า Motor Speed ในโปรแกรม

1.9.3. ควบคุมการทำงานของมอเตอร์ผ่านโปรแกรม Motor Pilot

1.9.3.1. ตั้งค่า Revolutions Per Minute หรือจำนวนรอบต่อหนึ่งนาทีผ่านโปรแกรม Motor Pilot สำหรับการควบคุมการทำงานของมอเตอร์ และบันทึกผลการทำงานทดลอง

คณะผู้จัดทำได้ทำการควบคุมการทำงานของมอเตอร์ BLDC ผ่านโปรแกรม Motor Pilot โดยการตั้งค่า Revolutions Per Minute (RPM) เพื่อกำหนดความเร็วในการหมุนของมอเตอร์ ซึ่ง RPM นี้เป็นตัวแปรสำคัญในการควบคุมความเร็วรอบของมอเตอร์ในแต่ละช่วงการทำงาน โดยได้ตั้งค่า Max App Speed ของมอเตอร์ที่ 9,864 RPM ซึ่งถือว่าเป็นความเร็วสูงสุดที่มอเตอร์สามารถทำงานได้ จากนั้นได้ทำการเก็บข้อมูลในช่วงที่ตั้งค่าความเร็วจาก 0% ถึง 100% ของความเร็วสูงสุดนี้ โดยที่ทุก 20% ของการตั้งค่าได้ทำการเก็บข้อมูล RPM ดังนี้:

0% RPM = 0 RPM (เริ่มต้น)
20% RPM = 1,973 RPM
40% RPM = 3,946 RPM
60% RPM = 5,918 RPM
80% RPM = 7,891 RPM
100% RPM = 9,864 RPM

ข้อมูลเหล่านี้จะช่วยให้คณะผู้จัดทำสามารถวิเคราะห์การเปลี่ยนแปลงของ RPM เมื่อปรับความเร็วของมอเตอร์ในช่วงต่าง ๆ และช่วยในการประเมินประสิทธิภาพของมอเตอร์ในช่วงความเร็วต่าง ๆ

ในขั้นตอนการทดลอง คณะผู้จัดทำได้เริ่มเก็บข้อมูลจากค่า 20% RPM ขึ้นไป เนื่องจากใน การทดสอบที่มีการตั้งค่า ต่ำกว่าค่า 20% RPM พบร่วมกับมอเตอร์สามารถทำงานได้เพียงระยะเวลาสั้น ๆ หรือเพียงแค่ ช่วงขณะ เนื่องจากค่าความเร็ว RPM ที่ตั้งเข้าไปนั้นต่ำกว่าความเร็วเริ่มต้นที่มอเตอร์ต้องใช้เพื่อให้สามารถหมุนได้อย่างต่อเนื่อง ซึ่งเป็นการทำงานในลักษณะของมอเตอร์ที่ต้องการแรงบิดเริ่มต้น (Starting Torque) ที่สูงพอสมควรในช่วงเริ่มต้นการหมุน

ดังนั้น คณะผู้จัดทำจึงตัดสินใจเริ่มเก็บข้อมูลที่ 20% RPM หรือ 1,972 RPM เป็นจุดเริ่มต้น เนื่องจากที่ความเร็วนี้มอเตอร์สามารถเริ่มทำงานได้อย่างมีประสิทธิภาพและสามารถหมุนได้อย่างต่อเนื่อง จึงทำให้ได้ข้อมูลที่มีความแม่นยำและสะท้อนถึงพฤติกรรมการทำงานของมอเตอร์ได้ดีขึ้น ในช่วงการทดสอบต่อไป

ในการทดลองที่ดำเนินการนั้น คณะผู้จัดทำได้ทดลองเก็บค่าสัญญาณ Revolutions Per Minute (RPM) ที่ 50% หรือ 4,932 RPM โดยทำการทดสอบในทิศทางเดียว ตามเข็มนาฬิกา (Clockwise) และ ทวนเข็มนาฬิกา (Counterclockwise) ซึ่งผลการทดลองพบว่า คาบเวลา (Time Period) ในการหมุนในทิศทางเดียวเท่ากัน แต่มีความแตกต่างที่ Phase หรือมุมไฟล์ของ สัญญาณที่เกิดขึ้น

เนื่องจากความแตกต่างใน Phase ที่เกิดขึ้นระหว่างการหมุนไปในทิศทางตามเข็มนาฬิกาและ ทวนเข็มนาฬิกา คณะผู้จัดทำจึงเลือกที่จะเก็บข้อมูลเฉพาะในทิศทาง ทวนเข็มนาฬิกา (Counterclockwise) เท่านั้น โดยจะทำการบันทึกค่าของ RPM ที่ 20%, 40%, 60%, 80%, และ 100% จากค่าความเร็วสูงสุด (Max App Speed) ที่ 9,864 RPM เท่านั้น

การเก็บข้อมูลในแต่ละจุดนี้จะช่วยให้สามารถทำการวิเคราะห์และเปรียบเทียบค่าของ RPM ในช่วงต่าง ๆ เพื่อศึกษาความสัมพันธ์ระหว่างการตั้งค่าความเร็วและค่าความเร็วที่มอเตอร์สามารถทำได้จริงในสภาพการทำงานต่าง ๆ ทั้งนี้จะช่วยในการประเมินประสิทธิภาพการทำงานของมอเตอร์ BLDC ในการควบคุมด้วยโปรแกรม Motor Pilot และทำให้สามารถปรับการตั้งค่าได้ตามความเหมาะสมในงานต่าง ๆ ที่ต้องการ

1.9.4. ตั้งค่า Oscilloscope เพื่อให้สามารถสังเกตและวิเคราะห์สัญญาณไฟฟ้า 3 เฟสที่เกิดขึ้นจากมอเตอร์ และบันทึกผล



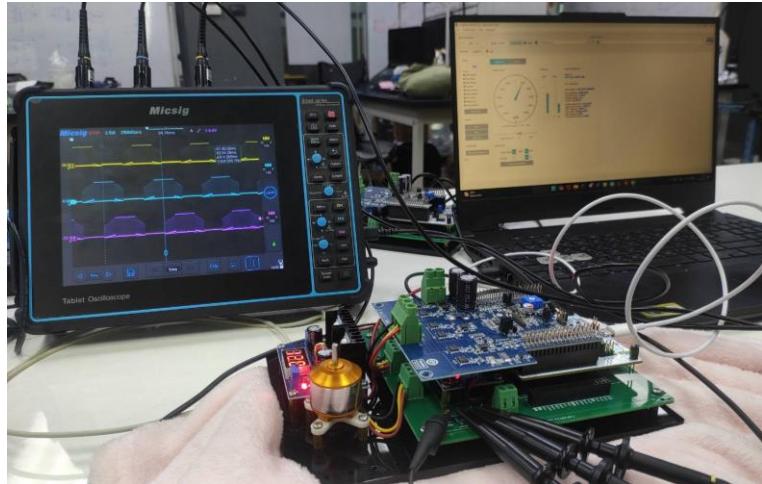
รูปแสดง การนีบสายสัญญาณ Oscilloscope กับตัวบอร์ด

โดยในการทดลอง คณะผู้จัดทำได้ใช้ Oscilloscope โดยเชื่อมต่อ Channel 1, 2 และ 3 เข้ากับขา ก้าง平坦บนบอร์ดควบคุมของมอเตอร์ที่ติดแผ่นงา ขา 1, 2 และ 3 ตามลำดับ ซึ่งขาเหล่านี้สอดคล้องกับขั้ว U, V และ W ของ BLDC Motor โดยการตรวจสอบดังกล่าวช่วยให้สามารถดูสัญญาณไฟฟ้า 3 เฟสของมอเตอร์ ในระหว่างการทำงานได้อย่างละเอียด

การตรวจสอบสัญญาณ 3 เฟส:

ขั้ว U, V, W เป็นสายไฟที่ต่อ กับ ชุดวงจรในสเตเตอร์ของมอเตอร์ ซึ่งแต่ละสายจะมีสัญญาณไฟฟ้าที่มีความแตกต่างทางเฟส (Phase Shift) อยู่ที่ 120 องศา (Electrical Degrees) ระหว่างกัน ลำดับเฟสของ สัญญาณจะเป็นไปตามลำดับ U → V → W ซึ่งถูกออกแบบมาเพื่อสร้างแรงบิดต่อเนื่องและทำให้มอเตอร์หมุนได้อย่างราบรื่น

1.9.4.1. ตั้งค่าแกน X และแกน Y ของ Oscilloscope เพื่อให้สามารถสังเกตและวิเคราะห์ สัญญาณไฟฟ้า 3 เฟสที่เกิดขึ้นจากมอเตอร์



รูปแสดง การตั้งค่า Oscilloscope เพื่อให้สามารถสังเกตและวิเคราะห์สัญญาณไฟฟ้า 3 เฟสที่เกิดขึ้น จากมอเตอร์และบันทึกผล

1) การตั้งค่าแกน X (เวลาหรือ Time Axis): แกน X ถูกตั้งให้แสดงช่วงเวลาในหน่วย มิลลิวินาที (ms) โดยมีความละเอียดที่เหมาะสมในช่วงเวลา 200 μs, 500 μs และ 1 ms เพื่อศึกษา ลักษณะของสัญญาณในแต่ละช่วงของ 1 Cycle ของมอเตอร์ เช่น ตำแหน่ง Zero Crossing, Floating Phase และ Conducting Phase การตั้งค่าความละเอียดในแนวแกน X ช่วยให้สามารถ แยกแยะรายละเอียดต่าง ๆ ของสัญญาณ 3 เฟสได้อย่างชัดเจนในช่วงเวลาที่เหมาะสมกับความถี่ของ สัญญาณที่ตรวจสอบ

2) การตั้งค่าแกน Y (แรงดันไฟฟ้า หรือ Voltage Axis): แกน Y ถูกตั้งให้แสดงค่าความ แรงดันไฟฟ้าของสัญญาณในหน่วยโวลต์ (V) โดยคณาจัดทำได้ปรับค่าความละเอียดให้เหมาะสมกับ ช่วงแรงดันที่ใช้ในมอเตอร์ โดยตั้งค่าที่ 10 V ต่อช่อง เพื่อให้สามารถแสดงระดับแรงดันไฟฟ้าในแต่ละ เฟส (U, V, W) ได้อย่างชัดเจนและง่ายต่อการเปรียบเทียบ

3) การปรับตำแหน่ง Trigger: เนื่องจากสัญญาณที่แสดงบนหน้าจอของ Oscilloscope จะ เคลื่อนที่ในแนวแกน X อย่างต่อเนื่อง คณาจัดทำจึงได้ทำการปรับตำแหน่ง Trigger ของสัญญาณแต่ ละ Channel เพื่อให้สัญญาณที่แสดงไม่เคลื่อนที่

4) การใช้เครื่องมือ Cursor ใน Oscilloscope: ช่วยในการวัด ความสูง (Amplitude) และ ความกว้าง (Width) ของลูกคลื่นทำได้แม่นยำและสะดวก โดยการวัด:

- ความสูง (Amplitude): วัดค่าความสูงสุดของลูกคลื่น (จากจุดต่ำสุดถึงจุดสูงสุด) เพื่อ ประเมินระดับแรงดันไฟฟ้าของสัญญาณแต่ละเฟส
- ความกว้าง (Width): วัดระยะเวลาที่ลูกคลื่นใช้ในการขึ้นลงระหว่างจุดต่ำสุดและสูงสุด ซึ่ง ช่วยในการศึกษาลักษณะของสัญญาณในแต่ละเฟส

การใช้ Cursor ช่วยให้สามารถอ่านค่าความสูงและความกว้างของลูกคลื่นได้อย่างแม่นยำและสะดวก, ทำให้สามารถวิเคราะห์พฤติกรรมของมอเตอร์ในระบบ Sensorless Six-step Control ได้อย่างมีประสิทธิภาพ

5) การบันทึกข้อมูล: ในการทดลอง คณะผู้จัดทำได้เก็บค่าสัญญาณจาก Oscilloscope ทั้งหมด 10 ครั้งในแต่ละ Duty Cycle เพื่อสังเกตพฤติกรรมของสัญญาณที่เกิดขึ้น ซึ่งข้อมูลที่เก็บรวบรวมจะถูกใช้ในการวิเคราะห์ความสัมพันธ์ระหว่างสัญญาณไฟฟ้า 3 เฟสและความเร็วรอบของมอเตอร์

จากผลการทดลอง พบร่วมกับความถี่ของสัญญาณไฟฟ้าสามเฟสมีความสัมพันธ์เชิงเส้นกับความเร็วรอบของมอเตอร์ ซึ่งสอดคล้องกับหลักการทำงานของมอเตอร์ BLDC ที่ความเร็วรอบจะเปรียบเท่ากับความถี่ของสัญญาณ

1.9.5. ตรวจสอบความสอดคล้องของค่า RPM ที่คำนวณได้จากการวัดเวลาใน 1 ลูกคลื่น (Cycle) ของสัญญาณกับค่า RPM ที่ตั้งค่าไว้

คณะผู้จัดทำได้ดำเนินการวัดเวลา (Time Period) ใน 1 ลูกคลื่น (Cycle) ของสัญญาณจากแต่ละเฟสผ่านเครื่อง Oscilloscope เพื่อนำค่าที่ได้เป็นความถี่ (Frequency) ของสัญญาณ และตรวจสอบว่าค่า RPM ที่คำนวณได้มีความสอดคล้องหรือใกล้เคียงกับค่า RPM ที่ตั้งค่าไว้สำหรับควบคุมมอเตอร์หรือไม่

การคำนวณค่า RPM ของมอเตอร์ BLDC ทำได้โดยใช้สูตรดังนี้:

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

รายละเอียด:

- Frequency: ความถี่ของสัญญาณไฟฟ้าในหนึ่งเฟสที่วัดได้จาก Oscilloscope มีหน่วยเป็นเฮิรตซ์ (Hz)
- Number of Pole Pairs: จำนวนคู่ขั้วแม่เหล็ก (Pole Pairs) ของมอเตอร์ BLDC โดยในที่นี้ใช้ A2212/13T ซึ่งมีค่า Number of Pole Pairs เป็น 7

การคูณด้วย 60 เป็นการแปลงความถี่จากการอบท่อวินาที (Hz) ให้เป็นรอบต่อนาที (RPM)

จากการคำนวณ พบร่วมกับค่า RPM ที่ได้จากการวัดและคำนวณมีความใกล้เคียงกับค่า RPM ที่ตั้งค่าไว้ในโปรแกรมควบคุมมอเตอร์ และให้เห็นถึงความถูกต้องของกระบวนการวัดและการคำนวณ

นอกจากนี้ ยังพบว่าเมื่อเพิ่มค่า RPM ของมอเตอร์ให้สูงขึ้น จะเกิดแรงดันย้อนกลับ (Back Electromotive Force หรือ Back EMF) มากขึ้นตามไปด้วย ซึ่ง Back EMF นี้เป็นแรงดันที่เกิดจากการหมุนของโรเตอร์ในมอเตอร์ BLDC และมีผลโดยตรงต่อการทำงานของวงจรควบคุมมอเตอร์ การเพิ่มขึ้นของ Back EMF ใน RPM ที่สูงขึ้นอาจส่งผลต่อประสิทธิภาพและความเสถียรของมอเตอร์ ซึ่งเป็นปัจจัยที่ควรคำนึงถึงในการออกแบบและควบคุมการทำงานในระบบจริง

1.10. ผลการทดลอง

1.10.1. การทดลองหาค่าบเวลา (Time Period) ของ RPM ในทิศทางตามเข็มนาฬิกาและวนเข็มนาฬิกา คณะผู้จัดทำได้ควบคุมการทำงานของมอเตอร์ BLDC ผ่านโปรแกรม Motor Pilot โดยตั้งค่า Revolutions Per Minute (RPM) เพื่อกำหนดความเร็วรอบของมอเตอร์ ซึ่ง RPM เป็นตัวแปรสำคัญในการควบคุมความเร็วรอบของมอเตอร์ในแต่ละช่วงการทำงาน

ในการทดลองนี้ ได้ตั้งค่า Max App Speed ของมอเตอร์ไว้ที่ 9,864 RPM ซึ่งเป็นความเร็วสูงสุดที่ มอเตอร์สามารถทำงานได้ จากนั้นได้ทำการทดลองโดยตั้งค่าความเร็วของมอเตอร์ที่ 50% ของความเร็วสูงสุด หรือ 4,932 RPM

การทดสอบดำเนินการในทั้งสองทิศทางการหมุน ได้แก่

- ตามเข็มนาฬิกา (Clockwise)
- วนเข็มนาฬิกา (Counterclockwise)

โดยได้เก็บข้อมูลสัญญาณ RPM จำนวน 10 ค่า ต่อทิศทางการหมุน โดยผลการทดลองแสดงดังรูป และทำการวัดค่าบเวลา (Time Period) ใน 1 ลูกลีน (Cycle)

- การตั้งค่าแกน X (เวลาหรือ Time Axis) โดยมีความละเอียดที่เหมาะสมในช่วงเวลา 500 μ s เพื่อศึกษาลักษณะของสัญญาณในแต่ละช่วงของ 1 Cycle ของมอเตอร์ เช่น ตำแหน่ง Zero Crossing, Floating Phase และ Conducting Phase การตั้งค่าความละเอียดในแนวแกน X ช่วยให้สามารถแยกแยะรายละเอียดต่าง ๆ ของสัญญาณ 3 เฟสได้อย่างชัดเจนในช่วงเวลาที่เหมาะสมกับความถี่ของสัญญาณที่ตรวจสอบ
- การตั้งค่าแกน Y (แรงดันไฟฟ้า หรือ Voltage Axis) แกน Y ถูกตั้งให้แสดงค่าความแรงดันไฟฟ้าของสัญญาณในหน่วยโวลต์ (V) โดยคณะผู้จัดทำได้ปรับค่าความละเอียดให้เหมาะสมกับช่วงแรงดันที่ใช้ในมอเตอร์ โดยตั้งค่าที่ 10 V ต่อช่อง เพื่อให้สามารถแสดงระดับแรงดันไฟฟ้าในแต่ละเฟส (U, V, W) ได้อย่างชัดเจนและง่ายต่อการเปรียบเทียบ
- การปรับตำแหน่ง Trigger: เนื่องจากสัญญาณที่แสดงบนหน้าจอของ Oscilloscope จะเคลื่อนที่ในแนวแกน X อย่างต่อเนื่อง คณะผู้จัดทำจึงได้ทำการปรับตำแหน่ง Trigger ของสัญญาณแต่ละ Channel เพื่อให้สัญญาณที่แสดงไม่เคลื่อนที่
- การใช้เครื่องมือ Cursor ใน Oscilloscope: ช่วยในการวัด ความสูง (Amplitude) และ ความกว้าง (Width) ของลูกลีนทำได้แม่นยำและสะดวก โดยการวัดและตรวจสอบว่าค่า RPM ที่คำนวณได้มีความสอดคล้องหรือใกล้เคียงกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์หรือไม่ รวมถึงหากค่าความคลาดเคลื่อนของข้อมูลที่เกิดขึ้น โดยใช้การคำนวณค่า RPM ของมอเตอร์ BLDC





รูปแสดง ค่าคาบเวลา (Time Period)ของมอเตอร์ที่ 50% ของความเร็วสูงสุด หรือ 4,932 RPM ในทิศทาง
ทวนเข็มนาฬิกา (Counterclockwise)



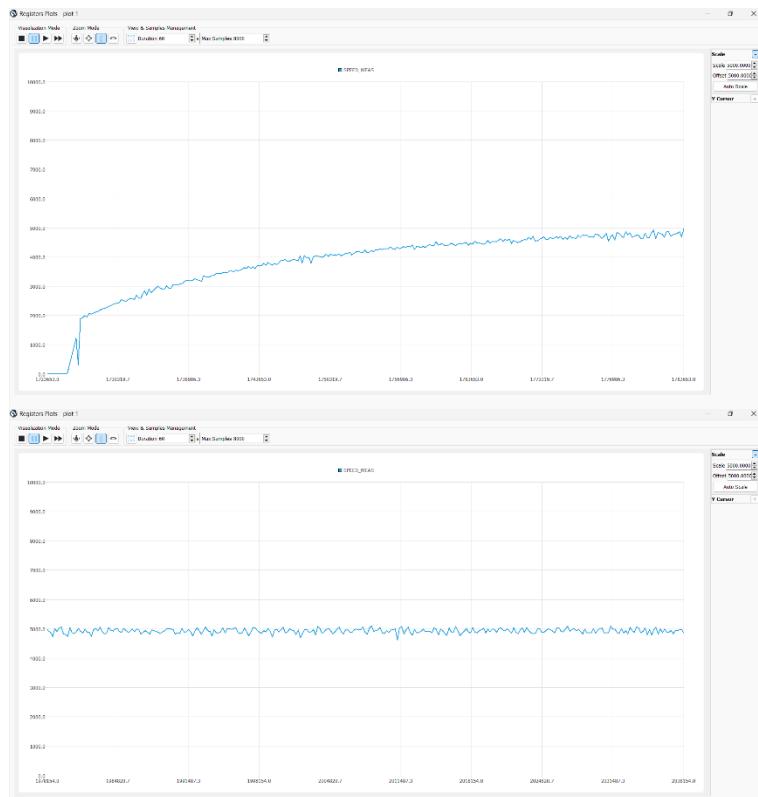


รูปแสดง ค่าคาบเวลา (Time Period)ของมอเตอร์ที่ 50% ของความเร็วสูงสุด หรือ 4,932 RPM ในทิศทาง
ตามเข็มนาฬิกา (Clockwise)

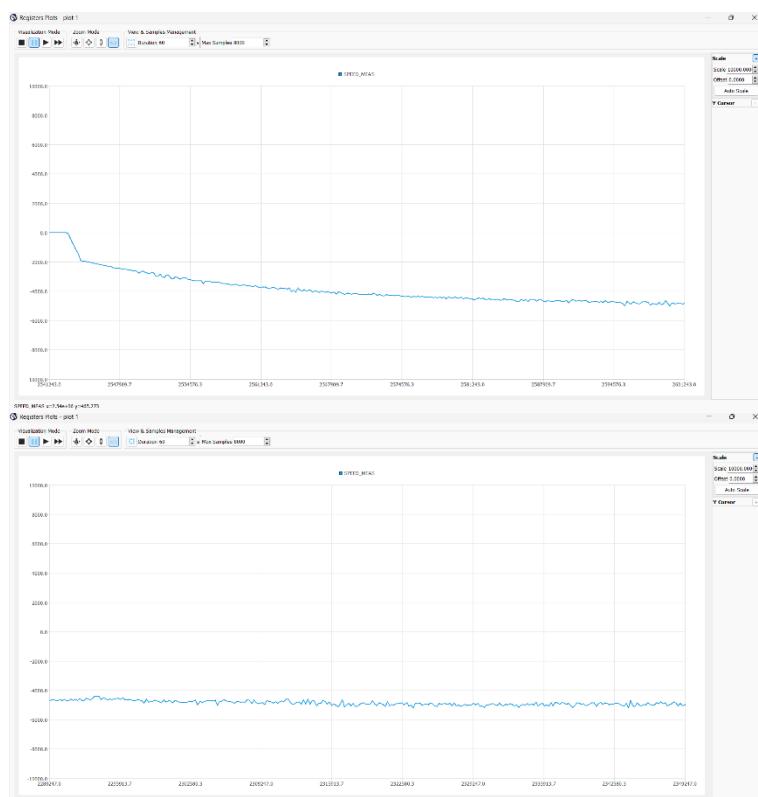
จำนวนครั้งที่ทำ การทดสอบ	ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 50% ในทิศทางวนเข็มนาฬิกา (Counterclockwise) หน่วย HZ	ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 50% ในทิศทางตามเข็มนาฬิกา (Clockwise) หน่วย HZ
1	578.0	584.8
2	574.7	565.0
3	574.7	581.4
4	574.7	565.0
5	584.8	574.7
6	568.2	574.7
7	584.8	561.8
8	584.8	568.2
9	578.0	561.8
10	581.4	565.0
ค่าเฉลี่ยข้อมูล	578.41	570.24

- ความเร็ว 4,932 RPM ในทิศทางวนเข็มนาฬิกา(Counterclockwise)
 - คำนวณค่า RPM จากค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% ได้ 4,957.80 RPM
 - ค่าความคลาดเคลื่อนของ RPM ที่เกิดขึ้น เมื่อเปรียบเทียบกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์ คือ -25.80 RPM
- ความเร็ว 4,932 RPM ในทิศทางตามเข็มนาฬิกา (Clockwise)
 - คำนวณค่า RPM จากค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% ได้ 4,887.77 RPM
 - ค่าความคลาดเคลื่อนของ RPM ที่เกิดขึ้น เมื่อเปรียบเทียบกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์ คือ 44.23 RPM

โดยจากการวิเคราะห์ค่าเฉลี่ยของข้อมูลที่เก็บได้ทั้งหมด 10 ค่า สามารถสังเกตได้ว่าค่ามีความแปรผวนของสัญญาณ โดยความอาจเกิดจากความไม่เสถียรของเซ็นเซอร์หรือการทำงานของมอเตอร์ในขณะนั้น ซึ่งทำให้สัญญาณมีการแกว่งเล็กน้อยดังรูป สัญญาณเหล่านี้อาจได้รับผลกระทบจากสัญญาณรบกวนหรือการกรองสัญญาณที่ไม่ดี ทำให้มีความแปรผวนในข้อมูลที่วัดได้



รูปแสดง ความแปรผันของสัญญาณของความเร็ว 4,932 RPM ในทิศทางวนเข็มนาฬิกา (Counterclockwise)



รูปแสดง ความแปรผันของสัญญาณของความเร็ว 4,932 RPM ในทิศทางตามเข็มนาฬิกา (Clockwise)

ซึ่งผลการทดลองพบว่า คาบเวลา (Time Period) ในการหมุนในทิศทางมีค่าเท่ากัน แต่มีความแตกต่างที่ Phase หรือมุมไฟฟ้าของสัญญาณที่เกิดขึ้น

เนื่องจากความแตกต่างใน Phase ที่เกิดขึ้นระหว่างการหมุนไปในทิศทางตามเข็มนาฬิกาและวนเข็มนาฬิกา คณะผู้จัดทำจึงเลือกที่จะเก็บข้อมูลเฉพาะในทิศทาง ทวนเข็มนาฬิกา (Counterclockwise) เท่านั้น โดยจะทำการบันทึกค่าของ RPM ที่ 20%, 40%, 60%, 80%, และ 100% จากค่าความเร็วสูงสุด (Max App Speed) ที่ 9,864 RPM เท่านั้น

1.10.2. การทดลองหาคาบเวลา (Time Period) ของ $20\% \text{ RPM} = 1,973 \text{ RPM}$

คณะผู้จัดทำได้ควบคุมการทำงานของมอเตอร์ BLDC ผ่านโปรแกรม Motor Pilot โดยตั้งค่า Revolutions Per Minute (RPM) เพื่อกำหนดความเร็วรอบของมอเตอร์ ซึ่ง RPM เป็นตัวแปรสำคัญในการควบคุมความเร็วรอบของมอเตอร์ในแต่ละช่วงการทำงาน

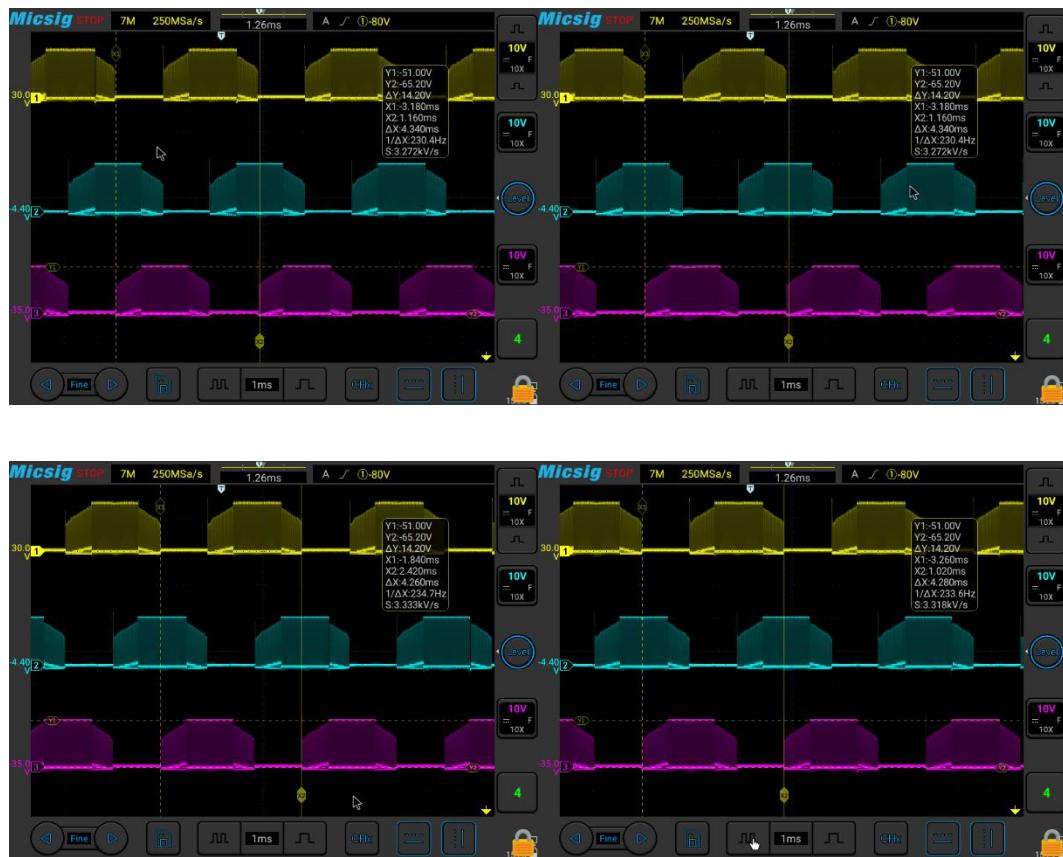
ในการทดลองนี้ได้ตั้งค่า Max App Speed ของมอเตอร์ไว้ที่ 9,864 RPM ซึ่งเป็นความเร็วสูงสุดที่มอเตอร์สามารถทำงานได้ จานนี้ได้ทำการทดลองโดยตั้งค่าความเร็วของมอเตอร์ที่ $20\% \text{ RPM} = 1,973 \text{ RPM}$ การทดสอบดำเนินการในทวนเข็มนาฬิกา (Counterclockwise)

โดยได้เก็บข้อมูลสัญญาณ RPM จำนวน 10 ค่า ต่อทิศทางการหมุน โดยผลการทดลองแสดงดังรูป และทำการวัดคาบเวลา (Time Period) ใน 1 ลูกคลื่น (Cycle)

- การตั้งค่าแกน X (เวลาหรือ Time Axis) โดยมีความละเอียดที่เหมาะสมในช่วงเวลา 1 ms เพื่อศึกษาลักษณะของสัญญาณในแต่ละช่วงของ 1 Cycle ของมอเตอร์ เช่น ตำแหน่ง Zero Crossing, Floating Phase และ Conducting Phase การตั้งค่าความละเอียดในแนวแกน X ช่วยให้สามารถแยกแยะรายละเอียดต่าง ๆ ของสัญญาณ 3 เฟสได้อย่างชัดเจนในช่วงเวลาที่เหมาะสมกับความถี่ของสัญญาณที่ตรวจสอบ
- การตั้งค่าแกน Y (แรงดันไฟฟ้า หรือ Voltage Axis) แกน Y ถูกตั้งให้แสดงค่าความแรงดันไฟฟ้าของสัญญาณในหน่วยโวลต์ (V) โดยคณะผู้จัดทำได้ปรับค่าความละเอียดให้เหมาะสมกับช่วงแรงดันที่ใช้ในมอเตอร์ โดยตั้งค่าที่ 10 V ต่อช่อง เพื่อให้สามารถแสดงระดับแรงดันไฟฟ้าในแต่ละเฟส (U, V, W) ได้อย่างชัดเจนและง่ายต่อการเปรียบเทียบ
- การปรับตำแหน่ง Trigger: เนื่องจากสัญญาณที่แสดงบนหน้าจอของ Oscilloscope จะเคลื่อนที่ในแนวแกน X อย่างต่อเนื่อง คณะผู้จัดทำจึงได้ทำการปรับตำแหน่ง Trigger ของสัญญาณแต่ละ Channel เพื่อให้สัญญาณที่แสดงไม่เคลื่อนที่
- การใช้เครื่องมือ Cursor ใน Oscilloscope: ช่วยในการวัด ความสูง (Amplitude) และ ความกว้าง (Width) ของลูกคลื่นทำได้แม่นยำและสะดวก โดยการวัด

และตรวจสอบว่าค่า RPM ที่คำนวณได้มีความสอดคล้องหรือไม่กับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์หรือไม่ รวมถึงหากค่าความคลาดเคลื่อนของข้อมูลที่เกิดขึ้น โดยใช้การคำนวณค่า RPM ของมอเตอร์ BLDC



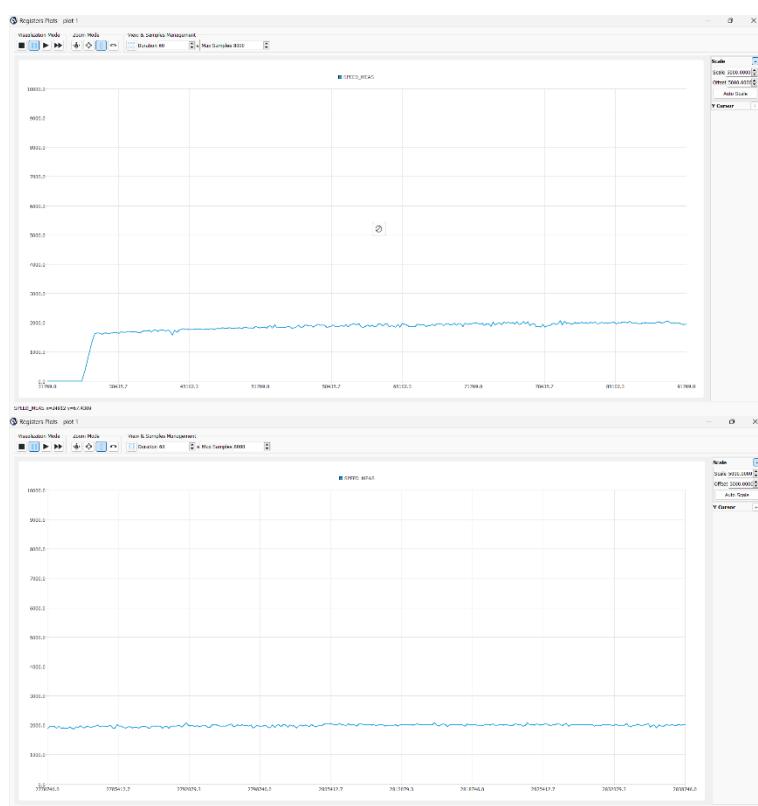


รูปแสดง ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 20% RPM = 1,973 RPM

จำนวนครั้งที่ทำ การทดสอบ	ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 20% ในทิศทางวนเข็มนาฬิกา (Counterclockwise) หน่วย Hz
1	232.6
2	227.3
3	232.6
4	232.6
5	234.7
6	237.0
7	230.4
8	230.4
9	234.7
10	233.6
ค่าเฉลี่ยข้อมูล	232.59

- คำนวณค่า RPM จากค่า czas เป็นเวลา (Time Period) ของมอเตอร์ที่ 100% ได้ 1,993.62 RPM
- ค่าความคลาดเคลื่อนของ RPM ที่เกิดขึ้น เมื่อเปรียบเทียบกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์ คือ -20.62 RPM

โดยจากการวิเคราะห์ค่าเฉลี่ยของข้อมูลที่เก็บได้ทั้งหมด 10 ค่า สามารถสังเกตได้ว่าค่ามีความแปรผันของสัญญาณ โดยความอาจเกิดจากความไม่เสถียรของเซ็นเซอร์หรือการทำงานของมอเตอร์ในขณะนั้น ซึ่งทำให้สัญญาณมีการแกว่งเล็กน้อยดังรูป สัญญาณเหล่านี้อาจได้รับผลกระทบจากสัญญาณรบกวนหรือการกรองสัญญาณที่ไม่ดี ทำให้มีความแปรผันในข้อมูลที่วัดได้



รูปแสดง ความแปรผันของสัญญาณของความเร็ว 1,973 RPM

1.10.3. การทดลองหาคาบเวลา (Time Period) ของ 40% RPM = 3,946 RPM
คณะผู้จัดทำได้ควบคุมการทำงานของมอเตอร์ BLDC ผ่านโปรแกรม Motor Pilot โดยตั้งค่า Revolutions Per Minute (RPM) เพื่อกำหนดความเร็วรอบของมอเตอร์ ซึ่ง RPM เป็นตัวแปรสำคัญในการควบคุมความเร็วรอบของมอเตอร์ในแต่ละช่วงการทำงาน

ในการทดลองนี้ได้ตั้งค่า Max App Speed ของมอเตอร์ไว้ที่ 9,864 RPM ซึ่งเป็นความเร็วสูงสุดที่มอเตอร์สามารถทำงานได้ จากนั้นได้ทำการทดลองโดยตั้งค่าความเร็วของมอเตอร์ที่ 40% RPM = 3,946 RPM การทดสอบดำเนินการในทวนเข็มนาฬิกา (Counterclockwise)

โดยได้เก็บข้อมูลสัญญาณ RPM จำนวน 10 ค่า ต่อทิศทางการหมุน โดยผลการทดลองแสดงดังรูป และทำการวัดคาบเวลา (Time Period) ใน 1 ลูปคลื่น (Cycle)

- การตั้งค่าแกน X (เวลาหรือ Time Axis) โดยมีความละเอียดที่เหมาะสมในช่วงเวลา 500 μs เพื่อศึกษาลักษณะของสัญญาณในแต่ละช่วงของ 1 Cycle ของมอเตอร์ เช่น ตำแหน่ง Zero Crossing, Floating Phase และ Conducting Phase การตั้งค่าความละเอียดในแนวแกน X ช่วยให้สามารถแยกรายละเอียดต่าง ๆ ของสัญญาณ 3 เฟสได้อย่างชัดเจนในช่วงเวลาที่เหมาะสมกับความถี่ของสัญญาณที่ตรวจสอบ
- การตั้งค่าแกน Y (แรงดันไฟฟ้า หรือ Voltage Axis) แกน Y ถูกตั้งให้แสดงค่าความแรงดันไฟฟ้าของสัญญาณในหน่วยโวลต์ (V) โดยคณะผู้จัดทำได้ปรับค่าความละเอียดให้เหมาะสมกับช่วงแรงดันที่ใช้ในมอเตอร์ โดยตั้งค่าที่ 10 V ต่อช่อง เพื่อให้สามารถแสดงระดับแรงดันไฟฟ้าในแต่ละเฟส (U, V, W) ได้อย่างชัดเจนและง่ายต่อการเปรียบเทียบ
- การปรับตำแหน่ง Trigger: เนื่องจากสัญญาณที่แสดงบนหน้าจอของ Oscilloscope จะเคลื่อนที่ในแนวแกน X อย่างต่อเนื่อง คณะผู้จัดทำจึงได้ทำการปรับตำแหน่ง Trigger ของสัญญาณแต่ละ Channel เพื่อให้สัญญาณที่แสดงไม่เคลื่อนที่
- การใช้เครื่องมือ Cursor ใน Oscilloscope: ช่วยในการวัด ความสูง (Amplitude) และ ความกว้าง (Width) ของลูปคลื่นทำได้แม่นยำและสะดวก โดยการวัดและตรวจสอบว่าค่า RPM ที่คำนวณได้มีความสอดคล้องหรือใกล้เคียงกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์หรือไม่ รวมถึงหากค่าความคลาดเคลื่อนของข้อมูลที่เกิดขึ้น โดยใช้การคำนวณค่า RPM ของมอเตอร์ BLDC



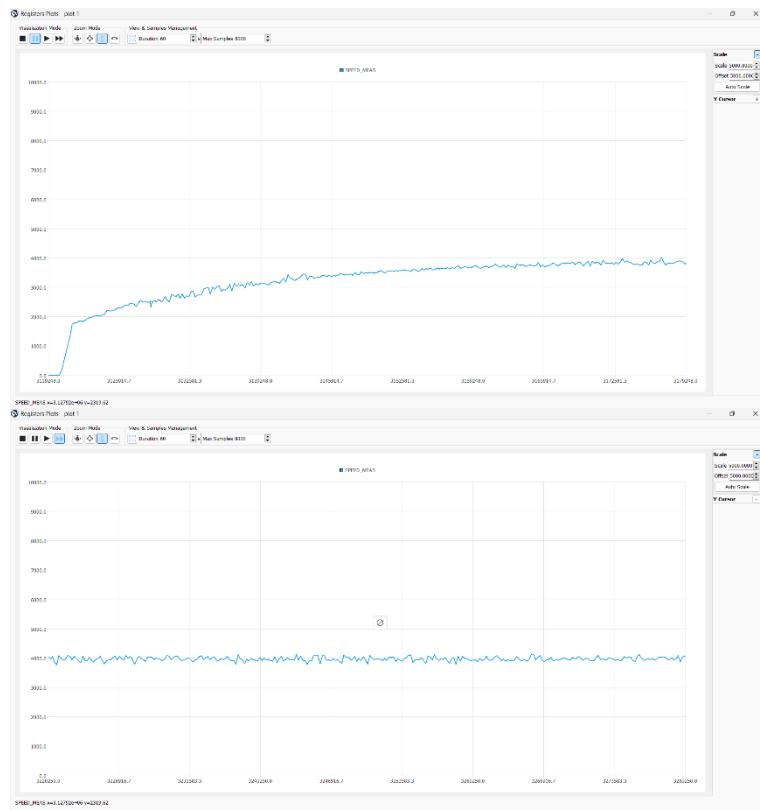


รูปแสดง ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 40% RPM = 3,946 RPM

จำนวนครั้งที่ทำ การทดสอบ	ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 40% ในทิศทางวนเข็มนาฬิกา (Counterclockwise) หน่วย HZ
1	454.5
2	460.8
3	454.5
4	458.7
5	456.6
6	460.8
7	467.3
8	460.8
9	460.8
10	460.8
ค่าเฉลี่ยข้อมูล	459.56

- คำนวณค่า RPM จากค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% ได้ 3,939.09 RPM
- ค่าความคลาดเคลื่อนของ RPM ที่เกิดขึ้น เมื่อเปรียบเทียบกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์ คือ 6.91 RPM

โดยจากการวิเคราะห์ค่าเฉลี่ยของข้อมูลที่เก็บได้ทั้งหมด 10 ค่า สามารถสังเกตได้ว่าค่ามีความแปรผวนของสัญญาณ โดยความอาจเกิดจากความไม่เสถียรของเซ็นเซอร์หรือการทำงานของมอเตอร์ในขณะนั้น ซึ่งทำให้สัญญาณมีการแก่งเล็กน้อยดังรูป สัญญาณเหล่านี้อาจได้รับผลกระทบจากสัญญาณรบกวนหรือการกรองสัญญาณที่ไม่ดี ทำให้มีความแปรผวนในข้อมูลที่วัดได้



รูปแสดง ความแปรผันของสัญญาณของความเร็ว 3,946 RPM

1.10.4. การทดลองหาคาบเวลา (Time Period) ของ 60% RPM = 5,918 RPM
คณะผู้จัดทำได้ควบคุมการทำงานของมอเตอร์ BLDC ผ่านโปรแกรม Motor Pilot โดยตั้งค่า Revolutions Per Minute (RPM) เพื่อกำหนดความเร็วรอบของมอเตอร์ ซึ่ง RPM เป็นตัวแปรสำคัญในการควบคุมความเร็วรอบของมอเตอร์ในแต่ละช่วงการทำงาน

ในการทดลองนี้ได้ตั้งค่า Max App Speed ของมอเตอร์ไว้ที่ 9,864 RPM ซึ่งเป็นความเร็วสูงสุดที่มอเตอร์สามารถทำงานได้ จากนั้นได้ทำการทดลองโดยตั้งค่าความเร็วของมอเตอร์ที่ 60% RPM = 5,918 RPM การทดสอบดำเนินการในทวนเข็มนาฬิกา (Counterclockwise)

โดยได้เก็บข้อมูลสัญญาณ RPM จำนวน 10 ค่า ต่อทิศทางการหมุน โดยผลการทดลองแสดงดังรูป และทำการวัดคาบเวลา (Time Period) ใน 1 ลูปคลื่น (Cycle)

- การตั้งค่าแกน X (เวลาหรือ Time Axis) โดยมีความละเอียดที่เหมาะสมในช่วงเวลา 500 μs เพื่อศึกษาลักษณะของสัญญาณในแต่ละช่วงของ 1 Cycle ของมอเตอร์ เช่น ตำแหน่ง Zero Crossing, Floating Phase และ Conducting Phase การตั้งค่าความละเอียดในแนวแกน X ช่วยให้สามารถแยกรายละเอียดต่าง ๆ ของสัญญาณ 3 เฟสได้อย่างชัดเจนในช่วงเวลาที่เหมาะสมกับความถี่ของสัญญาณที่ตรวจสอบ
- การตั้งค่าแกน Y (แรงดันไฟฟ้า หรือ Voltage Axis) แกน Y ถูกตั้งให้แสดงค่าความแรงดันไฟฟ้าของสัญญาณในหน่วยโวลต์ (V) โดยคณะผู้จัดทำได้ปรับค่าความละเอียดให้เหมาะสมกับช่วงแรงดันที่ใช้ในมอเตอร์ โดยตั้งค่าที่ 10 V ต่อช่อง เพื่อให้สามารถแสดงระดับแรงดันไฟฟ้าในแต่ละเฟส (U, V, W) ได้อย่างชัดเจนและง่ายต่อการเปรียบเทียบ
- การปรับตำแหน่ง Trigger: เนื่องจากสัญญาณที่แสดงบนหน้าจอของ Oscilloscope จะเคลื่อนที่ในแนวแกน X อย่างต่อเนื่อง คณะผู้จัดทำจึงได้ทำการปรับตำแหน่ง Trigger ของสัญญาณแต่ละ Channel เพื่อให้สัญญาณที่แสดงไม่เคลื่อนที่
- การใช้เครื่องมือ Cursor ใน Oscilloscope: ช่วยในการวัด ความสูง (Amplitude) และ ความกว้าง (Width) ของลูปคลื่นทำได้แม่นยำและสะดวก โดยการวัดและตรวจสอบว่าค่า RPM ที่คำนวณได้มีความสอดคล้องหรือใกล้เคียงกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์หรือไม่ รวมถึงหากค่าความคลาดเคลื่อนของข้อมูลที่เกิดขึ้น โดยใช้การคำนวณค่า RPM ของมอเตอร์ BLDC





รูปแสดง ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 60% RPM = 5,918 RPM

จำนวนครั้งที่ทำ การทดสอบ	ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 60% ในทิศทางวนเข็มนาฬิกา (Counterclockwise) หน่วย Hz
1	662.3
2	694.4
3	694.4
4	694.4
5	699.3
6	680.3
7	694.4
8	689.7
9	689.7
10	699.3
ค่าเฉลี่ยข้อมูล	689.82

- คำนวณค่า RPM จากค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% ได้ 5,912.74 RPM
- ค่าความคลาดเคลื่อนของ RPM ที่เกิดขึ้น เมื่อเปรียบเทียบกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์ คือ 5.26 RPM

โดยจากการวิเคราะห์ค่าเฉลี่ยของข้อมูลที่เก็บได้ทั้งหมด 10 ค่า สามารถสังเกตได้ว่าค่ามีความแปรผวนของสัญญาณ โดยความอาจเกิดจากความไม่เสถียรของเซ็นเซอร์หรือการทำงานของมอเตอร์ในขณะนั้น ซึ่งทำให้สัญญาณมีการแก่งเล็กน้อยตั้งรูป สัญญาณเหล่านี้อาจได้รับผลกระทบจากสัญญาณรบกวนหรือการกรองสัญญาณที่ไม่ดี ทำให้มีความแปรผวนในข้อมูลที่วัดได้



รูปแสดง ความแปรผันของสัญญาณของความเร็ว 5,918 RPM

1.10.5. การทดลองหาคาบเวลา (Time Period) ของ 80% RPM = 7,891 RPM
คณะผู้จัดทำได้ควบคุมการทำงานของมอเตอร์ BLDC ผ่านโปรแกรม Motor Pilot โดยตั้งค่า Revolutions Per Minute (RPM) เพื่อกำหนดความเร็วรอบของมอเตอร์ ซึ่ง RPM เป็นตัวแปรสำคัญในการควบคุมความเร็วรอบของมอเตอร์ในแต่ละช่วงการทำงาน

ในการทดลองนี้ ได้ตั้งค่า Max App Speed ของมอเตอร์ไว้ที่ 9,864 RPM ซึ่งเป็นความเร็วสูงสุดที่มอเตอร์สามารถทำงานได้ จากนั้นได้ทำการทดลองโดยตั้งค่าความเร็วของมอเตอร์ที่ 80% RPM = 7,891 RPM การทดสอบดำเนินการในทวนเข็มนาฬิกา (Counterclockwise)

โดยได้เก็บข้อมูลสัญญาณ RPM จำนวน 10 ค่า ต่อทิศทางการหมุน โดยผลการทดลองแสดงดังรูป และทำการวัดคาบเวลา (Time Period) ใน 1 ลูปคลื่น (Cycle)

- การตั้งค่าแกน X (เวลาหรือ Time Axis) โดยมีความละเอียดที่เหมาะสมในช่วงเวลา 200 μs เพื่อศึกษาลักษณะของสัญญาณในแต่ละช่วงของ 1 Cycle ของมอเตอร์ เช่น ตำแหน่ง Zero Crossing, Floating Phase และ Conducting Phase การตั้งค่าความละเอียดในแนวแกน X ช่วยให้สามารถแยกรายละเอียดต่าง ๆ ของสัญญาณ 3 เฟสได้อย่างชัดเจนในช่วงเวลาที่เหมาะสมกับความถี่ของสัญญาณที่ตรวจสอบ
- การตั้งค่าแกน Y (แรงดันไฟฟ้า หรือ Voltage Axis) แกน Y ถูกตั้งให้แสดงค่าความแรงดันไฟฟ้าของสัญญาณในหน่วยโวลต์ (V) โดยคณะผู้จัดทำได้ปรับค่าความละเอียดให้เหมาะสมกับช่วงแรงดันที่ใช้ในมอเตอร์ โดยตั้งค่าที่ 10 V ต่อช่อง เพื่อให้สามารถแสดงระดับแรงดันไฟฟ้าในแต่ละเฟส (U, V, W) ได้อย่างชัดเจนและง่ายต่อการเปรียบเทียบ
- การปรับตำแหน่ง Trigger: เนื่องจากสัญญาณที่แสดงบนหน้าจอของ Oscilloscope จะเคลื่อนที่ในแนวแกน X อย่างต่อเนื่อง คณะผู้จัดทำจึงได้ทำการปรับตำแหน่ง Trigger ของสัญญาณแต่ละ Channel เพื่อให้สัญญาณที่แสดงไม่เคลื่อนที่
- การใช้เครื่องมือ Cursor ใน Oscilloscope: ช่วยในการวัด ความสูง (Amplitude) และ ความกว้าง (Width) ของลูปคลื่นทำได้แม่นยำและสะดวก โดยการวัด และตรวจสอบว่าค่า RPM ที่คำนวณได้มีความสอดคล้องหรือใกล้เคียงกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์หรือไม่ รวมถึงหากค่าความคลาดเคลื่อนของข้อมูลที่เกิดขึ้น โดยใช้การคำนวณค่า RPM ของมอเตอร์ BLDC



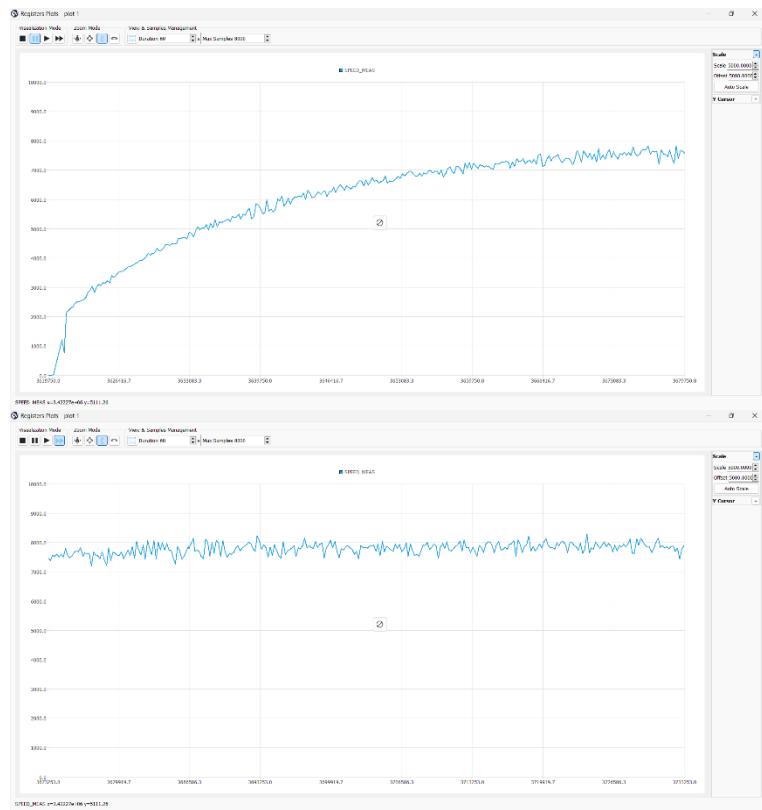


รูปแสดง ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 80% RPM = 7,891 RPM

จำนวนครั้งที่ทำ การทดสอบ	ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 80% ในทิศทางวนเข็มนาฬิกา (Counterclockwise) หน่วย Hz
1	883.4
2	912.4
3	902.5
4	909.1
5	912.4
6	905.8
7	909.1
8	905.8
9	912.4
10	905.8
ค่าเฉลี่ยข้อมูล	905.87

- คำนวณค่า RPM จากค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% ได้ 7,764.60 RPM
- ค่าความคลาดเคลื่อนของ RPM ที่เกิดขึ้น เมื่อเปรียบเทียบกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์ คือ 126.40 RPM

โดยจากการวิเคราะห์ค่าเฉลี่ยของข้อมูลที่เก็บได้ทั้งหมด 10 ค่า สามารถสังเกตได้ว่าค่ามีความแปรผวนของสัญญาณ โดยความอาจเกิดจากความไม่เสถียรของเซ็นเซอร์หรือการทำงานของมอเตอร์ในขณะนั้น ซึ่งทำให้สัญญาณมีการแก่งเล็กน้อยดังรูป สัญญาณเหล่านี้อาจได้รับผลกระทบจากสัญญาณรบกวนหรือการกรองสัญญาณที่ไม่ดี ทำให้มีความแปรผวนในข้อมูลที่วัดได้



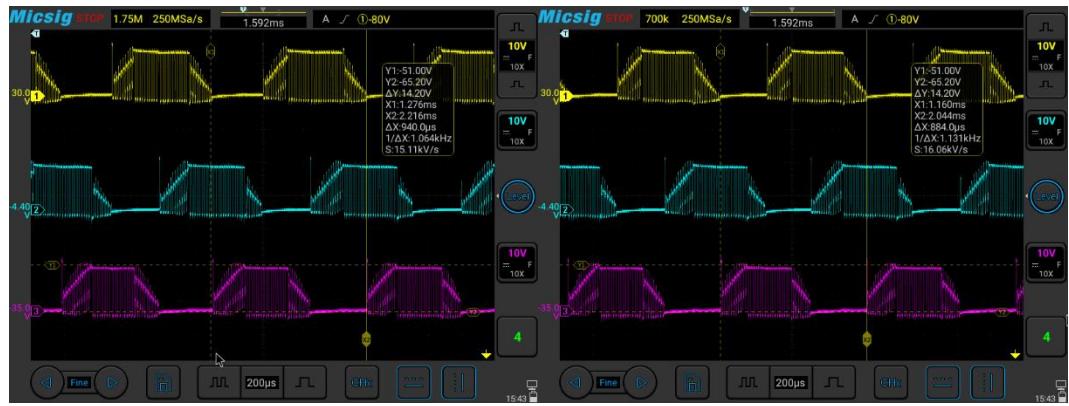
รูปแสดง ความแปรผันของสัญญาณของความเร็ว 7,891 RPM

1.10.6. การทดลองหาคาบเวลา (Time Period) ของ 100% RPM = 9,864 RPM
คณะผู้จัดทำได้ควบคุมการทำงานของมอเตอร์ BLDC ผ่านโปรแกรม Motor Pilot โดยตั้งค่า Revolutions Per Minute (RPM) เพื่อกำหนดความเร็วรอบของมอเตอร์ ซึ่ง RPM เป็นตัวแปรสำคัญในการควบคุมความเร็วรอบของมอเตอร์ในแต่ละช่วงการทำงาน

ในการทดลองนี้ ได้ตั้งค่า Max App Speed ของมอเตอร์ไว้ที่ 9,864 RPM ซึ่งเป็นความเร็วสูงสุดที่มอเตอร์สามารถทำงานได้ จากนั้นได้ทำการทดลองโดยตั้งค่าความเร็วของมอเตอร์ที่ 100% RPM = 9,864 RPM การทดสอบดำเนินการในทวนเข็มนาฬิกา (Counterclockwise)

โดยได้เก็บข้อมูลสัญญาณ RPM จำนวน 10 ค่า ต่อทิศทางการหมุน โดยผลการทดลองแสดงดังรูป และทำการวัดคาบเวลา (Time Period) ใน 1 ลูปคลื่น (Cycle)

- การตั้งค่าแกน X (เวลาหรือ Time Axis) โดยมีความละเอียดที่เหมาะสมในช่วงเวลา 200 μs เพื่อศึกษาลักษณะของสัญญาณในแต่ละช่วงของ 1 Cycle ของมอเตอร์ เช่น ตำแหน่ง Zero Crossing, Floating Phase และ Conducting Phase การตั้งค่าความละเอียดในแนวแกน X ช่วยให้สามารถแยกรายละเอียดต่าง ๆ ของสัญญาณ 3 เฟสได้อย่างชัดเจนในช่วงเวลาที่เหมาะสมกับความถี่ของสัญญาณที่ตรวจสอบ
- การตั้งค่าแกน Y (แรงดันไฟฟ้า หรือ Voltage Axis) แกน Y ถูกตั้งให้แสดงค่าความแรงดันไฟฟ้าของสัญญาณในหน่วยโวลต์ (V) โดยคณะผู้จัดทำได้ปรับค่าความละเอียดให้เหมาะสมกับช่วงแรงดันที่ใช้ในมอเตอร์ โดยตั้งค่าที่ 10 V ต่อช่อง เพื่อให้สามารถแสดงระดับแรงดันไฟฟ้าในแต่ละเฟส (U, V, W) ได้อย่างชัดเจนและง่ายต่อการเปรียบเทียบ
- การปรับตำแหน่ง Trigger: เนื่องจากสัญญาณที่แสดงบนหน้าจอของ Oscilloscope จะเคลื่อนที่ในแนวแกน X อย่างต่อเนื่อง คณะผู้จัดทำจึงได้ทำการปรับตำแหน่ง Trigger ของสัญญาณแต่ละ Channel เพื่อให้สัญญาณที่แสดงไม่เคลื่อนที่
- การใช้เครื่องมือ Cursor ใน Oscilloscope: ช่วยในการวัด ความสูง (Amplitude) และ ความกว้าง (Width) ของลูปคลื่นทำได้แม่นยำและสะดวก โดยการวัด และตรวจสอบว่าค่า RPM ที่คำนวณได้มีความสอดคล้องหรือใกล้เคียงกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์หรือไม่ รวมถึงหากค่าความคลาดเคลื่อนของข้อมูลที่เกิดขึ้น โดยใช้การคำนวณค่า RPM ของมอเตอร์ BLDC



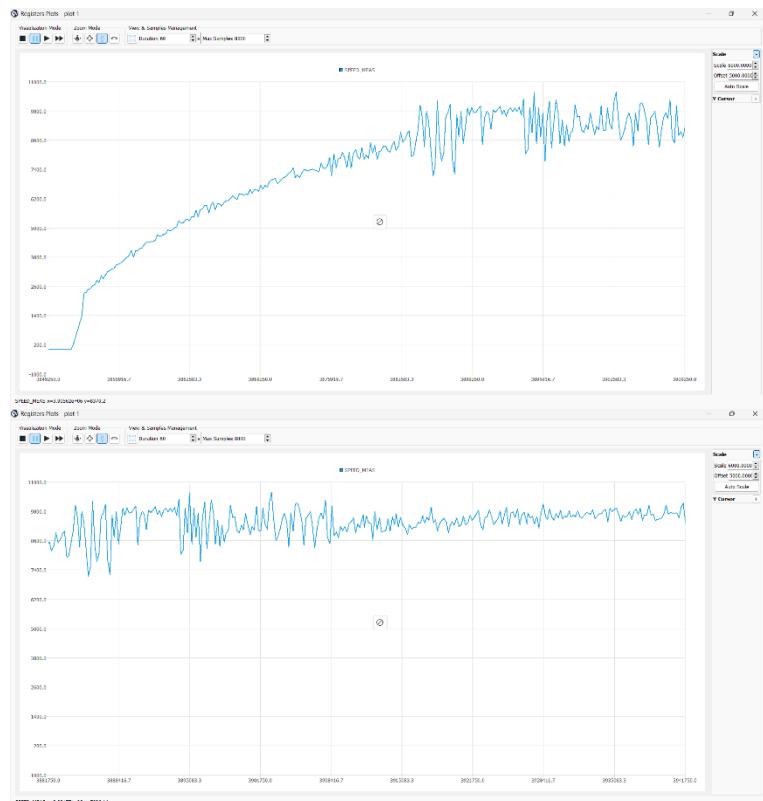


รูปแสดง ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% RPM = 9,864 RPM

จำนวนครั้งที่ทำการทดสอบ	ค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% ในทิศทางวนเข็มนาฬิกา (Counterclockwise) หน่วย HZ
1	1064
2	1131
3	1126
4	1142
5	1136
6	1136
7	1168
8	1126
9	1126
10	1126
ค่าเฉลี่ยข้อมูล	1128.1

- คำนวณค่า RPM จากค่าคาบเวลา (Time Period) ของมอเตอร์ที่ 100% ได้ 9,669.43 RPM
- ค่าความคลาเดคลื่อนของ RPM ที่เกิดขึ้น เมื่อเปรียบเทียบกับค่า RPM ที่ตั้งค่าไว้สำหรับการควบคุมมอเตอร์ คือ 194.57 RPM

โดยจากการวิเคราะห์ค่าเฉลี่ยของข้อมูลที่เก็บได้ทั้งหมด 10 ค่า สามารถสังเกตได้ว่าค่ามีความแปรผันของสัญญาณ โดยความอาจเกิดจากความไม่เสถียรของเซ็นเซอร์หรือการทำงานของมอเตอร์ในขณะนั้น ซึ่งทำให้สัญญาณมีการแก่งเล็กน้อยดังรูป สัญญาณเหล่านี้อาจได้รับผลกระทบจากสัญญาณรบกวนหรือการกรองสัญญาณที่ไม่ดี ทำให้มีความแปรผันในข้อมูลที่วัดได้



รูปแสดง ความแปรผันของสัญญาณของความเร็ว 9,864 RPM

1.11. สรุปผล

การทดลองนี้มุ่งเน้นการทดสอบและวิเคราะห์ความสัมพันธ์ระหว่างความเร็วรอบ (RPM) ของมอเตอร์ BLDC และสัญญาณไฟฟ้า 3 เฟสที่เกิดขึ้นในระบบควบคุมมอเตอร์ โดยใช้โปรแกรม STM32 Motor Control Workbench 6.3.2 สำหรับการตั้งค่าความเร็วและควบคุมมอเตอร์ในโหมด 6-Step (Trapezoidal Control) พร้อมทั้งบันทึกสัญญาณและคำนวณค่า RPM ที่ได้จากการวัดคาบเวลา (Time Period) ในแต่ละค่าความเร็วรอบ

การตั้งค่าระบบและการควบคุมมอเตอร์

- มอเตอร์ BLDC A2212 13T ถูกตั้งค่าความเร็วสูงสุดที่ 9,864 RPM โดยใช้บอร์ด NUCLEO-G474RE เป็น Control Board
- การทดลองเริ่มต้นที่ 20% RPM (1,973 RPM) เนื่องจากมอเตอร์ไม่สามารถหมุนได้อย่างต่อเนื่องที่ความเร็วต่ำกว่านี้ โดยทดสอบที่ 50% RPM (4,932 RPM) ในทั้งสองทิศทาง (ตามเข็มนาฬิกาและวนเข็มนาฬิกา) และพบว่าคาบเวลาในทั้งสองทิศทางมีค่าเท่ากัน แต่มีความแตกต่างในมุมไฟฟ้าของสัญญาณไฟฟ้า
- ขณะผู้จัดทำจึงเลือกเก็บข้อมูลเฉพาะในทิศทางทวนเข็มนาฬิกา

การทดสอบสัญญาณไฟฟ้า 3 เฟส

- การใช้ Oscilloscope เพื่อวัดสัญญาณ 3 เฟสจากมอเตอร์ช่วงให้เห็นลำดับการทำงานของมอเตอร์ในแต่ละเฟส
- ความถี่ของสัญญาณไฟฟ้า 3 เฟสมีความสัมพันธ์เชิงเส้นกับ RPM โดยเมื่อคำนวณ RPM จากการวัดคาบเวลาใน 1 Cycle ของสัญญาณพบว่า RPM ที่คำนวณตรงกับค่าที่ตั้งไว้ในโปรแกรม

การหาความคลาดเคลื่อนของ RPM

- ผลการทดลองแสดงให้เห็นค่าความคลาดเคลื่อนระหว่าง RPM ที่คำนวณจากค่าคาบเวลา กับ RPM ที่ตั้งค่าไว้ในการควบคุมมอเตอร์ โดยค่าความคลาดเคลื่อนมีค่าตั้งแต่ -25.80 RPM ถึง 194.57 RPM ขึ้นอยู่กับค่าความเร็วที่ตั้งไว้ (20% - 100% RPM)
- ความคลาดเคลื่อนในแต่ละช่วงของ RPM นั้นแสดงให้เห็นถึงความแม่นยำในการควบคุมความเร็วรอบของมอเตอร์ที่มีความแตกต่างกันตามช่วงความเร็วที่ทดสอบ

การวิเคราะห์ผล

- โดยรวมแล้ว การทดลองนี้ได้แสดงให้เห็นถึงการทำงานของมอเตอร์ BLDC ในแต่ละช่วง RPM และการวัดคาบเวลา (Time Period) ของมอเตอร์ในทั้งสองทิศทางการหมุน
- แม้ว่าค่าความคลาดเคลื่อนจะมีการแปรผัน แต่ผลการทดลองยังคงแสดงถึงการควบคุมที่มีความแม่นยำสูงในกรณีที่ทดสอบที่ความเร็วสูงขึ้น (ตั้งแต่ 40% ถึง 100% RPM)

- สัญญาณที่ได้จาก Oscilloscope ช่วยให้สามารถวิเคราะห์และประเมินการทำงานของมอเตอร์ในแต่ละเฟสได้อย่างมีประสิทธิภาพ

1.12. ภัณฑ์

จากการทดลองที่ได้ทำการวัดความเร็วรอบ (RPM) ของมอเตอร์ BLDC ในทิศทางต่าง ๆ ตามการตั้งค่า RPM ที่แตกต่างกัน พบว่าโดยรวมแล้วมอเตอร์มีการทำงานที่ค่อนข้างเสถียร แม้ว่าจะมีความคลาดเคลื่อนของ RPM เมื่อเปรียบเทียบกับค่าที่ตั้งไว้ แต่ความคลาดเคลื่อนที่เกิดขึ้นสามารถยอมรับได้เนื่องจากมีการแปรผันเล็กน้อยในค่าที่วัดได้ ทั้งนี้การวัดควบเวลาในแต่ละ RPM ได้แสดงถึงความสัมพันธ์ระหว่างเวลาการหมุนและความเร็วรอบของมอเตอร์ที่คำนวนได้จากค่าควบเวลา (Time Period) ซึ่งมีความสัมพันธ์เชิงเส้นกับการเปลี่ยนแปลงของ RPM อย่างชัดเจน

ผลการทดลองพบว่าในทิศทางการหมุนตามเข็มนาฬิกาและทวนเข็มนาฬิกามีควบเวลาที่เท่ากัน แต่ความแตกต่างของมุมเฟส (Phase) ของสัญญาณ 3 เฟสที่เกิดขึ้นนั้นแสดงให้เห็นถึงการเปลี่ยนแปลงของกระแสไฟฟ้าที่ส่งไปยังมอเตอร์ในแต่ละทิศทาง แม้ว่าค่าควบเวลาในแต่ละทิศทางจะเท่ากัน แต่มีความแตกต่างที่มุมเฟส ซึ่งเป็นผลจากการเปลี่ยนทิศทางการหมุนของมอเตอร์ นอกจากนี้การเลือกเก็บข้อมูลเฉพาะในทิศทางทวนเข็มนาฬิกา (Counterclockwise) ช่วยให้สามารถศึกษาลักษณะของสัญญาณได้อย่างละเอียดและมีความสอดคล้องกับผลที่ต้องการ

ความคลาดเคลื่อนของ RPM ที่เกิดขึ้นระหว่างการคำนวนจากควบเวลาและค่าที่ตั้งไว้ในโปรแกรมมีการแปรผันตามเปอร์เซ็นต์ของ RPM เช่น ในการทดลองที่ 20% RPM พบว่าค่าความคลาดเคลื่อนอยู่ที่ -20.62 RPM ซึ่งเป็นค่าคลาดเคลื่อนที่มีขนาดเล็ก ขณะที่ใน 100% RPM ค่าความคลาดเคลื่อนอยู่ที่ 194.57 RPM ซึ่งแสดงให้เห็นถึงผลกระทบของการตั้งค่า RPM ที่สูงขึ้น ซึ่งทำให้เกิดความผิดพลาดที่มากขึ้น การวัดควบเวลาและการคำนวนค่า RPM จากควบเวลาที่วัดได้ช่วยในการตรวจสอบความถูกต้องของการตั้งค่าและประสิทธิภาพการทำงานของมอเตอร์

จากการทดลองนี้สามารถสรุปได้ว่าความเร็วรอบของมอเตอร์ BLDC ในระบบ 6-Step Trapezoidal Control สามารถควบคุมได้ตามค่าที่ตั้งไว้ในโปรแกรม โดยมีความคลาดเคลื่อนที่ยอมรับได้ในแต่ละช่วงการทำงาน และมีการแปรผันในมุมเฟสของสัญญาณ 3 เฟสที่ควรจะพิจารณาในการออกแบบระบบควบคุมมอเตอร์ในอนาคต เพื่อให้มีประสิทธิภาพและความแม่นยำที่สูงขึ้น

นอกจากนี้ยังพบว่าเมื่อเพิ่มค่า RPM ของมอเตอร์ให้สูงขึ้น จะเกิดแรงดันย้อนกลับ (Back Electromotive Force หรือ Back EMF) มากขึ้นตามไปด้วย ซึ่ง Back EMF นี้เป็นแรงดันที่เกิดจากการหมุนของโรเตอร์ในมอเตอร์ BLDC และมีผลโดยตรงต่อการทำงานของวงจรควบคุมมอเตอร์ การเพิ่มขึ้นของ Back EMF ใน RPM ที่สูงขึ้นอาจส่งผลต่อประสิทธิภาพและความเสถียรของมอเตอร์ ซึ่งเป็นปัจจัยที่ควรคำนึงถึงในการออกแบบและควบคุมการทำงานในระบบจริง โดยเฉพาะในช่วงที่ RPM สูง ซึ่งอาจทำให้เกิดการสะท้อนกลับของกระแสไฟฟ้าที่ส่งผลให้วงจร

ควบคุมทำงานไม่เสถียร นอกจากนี้ยังอาจส่งผลต่อการใช้พลังงานและการระบายความร้อนของมอเตอร์ ทำให้ต้องมีการควบคุมและออกแบบวงจรให้สามารถรองรับแรงดันย้อนกลับนี้ได้อย่างเหมาะสม

1.13. ข้อเสนอแนะ

1.13.1. การพัฒนาความแม่นยำของการตั้งค่า RPM: ถึงแม้ผลการทดลองจะพบว่าความคลาดเคลื่อนของ RPM สามารถรับได้ในระดับหนึ่ง แต่ยังมีความแตกต่างที่สังเกตได้เมื่อ RPM สูงขึ้น จึงควรมีการพัฒนาวิธีการควบคุมความเร็วรอบที่มีความแม่นยำมากขึ้น เช่น การใช้ระบบควบคุมที่มีความละเอียดสูงขึ้น หรือการปรับแต่งค่าพารามิเตอร์ในการตั้งค่า RPM เพื่อให้เกิดความเสถียรและความแม่นยำที่ดียิ่งขึ้น

1.13.2. การออกแบบวงจรควบคุมที่รองรับแรงดันย้อนกลับ (Back EMF): ในการเพิ่มค่า RPM ของมอเตอร์ ควรมีการออกแบบวงจรควบคุมที่สามารถรองรับแรงดันย้อนกลับที่เกิดจากการหมุนของมอเตอร์ได้ โดยเฉพาะในช่วง RPM สูง ซึ่งอาจทำให้เกิดการสะท้อนกลับของกระแสไฟฟ้าและส่งผลต่อความเสถียรของมอเตอร์ จึงควรมีการพัฒนาวงจรที่มีฟังก์ชันในการลดหรือควบคุมการเกิด Back EMF อย่างมีประสิทธิภาพ เช่น การใช้วงจรกรองหรือวงจรที่ช่วยลดการสะท้อนกลับในช่วงที่ RPM สูง

1.14. เอกสารอ้างอิง

TWMotor. How Brushless DC Motors Work. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://th.twmotor.net/info/how-brushless-dc-motors-work-97376907.html>

9Engineer. บทความที่เกี่ยวข้อง. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from https://www.9engineer.com/index.php?m=article&a=show&article_id=2606

All About Circuits. Sensorless Brushless DC (BLDC) Motor Control. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://www.allaboutcircuits.com/technical-articles/sensorless-brushless-dc-bldc-motor-control/>

Texas Instruments E2E Forum. DRV8308EVM 3-Phase BLDC Motor - Back-EMF Measurement. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://e2e.ti.com/support/motor-drivers-group/motor-drivers/f/motor-drivers-forum/827237/drv8308evm-3-phase-bldc-motor--back-emf-measurement>

About Motors. PMSM Motor Control. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://about-motors.com/motorcontrol/pmsm/>

LinkTech Thailand. Synchronous Motor (PMSM) EV Conversion. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://linktechthailand.com/portfolio/synchronous-motor-pmsm-ev-conversion/>

ResearchGate. BLDC Motor with Commutation Logic. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from https://www.researchgate.net/figure/BLDC-Motor-with-Commutation-Logic_fig1_308571505

MathWorks. Six-Step Commutation. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://www.mathworks.com/help/mcb/ref/sixstepcommutation.html>

Scribd. BLDC A2212 13T. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://www.scribd.com/document/343523843/BLDC-A2212-13T>

2. Brush DC Motor

2.1. จุดประสงค์

- คณะผู้จัดทำต้องดำเนินการตามการวิเคราะห์ผลการเรียนรู้หลักของการทดลองนี้
- คณะผู้จัดทำต้องสามารถอธิบายหลักการทำงานของ DC Motor รวมถึงความสามารถของ Motor-Torque Constant และ Back-EMF Constant ได้อย่างถูกต้อง
- คณะผู้จัดทำต้องสามารถอธิบายความสัมพันธ์ของ Speed, Torque, Current, Power, และ %Efficiency เมื่อ Load Torque ที่กระทะต่อ DC Motor มีการเปลี่ยนแปลง ผ่านการปรับ Load Torque ด้วย Magnetic Particle Clutches และเมื่อแรงดันไฟฟ้า Input เปลี่ยนแปลงไปด้วยการปรับ Duty Cycle และ Frequency ของ PWM
- คณะผู้จัดทำต้องสามารถอธิบายหลักการทำงานของ H-Bridge Drive Mode ทั้ง 3 โหมด ได้แก่ Sign-Magnitude, Locked Anti-Phase, และ Async Sign-Magnitude พร้อมทั้งควบคุม DC Motor ได้ในโหมด Sign-Magnitude และ Locked Anti-Phase
- คณะผู้จัดทำต้องสามารถอธิบายกระบวนการ Signal Conditioning และ Signal Processing ทั้งหมด ตั้งแต่การอ่านค่าจาก Incremental Encoder และ Hall Current Sensor อธิบายถึงกระบวนการคิด ขั้นตอนก่อนและหลังการ Calibrate Sensor รวมถึงวิธีการจัดการข้อมูล และหาสมการความสัมพันธ์ระหว่างกระแสไฟฟ้าที่ใช้จริงในมอเตอร์ กับแรงดันไฟฟ้าที่ออกมายจาก Hall Current Sensor พร้อมทั้งกระบวนการ Unwrap ค่า
- คณะผู้จัดทำต้องสามารถเขียนโปรแกรมโดยประยุกต์ใช้ MATLAB และ Simulink เพื่อสั่งการ หรือรับค่าร่วมกับบอร์ด Nucleo STM32G474RE โดยใช้สัญญาณจาก Incremental Encoder และ Hall Current Sensor เป็น Input และแสดงผล Output ผ่านการ Log สัญญาณใน MATLAB Simulink โดยแสดงผลแบบ Real-Time เป็นความเร็วเชิงมูน และกระแสไฟฟ้าในหน่วย SI derived.

2.2. สมมติฐาน

- หากคณะผู้จัดทำปรับแรงดันไฟฟ้า Input ที่จ่ายเข้าสู่ DC Motor ด้วยการปรับ Duty Cycle และ Frequency ของ PWM จะทำให้ความเร็วเชิงมูนของมอเตอร์เปลี่ยนแปลงอย่างเป็นสัดส่วน
- หากคณะผู้จัดทำปรับ Load Torque ด้วย Magnetic Particle Clutches จะสามารถสังเกตเห็นการเปลี่ยนแปลงของความสัมพันธ์ระหว่าง Speed, Torque, Current, Power, และ %Efficiency ของ DC Motor
- การใช้ Incremental Encoder และ Hall Current Sensor ร่วมกับการทำ Signal Conditioning และ Signal Processing ที่เหมาะสม จะช่วยให้สามารถอ่านค่าความเร็วเชิงมูนและกระแสไฟฟ้าของ DC Motor ได้อย่างแม่นยำ

- การควบคุม DC Motor ด้วยโหนด H-Bridge (Sign-Magnitude และ Locked Anti-Phase) จะส่งผลต่อพฤติกรรมการทำงานของมอเตอร์แตกต่างกัน และสามารถวิเคราะห์ได้ผ่านการเปรียบเทียบสัญญาณ Input และ Output
- การใช้ MATLAB และ Simulink ในการประมวลผลและแสดงผลแบบ Real-Time จะช่วยให้สามารถตรวจสอบความสัมพันธ์ระหว่างสัญญาณ Input และ Output ของระบบ DC Motor ได้อย่างมีประสิทธิภาพ
- การปรับค่าต่าง ๆ ในกระบวนการทดลอง เช่น การ Calibrate Sensor และการใช้ Unwrap ค่าจะช่วยลดความคลาดเคลื่อนของข้อมูลที่ได้จาก Incremental Encoder และ Hall Current Sensor

2.3. ตัวแปร

2.3.1. ตัวแปรต้น (Independent Variables)

- แรงดันไฟฟ้า Input ที่จ่ายเข้าสู่ DC Motor (ปรับผ่าน Duty Cycle และ Frequency ของ PWM)
- ค่า Load Torque ที่กระทำต่อ DC Motor (ปรับด้วย Magnetic Particle Clutches)
- โหนดการควบคุมของ H-Bridge Drive (Sign-Magnitude และ Locked Anti-Phase)

2.3.2. ตัวแปรตาม (Dependent Variables)

- ความเร็วเชิงมุม (Angular Velocity) ของ DC Motor (วัดจาก Incremental Encoder)
- กระแสไฟฟ้าที่มอเตอร์ใช้ (Current) (วัดจาก Hall Current Sensor)
- ค่า Power และ %Efficiency ของระบบ

2.3.3. ตัวแปรควบคุม (Control Variables)

- ชนิดของ DC Motor ที่ใช้ในการทดลอง
- ชนิดและคุณสมบัติของเซ็นเซอร์ (Incremental Encoder, Hall Current Sensor)
- อุปกรณ์ควบคุมที่ใช้ เช่น บอร์ด Nucleo STM32G474RE
- อุณหภูมิของห้องปฏิบัติการที่อาจมีผลต่อค่าการวัด
- โปรแกรม MATLAB และ Simulink ที่ใช้สำหรับประมวลผลและเก็บข้อมูล
- วิธีการตั้งค่าและ Calibrate Sensor ก่อนเริ่มการทดลอง

2.4. นิยามศัพท์เฉพาะ

- Motor-Torque Constant เป็นค่าคงที่ที่ใช้แสดงความสัมพันธ์ระหว่างกระแสไฟฟ้าที่จ่ายเข้าไปในมอเตอร์ (Input Current) และแรงบิด (Torque) ที่มอเตอร์สามารถผลิตได้ โดยมีหน่วยวัดเป็น นิวตัน-เมตรต่อแอม培ร์ (Nm/A) ค่านี้เป็นคุณสมบัติเฉพาะของมอเตอร์ ซึ่งแสดงถึงประสิทธิภาพในการเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล หากค่า Torque Constant สูงจะหมายถึงมอเตอร์สามารถสร้างแรงบิดได้มากเมื่อจ่ายกระแสไฟฟ้าในปริมาณเท่ากัน

- Back-EMF Constant เป็นค่าคงที่ที่แสดงความสัมพันธ์ระหว่างความเร็วเชิงมุมของมอเตอร์ (Angular Velocity) กับแรงดันไฟฟ้าที่เกิดขึ้นในขดลวดของมอเตอร์ (Back Electromotive Force หรือ Back-EMF) โดยมีหน่วยวัดเป็น โวลต์ต่อเรเดียนต่อวินาที ($V \cdot s/rad$) หรือ โวลต์ต่อรอบต่อนาที (V/rpm) แรงดันไฟฟ้า Back-EMF นี้เกิดจากการเหนี่ยวนำแม่เหล็กไฟฟ้า (Electromagnetic Induction) เมื่อมอเตอร์หมุน ซึ่งแรงดันไฟฟ้านี้จะมีขั้วตรงข้ามกับแรงดันไฟฟ้าที่ป้อนเข้าไป ทำหน้าที่ลดกระแสไฟฟ้าที่จ่ายเข้าสู่มอเตอร์
- PWM (Pulse Width Modulation) คือเทคนิคที่ใช้ในการปรับสัดส่วนระหว่างช่วงเวลาที่สัญญาณอยู่ในสถานะ "เปิด" (On) กับ "ปิด" (Off) ในรอบสัญญาณ (Cycle) เพื่อควบคุมปริมาณพลังงานที่ส่งผ่านไปยังโหลด เช่น มอเตอร์ไฟฟ้า
- Load Torque: แรงบิดที่ถูกกระทำกับมอเตอร์ ซึ่งมีผลต่อการทำงานและประสิทธิภาพของมอเตอร์
- Magnetic Particle Clutches: อุปกรณ์ที่ใช้ในการควบคุมแรงบิด โดยอาศัยหลักการของสนามแม่เหล็กในการเข้มต่อระหว่างเพลาขับและเพลาตาม
- H-Bridge Drive: วงจรควบคุมการหมุนของมอเตอร์ DC โดยการเปลี่ยนทิศทางกระแสไฟฟ้าผ่านขดลวดของมอเตอร์
- Hall Current Sensor: อุปกรณ์ที่ใช้วัดกระแสไฟฟ้าโดยอาศัยหลักการของ Hall Effect เพื่อให้ได้ข้อมูลที่แม่นยำ
- Signal Conditioning: กระบวนการปรับแต่งสัญญาณที่ได้รับจากเซ็นเซอร์ให้อยู่ในรูปแบบที่เหมาะสมกับการประมวลผล
- Signal Processing: กระบวนการประมวลผลสัญญาณเพื่อแยกข้อมูลที่มีความหมายจากข้อมูลดิบ
- Unwrap: กระบวนการแก้ไขค่าเฟสของสัญญาณที่ถูกคำนวณให้ต่อเนื่อง เพื่อป้องกันความผิดพลาดในการวิเคราะห์
- Angular Velocity: ความเร็วเชิงมุมของวัตถุที่กำลังหมุน โดยแสดงเป็นหน่วย radians per second (rad/s)
- %Efficiency: ค่าที่แสดงประสิทธิภาพของระบบ โดยคำนวณจากอัตราส่วนของพลังงานที่ส่งออกมีประโยชน์ต่อพลังงานที่ใส่เข้าระบบ

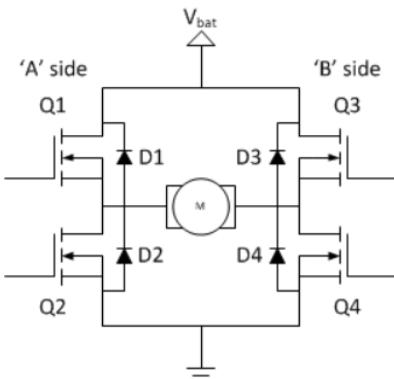
2.5. นิยามเชิงปฏิบัติการ

- ความเร็วเชิงมุม: ค่าที่วัดได้จาก Incremental Encoder และแสดงผลในหน่วย radians per second (rad/s)
- กระแสไฟฟ้า: ค่าที่วัดได้จาก Hall Current Sensor และผ่านการปรับเทียบเพื่อแสดงผลในหน่วยแอมเปอร์ (A)

- การปรับแรงดันไฟฟ้า Input: กระบวนการปรับ Duty Cycle และ Frequency ของ PWM เพื่อเปลี่ยนแปลงแรงดันไฟฟ้าที่จ่ายเข้าสู่มอเตอร์
- การปรับ Load Torque: การใช้งาน Magnetic Particle Clutches เพื่อเปลี่ยนแรงบิดที่กระทำต่อมอเตอร์

2.6. เอกสารและงานวิจัยที่เกี่ยวข้อง

2.6.1. H-Bridge Drive Mode



รูปแสดงวงจร H-Bridge Drive

https://www-modularcircuits-com.translate.goog/blog/articles/h-bridge-secrets/h-bridges-the-basics/?_x_tr_sl=en&_x_tr_tl=th&_x_tr_hl=th&_x_tr_pto=tc

H-Bridge Drive คือวงจรที่ใช้ควบคุมทิศทางและแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ DC โดยใช้สวิตซ์ 4 ตัว (เช่น MOSFET หรือ BJT) จัดเรียงในรูปแบบของตัวอักษร H ช่วยให้สามารถควบคุมการหมุนไปข้างหน้า (Forward) ถอยหลัง (Reverse) และหยุด (Stop) ได้

H-Bridge Drive มี 3 รูปแบบหลัก ได้แก่ Sign-Magnitude, Locked Anti-Phase, และ Async Sign-Magnitude แต่ละรูปแบบมีลักษณะการทำงานดังนี้:

2.6.1.1. Sign-Magnitude Mode

Sign-Magnitude เป็นโหมดที่ควบคุมมอเตอร์โดยแยกการควบคุมทิศทาง และความเร็วอย่างอิสระ

หลักการทำงาน:

- ใช้สัญญาณ PWM ควบคุมความเร็ว (Magnitude)
- ใช้ Logic Signal กำหนดทิศทาง (Sign)
- MOSFET สองตัวใน H-Bridge จะทำงานพร้อมกันเพื่อตั้งทิศทางกระแสไฟฟ้า และ PWM จะถูกนำไปใช้กับ MOSFET อีกสองตัวเพื่อควบคุมความเร็ว

การทำงานของ MOSFET:

- เมื่อ PWM เปิด-ปิด MOSFET ด้านหนึ่งในจังหวะเร็ว

- MOSFET ด้านตรงข้ามจะเปิดค้างไว้เพื่อตั้งทิศทางของกระแส

ข้อดี:

- ควบคุมง่าย
- มีความแม่นยำในการควบคุมความเร็ว

ข้อเสีย:

- พลังงานสูญเสียระหว่างช่วงหยุด (มีแรงนิ่อของมอเตอร์เมื่อ PWM อยู่ในสถานะ Off)

2.6.1.2. Locked Anti-Phase Mode

Locked Anti-Phase เป็นโหมดที่ใช้สัญญาณ PWM เพียงสัญญาณเดียวในการควบคุมทั้งทิศทางและความเร็ว

หลักการทำงาน:

- สัญญาณ PWM จะมีค่า Duty Cycle อยู่ในช่วง -100% ถึง +100%
 - ค่า Duty Cycle เป็นบวก: มอเตอร์หมุนไปในทิศทางหนึ่ง
 - ค่า Duty Cycle เป็นลบ: มอเตอร์หมุนในทิศทางตรงข้าม
- MOSFET ทั้งสองคู่ใน H-Bridge จะทำงานตรงกันข้ามกัน โดยสัญญาณ PWM จะสลับเปิด-ปิดระหว่าง MOSFET ด้านตรงข้าม

การทำงานของ MOSFET:

- MOSFET หนึ่งคู่ทำงานเพื่อตีดึงกระแสจากแหล่งจ่ายไฟ
- อีกคู่ตีดึงกระแสกลับผ่านโหลด

ข้อดี:

- ควบคุมทั้งความเร็วและทิศทางด้วยสัญญาณเดียว
- มีการตอบสนองที่รวดเร็วและมีความต่อเนื่องของแรงบิด

ข้อเสีย:

- มีการสูญเสียพลังงานมากกว่าเนื่องจากกระแสย้อนกลับ

2.6.1.3. Async Sign-Magnitude Mode

Async Sign-Magnitude เป็นโหมดที่พัฒนามาจาก Sign-Magnitude โดยเน้นประสิทธิภาพการใช้พลังงาน

หลักการทำงาน:

- คล้ายกับ Sign-Magnitude แต่เพิ่มการควบคุม Active Freewheeling

- MOSFET ที่ไม่ทำงานจะเปิดเพื่อให้กระแสไฟฟ้าผ่านอย่างอิสระ (แทนการใช้ไดโอดแบบปกติ)

การทำงานของ MOSFET:

- MOSFET จะสลับทำงานและหยุดทำงานสลับกันในลักษณะ Asynchronous เพื่อลดแรงดันไฟฟ้าต่อกัน (Voltage Drop)

ข้อดี:

- ประหยัดพลังงานมากกว่า Sign-Magnitude
- ลดการสูญเสียพลังงานในระหว่างการหยุด

ข้อเสีย:

- ซ้ำซ้อนในการออกแบบวงจร

2.7. วิธีดำเนินการทดลอง

การทดลองนี้เริ่มต้นด้วยการตั้งค่าระบบควบคุม โดยใช้บอร์ด **Nucleo STM32G474RE** ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพสูงสำหรับการควบคุมมอเตอร์และประมวลผลข้อมูล ร่วมกับโปรแกรม **MATLAB Simulink** เพื่อสร้างแบบจำลองควบคุมที่สามารถส่งสัญญาณไปยังมอเตอร์ **Nidec Components Geared DC Motor (12 V, 20 Ncm, 70 rpm)** พร้อมทั้งบันทึกข้อมูลสำคัญ เช่น ความเร็ว เสียงมุ่งและกระแสไฟฟ้า โดยใช้เซ็นเซอร์ **Incremental Encoder AMT103-V** สำหรับตรวจสอบตำแหน่งและ ความเร็ว และ **WCS1700 Hall Current Sensor** สำหรับดักกระแสไฟฟ้าที่มอเตอร์ใช้งาน

ก่อนเริ่มการทดลอง มีการปรับเทียบเซ็นเซอร์เพื่อให้ข้อมูลที่ได้มีความแม่นยำที่สุด ในส่วนของ Hall Current Sensor WCS1700 มีการทดสอบการวัดกระแสไฟฟ้าว่าตรงกับค่ากระแสที่จ่ายเข้าไปหรือไม่ หากพบว่าค่าที่ตรวจวัดไม่ตรง จะมีการปรับค่า Offset ผ่านโปรแกรม MATLAB Simulink เพื่อให้ได้ค่ากระแสที่ถูกต้องและลดความคลาดเคลื่อนของข้อมูล

จากนั้นจะมีการควบคุมแรงดันไฟฟ้า Input โดยใช้ไดรเวอร์ **Cytron MDD20A Motor Driver** ผ่านเทคนิค **PWM (Pulse Width Modulation)** ซึ่งสามารถปรับ Duty Cycle และความถี่ของ PWM เพื่อควบคุมพลังงานที่จ่ายให้มอเตอร์ นอกจากนี้ ยังใช้อุปกรณ์ **Warner Electric Magnetic Particle Clutches MPB12** เพื่อสร้างและปรับแรงบิดโหลด (Load Torque) ได้อย่างแม่นยำ

การดำเนินการทั้งหมดติดตั้งบนฐานประกอบ **MotorXplorer** ซึ่งชั้นตอนการตั้งค่าและการปรับแต่ง ทั้งหมดช่วยให้สามารถเก็บข้อมูลสำหรับวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรสำคัญ ได้แก่ ความเร็ว (Speed), แรงบิด (Torque), กระแสไฟฟ้า (Current), กำลังไฟฟ้า (Power) และประสิทธิภาพ (%Efficiency)

2.8. วัสดุอุปกรณ์

- Nidec Components Geared DC Geared Motor, 12 V dc, 20 Ncm, 70 rpm, 6mm Shaft Diameter จำนวน 1 อัน
- Incremental Encoder AMT103-V จำนวน 1 อัน
- Warner Electric Magnetic Particle Clutches MPB12 จำนวน 1 อัน
- WCS1700 Hall Current Sensor จำนวน 1 อัน
- Cytron MDD20A Motor Driver จำนวน 1 อัน
- Nucleo STM32G474RE พร้อมสายอัปโหลด จำนวน 1 ชุด
- MotorXplorer จำนวน 1 ชุด - ฐานสามารถจับต่อ Breadboard, 3D-Print ใช้สำหรับการประกอบกับ DC Motor

2.9. ขั้นตอนการดำเนินงาน

2.9.1. การตั้งค่าโปรแกรมควบคุม

1. เชื่อมต่อ Nucleo STM32G474RE กับคอมพิวเตอร์ผ่านสายอัปโหลด
2. ตั้งค่าโปรแกรม MATLAB Simulink สำหรับการควบคุมมอเตอร์และรวมข้อมูล

2.9.2. การปรับเทียบเซ็นเซอร์

1. ทดสอบความแม่นยำของ WCS1700 Hall Current Sensor โดยจ่ายกระแสไฟฟ้าทดสอบ
2. ปรับค่า Offset ในโปรแกรม MATLAB Simulink เพื่อแก้ไขค่ากระแสที่อ่านได้ให้มีความถูกต้อง
3. ตรวจสอบการอ่านค่าของ Incremental Encoder เพื่อให้มั่นใจว่าการวัดตำแหน่งและความเร็วถูกต้อง

2.9.3. การควบคุมมอเตอร์ด้วย PWM

1. เชื่อมต่อมอเตอร์เข้ากับ Cytron MDD20A Motor Driver และกำหนดค่า PWM (Pulse Width Modulation) ในโปรแกรม MATLAB Simulink
2. ปรับ Duty Cycle และความถี่ของ PWM เพื่อควบคุมแรงดันไฟฟ้าที่จ่ายให้มอเตอร์

2.9.4. การปรับแรงบิดโหลด (Load Torque)

1. ใช้อุปกรณ์ Warner Electric Magnetic Particle Clutches MPB12 เพื่อสร้างแรงบิดโหลดตามที่กำหนด

2.9.5. การเก็บข้อมูลและวิเคราะห์ผลการทดลอง

1. บันทึกค่าความเร็ว (Speed) จาก Incremental Encoder และกระแสไฟฟ้า (Current) จาก WCS1700 Hall Current Sensor

2. วิเคราะห์ข้อมูลที่ได้เพื่อหาแรงบิด (Torque), กำลังไฟฟ้า (Power) และประสิทธิภาพ (%) Efficiency)

2.10. ผลการทดลอง

ไม่สามารถดำเนินการทดลองได้เนื่องจากปัญหาการควบคุม DC Motor ซึ่งพบว่า:

- มอเตอร์ไม่ตอบสนองต่อการควบคุมด้วย MATLAB Simulink อย่างสม่ำเสมอ
- ข้อมูลที่ได้รับจาก Incremental Encoder และ Hall Current Sensor มีความผิดพลาด และไม่สามารถใช้ในการวิเคราะห์ได้
 - การปรับ Load Torque ด้วย Magnetic Particle Clutches ไม่ส่งผลต่อพฤติกรรมของมอเตอร์ตามที่คาดไว้

2.11. สรุปผล

จากการทดลองพบว่าปัญหาการควบคุม DC Motor ส่งผลให้ไม่สามารถดำเนินการทดลองได้สำเร็จ การตั้งค่าระบบควบคุมและการเก็บข้อมูลจากเซ็นเซอร์มีความผิดพลาด ซึ่งทำให้ข้อมูลที่ได้ไม่สามารถนำไปวิเคราะห์ได้อย่างถูกต้อง

2.12. อภิปรายผล

การที่ไม่สามารถดำเนินการทดลองได้สำเร็จเน้นให้เห็นถึงความสำคัญของการตรวจสอบระบบควบคุมและการปรับเทียบเซ็นเซอร์ก่อนเริ่มการทดลอง ปัญหาอาจเกิดจากความไม่สมบูรณ์ของอุปกรณ์หรือข้อผิดพลาดในการเชื่อมต่อ

2.13. ข้อเสนอแนะ

- ตรวจสอบความสมบูรณ์ของ DC Motor และการเชื่อมต่อ ก่อนการทดลอง
- ใช้การจำลองใน MATLAB Simulink เพื่อวิเคราะห์ความสมมติเบื้องต้นก่อนทดลองจริง
- เตรียมบอร์ดสำรองและอุปกรณ์ที่มีฟังก์ชันคล้ายกันเพื่อใช้งานในกรณีฉุกเฉิน
- ผู้ผลิตควรปรับปรุงคุณภาพของอุปกรณ์ เพื่อเพิ่มความเสถียรในการใช้งาน
- เพิ่มการฝึกอบรมเกี่ยวกับการใช้งานและการตรวจสอบระบบเพื่อเพิ่มความเข้าใจและลดความเสี่ยงที่เกิดขึ้นจากข้อผิดพลาดในการตั้งค่า

2.14. เอกสารอ้างอิง

Modular Circuits. H-Bridges: The Basics. สืบคืบวันที่ 29 ตุลาคม 2024. Retrieved from https://www-modularcircuits-com.translate.goog/blog/articles/h-bridge-secrets/h-bridges-the-basics/?_x_tr_sl=en&_x_tr_tl=th&_x_tr_hl=th&_x_tr_pto=tc

Motion Control Tips. Difference Between Torque Constant and Back EMF Constant in a Motor. สืบคืบวันที่ 29 ตุลาคม 2024. Retrieved from <https://www-motioncontrolltips->

com.translate.goog/faq-difference-between-torque-back-emf-motor-constant/?_x_tr_sl=en&_x_tr_tl=th&_x_tr_hl=th&_x_tr_pto=tc

iCreative Systems. To Know PWM. สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from
<https://medium.com/icreativesystems/to-know-pwm-373efa1322ce>

3. Stepper Motor

3.1. จุดประสงค์

- คณะผู้จัดทำต้องสามารถทำตามการวิเคราะห์ผลการเรียนรู้หลัก เพื่อพัฒนาความเข้าใจ และการปฏิบัติที่ถูกต้องเกี่ยวกับ Stepper Motor และการควบคุม
- คณะผู้จัดทำต้องสามารถอธิบายหลักการทำงานของ Stepper Motor ได้อย่างครบถ้วน
- คณะผู้จัดทำต้องสามารถอธิบายความสัมพันธ์ระหว่างความเร็วของ Stepper Motor กับค่าความถี่ (Frequency) ของสัญญาณที่จ่ายเข้า Stepper Motor เมื่อเปลี่ยนแปลงค่าความถี่
- คณะผู้จัดทำต้องสามารถอธิบายรูปแบบการไดร์ฟของ Stepper Motor แบบ Full-Step และ Half-Step และอธิบายผลของรูปแบบการไดร์ฟต่อการควบคุมความเร็วและตำแหน่งของมอเตอร์
- คณะผู้จัดทำต้องสามารถอธิบายกระบวนการปรับแต่งสัญญาณ (Signal Conditioning) และกระบวนการประมวลผลสัญญาณ (Signal Processing) ของข้อมูลจาก Incremental Encoder และ Hall Current Sensor ตั้งแต่การได้มาซึ่งข้อมูลจนถึงการประมวลผล พร้อมทั้งสามารถแสดงวิธีคิดและขั้นตอนที่เกี่ยวข้อง ทั้งก่อนและหลังการปรับเทียบ (Calibrate) เช่นเซอร์รวมถึงการจัดการข้อมูลและการ Unwrap ค่า
- คณะผู้จัดทำต้องสามารถหาสมการความสัมพันธ์ระหว่างกระแสไฟฟ้าที่มอเตอร์ใช้จริง กับแรงดันไฟฟ้าที่ออกมายจาก Hall Current Sensor ได้
- คณะผู้จัดทำต้องสามารถเขียนโปรแกรมโดยใช้ MATLAB และ Simulink เพื่อควบคุมความเร็วของ Stepper Motor ในรูปแบบความถี่และความเร็วเชิงมุม พร้อมกับการรับค่า Input จาก Incremental Encoder และ Hall Current Sensor
- คณะผู้จัดทำต้องสามารถแสดงผลสัญญาณ Output จากการบันทึก (Log) สัญญาณและแสดงผลเป็นกราฟผ่าน Data Inspector ใน MATLAB Simulink โดยแสดงให้เห็นว่าสัญญาณ Output มีการแปรผันตามสัญญาณ Input แบบ Real Time
- คณะผู้จัดทำต้องสามารถอธิบายและอภิรายผลการทดลองที่ได้ โดยเปรียบเทียบค่าความเร็วเชิงมุม และกระแสไฟฟ้าที่ได้ในหน่วย SI Derived พร้อมทั้งสรุปผลการทดลองด้วยเหตุผลที่ สอดคล้องกับข้อมูลที่รวบรวมได้

3.2. สมมติฐาน

- หากเพิ่มความถี่ของสัญญาณ PWM ที่ป้อนเข้าสู่ Stepper Motor ความเร็วเชิงมุมของมอเตอร์ จะเพิ่มขึ้นในอัตราส่วนที่สอดคล้องกับความถี่
- การใช้ Full-Step จะให้การหมุนที่มีกำลังแรงและความแม่นยำในตำแหน่งลดลงเมื่อเทียบกับ Half-Step ซึ่งจะเพิ่มความละเอียดและความแม่นยำในตำแหน่งที่มอเตอร์สามารถควบคุมได้ แต่จะลดแรงบิดที่ Stepper Motor สร้างขึ้น

- กระแสไฟฟ้าที่มอเตอร์ใช้จริงจะเพิ่มขึ้นเมื่อความถี่ที่ป้อนเข้าสูงขึ้น เนื่องจากต้องใช้พลังงานมากขึ้นเพื่อรักษาความเร็วในการหมุน และรูปแบบการไดร์ฟแบบ Half-Step อาจส่งผลให้การใช้พลังงานเพิ่มขึ้นเพื่อซัดขยายแรงบิดที่ลดลง
- การตั้งค่า Incremental Encoder และ Hall Current Sensor จะทำให้สามารถตรวจจับข้อมูลเกี่ยวกับตำแหน่งและกระแสไฟฟ้าได้อย่างแม่นยำ โดยแสดงผลบน MATLAB Simulink แบบเรียลไทม์ ซึ่งช่วยตรวจสอบความถูกต้องของการควบคุม Stepper Motor
- การจัดการสัญญาณ เช่น การ Calibrate Sensor หรือการ Unwrap ค่า จะช่วยปรับปรุงความแม่นยำของข้อมูลที่ได้รับจาก Incremental Encoder และ Hall Current Sensor ซึ่งส่งผลให้การวิเคราะห์ค่าความเร็วเชิงมุมและกระแสไฟฟ้ามีความแม่นยำสูง

3.3. ตัวแปร

3.3.1. ตัวแปรต้น (Independent Variables)

- ความถี่ (Frequency) ของสัญญาณที่จ่ายเข้า Stepper Motor
- รูปแบบการไดร์ฟ (Drive Mode) ได้แก่ Full-Step และ Half-Step

3.3.2. ตัวแปรตาม (Dependent Variables)

- ความเร็วเชิงมุม (Angular Speed) ของ Stepper Motor
- การเปลี่ยนแปลงตำแหน่งของ Stepper Motor
- กระแสไฟฟ้าที่รับได้จาก Hall Current Sensor

3.3.3. ตัวแปรควบคุม (Control Variables)

- แรงดันไฟฟ้าที่จ่ายเข้า Stepper Motor
- สภาพแวดล้อมในการทดลอง (เช่น อุณหภูมิ และความชื้น)
- ชุดบอร์ดและอุปกรณ์ที่ใช้ (Nucleo STM32G474RE, Incremental Encoder, Hall Current Sensor)
- ซอฟต์แวร์ที่ใช้ในการควบคุมและเก็บข้อมูล (MATLAB และ Simulink)
- การตั้งค่าการปรับเทียบ (Calibration) ของเซ็นเซอร์ที่ใช้

3.4. นิยามศัพท์เฉพาะ

- **Stepper Motor:** มอเตอร์ไฟฟ้าชนิดหนึ่งที่มีการเคลื่อนที่แบบขั้นตอน ซึ่งแต่ละขั้นตอนของการหมุนจะสอดคล้องกับสัญญาณควบคุมที่ได้รับ
- **Full-Step Drive:** รูปแบบการควบคุม Stepper Motor ที่เปิดใช้งานขาด漉ดสองชุดพร้อมกันเพื่อให้เกิดแรงบิดสูงสุดในแต่ละขั้นตอน
- **Half-Step Drive:** รูปแบบการควบคุม Stepper Motor ที่สลับการเปิดใช้งานขาด漉ดหนึ่งชุดและสองชุด ทำให้ได้ความละเอียดที่สูงขึ้นแต่แรงบิดลดลง

- **Incremental Encoder:** อุปกรณ์ที่ใช้ตรวจจับตำแหน่งและการหมุนของมอเตอร์ โดยให้สัญญาณแบบดิจิทัลเพื่อบอกจำนวนการหมุน
- **Hall Current Sensor:** อุปกรณ์ที่ใช้วัดกระแสไฟฟ้าผ่านหลักการของ Hall Effect
- **MATLAB Simulink:** ซอฟต์แวร์สำหรับการจำลองและควบคุมระบบ รวมถึงการประมวลผลสัญญาณแบบเรียลไทม์

3.5. นิยามเชิงปฏิบัติการ

- ความเร็วเชิงมุม: ค่าที่คำนวณจากข้อมูลการหมุนของ Stepper Motor ที่ตรวจจับได้จาก Incremental Encoder โดยแสดงผลในหน่วย radians per second (rad/s)
- กระแสไฟฟ้า: ค่าที่รับจาก Hall Current Sensor ซึ่งผ่านการปรับเทียบเพื่อแสดงผลในหน่วยแอมเปอร์ (A)
- การปรับเทียบเซ็นเซอร์: การตั้งค่าหรือปรับค่าที่ได้จากเซ็นเซอร์ให้ตรงกับค่ามาตรฐานเพื่อให้ข้อมูลมีความแม่นยำมากขึ้น

3.6. เอกสารและงานวิจัยที่เกี่ยวข้อง

3.6.1. รูปแบบการ Drive Stepper Motor

การ Drive Stepper Motor ในรูปแบบต่าง ๆ ส่งผลต่อความละเอียดในการควบคุมตำแหน่ง การเคลื่อนที่ ความเร็ว และแรงบิดของมอเตอร์ แต่ละรูปแบบมีลักษณะการทำงานดังนี้:

3.6.1.1. Full-Step Drive

Sign-Magnitude เป็นโหมดที่ควบคุมมอเตอร์โดยแยกการควบคุมทิศทาง และความเร็วอย่างอิสระ หลักการทำงาน:

- ขดลวดสองชุดของมอเตอร์จะถูกกระตุ้นทีละชุดหรือพร้อมกัน
- ลำดับการกระตุ้นมักเป็นแบบ สองขดลวด ที่เปิดพร้อมกัน (2 Phase On) เพื่อเพิ่มแรงบิด
- มอเตอร์หมุนในมุมที่แน่นอนต่อการกระตุ้นหนึ่งรอบ เช่น 1.8° ต่อสเต็ป (ขึ้นอยู่กับスペกมอเตอร์)

ผลกระทบต่อการควบคุม:

- ตำแหน่ง: ควบคุมตำแหน่งหยาบ เนื่องจากตำแหน่งเปลี่ยนทีละ Step Angle ที่กำหนด
- ความเร็ว: เคลื่อนที่ได้เร็วกว่า Half-Step และ Micro-Step เนื่องจากมีจุดหยุดน้อยกว่า
- แรงบิด: มีแรงบิดสูงสุด เนื่องจากขดลวดสองชุดถูกกระตุ้นพร้อมกัน

ข้อดี:

- แรงบิดสูงสุด
- ระบบง่าย ไม่ซับซ้อน
- ใช้พลังงานได้เต็มที่

ข้อเสีย:

- การเคลื่อนที่ไม่ราบรื่น (มีการสั่นสะเทือน)
- ความละเอียดต่ำ

3.6.1.2. Half-Step Drive

ในรูปแบบ Half-Step มอเตอร์เคลื่อนที่ได้ที่ลักษณะเดียวกันโดยการสลับระหว่างการกระตุนขดลวดหนึ่งชุดและสองชุด

หลักการทำงาน:

- มีการเพิ่ม "จุดหยุด" ระหว่าง Full Steps โดยการกระตุนขดลวดตั้งนี้:
 - ขดลวด A ทำงาน
 - ขดลวด A และ B ทำงานพร้อมกัน
 - ขดลวด B และ A' ทำงานพร้อมกัน
- ทำให้จำนวนสเต็ปเพิ่มเป็นสองเท่าจาก Full-Step (เช่น จาก 200 สเต็ปต่อรอบ เป็น 400 สเต็ปต่อรอบ)

ผลกระทบต่อการควบคุม:

- ตำแหน่ง: มีความละเอียดมากกว่า Full-Step (มุ่งต่อสเต็ปเล็กลง)
- ความเร็ว: ช้ากว่า Full-Step เพราะต้องเคลื่อนที่หลายจุดมากขึ้น
- แรงบิด: แรงบิดลดลงเล็กน้อยเมื่อกระตุนเพียงขดลวดเดียว

ข้อดี:

- ลดการสั่นสะเทือนของมอเตอร์
- ความละเอียดในการควบคุมตำแหน่งเพิ่มขึ้น

ข้อเสีย:

- แรงบิดลดลงในบางช่วง
- ใช้พลังงานไม่เต็มที่

3.6.1.3. Micro-Step Drive

ในรูปแบบ Micro-Step มอเตอร์เคลื่อนที่ในมุ่งเล็กกว่า Half-Step โดยการปรับความเข้มของกระแสในขดลวดทั้งสองอย่างละเอียด

หลักการทำงาน:

- ใช้การปรับความเข้มกระแส (Current) ในขดลวดสองชุดตามลำดับ ของคลีนไชน์ และโคลาเซ่น
 - ชั้น กระแสในขดลวด A เป็น $\sin(\theta)$ และในขดลวด B เป็น $\cos(\theta)$
- ทำให้มอเตอร์เคลื่อนที่ได้ทีละ "จุด" ที่เล็กกว่าหนึ่งสเต็ปของ Half-Step

ผลกระทบต่อการควบคุม:

- ตำแหน่ง: มีความละเอียดสูงสุด สามารถแบ่งย่อยตำแหน่งได้ เช่น 16, 32 หรือ 256 Micro-Steps ต่อ Full Step
- ความเร็ว: ช้ากว่า Full-Step และ Half-Step หากใช้ความละเอียดสูง
- แรงบิด: แรงบิดลดลงเมื่อเพิ่มความละเอียด เนื่องจากการกระจายแรงบิดในจุดเล็ก ๆ

ข้อดี:

- การเคลื่อนที่ราบรื่นมากที่สุด
- ลดการสั่นสะเทือนของมอเตอร์
- เหมาะสมสำหรับงานที่ต้องการความแม่นยำสูง

ข้อเสีย:

- แรงบิดต่ำที่สุดใน 3 รูปแบบ
- ซับซ้อนกว่า และต้องการไดรเวอร์คุณภาพสูง

3.7. วิธีดำเนินการทดลอง

การทดลองนี้เริ่มต้นด้วยการตั้งค่าระบบควบคุมโดยใช้บอร์ด Nucleo STM32G474RE ซึ่งเป็นไมโครคอนโทรลเลอร์ที่รองรับการส่งสัญญาณควบคุมและรับข้อมูลจากเซ็นเซอร์หลายประเภท บอร์ดนี้ถูกใช้ร่วมกับมอเตอร์ Stepper Motor RS PRO Hybrid (0.22 Nm Torque, 2.8 V, 1.8°, 42.3 x 42.3 mm Frame, 5 mm Shaft) เพื่อควบคุมการเคลื่อนที่ที่แม่นยำ โปรแกรม MATLAB Simulink ถูกใช้สำหรับออกแบบโมเดลควบคุมที่จำลองและส่งสัญญาณ PWM (Pulse Width Modulation) ไปยังมอเตอร์ ขณะเดียวกันยังบันทึกข้อมูลสำคัญ เช่น ความเร็วเชิงมุมจาก Incremental Encoder AMT103-V และกระแสไฟฟ้าที่วัดด้วย WCS1700 Hall Current Sensor เพื่อให้มั่นใจว่าข้อมูลมีความถูกต้อง ก่อนเริ่มการทดลองได้มีการปรับเทียบเซ็นเซอร์ทุกตัว โดยตรวจสอบความแม่นยำของ Incremental Encoder ในการวัดตำแหน่งและความเร็ว รวมถึงการทดสอบการวัดกระแสไฟฟ้าของ Hall Current Sensor หากพบว่ามีความคลาดเคลื่อน จะมีการปรับค่า Offset ผ่าน MATLAB Simulink เพื่อแก้ไขค่าที่วัดได้ กระบวนการทดลองเริ่มจากการส่งสัญญาณ PWM ผ่าน

Cytron MDD20A Motor Driver เพื่อควบคุมแรงดันไฟฟ้าและพลังงานที่จ่ายให้มอเตอร์ โดยสามารถปรับความถี่และ Duty Cycle เพื่อควบคุมการทำงานของมอเตอร์ได้อย่างละเอียด กระบวนการนี้ช่วยให้เคราะห์ความสัมพันธ์ระหว่างตัวแปร เช่น ความเร็วเชิงมุม กระแสไฟฟ้า และแรงบิดที่เกิดขึ้น ข้อมูลทั้งหมดถูกบันทึกไว้เพื่อนำไปวิเคราะห์ต่อ โดยอุปกรณ์ทั้งหมดถูกติดตั้งบนฐาน MotorXplorer เพื่อรองรับการประกอบมอเตอร์ และระบบควบคุม

3.8. วัสดุอุปกรณ์

- RS PRO Hybrid, Permanent Magnet Stepper Motor, 0.22Nm Torque, 2.8 V, 1.8°, 42.3 x 42.3mm Frame, 5mm
- Shaft จำนวน 1 อัน
- Incremental Encoder AMT103-V จำนวน 1 อัน
- WCS1700 Hall Current Sensor จำนวน 1 อัน
- Cytron MDD20A Motor Driver จำนวน 1 อัน
- Nucleo STM32G474RE พร้อมสายอัปโหลด จำนวน 1 ชุด
- MotorXplorer จำนวน 1 ชุด - ฐานสามารถบรรจุบอร์ดควบคุม, Breadboard, 3D-Print ใช้สำหรับการประกอบกับ StepperMotor

3.9. ขั้นตอนการดำเนินงาน

3.9.1. ตั้งค่าระบบควบคุมโดยใช้บอร์ด Nucleo STM32G474RE เพื่อส่งสัญญาณควบคุมและรับข้อมูลจากเซ็นเซอร์ต่างๆ

3.9.2. ใช้โปรแกรม MATLAB Simulink ออกแบบโมเดลควบคุมสำหรับจำลองและส่งสัญญาณ PWM (Pulse Width Modulation) ไปยัง Stepper Motor RS PRO Hybrid

3.9.3. ตรวจสอบความถูกต้องของเซ็นเซอร์ก่อนเริ่มการทดลอง:

1. ปรับเทียบ Incremental Encoder AMT103-V เพื่อให้วัดค่าความเร็วเชิงมุมและตำแหน่งได้อย่างแม่นยำ

2. ทดสอบการวัดกระแสไฟฟ้าของ WCS1700 Hall Current Sensor และปรับค่า Offset ผ่าน MATLAB Simulink หากพบความคลาดเคลื่อน

3.9.4. เริ่มทดลองโดยส่งสัญญาณ PWM ผ่าน Cytron MDD20A Motor Driver เพื่อควบคุมแรงดันไฟฟ้าและพลังงานที่จ่ายเข้าสู่มอเตอร์

3.9.5. ปรับความถี่และ Duty Cycle ของสัญญาณ PWM เพื่อควบคุมการทำงานของมอเตอร์อย่างละเอียด

3.9.6 เก็บข้อมูลตัวแปรสำคัญ ได้แก่:

- ความเร็วเชิงมุมของมอเตอร์จาก Incremental Encoder
- กระแสไฟฟ้าที่มอเตอร์ใช้จาก Hall Current Sensor
- แรงบิดที่เกิดขึ้นจากสัญญาณควบคุม

3.9.7 บันทึกผลการทดลองเพื่อนำไปวิเคราะห์ความสัมพันธ์ระหว่างตัวแปร เช่น ความเร็ว กระแสไฟฟ้า และแรงบิด เพื่อประเมินประสิทธิภาพของมอเตอร์

3.10. ผลการทดลอง

จากการทดลองพบว่าไม่สามารถดำเนินการทดลองได้สำเร็จเนื่องจากเกิดปัญหาทางเทคนิคหลายประการ กับบอร์ด Nucleo STM32G474RE และ Stepper Motor ดังนี้:

- บอร์ดไม่สามารถควบคุม Stepper Motor ได้อย่างเสถียรต้องกดปุ่ม Reset บอร์ดหรือ ปิดเปิดการจ่ายไฟใหม่เพื่อให้มันทำงาน โดย Stepper Motor มีการหมุนแบบไม่สามารถคาดเดาได้ หมุนบ้างไม่หมุนบ้าง แม้ว่าจะตั้งค่าความเร็วของสัญญาณ PWM อย่างถูกต้องแล้ว
- การควบคุม Stepper Motor ผ่าน MATLAB Simulink ไม่ทำงานตามที่คาดหวัง สัญญาณควบคุมที่ส่งออกอาจมีความผิดพลาดหรือไม่สามารถสื่อสารกับฮาร์ดแวร์ได้อย่างสมบูรณ์
- เมื่อทำการเชื่อมต่อ DC Motor ในบางครั้ง Stepper Motor กลับมีการหมุนเองโดยไม่ตั้งใจ ซึ่งบ่งชี้ถึงความผิดพลาดในการจัดการสัญญาณหรือการรบกวนจากอุปกรณ์อื่นในวงจร
- สัญญาณที่ได้จาก Incremental Encoder และ Hall Current Sensor ไม่สอดคล้องกับการทำงานของมอเตอร์ในบางจังหวะ ซึ่งทำให้ไม่สามารถวิเคราะห์ค่าความเร็วเชิงมุมหรือกระแสไฟฟ้าได้อย่างแม่นยำ

จากปัญหาดังกล่าว การทดลองในครั้งนี้ไม่สามารถดำเนินการได้จนจบกระบวนการ อย่างไรก็ตาม ข้อมูลปัญหาที่พบสามารถนำไปใช้ในการวิเคราะห์และปรับปรุงระบบในอนาคตได้

3.11. สรุปผล

จากการทดลองพบว่าปัญหาทางเทคนิคเกี่ยวกับ Stepper Motor ทำให้ไม่สามารถดำเนินการทดลองได้ตามแผนที่กำหนดไว้

3.12. อภิรายผล

การที่ไม่สามารถดำเนินการทดลองได้แสดงให้เห็นถึงความสำคัญของการตรวจสอบอุปกรณ์ก่อนการทดลอง หากมีการตรวจสอบระบบและบอร์ดควบคุมอย่างละเอียดก่อนเริ่มต้น อาจช่วยลดความเสี่ยงในการเกิดปัญหา

3.13. ข้อเสนอแนะ

- ตรวจสอบบอร์ด Nucleo STM32G474RE และการซื้อมาต่อ ก่อนเริ่มการทดลอง
- เตรียมบอร์ดสำรองหรืออุปกรณ์ที่มีฟังก์ชันใกล้เคียงเพื่อใช้งานในกรณีฉุกเฉิน
- ใช้การจำลองใน MATLAB Simulink เพื่อวิเคราะห์ความสัมพันธ์เบื้องต้นก่อนทดลองจริง
- ควรมีการประเมินคุณภาพของบอร์ดก่อนนำมาใช้งาน โดยเฉพาะในงานที่ต้องการความแม่นยำสูง เช่น การควบคุม Stepper Motor หากบอร์ด Nucleo STM32G474RE ไม่สามารถตอบสนองได้ตามที่ควรจะเป็น
 - ผู้ผลิตควรปรับปรุงกระบวนการผลิตและการตรวจสอบคุณภาพบอร์ด เพื่อให้ผู้ใช้งานได้รับอุปกรณ์ที่มีประสิทธิภาพจริงและสามารถทำการทดลองได้โดยไม่ต้องซ่อม หรือแก้ไขโดยไม่จำเป็น หลีกเลี่ยงการเสียเวลาในการแก้ปัญหาที่ไม่ควรเกิดขึ้นตั้งแต่ต้น

3.14. เอกสารอ้างอิง

Automate.org. What is the Difference Between Full Stepping, Half Stepping, and Microstepping? สืบค้นวันที่ 29 ตุลาคม 2024. Retrieved from <https://www.automate.org/motion-control/case-studies/what-is-the-difference-between-full-stepping-the-half-stepping-and-the-micro-drive>