# บทที่ 7
# ตัวควบคุมแบบฐานกฎสำหรับการประหยัดพลังงาน

## 7.1 บทนำ

การควบคุมมอเตอร์ไฟฟ้ากระแสตรงชนิดแยกกระตุ้นเพื่อประหยัดพลังงานสำหรับงานวิจัยนี้ใช้ตัวควบคุมที่เรียกว่า ตัวควบคุมแบบฐานกฎ (rule based controller) โดยมีหลักการการคำนวณที่ไม่ยุ่งยากและซับซ้อนมากนัก ตัวควบคุมดังกล่าวจะใช้สำหรับวิธีการประหยัดพลังงานซึ่งมีการคำนวณหาค่ากระแสสนามที่เหมาะสมค่าหนึ่งตามสภาวะการทำงานของมอเตอร์ ที่ทำให้กำลังงานสูญเสียในมอเตอร์มีค่าน้อยที่สุด รวมถึงควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงชนิดแยกกระตุ้นให้ได้ตามความต้องการอีกด้วย ซึ่งในบทนี้ได้นำเสนอรายละเอียดต่าง ๆ เกี่ยวกับตัวควบคุมแบบฐานกฎที่ใช้ในงานวิจัย

## 7.2 การออกแบบตัวควบคุมแบบฐานกฎ

การควบคุมมอเตอร์ไฟฟ้ากระแสตรงชนิดแยกกระตุ้นเพื่อประหยัดพลังงานสำหรับงานวิจัยนี้ได้เลือกลักษณะของตัวควบคุมเป็นตัวควบคุมแบบฐานกฎ ซึ่งจัดอยู่ในกลุ่มตัวควบคุมชำนาญการ (expert controller) ชนิดหนึ่ง โดยองค์ประกอบของตัวควบคุมดังกล่าว แบ่งออกเป็น 2 ส่วน คือ เครื่องอนุมาน (inference engine) และฐานความรู้ (knowledge base) เครื่องอนุมานในที่นี้คือคอมพิวเตอร์ ส่วนฐานความรู้นั้นประกอบไปด้วยส่วนที่เป็นข้อมูล (database) และส่วนที่เป็นกฎ (rule base) ให้การตัดสินสั่งการ โดยมีรูปแบบเป็นตรรกเชิงศึกษาสำนึก (heuristic logic) (กองพล อารีรักษ์, 2545) ดังนี้

ถ้า.....แล้ว..... (if.....then..... )

การออกแบบตัวควบคุมแบบฐานกฎสำหรับงานวิจัยนี้ กฎการควบคุมได้มาจากการศึกษาวิเคราะห์พฤติกรรมของระบบ ซึ่งการศึกษาวิเคราะห์พฤติกรรมดังกล่าวอาศัยการสังเกตระบบ รวมถึงมีการเก็บข้อมูลต่าง ๆ ที่เป็นประโยชน์ ทั้งนี้เนื่องจากการออกแบบกฎในงานวิจัยมีจุดประสงค์เพื่อควบคุมการทำงาน 2 โหมด คือ กระแสสนามของมอเตอร์และความเร็วรอบของมอเตอร์ ซึ่งการออกแบบกฎมีรายละเอียดดังต่อไปนี้

### 7.2.1 โหมดการทำงานที่ 1 การควบคุมกระแสสนามของมอเตอร์

โหมดการทำงานที่ 1 จะคำนวณหาค่ากระแสสนามที่เหมาะสมเพื่อประหยัดพลังงาน โดยก่อนการคำนวณโปรแกรมจะรับข้อมูลความเร็วรอบและแรงบิดตามที่ต้องการ หลังจากโปรแกรมคำนวณค่ากระแสสนามเป็นที่เรียบร้อยแล้ว โปรแกรมจะส่งค่ารอบทำงานเพื่อทำการปรับมุมจุดชนวนที่มอสเฟตของวงจรแปลงผันแบบบัคก์ฝั่งวงจรสนามให้ได้กระแสสนามตามที่ต้องการ จึงจะเปลี่ยนไปทำงานในโหมดการทำงานที่ 2 (อธิบายไว้ในหัวข้อต่อไป)

การควบคุมกระแสสนามด้วยวิธีการข้างต้นนั้น คือการปรับค่าแรงดันฝั่งวงจรสนาม เนื่องจากกระแสสนามมีค่าแปรผันตรงกับแรงดันทางฝั่งสนาม ดังนั้นจึงได้ทำการออกแบบและปรับเปลี่ยนกฎในการควบคุม ดังตารางที่ 7.1 ดังนี้

ตารางที่ 7.1 การทดสอบกฎ

| ชุดของกฎ | error1 > 0 | | error1 < 0 | | เวลาที่ใช้ (นาที) |
|---|---|---|---|---|---|
| | ขนาดของ error1 | เพิ่มรอบทำงาน | ขนาดของ error1 | ลดรอบทำงาน | |
| 1 | >0.015 | 1.5 | <-0.015 | 1.5 | 1.12 |
| | >0.010 | 0.5 | <-0.010 | 0.5 | |
| | >0.005 | 0.1 | <-0.005 | 0.1 | |
| 2 | >0.015 | 2.0 | <-0.015 | 2.0 | 0.53 |
| | >0.012 | 1.0 | <-0.012 | 1.0 | |
| | >0.010 | 0.5 | <-0.010 | 0.5 | |
| | >0.007 | 0.3 | <-0.007 | 0.3 | |
| | >0.005 | 0.1 | <-0.005 | 0.1 | |
| 3 | >0.015 | 2.5 | <-0.015 | 2.5 | 0.42 |
| | >0.012 | 1.5 | <-0.012 | 1.5 | |
| | >0.010 | 1.0 | <-0.010 | 1.0 | |
| | >0.007 | 0.5 | <-0.007 | 0.5 | |
| | >0.005 | 0.1 | <-0.005 | 0.1 | |

โดย error1 คือ ค่าผิดพลาดของกระแสสนาม คำนวณได้ดังสมการที่ 7.1 ดังนี้

$$error1 = Value.Ifcal - Value.Ifact \tag{7.1}$$

ส่วนเปอร์เซ็นต์ค่าผิดพลาดของกระแสสนาม คำนวณได้ดังสมการที่ (7.2) คือ

$$percent_error1 = \frac{Value.Ifcal - Value.Ifact}{Value.Ifcal} \times 100 \tag{7.2}$$

เมื่อ Value.Ifcal คือ กระแสสนามจากการคำนวณด้วยวิธีฐานแบบจำลอง

Value.Ifact คือ กระแสสนามในขณะใด ๆ

การทดสอบกฎดังกล่าว ทำการทดสอบที่แรงบิด 0.2 N·m และความเร็วรอบ 1500 rpm จากตารางที่ 7.1 จะพบว่าเมื่อปรับกฎเปลี่ยนไป จะทำให้เวลาในการลู่เข้าเปลี่ยนแปลงไปด้วย ดังนั้นจึงได้เลือกกฎชุดที่ 3 ในการควบคุมกระแสสนามของมอเตอร์เนื่องจากใช้เวลาในการลู่เข้าน้อยที่สุด และสามารถอธิบายรายละเอียดได้ดังต่อไปนี้

กฎที่ 1 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นบวกและมีขนาดเกินกว่า 0.015 A> แล้ว <เพิ่มรอบทำงาน 2.5 เปอร์เซ็นต์>

กฎที่ 2 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นบวกและมีขนาดเกินกว่า 0.012 A> แล้ว <เพิ่มรอบทำงาน 1.5 เปอร์เซ็นต์>

กฎที่ 3 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นบวกและมีขนาดเกินกว่า 0.010 A> แล้ว <เพิ่มรอบทำงาน 1.0 เปอร์เซ็นต์>

กฎที่ 4 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นบวกและมีขนาดเกินกว่า 0.007 A> แล้ว <เพิ่มรอบทำงาน 0.5 เปอร์เซ็นต์>

กฎที่ 5 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นบวกและมีขนาดเกินกว่า 0.005 A> แล้ว <เพิ่มรอบทำงาน 0.1 เปอร์เซ็นต์>

กฎที่ 6 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นลบและมีขนาดเกินกว่า 0.015 A> แล้ว <ลดรอบทำงาน 2.5 เปอร์เซ็นต์>

กฎที่ 7 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นลบและมีขนาดเกินกว่า 0.012 A> แล้ว <ลดรอบทำงาน 1.5 เปอร์เซ็นต์>

กฎที่ 8 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นลบและมีขนาดเกินกว่า 0.010 A> แล้ว <ลดรอบทำงาน 1.0 เปอร์เซ็นต์>

กฎที่ 9 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นลบและมีขนาดเกินกว่า 0.007 A>
แล้ว <ลดรอบทำงาน 0.5 เปอร์เซ็นต์>

กฎที่ 10 ถ้า <ค่าผิดพลาดของกระแสสนามเป็นลบและมีขนาดเกินกว่า 0.005 A>
แล้ว <ลดรอบทำงาน 0.1 เปอร์เซ็นต์>

### 7.2.2 โหมดการทำงานที่ 2 การควบคุมความเร็วรอบของมอเตอร์

โหมดการทำงานที่ 2 สำหรับควบคุมความเร็วรอบให้ได้ตามที่ต้องการ การควบคุมดังกล่าวทำได้โดยการปรับค่าแรงดันไฟฟ้าของวงจรฝั่งอาร์เมเจอร์ ด้วยการปรับมุมจุดชนวนของมอสเฟตของวงจรแปลงผันแบบบัคก์ฝั่งวงจรอาร์เมเจอร์ โดยเงื่อนไขการปรับค่าแรงดันไฟฟ้าพิจารณาจากค่าผิดพลาดของความเร็วรอบ (error2) ดังสมการที่ (7.3) ดังนี้

$$error2 = N_t - Value.Nact \tag{7.3}$$

ส่วนเปอร์เซ็นต์ค่าผิดพลาดของความเร็วรอบ คำนวณได้ดังสมการที่ (7.4) คือ

$$percent_error2 = \frac{N_t - Value.Nact}{N_t} \times 100 \tag{7.4}$$

เมื่อ N_t คือ ความเร็วรอบของมอเตอร์ที่ต้องการ
Value.Nact คือ ความเร็วรอบของมอเตอร์ในขณะใด ๆ

การควบคุมความเร็วรอบของมอเตอร์ หมายถึง การปรับค่าแรงดันฝั่งวงจรอาร์เมเจอร์ เนื่องจากความเร็วรอบมีลักษณะแปรผันตรงกับแรงดันอาร์เมเจอร์ จึงได้ออกแบบกฎของการควบคุมนี้ไว้เพียง 1 ชุด ไม่ได้ทำการปรับเปลี่ยนกฎและเปรียบเทียบเวลาในการลู่เข้าดังเช่นกฎการควบคุมกระแสสนาม เพราะกฎชุดดังกล่าวใช้เวลาในการลู่เข้าไม่แตกต่างกันในแต่ละครั้งที่มีการปรับเปลี่ยนกฎการควบคุมกระแสสนาม ผู้วิจัยจึงเห็นว่าไม่จำเป็นต้องทำการปรับเปลี่ยนอะไร ดังนั้นกฎที่ได้จากการออกแบบจึงมีรายละเอียดดังต่อไปนี้

กฎที่ 1 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นบวกและมีขนาดเกินกว่า 200 rpm>
แล้ว <เพิ่มรอบทำงาน 1.5 เปอร์เซ็นต์>

กฎที่ 2 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นบวกและมีขนาดเกินกว่า 100 rpm>
แล้ว <เพิ่มรอบทำงาน 1.0 เปอร์เซ็นต์>

กฎที่ 3 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นบวกและมีขนาดเกินกว่า 50 rpm> แล้ว <เพิ่มรอบทำงาน 0.5 เปอร์เซ็นต์>

กฎที่ 4 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นบวกและมีขนาดเกินกว่า 10 rpm> แล้ว <เพิ่มรอบทำงาน 0.1 เปอร์เซ็นต์>

กฎที่ 5 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นลบและมีขนาดเกินกว่า 200 rpm> แล้ว <ลดรอบทำงาน 1.5 เปอร์เซ็นต์>

กฎที่ 6 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นลบและมีขนาดเกินกว่า 100 rpm> แล้ว <ลดรอบทำงาน 1.0 เปอร์เซ็นต์>

กฎที่ 7 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นลบและมีขนาดเกินกว่า 50 rpm> แล้ว <ลดรอบทำงาน 0.5 เปอร์เซ็นต์>

กฎที่ 8 ถ้า <ค่าผิดพลาดของความเร็วรอบเป็นลบและมีขนาดเกินกว่า 10 rpm> แล้ว <ลดรอบทำงาน 0.1 เปอร์เซ็นต์>

## 7.3 การสร้างตัวควบคุมแบบฐานกฎด้วยบอร์ด dsPIC30F2010

การสร้างตัวควบคุมแบบฐานกฎด้วยบอร์ด dsPIC30F2010 มีแผนภาพการทำงานดังรูปที่ 7.1 จากแผนภาพดังกล่าวจะเป็นระบบขับเคลื่อนมอเตอร์ที่มีตัวควบคุมแบบฐานกฎ ซึ่งตัวควบคุมดังกล่าวจะควบคุมกระแสสนามและความเร็วรอบของมอเตอร์ สำหรับการประหยัดพลังงาน โดยโปรแกรมภาษาซีเฉพาะส่วนของตัวควบคุมแบบฐานกฎบนบอร์ด dsPIC30F2010 สามารถอธิบายได้ดังนี้

- *โปรแกรมในโหมดการควบคุมกระแสสนาม*

```
error1 = Value.Ifcal - Value.Ifact;                    //คำนวณค่าความผิดพลาดของกระแสสนาม

if(error1 >= 0)
percent_error1 = (Value.Ifcal - Value.Ifact)*100/Value.Ifcal;
else
percent_error1 = (Value.Ifact - Value.Ifcal)*100/Value.Ifcal;
                                                       //ตรวจสอบค่าความผิดพลาดว่าเป็นบวกหรือ
                                                       ลบแล้วคำนวณหาเปอร์เซ็นต์ค่าความผิดพลาด
%เริ่มโหมดการควบคุมกระแสสนาม
{
```

```
if (percent_error1 <= 5)                    //ถ้าเปอร์เซ็นต์ค่าความผิดพลาดน้อยกว่าหรือ
                                            เท่ากับ 5
        {
 System.EnaCtrlIFieldMode = false;          //ออกจากโหมดการควบคุมกระแส
System.EnaCal_IField = false;               //หยุดการคำนวณกระแสสนาม
System.EnaCtrlSpeedMode = true;             //เริ่มโหมดการควบคุมความเร็วรอบ
Value.CalPWM1Duty = Value.PWM1Duty;         //ส่งค่ารอบทำงานของฝั่งวงจรสนาม
        }
else
        {
if (error1>0)                               //ถ้าค่าความผิดพลาดเป็นบวก
                {
if (error1 > 15)                            //ถ้าค่าความผิดพลาดมากกว่า 15
Value.CalPWM1Duty = Value.PWM1Duty+25;      //เพิ่มรอบทำงาน 25 (2.5 เปอร์เซ็นต์)
else if (error1 > 12)                       //ถ้าค่าความผิดพลาดมากกว่า 12
Value.CalPWM1Duty = Value.PWM1Duty+15;      //เพิ่มรอบทำงาน 15 (1.5 เปอร์เซ็นต์)
else if (error1 > 10)                       //ถ้าค่าความผิดพลาดมากกว่า 10
Value.CalPWM1Duty = Value.PWM1Duty+10;      //เพิ่มรอบทำงาน 10 (1.0 เปอร์เซ็นต์)
else if (error1 > 7)                        //ถ้าค่าความผิดพลาดมากกว่า 7
Value.CalPWM1Duty = Value.PWM1Duty+5;       //เพิ่มรอบทำงาน 5 (0.5 เปอร์เซ็นต์)
else if (error1 > 5)                        //ถ้าค่าความผิดพลาดมากกว่า 5
Value.CalPWM1Duty = Value.PWM1Duty+1;       //เพิ่มรอบทำงาน 1 (0.1 เปอร์เซ็นต์)
if(Value.CalPWM1Duty > 950)                 //ถ้ารอบทำงานของฝั่งวงจรสนาม
                                            มากกว่า 950 (95 เปอร์เซ็นต์)
        Value.CalPWM1Duty = 950;            //ให้รอบทำงานของฝั่งวงจรสนาม
                                            เท่ากับ 950 (95 เปอร์เซ็นต์)
                }
else                                        //ถ้าค่าความผิดพลาดเป็นลบ
                {
if (error1 < -15)                           //ถ้าค่าความผิดพลาดน้อยกว่า -15
Value.CalPWM1Duty = Value.PWM1Duty-25;      //ลดรอบทำงาน 25 (2.5 เปอร์เซ็นต์)
```

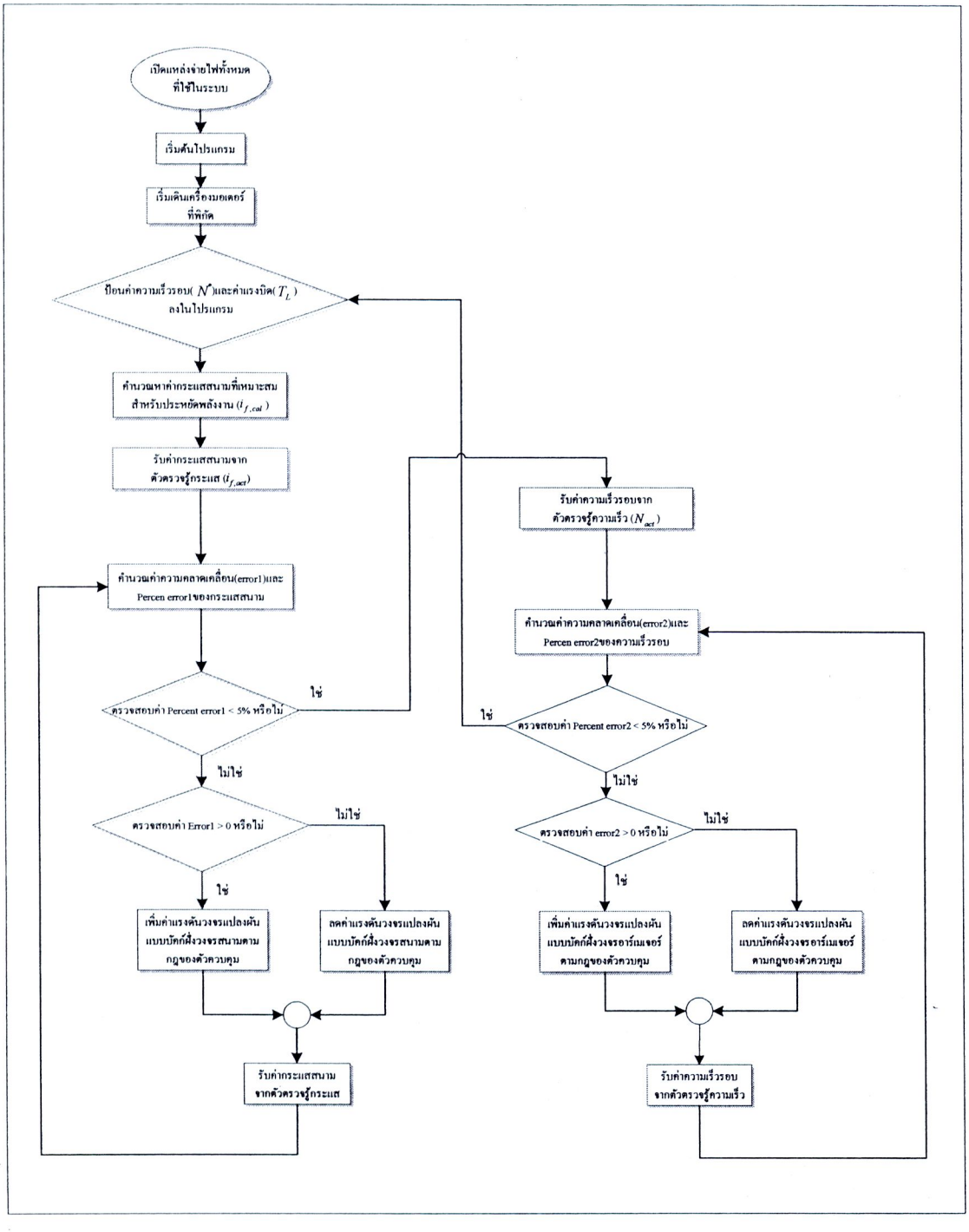
```
else if (error1 < -12)                          //ถ้าค่าความผิดพลาดน้อยกว่า -12
Value.CalPWM1Duty = Value.PWM1Duty-15;          //ลดรอบทำงาน 15 (1.5 เปอร์เซ็นต์)
else if (error1 < -10)                          //ถ้าค่าความผิดพลาดน้อยกว่า -10
Value.CalPWM1Duty = Value.PWM1Duty-10;          //ลดรอบทำงาน 10 (1.0 เปอร์เซ็นต์)
else if (error1 < -7)                           //ถ้าค่าความผิดพลาดน้อยกว่า -7
Value.CalPWM1Duty = Value.PWM1Duty-5;           //ลดรอบทำงาน 5 (0.5 เปอร์เซ็นต์)
else if (error1 < -5)                           //ถ้าค่าความผิดพลาดน้อยกว่า -5
Value.CalPWM1Duty = Value.PWM1Duty-1;           //ลดรอบทำงาน 1 (0.1 เปอร์เซ็นต์)
if(Value.CalPWM1Duty < 100)                     //ถ้ารอบทำงานของฝั่งวงจรสนาม
น้อยกว่า 100 (10 เปอร์เซ็นต์)
Value.CalPWM1Duty = 100;                        //ให้รอบทำงานของฝั่งวงจรสนาม
                                                เท่ากับ 100 (10 เปอร์เซ็นต์)
                }
        }
}
else Value.CalPWM1Duty = Value.PWM1Duty;        //ส่งค่ารอบทำงานของฝั่งวงจรสนาม
```

- *โปรแกรมในโหมดการควบคุมความเร็วรอบ*

```
error2 = N_t - Value.Nact;                      //คำนวณค่าความผิดพลาดของความเร็วรอบ

if(error2 >= 0)
 Value.percent_error2 = (unsigned long)( N_t - Value.Nact)*100/N_t;
else
Value.percent_error2 = (unsigned long)( Value.Nact - N_t)*100/N_t;
                                                //ตรวจสอบค่าความผิดพลาดว่าเป็นบวกหรือ
                                                ลบแล้วคำนวณหาเปอร์เซ็นต์ค่าความผิดพลาด
%เริ่มโหมดการควบคุมความเร็วรอบ
if (Value.percent_error2 <= 5)                  //ถ้าเปอร์เซ็นต์ค่าความผิดพลาดน้อยกว่าหรือ
        {                                       เท่ากับ 5
Value.CalPWM2Duty = Value.PWM2Duty;             //ส่งค่ารอบทำงานของฝั่งวงจรอาร์เมเจอร์
```

```
        }
else
        {
if (error2 > 0)                                 //ถ้าค่าความผิดพลาดเป็นบวก
                {
if (error2 > 200)                               //ถ้าค่าความผิดพลาดมากกว่า 200
Value.CalPWM2Duty = Value.PWM2Duty+15;          //เพิ่มรอบทำงาน 15 (1.5 เปอร์เซ็นต์)
else if (error2 > 100)                          //ถ้าค่าความผิดพลาดมากกว่า 100
Value.CalPWM2Duty = Value.PWM2Duty+10;          //เพิ่มรอบทำงาน 10 (1.0 เปอร์เซ็นต์)
else if (error2 > 50)                           //ถ้าค่าความผิดพลาดมากกว่า 50
Value.CalPWM2Duty = Value.PWM2Duty+5;           //เพิ่มรอบทำงาน 5 (0.5 เปอร์เซ็นต์)
else //if (error2 > 10)                         //ถ้าค่าความผิดพลาดมากกว่า 10
Value.CalPWM2Duty = Value.PWM2Duty+1;           //เพิ่มรอบทำงาน 1 (0.1 เปอร์เซ็นต์)
                }
else                                            //ถ้าค่าความผิดพลาดเป็นลบ
                {
if (error2 < -200)                              //ถ้าค่าความผิดพลาดน้อยกว่า -200
Value.CalPWM2Duty = Value.PWM2Duty-15;          //ลดรอบทำงาน 15 (1.5 เปอร์เซ็นต์)
else if (error2 < -100)                         //ถ้าค่าความผิดพลาดน้อยกว่า -100
Value.CalPWM2Duty = Value.PWM2Duty-10;          //ลดรอบทำงาน 10 (1.0 เปอร์เซ็นต์)
else if (error2 < -50)                          //ถ้าค่าความผิดพลาดน้อยกว่า -50
Value.CalPWM2Duty = Value.PWM2Duty-5;           //ลดรอบทำงาน 5 (0.5 เปอร์เซ็นต์)
else //if (error2 < -10)                        //ถ้าค่าความผิดพลาดน้อยกว่า -10
Value.CalPWM2Duty = Value.PWM2Duty-1;           //ลดรอบทำงาน 1 (0.1 เปอร์เซ็นต์)
                }
        }
}
else Value.CalPWM2Duty = Value.PWM2Duty;        //ส่งค่ารอบทำงานของฝั่งวงจรอาร์เมเจอร์
```




รูปที่ 7.1 แผนภาพการทำงานของระบบขับเคลื่อนมอเตอร์ที่มีตัวควบคุมแบบฐานกฎ

จากรูปที่ 7.1 การทำงานของระบบขับเคลื่อนมอเตอร์ ขั้นแรกจะทำการเปิดแหล่งจ่ายไฟทั้งหมดที่ใช้ในระบบ หลังจากนั้นจึงเริ่มต้นการทำงานของโปรแกรมซึ่งโปรแกรมดังกล่าวจะควบคุมกระบวนการทำงานทั้งหมด เพื่อประหยัดพลังงานในระบบขับเคลื่อนมอเตอร์ไฟฟ้า

กระแสตรงชนิดแยกกระตุ้น โดยในช่วงต้นของกระบวนการทำงานจะเริ่มเดินเครื่องมอเตอร์ จนกระทั่งได้แรงดันวงจรอาร์เมเจอร์และวงจรสนามที่พิกัด (220 V) ต่อมาจึงทำการป้อนค่าความเร็วรอบและแรงบิดที่ต้องการ เพื่อเป็นข้อมูลสำหรับคำนวณหาค่ากระแสสนามที่เหมาะสมสำหรับประหยัดพลังงาน หลังจากนั้นจะทำการปรับค่ากระแสสนามดังกล่าวจากวงจรแปลงผันแบบบัคก์ฝั่งวงจรสนามด้วยไมโครคอนโทรลเลอร์ และรับค่ากระแสสนามของมอเตอร์ผ่านตัวตรวจรู้กระแส เพื่อตรวจสอบว่ากระแสสนามดังกล่าวมีค่าใกล้เคียงกับกระแสสนามที่ต้องการหรือไม่ ซึ่งการตรวจสอบจะทำการคำนวณค่าเปอร์เซ็นต์ความผิดพลาดของกระแสสนามตามสมการที่ (7.2) ถ้าค่าเปอร์เซ็นต์ความผิดพลาดมีค่ามากกว่า 5 เปอร์เซ็นต์ วงจรแปลงผันแบบบัคก์จะทำหน้าที่เพิ่มและลดแรงดันฝั่งวงจรสนามตามกฎของตัวควบคุมที่อธิบายไว้ในหัวข้อที่ 7.2 จนกว่าค่าเปอร์เซ็นต์ความผิดพลาดของกระแสสนามจะมีค่าน้อยกว่า 5 เปอร์เซ็นต์ แล้วไมโครคอนโทรลเลอร์จึงจะเปลี่ยนไปควบคุมความเร็วรอบของมอเตอร์ โดยมีการทำงานเช่นเดียวกับการควบคุมกระแสสนาม ต่างกันเพียงแค่การควบคุมความเร็วรอบนั้น วงจรแปลงผันแบบบัคก์ที่ทำหน้าที่เพิ่มลดแรงดัน คือ ฝั่งวงจรอาร์เมเจอร์ และเมื่อต้องการหยุดการทำงานของโปรแกรมหรือเปลี่ยนจุดการทำงานใหม่ สามารถทำได้โดยลดแรงดันจากแหล่งจ่ายไฟจนหมดแล้วจึงป้อนค่าความเร็วรอบและแรงบิดค่าใหม่ลงในโปรแกรม

## 7.4 สรุป

เนื้อหาในบทนี้นำเสนอตัวควบคุมแบบฐานกฎที่ใช้สำหรับการประหยัดพลังงาน และควบคุมกระบวนการปรับความเร็วรอบของมอเตอร์ให้ได้ตามความต้องการ โดยตัวควบคุมแบบฐานกฎมีกระบวนการทำงานที่ไม่ยุ่งยากซับซ้อน คือ จะตรวจสอบจากการคำนวณค่าผิดพลาด แล้วตัวควบคุมดังกล่าวจึงจะทำการปรับเพิ่มหรือลดแรงดันตามกฎที่ได้ออกแบบไว้ และจากการทดสอบกฎจะพบว่าตัวควบคุมชนิดนี้ใช้เวลาในการควบคุมไม่นานมาก จึงถือเป็นข้อดีของตัวควบคุมแบบฐานกฎ คือ ไม่ซับซ้อนและใช้เวลาไม่มาก และต่อไปจะเป็นการทดสอบการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงชนิดแยกกระตุ้นสำหรับประหยัดพลังงาน โดยมีผลเปรียบเทียบกับวิธีการขับเคลื่อนด้วยวิธีดั้งเดิม ซึ่งจะนำเสนอในบทถัดไป