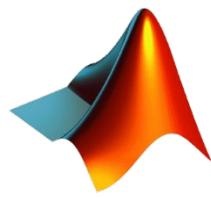


RMXplorer



Lab2 : ... Report

Name

- นายชัยกร ชื่นประเสริฐ 66340500012
- นายนาราชล นราภุลพัชร์ 66340500027
- นายพิสิฐ มงคลวิสุทธิ์ 66340500033

จุดประสงค์

- นักศึกษาต้องสามารถออกแบบการทดลองโดยใช้ทักษะกระบวนการทางวิทยาศาสตร์ ในการ สืบเสาะพูดคิดรวม ประการณ์ทดลอง บันทึกผลการทดลอง สรุปผล และอภิปรายผลการทดลอง เข้าใจหลักการทำงานของเซ็นเซอร์ และอุปกรณ์ทั้งหมดที่เกี่ยวข้องได้ ตลอดจนใช้หลักการทำงานทางวิทยาศาสตร์ คณิตศาสตร์ สถิติ และศาสตร์อื่นๆ ที่เกี่ยวข้อง ประยุกต์ใช้ร่วมกับโปรแกรม MATLAB เพื่อเก็บผลการทดลอง วิเคราะห์ผลการทดลอง วิเคราะห์ความเที่ยงตรง ความแม่นยำ ได้อย่างถูกต้อง และมีเหตุผลรองรับ ตรวจสอบความถูกต้องเทียบกับทฤษฎีที่นำเข้าถือ
- นักศึกษาต้องสามารถอธิบายความสามารถในการรับรู้ปริมาณทางฟิสิกส์ของเซ็นเซอร์ทั้งหมดตั้งแต่ต้นจนจบกระบวนการ ได้ เช่น การอธิบายการวัดความเร็วของ Brushless DC Motor จาก Hall Sensor (จากคลื่นสัญญาณไฟฟ้าแปลงเป็นความเร็วได้อย่างไร)
- นักศึกษาต้องสามารถกำหนด ตัวแปรในการทดลองได้อย่างถูกต้องและสมเหตุสมผล ไม่ว่าจะเป็น ตัวแปรต้น ตัวแปรตาม ตัวแปรควบคุม อธิบายจุดประสงค์การทดลอง อธิบายสมมติฐานให้สอดคล้องกับตัวแปรที่กำหนด นิยามเชิงปฏิบัติการ และมีทฤษฎีที่นำเข้าถือรองรับ เช่น ทฤษฎีทางฟิสิกส์ หรือข้อมูลจาก Datasheet
- นักศึกษาต้องสามารถอธิบาย ผลการทดลองที่ได้จากการทดลองที่ได้รับ ได้ด้วยตนเองอย่างถูกต้องตามหลักวิทยาศาสตร์ รวมทั้ง บันทึกผล สรุปผล อภิปรายผล ตามข้อมูลที่บันทึกได้จริง มีกระบวนการทำการทำซ้ำ อธิบายที่มาของผลการทดลองนั้นได้ โดยใช้อุปกรณ์ เครื่องมือ ไฟล์ Simulink, mlx ฯลฯ และชุดการทดลองพร้อมบอร์ด Microcontroller ที่ TA จัดเตรียมให้เบื้องต้น
- นักศึกษาต้องสามารถเขียนรายงาน สัญลักษณ์ และสมการทางวิทยาศาสตร์ ได้อย่างถูกต้อง ทั้งขนาด และรูปแบบอักษร การเว้นช่องไฟ การเว้นขอบกระดาษ การเว้นระยะพิมพ์ ให้ได้ระยะที่เหมาะสมตามหลักสากล

1. การทดลองเพื่อศึกษาหลักการทำงานของ Brushless Motor

1.1 วัตถุประสงค์

- 1.1.1 ต้องทำความรู้จักการเรียนรู้หลัก
- 1.1.2 สามารถอธิบายหลักการทำงานของ Brushless DC Motor ได้
- 1.1.3 เข้าใจวิธีการตั้งค่า Motor ในโปรแกรม Motor Workbench 6.3.2
- 1.1.4 เข้าใจวิธีการตั้งค่า PWM Generation Frequency และ Speed Sensing
- 1.1.5 เข้าใจ Electrical parameters, Mechanical parameters, และ Motor magnetic structure ของ BLDC Motor
- 1.1.6 เข้าใจ Electrical parameters, Mechanical parameters, และ Motor magnetic structure ของ BLDC Motor
- 1.1.7 เข้าใจ และ อธิบาย รูป ณ ทาง ของ BLDC Motor ทั้ง 3 Phases จากพฤติกรรมที่สังเกตว่ามาจากการควบคุมแบบไหน และสาเหตุที่กราฟมีลักษณะเช่นนี้โดยละเอียด และวิเคราะห์โดยใช้ผลการเรียนรู้อย่างที่เกี่ยวข้องมาอธิบายและอภิปรายในผลการทดลอง ครอบคลุมถึง
 1. Brushless DC (BLDC) Motor
 2. Trapezoidal Back EMF
 3. Permanent Magnet Synchronous Machine (PMSM)
 4. Trapezoidal Control
 5. Field-Oriented Control (FOC)
 6. Pulse Width Modulation (PWM)
 7. Commutation Logic
 8. Six-Step Commutation
 9. Space Vector Modulation (SVM)
 10. Sensorless Control
 11. Hall Effect Sensors

โดยสามารถนำผลการทดลองไปเปรียบเทียบกับพฤติกรรมการควบคุม PMSM แบบ FOC ด้วยการสืบค้นจากแหล่งข้อมูลภายนอก เช่น MATLAB

- 1.1.8 เข้าใจวิธีการคำนวณหาความเร็วของ BLDC Motor จาก Frequency ของสัญญาณที่อ่านได้จาก Oscilloscope
- 1.1.9 เข้าใจวิธีการใช้งาน Oscilloscope 4 Channels ในการจับสัญญาณของ BLDC Motor ทั้ง 3 Phases

1.2 สมมติฐาน

การจ่ายสัญญาณ PWM มี Duty Cycle จะส่งผลต่อความเร็วของ BLDC Motor โดยความเร็วที่เพิ่มขึ้นจะอ้างอิงจากการจ่าย % Duty Cycle ที่มีค่ามากขึ้น

การกำหนดค่าบวกลบของ RPM มีผลต่อทิศทางการหมุนของ Motor

1.3 ตัวแปร

1.3.1 ตัวแปรต้น

Duty cycle

1.3.2 ตัวแปรตาม

Back emf

ค ว า น น ร ะ ว น น อ ล ต ว ร

Phase

1.3.3 ตัวแปรควบคุม

ม อ ล ต อ ร ะ A 2 2 1 2 / 1 3 T

บ อ ร ะ D Nucleo STM32G474RE

STMICROELECTRONICS X-NUCLEO-IHM08M1

BLDCXplorer

2. นิยามศัพท์เฉพาะ

2.1. การตั้งค่า Motor Work Bench คือการตั้งค่าสำหรับใช้คุณ Motor

3. นิยามเชิงปฏิบัติการ

3.1. Brushless DC (BLDC) Motor หมายถึง มอเตอร์กระแสตรงชนิดไร้แปรงถ่าน ซึ่งใช้แม่เหล็กถาวรในโรเตอร์ และขดลวดในสเตเตอร์ที่ควบคุมการท างานผ่านวงจรยิเลกทรอนิกส์

3.2. ซอฟชิลโลสโคป คือ เครื่องที่ใช้วัดสัญญาณไฟฟ้า คลื่นไฟฟ้า วัดค่าแรงดันของไฟฟ้า การใช้วัดความถี่ วัดเฟสของสัญญาณ และใช้สำหรับการวัดคาบเวลา ซึ่งซอฟชิลโลสโคปจะแสดงผลออกมารูปกราฟ ซึ่งจะแสดงผลผ่านหลอดภาพที่ฉายด้วยฟลูออเรสเซนต์

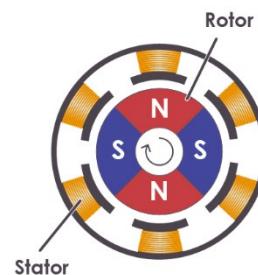
3.3. Pole Pairs หมายถึง จำนวนคู่ขั้วแม่เหล็กในโรเตอร์ ซึ่งมีผลต่อความเร็วรอบของมอเตอร์

4. เอกสารและงานวิจัยที่เกี่ยวข้อง

4.1. Motor Control Workbench เป็นซอฟต์แวร์สำหรับช่วยในการออกแบบ และตั้งค่าการควบคุมมอเตอร์ โดยช่วยลดเวลาและความยุ่งยากในการพัฒนาระบบควบคุมมอเตอร์

4.2. STM32MCMotorPilot เป็นเครื่องมือที่ออกแบบมาเพื่อแสดงภาพและควบคุมแอปพลิเคชันที่เกี่ยวข้องกับการควบคุมมอเตอร์โดยใช้ไมโครคอนโทรลเลอร์ STM32

4.3. Brushless DC Motor คือ มอเตอร์ที่ทำงานโดยใช้ตัวควบคุมอิเล็กทรอนิกส์ (Electronic Controller) ซึ่งทำหน้าที่สวิตซ์เพื่อควบคุมการจ่ายแรงดันไฟฟ้าดิจิทัลเพื่อปรับความเร็วของมอเตอร์ที่สร้างสนามแม่เหล็กต่างๆ ตามที่ต้องการ สำหรับการใช้งานในรถบินไร้คนขับ ตัวควบคุมอิเล็กทรอนิกส์จะใช้ตัวควบคุมอิเล็กทรอนิกส์แทนแบตเตอรี่ในการสร้างสนามแม่เหล็ก ซึ่งทำให้เกิดสนามแม่เหล็กและการหมุนของมอเตอร์ โดยมอเตอร์ BLDC จะใช้ตัวควบคุมอิเล็กทรอนิกส์แทนแบตเตอรี่ในการควบคุมมอเตอร์ ทิศทางการหมุน และความเร็วของมอเตอร์



Inner Rotor Motor

รูปที่ 1 แสดงถึงการทำงานของ Brushless DC Motor

Roter คือ ส่วนประกอบที่หมุนได้ ในมอเตอร์ซึ่งทำหน้าที่ในการสร้างการหมุนทางกล หรือทำปฏิกริยา กับสนามแม่เหล็กจาก Stator เพื่อสร้างพลังงานไฟฟ้า

Stator

คือส่วนประกอบที่อยู่กับที่ในมอเตอร์ซึ่งมีบทบาทสำคัญในการทำให้เกิดสนามแม่เหล็กหรือกระแสไฟฟ้าในระบบไฟฟ้า

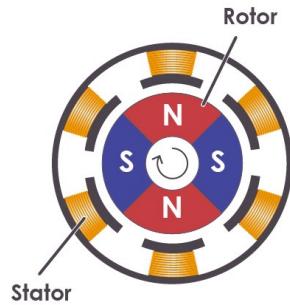
Pole pair คือ จุดที่นวนวนคู่ ซึ่งเป็นค่าที่มีผลต่อคุณสมบัติการทำงานของมอเตอร์ ที่อยู่ในสเตเตอร์หรือโรเตอร์ของมอเตอร์ซึ่งเป็นค่าที่มีผลต่อคุณสมบัติการทำงานของมอเตอร์

ประเภทของ Brushless

Inrunner และ Outrunner: ความแตกต่างหลักระหว่างทั้งสองประเภทนี้คือ ตำแหน่งของโรเตอร์:

4.3.1. Inrunner:

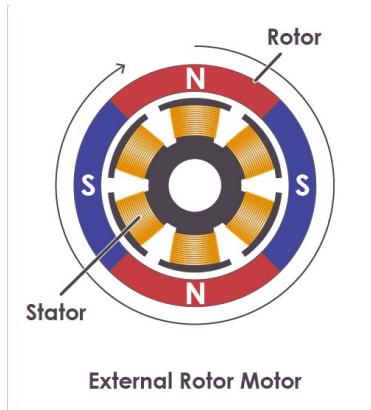
โรเตอร์อยู่ตรงกลางมอเตอร์โดยมีขดลวดแม่เหล็กไฟฟ้าอยู่ภายในตัวโครงสร้างของมอเตอร์ นิรันเนอร์ จะได้รับการปักป้องจากสภาพแวดล้อมได้ดีกว่าและระยะความร้อนได้ดีกว่า อ出租跑者



Inner Rotor Motor

รูปที่ 2 แสดงถึงการทำงานของ Brushless DC Motor ประเภท inrunner

4.3.2. Outrunner: โรเตอร์อยู่ที่ตัวเคสภายนอกโดยมีแม่เหล็กถาวรอยู่ที่ส่วนภายนอก มองเห็นได้ชัดเจน 而出runner จะมีกำลังบิด (Torque) น้อยกว่าเมื่อเทียบกับขนาดเดียวกันของ inrunner และทำงานที่ความเร็วต่ำกว่า



รูปที่ 3 แสดงถึงการทำงานของ Brushless DC Motor ประเภท outrunner

4.4. การควบคุมแบบไม่มีเซ็นเซอร์ (Sensorless control) คือการควบคุมประเภทนี้ใช้ข้อมูลแรงดันไฟฟ้าและกระแสจากมอเตอร์ในการประมาณตำแหน่งของโรเตอร์ โดยไม่ต้องใช้เซ็นเซอร์ในการระบุตำแหน่งของโรเตอร์

4.5. การควบคุมแบบใช้เซ็นเซอร์ (Sensor-based control) คือการควบคุมประเภทนี้ใช้เซ็นเซอร์ทางกายภาพ เช่น เซ็นเซอร์ Hall effect, Resolvers หรือ Encoders เพื่อรับตำแหน่งของโรเตอร์

4.6. Hall Effect Sensor มีบทบาทสำคัญในระบบ Sensor-Based Control โดยทั่วไปจะถูกนำไปใช้ในการตรวจสอบความเร็ว เช่น และแปลงเป็นสัญญาณไฟฟ้าเพื่อใช้ในวัตถุประสงค์ต่อไปนี้

4.6.1. การตรวจจับตำแหน่งของโรเตอร์ (Rotor Position Sensing)

1. Hall Effect Sensor ใช้ตรวจจับตำแหน่งของโรเตอร์เมื่อหมุนผ่านสนามแม่เหล็ก
2. ตำแหน่งของโรเตอร์จะถูกส่งไปยังไมโครคอนโทรลเลอร์ (MCU) หรือไดรเวอร์มอเตอร์เพื่อกำหนดการสลับเฟสของขดลวดสเตเตอร์ (Stator)

4.6.2. การควบคุมการสลับเฟส (Commutation Control)

1. BLDC Motor ต้องการการสลับเฟสของขดลวดอย่างถูกต้องเพื่อสร้างแรงบิดที่เหมาะสม
2. Hall Effect Sensor จะส่งสัญญาณดิจิตอล (High / Low) ตามตำแหน่งของโรเตอร์ซึ่งไมโครคอนโทรลเลอร์ใช้ในการควบคุมลำดับการจ่ายกระแส

4.6.3. การตรวจจับทิศทางการหมุน (Direction Detection)

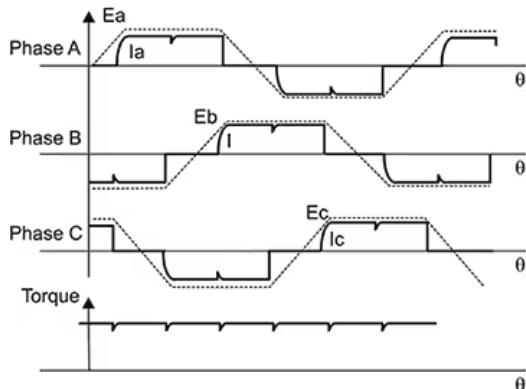
1. Hall Effect Sensor สามารถตรวจจับทิศทางการหมุนของโรเตอร์โดยการวิเคราะห์ลำดับของสัญญาณ

4.6.4. การเริ่มต้นหมุนมอเตอร์ (Startup Assistance)

1. ในช่วงเริ่มต้น (Startup), Hall Effect Sensor ช่วยระบุตำแหน่งเริ่มต้นของโรเตอร์เพื่อเริ่มสลับเฟสได้อย่างเหมาะสม
2. หากไม่มีการรับสัญญาณจากเซ็นเซอร์อาจไม่สามารถเริ่มหมุนได้หรือหมุนผิดทิศทาง

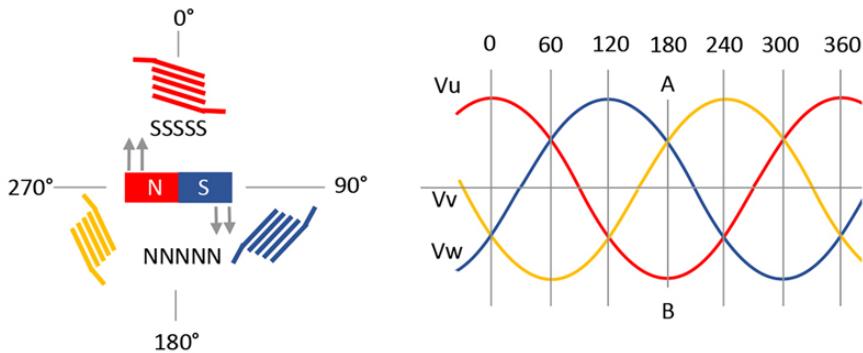
4.7. วิธีการควบคุมแบบทรายเปซอยด์ (Trapezoidal Commutation) คือ การกำหนดรูปแบบของคลื่นแรงดันไฟฟ้ากลับ (Back EMF) และคลื่นกระแสที่ใช้ในการขับเคลื่อนมอเตอร์ โดยเพียงพอให้ได้ทิศทางและแรงบิดที่ต้องการ กระบวนการนี้เรียกว่า “6-step Commutation”

เนื่องจากมีขั้นตอนการขับกระแสทั้งหมด 6 ขั้นตอนในการหมุนโรเตอร์ให้ครบหนึ่งรอบ ในวิธีนี้ จะมีการขับเคลื่อนมอเตอร์ด้วยเพียงสองเฟสในแต่ละช่วงเวลาเท่านั้น โดยวิธีการควบคุมนี้มีอัลกอริズึมที่เรียบง่าย แต่จะมีการสั่นสะเทือนของทอร์ค (Torque Ripple) เกิดขึ้นทุกครั้งที่มีการสลับเฟส (ทุก 60 องศา) ซึ่งอาจส่งผลกระทบต่อความนิ่มนวลในการทำงานของมอเตอร์



รูปที่ 4 แสดง output สัญญาณของ brushless dc motor ที่ใช้การควบคุมแบบควบคุมแบบตราเปซอยด์

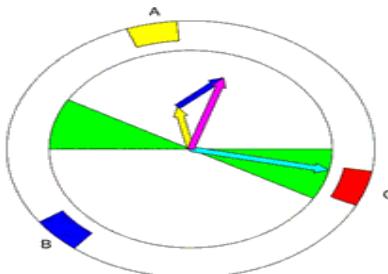
4.8. การควบคุมแบบไซนุซอยดัล (Sinusoidal Commutation) ถูกนำมาใช้ในมอเตอร์ Permanent Magnet Synchronous Motors (PMSM) เนื่องจากมอเตอร์ PMSM มีแรงดันไฟฟ้ากลับ (Back EMF) ที่มีลักษณะเป็นรูปไซนุซอยดัล แม้ว่าทฤษฎีจะระบุว่าแรงดันไฟฟ้ากลับของมอเตอร์ BLDC มีลักษณะเป็นรูปไซนุซอยดัล แต่ความเห็นยังคงมาในมอเตอร์จะทำให้รูปคลื่นของแรงดันไฟฟ้ากลับกลายเป็นรูปไซนุซอยดัล ซึ่งทำให้มอเตอร์ BLDC สามารถใช้วิธีการควบคุมได้ทั้งสองแบบ ได้แก่ การควบคุมแบบตราเปซอยด์ และ ไซนุซอยดัล ตามที่ขอของมัน รูปคลื่นของแรงดันไฟฟ้ากลับและกระแสที่ใช้ในการขับเคลื่อนมอเตอร์ในวินิจฉัยลักษณะเป็นไซนุซอยดัล การควบคุมแบบไซนุซอยดัลช่วยลดการสั่นสะเทือนของทอร์ค (Torque Ripple) และ ให้ก้าวคล่องตัว อนุทันท่วงท่า บรรลุ หลักการพื้นฐานของการควบคุมแบบไซนุซอยดัลคือการจ่ายกระแสไซนุซอยดัลที่มีการเปลี่ยนแปลงตาม ตำแหน่งของโรเตอร์ โดยกระแสจะมีการเลื่อนเฟสออกไป 120 องศา การควบคุมแบบไซนุซอยดัลจะใช้ Field Oriented Control Algorithm ซึ่งหลักการสำคัญของอัลกอริธึมนี้คือการผลิตทอร์คสูงสุดเมื่อ spanning แม่เหล็กของโรเตอร์และสเตเตอเรียมีความ ตรงรูปแบบ ขนาด กกน ดังนั้น กระแสไซนุซอยดัลจะเปลี่ยนแปลงตามตำแหน่งของโรเตอร์เพื่อให้ได้ทอร์คสูงสุดและการเคลื่อนที่ที่ราบรื่น



รูปที่ 5 แสดง output สัญญาณของ brushless dc motor ที่ใช้การควบคุมแบบควบคุมแบบไชนุชอยด์ลัล

4.9. การควบคุมแบบมุ่งเน้นสนามแม่เหล็ก (Field-Oriented Control, FOC)
หรือที่เรียกว่าการควบคุมเวกเตอร์ (Vector Control) เป็นเทคนิคที่ใช้ในการควบคุมมอเตอร์ Permanent Magnet Synchronous Motor (PMSM) และมอเตอร์เหนี่ยวนำแบบ AC (ACIM) ซึ่งช่วยให้สามารถควบคุมทอร์คและความเร็วได้ในช่วงกว้าง การนำเทคนิค FOC ไปใช้ต้องมีการแปลงกระเสสจากกรอบอ้างอิงของสเตเตอร์ที่อยู่กับที่ไปยังกรอบอ้างอิงของสนามแม่เหล็กของโรเตอร์ (กรอบอ้างอิง d-q) โดยมีการควบคุมที่ใช้บอทีสุดใน FOC คือโดยมีการควบคุมทอร์คและการควบคุมความเร็ว ในโดยมีการควบคุมทอร์คระบบควบคุมมอเตอร์จะติดตามค่าทอร์คอ้างอิง ส่วนในโดยมีการควบคุมความเร็วระบบจะติดตามค่าความเร็วอ้างอิงและสร้างทอร์คอ้างอิงเพื่อควบคุมทอร์คในขณะที่โดยมีการควบคุมที่ต้องมีข้อมูลย้อนกลับ (feedback) ของกระเสสและตำแหน่งของโรเตอร์ในแบบเรียลไทม์ ซึ่งสามารถวัดได้จากเซ็นเซอร์ หรือใช้เทคนิคแบบไรเซ็นเซอร์ที่ประมาณค่าของข้อมูลย้อนกลับแทนการใช้เซ็นเซอร์

BLDC Six-Sector Commutation
Two Pole-Pairs

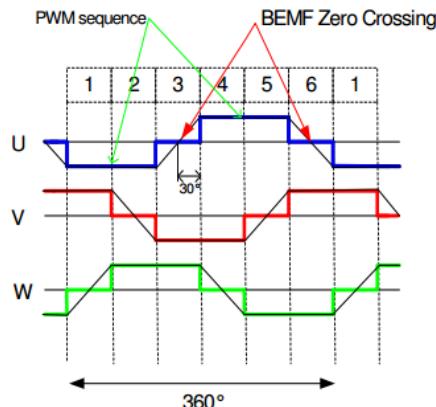


Commutes every 30 mechanical degrees

One mechanical rotation sees two electrical rotations

รูปที่ 6 แสดงการทำงานของbrushless dc motor โดยใช้การควบคุมแบบมุ่งเน้นสนามแม่เหล็ก (FOC)

4.10. Zero Crossing คือ จุดที่แรงดันไฟฟ้ากลับ (Back-EMF) ตัดกับระดับศูนย์ หรือจุดที่สัญญาณ Back-EMF เปลี่ยนทิศทางจากบวกเป็นลบ หรือจากลบเป็นบวก ซึ่งมีความสัมภัยใน การควบคุมการหมุนของมอเตอร์ เนื่องจากการตรวจจับจุดนี้ช่วยให้สามารถควบคุมการสวิตช์กระແສไฟฟ้าให้ตรงกับตำแหน่งของโรเตอร์ เพื่อให้มอเตอร์ทำงานได้อย่างมีประสิทธิภาพและไม่มีการกระตุกในระหว่างการหมุน



รูปที่ 7 แสดงจุด zero crossing บนกราฟของสัญญาณ Trapezoidal

- 4.11. Back-EMF หมายถึง แรงดันไฟฟ้ากลับ (Back Electromotive Force) ซึ่งเกิดขึ้นจากการหมุนของโรเตอร์ในมอเตอร์แบบไร้แปรงถ่าน (BLDC Motor) แรงดันนี้มีลักษณะเป็นคลื่นที่เปลี่ยนแปลงตามการหมุนของโรเตอร์ โดยมีความสัมภัยในการควบคุมการทำงานของมอเตอร์ เนื่องจากสามารถใช้แรงดันไฟฟ้านี้ในการประเมินตำแหน่งของโรเตอร์และควบคุมการจ่ายกระแสไฟฟ้าให้เหมาะสมกับการหมุนของมอเตอร์
- 4.12. Pulse with Modulation คือ เทคนิคที่ใช้สำหรับควบคุมความเร็วของมอเตอร์แบบไร้แปรงถ่าน (Brushless DC Motor: BLDC) โดยการปรับแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ เทคนิค PWM ทำงานโดยการปรับความกว้างของพัลส์แรงดันไฟฟ้าที่ส่งไปยังชุดลวดของมอเตอร์ ซึ่งการปรับความกว้างของพัลสนี้จะส่งผลต่อแรงดันไฟฟ้าเฉลี่ยที่มอเตอร์ได้รับ และทำให้สามารถควบคุมความเร็วและแรงบิดของมอเตอร์ได้อย่างมีประสิทธิภาพ การทำงานของ PWM จะควบคู่กับตัวควบคุมความเร็วอิเล็กทรอนิกส์ (Electronic Speed Controller: ESC) ซึ่งมีหน้าที่สลับแรงดันไฟฟ้าที่จ่ายให้กับชุดลวดของมอเตอร์อย่างรวดเร็ว หากความกว้างของพัลส์เพิ่มขึ้น มอเตอร์จะได้รับแรงดันไฟฟ้าสูงขึ้น ส่งผลให้ความเร็วของมอเตอร์เพิ่มขึ้น ในทางกลับกัน หากความกว้างของพัลส์ลดลง มอเตอร์จะได้รับแรงดันไฟฟ้าต่ำลง ส่งผลให้ความเร็วลดลง ซึ่งสามารถนำค่าความกว้างของ pulse มาคิดเป็น duty cycle ได้ซึ่ง duty cycle คืออัตราส่วนระหว่างเวลาที่สัญญาณอยู่ในสถานะ "เปิด" (ON) เทียบกับ ช่วงเวลาทั้งหมดของสัญญาณในหนึ่งรอบการทำงาน โดย Duty Cycle แสดงออกในรูปแบบเปอร์เซ็นต์ (%) และเป็นปัจจัยสำคัญในการควบคุมการทำงานของระบบที่ใช้ Pulse Width Modulation (PWM)

5. วิธีการดำเนินการทดลอง

- 5.1. ศึกษาการทำงานของ STMICROELECTRONICS X-NUCLEO-IHM08M1
- 5.2. ศึกษาการทำงานของ Motor Control Workbench
- 5.3. ศึกษาการทำงานของ BLDC Motor
- 5.4. เชื่อมต่อบอร์ด STM32G474RE เข้ากับ STMICROELECTRONICS X-NUCLEO-IHM08M1
- 5.5. ทำการ set up ค่าต่างๆใน Motor Control Workbench
- 5.6. ทำการทดลองโดยตั้งค่าความเร็วของ Motor ตั้งแต่ 2000 – 10000 RPM เพราะถ้าเริ่มประมาณ 1000 RPM จะแรงดันไฟฟ้าจ่ายจะจ่ายน้อยกว่าที่มอเตอร์ต้องการทำให้ Motor ไม่หมุน และปรับค่าเพิ่มขึ้นทีละ 2000 RPM เพื่อให้จ่ายต่อการเก็บค่า และเก็บค่าการทดลองทั้งค่าบวก และค่าลบ โดยจะเก็บค่าโดยการใช้ Oscilloscope เก็บค่า Duty cycle และ Back EMF เพื่อมาคำนวณ

และเปรียบเทียบความคลาดเคลื่อนของความเร็วมอเตอร์ จากที่คำนวณได้กับค่าที่แสดงในโปรแกรม Motor Pilot

6. วัสดุอุปกรณ์

- 6.1. BLDC Motor จำนวน 1 อัน Incremental Encoder AMT103-V จำนวน 1 อัน
- 6.2. STMICROELECTRONICS X-NUCLEO-IHM08M1 จำนวน 1 อัน
- 6.3. Nucleo STM32G474RE พร้อมสายอัปโหลด จำนวน 1 ชุด
- 6.4. BLDCXplorer จำนวน 1 ชุด - ฐานสามารถรับจูบอร์ดควบคุม, Breadboard, 3D-Print ใช้สำหรับการประกอบกับ BLDC Motor

7. ขั้นตอนการดำเนินงาน

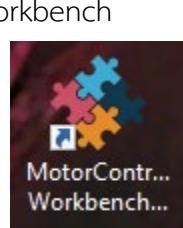
7.1. ศึกษาการทำงานของ STMICROELECTRONICS-X-NUCLEO-IHM08M1, Motor Control Workbench, BLDC Motor และ Motor Pilot

7.2. ศึกษารูปแบบการควบคุมของ Brushless DC Motor แบบต่างๆ

7.3. ทำการทดลอง

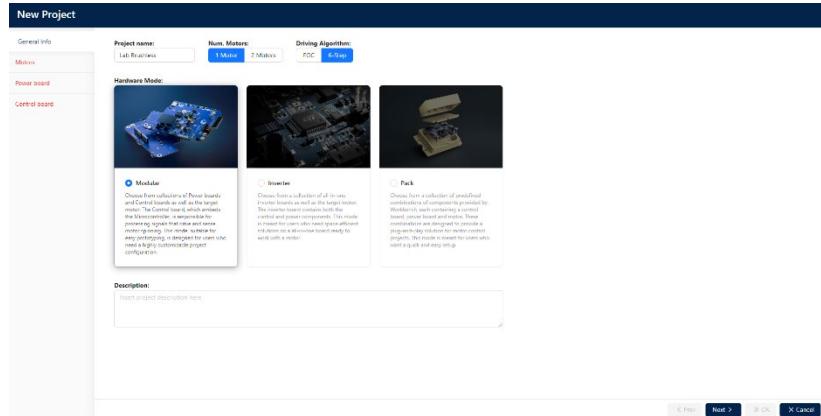
- 7.3.1. เชื่อมต่อ STM32G474RE เข้ากับ STMICROELECTRONICS X-NUCLEO-IHM08M1
- 7.3.2. เชื่อมต่อบอร์ด STM32G474Re เข้ากับคอมพิวเตอร์
- 7.3.3. ตั้งค่า Oscilloscope
 1. ตั้งค่าความละเอียดแกน X และ Y โดยให้แกน X เป็น 10V และ แกน Y เป็น 50 millisecond เพื่อใช้ในการวัดความถี่
 2. ตั้งค่าความละเอียดแกน X และ Y โดยให้แกน X เป็น 10V และ แกน Y เป็น 5 millisecond เพื่อใช้ในการวัดสัญญาณ PWM
- 7.3.4. ตั้งค่าการใช้งาน STMICROELECTRONICS-X-NUCLEO-IHM08M1 ใน โปรแกรม MotorControl Workbench

1. เปิดโปรแกรม Motor Control Workbench



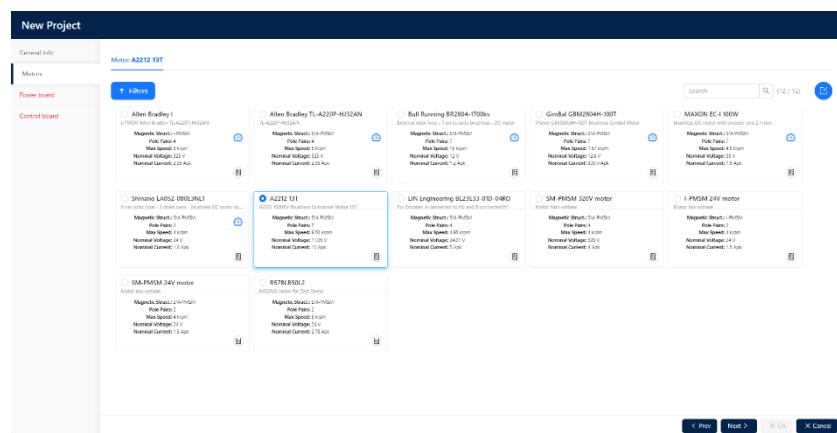
รูปที่ 8 โปรแกรม Motor WorkBench

2. ตั้งชื่อโปรเจกต์, เลือกจำนวน Motor ที่จะควบคุม และเลือกการควบคุมแบบ 6 Step



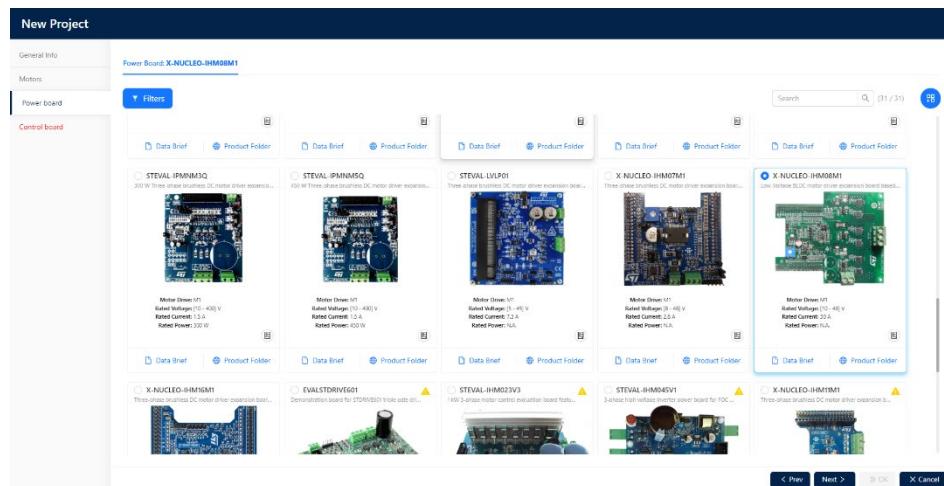
รูปที่9 แสดงหน้าต่าง New Project ที่ใช้ในการตั้งค่า

3. เลือก Motor A2212 13T ซึ่งเป็น Motor ที่จะใช้ในการทดลอง



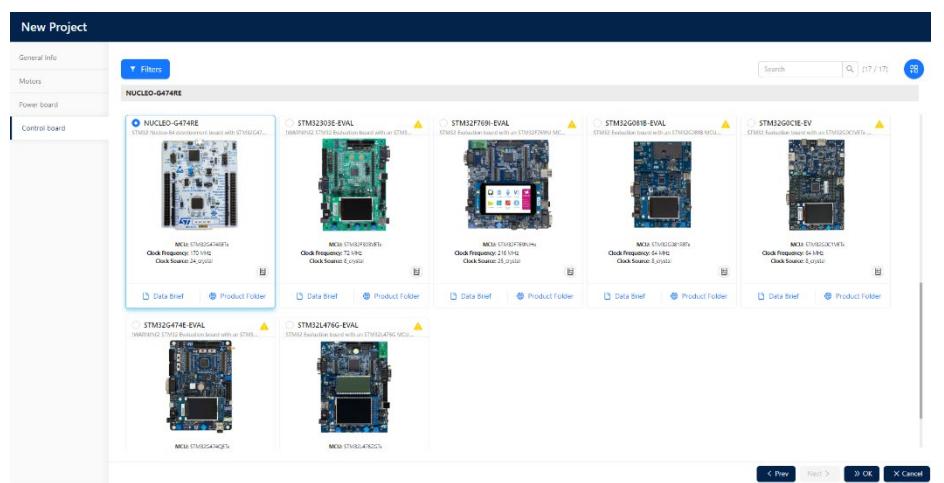
รูปที่10 แสดงหน้าต่าง New Project ที่ใช้ในการเลือก Motor

4. เลือก X-NUCLEO-IHM08M1 ซึ่งเป็น Drive ที่จะใช้ในการทดลอง



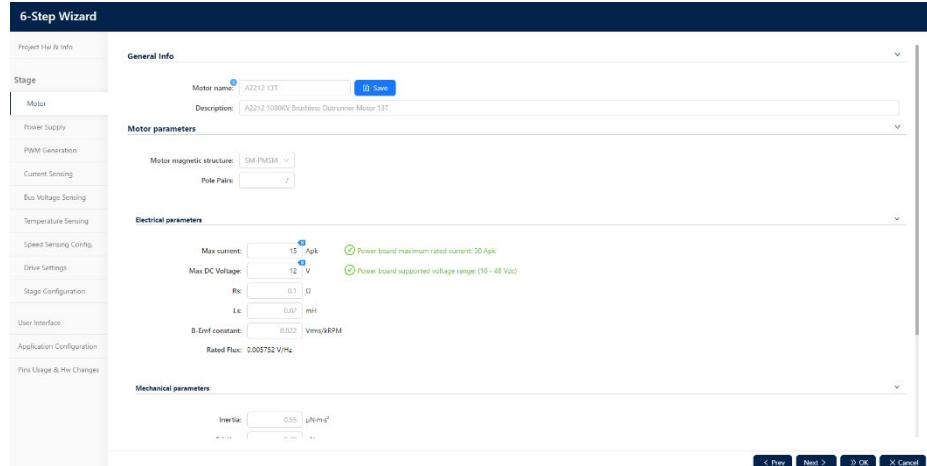
New Project ที่ใช้ในการเลือก Drive

5. เลือกบอร์ด STM32G474RE ซึ่งเป็น บอร์ดที่จะใช้ในการทดลอง



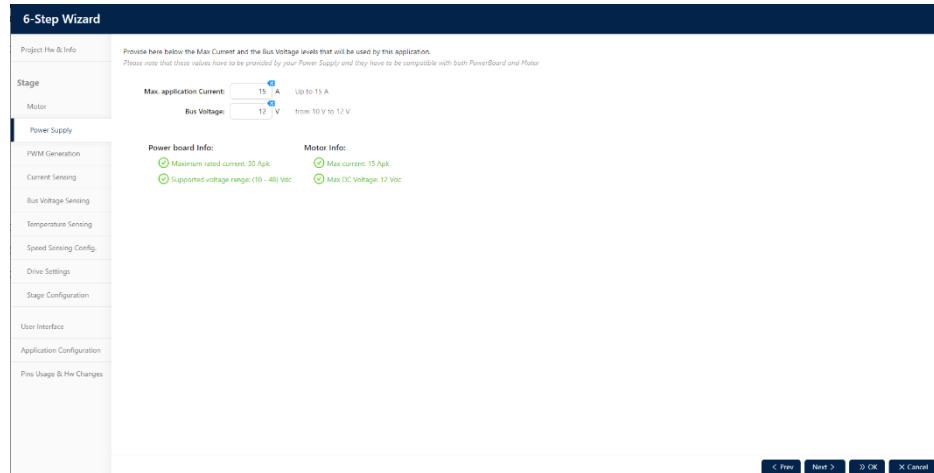
รูปที่12 แสดงหน้าต่าง New Project ที่ใช้ในการเลือก บอร์ดที่ใช้ในการควบคุม

6. การตั้งค่าการควบคุมแบบ 6 Step โดยกำหนดให้กระแสไฟที่ Motor รับได้เท่ากับ 15A และ แรงดันที่ Motor รับได้เท่ากับ 12V



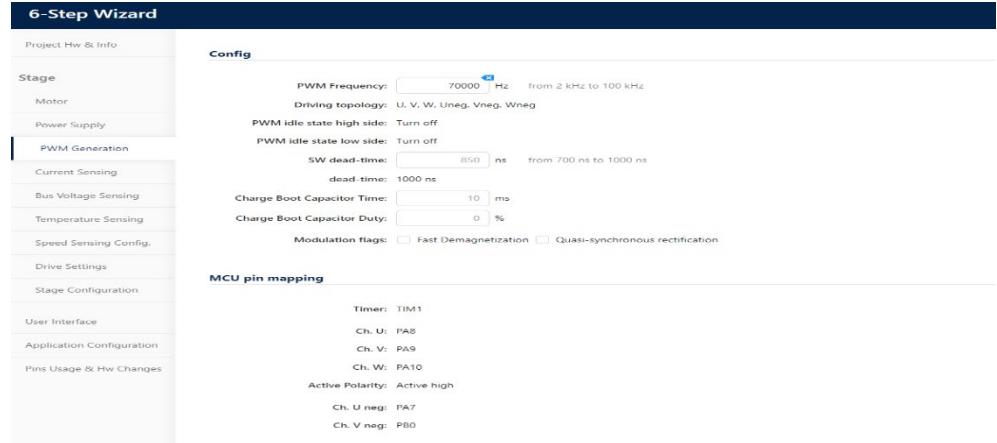
รูปที่13 แสดงหน้าต่าง 6 - Step ที่ใช้ในการตั้งค่า แรงดันและกระแสของ Motor

7. ตั้งค่าแหล่งจ่ายไฟที่จะใช้ในการควบคุม Motor กำหนดให้จ่ายกระแสไฟได้มากสุด 15 A และแรงดันไฟฟ้ามากสุด 20V



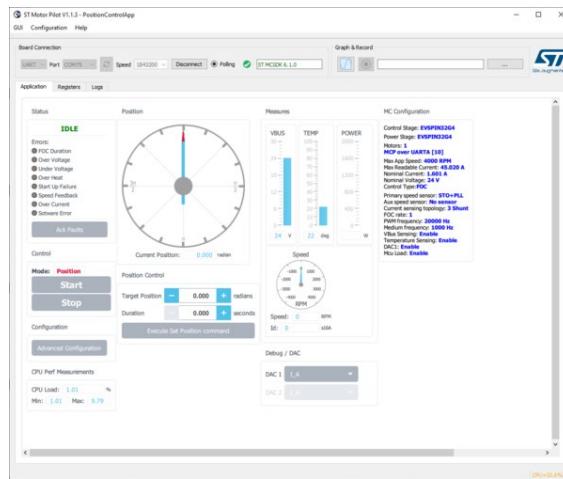
รูปที่14 แสดงหน้าต่าง 6 - Step ที่ใช้ในการตั้งค่า แรงดันและกระแสไฟในการคุ้ม Motor

8. ตั้งค่า PWM Frequency ที่ 70,000 Hz เพราะเป็นความถี่ที่ทำให้ Motor เสถียร ลดเสียง, ความร้อน และทำให้เพิ่มประสิทธิภาพของระบบ



รูปที่ 15 แสดงหน้าต่าง 6 - Step ที่ใช้ในการตั้งค่า PWM Frequency ของ Motor

9. เปิดโปรแกรม Motor Pilot เพื่อใช้ในการควบคุม RPM และ ดูกราฟ RPM



รูปที่ 16 แสดงหน้าต่างการตั้งค่า RPM ใน Motor Pilot

7.3.5. การเก็บค่า

1. วัดสัญญาณ Feedback ในช่อง U, V และ W จาก Oscilloscope

2. ทำการเก็บค่าตั้งแต่ 2,000 – 10,000 RPM เพิ่ม RPM ขึ้นทีละ 2000 RPM เก็บทั้งค่า

บวกและลบ

3. วัดความถี่ที่ได้

7.3.6. การวิเคราะห์

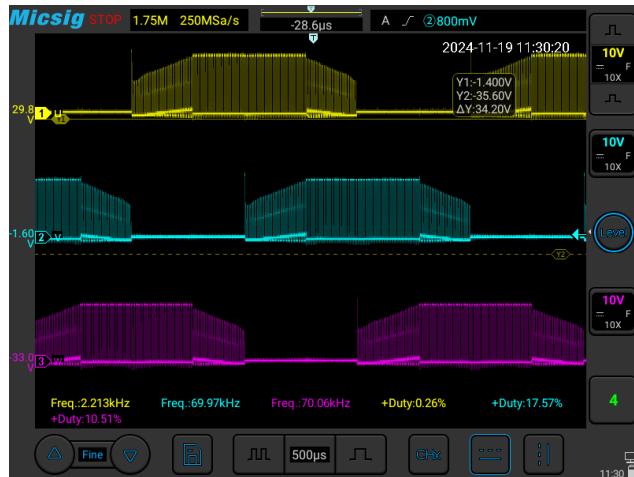
1. น.๊าค' ฯ ค ว า ม ถ.' ท.' ไ ด ' ' ป ค ' ฯ น ว ณ ห ฯ R P M
และเปรียบเทียบกับความเร็วที่กำหนดในโปรแกรม Motor Pilot

2.8.1 สรุปและอภิปรายผลการทดลอง

8. ผลการทดลอง

8.1. การแสดงผลการทดลองเก็บข้อมูลค่าความถี่ที่ได้จากการวัดค่าในหนึ่งลูกคื่นจากสัญญา Feedback ซึ่งเกิดจากการควบคุม มอเตอร์ Brushless DC Motor แบบ 6-step ทั้งค่าบวกและลบ

8.1.1. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 2000 RPM ค่าบวก



รูปที่ 17 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 2,000 RPM ค่าบวก

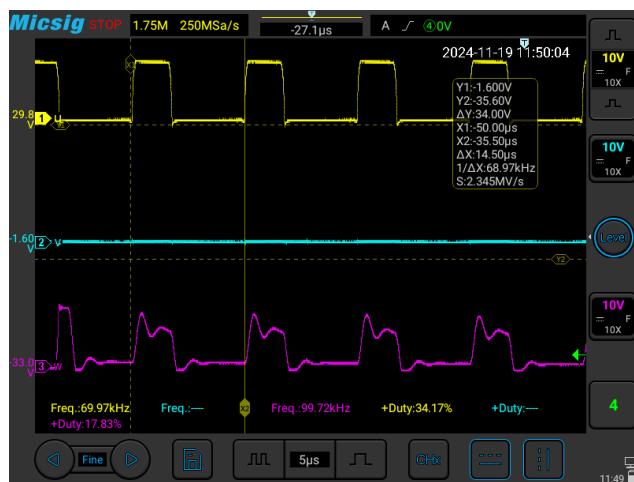


รูปที่ 18 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 2,000 RPM ค่าบวก

8.1.2. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 4000 RPM ค่าบวก



รูปที่19 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 4,000 RPM ค่าบวก

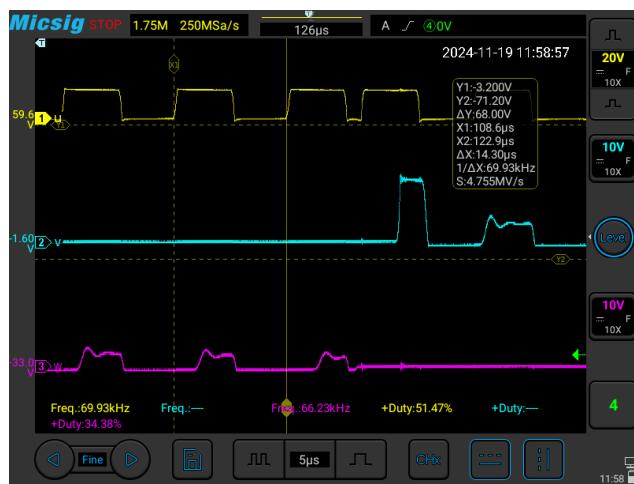


รูปที่20 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 4,000 RPM ค่าบวก

8.1.3. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 6000 RPM ค่าบวก

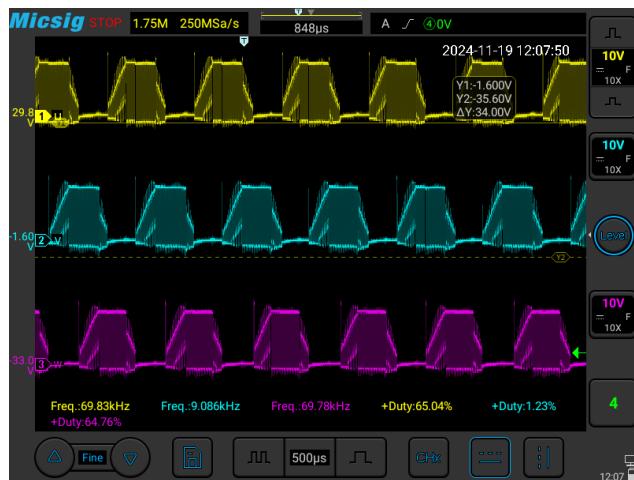


รูปที่21 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 6,000 RPM ค่าบวก



รูปที่22 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 6,000 RPM ค่าบวก

8.1.4. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 8,000 RPM ค่าบวก

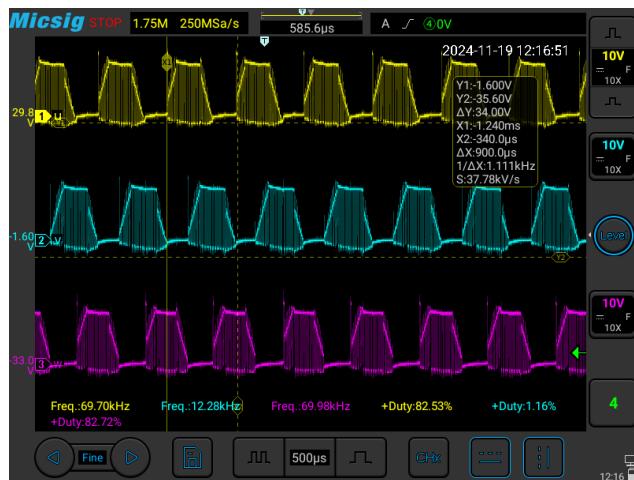


รูปที่23 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 8,000 RPM ค่าบวก

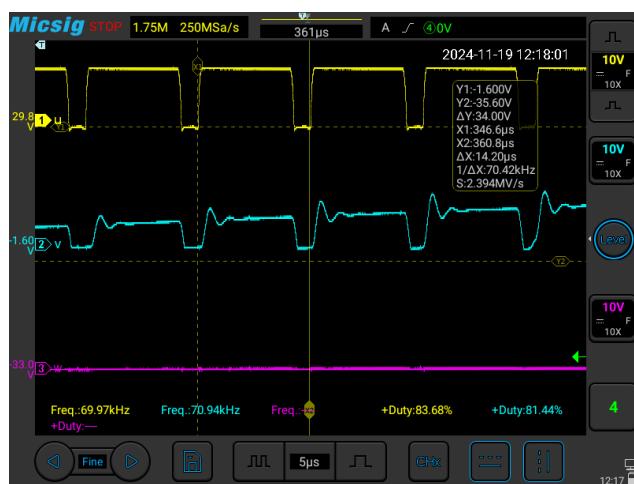


รูปที่24 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 8,000 RPM ค่าบวก

8.1.5. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 10,000 RPM ค่าบวก



รูปที่25 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 10,000 RPM ค่าบวก

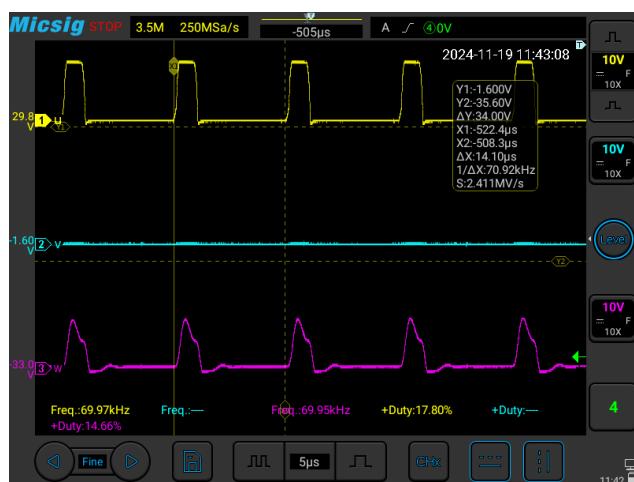


รูปที่26 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 10,000 RPM ค่าบวก

8.1.6. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 2000 RPM ค่าลบ



รูปที่27 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 2,000 RPM ค่าlab

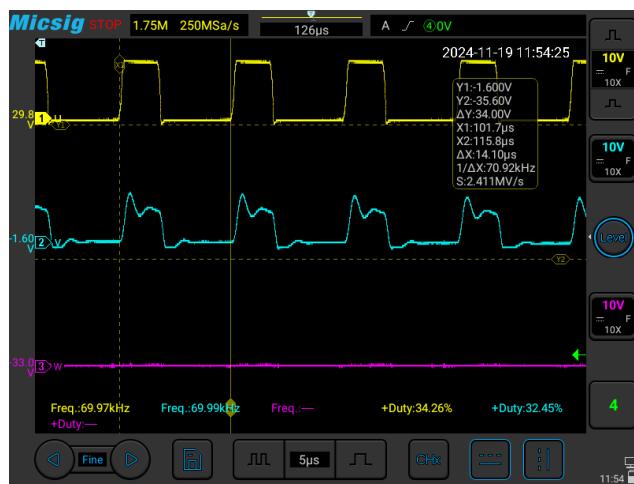


รูปที่28 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 2,000 RPM ค่าlab

8.1.7. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 4,000 RPM ค่าlab



รูปที่29 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 4,000 RPM ค่าlab



รูปที่30 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 4,000 RPM ค่าlab

8.1.8. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 6,000 RPM ค่าlab



รูปที่31 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 6,000 RPM ค่าlab

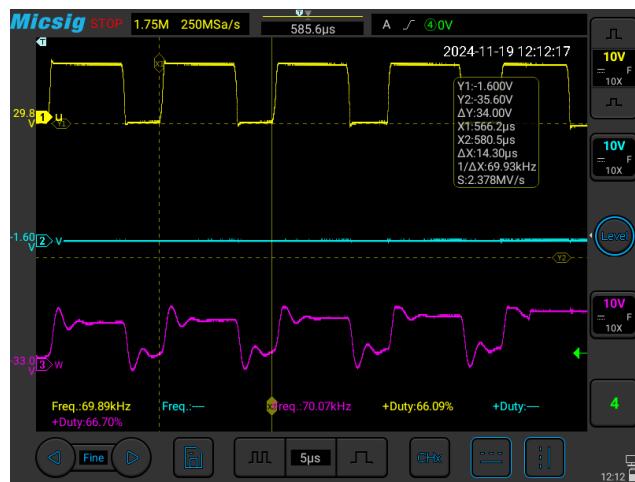


รูปที่32 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 6,000 RPM ค่าlab

8.1.9. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 8,000 RPM ค่าlab

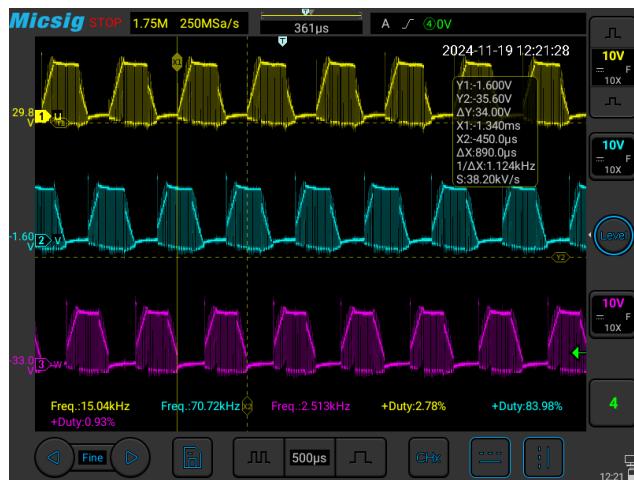


รูปที่33 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 8,000 RPM ค่าลบ

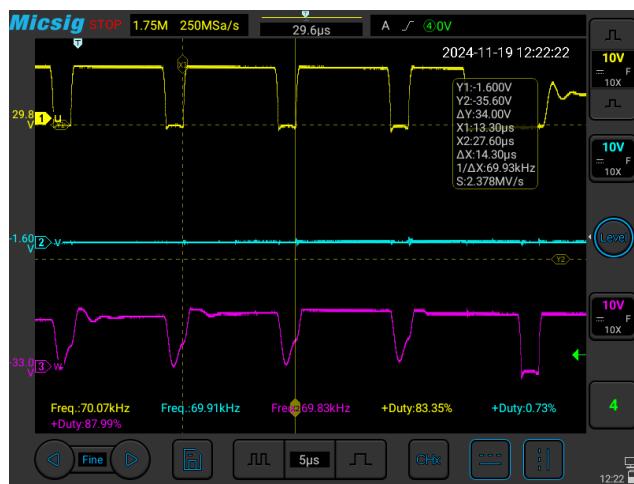


รูปที่34 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 8,000 RPM ค่าลบ

8.1.10. ค่าความถี่ที่ได้ เมื่อกำหนดความเร็วในโปรแกรมเป็น 10,000 RPM ค่าลบ



รูปที่35 ภาพ Feedback ที่เกิดขึ้นจาก Motor ที่ความเร็ว 10,000 RPM ค่าลับ



รูปที่36 ภาพ Duty Cycle ของสัญญาณ PWM ขณะความเร็ว 10,000 RPM ค่าลับ

8.2. วิเคราะห์ข้อมูล

8.2.1. วิเคราะห์ความสัมพันธ์ระหว่างความถี่ที่วัดได้และความเร็วที่กำหนดในโปรแกรม ตารางแสดงข้อมูลค่าความถี่ที่วัดได้และค่าความถี่เฉลี่ย ณ ความเร็วที่สนใจ
ตารางแสดง RPM ที่ตั้งไว้ใน Motor Work Bench กับความถี่ที่วัดได้ในค่าบวก

RPM ที่ตั้งไว้ใน Motor Work Bench	ความถี่ที่วัดได้
2,000 RPM	237 Hz
4,000 RPM	463 Hz
6,000 RPM	649 Hz
8,000 RPM	900.9 Hz
10,000 RPM	1176 Hz

ตารางแสดง RPM ที่ตั้งไว้ใน Motor Work Bench กับความถี่ที่วัดได้ในค่าลบ

RPM ที่ตั้งไว้ใน Motor Work Bench	ความถี่ที่วัดได้
2,000 RPM	234.7 Hz
4,000 RPM	467.3 Hz
6,000 RPM	704.2 Hz
8,000 RPM	909.1 Hz
10,000 RPM	1124 Hz

จากนั้นนำความถี่ที่ได้มาคำนวณหา RPM เพื่อหาความเร็วที่ Motor หมุนโดยใช้สูตร

$$RPM = \frac{f \times 60}{p}$$

โดย RPM คือ ความเร็วการหมุนของ Motor มีหน่วยเป็น Round/Min.

f คือ ความถี่ของสัญญาณที่วัดได้ มีหน่วยเป็น Hz

p คือ จำนวนขั้วแม่เหล็กใน Motor

ตาราง แสดง RPM ที่ตั้งไว้ใน Motor Work Bench และ ความเร็วที่วัดได้จากการแปลงความถี่ ที่วัดได้ค่าบวก

RPM ที่ตั้งไว้ใน Motor Work Bench	ความเร็วที่วัดได้จากการแปลงความถี่
2,000 RPM	2031.42 Hz
4,000 RPM	3968.57 Hz
6,000 RPM	5562.85 Hz
8,000 RPM	7722 Hz
10,000 RPM	10080 Hz

ตาราง แสดง RPM ที่ตั้งไว้ใน Motor Work Bench และ ความเร็วที่วัดได้จากการแปลงความถี่ ที่วัดได้ค่าlab

RPM ที่ตั้งไว้ใน Motor Work Bench	ความเร็วที่วัดได้จากการแปลงความถี่
2,000 RPM	211.71 Hz
4,000 RPM	4005.42 Hz
6,000 RPM	6036 Hz
8,000 RPM	7792.28 Hz
10,000 RPM	9634.28 Hz

8.2.2. การวิเคราะห์ความสัมพันธ์ระหว่าง RPM ที่กำหนดในโปรแกรมและ Duty Cycle ที่วัดได้

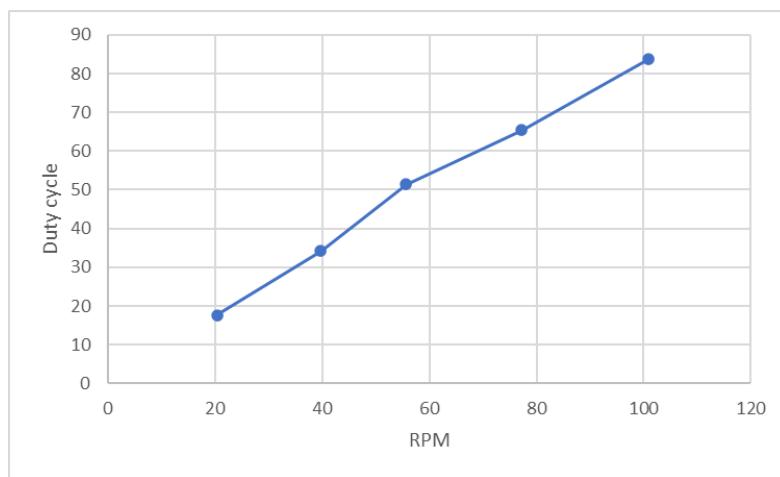
ตารางแสดง RPM ที่ตั้งไว้ใน Motor Work Bench กับ Duty Cycle ที่วัดได้ในค่าlab

RPM ที่ตั้งไว้ใน Motor Work Bench	Duty Cycle
2,000 RPM	17.64
4,000 RPM	34.17
6,000 RPM	51.47
8,000 RPM	65.36
10,000 RPM	83.68

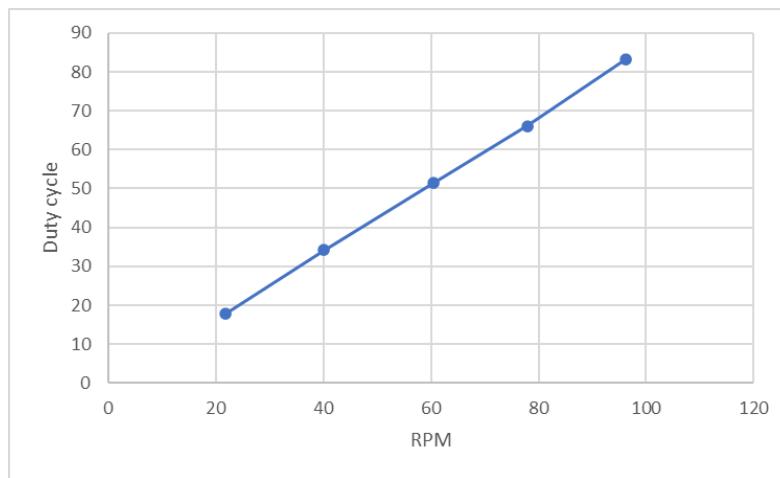
ตารางแสดง RPM ที่ตั้งไว้ใน Motor Work Bench กับ Duty Cycle ที่วัดได้ในค่าlab

RPM ที่ตั้งไว้ใน Motor Work Bench	Duty Cycle
2,000 RPM	17.80
4,000 RPM	34.26
6,000 RPM	51.39
8,000 RPM	66.09
10,000 RPM	83.35

กราฟแสดงความสัมพันธ์ระหว่าง Duty cycle กับ RPM ค่าบวก



กราฟแสดงความสัมพันธ์ระหว่าง Duty cycle กับ RPM ค่าลบ



9.สรุปผล

จากภาระที่ผลการทดลองการก้าว RPM ใน Motor Pilot เป็นบวกหรือลบคือการกำหนดทิศทางการหมุนของ Motor และค่า RPM ที่วัดได้กับค่า RPM ที่กำหนดมีความคลาดเคลื่อนกัน ผลการทดลองยังแสดงความสัมพันธ์ระหว่าง Duty cycle และ RPM มีความสัมพันธ์แบบแปรผันตรงกันถ้า Duty cycle เพิ่ม RPM ก็จะเพิ่มขึ้น

10. อภิรายผล

การควบคุมแบบ 6 – Step คือการใช้การจ่ายแรงดันไฟฟ้าเข้าไปในแต่ละเฟสเพื่อสร้างการหมุนของ Motor โดยการจ่ายแรงดันจะจ่ายบวกและลบในเฟสที่ต่างกันในเวลาเดียวกันเพื่อสร้างสนามแม่เหล็กที่ดึงหรือผลักโรเตอร์ซึ่งมีแม่เหล็กถาวร ทำให้เกิดแรงบิดและการหมุนส่วนอีกหนึ่งเฟสจะอยู่ในสถานะ Float เฟสนี้จะไม่มีแรงดันไฟฟ้า ทำให้กระแสไม่ไหล ขาดลวดของเฟสนี้แต่จะมีแรงดันที่แสดงลักษณะของ Back-EMF ที่มีลักษณะคล้ายคลื่น Trapezoidal ซึ่งเกิดจากการเคลื่อนที่ของโรเตอร์ในสนามแม่เหล็ก

11. ข้อเสนอแนะ

ลองใช้การควบคุมแบบ Foc

12. เอกสารอ้างอิง

<https://medium.com/icreativesystems/to-know-pwm-373efa1322ce>

https://www.rohde-schwarz.com/products/test-and-measurement/oscilloscopes_63663.html?mid=10913&midx=general_oscilloscope-individual_search_text-ad_r-ap_&kw=oscilloscope

<https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

2. การทดลองเพื่อศึกษาหลักการทำงานของ Stepper Motor

2.1 วัตถุประสงค์

- 1.1 การทำตามจุดประสงค์การวิเคราะห์ในผลการเรียนรู้หลัก
- 1.2 สามารถอธิบายหลักการทำงานของ Stepper Motor ได้
- 1.3 สามารถหาและอธิบายความสัมพันธ์ของความเร็ว เมื่อความถี่ของสัญญาณที่จ่ายให้กับ Stepper Motor เกิดการเปลี่ยนแปลงได้
- 1.4 สามารถอธิบายหลักการทำงานของการ Drive ตัวของ Stepper Motor ในแต่ละรูปแบบช่วงของ Full-Step, Half-Step และ Micro-Step ได้ และการทำงานของแต่ละรูปแบบสามารถควบคุมความเร็วของ Stepper Motor ได้อย่างไร
- 1.5 สามารถอธิบายกระบวนการ Signal Conditioning, Signal Processing ทั้งหมดได้ตั้งแต่ต้นจนจบกระบวนการ ว่าค่าที่อ่านได้จาก Incremental Encoder, Hall Current Sensor มีที่มาอย่างไร อธิบายให้เห็นถึงวิธีคิดและขั้นตอนทั้งหมด ทั้งก่อนและ หลัง Calibrate Sensor หรือ วิธีจัดการข้อมูลที่ได้มา จัดการอย่างไร รวมถึงความสามารถความสัมพันธ์ระหว่างกระแสไฟฟ้าที่มอเตอร์ใช้จริง กับแรงดันไฟฟ้าที่ออกมายจาก Hall Current Sensor และอธิบายกระบวนการการ Unwrap ค่า
- 1.6 สามารถเขียนโปรแกรม โดยประยุกต์ใช้ MATLAB และ Simulink ในการสั่งการควบคุมความเร็วของ Stepper Motor ในรูปแบบของความถี่ ความเร็วเชิงมุม และรับค่า ร่วมกับบอร์ด Nucleo STM32G474RE โดยใช้สัญญาณจาก Incremental Encoder, Hall Current Sensor เป็น Input และแสดงสัญญาณ Output จากการ Log สัญญาณ แสดงผลเป็นกราฟจาก Data Inspector ใน MATLAB Simulink และแสดงให้เห็นว่าสัญญาณ Output แปรผันตามสัญญาณ Input แบบ Real Time โดยมี Output เป็นความเร็วเชิงมุม และกระแสไฟฟ้า ในหน่วย SI derived

2.2 สมมติฐาน

ความถี่ของสัญญาณที่จ่ายไป ส่งผลต่อความเร็วเชิงมุมของ Stepper Motor แบบแปรผันกัน โดยความเร็วที่เพิ่มขึ้น จะอ้างอิงจากสัญญาณที่จ่ายถี่มากยิ่งขึ้น แต่ก็ทำให้แรงบิดที่ Stepper Motor ทำได้มีแรงบิดที่น้อยลง และความเร็วที่เกิดขึ้น ส่งผลต่อการเกิด Step loss ใน Stepper Motor

2.3 ตัวแปรการศึกษา

2.4.1 ตัวแปรต้น

การจ่าย Stepper frequency ของสัญญาณที่จ่ายตามความถี่ที่เพิ่มขึ้นให้ Stepper Motor ในแต่ละ Mode

2.4.2 ตัวแปรตาม

1. ความเร็วเชิงมุมของ Stepper Motor ที่เปลี่ยนแปลงไป
2. ความถี่สูงสุดที่ Stepper Motor ยังสามารถทำงานได้

2.4.3 ตัวแปรควบคุม

1. ระยะเวลาการเปลี่ยนแปลงความถี่ที่จ่ายให้ Stepper Motor เปลี่ยนแปลงไปทุก 1 วินาที
2. ข้อมูล MATLAB Simulink ทั้งค่าที่รับจาก Incremental Encoder, Hall Current Sensor และ การตั้งค่าใน MATLAB Simulink เพื่อตั้งค่า Stepper frequency ให้ค่อยๆเพิ่มความถี่ตามที่กำหนดได้อย่างถูกต้อง

2.4 นิยามศัพท์เฉพาะ

การตั้งค่า waijung ในไฟล์ motorXploer หมายถึงการ setup โปรแกรมเสริมชื่อ waijung เพื่อให้การอ่านค่าที่ได้อายก STM32 สามารถอ่านค่าโดยมี Delay ที่ลดลงอย่างมากได้

2.5 นิยามเชิงปฏิบัติการ

- 2.5.1 Stepper Motor หมายถึง มอเตอร์ไฟตรงประเภทหนึ่งที่การหมุนแบ่งออกเป็นสเต็ปหรือขั้นย่อย ๆ ที่เท่ากัน
- 2.5.2 MotorXploer หมายถึงชื่อของไฟล์ MATLAB Simulink ที่ทำการใช้งาน ซึ่งใช้ในการวัดความเร็ว เชิงมุมที่เกิดขึ้นใน Stepper Motor และใช้เพื่อสั่งความถี่ไฟฟ้าที่จ่ายไปให้ Stepper Motor เพื่อให้ Stepper Motor หมุนตามที่กำหนดไว้
- 2.5.3 Step loss หมายถึง เหตุการณ์ที่ Stepper Motor ไม่สามารถหมุนได้ตามเงื่อนไข ทำให้มอเตอร์หยุดหมุนในที่สุด
- 2.5.4

2.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.6.1 หลักการทำงานของ Stepper Motor

การทำงานของ Stepper Motor จะใช้การทำงานโดยการส่งสัญญาณไฟฟ้าไปยังขดลวดแบบเป็นช่วงๆ ซึ่งในแต่การสั่งจะทำให้การหมุนแต่ละครั้งจะหมุนเป็นขั้น (Step) ต่อการส่งสัญญาณไฟฟ้าไปให้ตัวขดลวด 1 ครั้ง โดยการหมุนจะเกิดจากการที่ไฟฟ้าที่จ่ายเป็นความถี่ในแต่ละครั้งจะจ่ายให้ตัวขดลวด (coil) เพื่อดึงหรือผลักกับตัวของสนามแม่เหล็กของโรเตอร์ (rotor) ที่อยู่ใน Stepper Motor จึงทำให้โรเตอร์เกิดการหมุนในแต่ละขั้น (step) ซึ่งในแต่ละขั้นจะหมุนไปโดยอิงอาศัยของศาสตร์การหมุนแต่ละครั้งตามจำนวนโรเตอร์ กับจำนวนของ Stator ที่มีอยู่ใน Stepper Motor นั้นๆ โดยจาก Stepper Motor ที่มีอยู่ (RS PRO Hybrid, Permanent Magnet Stepper Motor, 0.22Nm Torque, 2.8 V, 1.8°, 42.3 x 42.3mm Frame, 5mm Shaft) สามารถหมุนได้ขั้นละ 1.8°

2.6.2 ส่วนประกอบหลักของ Stepper Motor

1. Stator เป็นส่วนที่อยู่ภายนอก จะมีลักษณะเป็นชุดลวดพันรอบๆไว้ ใช้เพื่อสร้างสนามแม่เหล็กเพื่อควบคุมตัวของ Rotor
 2. Rotor คือส่วนแกนตรงกลางที่สามารถหมุนได้ จากอุปกรณ์ทดลองที่นำมาเป็นแบบ Hybrid จึงมีทั้งแม่เหล็กการและโลหะเฟอร์โรแมกเนติกอยู่ด้านใน
 3. Winding เป็นชุดลวดที่พันรอบๆ Stator เพื่อรับไฟฟ้าแล้วสร้างสนามแม่เหล็กภายใน Stepper Motor ให้ได้ทิศทางที่กำหนดไว้
 4. Shaft เป็นแท่งตรงกลาง เป็นตัวหมุนที่อยู่แกนกลางที่ออกมายกจาก Stepper Motor เพื่อนำการหมุนที่ได้ไปใช้งาน การ Loss Step ของ Stepper Motor
- การ Loss Step คือการที่ Stepper Motor ไม่สามารถหมุนไปยังตำแหน่งที่ตั้งไว้ได้อย่างถูกต้อง ทำให้เกิดการคลาดเคลื่อนตำแหน่งเกิดขึ้นซึ่งอาจเกิดสาเหตุได้ดังนี้
1. แรงบิดไม่เพียงพอ
 2. ความถี่ที่จ่ายให้ตัว Stepper Motor สูงเกินไปที่จะสั่งให้หมุนในระยะเวลาได้
 3. แรงดันไฟฟ้าที่จ่ายให้ Stepper Motor มีไม่เพียงพอไม่จะทำให้ Stepper Motor สามารถหมุนได้
 4. การเร่งจ่ายความถี่ที่จ่ายให้ Stepper Motor ที่สูงโดยทันทีโดยไม่มีการเร่งความเร็วมาก่อนหน้านั้น อาจทำให้ Stepper Motor หมุนตามความถี่ที่จ่ายไปให้ไม่ทัน ทำให้เกิดปัญหาดังข้อ 2.

2.6.3 การสั่งความเร็วด้วยความเร่งของ Stepper Motor

การสั่งความเร็วด้วยความเร่งนั้น เพื่อให้ Stepper Motor ยังสามารถมีความเร็วหมุนที่มากพอต่อการจ่ายแรงดันไฟฟ้าตามความถี่ที่เกิดขึ้น ทำให้สามารถลดปัญหา Step loss ที่เกิดจากการสั่งความเร็วเริ่มต้นที่สูงเกินไปได้

2.6.4 Datasheet ของ Stepper Motor 11HS2810



รูปที่ ... แสดงถึงตัวอย่างรูปของ Stepper Motor

1. ชนิดของ Stepper Motor : Hybrid, Permanent Magnet Stepper
2. Step Angle : 1.8°
3. Step Angle Accuracy (Full Step, No Load) : $\pm 5\%$
4. Holding Torque : 0.36 Nm
5. Detent Torque : 25 mNm
6. Number of Pole Pairs : 1
7. Number Of Leads : 4
8. Voltage Rating : 2.8 V
9. Current Per Phase : 1.68 A
10. Number Of Wires : 4Stepper Motor Specification

2.6.5 Stepper Motor Mode

1. Full mode
2. Half mode
3. 1/2 mode
4. 1/4 mode
5. 1/8 mode

6. 1/16 mode
7. 1/32 mode

2.7 วิธีดำเนินการทดลอง

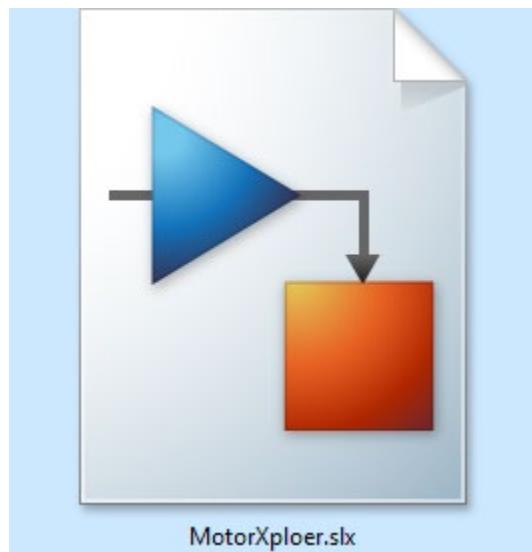
ศึกษาข้อมูลพื้นฐานเกี่ยวกับ Stepper Motor รวมไปถึงวิธีการควบคุมมอเตอร์ใน Full mode, Half mode, 1/2 mode, 1/4 mode, 1/8 mode, 1/16 mode และ 1/32 mode เพื่อดูความแตกต่างในแต่ละ mode เมื่อปรับ stepper frequency ในแต่ละแบบ เกิดความเปลี่ยนแปลงของความเร็วเป็นอย่างไร และ ความถี่สูงสุดที่จ่ายให้ Stepper Motor เมื่อความเร็วเปลี่ยนแปลงไป ความเร็วสูงสุดที่เท่าไหร่ถึงจะทำให้ Stepper Motor ไม่สามารถหมุนได้ตามปกติ พร้อมกับการทำความเข้าใจการใช้งาน Simulink ที่ได้รับกับการใช้งาน waijang โดยในการทดลองนั้นจะค่อยๆทดลองด้วยการปรับค่าที่จะเปลี่ยนความถี่ในแต่ละวินาทีและ ตรวจสอบดูความถี่สูงสุดที่สามารถวัดได้ก่อนเกิดการ Stop loss และจะค่อยๆเพิ่มความเร็วไปเรื่อยๆจนถึงจุด ที่ไม่สามารถหมุนได้มีกระทั้งความเร็วที่เปลี่ยนไปช่วงแรก ทำรูปแบบนี้ซ้ำไปเรื่อยๆกับทุกๆ Mode ที่มีอยู่

2.8 วัสดุอุปกรณ์

- 2.8.2 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
- 2.8.3 Incremental Encoder AMT103-V จำนวน 1 อัน
- 2.8.4 WCS1700 Hall Current Sensor จำนวน 1 อัน
- 2.8.5 Cytron MDD20A Motor Driver จำนวน 1 อัน
- 2.8.6 Nucleo STM32G474RE พร้อมสายอปโปโลด จำนวน 1 ชุด
- 2.8.7 MotorXplorer จำนวน 1 ชุด - ฐานสามารถบรรจุบอร์ดควบคุม, Breadboard, 3D-Print ใช้สำหรับการประกอบกับ Stepper Motor

2.9 ขั้นตอนการดำเนินงาน

1. ศึกษาข้อมูลเกี่ยวกับ Stepper Motor รุ่น 11HS2810 ในเรื่องของวิธีการใช้งาน การตั้งค่า Mode ของ Stepper Motor
2. ศึกษาวิธีการติดตั้งและการนำ Waijung มาใช้งาน
3. ศึกษาข้อมูลวิธีการใช้งาน MotorXploer ที่ใช้สั่งการปรับ Mode ความเร็ว การรับค่าต่างๆที่มีอยู่
4. ออกรูปแบบการทดลอง
5. ทำการตั้งค่าและเขียน Program ลงใน MotorXploer.slx
6. เริ่มทำการทดลอง
 - 6.1 การตั้งค่าบอร์ดที่ใช้งาน โดยทำตามวิธีการที่แนบมาในเอกสาร
 - 6.2 ทำการตั้งค่า waijung ในไฟล์ MotorXploer โดยการตั้งค่า COM ที่เสียบเชื่อมกับ NUCLEO STM32 G474 RE ให้เรียบร้อยโดยปรับให้ความถี่ของ waijung ให้เป็น 2,000,000 ในการรับ และส่งค่าต่างๆจะรับค่าโดยการใช้ MATLAB Simulink จากไฟล์ MotorXploer.slx และใช้ waijung18_11a ในการรับข้อมูลที่ได้จาก Incremental Encoder เพื่อคำนวนหาความเร็วที่เกิดขึ้น



รูปที่ ... แสดงถึงไฟล์ที่ใช้เพื่อควบคุมและรับค่าจาก Stepper Motor

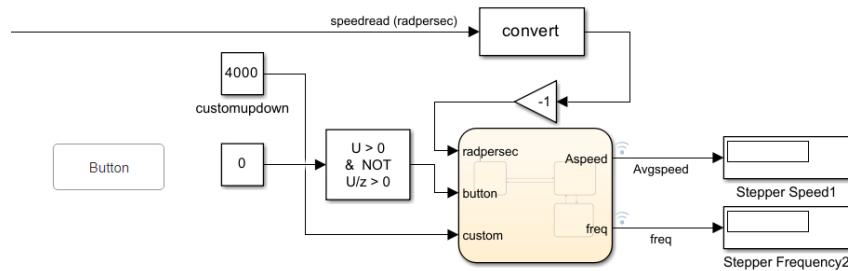
โดยสามารถนำค่าที่รับจะอ่านเป็น pulse ที่เปลี่ยนแปลงไป ซึ่งตาม Incremental Encoder ที่ได้รับนั้น 1 pulse จะหมายถึงการเปลี่ยนแปลงมุมไป 1.8 องศา ซึ่งสามารถแปลงเป็นความเร็ว ในหน่วย Radian ต่อวินาทีได้ดังสมการนี้

$$\text{ความเร็วเชิงมุมที่เกิดขึ้น} = \frac{\text{pulse} \times 2\pi}{8192}$$

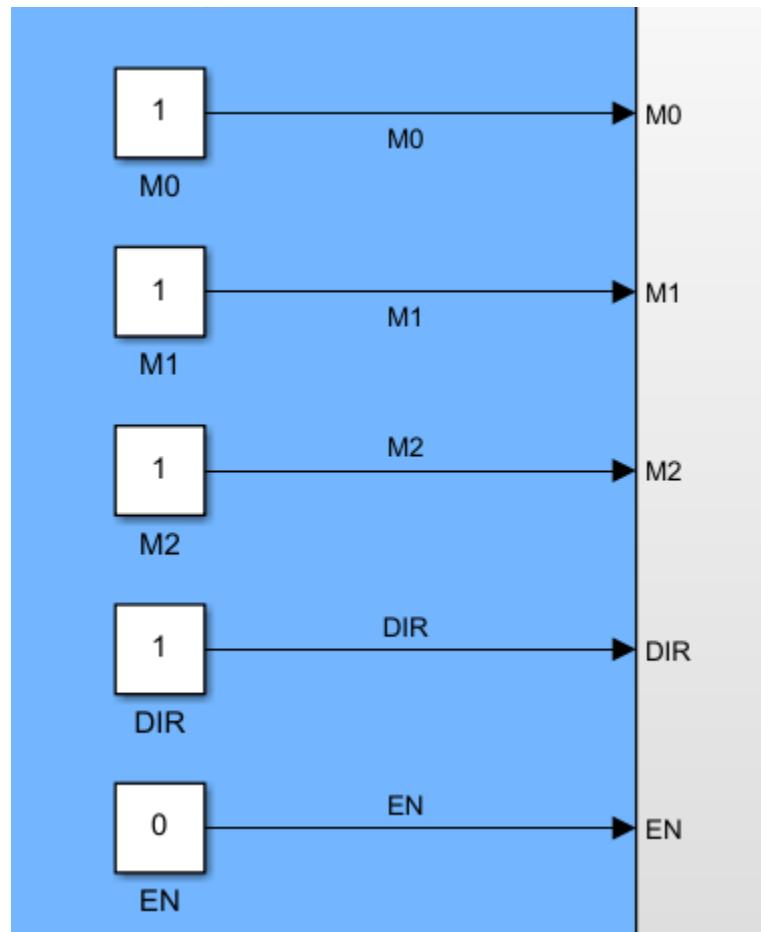
จากการทดสอบ พบร้าความเร็วที่ได้จากการรับค่ามีความไม่แน่นอนแต่ยังอยู่ในช่วงเวลาหนึ่ง จึงได้เปลี่ยนแปลงวิธีการรับค่าโดยนำค่าที่รับได้ในแต่ละครั้งมาเฉลี่ยเป็นดังสมการ

$$\text{ความเร็วเฉิงมุนที่เฉลี่ย} = \frac{\sum_{n=1}^{10000} \text{ความเร็วเฉิงมุนที่เกิดขึ้น}}{10000}$$

โดยในทุกๆ 1 วินาที จะทำการสรุปค่าเฉลี่ยความเร็วเฉิงมุนที่เกิดขึ้น ไปพร้อมกับการเพิ่มความถี่ของสัญญาณไฟฟ้าที่เพิ่มมากขึ้นตามค่าที่กำหนด



รูปที่ ... แสดงถึงการตั้งค่าเพื่อให้สามารถปรับความถี่ให้เปลี่ยนแปลงไปตามที่กำหนดได้
ในการเลือก Mode ของ Stepper Motor จะทำการปรับซองจ่ายไปทาง M0 M1 M2 ใน Matlab



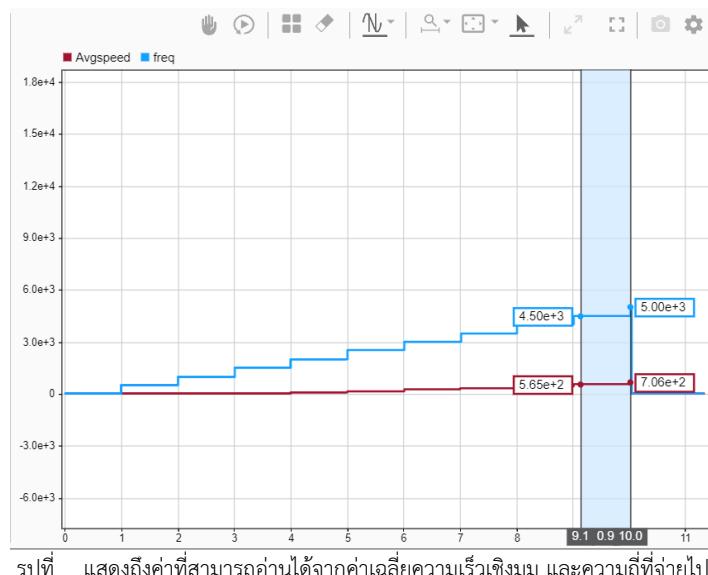
รูปที่ ... แสดงถึงค่าที่สามารถปรับได้ใน Stepper Motor

6.3 ปรับ Mode ของ Stepper Motor สามารถตั้งค่า M1 M2 M3 ตามค่าที่กำหนดได้ดังรูปภาพ

M0	M1	M2	Microstepping Mode
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Half Step
LOW	HIGH	LOW	1/4 Step
HIGH	HIGH	LOW	1/8 Step
LOW	LOW	HIGH	1/16 Step
HIGH	LOW	HIGH	1/32 Step
LOW	HIGH	HIGH	1/32 Step
HIGH	HIGH	HIGH	1/32 Step

รูปที่ ... แสดงถึงตัวอย่างการปรับค่าใน Stepper Motor

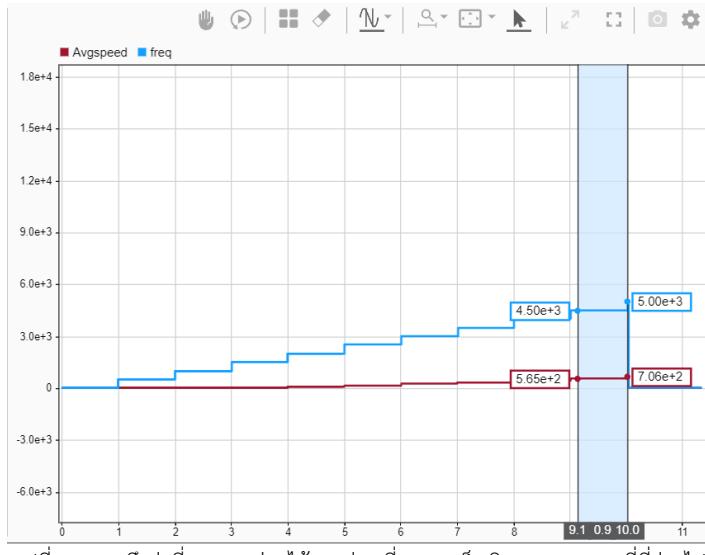
- 6.4 หลังจากการตั้งค่า Mode เรียบร้อยแล้ว ให้ทำการกดปุ่ม Button เพื่อปรับความถี่ให้เพิ่มขึ้น เรื่อยๆ โดยความถี่เริ่มต้นจะอยู่ที่ 0 Hz และจำเพิ่มขึ้นเรื่อยๆ หาก Stepper Motor ถึงจุด Step loss ค่าที่อ่านได้จาก Incremental Encoder จะอยู่ที่ 0 หรือน้อยกว่า 1 ใน Code จะทำการสั่ง ให้มอเตอร์หยุดหมุนทันทีโดยการจ่ายความถี่ที่ 0 Hz เพื่อให้ Stepper Motor หยุดหมุน หลังจากนั้นจะทำการตรวจสอบสัญญาณของความเร็วและความถี่ที่จ่ายไปในแต่ละ step ที่เกิดขึ้น ได้ เป็นดังกราฟนี้



รูปที่ ... แสดงถึงค่าที่สามารถอ่านได้จากค่าเฉลี่ยความเร็วเชิงมุม และความถี่ที่จ่ายไป

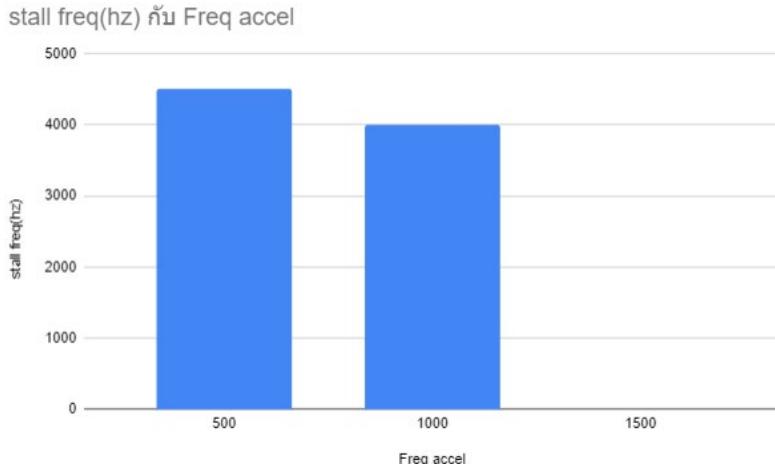
- 6.5 หลังจากการตั้งค่า Mode เรียบร้อยแล้ว ให้ทำการกดปุ่ม Button เพื่อปรับความถี่ให้เพิ่มขึ้น เรื่อยๆ โดยความถี่เริ่มต้นจะอยู่ที่ 0 Hz และจำเพิ่มขึ้นเรื่อยๆ หาก Stepper Motor ถึงจุด Step loss ค่าที่อ่านได้จาก Incremental Encoder จะอยู่ที่ 0 หรือน้อยกว่า 1 ใน Code จะทำการสั่ง

ให้มอเตอร์หยุดหมุนทันทีโดยการจ่ายความถี่ที่ 0 Hz เพื่อให้ Stepper Motor หยุดหมุนหลังจากนั้นจะทำการตรวจสอบความเร็วและความถี่ที่จ่ายไปในแต่ละ step ที่เกิดขึ้น ได้เป็นดังกราฟนี้



รูปที่ ... แสดงถึงค่าที่สามารถอ่านได้จากค่าเฉลี่ยความเร็วเชิงมุม และความถี่ที่จ่ายไป

6.6 และจากการดังกล่าว จะนำความถี่สูงสุดที่ได้ในแต่ละความเร่งที่ต่างกันเพื่อเปรียบเทียบความถี่สูงสุดที่ทำได้ต่อความเร่งที่เปลี่ยนแปลงไป โดยนำมาทำเป็นตารางและ Plot ค่าอุอกมาดังนี้



รูปที่ ... แสดงถึงการแสดงความถี่สูงสุดในแต่ละความเร่งที่ปรับ

6.7 ทำการทำข้อตามข้อที่ 6.3-6.6 จนกว่าจะครบทุก Mode ที่สามารถตั้งค่าได้

6.8 ทำการวิเคราะห์ผลการทดลอง

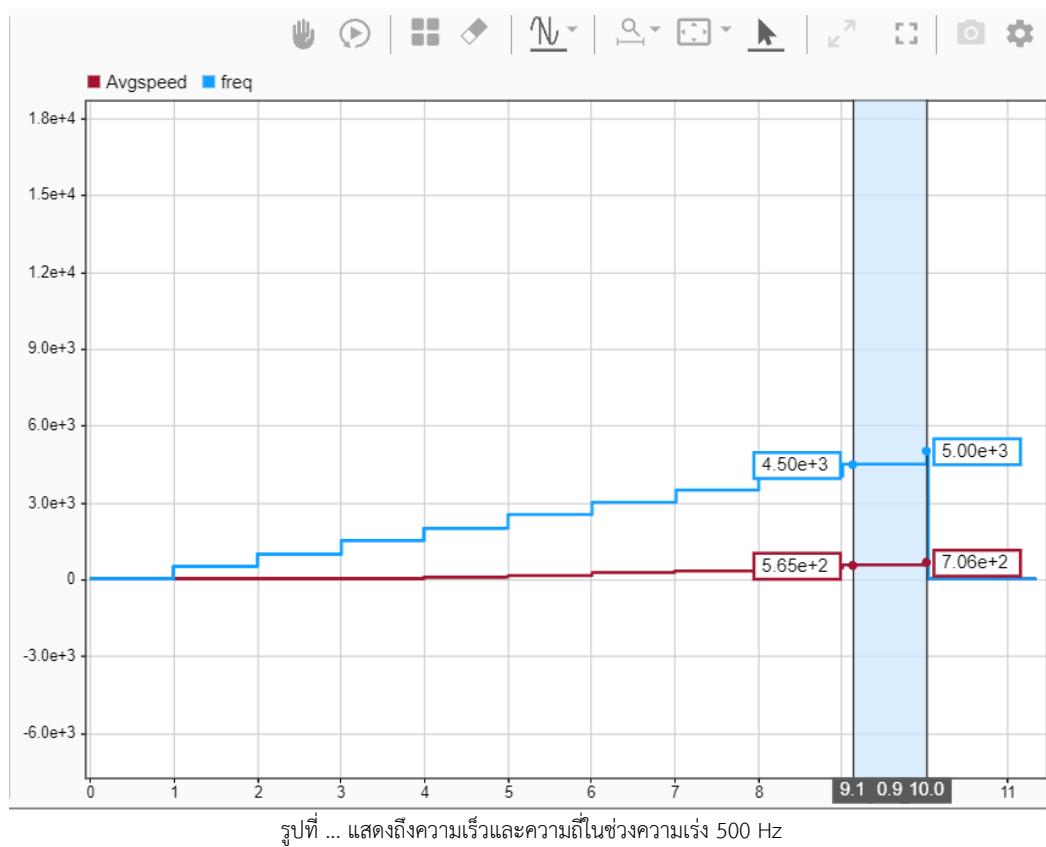
6.9 สรุปและอภิปรายผลที่ได้จากการทดลอง รวมถึงข้อเสนอแนะสำหรับการทดลอง

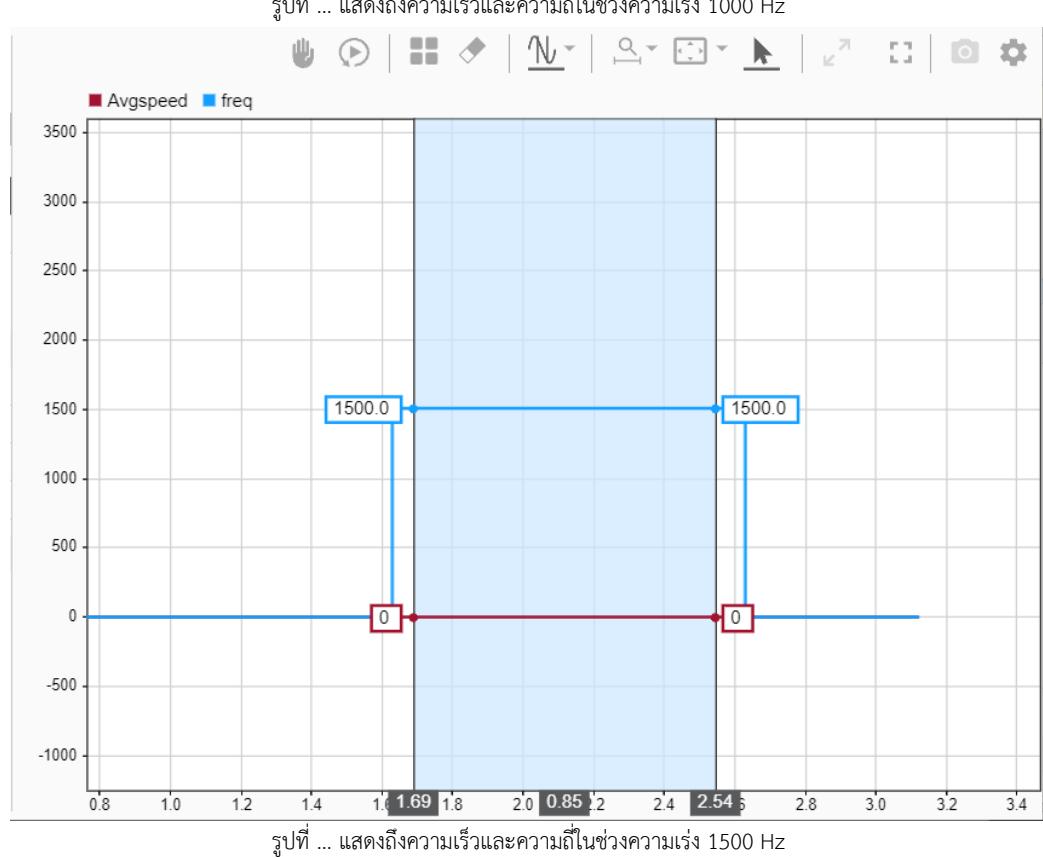
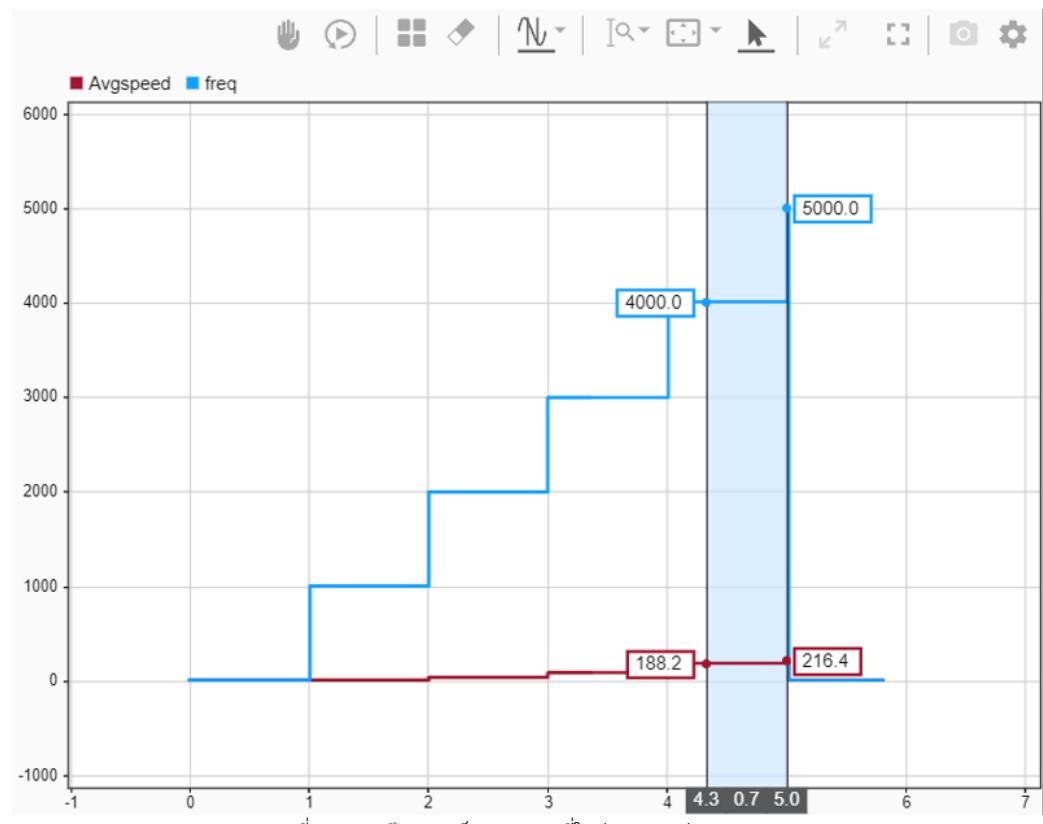
2.10 ผลการทดลอง

การแสดงผลการทดลอง

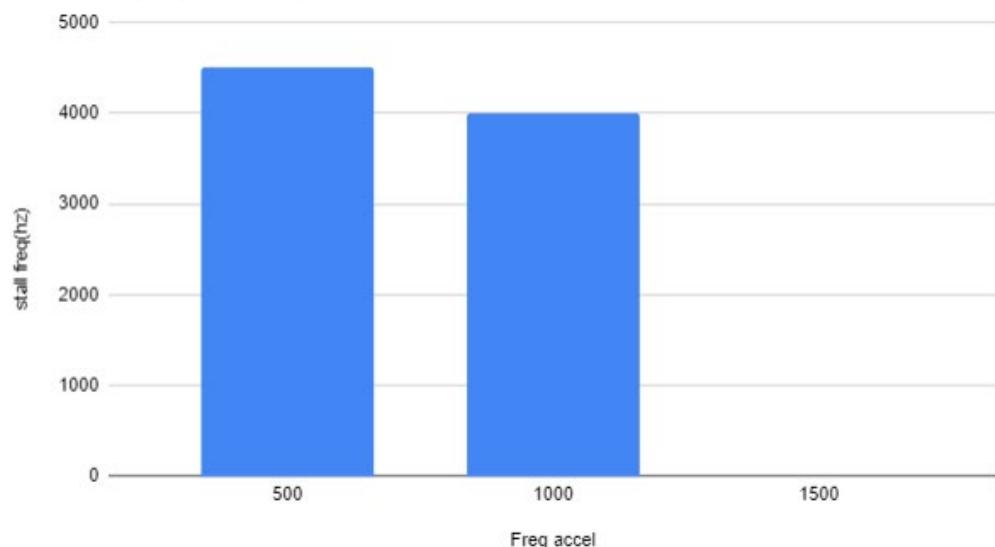
เก็บข้อมูลความเร็วและความถี่ในช่วงทุกๆ 1 วินาที และเทียบค่าความถี่สูงสุดที่ได้ก่อนการ Step loss

1. การเก็บข้อมูลค่าความถี่ที่ได้ต่อความเร็วที่ทำได้เมื่อให้ความเร่งความถี่ในรูปแบบต่างๆ เมื่อยุ่นในการทำงานแบบ Full Step mode



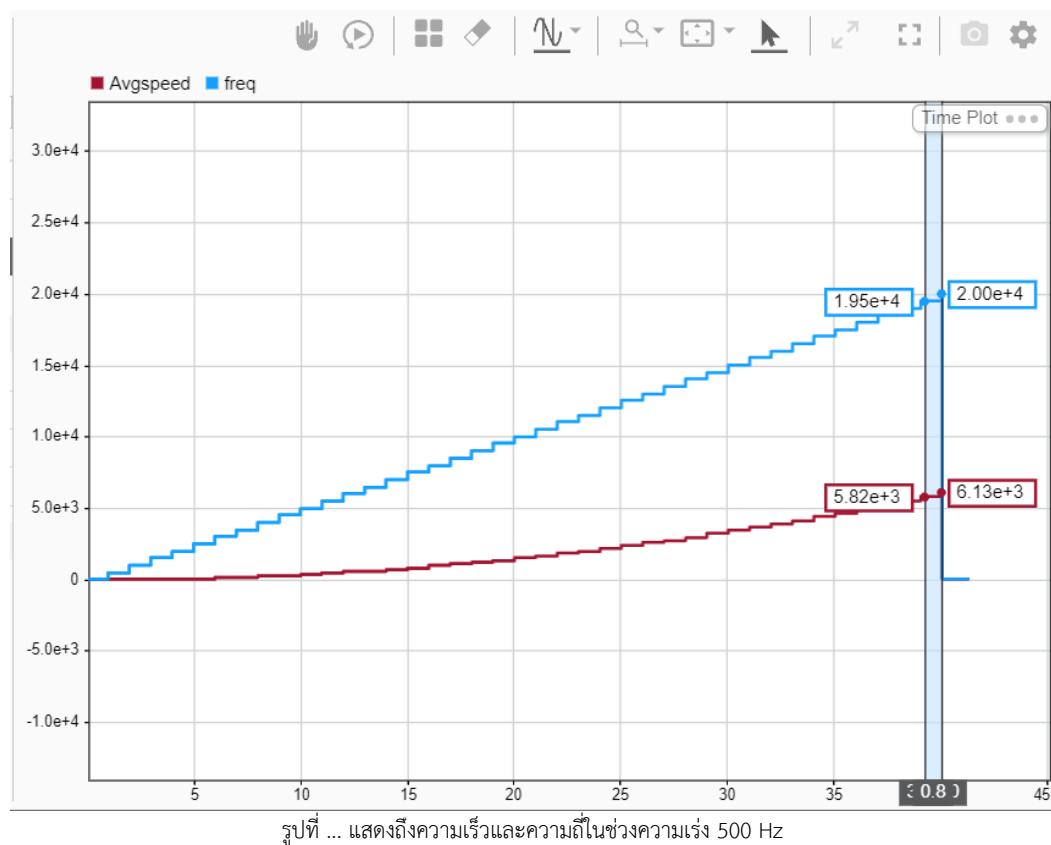


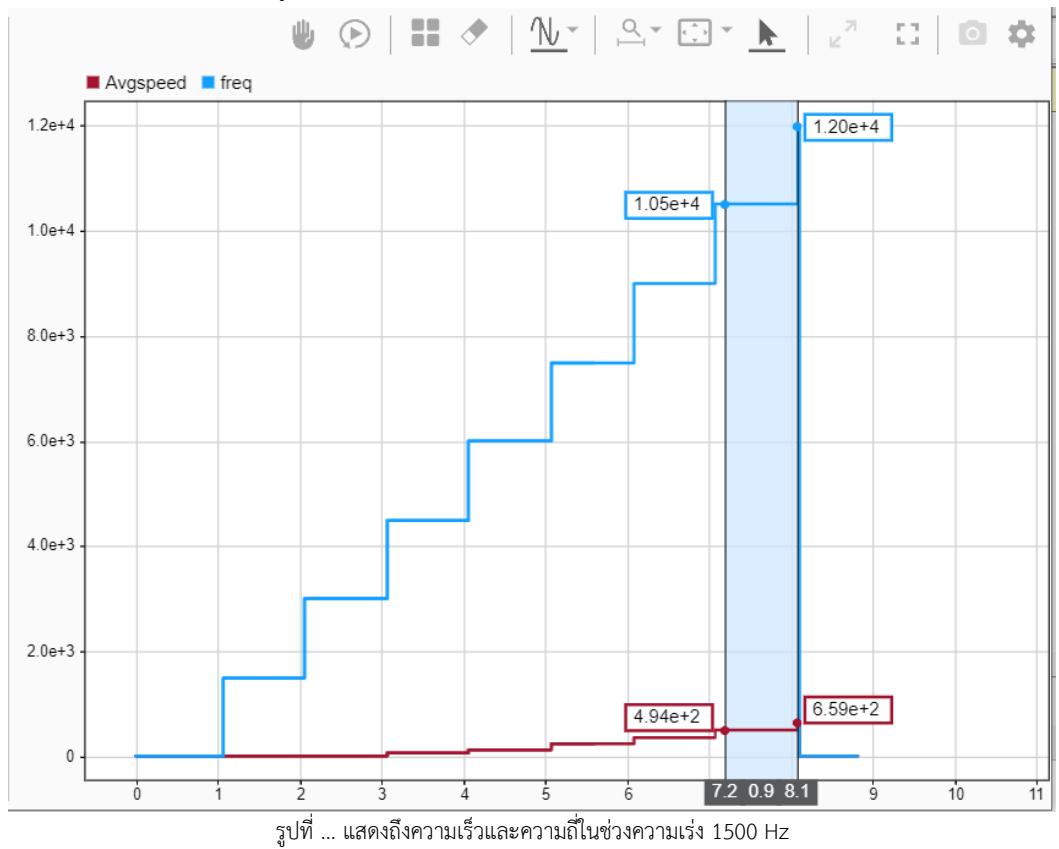
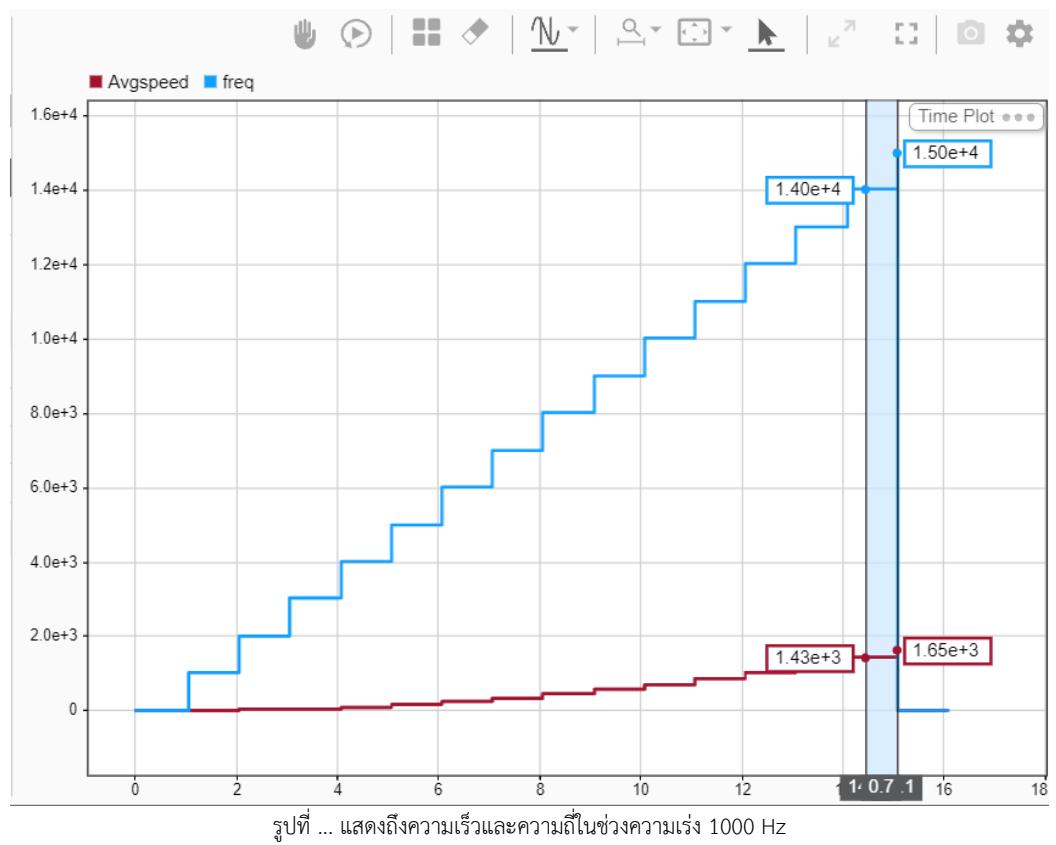
stall freq(hz) กับ Freq accel

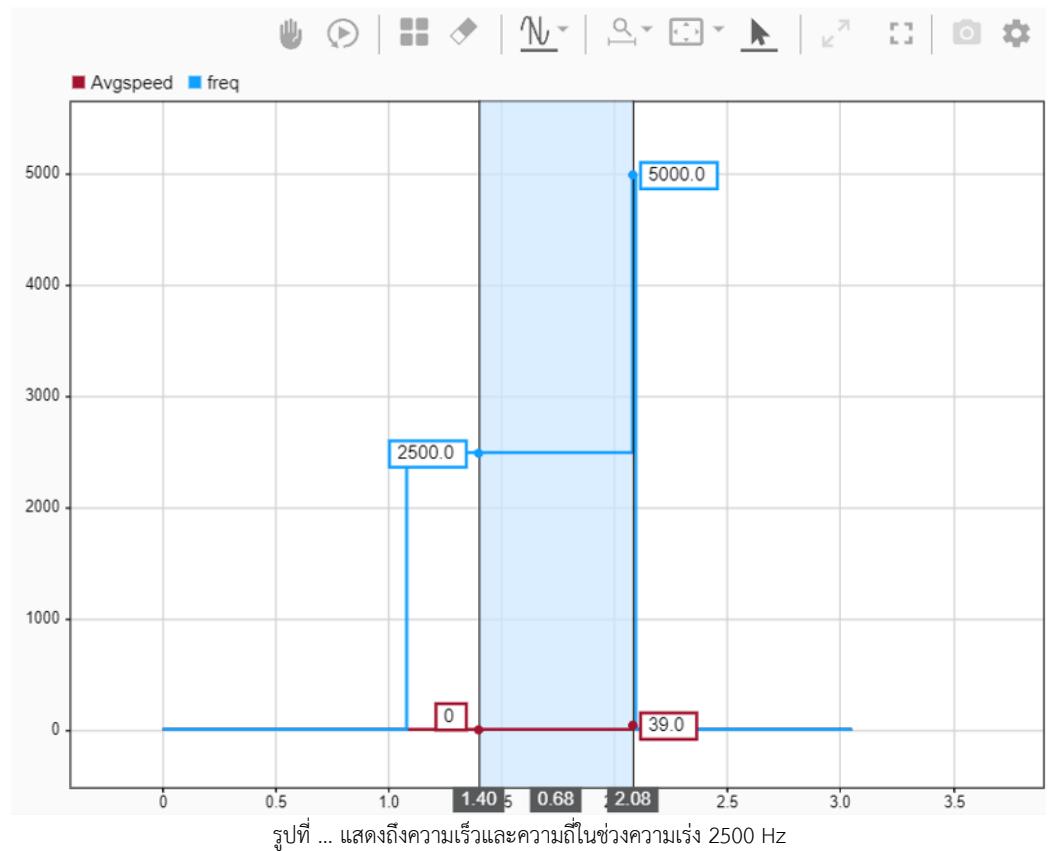
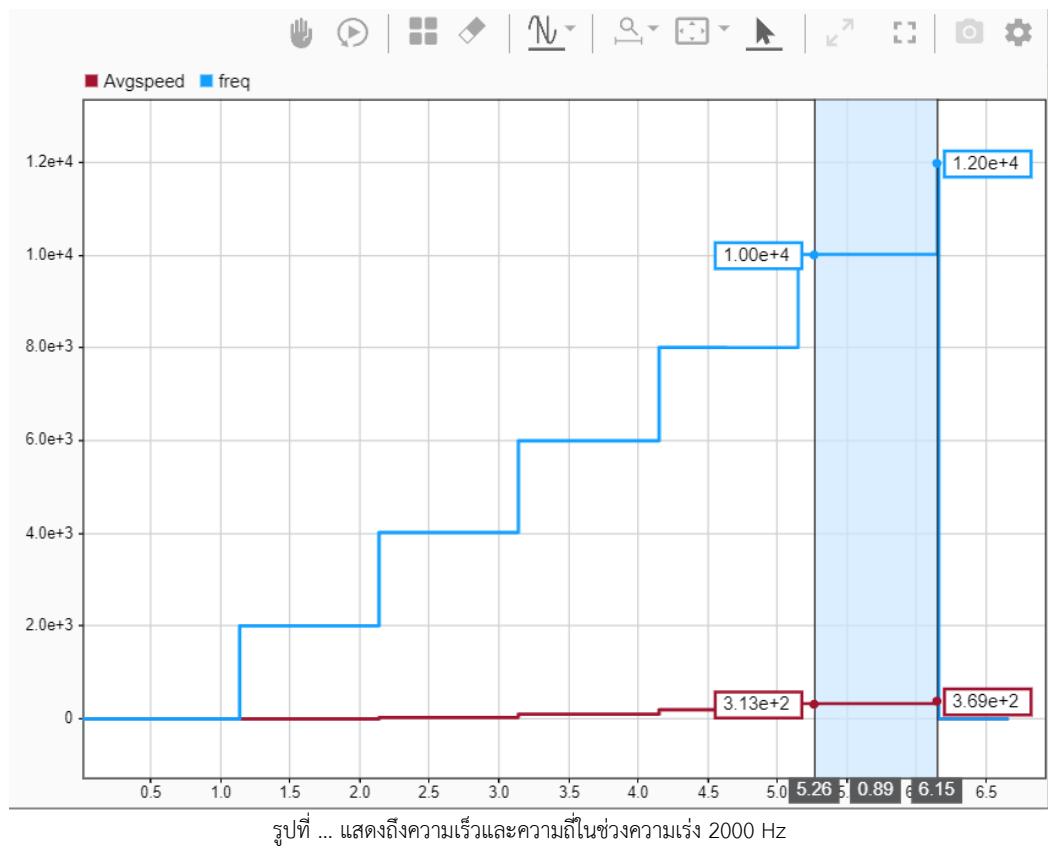


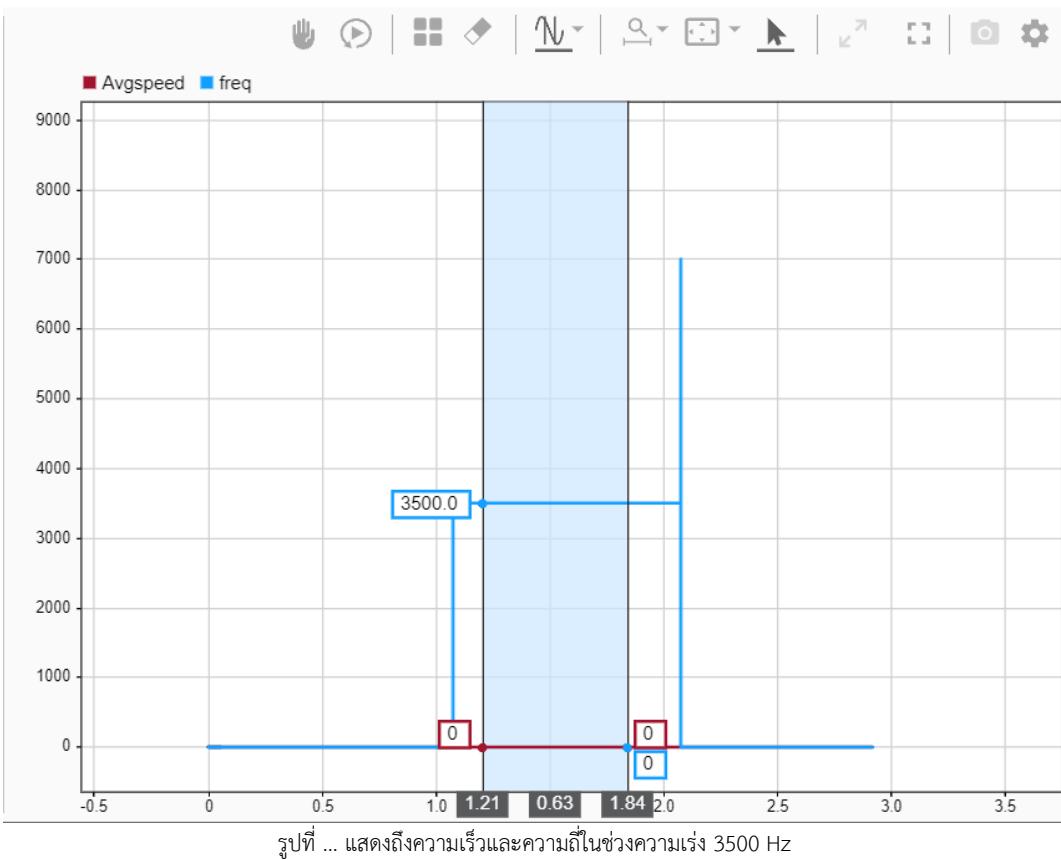
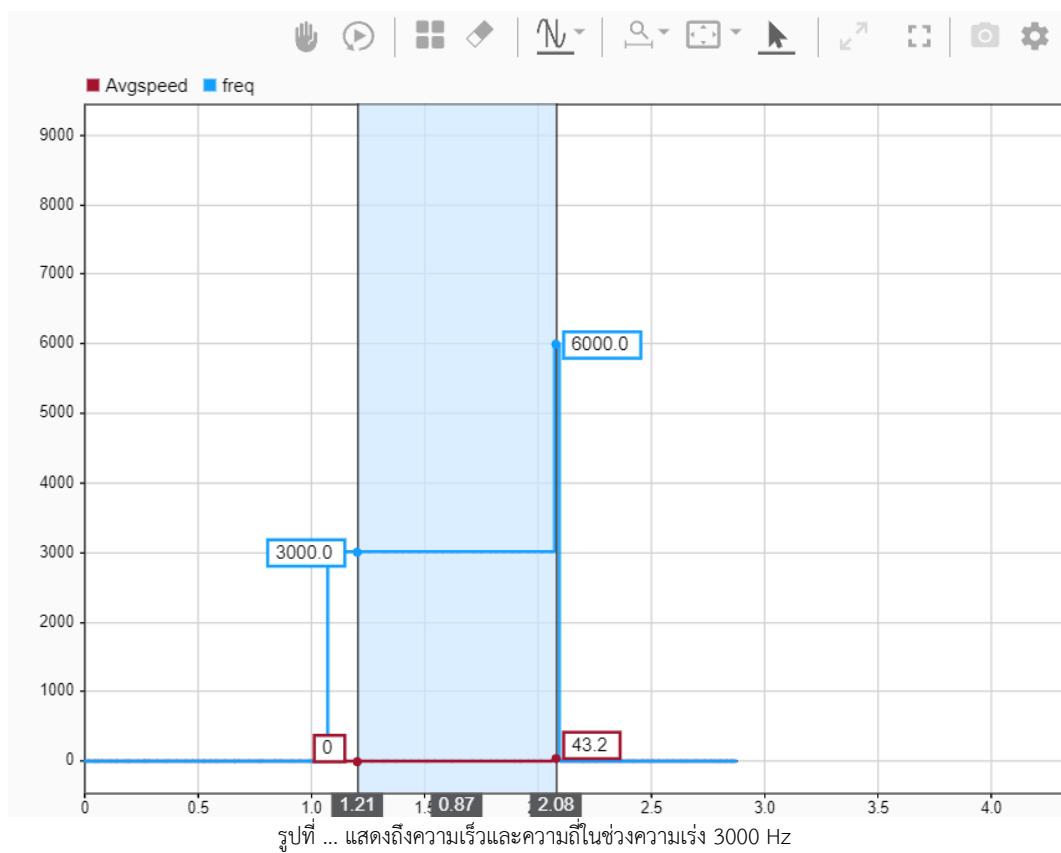
รูปที่ ... แสดงถึงความถี่สูงสุดและความเร่งในช่วงความเร่งต่างๆ ใน Full Step mode

- การเก็บข้อมูลค่าความถี่ที่ได้ต่อความเร็วที่ทำได้มีให้ความเร่งความถี่ในรูปแบบต่างๆ เมื่ออุปกรณ์ในการทำงานแบบ Half Step mode

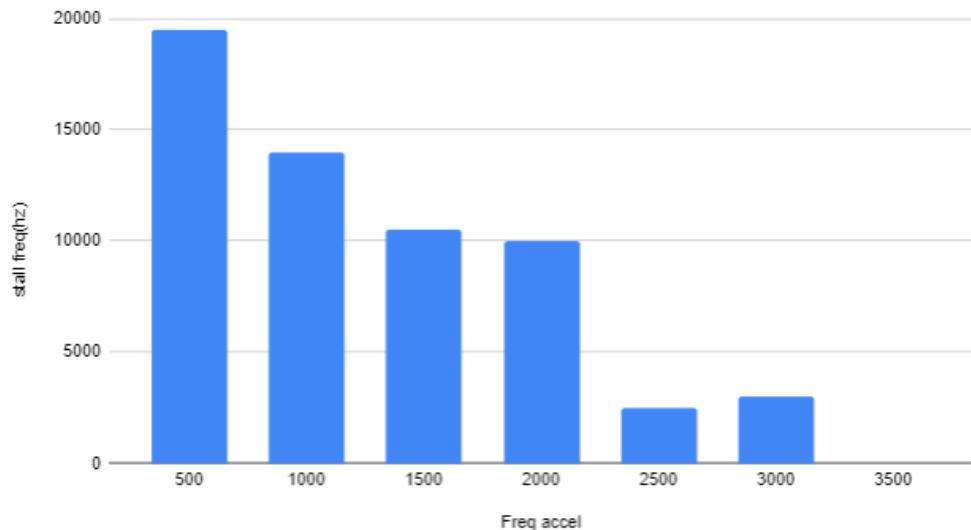






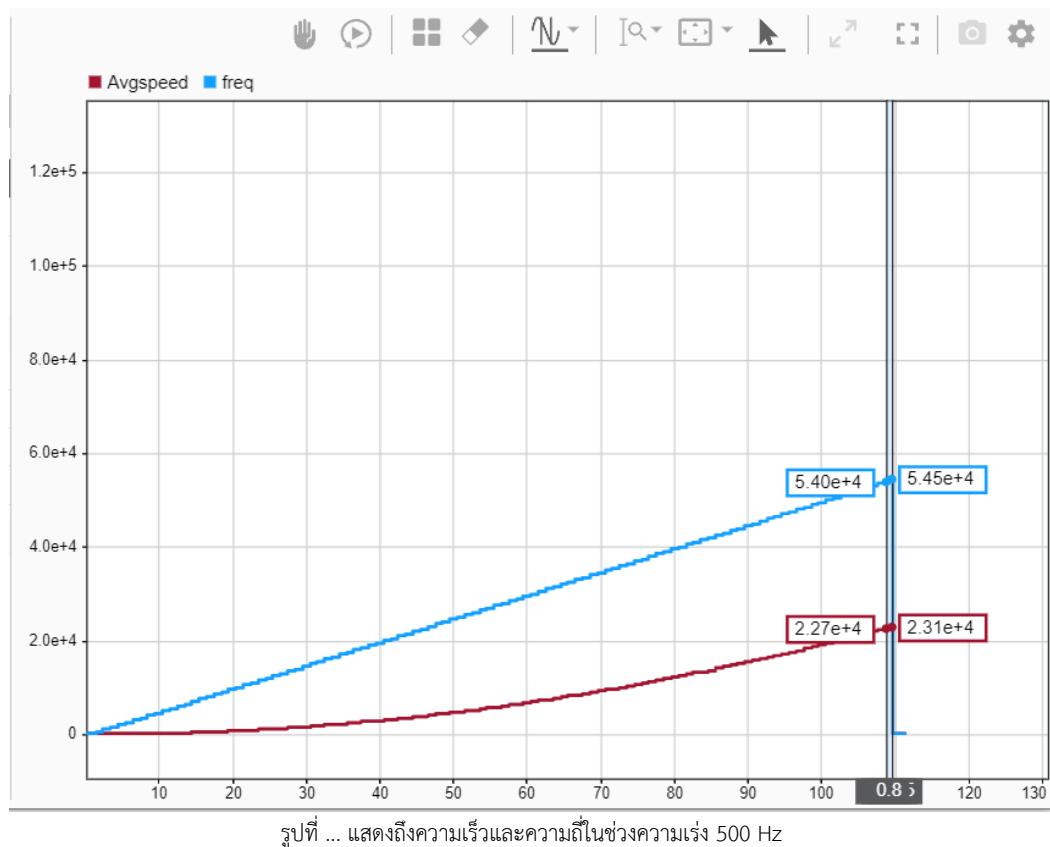


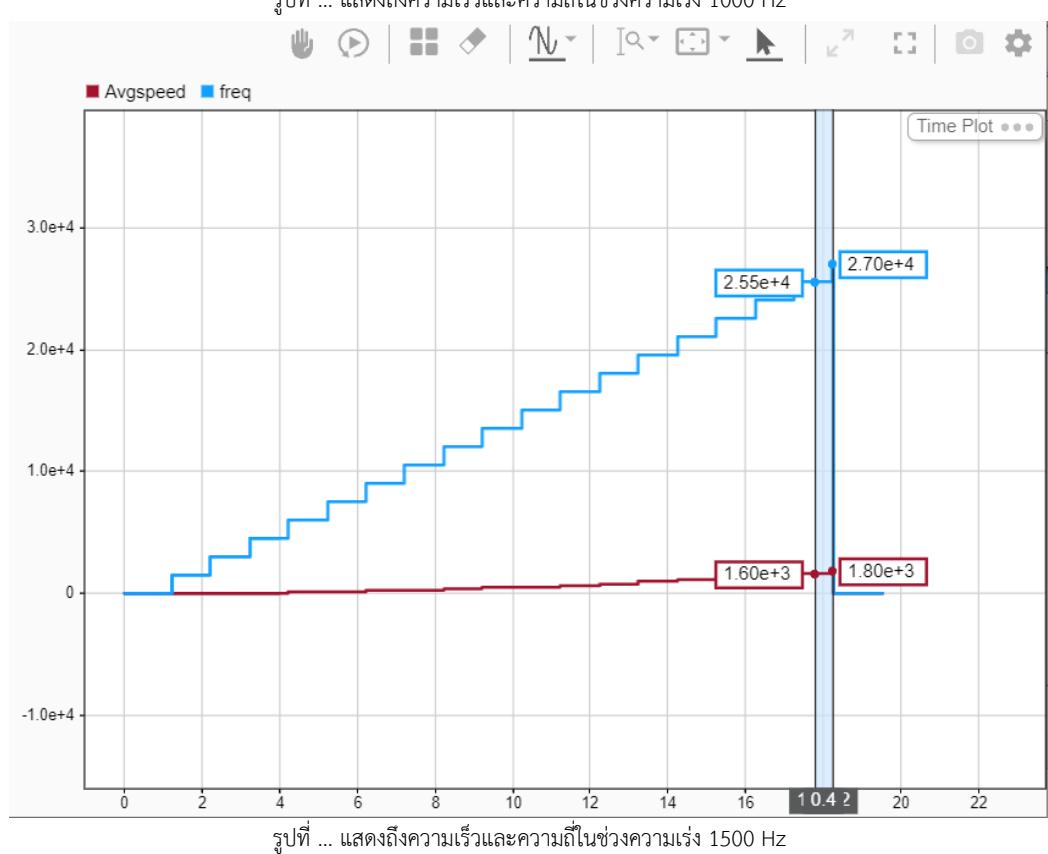
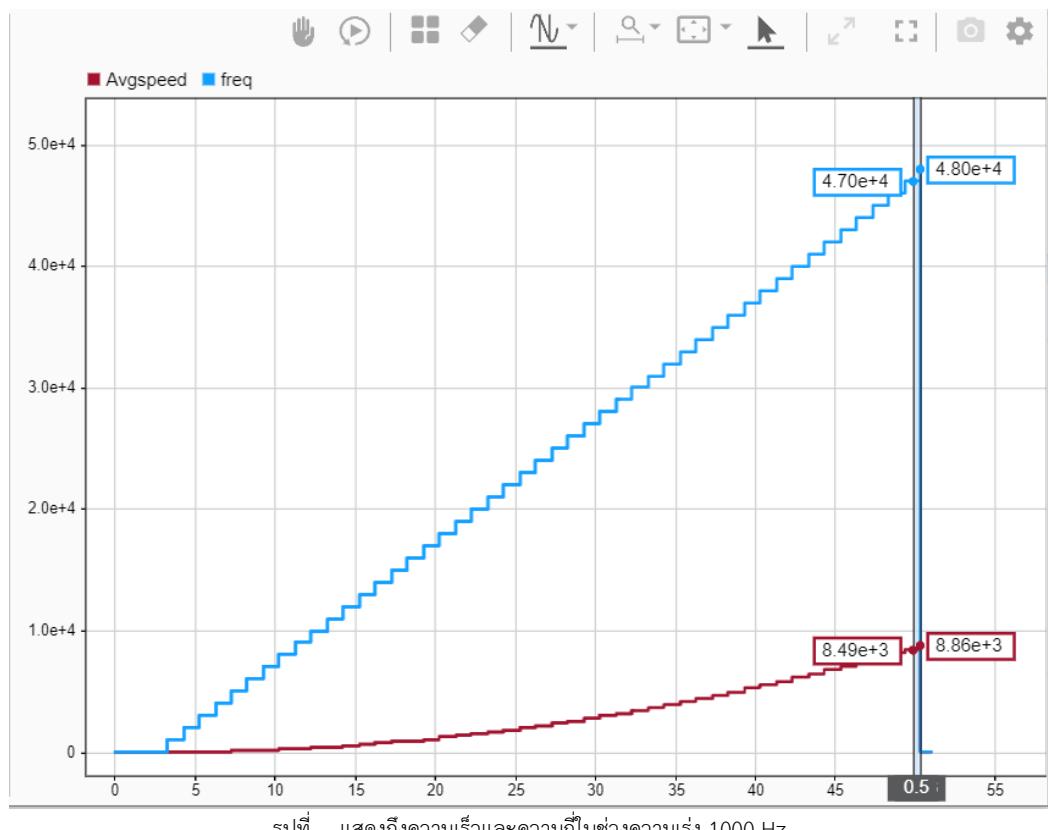
stall freq(hz) กับ Freq accel

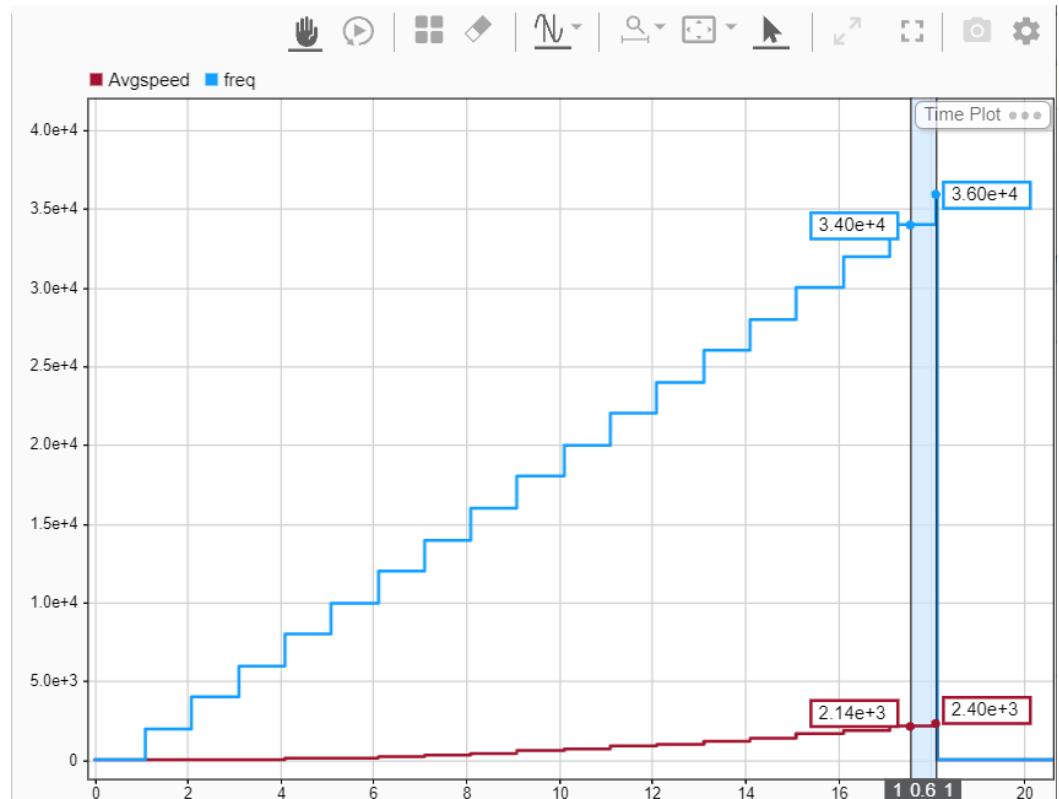


รูปที่ ... แสดงถึงความถี่สูงสุดและความเร่งในช่วงความเร่งต่างๆ ใน Half Step mode

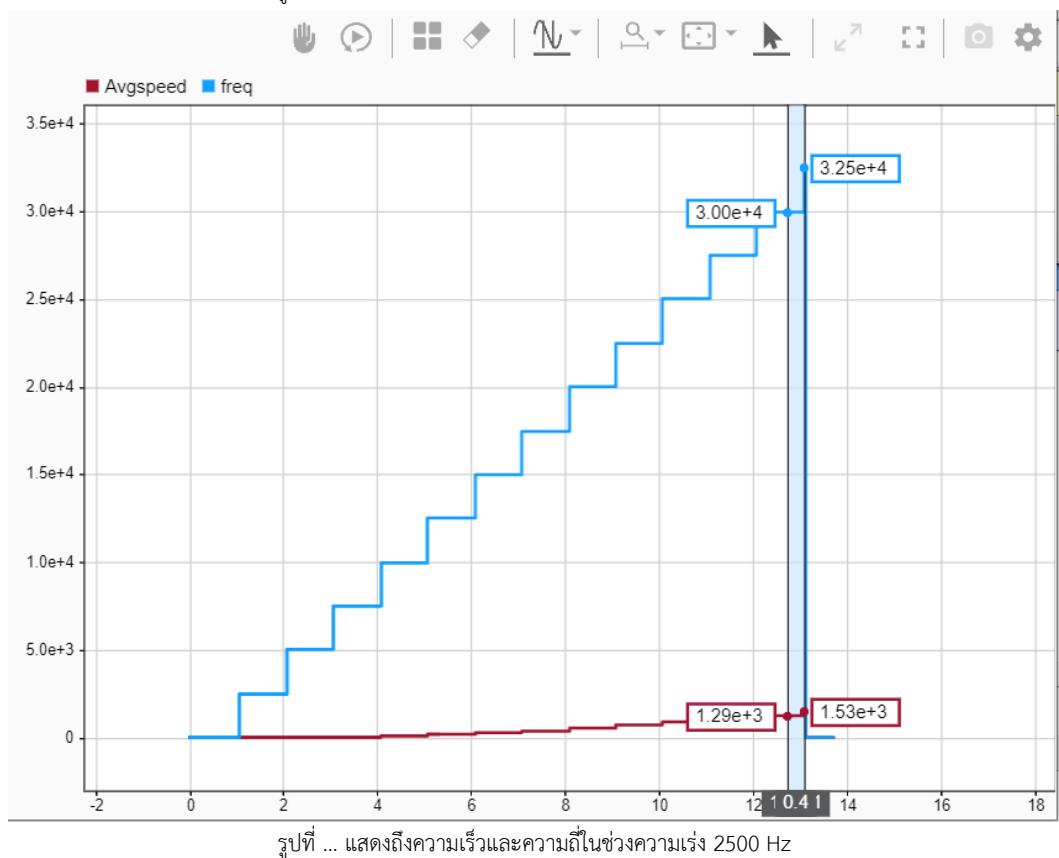
- การเก็บข้อมูลค่าความถี่ที่ได้ต่อความเร็วที่ทำได้มีให้ความเร่งความถี่ในรูปแบบต่างๆ เมื่ออุปกรณ์ในการทำงานแบบ 1/4 Step mode



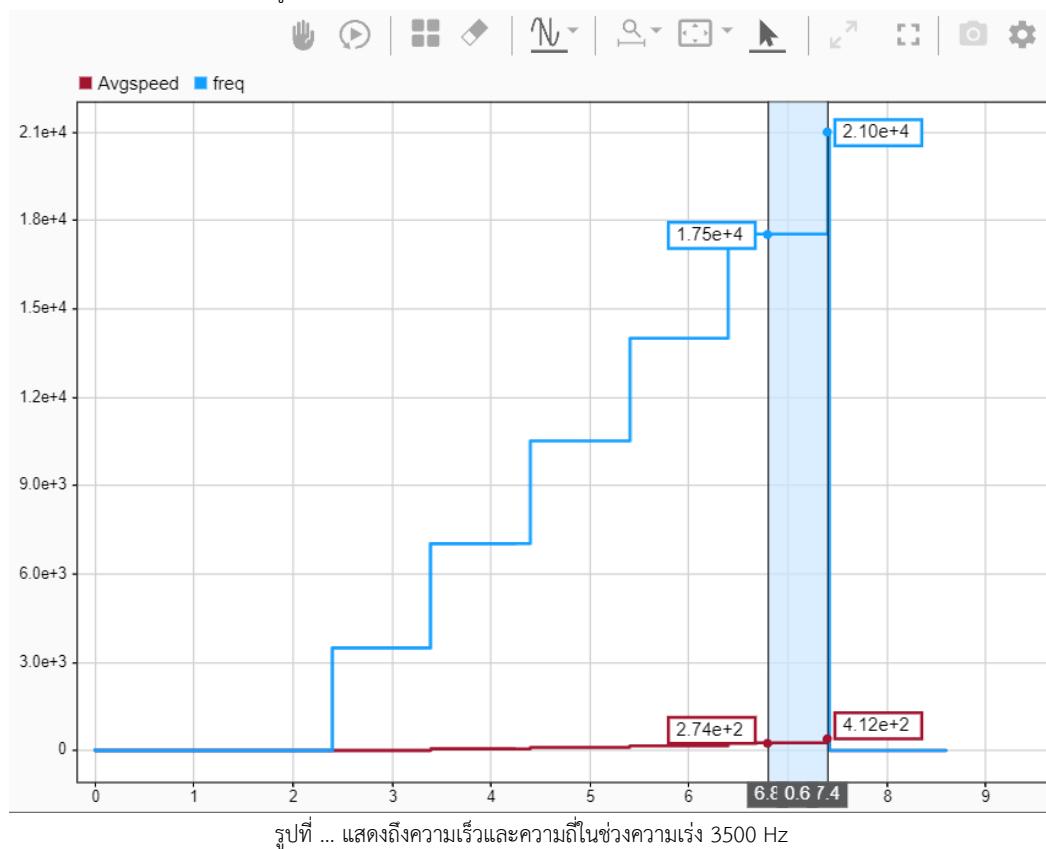
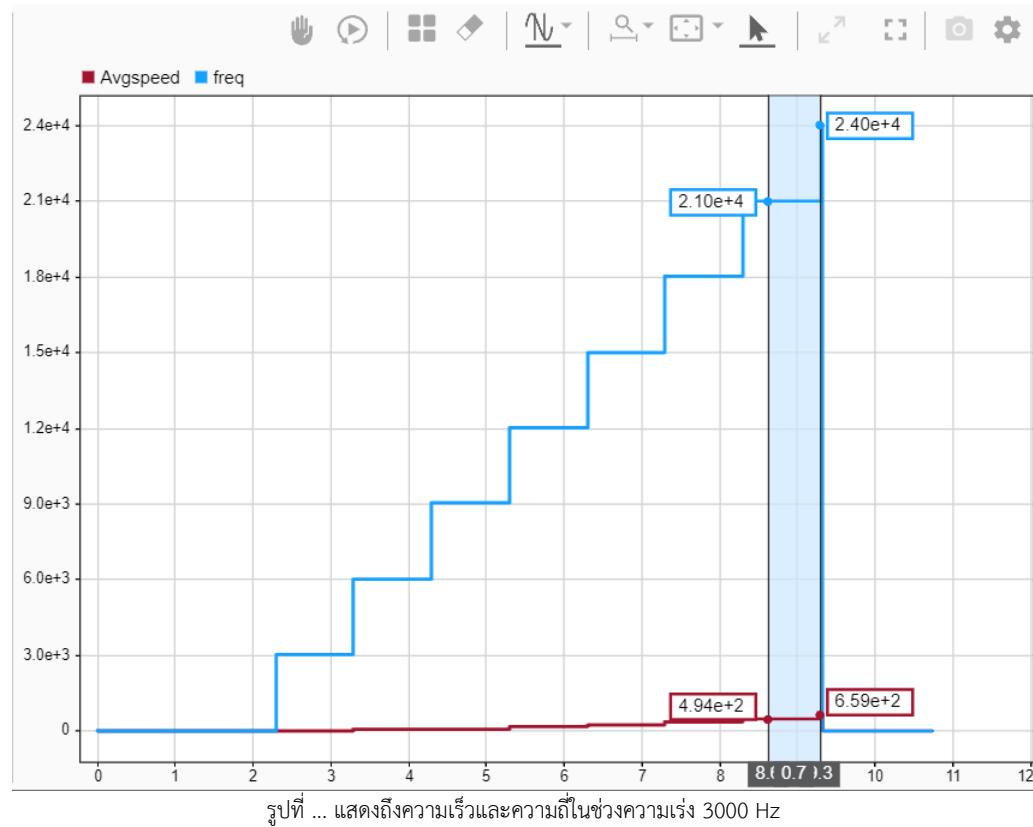


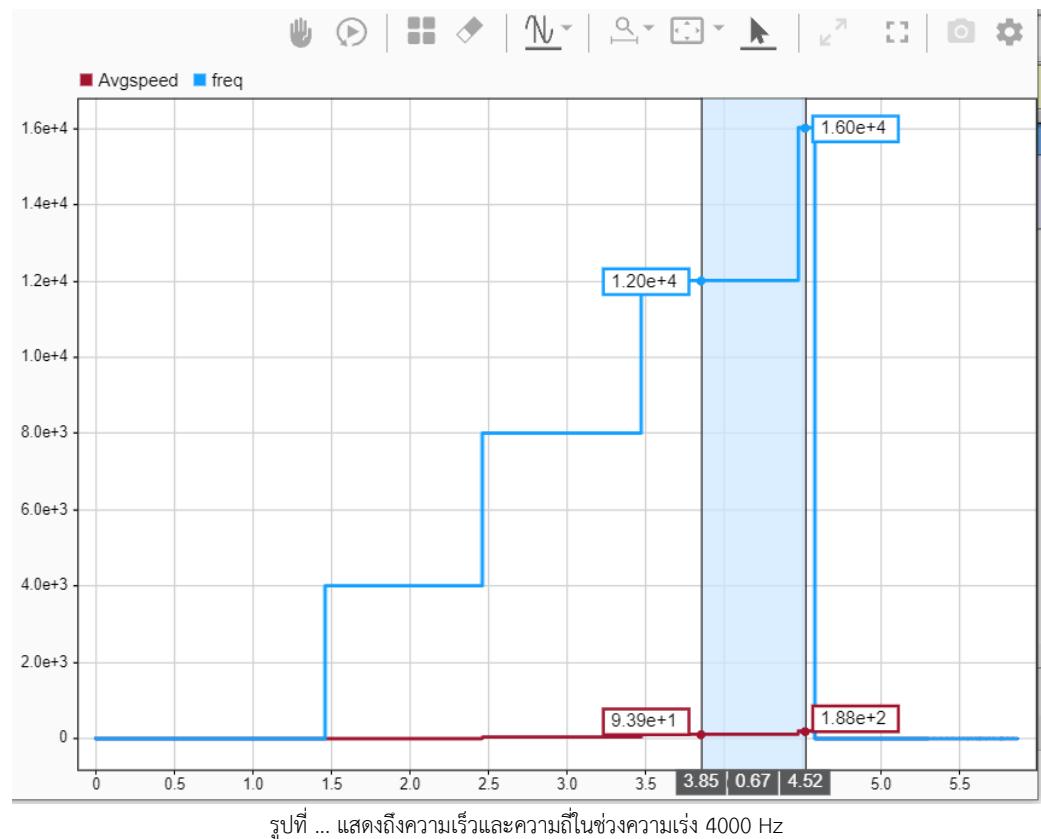


รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 2000 Hz

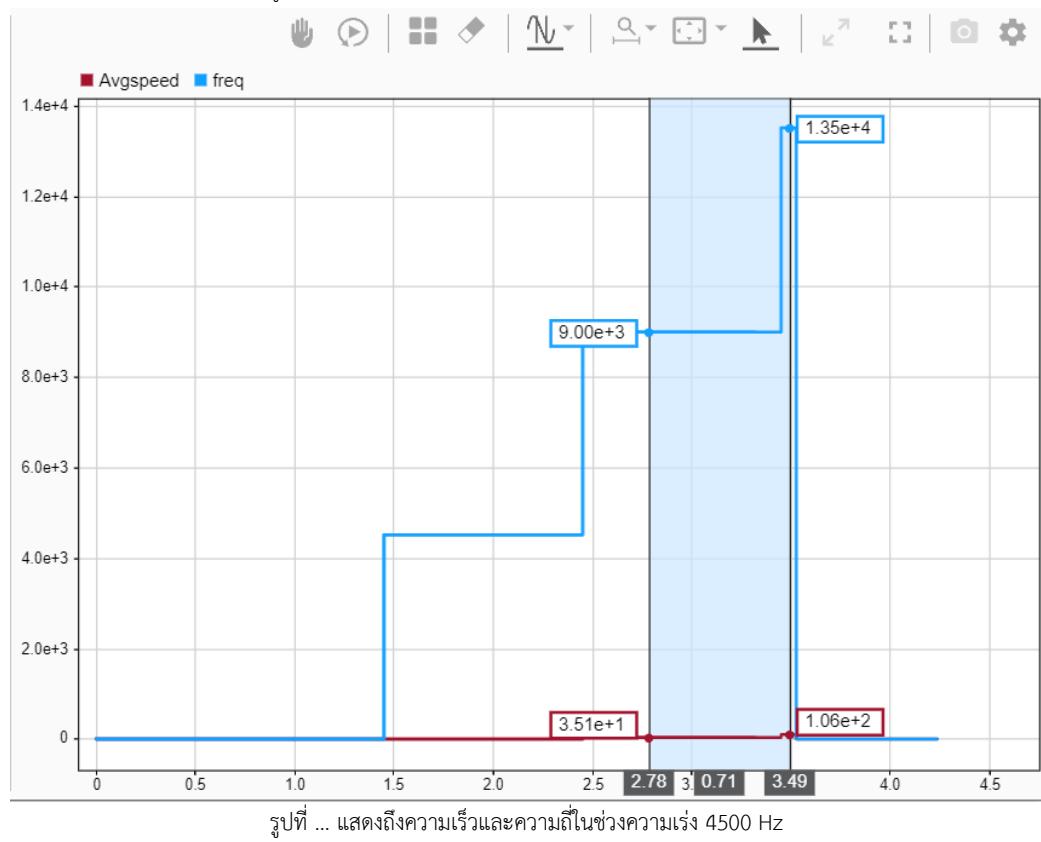


รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 2500 Hz

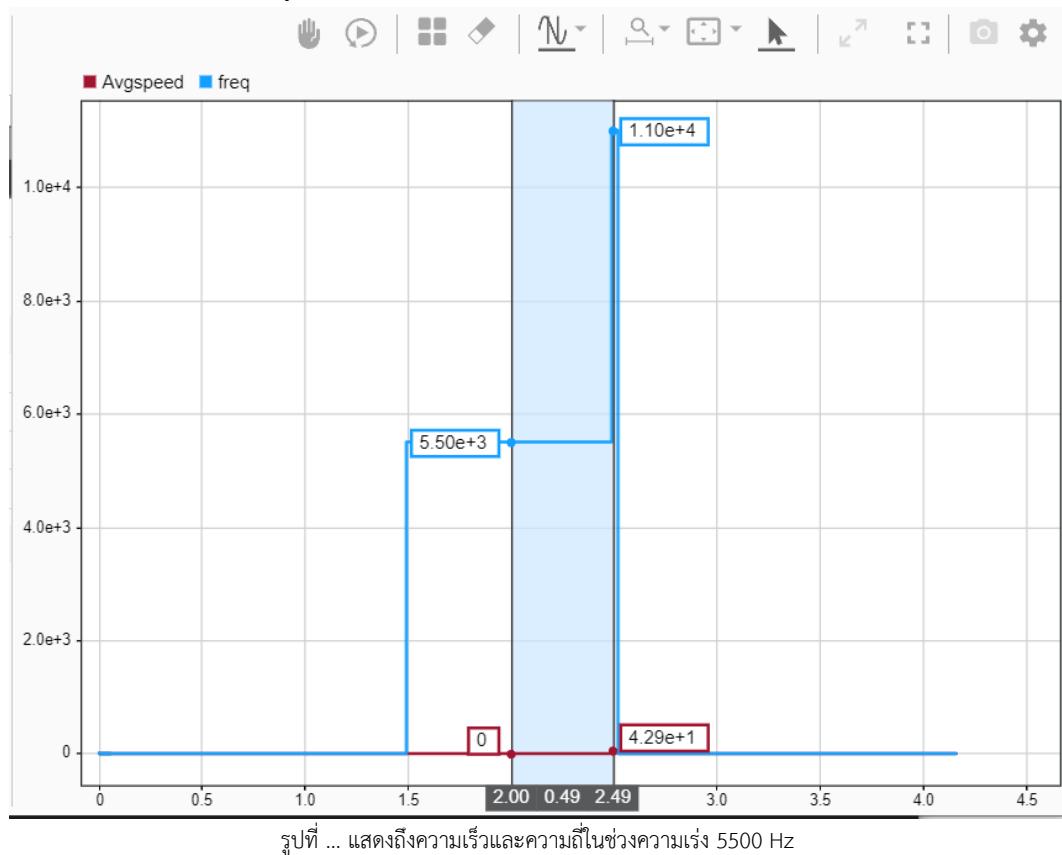
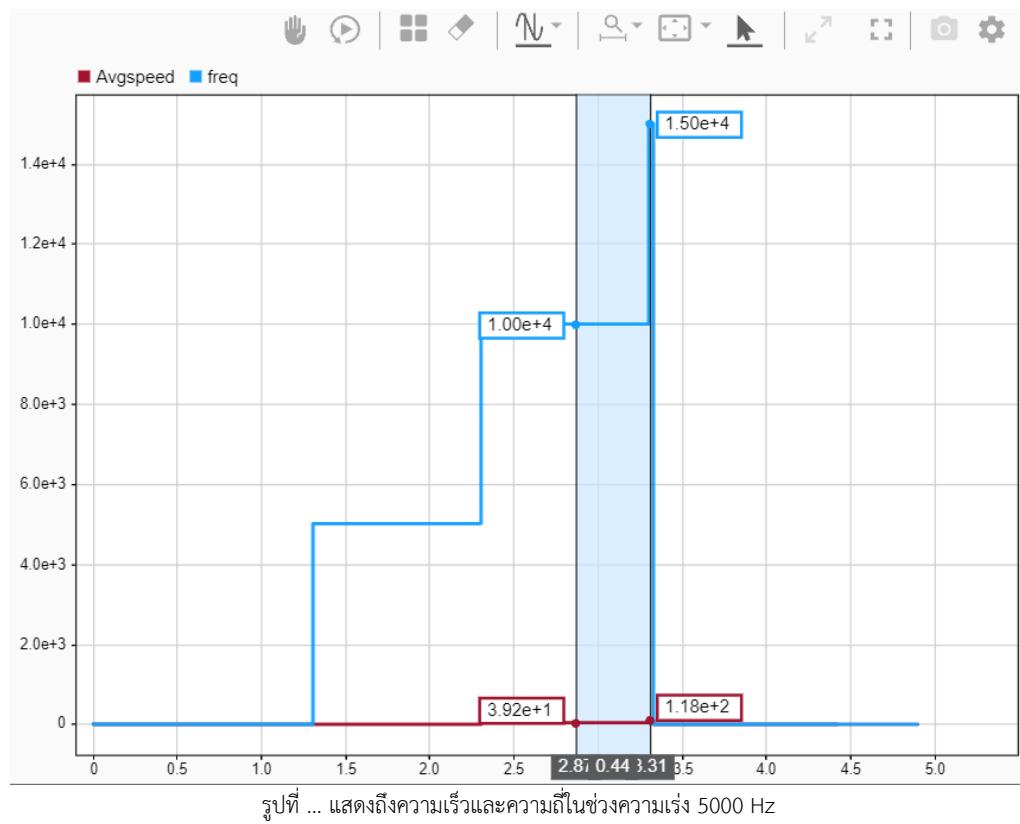


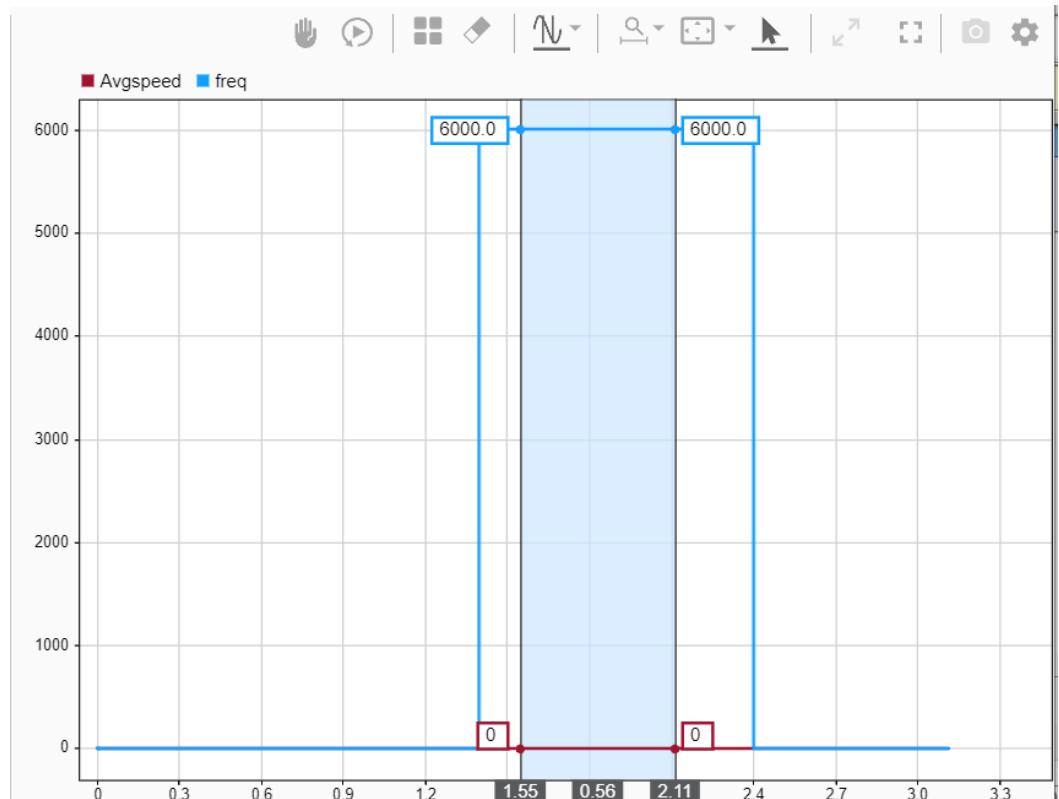


รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 4000 Hz

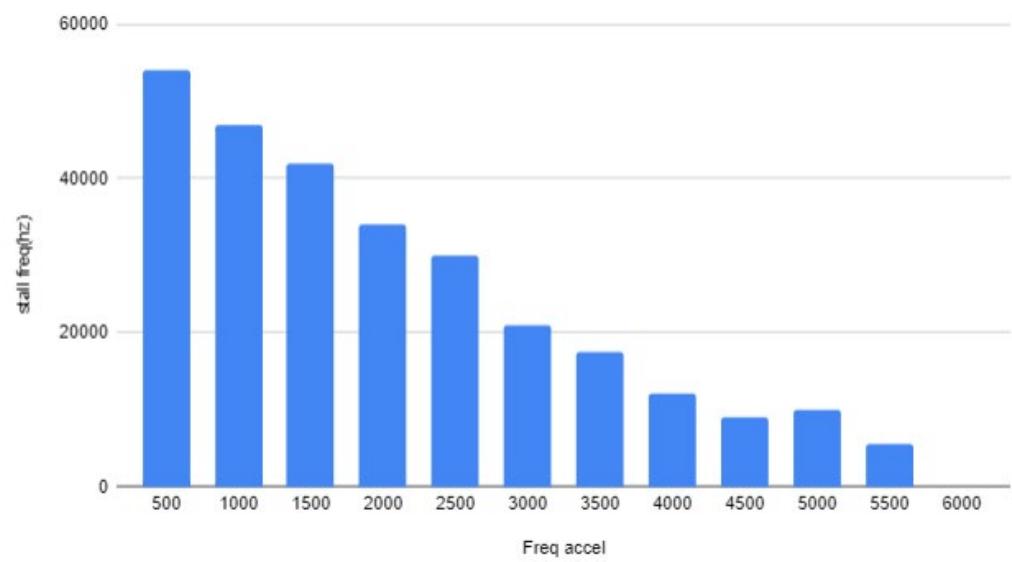


รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 4500 Hz





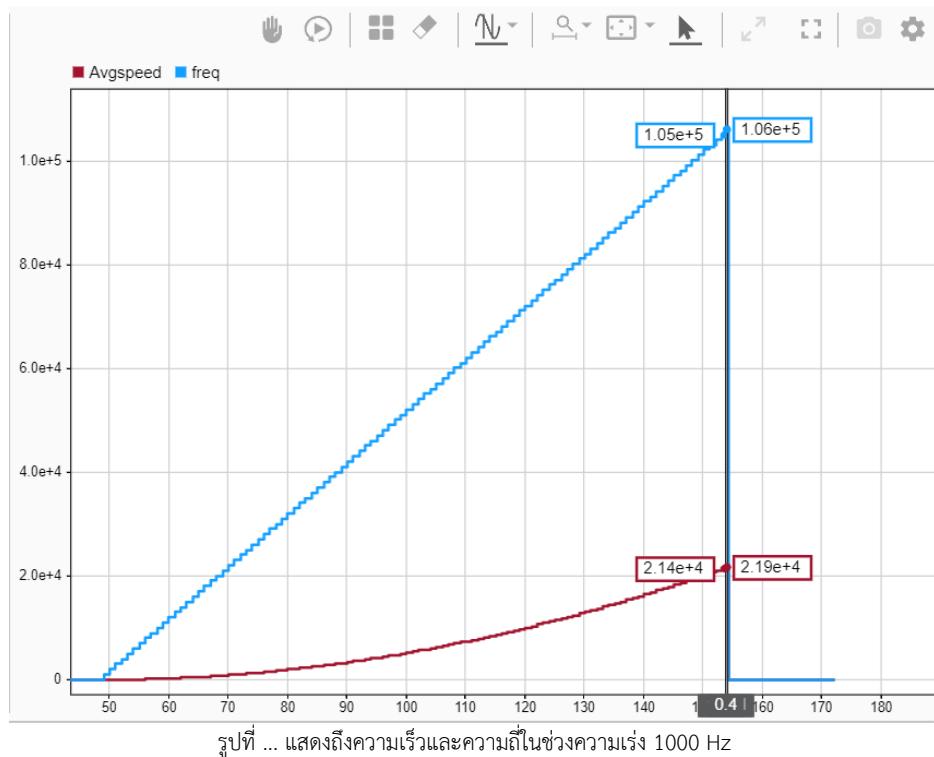
stall freq(hz) กับ Freq accel



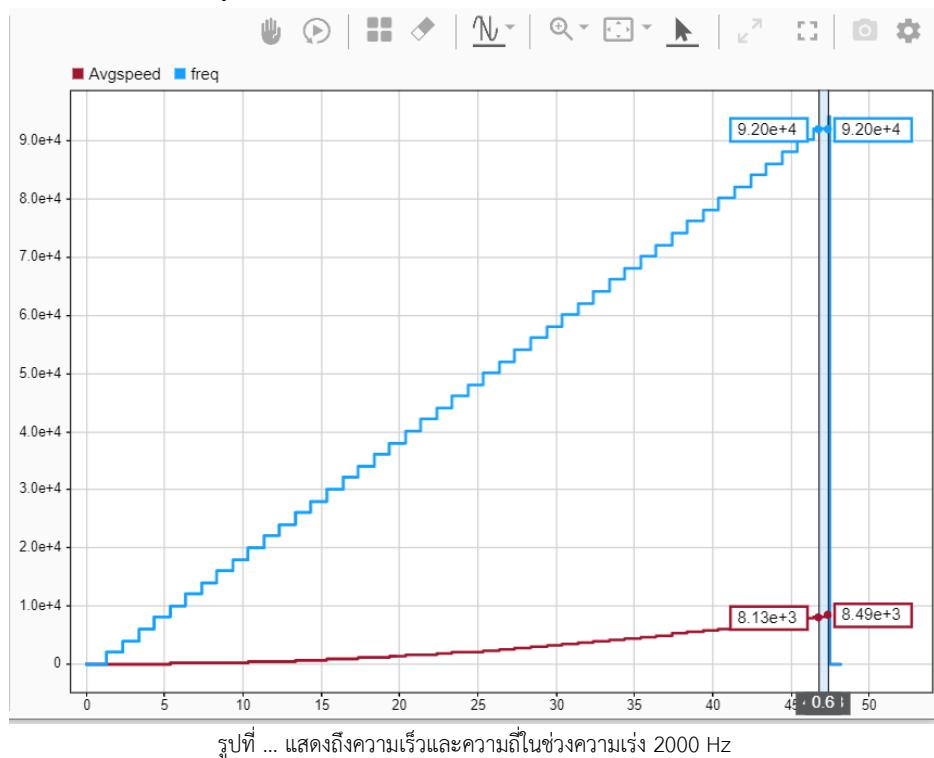
รูปที่ ... แสดงถึงความถี่สูงสุดและความเร่งในช่วงความเร่งต่างๆ ใน 1/4 Step mode

- การเก็บข้อมูลค่าความถี่ที่ได้ต่อความเร็วที่ทำได้เมื่อให้ความเร่งความถี่ในรูปแบบต่างๆ เมื่อยุ่งในการทำงานแบบ 1/8 Step mode โดยเพิ่มค่าความเร่งความถี่ที่ลักษณะ 1000 เพราะเมื่อตรวจสอบจากรูปแบบ

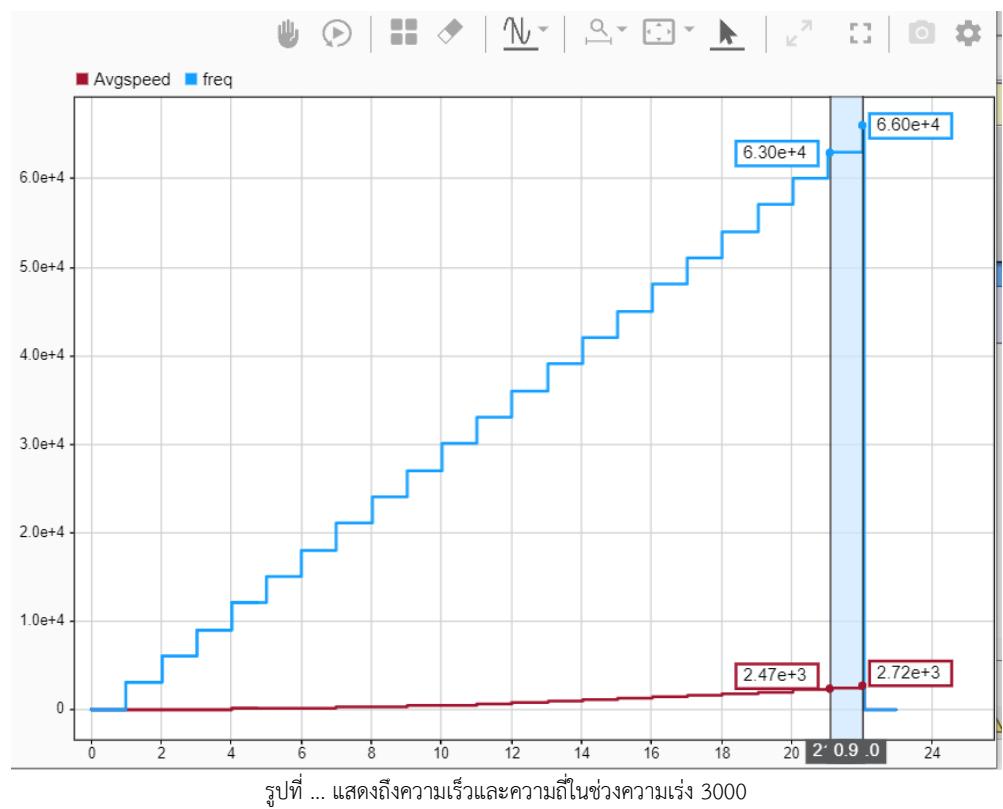
1/4 Step mode จะเริ่มไม่เห็นความแตกต่างของการเปรียบเทียบค่าแบบบล๊อคเมื่อเปรียบเทียบกัน
แบบความแตกต่าง 500 Hz แล้ว



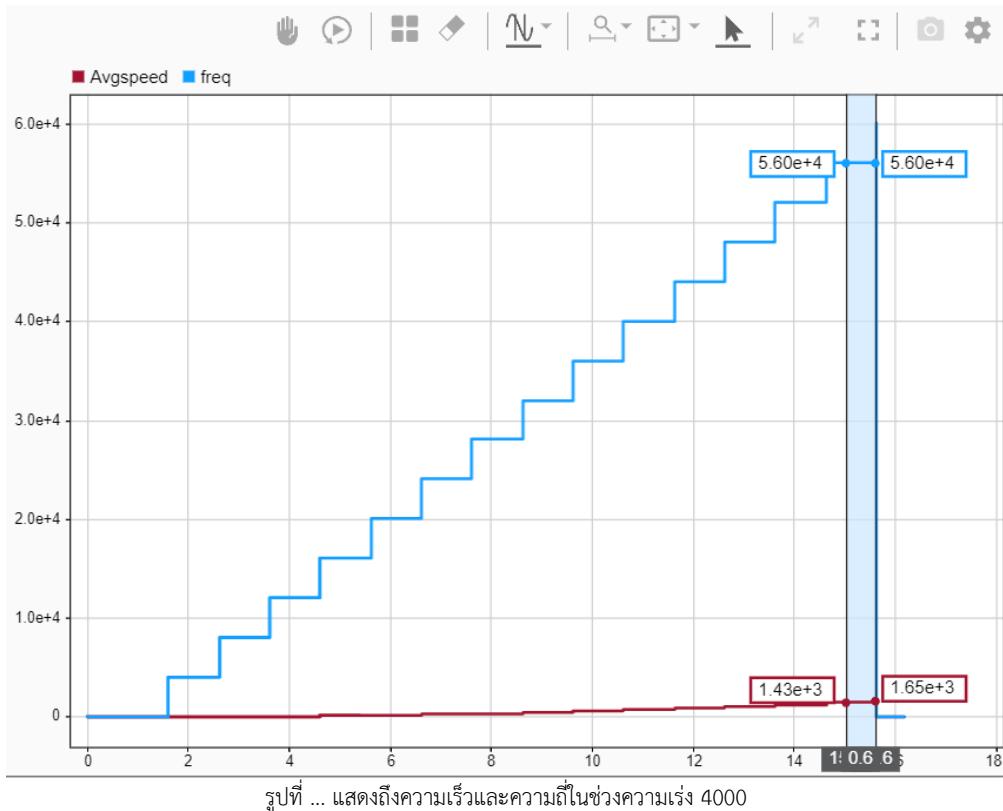
รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 1000 Hz



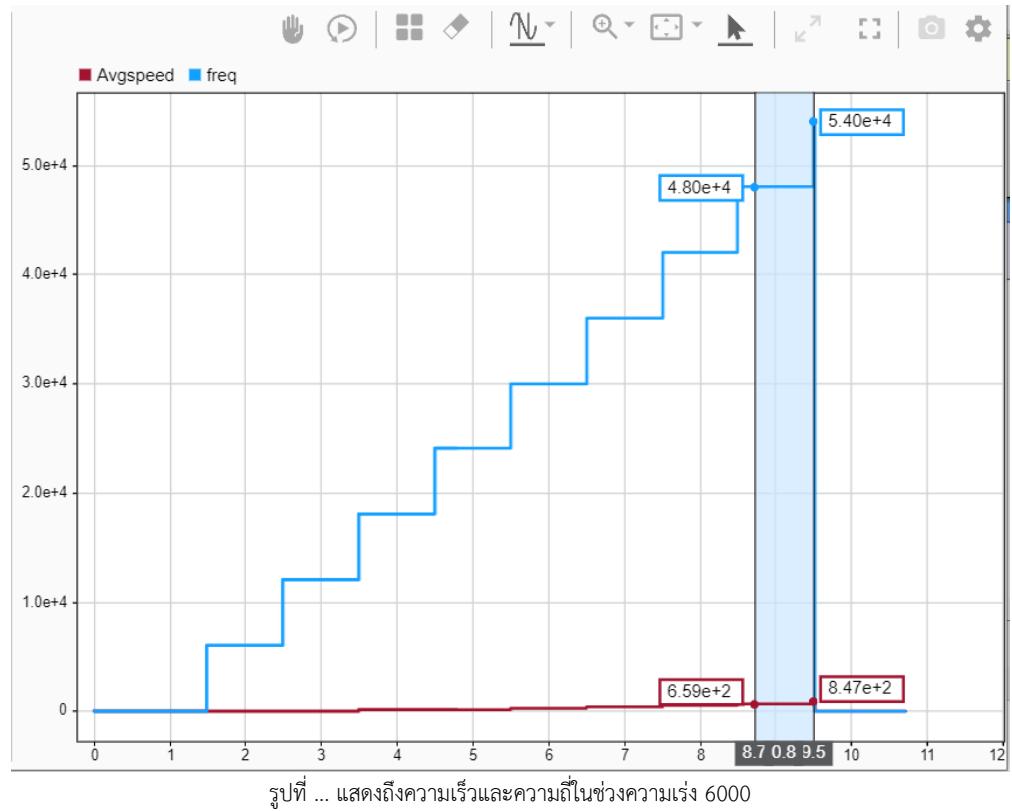
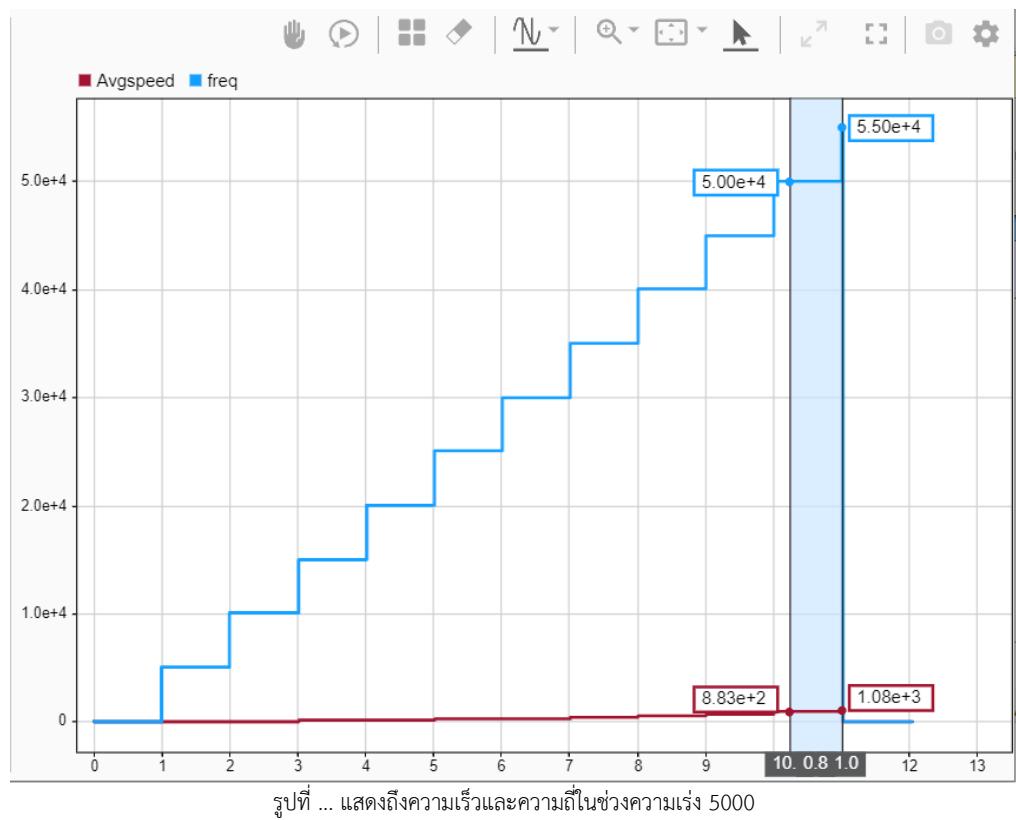
รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 2000 Hz

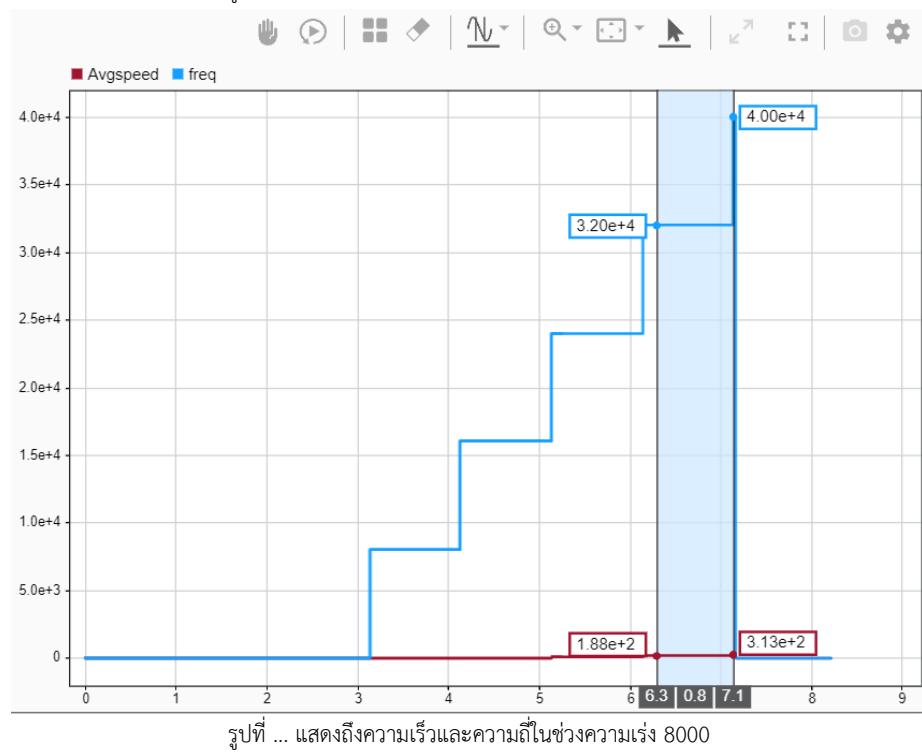
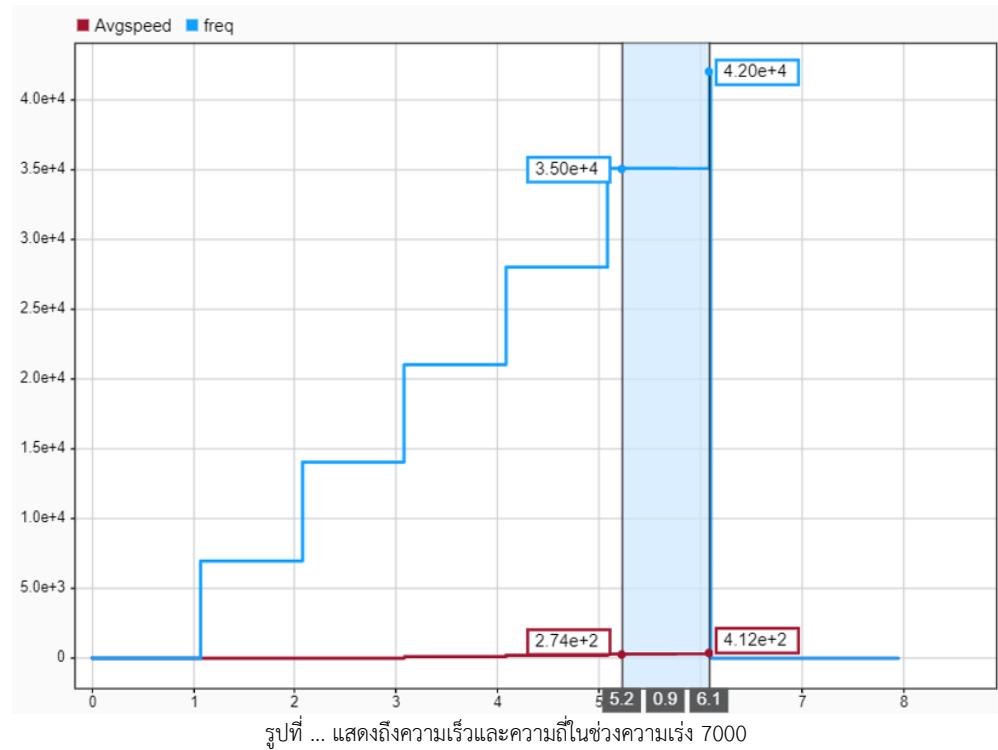


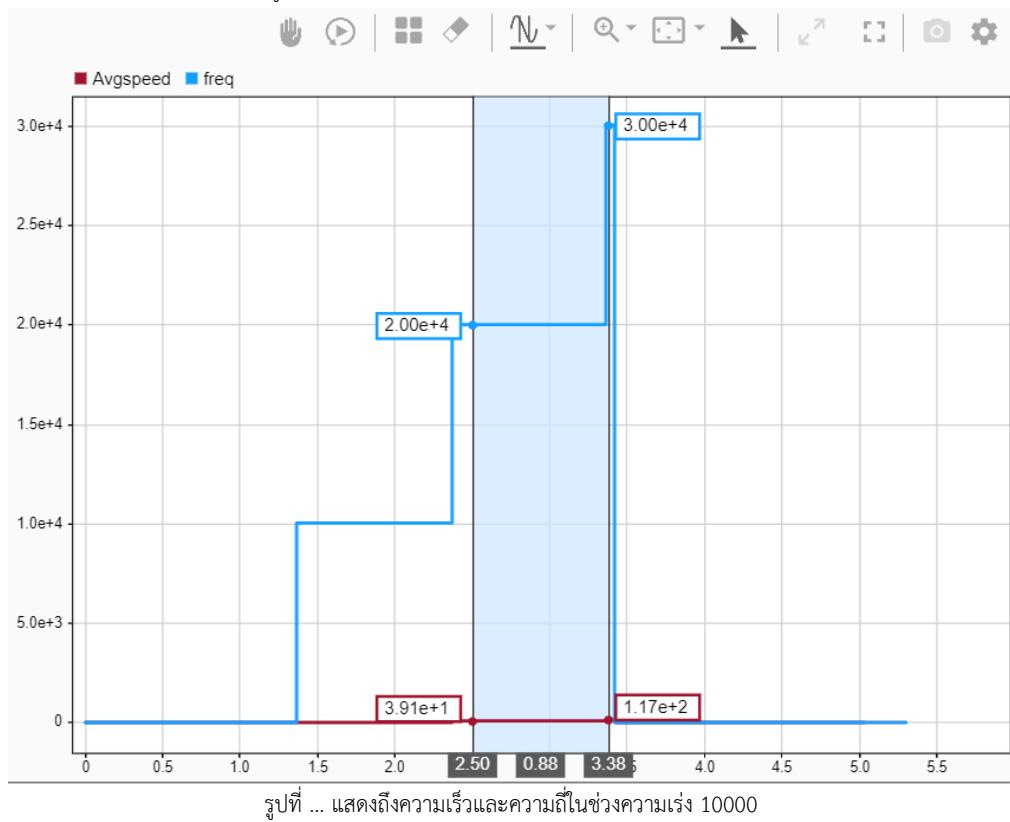
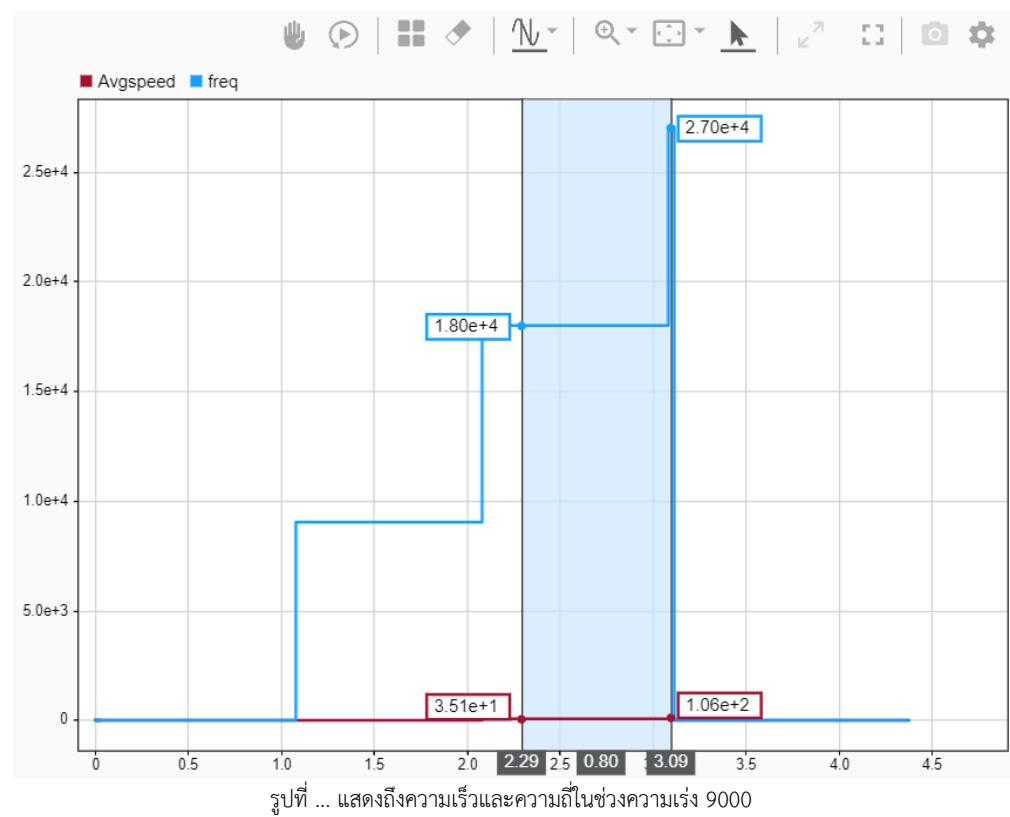
รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 3000

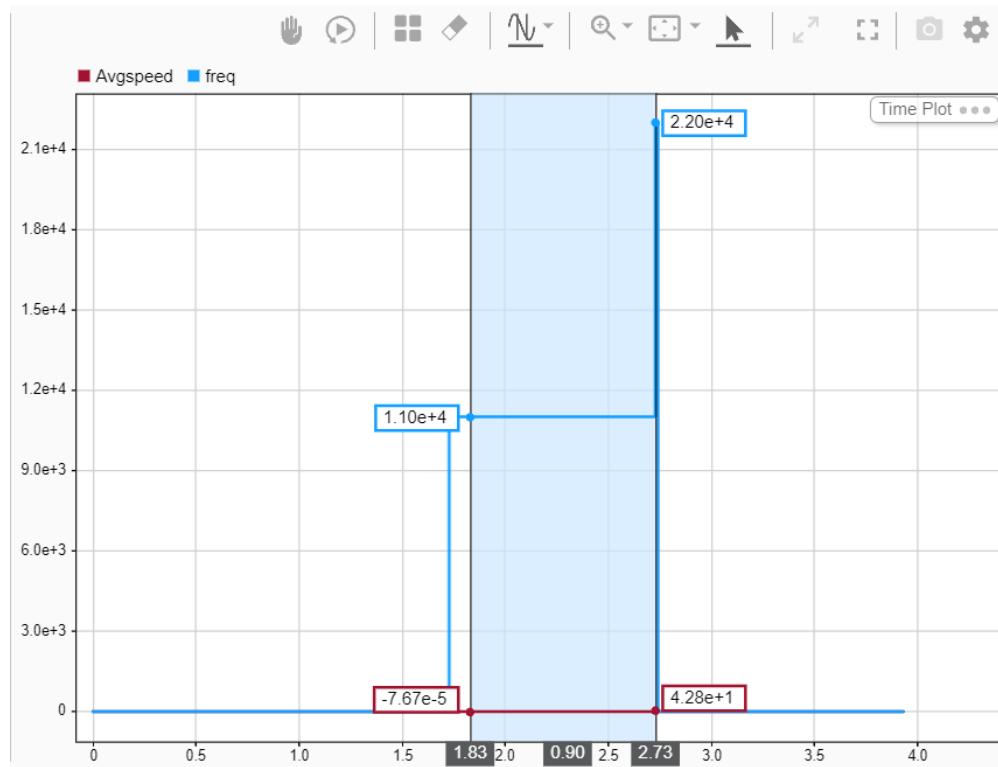


รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 4000

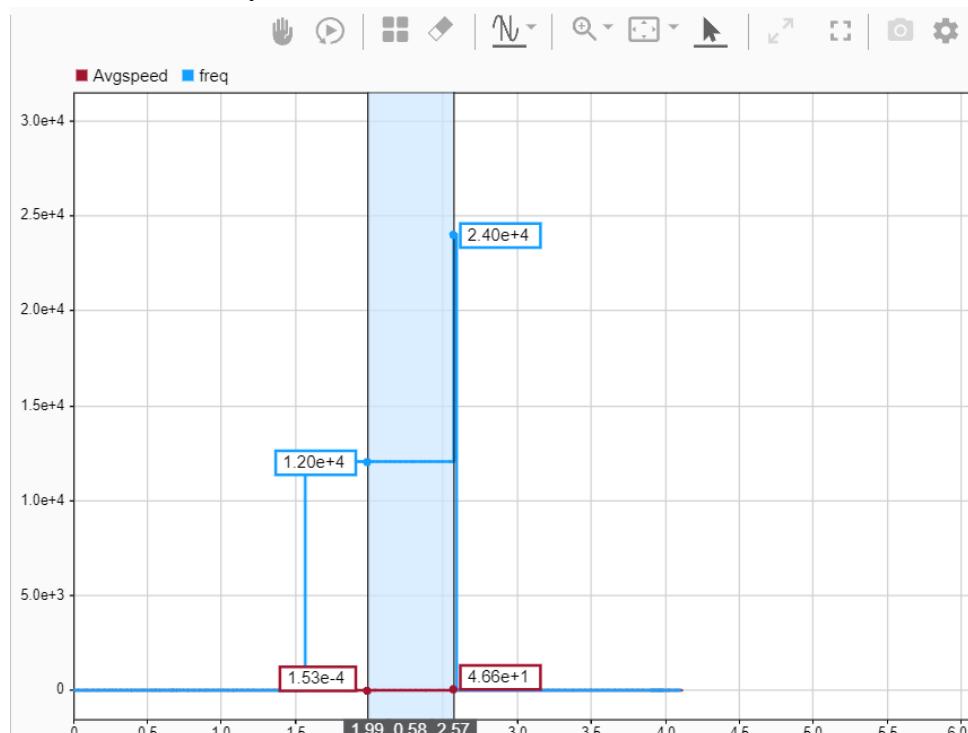




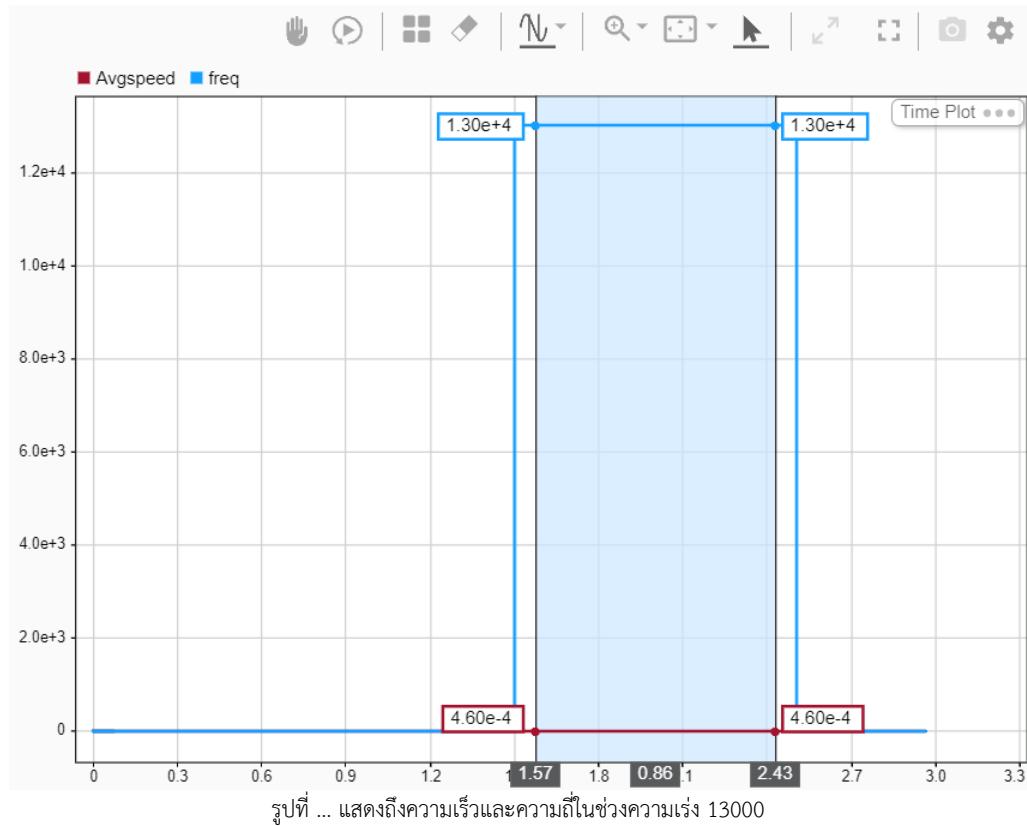




รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 11000

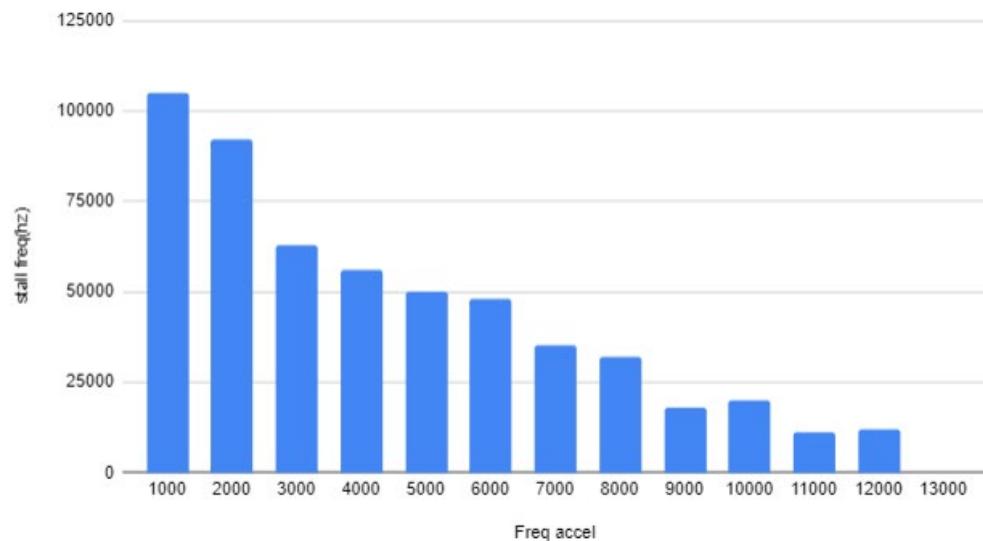


รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 12000



รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 13000

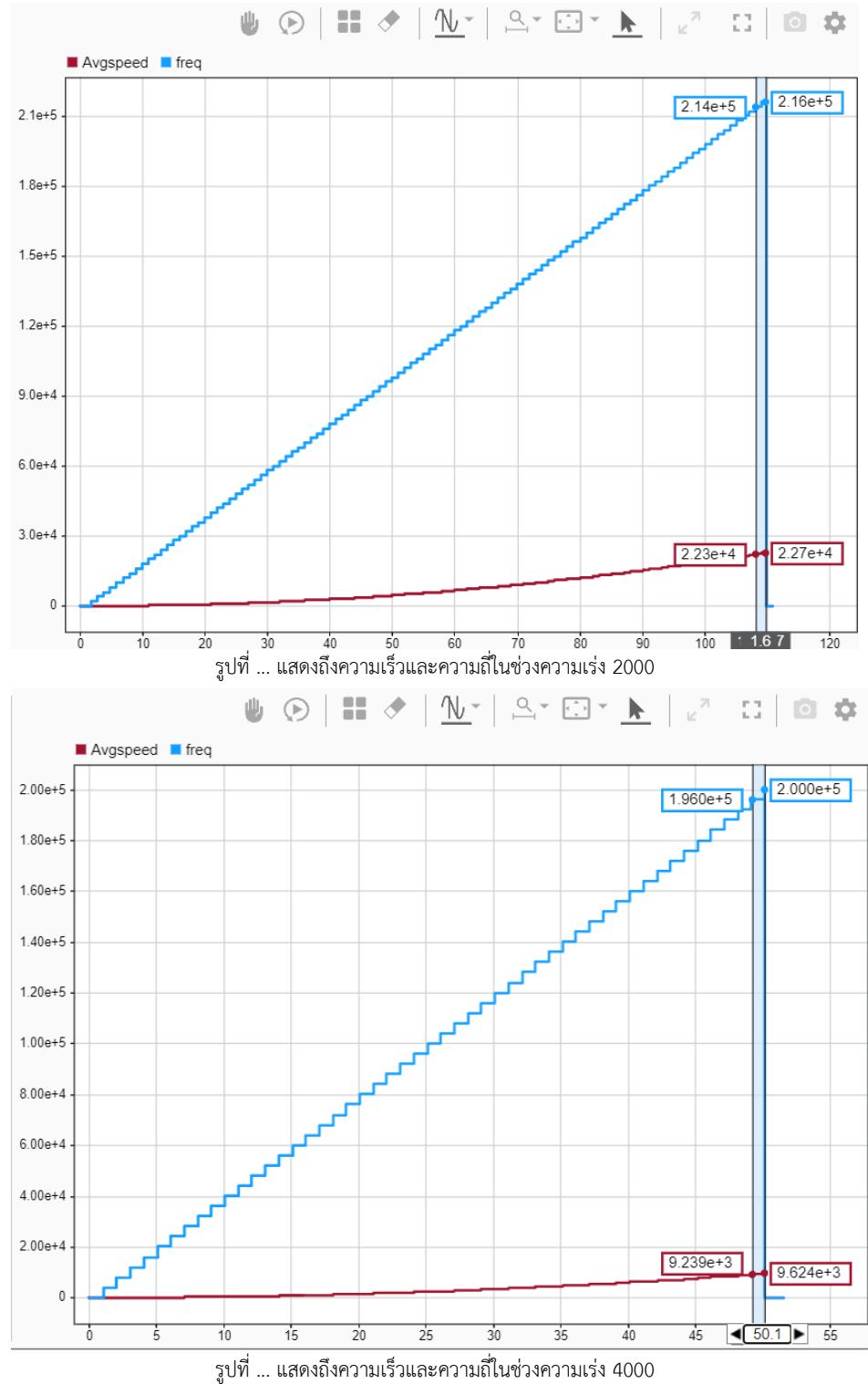
stall freq(hz) กับ Freq accel

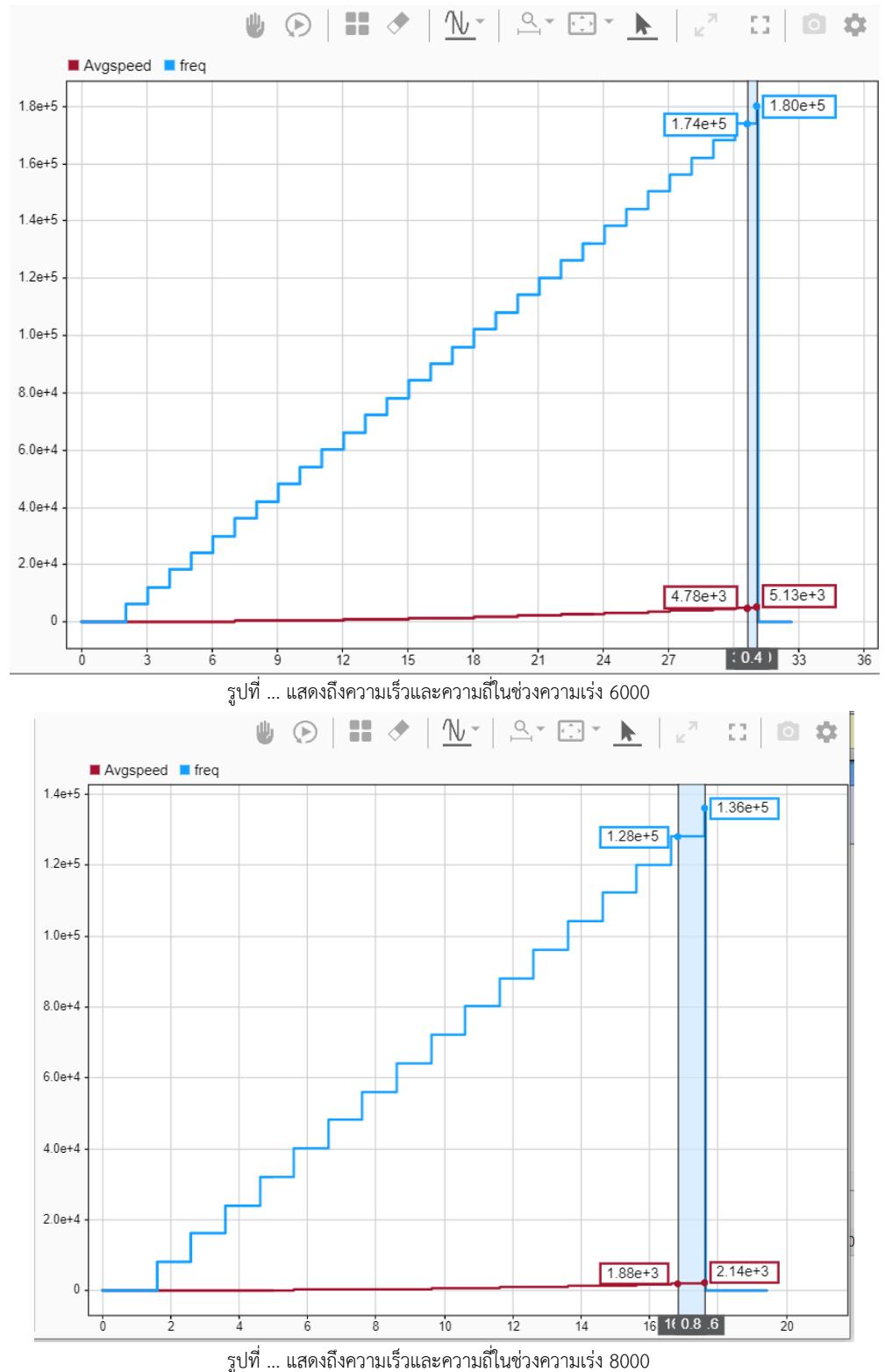


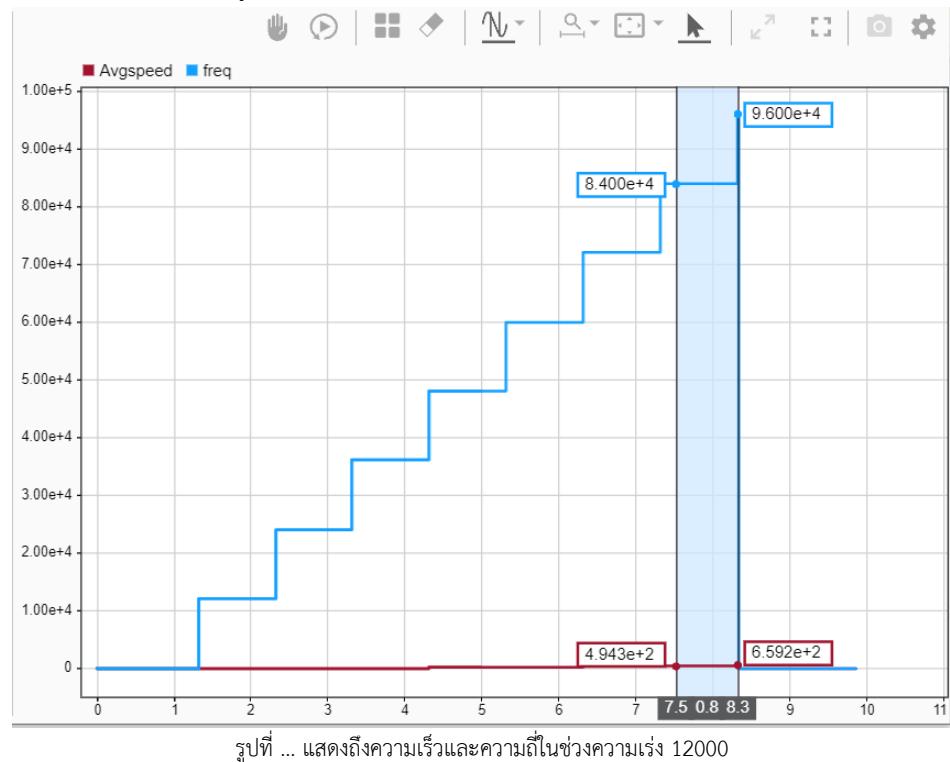
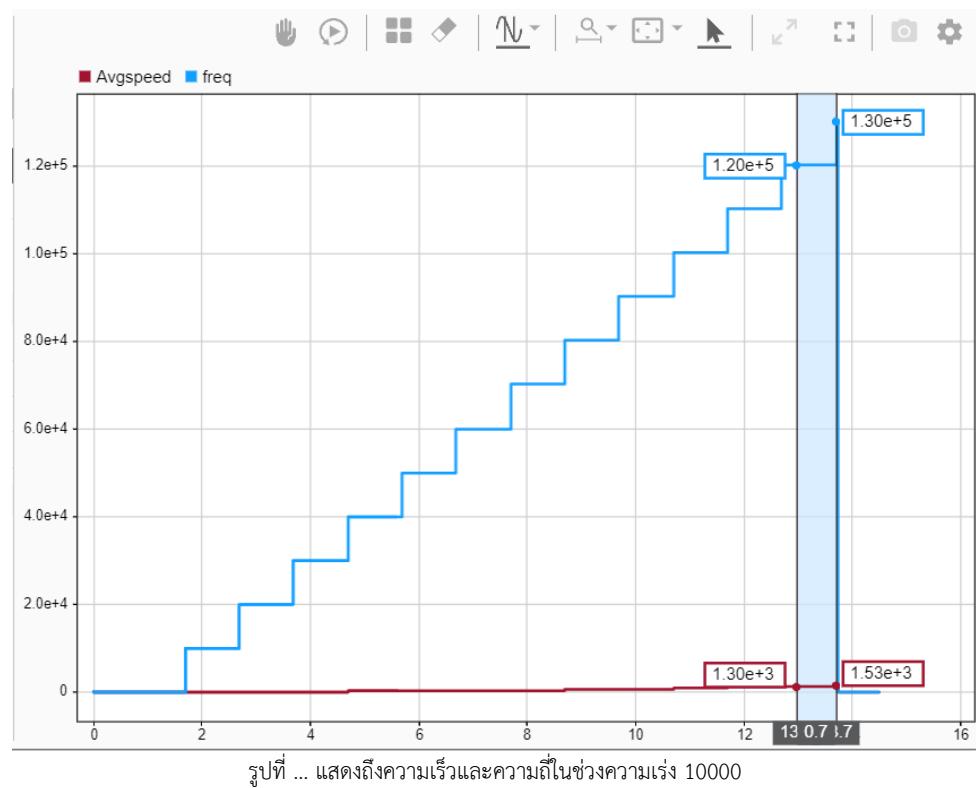
รูปที่ ... แสดงถึงความถี่สูงสุดและความเร่งในช่วงความเร่งต่างๆ ใน 1/8 Step mode

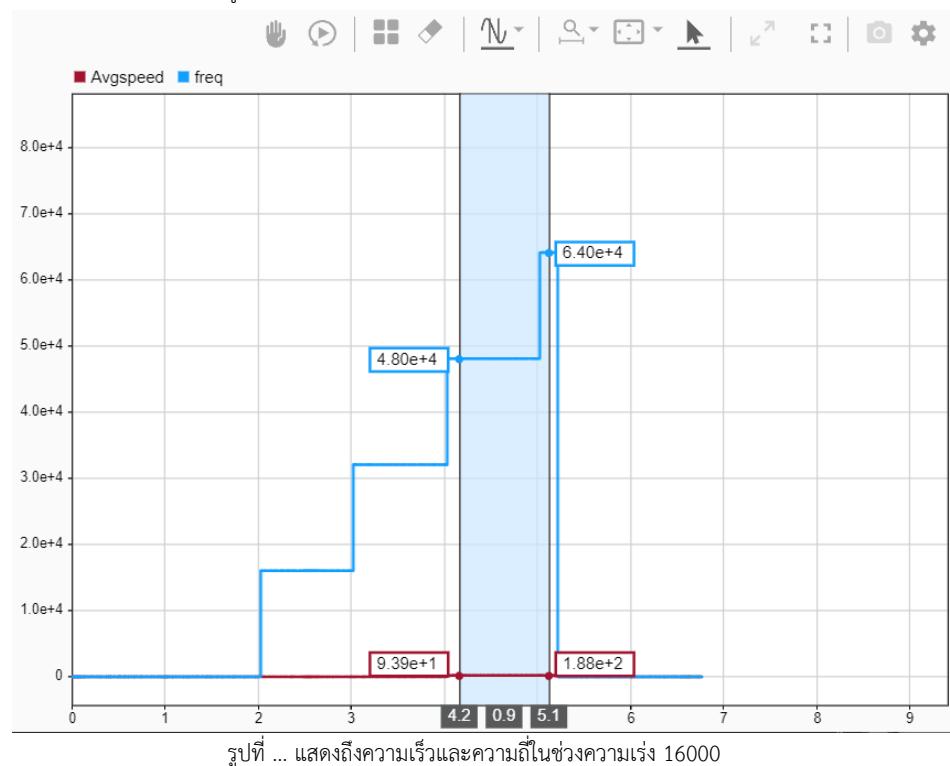
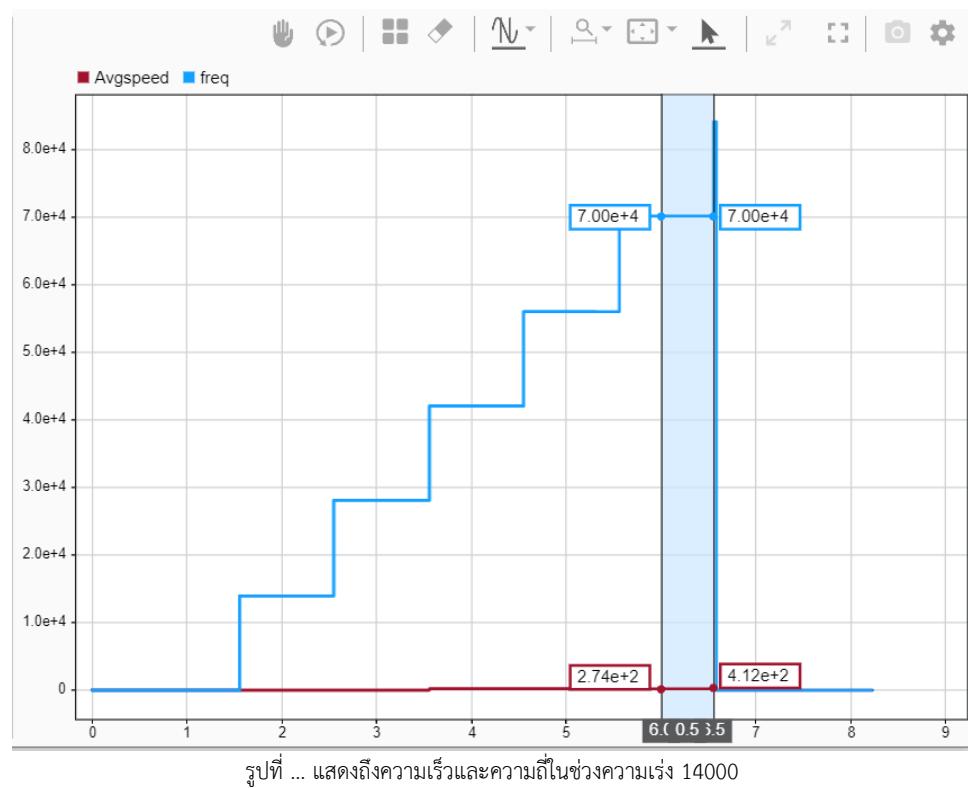
5. การเก็บข้อมูลค่าความถี่ที่ได้ต่อความเร็วที่ทำได้เมื่อให้ความเร่งความถี่ในรูปแบบต่างๆ เมื่ออุปกรณ์ในการทำงานแบบ 1/16 Step mode โดยเพิ่มค่าความเร่งความถี่ที่ลักษณะ 2000 เพราะเมื่อตรวจสอบ

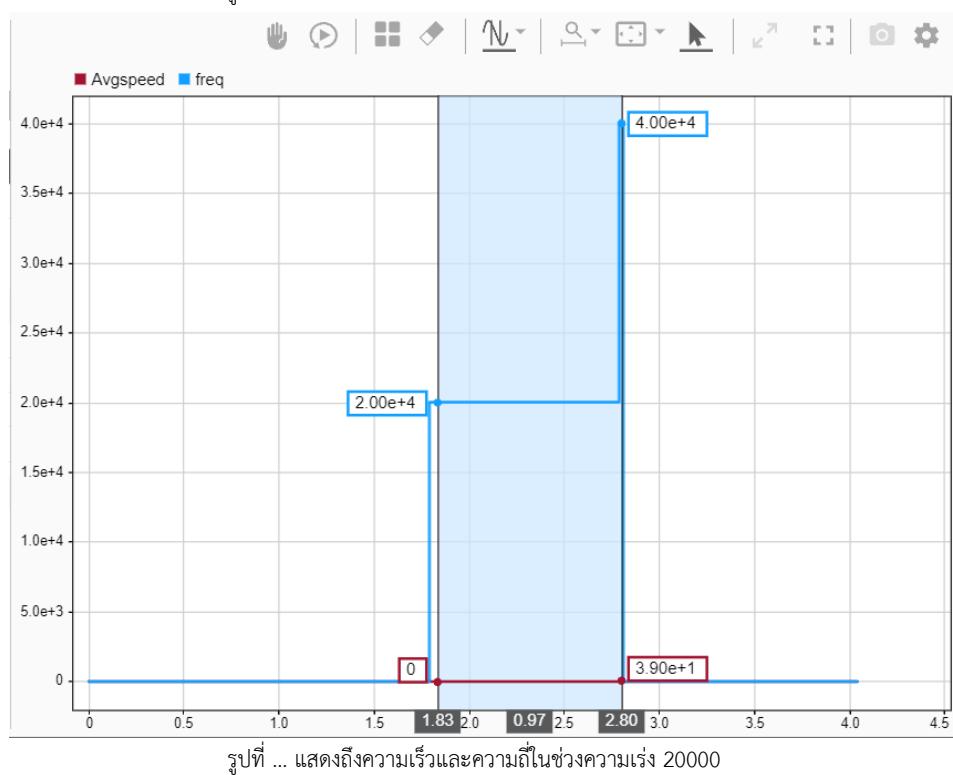
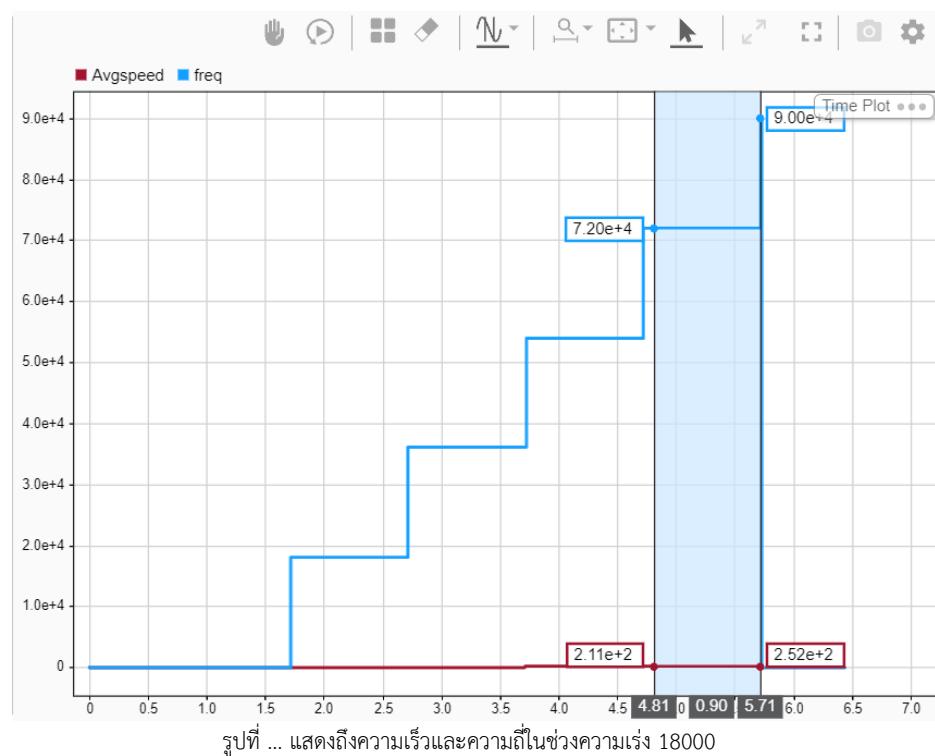
จากรูปแบบ 1/8 Step mode จะเริ่มไม่เห็นความแตกต่างของการเปรียบเทียบค่าแบบละเอียด เมื่อเปรียบเทียบกันแบบความแตกต่าง 1000 Hz แล้ว และไม่ Stepper Motor ไม่สามารถหมุนได้เมื่อมีความถี่เริ่มต้นที่ 1000 อีกต่อไป

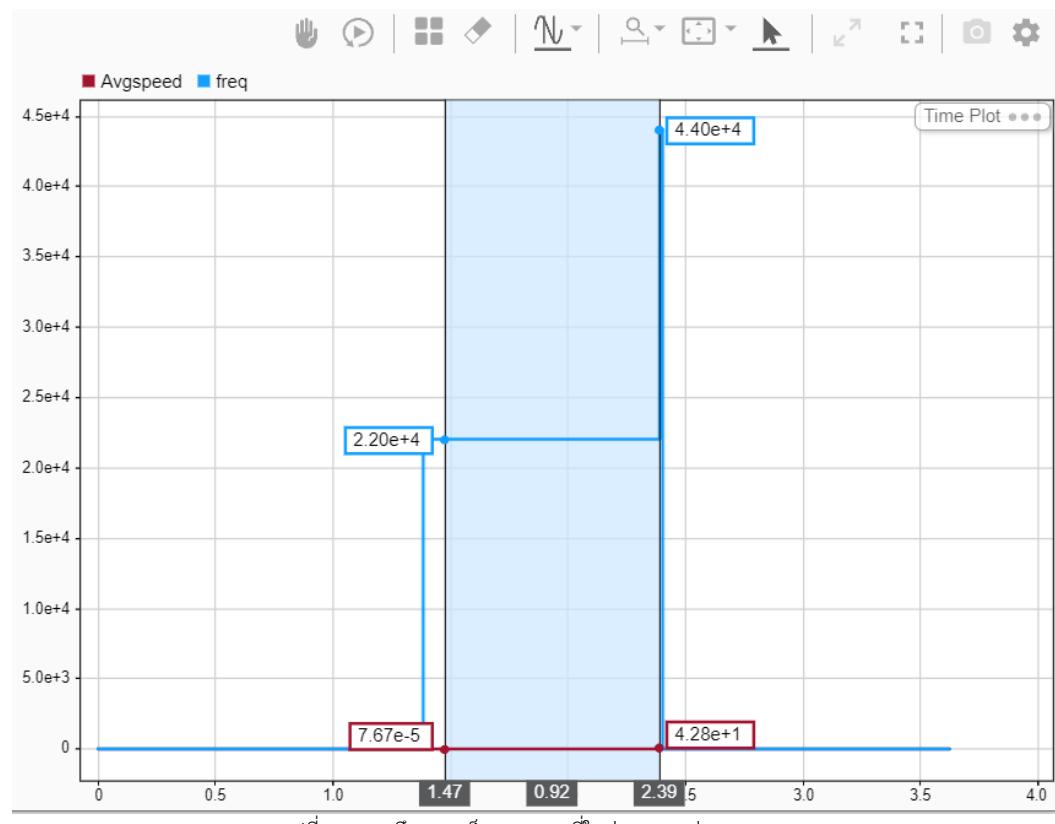




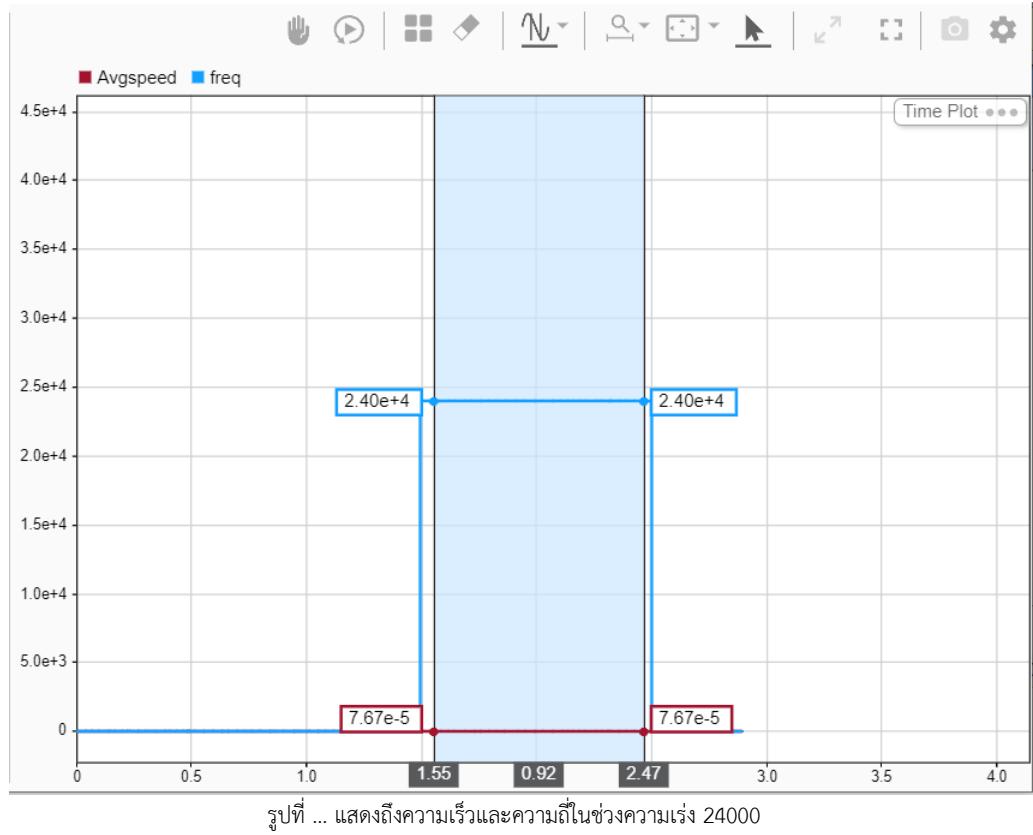




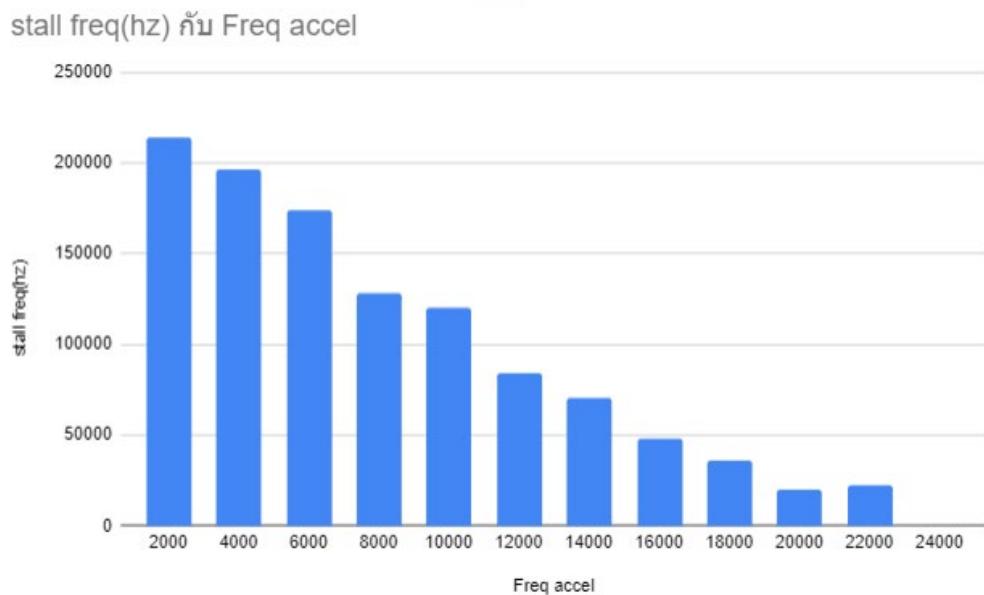




รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 22000

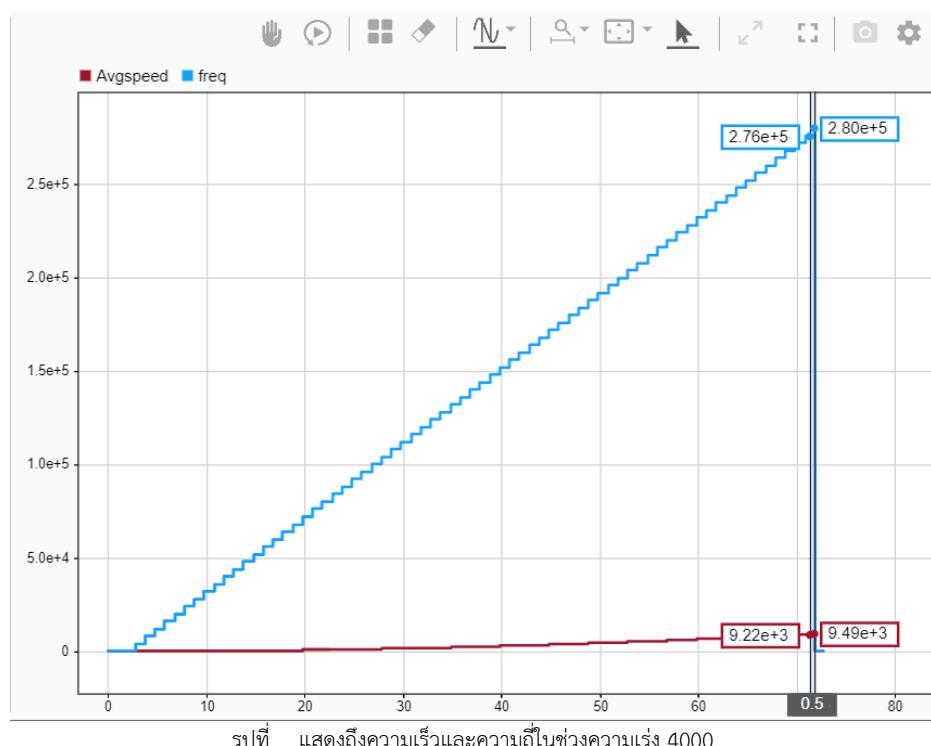


รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 24000

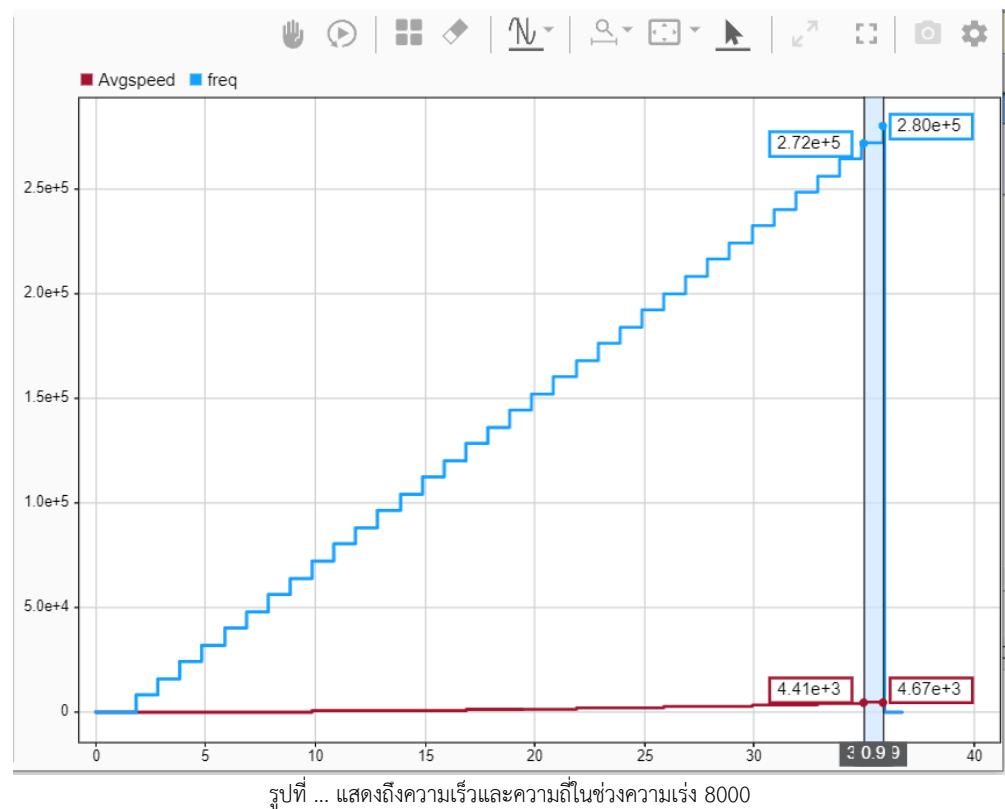


รูปที่ ... แสดงถึงความถี่สูงสุดและความเร่งในช่วงความเร่งต่างๆ ใน 1/16 Step mode

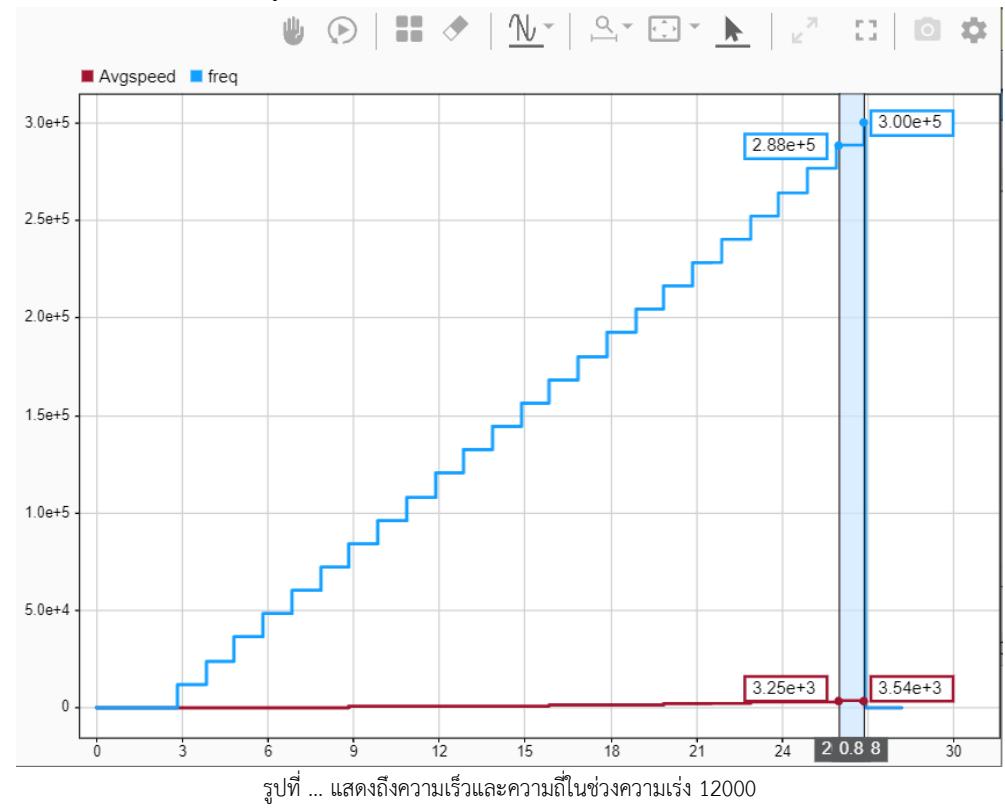
- การเก็บข้อมูลค่าความถี่ที่ได้ต่อความเร็วที่ทำได้มีอิทธิพลต่อความเร่งความถี่ในรูปแบบต่างๆ เมื่ออุปกรณ์ในการทำงานแบบ 1/32 Step mode โดยเพิ่มค่าความเร่งความถี่ที่ลาก 4000 เพราะเมื่อตรวจสอบจากรูปแบบ 1/16 Step mode จะเริ่มไม่เห็นความแตกต่างของการเปรียบเทียบค่าแบบละเอียดเมื่อเปรียบเทียบกันแบบความแตกต่าง 2000 Hz แล้ว และไม่ Stepper Motor ไม่สามารถหมุนได้เมื่อมีความถี่เริ่มต้นที่ 2000 อิกต่อไป



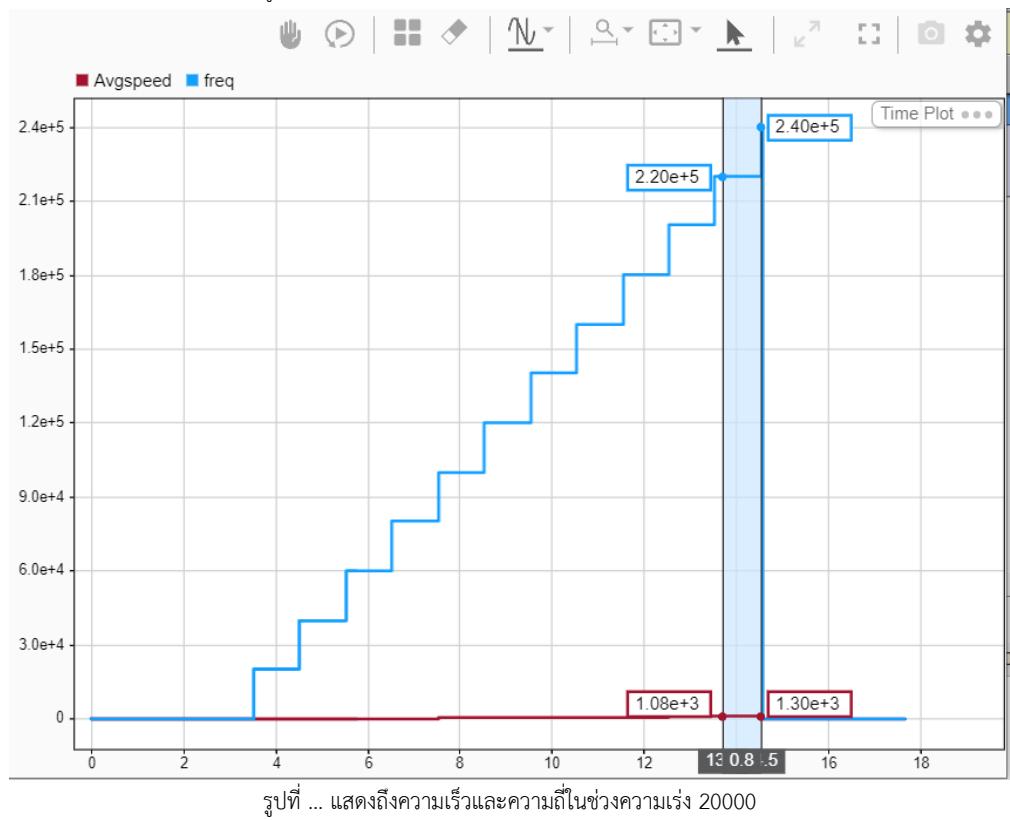
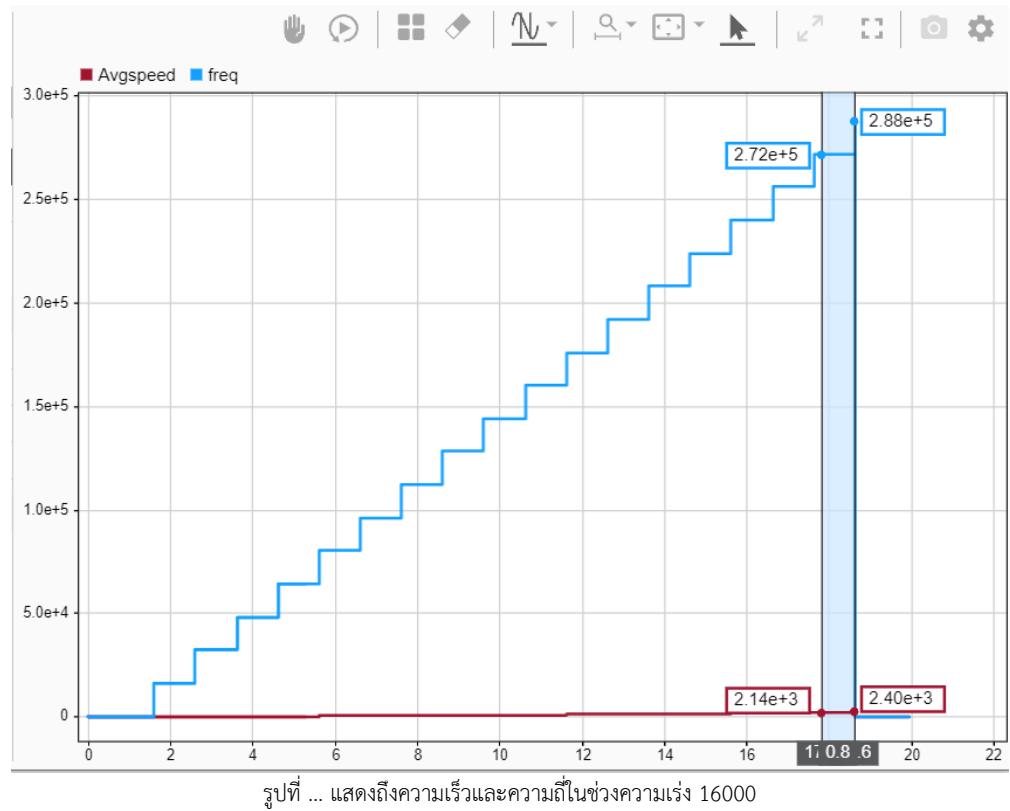
รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 4000

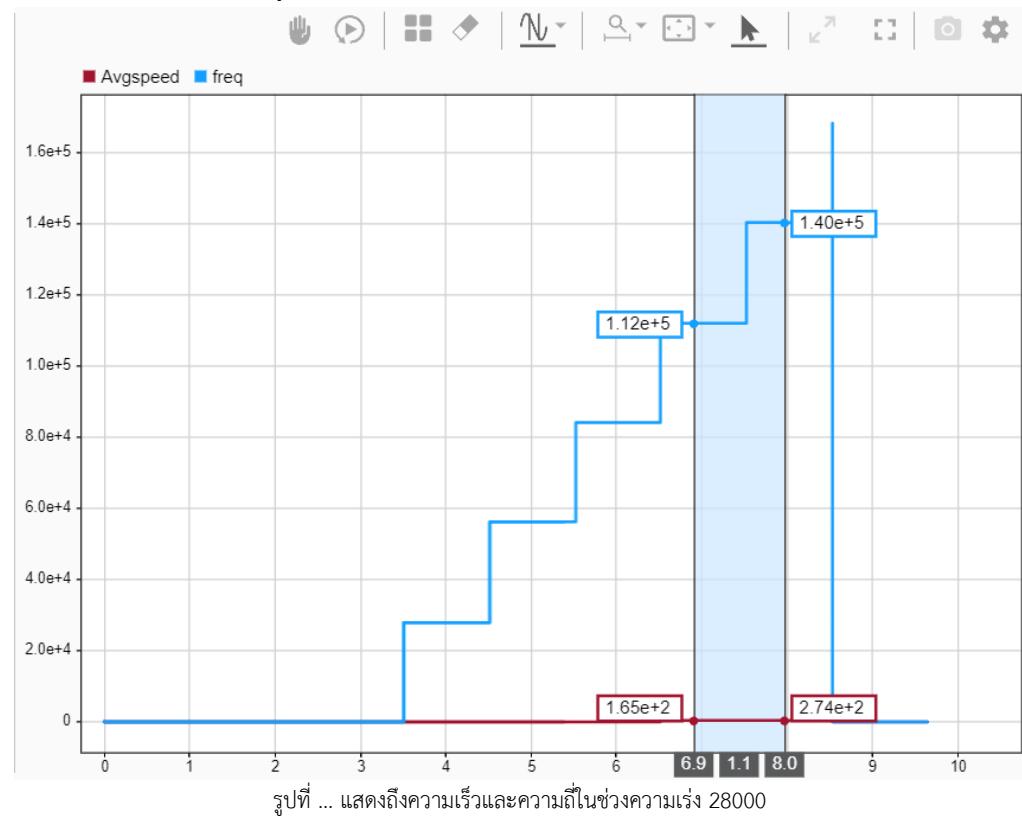
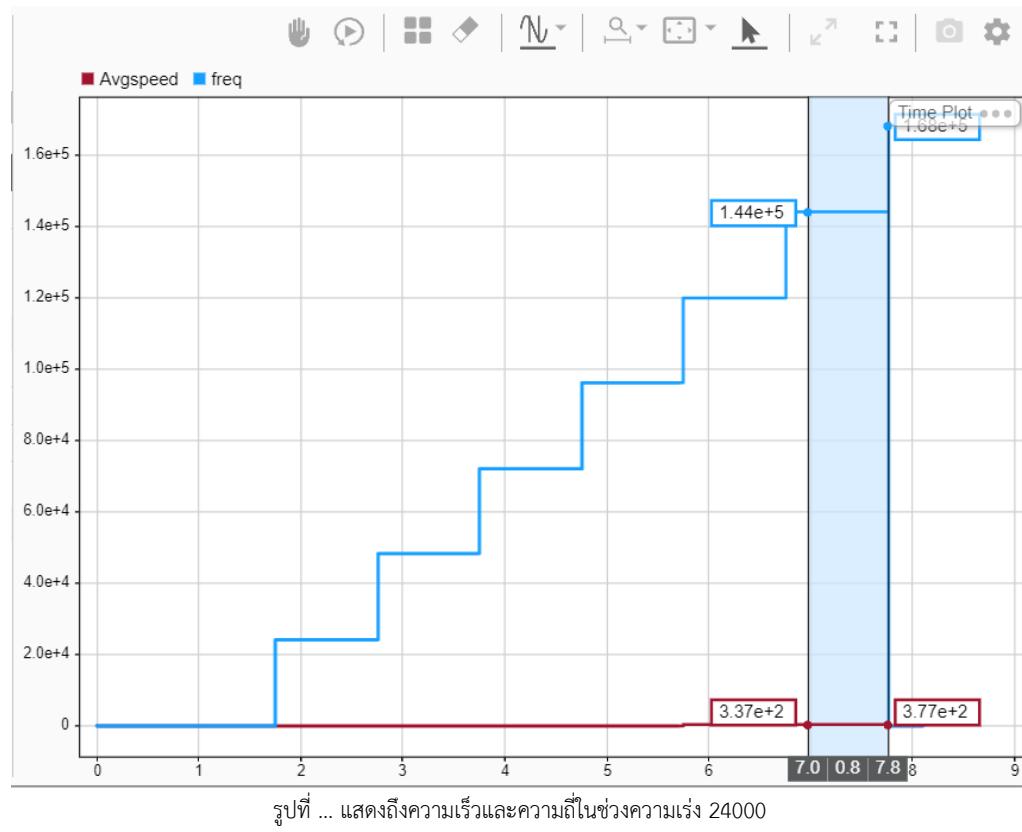


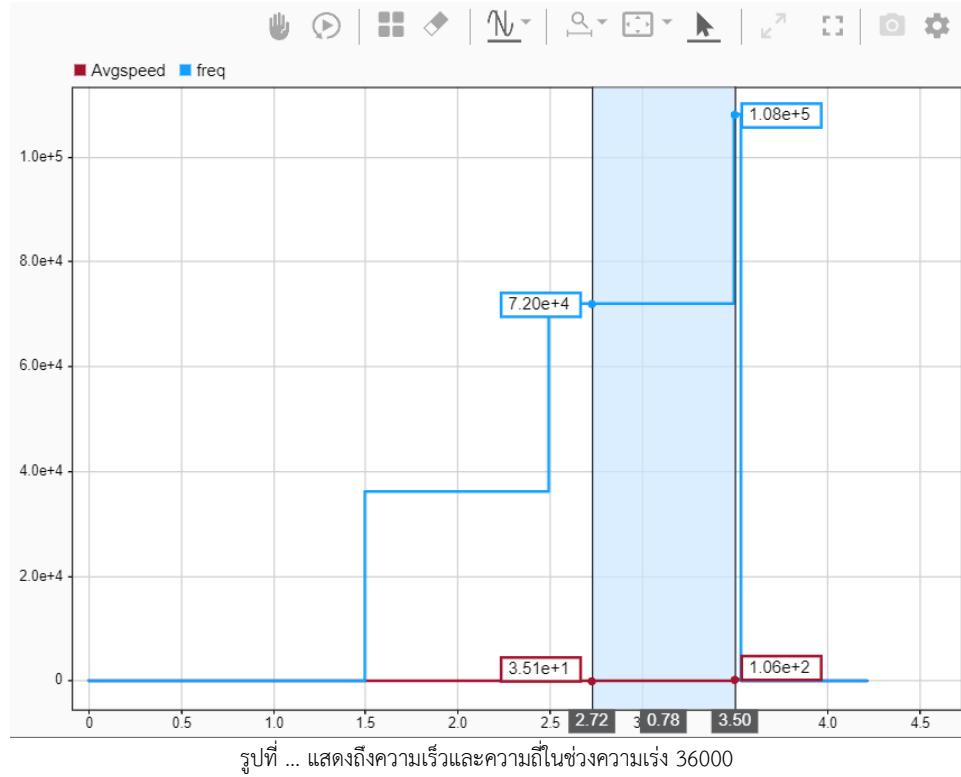
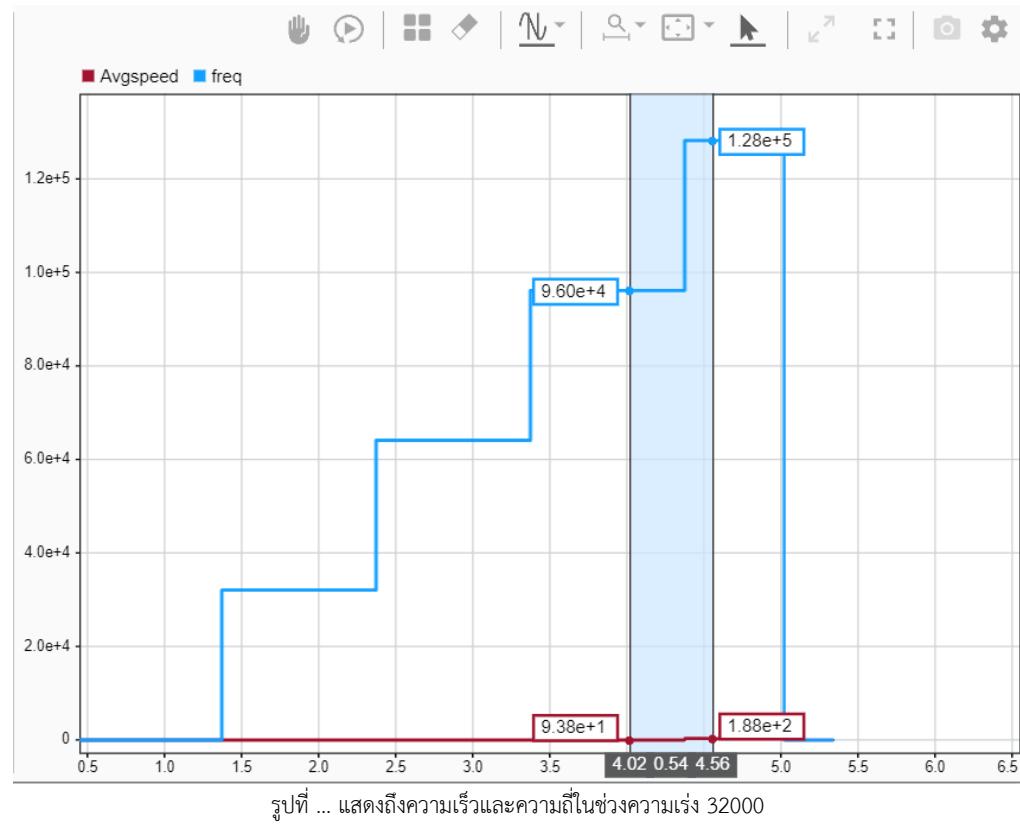
รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 8000

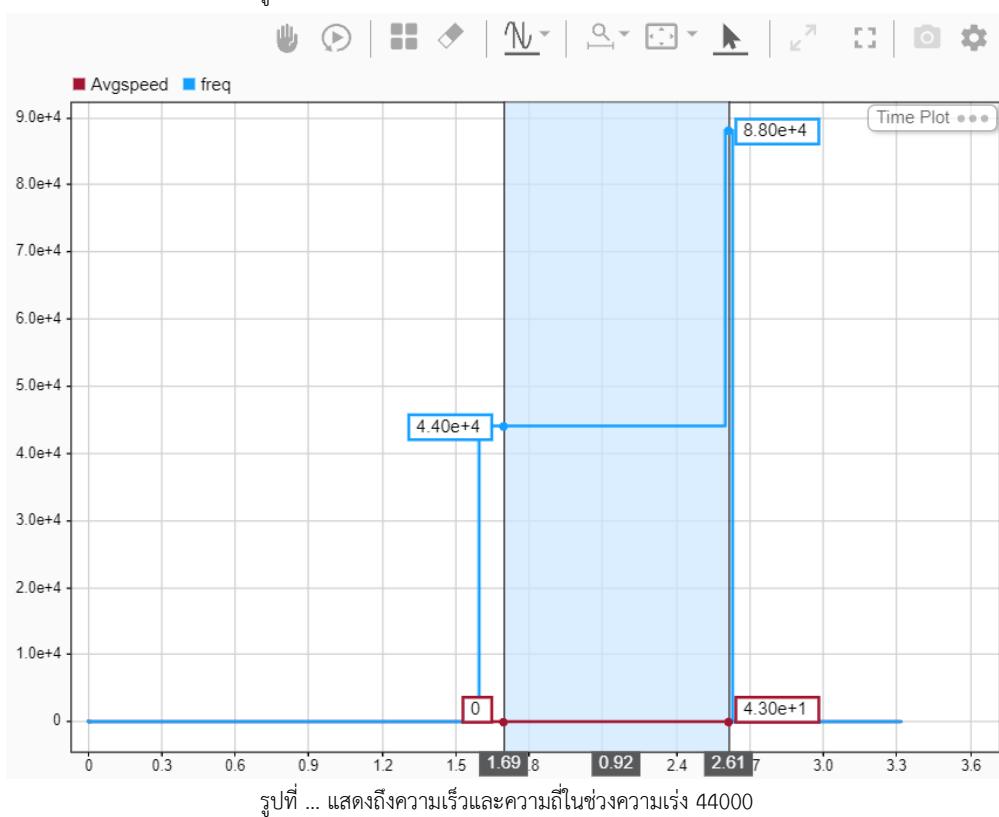
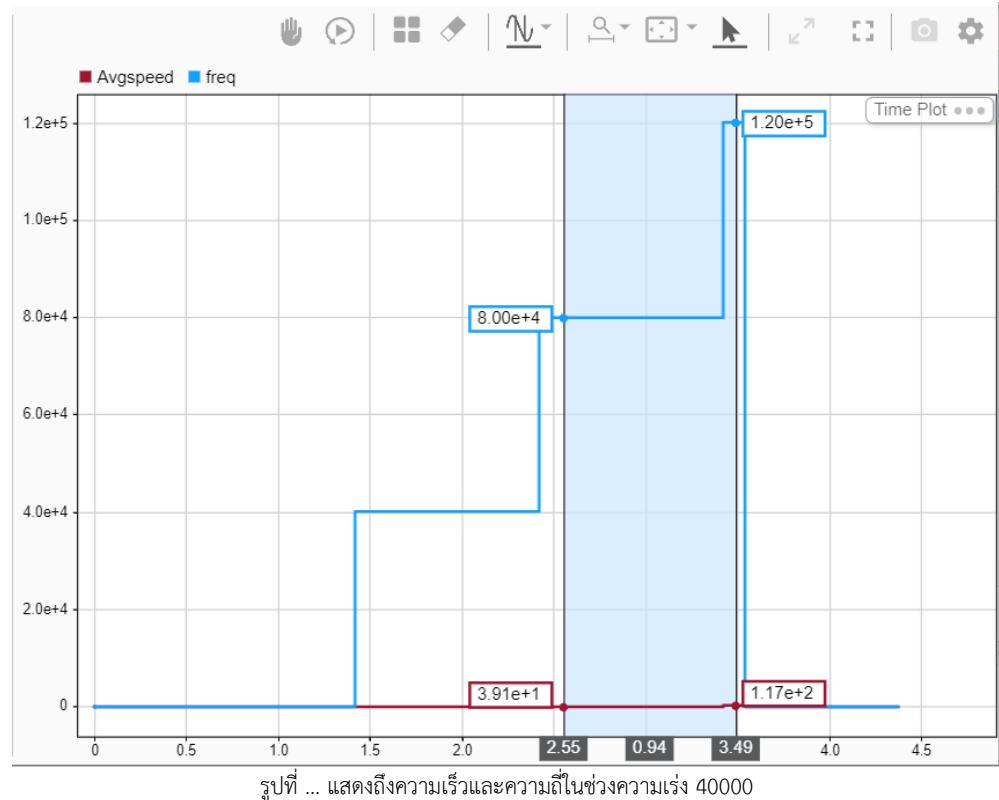


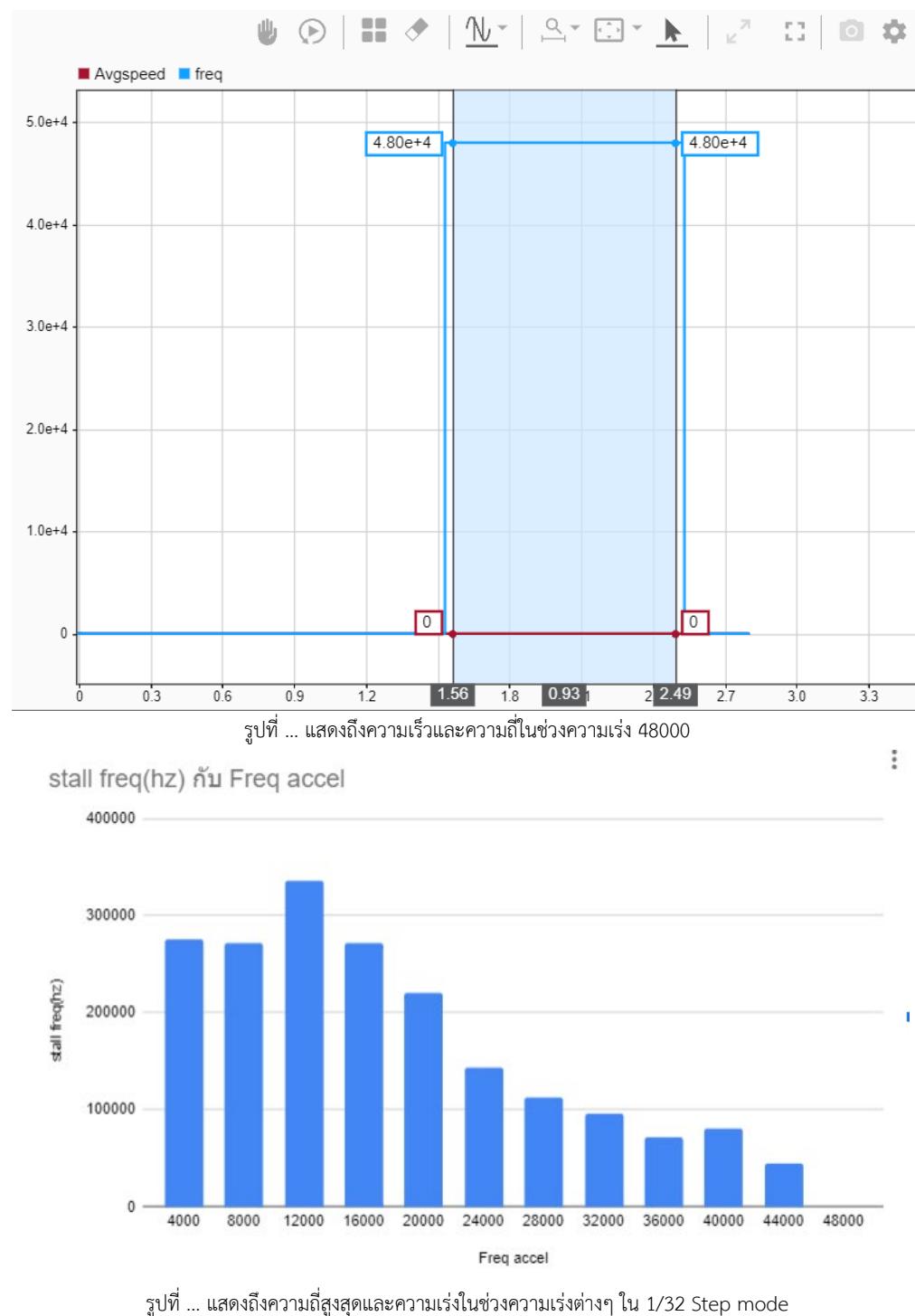
รูปที่ ... แสดงถึงความเร็วและความถี่ในช่วงความเร่ง 12000











รูปที่ ... แสดงถึงความถี่สูงสุดและความเร่งในช่วงความเร่งต่างๆ ใน 1/32 Step mode

2.11 สรุปผลการทดลอง

การวิเคราะห์ความสัมพันธ์ระหว่างความเร่งที่กำหนด เปรียบเทียบกับความถี่สูงสุดที่ทำได้ พบรเห็น ความสัมพันธ์ที่เกิดขึ้น โดยจากการเปรียบเทียบพบว่า ช่วงความเร่งความถี่ที่สูงขึ้นนั้น อาจทำให้ความเร็วروب ที่หมุนนั้นมีความถี่ที่สูงเกินไปจน Stepper Motor หมุนตามไม่ทัน จึงเกิดการหยุดหมุนขึ้นได้ ซึ่งจะมีช่วงของ

1/32 Step mode ที่ทำให้รู้ว่า ก่อนช่วงนั้นจะมีช่วงความถี่ที่น้อยเกินไปจนทำให้ Stepper Motor นั้นเคลื่อนที่พลาดได้เหมือนกัน ทำให้เกิดการ Step พลาดขึ้นได้จากการจ่ายความถี่ที่น้อยเกินไปใน Mode ที่ต้องการความถี่ขั้นต่ำสูง ซึ่งเริ่มพบความเปลี่ยนแปลงในช่วง 1/4 เป็นต้นไป

2.12 อภิปรายผลการทดลอง

Stepper Motor ที่นำมาใช้ในการทดลอง จะใช้การจ่ายกระแสไฟเป็นช่วงๆเพื่อทำการเลื่อน Step ไปทีละ Pulse เพื่อกำการขับเคลื่อนไปในแต่ละครั้ง โดยสิ่งที่วัดค่าได้มาสามารถสะท้อนถึงพฤติกรรมการเกิด Step loss ที่เกิดขึ้นใน Stepper Motor ซึ่งเกิดจากการที่ Stepper motor นั้นไม่สามารถหมุนความความถี่ที่จ่ายให้ทัน ทำให้ Stepper Motor เกิดการก้าวพลาดแล้วค้างอยู่ที่ตำแหน่งนั้นในที่สุด

2.13 ข้อเสนอแนะ

ในการทดลอง ให้พยายามเช็คความร้อนที่เกิดขึ้นกับตัวของ Stepper Motor หรือ STM32 อยู่เสมอ เพื่อความปลอดภัยในการใช้งาน และงดการทดลองช่วงคราว หากพบว่า Stepper Motor หรือ STM32 มีความร้อนสูงกว่าปกติ ซึ่งจากการตรวจสอบ ถึงแม้การทดลองจะทำการคำแนะนำอย่างครบถ้วน พบเจอบัญหาบอร์ด STM32 เกิดการไหม้ที่ชิปเกิดขึ้น เพราะได้ทำการทดลองอย่างต่อเนื่องโดยมิได้ตรวจสอบความร้อนที่เกิดขึ้นในบอร์ด STM32

2.14 เอกสารอ้างอิง

[1] <https://shorturl.at/RveU0>

[2] <https://shorturl.at/N8Umy>

[3] <https://shorturl.at/IISLa>