

การควบคุมมอเตอร์กระแสตรงแบบกระตุ้นแยก ด้วยไมโครคอนโทรลเลอร์

ตระกูล PIC

SEPARATE EXCITE DC MOTOR CONTROLLING BY PIC

MICROCONTROLLING

บัญชา ศรีวิโรจน์¹, วันชัย ทรัพย์สิงห์², จีระศักดิ์ วงศา³

Bancha Sreewirote¹, Wanchai Subsingha², Jeerasak Wongsas³

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏวชิรบุรี^{1,2}

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธนบุรี³

บทคัดย่อ

บทความนี้ได้นำเสนอการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นแยกให้มีประสิทธิภาพที่ดีขึ้น ที่มอเตอร์ขนาด 1,000 วัตต์ 220 โวลต์ ใช้ไมโครคอนโทรลเลอร์ 8 บิตตระกูล PIC เบอร์ PIC18F4431 เป็นตัวควบคุมการทำงานทั้งหมด การควบคุมมอเตอร์จะประกอบไปด้วยทฤษฎีควบคุมแบบลูบปิด PIcontrol ประมวลผลด้วยการเขียนโปรแกรมสร้างสมการให้สามารถควบคุมการสร้างสัญญาณ PWM (Pulse Width Modulation) ไปควบคุมมอเตอร์ และอาศัยการเชื่อมต่อผ่านพอร์ตอนุกรมที่เชื่อมต่อกับคอมพิวเตอร์เพื่อเก็บผลการทดสอบทั้งในส่วนของอาเมเจอร์โวลต์เตจคอนโทรล (V_a) และในส่วนของฟิลด์คอนโทรล (V_f) ด้วยโปรแกรม Hyper Terminal การทดสอบได้ใช้ ไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877A เป็นตัวกำหนดความเร็วรอบ (set point) ให้กับ PIC18F4431 เป็นตัวประมวลผลการทำงาน โดยมีการเปลี่ยนแปลงความเร็วรอบอยู่ตลอดเวลาเป็นช่วงๆไป ผลการทดสอบพบว่า ขณะมีการเปลี่ยนแปลงค่าความเร็วที่กำหนดไว้หรือมีการเปลี่ยนแปลงโหนดของมอเตอร์ ความเร็วจริงของมอเตอร์มีค่าใกล้เคียงความเร็วควบคุมที่กำหนดไว้ และเวลาเปลี่ยนแปลงความเร็วของมอเตอร์จริงที่เข้าใกล้ความเร็วที่กำหนดไว้ลดลงเฉลี่ยเหลือเพียง 0.85 วินาทีเท่านั้น

คำสำคัญ: มอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นแยก, ไมโครคอนโทรลเลอร์ตระกูล PIC, อาเมเจอร์โวลต์เตจคอนโทรล, ฟิลด์คอนโทรล

Abstract

This article describes the better performance of separately excited dc motor controlling for 1000 watts 220 volt by using 8-bit PIC18F4431 microcontroller controls all the functions. The motor controlling is consisting of the closed loop PI control theory evaluated the result by equation programming able to generate the PWM (Pulse Width Modulation) signal to control motor. The result collection both armature voltage control (V_a) and field control (V_f) has performed by Hyper Terminal programming connected to the computer via the serial port. The evaluation has been using the microcontroller PIC16F877A to be the set point rotating speed for PIC18F4431. There is the speed variation periodically. The evaluation result found the actual rotating speed of motor is approximated to the set point value during the speed changing of motor load and spending average time reduce to 0.85 second only.

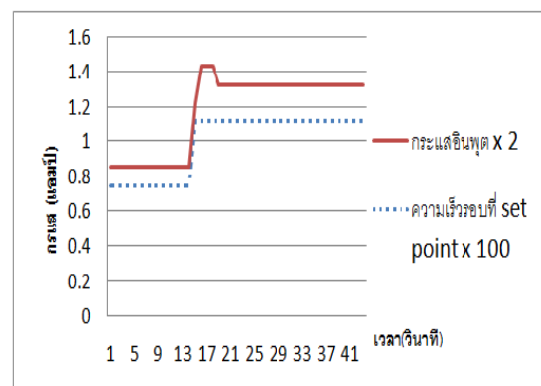
Keywords: separate excite dc motor, PIC microcontroller, amateur voltage control, field control

บทนำ

การควบคุมความมอดเตอร์ไฟฟ้ากระแสตรงแบบ กระตุ้นแยกส่วนใหญ่มีบทบาทสำคัญที่ใช้ในการ ควบคุมความเร็วในเครื่องจักรและกระบวนการผลิต อัตโนมัติในระบบอุตสาหกรรมนั้น จะควบคุมในส่วน ของแรงดันอาเมเจอร์ โดยการเพิ่มแรงดันอาร์เมเจอร์ หรือลดแรงดันอาร์เมเจอร์ หากเพิ่มแรงดันอาเมเจอร์ ความเร็วรอบก็จะสูงขึ้นและหากลดแรงดันความเร็ว รอบก็จะต่ำลง ในส่วนของแรงดันฟิลด์นั้นในงานส่วนใหญ่จะกำหนดให้คงที่ แต่ในงานวิจัยนี้จะแตกต่าง ออกไปคือมีการควบคุมแรงดันฟิลด์คือหากลดแรงดัน ฟิลด์ความเร็วรอบจะเพิ่มสูงขึ้นและหากเพิ่มแรงดัน ฟิลด์ความเร็วรอบก็จะลดต่ำลง ซึ่งขณะช่วงมอดเตอร์ กำลังเริ่มหมุนหรือเปลี่ยนแปลงความเร็วรอบนั้น จะเกิดการกระชากทำให้กระแสไฟฟ้าพุ่งสูงขึ้น จึงเกิด การสูญเสียทางกำลังไฟฟ้า แสดงดังภาพที่ 1 อีกทั้ง ความเร็วรอบในขณะที่เพิ่มโหลดให้กับมอดเตอร์นั้นจะมี การเปลี่ยนแปลงเป็นระยะเวลานาน กว่าที่จะเข้าใกล้ ความเร็วรอบที่กำหนดไว้จริง หรือตอบสนองช้านั้นเอง

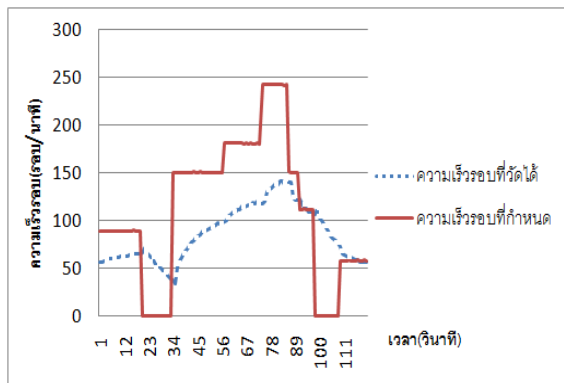
งานวิจัยนี้จึงได้นำไมโครคอนโทรลเลอร์ตระกูล PIC เข้ามาควบคุมในส่วนของวงจรควบคุมทั้งหมด เนื่องจาก ไมโครคอนโทรลเลอร์มีขนาดเล็กกะทัดรัด และยังมีฟังก์ชันการใช้งานที่หลากหลาย ตัวอย่างที่ใช้ ในงานวิจัยนี้ได้แก่ สามารถรับค่าความเร็วรอบของ

มอดเตอร์ได้ สามารถรับค่าของกระแสโดยการแปลง สัญญาณจากอนาล็อกเป็นดิจิตอล (Analog to Digital Converter, ADC) โดยที่ไม่จำเป็นต้องซื้อโมดูลแปลง สัญญาณมาใช้ จึงทำให้สามารถประหยัดค่าใช้จ่าย โดยสามารถสร้างสัญญาณ pulse with modulate (PWM) ที่จะสร้างสัญญาณขับเกตให้กับมอสเฟต เพื่อที่จะสับสวิตซ์ในการขับมอดเตอร์ อีกทั้งยังสามารถ สร้างสมการในการควบคุม PI ให้กับมอดเตอร์ได้จึง สามารถควบคุมแบบย้อนกลับได้(Feedback Control) ดังนั้นไมโครคอนโทรลเลอร์ตระกูล PIC จึงมีความ เหมาะสมสำหรับการควบคุมในงานวิจัยนี้ ซึ่งเป็นการ พัฒนาระบบการควบคุมให้มีขนาดเล็กกะทัดรัดได้อีก ด้วย



ภาพที่ 1 กระแสช่วงที่ เปลี่ยนแปลงความเร็วรอบของ มอดเตอร์ที่กำหนดไว้ (set point, SV)





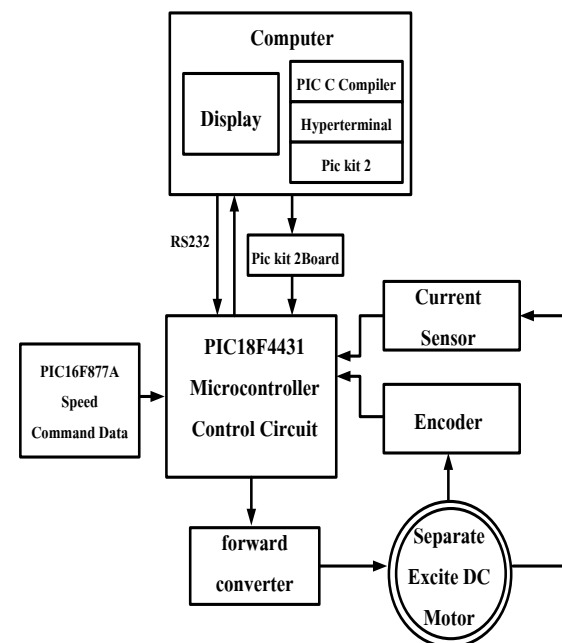
ภาพที่ 2 ความเร็วรอบในช่วงที่มีการเปลี่ยนแปลงแรงดันไฟฟ้า ของมอเตอร์ที่กำหนดไว้ (set point, SV)

การออกแบบ

ในส่วนของงานวิจัยนี้ได้ใช้การควบคุมแบบลูปปิดคือการรักษาความเร็วรอบให้คงที่ตามค่าความเร็วรอบที่กำหนดไว้ ซึ่งต้องอาศัยตัววัดความเร็วรอบ (Encoder) และเซนเซอร์กระแส (current sensor) เพื่อนำค่าที่ได้มาคำนวณหาค่าในทฤษฎี PI Control และสร้างสัญญาณ PWM เพื่อนำไปขับขาคงของมอเตอร์ที่เป็นตัวสับสวิตซ์ให้กับมอเตอร์การควบคุมนี้จะอาศัยการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์และคอมพิวเตอร์ โดยสื่อสารผ่านทางพอร์ตอนุกรม ซึ่งโปรแกรมที่ใช้เชื่อมต่อเพื่อรับค่าคือโปรแกรม Hyper Terminal และคอมไพล์ด้วยโปรแกรม PIC kit 2 ซึ่งใช้คู่กันกับบอร์ด PIC kit 2 ในส่วนของการเขียนโปรแกรมภาษาซีเพื่อควบคุมและสั่งงานไมโครคอนโทรลเลอร์นั้น จะใช้โปรแกรม PIC C Compiler โดยใช้ไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC18F4431 ซึ่งเหตุผลที่เลือกใช้เบอร์นี้เนื่องจากสามารถรับค่ากระแสได้โดยตรงที่รับมาจากเซนเซอร์กระแส เนื่องจากมีโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอลภายในตัว ทำให้ประหยัดค่าใช้จ่าย และยังมีฟังก์ชันในส่วนของการสร้างสัญญาณ PWM มากเพียงพอต่อการใช้งาน อีกทั้งการรับค่าความเร็วรอบจาก encoder สามารถที่จะต่อเข้าใช้งานได้โดยตรงซึ่งไม่ต้องต่อวงจรใดๆเพิ่มเติมเนื่องจากในตัวของ PIC18F4431 มีโมดูลที่รองรับการอ่านค่าของ encoder อยู่แล้ว จึงทำให้ประหยัดในส่วน

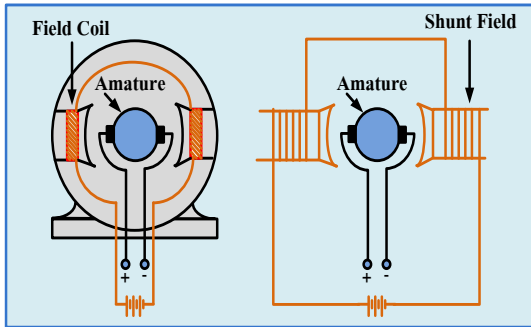
ของการสร้างวงจรแปลงความถี่ที่ได้รับจาก encoder ให้เป็นแรงดันไฟฟ้ากระแสตรง (Frequency convert to voltage) 0-5 โวลต์ ที่ป้อนเป็นสัญญาณอินพุตให้กับไมโครคอนโทรลเลอร์ จึงเป็นการลดขนาดของวงจรให้มีขนาดเล็กลงได้อีกด้วย โดยมีบล็อกไดอะแกรมระบบการควบคุมดังแสดงในภาพที่ 3 การควบคุมความเร็วรอบของมอเตอร์แบบลูปปิดจะถูกควบคุมด้วย PIC18F4431 เพียงตัวเดียวเท่านั้นโดยรับคำสั่งเป็นข้อมูลจาก PIC16F877A

เป็นตัวกำหนดความเร็วรอบของมอเตอร์เมื่อไมโครคอนโทรลเลอร์ PIC18F4431 รับสัญญาณอินพุตเข้ามา ไมโครคอนโทรลเลอร์จะทำการรับค่าและส่งค่าเข้ามาแสดงบนคอมพิวเตอร์ในหน้าต่างของโปรแกรม Hyper Terminal ในส่วนของการควบคุมแบบลูปปิดนั้น ไมโครคอนโทรลเลอร์จะเป็นตัวประมวลผลเปรียบเทียบข้อมูลเอง โดยรับคำสั่งภาษาซีที่ถูกส่งไว้ และทำการประมวลผลวนลูปอย่างต่อเนื่องจนกว่าค่าที่ควบคุม (set point, SV) จะเท่ากับค่าที่วัดได้ (Process value, PV)



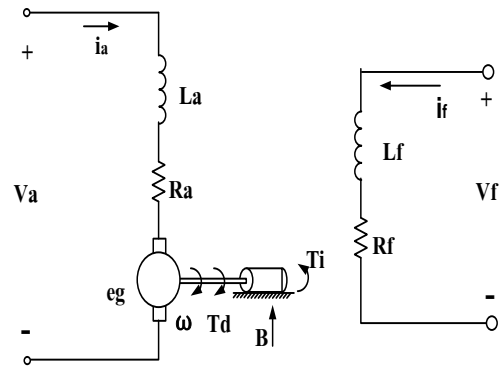
ภาพที่ 3 บล็อกไดอะแกรมแสดงระบบควบคุมความเร็วแบบลูปปิดด้วยไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC18F4431

1. มอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นแยก



ภาพที่ 4 ลักษณะโครงสร้างของมอเตอร์กระแสตรงแบบกระตุ้นแยก

มอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นแยก (separate excite dc motor) เป็นมอเตอร์ที่ใช้แรงดันไฟฟ้ากระแสตรง ประกอบไปด้วยขดลวดที่สำคัญอยู่ 2 ขด คือ ขดลวดอาเมเจอร์และขดลวดฟิลด์โดยพื้นฐานของดีซีมอเตอร์แบบ Separately Excited จะคล้ายกันกับ Self - excited ลักษณะโครงสร้างหลักจะประกอบด้วยส่วนที่อยู่กับที่ (Stator) และส่วนที่หมุนเคลื่อนที่ (Rotor) หรือหากพิจารณาในรูปของวงจรสมมูลทางไฟฟ้าก็สามารถแยกออกเป็น 2 วงจร คือวงจรฟิลด์ (Field Circuit) ซึ่งทำหน้าที่ในการสร้างสนามแม่เหล็กหลัก และ วงจรอาร์เมเจอร์ (Armature circuit) ที่ทำหน้าที่สร้างสนามแม่เหล็กรอบๆ อาร์เมเจอร์ แหล่งจ่ายกระแสไฟฟ้าของมอเตอร์ที่ต่อเข้ากับวงจรฟิลด์ และวงจรอาร์เมเจอร์ จะแยกเป็นอิสระซึ่งกันและกัน การควบคุมความเร็วจะควบคุมในส่วนของแรงดันอาเมเจอร์ โดยการเพิ่มแรงดันอาร์เมเจอร์หรือลดแรงดันอาร์เมเจอร์ หากเพิ่มแรงดันอาเมเจอร์ความเร็วรอบก็จะสูงขึ้นและหากลดแรงดันความเร็วรอบก็จะต่ำลง ในส่วนของแรงดันฟิลด์นั้นในงานส่วนใหญ่จะกำหนดให้คงที่ แต่ในงานวิจัยนี้จะแตกต่างออกไปคือมีการควบคุมแรงดันฟิลด์คือหากลดแรงดันฟิลด์ความเร็วรอบจะเพิ่มสูงขึ้นและหากเพิ่มแรงดันฟิลด์ความเร็วรอบก็จะลดต่ำลง



ภาพที่ 5 วงจรสมมูลของมอเตอร์กระแสตรงแบบกระตุ้นแยก

1.1 ค่าพารามิเตอร์ของมอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นแยก

จากภาพที่ 5 เมื่อพิจารณาการหาค่าพารามิเตอร์ ตามกฎของเคอร์ชอฟฟ์สามารถเขียนเป็นสมการได้ดังนี้ [1]

$$V_a = i_a R_a + L_a \frac{di_a}{dt} + e_g \quad (1)$$

$$\text{ซึ่ง } e_g = K_v i_f \omega \quad (2)$$

$$\text{จะได้ } V_a = i_a R_a + L_a \frac{di_a}{dt} + K_v i_f \omega \quad (3)$$

เมื่อกำหนดให้ K_v เป็นแรงดันคงที่ (V.s/rad).

ทำการแปลงลาปลาซในสมการที่ 2 จะได้

$$I_a(s) = I_a(s)R_a + L_a s I_a(s) + K_v I_f \omega(s) \quad (4)$$

จากสมการที่ 3 สามารถเขียนใหม่ได้เป็น

$$I_a(s) = \frac{V_a(s) - K_v I_f \omega(s)}{sL_a + R_a} \quad (5)$$

จากภาพที่ 4 เมื่อพิจารณาทางกลจะได้

$$T_d = J \frac{d\omega}{dt} + B\omega + T_L \quad (6)$$

ทำการแปลงลาปลาซในสมการที่ 5 และกำหนดให้

$$T_d(s) = K_t I_f I_a(s) \text{ และ } T_L(s) = 0 \text{ จะได้}$$

$$\omega(s) = \frac{K_t I_f I_a(s)}{sJ + B} \quad (7)$$

เมื่อ K_t เป็นค่าคงที่ของแรงบิด (N.m/A)

แทนค่า $I_a(s)$ ในสมการที่ 4 และ 6 ตามลำดับทรานเฟอร์ฟังก์ชันระหว่าง armature voltage (V_a) และ ความเร็วมอเตอร์ (V_f) จะได้เป็น



$$\frac{\omega(s)}{V_a(s)} = \frac{K_f I_f / R_a R_f B}{\left[\frac{L_a J}{R_a B} \right] s^2 + \left[\frac{L_a + J}{R_a + B} \right] s + \left[1 + \frac{(K_v V_f)^2}{R_a R_f^2 B} \right]} \quad (8)$$

กำหนดให้ $K_v = K_f$ จะได้

$$\frac{\omega(s)}{V_a(s)} = \frac{K I_f / R_a R_f B}{\left[\frac{L_a J}{R_a B} \right] s^2 + \left[\frac{L_a + J}{R_a + B} \right] s + \left[1 + \frac{(K V_f)^2}{R_a R_f^2 B} \right]} \quad (9)$$

การหาทรานเฟอ์ฟังก์ชันด้วยวิธีการ system Identify เป็นวิธีการเก็บผลการทดสอบ โดยวิธีการเริ่มการหมุนมอเตอร์เพื่อที่จะเก็บค่าของกระแส แรงดันและความเร็วรอบ หลายหมื่นค่า แล้วนำค่าที่ได้นั้น มาทำการหาทรานเฟอ์ฟังก์ชันในโปรแกรม Matlab ซึ่งได้ทรานเฟอ์ฟังก์ชันของมอเตอร์กระแสตรงได้ดังนี้

$$T_f = \frac{K}{1 + T_{p1}s} \quad (10)$$

ลูป Amature

$$K = \frac{2.7036}{1 + 1.0079s} \quad (11)$$

ลูป speed

$$K = \frac{0.020006}{1 + 0.51274s} \quad (12)$$

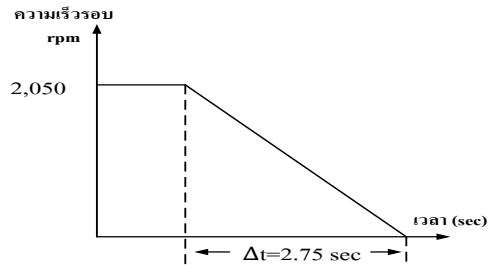
จากสมการข้างต้นเมื่อนำมาหาค่าพารามิเตอร์จะได้ดังตารางที่ 1

ตารางที่ 1 ค่าพารามิเตอร์ของมอเตอร์

ค่าพารามิเตอร์	
แรงดันไฟฟ้า	$V_a = 220 \text{ V}$
กระแสอาร์เมเจอร์	$I_a = 5.6 \text{ A}$
ความต้านทานอาร์เมเจอร์	$R_a = 2.2 \ \Omega$
ความต้านทานฟิลต์	$R_f = 400 \ \Omega$
ค่าความเหนี่ยวนำอาร์เมเจอร์	$L_a = 0.015 \text{ H}$
ค่าความเหนี่ยวนำฟิลต์	$L_f = 7 \text{ H}$
Mechanical time constant	$T_m = 2.75 \text{ sec}$
Moment of inertia	$J = 2.92 \times 10^{-3} \text{ kg} \cdot \text{m}^2$
Viscous friction coefficient	$B = 37.77 \times 10^{-3} \text{ N} \cdot \text{m} \cdot \text{s/rad}$

โดยที่ Mechanical time constant (T_m) คือค่าเวลาที่มอเตอร์ใช้ในการหยุดหมุนซึ่งขึ้นอยู่กับรัศมีและ

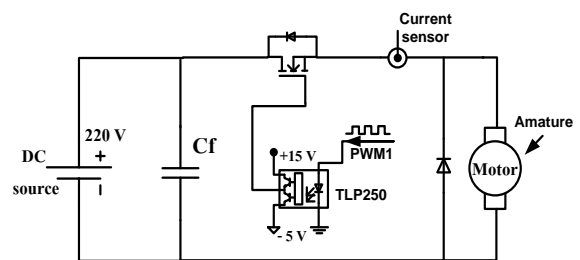
น้ำหนักตัวมอเตอร์ พิจารณาดังภาพที่ 6 ซึ่งมีค่าเท่ากับ 2.75 วินาที



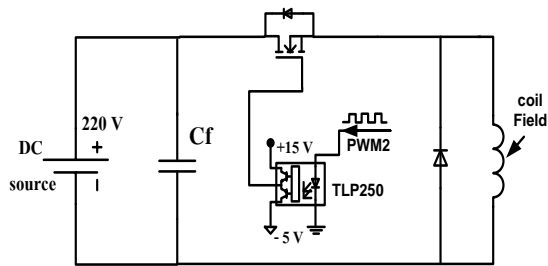
ภาพที่ 6 mechanical time constant

2. วงจรกำลังภาคขับเคลื่อนมอเตอร์ด้วยมอสเฟต

วงจรมีเป็นวงจรที่ทำหน้าที่สับสวิตช์มอสเฟตที่มีความถี่ที่เหมาะสมที่ 30 kHz โดยรับสัญญาณ PWM 1 และ PWM 2 มาจาก OPTO เบอ์ TLP250 ซึ่งทำหน้าที่ขยายสัญญาณ PWM ที่มาจากไมโครคอนโทรลเลอร์ PIC18F4431 ให้มีขนาดแรงดันที่เหมาะสมสำหรับมอสเฟต ซึ่งในที่นี้ได้เลือกใช้มอสเฟตเบอ์ IRFP460LC สามารถทนแรงดันได้ที่ 500 โวลต์และรับกระแสได้ถึง 20 แอมป์ ทั้งนี้ก็ขึ้นอยู่กับอุณหภูมิของตัวมอสเฟต แรงดันระหว่างขา drain และ source และแรงดันที่ใช้ขับเคลื่อนของมอสเฟต ซึ่งต้องใช้แรงดันในการขับเคลื่อนที่เหมาะสมคือ 15 โวลต์และ -5 โวลต์ อีกทั้ง OPTO TLP250 ยังเป็นวงจรที่ช่วยแยกกราวด์ระหว่างวงจรกำลังภาคขับเคลื่อนกับวงจรไมโครคอนโทรลเลอร์เมื่อเกิดการลัดวงจรในส่วนภาคขับเคลื่อนมอสเฟตความเสียหายที่เกิดขึ้นจะไม่ส่งผลกระทบต่อภาไมโครคอนโทรลเลอร์



ภาพที่ 7 วงจรกำลังภาคขับเคลื่อนแรงดันอาร์เมเจอร์ (V_a)



ภาพที่ 8 วงจรกำลังภาคขับแรงดันขดลวดฟิลด์ (Vf)

3. วงจรควบคุมไมโครคอนโทรลเลอร์ตระกูล PIC

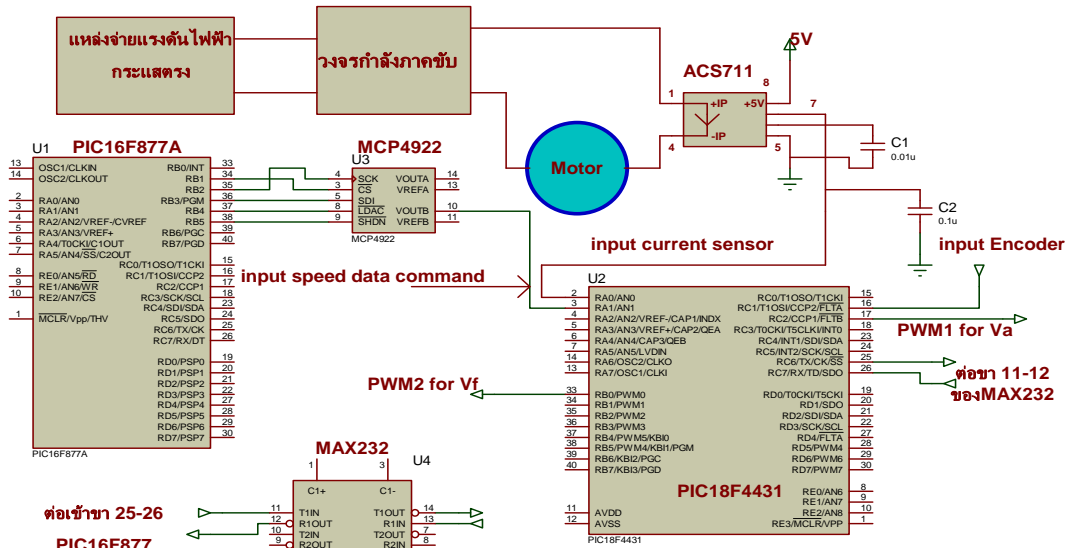
ในส่วนของ การเขียนโปรแกรมภาษาซี ใช้โปรแกรม PIC C Compiler ในการเขียนโปรแกรม ซึ่งมีโมดูลสร้างสัญญาณ PWM ไว้ให้ 8 ชุด และในงานวิจัยนี้ใช้เพียง 2 ชุดเท่านั้นใช้รีจิสเตอร์ขนาด 16 บิต เพื่อควบคุมแรงดันอาร์เมเจอร์และแรงดันฟิลด์ การคอมไพล์ใช้โปรแกรม Pickit 2 คู่กับบอร์ด Pickit 2 โดยการดิงไฟล์นามสกุล .hex ซึ่งโปรแกรม CCS PIC 'C' Compiler สามารถรองรับโปรแกรม Pickit 2 จึงทำให้ง่ายต่อการคอมไพล์ ซึ่งมีฟังก์ชันการคอมไพล์เฮกไฟล์แบบอัตโนมัติให้ใช้ จึงสามารถที่จะใช้งานได้อย่างรวดเร็วมากขึ้นวงจรควบคุมนี้จะรับสัญญาณ pulse ที่มาจาก encoder ซึ่งยึดติดกับเพลลาของมอเตอร์รับค่ากระแสที่ไหลผ่านตัวมอเตอร์และรับค่าความเร็วรอบของมอเตอร์ที่กำหนดมาจาก PIC16F877A เข้ามาเป็นสัญญาณ ADC และจะเข้าสู่เงื่อนไข คือความเร็วที่พิกัดของมอเตอร์ กำหนดให้เท่ากับ 80 % ของความเร็วที่พิกัดจริงของมอเตอร์นั่นคือ 2,050 rpm

เนื่องจากการทดสอบจะมีการทดสอบที่สูงกว่าพิกัดของมอเตอร์ จึงต้องกำหนดให้มอเตอร์ทำงานที่พิกัดเท่ากับ 80% เพื่อเป็นการถนอมอุปกรณ์ซึ่งมีราคาสูง การเขียนโปรแกรมจะสอดคล้องกับการทดลอง นั่นคือมีการกำหนดให้

1. มอเตอร์ทำงานที่ความเร็วรอบต่ำกว่าความเร็วรอบที่พิกัดของมอเตอร์
2. มอเตอร์ทำงานที่ความเร็วรอบที่พิกัดของมอเตอร์
3. มอเตอร์ทำงานที่ความเร็วรอบสูงเกินกว่าพิกัดของมอเตอร์

โดยข้อมูลการสั่งงานให้ไมโครคอนโทรลเลอร์ทำงานที่ความเร็วต่างกันนั้น จะรับข้อมูลมาจากไมโครคอนโทรลเลอร์อีกตัวหนึ่งนั่นคือ PIC16F877A ส่งค่าเป็นสัญญาณดิจิตอลและเข้าสู่วงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อกแล้วจึงนำสัญญาณอนาล็อกนี้ต่อเข้ากับขาสัญญาณ ADC ของ PIC18F4431 นอกจากนี้แล้วการควบคุมความเร็วรอบนั้น จะอาศัยทฤษฎี PI control และยังมีการควบคุมกระแสที่ไหลผ่านมอเตอร์อีกด้วยในส่วนของ การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์และคอมพิวเตอร์นั้น อาศัยการสื่อสารผ่านพอร์ตอนุกรม RS-232 โดยต่อผ่าน MAX232 ที่เป็นอุปกรณ์ในการเชื่อมต่อ ซึ่งวงจรควบคุมนี้จะรับสัญญาณ pulse ที่มาจาก encoder ซึ่งยึดติดกับเพลลาของมอเตอร์รับค่ากระแสที่ไหลผ่านตัวมอเตอร์และรับค่าความเร็วรอบของมอเตอร์ที่กำหนดมาจาก PIC16F877A เข้ามาเป็นสัญญาณ ADC และจะเข้าสู่เงื่อนไข

วงจรควบคุมความเร็วมอเตอร์แบบกระตุ้นแยกด้วยไมโครคอนโทรลเลอร์ตระกูล PIC



ภาพที่ 9 วงจรควบคุมความเร็วมอเตอร์แบบกระตุ้นแยกด้วยไมโครคอนโทรลเลอร์ตระกูล PIC

ในส่วนของการรับค่าต่างๆ ได้แก่

1. สัญญาณความเร็ว ได้เลือกใช้ encoder ที่สร้างสัญญาณสี่เหลี่ยม 600 สัญญาณต่อรอบ ของบริษัท Shimpo รุ่น RE1-600-C มาใช้ โดยต่อสายสัญญาณเข้ากับขา CCP2 ของไมโครคอนโทรลเลอร์ PIC 18F4431 โดยการเขียนภาษาซีรับค่าความเร็วรอบ
2. ค่ากระแสที่ไหลผ่านมอเตอร์ใช้เซนเซอร์กระแสเบอร์ ACS711ELCTR-25AB ที่สามารถยอมให้กระแสไหลผ่านได้สูงสุดถึง 25 แอมป์ โดยสัญญาณนี้จะต่อเข้ากับ AN0 ของ PIC18F4431
3. สัญญาณข้อมูลความเร็วรอบที่รับมาจาก PIC16F877A เป็นตัวกำหนดความเร็วรอบให้ PIC 18F4431 นั่นจะเป็นสัญญาณดิจิตอลจึงทำการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกโดยใช้ไอซีแปลงสัญญาณเบอร์ MCP4922 เมื่อได้สัญญาณแล้วจึงนำไปต่อเข้ากับขา AN1 ของ PIC18F4431 ทั้ง 3 สัญญาณที่ต่อเข้า PIC18F4431 จะเป็นข้อมูลเพื่อนำไปคำนวณค่าในทฤษฎี PI Control ต่อไป

4. การออกแบบในส่วนของ Software

จะมีซอฟต์แวร์ที่เป็นการขับมอเตอร์แบบ open loop control และซอฟต์แวร์แบบ close loop control เพื่อเปรียบเทียบผลการทดลอง

1. open loop control ในการออกแบบซอฟต์แวร์นี้ เมื่อรับค่าความเร็วรอบที่ PIC16F877A กำหนดมา หลังจากนั้น PIC18F4431 ก็สร้างสัญญาณ PWM ให้สอดคล้องกับค่าความเร็วรอบ โดยการปรับ duty cycle นอกจากนี้จะสร้างสัญญาณ PWM แล้ว PIC18F4431 จะต้องส่งข้อมูลความเร็วรอบที่วัดได้ (Process value, Pv) และค่ากระแสที่วัดได้ (Current Value, Iv) ไปยังคอมพิวเตอร์อีกด้วย

2. close loop control ซอฟต์แวร์นี้จะเงื่อนไขทฤษฎี PI Control เพื่อที่จะให้ค่าความเร็วรอบที่วัดได้มีค่าเข้าใกล้ค่าเป้าหมายที่ต้องการ (set point, Sv) ในส่วน of ค่ากระแสให้เป็นไปตามค่าของกระแสที่ตั้งไว้ (Current set point ,Is) ซึ่งต้องกำหนดเงื่อนไขที่ถูกต้อง เพื่อที่จะสร้างสัญญาณ PWM ที่เหมาะสม (Manipulate value, Mv) ซึ่งมีเงื่อนไขดังต่อไปนี้

จากสมการ PI-Control [2]

$$M_v(t) = M_{v,ss} \pm K_c \left[e(t) + \frac{1}{T_i} \int e(t) dt \right] \quad (13)$$

กำหนดให้

เงื่อนไขที่ 1 ถ้าค่ากระแสที่วัดได้จริงน้อยกว่าค่ากระแสที่ได้จากการคำนวณ โปรแกรมจะเพิ่ม

ค่ากระแสให้สูงขึ้นโดยการเพิ่มค่า M_v คำนวณตามสมการ

$$err_1 = I_s - I_v \quad (14)$$

$$M_v(t) = M_{v_1} + gain[(err_1 - err_2) + err_1 / T_i] \quad (15)$$

เมื่อค่า $M_v \geq 254$ ให้ M_v เท่ากับ 254

ถ้าเงื่อนไขที่ 1 เป็นจริงแล้ว ไมโครคอนโทรลเลอร์จะทำการตรวจสอบความเร็วรอบที่รับมาจาก PIC16F877A (S_v) อีกครั้งในเงื่อนไขที่ 3 และ 4

เงื่อนไขที่ 2 ถ้าค่ากระแสที่วัดได้จริงมากกว่าค่ากระแสที่ได้จากการคำนวณ โปรแกรมจะลดค่ากระแสลงโดยการลดค่า M_v คำนวณตามสมการ

$$err_1 = I_v - I_s \quad (16)$$

$$M_v(t) = M_{v_1} - gain[(err_1 - err_2) + err_1 / T_i] \quad (17)$$

เมื่อค่า $M_v \leq 1$ ให้ M_v มีเท่ากับ 1

ถ้าเงื่อนไขที่ 2 เป็นจริงแล้ว ไมโครคอนโทรลเลอร์จะทำการตรวจสอบความเร็วรอบที่รับมาจาก PIC16F877A (S_v) อีกครั้งในเงื่อนไขที่ 3 และ 4 หากเงื่อนไขที่ 1 และ 2 ไม่เป็นจริงก็โปรแกรมก็จะข้ามมาตรวจสอบเงื่อนไขที่ 3 และ 4 เลย

เงื่อนไขที่ 3 หากค่าความเร็วรอบที่วัดได้จริงน้อยกว่าค่าความเร็วรอบที่รับมาจาก PIC16F877A โปรแกรมจะเพิ่มค่าความเร็วรอบให้สูงขึ้นโดยการเพิ่มค่า M_v คำนวณตามสมการ

$$err_1 = S_v - P_v \quad (18)$$

$$M_v(t) = M_{v_1} + gain[(err_1 - err_2) + err_1 / T_i] \quad (19)$$

เมื่อค่า $M_v \geq 254$ ให้ M_v เท่ากับ 254

เงื่อนไขที่ 4 หากค่าความเร็วรอบที่วัดได้จริงมากกว่าค่าความเร็วรอบที่รับมาจาก PIC16F877A โปรแกรมจะลดค่าความเร็วรอบให้ต่ำลงโดยการลดค่า M_v คำนวณตามสมการ

$$err_1 = P_v - S_v \quad (20)$$

$$M_v(t) = M_{v_1} - gain[(err_1 - err_2) + err_1 / T_i] \quad (21)$$

เมื่อค่า $M_v \leq 1$ ให้ M_v มีเท่ากับ 1

เงื่อนไขที่ 5 หาก $I_s = I_v$ แล้วตรวจสอบ

$P_v = S_v$ แล้วจะได้

$$M_v = I_s = I_v = P_v = S_v \quad (22)$$

5. การเขียนโปรแกรมภาษาซี

ในส่วนของโปรแกรม PIC C Compiler นั้น จะต้องมีการเขียนรับค่าสัญญาณต่าง ๆ ได้แก่ การเขียนโปรแกรมเชื่อมต่อกับพอร์ตอนุกรม RS232, การเขียนโปรแกรมรับค่าความเร็วรอบจาก encoder, การเขียนโปรแกรมรับค่าคำสั่งความเร็วรอบของมอเตอร์จาก PIC16F877A ,การเขียนโปรแกรมสร้างความเร็วรอบหลายๆค่าในตัว PIC16F877A ,การเขียนโปรแกรมรับค่ากระแสที่ไหลผ่านตัวมอเตอร์ และการสร้างสัญญาณ PWM ซึ่งโปรแกรมทั้งหมดจะมีความยาวมาก จึงได้ยกตัวอย่างโปรแกรมมาบางส่วนมีดังนี้

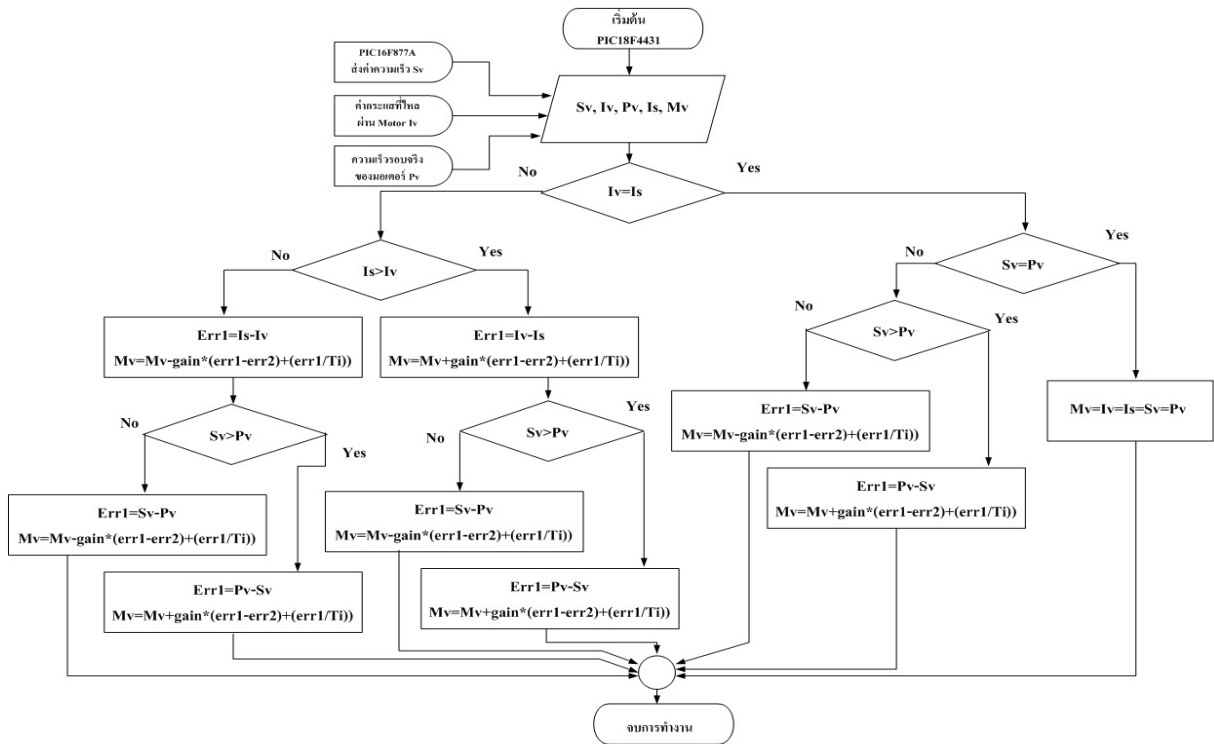
1. ตัวอย่างการเขียนโปรแกรมเชื่อมต่อกับพอร์ตอนุกรม RS232

```
#include <18f4431.h>
#device adc=8
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=8000000)
#use rs232(baud=9600, xmit=PIN_C6,
rcv=PIN_C7)
```

2. ตัวอย่างโปรแกรมรับค่าความเร็วรอบจาก encoder

```
pos1 = (float)pos;
delay_ms(5);
pos2 = (float)pos;
speed = (pos2-pos1);
speed = (speed*2*3.14)/600;
speed2=speed;
dt = get_timer1();
dt = dt*8*0.0000005;
printf("%.3f\t%f\t%f\t%f\n",speed,dt,v,vf);
```





ภาพที่ 10 ผังการทำงานของโปรแกรมที่ใช้ในการควบคุม

3. ตัวอย่างการเขียนโปรแกรมรับค่าคำสั่งความเร็วรอบของมอเตอร์จาก PIC16F877A ที่เป็นสัญญาณอนาล็อก

```
set_adc_channel(1);
delay_us(10);
value = read_adc();
```

4. ตัวอย่างการเขียนโปรแกรมสร้างความเร็วรอบหลายๆค่าในตัว PIC16F877Aที่จ่ายสัญญาณนี้ให้กับ PIC18F4431

```
#define MCP4922_CS PIN_B1
#define MCP4922_CLK PIN_B2
#define MCP4922_DIN PIN_B3
#define MCP4922_LDAC PIN_B4
#define MCP4922_SHDN PIN_B5
```

5. ตัวอย่างการเขียนโปรแกรมรับค่ากระแสที่ไหลผ่านตัวมอเตอร์

```
set_adc_channel(0);
```

```
delay_us(10);
value = read_adc();
value2=read_adc();
trvalue2=(value2*(5.0/255));
i= abs((trvalue2)/0.186);
```

6. ตัวอย่างการสร้างสัญญาณ PWM ซึ่งโปรแกรมมีดังนี้

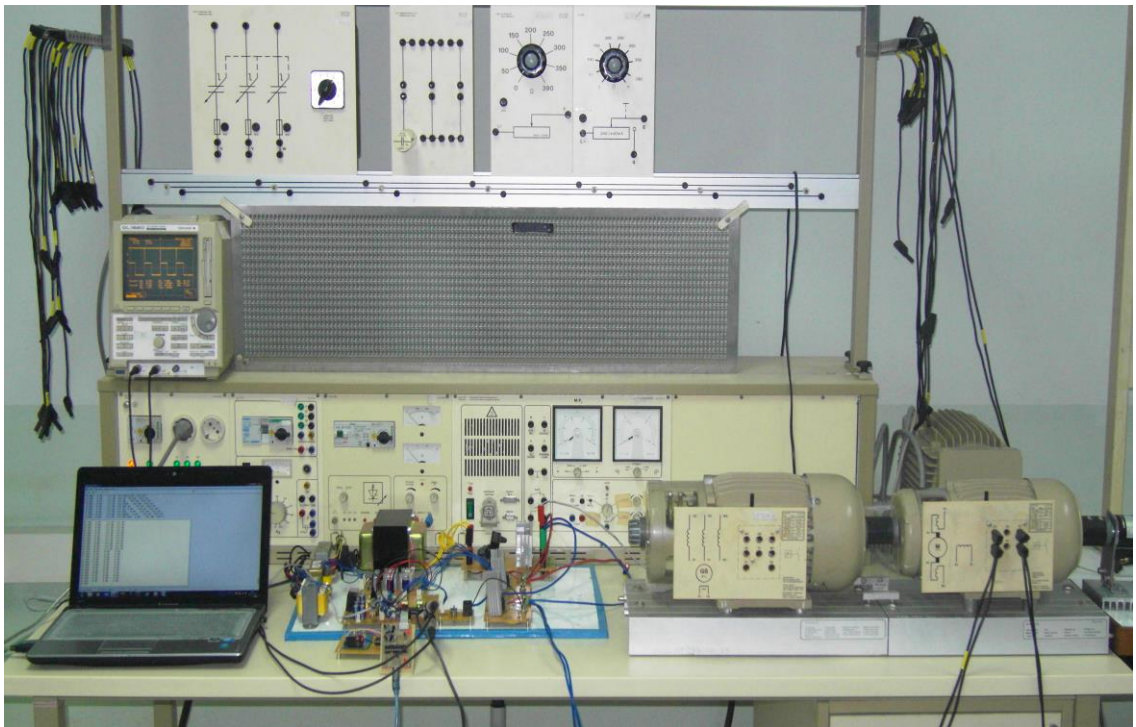
```
SETUP_CCP2(CCP_CAPTURE_RE);
enable_interrupts(INT_CCP1);
setup_ccp1(CCP_PWM);
```

ในส่วนของการสร้างสัญญาณสร้าง PWM ของ Va และ Vf คือ

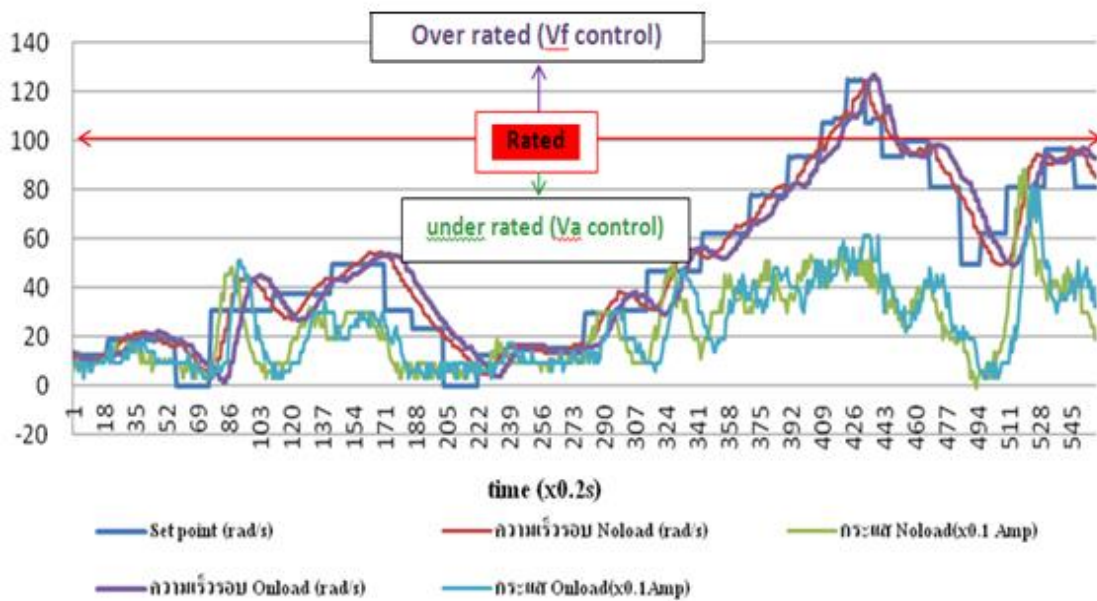
$$PWM = PWM + (Kp0 * error) + (Ki0 * (error+error2) * dt);$$

$$FPWM = FPWM + (Kp1 * error) + (Ki1 * (error+error2) * dt);$$





ภาพที่11 ชุดสาธิตการควบคุมมอเตอร์กระแสตรงแบบกระตุ้นแยก ด้วยไมโครคอนโทรลเลอร์ตระกูล PIC



ภาพที่12 ผลการทดลองกระแสและความเร็วรอบขณะ No load และ On load เทียบกับความเร็วรอบที่ตั้งไว้ (set point)

ผลการวิจัย

จากการทดสอบเพื่อการควบคุมที่มีประสิทธิภาพของมอเตอร์กระแสตรงแบบกระตุ้นแยกแบบลูบปิดด้วยไมโครคอนโทรลเลอร์นั้น มีผลการดำเนินงานดังนี้

1. ผลการทดสอบชุดควบคุมไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC18F4431 สามารถควบคุมความเร็วรอบและกระแสของมอเตอร์กระแสตรงแบบกระตุ้นแยกได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ตามทฤษฎี PI control ที่ขนาดของมอเตอร์ 1000 วัตต์ แรงดัน 220 โวลต์ และขนาดกระแส 5.6 แอมป์ ที่ความเร็วรอบตั้งแต่ 0-2055 รอบต่อนาที ซึ่งเป็นมอเตอร์ที่มีขนาดใหญ่ โดยค่าของความเร็ว กระแส จะแสดงบนหน้าต่างของโปรแกรม Hyperterminal ซึ่งสื่อสารผ่านพอร์ตอนุกรมขณะมีการเปลี่ยนแปลงค่าความเร็วที่กำหนดไว้หรือมีการเปลี่ยนแปลงโพลด์ของมอเตอร์ ความเร็วจริงของมอเตอร์มีค่าใกล้เคียงความเร็วควบคุมที่กำหนดไว้ และเวลาเปลี่ยนแปลงความเร็วของมอเตอร์จริงที่เข้าใกล้ความเร็วที่กำหนดไว้ลดลงเฉลี่ยเหลือเพียง 0.85 วินาทีเท่านั้นเมื่อนำผลที่ได้มาพลอตกราฟจะได้ดังภาพที่ 12

2. ผลการทดสอบการทำงานของวงจรถ้าสามารถขับมอเตอร์ได้ถึงพิกัดที่กำหนด ในช่วงที่ความเร็วรอบเกินพิกัด สามารถลดค่าแรงดันฟิลต์ (Vf) ลงได้ แต่จะเกิดความผิดเพี้ยนของกระแสเนื่องจากสัญญาณรบกวนทำให้ส่งผลต่อการรับค่าของชุดควบคุม ค่าที่ส่งจึงมีความผิดเพี้ยนบ้างแต่ไม่ถึง 10 ใน 10,000 ค่า หรือคิดเป็น 0.1%

สรุป

จากการทดลองของชุดควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นแยกขนาด 1000 วัตต์ ด้วยไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC18F4431 สามารถที่จะปรับความเร็วรอบที่เหมาะสมให้สอดคล้องกับคำสั่งความเร็วรอบของ PIC16F877A ได้ ซึ่งมีความคลาดเคลื่อนต่ำกว่า 2 % เมื่อไม่คิดค่า Mechanical time constants (T_m) เนื่องจากช่วงนี้ไม่สามารถควบคุมได้ สาเหตุเพราะถึงแม้ว่าจะจ่าย

สัญญาณ Mv ให้ต่ำลงมากที่สุดคือ 0 มอเตอร์ก็จะไม่หยุดทันที

การทดสอบเมื่อเพิ่มโพลด์ให้กับมอเตอร์ วงจรควบคุมสามารถที่จะรักษาความเร็วรอบให้เข้าใกล้ความเร็วที่กำหนดไม่ว่าความเร็วที่ PIC16F877A จะกำหนดมาเท่าไรก็ตาม หรือไม่ว่าจะมีการเปลี่ยนแปลงโพลด์ของมอเตอร์ ค่าความเร็วรอบก็ยังรักษาระดับความเร็วรอบให้เข้าใกล้ค่าความเร็วรอบที่กำหนดอยู่เสมอ

ข้อเสนอแนะ

1. การเก็บผลการทดลองจะเกิดปัญหาในส่วนของสัญญาณรบกวนเนื่องจากความถี่สูงในการสวิตช์ของมอสเฟสโดยเฉพาะเมื่อขับมอเตอร์ด้วยความเร็วรอบสูงๆ ทำให้ค่ารอบและกระแสที่วัดได้มีค่าความคลาดเคลื่อนจากค่าจริง แนวทางการแก้ไขคือต้องสร้างวงจรกรองความถี่ให้สัญญาณที่เข้าสู่ไมโครคอนโทรลเลอร์เรียบขึ้น

2. การเขียนโปรแกรมควบคุมในส่วนการรับสัญญาณกระแส จะมีกระแสในส่วนหนึ่งที่ทำให้เกิดความผิดเพี้ยนเนื่องจากเป็นกระแสย้อนกลับ ทำให้ไมโครคอนโทรลเลอร์อ่านค่าเป็นค่าติดลบ วิธีการแก้ไขคือในส่วนของตัวโปรแกรมจะมีฟังก์ชันทางคณิตศาสตร์ให้ใช้งานให้กลับค่าของกระแสจากลบให้เป็นบวกและกำหนดได้ว่าถ้ากระแสที่มีค่าเกินกว่าค่าความเป็นจริงให้โปรแกรมไม่นำมาคิด

3. การทดสอบรับสัญญาณกระแสหากศึกษาจาก data sheet แล้วกระแสที่ 0 แอมป์ สัญญาณจะออกมาให้ไมโครคอนโทรลเลอร์เท่ากับ 2.5 โวลต์ แต่เมื่อทดสอบจริงจะได้กระแสเพียง 2.482 โวลต์ ดังนั้นควรกำหนดการเขียนโปรแกรมรับค่ากระแสที่ 0 โวลต์ เป็น 2.482 โวลต์ไม่ควรกำหนดตาม data sheet



เอกสารอ้างอิง

- [1] Sasiya Udomsuk; et al. (2011). Power Loss Identification of Separately Excited DC Motor Using Adaptive Tabu Search. **European Journal of Scientific Research**. 60(4): 488- 497.
- [2] จิรายุทธ แก้วอาสา. (2549). ชุดสาธิตการควบคุมความเร็วมอเตอร์กระแสตรงแบบลููปปิดบนจอคอมพิวเตอร์ด้วยโปรแกรม visual basic. วารสารเทคโนโลยีสารสนเทศ.

