

ระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วสำหรับขับเคลื่อนมอเตอร์
เหนี่ยวนำที่ใช้การประมาณค่าความเร็วรอบจากระบบจำลองอ้างอิงแบบปรับตัว

A SPEED SENSORLESS VECTOR CONTROL SYSTEM OF INDUCTION
MOTOR DRIVE WITH SPEED ESTIMATION BASED ON
MODEL REFERENCE ADAPTIVE SYSTEM



ณัฐพงศ์ เมืองจันทร์

NUTTAPONG MUANGCHAN

ดงพ.
ท.336
2552

เลขหมู่.....
เลขทะเบียน.....105260
วัน,เดือน,ปี...1.7 พ.ย. 2552

b. 121669A7
i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-EN-M-020-097

**A SPEED SENSORLESS VECTOR CONTROL SYSTEM OF INDUCTION
MOTOR DRIVE WITH SPEED ESTIMATION BASED ON
MODEL REFERENCE ADAPTIVE SYSTEM**

NUTTAPONG MUANGCHAN

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2009
KMITL-2009-EN-M-020-097**

COPYRIGHT 2009

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ ระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วสำหรับขับเคลื่อนมอเตอร์เหนี่ยวนำที่ใช้การประมาณค่าความเร็วรอบจากระบบจำลองอ้างอิงแบบปรับตัว

Thesis Title A Speed Sensorless Vector Control System of Induction Motor Drive with Speed Estimation Based on Model Reference Adaptive System

นักศึกษา นายณัฐพงศ์ เมืองจันทร์

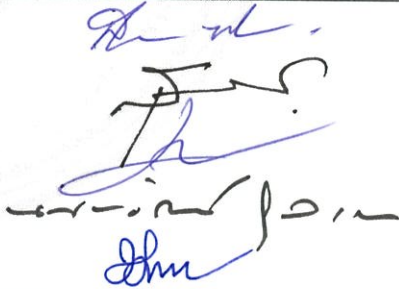
รหัสประจำตัว 47060254

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.วิจิตร กิณเรศ

หมายเลขวิทยานิพนธ์ KMITL-2009-EN-M-020-097

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ดร.สมภพ	ผลไม้	
ดร.สุรินทร์	คำฟอย	
ผศ.ดร.เสถียร	ชัยยศวีร์รัตน์	
ผศ.ดร.นนทวัฒน์	จุลเดชะ	
รศ.ดร.วิจิตร	กิณเรศ	

วัน/เดือน/ปี ที่สอบ วันศุกร์ที่ 15 พฤษภาคม พ.ศ. 2552 เวลา 09.30-11.30 น.

สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 3

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 15 พฤษภาคม พ.ศ. 2552

หัวข้อวิทยานิพนธ์	ระบบควบคุมแบบเวกเตอร์ไ้เซนเซอร์วัดความเร็วสำหรับ ขับเคลื่อนมอเตอร์เหนี่ยวนำที่ใช้การประมาณค่าความเร็วรอบ จากระบบจำลองอ้างอิงแบบปรับตัว
นักศึกษา	นาย ณัฐพงศ์ เมืองจันทร์
รหัสนักศึกษา	47060254
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร. วิจิตร กิณเรศ

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอระบบควบคุมแบบเวกเตอร์ไ้เซนเซอร์วัดความเร็วสำหรับขับเคลื่อนมอเตอร์เหนี่ยวนำที่ใช้การประมาณค่าความเร็วรอบจากระบบจำลองอ้างอิงแบบปรับตัวโดยใช้วิธีฟลักซ์โรเตอร์จากระบบประมาณค่าฟลักซ์ในแบบจำลองกระแสและแบบจำลองแรงดัน โดยนำความแตกต่างของฟลักซ์โรเตอร์ทั้งสองไปปรับเอาต์พุตของค่าประมาณความเร็วโดยผ่านกระบวนการควบคุมแบบพีไอ นอกจากนี้ยังได้บูรณาการระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัวเพื่อลดความซ้ำซ้อนของแบบจำลอง รวมทั้งได้นำเสนอการออกแบบอัตราขยายปรับตัวโดยพิจารณาถึงความเร็วประมาณติดตามความเร็วจริงในช่วงมอเตอร์เปลี่ยนแปลงความเร็ว โดยที่เงื่อนไขการออกแบบนี้จะถูกกำหนดด้วยค่าความเร็วผิดพลาดที่ต้องการและนำไปสู่การหาค่าอัตราขยายปรับตัวแบบอินทิเกรตของระบบประมาณค่าความเร็ว โดยที่อัตราขยายปรับตัวแบบสัดส่วนจะถูกพิจารณาจากความถี่ของฟลักซ์โรเตอร์ ซึ่งเป็นการวิเคราะห์ที่มีเงื่อนไขชัดเจนและเลือกค่าอัตราขยายปรับตัวที่เหมาะสมกับระบบประมาณค่าความเร็วได้ และเพื่อเป็นการตรวจสอบความถูกต้องของแนวคิดที่ได้นำเสนอ วิทยานิพนธ์นี้ได้มีการจำลองการทำงานของระบบโดยรวมทั้งหมดด้วยโปรแกรม Matlab/Simulink เพื่อดูผลการตอบสนองทางพลวัต อีกทั้งในวิทยานิพนธ์นี้ได้ทำการทดสอบระบบจริงกับมอเตอร์เหนี่ยวนำ 3 เฟส ขนาด 3 แรงม้า เพื่อดูผลการตอบสนองของระบบในสภาวะชั่วคราวและสภาวะอยู่ตัวภายใต้ความเร็วคำสั่งและโหลดเป็นแบบขั้น ซึ่งผลจากการทดสอบกับระบบที่ได้เสนอขึ้นนี้ถูกต้องตามแนวคิดที่ได้นำเสนอ โดยที่ผลการตอบสนองของระบบขณะเปลี่ยนแปลงความเร็วจาก 500 ถึง 1410 รอบต่อนาที มีช่วงเวลาขาขึ้น 0.3 วินาที เข้าสู่ความเร็วคำสั่งที่เวลา 0.5 วินาที และใช้เวลาในการกลับทิศความเร็ว 0.8 วินาที นอกจากนี้ในขณะป้อนโหลดและปลดโหลดแบบขั้นอย่างทันทีทันใดขนาด 2 นิวตัน-เมตร ใช้เวลาเพียง 0.8 วินาที ในการกลับเข้าสู่สภาวะอยู่ตัว

Thesis Title	A Speed Sensorless Vector Control System of Induction Motor Drive with Speed Estimation Based on Model Reference Adaptive System
Student	Mr. Nuttapong Muangchan
Student ID.	47060254
Degree	Master of Engineering
Program	Electrical Engineering
Year	2009
Thesis Advisor	Assoc. Prof. Dr. Vijit Kinnares

ABSTRACT

This thesis presents the speed sensorless vector control system of an induction motor drive which speed estimation is based on model reference adaptive system (MRAS). The rotor flux is estimated in both current and voltage models from flux estimation system. The difference between rotor flux obtained from voltage and current models is used to adjust the output of the estimated speed via a PI control process. In addition, the estimation of the closed loop stator flux observer is integrated with the model reference adaptive system in order to reduce the model redundancy. The design condition is specified with required speed error value leading to find the gain of an integrate controller of the speed estimation system. The gain of a proportional controller is determined from the rotor flux frequency. In order to verify the correctness of the proposed concept, the simulation of the overall system using Matlab/Simulink for dynamic response is included. In addition, an experimental test with a 3-phase, 3 HP induction motor has been performed for verifying the dynamic and steady state response under speed command and step load change conditions. The experimental results show that the proposed concept can be used for drive applications with satisfied results. The system response of a speed change from 500 rpm to 1410 rpm gives a rise time of 0.3 sec and a settling time within 0.5 sec. The response of a speed reversal is within 0.8 sec. Additionally, the response of a load disturbance by taking on and off of 2 Nm load torque is within 0.8 sec for changing to a steady state.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาเป็นอย่างสูงจากรศ.ดร. วิจิตร กิณเรศ ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้คำปรึกษาแนะนำและช่วยเหลือเกี่ยวกับการทำวิทยานิพนธ์มาโดยตลอด ผู้วิจัยเองมีความรู้สึกรักซาบซึ้งในความอนุเคราะห์ของท่านและขอกล่าวขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ ประธานและคณะกรรมการสอบวิทยานิพนธ์ผู้ทรงคุณวุฒิทุกท่าน ที่ได้เสนอการแก้ไขข้อผิดพลาดและชี้แนะเพื่อให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณ มหาวิทยาลัยอีสเทิร์นเอเซีย ที่ได้ช่วยสนับสนุนทุนการศึกษาและเวลาในการทำงานวิจัยมาโดยตลอด

ขอขอบคุณ พี่จรรย์ ตั้งวงศ์ชูเกตุ ที่มอบวัสดุและอุปกรณ์สำหรับการทำวิจัยบางส่วนโดยไม่คิดมูลค่า

ขอกราบขอบพระคุณ บิดา มารดา ผู้ให้กำเนิด และเครือญาติ ที่เป็นผู้ให้กำลังใจในการศึกษาเล่าเรียนและชี้ทางที่ถูกที่ควรให้มาโดยตลอด

สุดท้ายนี้ขอบขอบคุณพี่ ๆ และเพื่อนๆ นักศึกษากลุ่ม PMC ที่ให้ความช่วยเหลือและให้กำลังใจอย่างต่อเนื่องตลอดการศึกษาของผู้วิจัย

คุณค่าและประโยชน์อันพึงจะมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอบมอบให้แก่ บิดา มารดา ครุบาอาจารย์ผู้ประสิทธิ์ประสาทวิชาและผู้มีพระคุณทุกท่าน

ณัฐพงศ์ เมืองจันทร์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 วิธีการวิจัย.....	2
1.4 ขอบเขตของงานวิจัย.....	2
1.5 โครงสร้างของวิทยานิพนธ์.....	3
1.6 ประโยชน์ที่ได้รับจากวิทยานิพนธ์.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย.....	4
2.1 แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำ.....	4
2.1.1 การนำเสนอในรูปแบบสเปซเวกเตอร์.....	5
2.1.2 การย้ายแกนอ้างอิงของสเปซเวกเตอร์.....	7
2.1.3 แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำบนแกนอ้างอิงแกนนิ่ง.....	7
2.1.4 แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำบนแกนอ้างอิงแกนหมุน.....	11
2.2 อินเวอร์เตอร์และสเปซเวกเตอร์แรงดัน.....	12
2.2.1 ระบบแหล่งจ่ายแรงดัน 3 เฟส.....	12
2.2.2 ระบบแหล่งจ่ายไฟแบบอินเวอร์เตอร์ 3 เฟส.....	13
2.2.3 เวกเตอร์แรงดันในกรอบอ้างอิงสเตเตอร์.....	16
2.3 เทคนิคสเปซเวกเตอร์พีดับบีเอ็ม.....	18
2.4 แนวคิดของการควบคุมแบบเวกเตอร์.....	22
2.5 การควบคุมอินเวอร์เตอร์ประเภทต่าง ๆ ในอุตสาหกรรม.....	26
2.5.1 การควบคุมอินเวอร์เตอร์แบบ V/F.....	27

สารบัญ (ต่อ)

	หน้า
2.5.2 การควบคุมอินเวอร์เตอร์แบบเวกเตอร์.....	27
2.5.3 การควบคุมอินเวอร์เตอร์แบบเวกเตอร์ไร้เซนเซอร์วัดความเร็ว	28
บทที่ 3 ระบบควบคุมแบบเวกเตอร์.....	32
3.1 การควบคุมแบบเวกเตอร์.....	32
3.2 การควบคุมกระแสบนแกนหมุนสำหรับการควบคุมแบบเวกเตอร์.....	35
3.3 แนวทางการออกแบบอัตราขยายตัวควบคุม PI.....	37
3.3.1 การออกแบบระบบควบคุมกระแส.....	37
3.3.2 การออกแบบระบบควบคุมความเร็ว.....	39
3.4 ระบบประมาณค่าฟลักซ์โดยใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์.....	40
บทที่ 4 ระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็ว.....	45
4.1 ระบบจำลองอ้างอิงแบบปรับตัว.....	45
4.2 การบูรณาการตัวสังเกตวงรอบปิดฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัว.....	47
4.3 ระบบการประมาณค่าความเร็วบนแกนหมุน.....	50
4.4 การออกแบบอัตราขยายปรับตัว.....	54
4.5 การจำลองระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็วบนระบบจำลองอ้างอิงแบบปรับตัวด้วยโปรแกรมคอมพิวเตอร์.....	56
บทที่ 5 การสร้างระบบทดสอบจริง.....	62
5.1 ชุดวงจรคอนเวอร์เตอร์.....	63
5.2 ชุดวงจรอินเวอร์เตอร์.....	64
5.3 ชุดวงจรขับนำสวิตช์.....	66
5.4 ชุดวงจรตรวจจับกระแส.....	67
5.5 วงจรตรวจจับแรงดัน.....	68
5.6 ชุดวงจรขยายและยกกระดับแรงดัน.....	69
5.7 ชุดวงจรป้องกันต่าง ๆ.....	69
5.8 ชุดประมวลผลสัญญาณดิจิทัล.....	71

สารบัญ (ต่อ)

	หน้า
5.9 ชุดการแสดงผลของค่าต่างๆ.....	72
5.10 โครงสร้างทางซอฟต์แวร์ที่ใช้ในการควบคุมมอเตอร์.....	73
บทที่ 6 ผลการทดสอบการทำงานของระบบจริง.....	78
บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ.....	91
เอกสารอ้างอิง.....	93
ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่.....	95
ภาคผนวก ข. โปรแกรมที่ใช้ในการควบคุมระบบ.....	103
ประวัติผู้วิจัย.....	124

สารบัญตาราง

ตารางที่	หน้า
2.1 รูปแบบการสวิตช์และแรงดันเทียบจุดสเทินเสมือนของอินเวอร์เตอร์.....	14
2.2 รูปแบบการสวิตช์และแรงดันเทียบจุดนิวตรอลของอินเวอร์เตอร์.....	14
2.3 รูปแบบการสวิตช์และค่าแรงดันของอินเวอร์เตอร์ในกรอบอ้างอิงสเตเตอร์.....	17
2.4 แสดงการคำนวณเวลาในแต่ละเซกเตอร์.....	20
4.1 พิกัดและพารามิเตอร์ของมอเตอร์เหนี่ยวนำ.....	56
4.2 อัตราขยายระบบควบคุมที่ได้ออกแบบ.....	56

สารบัญรูป

รูปที่	หน้า
2.1 วงจรสมมูลทางพลวัตที่มีแกนอ้างอิงบนแกนหมุนซิงโครนัส.....	4
2.2 โครงสร้างของมอเตอร์เหนี่ยวนำแบบกรงกระรอก.....	6
2.3 วงจรสมมูลทางพลวัตที่มีแกนอ้างอิงบนแกนนิ่งสำหรับมอเตอร์เหนี่ยวนำ ชนิดกรงกระรอก.....	8
2.4 แหล่งจ่ายไฟอิสระ 3 ชุด.....	12
2.5 แหล่งจ่ายไฟแบบอินเวอร์เตอร์ 3 เฟส.....	13
2.6 สถานะอินเวอร์เตอร์ขณะสวิตช์ 100.....	14
2.7 ไดอะแกรมแสดงความสัมพันธ์ของแรงดันเฟส v_{AN}, v_{BN} และ v_{CN}	15
2.8 ไดอะแกรมแสดงความสัมพันธ์ของแรงดันไลน์ v_{AB}, v_{BC} และ v_{CA}	16
2.9 เวกเตอร์แรงดันทั้ง 8 เวกเตอร์.....	17
2.10 แสดงความสัมพันธ์ของแรงดัน v_α และ v_β	18
2.11 การสวิตช์ 8 สถานะของอินเวอร์เตอร์.....	18
2.12 สเปซเวกเตอร์แรงดัน.....	19
2.13 แผนผังการสวิตช์จากการคำนวณเวลาในแต่ละเซกเตอร์.....	21
2.14 สเปซเวกเตอร์แรงดัน v_{sA}, v_{sB}, v_{sC}	22
2.15 เครื่องกลไฟฟ้ากระแสตรงแบบกระตุ้นแยก.....	23
2.16 มอเตอร์เหนี่ยวนำและเครื่องกลไฟฟ้ากระแสตรงในรูปของ Vector Control.....	23
2.17 เฟสเซอร์ไดอะแกรมในการควบคุมเวกเตอร์แบบทางตรง.....	25
2.18 การหาค่าต่างๆโดยวิธี Vector Control โดยใช้แบบจำลองทางเครื่องกลไฟฟ้า.....	26
2.19 โครงสร้างของระบบควบคุมแบบ V/F.....	27
2.20 โครงสร้างของระบบควบคุมแบบ Vector Control.....	28
2.21 โครงสร้างของระบบควบคุมแบบ Sensorless Vector Control.....	28
2.22 การประมาณค่าความเร็วโรเตอร์ด้วยการคำนวณความถี่สลิปโดยตรง.....	29
2.23 การประมาณค่าความเร็วโรเตอร์ด้วยวิธี MRAS.....	30
2.24 การประมาณค่าความเร็วโรเตอร์ด้วย Adaptive Observer.....	31
3.1 การอ้างอิงสเปซเวกเตอร์กระแสและแรงดันบนแกนนิ่งและฟลักซ์โรเตอร์.....	34
3.2 การควบคุมกระแสด้วย PI บนแกนหมุน.....	35
3.3 บล็อกไดอะแกรมระบบควบคุมกระแสบนแกน d	36
3.4 บล็อกไดอะแกรมระบบควบคุมกระแสบนแกน q	36

สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.5	บล็อกไดอะแกรมของระบบควบคุมกระแส.....	37
3.6	แผนภาพโบเดของระบบควบคุมกระแสจากการออกแบบ.....	38
3.7	บล็อกไดอะแกรมของระบบควบคุมความเร็ว.....	39
3.8	แผนภาพโบเดของระบบควบคุมความเร็วจากการออกแบบ.....	40
3.9	บล็อกไดอะแกรมคำนวณฟลักซ์สเตเตอร์ในรูปแบบจำลองแรงดัน (Voltage model) และรูปแบบจำลองกระแส (Current model).....	41
4.1	ระบบจำลองอ้างอิงปรับตัวโดยใช้วิธีฟลักซ์โรเตอร์.....	47
4.2	แสดงความซ้ำซ้อนของระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิด ฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัว.....	48
4.3	ระบบจำลองอ้างอิงแบบปรับตัว (MRAS).....	49
4.4	การบูรณาการระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์ และระบบจำลองอ้างอิงแบบปรับตัว.....	50
4.5	บล็อกไดอะแกรมความผิดพลาดของระบบประมาณความเร็ว.....	51
4.6	ระบบประมาณค่าความเร็วบนแกนนิ่ง.....	52
4.7	ระบบประมาณค่าความเร็วบนแกนหมุน.....	52
4.8	ระบบประมาณค่าความเร็วบนแกนหมุนที่ใช้ฟังก์ชันการโอนย้าย $G'_{22}(s)$	53
4.9	ระบบประมาณค่าความเร็วบนแกนหมุนในลักษณะสัญญาณเข้าออกเดี่ยว (SISO).....	54
4.10	แผนภาพโบเดของ $G'_{22}(s)$ ที่จุดทำงานใดๆ.....	55
4.11	แผนภาพโบเดของระบบประมาณค่าความเร็วจากการออกแบบ.....	55
4.12	โครงสร้างของระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็ว.....	56
4.13	ทดสอบขณะเปลี่ยนแปลงความเร็วของมอเตอร์ เมื่อ $k_p = 600, k_I = 15000$	57
4.14	ทดสอบขณะเปลี่ยนแปลงความเร็วของมอเตอร์ เมื่อ $k_p = 600, k_I = 30000$	58
4.15	ทดสอบขณะเปลี่ยนแปลงความเร็วของมอเตอร์เมื่อ $\omega_c > \omega_{mr}$	59
4.16	ทดสอบขณะป้อนโหลดและปลดโหลดขนาด 2 Nm. อย่างทันทีทันใดที่ความเร็ว คงที่ 1410 rpm.....	60
4.17	ทดสอบขณะกลับทิศความเร็วของมอเตอร์ 1410 ถึง -1410 rpm.....	61
5.1	โครงสร้างของระบบทดสอบจริง.....	62
5.2	วงจรเรียงกระแสแบบบริดจ์และวงจรกรอง.....	63
5.3	ทรานซิสเตอร์โมดูลกำลังแบบ IGBT รุ่น PM30CSJ060.....	64

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.4	วงจรแยกโคคและชุดขับนำทรานซิสเตอร์โมดูลกำลังแบบ IGBT 65
5.5	วงจรควบคุมและหยุดพัลส์สัญญาณขับนำสวิตช์..... 67
5.6	วงจรตรวจจับกระแส..... 68
5.7	วงจรตรวจจับแรงดัน..... 68
5.8	วงจรขยายและยกระดับแรงดัน..... 69
5.9	วงจรป้องกันแรงดันเกิน..... 70
5.10	วงจรป้องกันกระแสเกิน..... 71
5.11	ชุดประมวลผลสัญญาณดิจิทัล รุ่น eZdsp2812 ของบริษัท Texas Instruments 72
5.12	การแสดงผลออกของ TLV5618 ของ Texas Instruments 72
5.13	การแสดงผลออกโมดูล PWM_DAC_DRV ของ eZdsp2812 73
5.14	ไดอะแกรมสถานะและการเรียกใช้อินเตอร์รัปต์..... 74
5.15	ไดอะแกรมเวลาในส่วนของโปรแกรมหลัก..... 75
5.16	ไดอะแกรมเวลาในส่วนของโปรแกรมควบคุมกระแส..... 77
6.1	พล็อตซีโรเตอร์จากแบบจำลองแรงดัน (Voltage model) ที่ความถี่ 5 Hz 78
6.2	พล็อตซีโรเตอร์จากแบบจำลองแรงดัน (Voltage model) ที่ความถี่ 50 Hz 79
6.3	พล็อตซีโรเตอร์จากแบบจำลองกระแส (Current model) ที่ความถี่ 5 Hz 79
6.4	พล็อตซีโรเตอร์จากแบบจำลองกระแส (Current model) ที่ความถี่ 50 Hz 79
6.5	รูปคลื่นของพล็อตซีโรเตอร์จากแบบจำลองกระแส(Current model) ก่อนบูรณาการ เข้ากับระบบประมาณค่าความเร็ว..... 80
6.6	รูปคลื่นของพล็อตซีโรเตอร์จากแบบจำลองกระแส(Current model) หลังบูรณาการ เข้ากับระบบประมาณค่าความเร็ว..... 80
6.7	ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 150 rpm..... 81
6.8	ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 500 rpm..... 81
6.9	ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1000 rpm..... 82
6.10	ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1200 rpm..... 82
6.11	ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1410 rpm..... 83
6.12	ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1500 rpm..... 83
6.13	ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 500 rpm → 1410 rpm 84
6.14	ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 1410 rpm → 500 rpm 84

สารบัญรูป (ต่อ)

รูปที่	หน้า
6.15 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก 1410 rpm → -1410 rpm เมื่อใช้อัตราขยาย $k_p = 600, k_I = 30000$	85
6.16 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก -1410 rpm → 1410 rpm เมื่อใช้อัตราขยาย $k_p = 600, k_I = 30000$	85
6.17 ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 1000 rpm → 1200 rpm	86
6.18 ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 1200 rpm → 1000 rpm	86
6.19 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 300 rpm	87
6.20 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 500 rpm	87
6.21 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 1000 rpm	88
6.22 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 1410 rpm	88
6.23 ผลการทำงานของระบบ เมื่อใช้อัตราขยาย $k_p = 100, k_I = 30000$	89
6.24 ผลการทำงานของระบบ เมื่อใช้อัตราขยาย $k_p = 600, k_I = 30000$	89
6.25 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก 1410 rpm → -1410 rpm เมื่อใช้อัตราขยาย $k_p = 600, k_I = 15000$	90
6.26 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก -1410 rpm → 1410 rpm เมื่อใช้อัตราขยาย $k_p = 600, k_I = 15000$	90

ดังนั้นในงานวิจัยนี้ผู้วิจัยจะทำการแก้ปัญหาที่ตั้งกล่าวมา โดยจะบูรณาการระบบประมาณค่าฟังก์ชันและระบบประมาณค่าความเร็วเข้าด้วยกันเพื่อลดความซ้ำซ้อนของแบบจำลอง และหาเงื่อนไขการออกแบบอัตราขยายปรับตัวที่ง่ายไม่ซับซ้อนและสามารถนำไปใช้ในทางปฏิบัติจริงได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

บูรณาการระบบประมาณค่าฟังก์ชันที่ใช้ตัวสังเกตแบบวงรอบปิดฟังก์ชันสเตเตอร์และระบบประมาณค่าความเร็วเพื่อลดความซ้ำซ้อนของแบบจำลอง ทำการออกแบบอัตราขยายปรับตัวของระบบจำลองอ้างอิงแบบปรับตัวให้มีเงื่อนไขในการออกแบบที่ชัดเจน และออกแบบอัตราขยายระบบควบคุมกระแส ระบบควบคุมความเร็ว เพื่อให้ระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็วโดยรวมมีเสถียรภาพในการทำงาน

1.3 วิธีการวิจัย

1. ศึกษาทฤษฎีการควบคุมมอเตอร์เหนี่ยวนำ 3 เฟส แบบเวกเตอร์ไร้เซนเซอร์วัดความเร็ว
2. ศึกษาระบบประมาณค่าฟังก์ชันที่ใช้ตัวสังเกตแบบวงรอบปิดฟังก์ชันสเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัว
3. ศึกษาวิธีการออกแบบอัตราขยายปรับตัว อัตราขยายระบบควบคุมกระแส และระบบควบคุมความเร็ว
4. จำลองการทำงานด้วยโปรแกรมคอมพิวเตอร์ เพื่อทดสอบตามแนวคิด
5. สร้างและติดตั้งชุดต้นแบบสำหรับการทดสอบ
6. ออกแบบระบบซอฟต์แวร์ พร้อมทดสอบการทำงาน
7. ปรับปรุงและแก้ไขซอฟต์แวร์ที่ได้พัฒนาขึ้น
8. เก็บผลการทดลองเทียบกับการจำลองจากโปรแกรมคอมพิวเตอร์
9. ประเมินผล สรุป และเขียนวิทยานิพนธ์

1.4 ขอบเขตของงานวิจัย

1. บูรณาการระบบประมาณค่าฟังก์ชันที่ใช้ตัวสังเกตแบบวงรอบปิดฟังก์ชันสเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัวเพื่อลดความซ้ำซ้อนของแบบจำลอง
2. ออกแบบอัตราขยายปรับตัวของระบบจำลองอ้างอิงแบบปรับตัวโดยพิจารณาขณะมอเตอร์เปลี่ยนแปลงความเร็ว
3. สร้างชุดต้นแบบสำหรับทดสอบจริง

1.5 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์นี้ได้ศึกษา วิเคราะห์ วิจัย ระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็วที่ใช้การประมาณค่าความเร็วจากระบบจำลองอ้างอิงแบบปรับตัว โดยรายละเอียดเนื้อหาที่จะกล่าวถึงในวิทยานิพนธ์ สามารถแบ่งออกเป็นบทได้ดังนี้

บทที่ 1 กล่าวถึงความเป็นมาของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา วิธีการวิจัย ขอบเขต โครงสร้างวิทยานิพนธ์ และประโยชน์ที่ได้รับจากวิทยานิพนธ์

บทที่ 2 กล่าวถึงแบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำ อินเวอร์เตอร์และสเปซเวกเตอร์แรงดัน เทคนิคสเปซเวกเตอร์พีดับบีวเอ็ม แนวคิดของการควบคุมแบบเวกเตอร์และการควบคุมอินเวอร์เตอร์ประเภทต่างๆ ในอุตสาหกรรม

บทที่ 3 กล่าวถึงการควบคุมแบบเวกเตอร์ การควบคุมกระแสบนแกนหมุนสำหรับการควบคุมแบบเวกเตอร์ แนวทางการออกแบบอัตราขยายตัวควบคุม PI และระบบประมาณค่าฟลักซ์โดยใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์

บทที่ 4 กล่าวถึงระบบจำลองอ้างอิงแบบปรับตัว การบูรณาการตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัว ระบบประมาณค่าความเร็วบนแกนหมุน การออกแบบอัตราขยายปรับตัว และการจำลองระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็วบนระบบจำลองอ้างอิงแบบปรับตัวด้วยโปรแกรมคอมพิวเตอร์

บทที่ 5 กล่าวถึงการสร้างระบบทดสอบจริง ชุดวงจรคอนเวอร์เตอร์ ชุดวงจรอินเวอร์เตอร์ ชุดวงจรขับนำสวิตช์ ชุดวงจรตรวจจับกระแส ชุดวงจรตรวจจับแรงดัน ชุดวงจรขยายและยกระดับแรงดัน ชุดวงจรป้องกันต่าง ๆ ชุดประมวลผลสัญญาณดิจิทัล ชุดการแสดงผลของค่าต่างๆ และ โครงสร้างทางซอฟต์แวร์ที่ใช้ในการควบคุมมอเตอร์

บทที่ 6 กล่าวถึงผลการทดสอบการทำงานของระบบจริง ผลการทดสอบสภาวะอยู่ตัวและสภาวะชั่วคราว การตอบสนองต่อคำสั่งความเร็ว การตอบสนองต่อโหลด ผลของอัตราขยาย k_p และ k_i ที่มีต่อเสถียรภาพของระบบ

บทที่ 7 กล่าวถึงสรุปผลการวิจัยและข้อเสนอแนะ

ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

ภาคผนวก ข. โปรแกรมที่ใช้ในการควบคุมระบบ

1.6 ประโยชน์ที่ได้รับจากวิทยานิพนธ์

1. การบูรณาการระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์และระบบประมาณค่าความเร็ว สามารถลดความซ้ำซ้อนของแบบจำลองได้

2. การออกแบบอัตราขยายปรับตัวของระบบประมาณค่าความเร็ว มีเงื่อนไขที่ชัดเจนและสามารถนำไปใช้ในทางปฏิบัติได้จริง

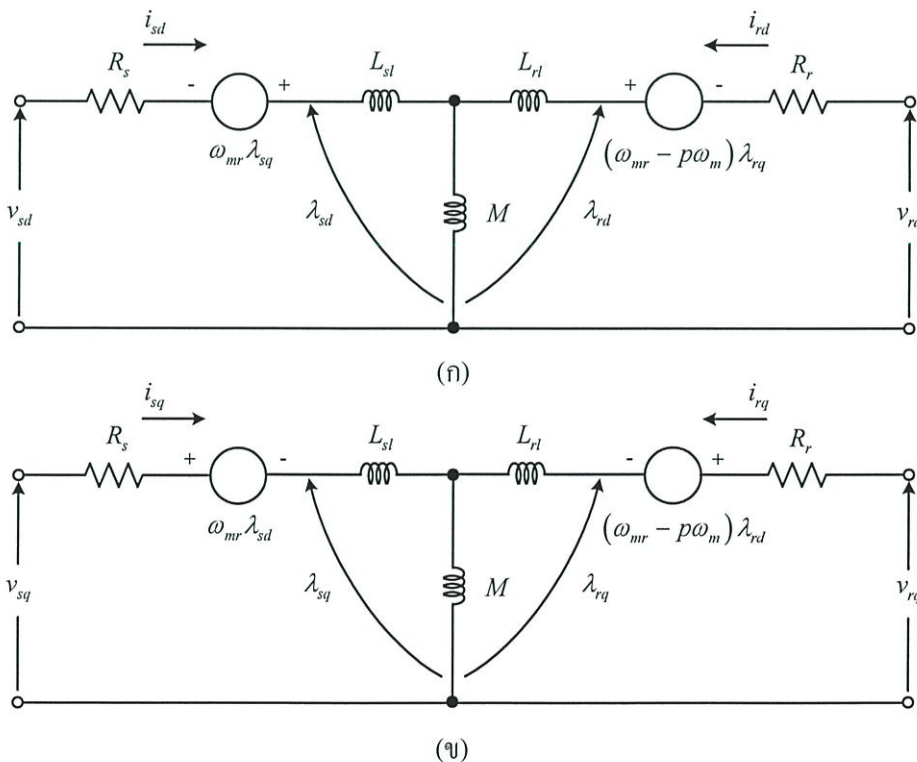
บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

ในบทนี้จะกล่าวถึงทฤษฎีเบื้องต้นที่ใช้ในงานวิจัยและพื้นฐานของระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็ว ซึ่งจะประกอบไปด้วยแบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำ อินเวอร์เตอร์และสเปซเวกเตอร์แรงดัน เทคนิคสเปซเวกเตอร์พีดับบีเอ็ม แนวคิดของการควบคุมแบบเวกเตอร์ และสุดท้ายจะกล่าวถึงการควบคุมอินเวอร์เตอร์ประเภทต่าง ๆ ในอุตสาหกรรม

2.1 แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำ

วงจรมูลทางพลวัต $d-q$ (Dynamic d-q equivalent circuit) ที่มีแกนอ้างอิงบนแกนหมุนซิงโครนัส (Synchronously rotating reference frame) ของเครื่องจักรกลไฟฟ้าเหนี่ยวนำ แสดงได้ดังรูปที่ 2.1 โดยมีแกน d และแกน q แยกอิสระต่อกัน ปริมาณดังกล่าวอยู่ในรูปของไฟตรงและแสดงถึงการป้อนแรงดัน 2 ทิศทาง คือสามารถป้อนแรงดันเข้าด้านสเตเตอร์ (v_{sd}, v_{sq}) และป้อนแรงดันเข้าด้านโรเตอร์ (v_{rd}, v_{rq}) ซึ่งวิธีการป้อนแรงดันแบบดังกล่าวนี้ จะเป็นลักษณะการควบคุมเครื่องจักรกลไฟฟ้าชนิดวาวด์โรเตอร์ (Wound rotor machine) [2]



รูปที่ 2.1 วงจรมูลทางพลวัตที่มีแกนอ้างอิงบนแกนหมุนซิงโครนัส (Synchronously rotating reference frame) (ก) วงจรมูลแกน d (ข) วงจรมูลแกน q

พารามิเตอร์ที่ปรากฏดังรูปที่ 2.1 มีความหมายดังนี้[3]

v_{sd}, v_{sq} : แรงดันด้านสเตเตอร์บนแกน $d - q$ ตามลำดับ

v_{rd}, v_{rq} : แรงดันด้านโรเตอร์บนแกน $d - q$ ตามลำดับ

i_{sd}, i_{sq} : กระแสด้านสเตเตอร์บนแกน $d - q$ ตามลำดับ

i_{rd}, i_{rq} : กระแสด้านโรเตอร์บนแกน $d - q$ ตามลำดับ

$\lambda_{sd}, \lambda_{sq}$: ฟลักซ์สเตเตอร์บนแกน $d - q$ ตามลำดับ

$\lambda_{rd}, \lambda_{rq}$: ฟลักซ์โรเตอร์บนแกน $d - q$ ตามลำดับ

R_s, R_r : ค่าความต้านทานของขดลวดสเตเตอร์และโรเตอร์ตามลำดับ

L_{sl}, L_{rl} : ค่าความเหนี่ยวนำรั่วไหลของขดลวดสเตเตอร์และโรเตอร์ตามลำดับ

M : ค่าความเหนี่ยวนำร่วมระหว่างขดลวดสเตเตอร์และโรเตอร์

ω_{mr} : ความถี่ของฟลักซ์โรเตอร์ (Rotor flux frequency)

ω_m : ความเร็วโรเตอร์ (Rotor speed)

p : จำนวนคู่ของขั้วแม่เหล็ก (Poles pair)

2.1.1 การนำเสนอในรูปแบบสเปซเวกเตอร์

สำหรับรูปแบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำที่ใช้ในการขับเคลื่อนจะใช้ทฤษฎีของสเปซเวกเตอร์ ซึ่งเราจะเขียนปริมาณต่างๆ ของมอเตอร์เหนี่ยวนำให้เป็นเวกเตอร์ได้ดังนี้[3]

$$\vec{v}_s(t) = \frac{2}{3} [v_{sA}(t) + v_{sB}(t)e^{j\theta} + v_{sC}(t)e^{j2\theta}] = v_{s\alpha} + jv_{s\beta} \quad (2.1)$$

$$\vec{i}_s(t) = \frac{2}{3} [i_{sA}(t) + i_{sB}(t)e^{j\theta} + i_{sC}(t)e^{j2\theta}] = i_{s\alpha} + ji_{s\beta} \quad (2.2)$$

$$\vec{i}_r(t) = \frac{2}{3} [i_{rA}(t) + i_{rB}(t)e^{j\theta} + i_{rC}(t)e^{j2\theta}] = i_{r\alpha} + ji_{r\beta} \quad (2.3)$$

$$\vec{\lambda}_s(t) = \frac{2}{3} [\lambda_{sA}(t) + \lambda_{sB}(t)e^{j\theta} + \lambda_{sC}(t)e^{j2\theta}] = \lambda_{s\alpha} + j\lambda_{s\beta} \quad (2.4)$$

$$\vec{\lambda}_r(t) = \frac{2}{3} [\lambda_{rA}(t) + \lambda_{rB}(t)e^{j\theta} + \lambda_{rC}(t)e^{j2\theta}] = \lambda_{r\alpha} + j\lambda_{r\beta} \quad (2.5)$$

เมื่อ $\vec{v}_s(t)$ คือ สเปซเวกเตอร์แรงดันที่ขดลวดสเตเตอร์

$\vec{i}_s(t)$ คือ สเปซเวกเตอร์กระแสที่ขดลวดสเตเตอร์

$\vec{i}_r(t)$ คือ สเปซเวกเตอร์กระแสที่ขดลวดโรเตอร์

$\vec{\lambda}_s(t)$ คือ สเปซเวกเตอร์ของฟลักซ์ที่เกิดขึ้นในขดลวดสเตเตอร์

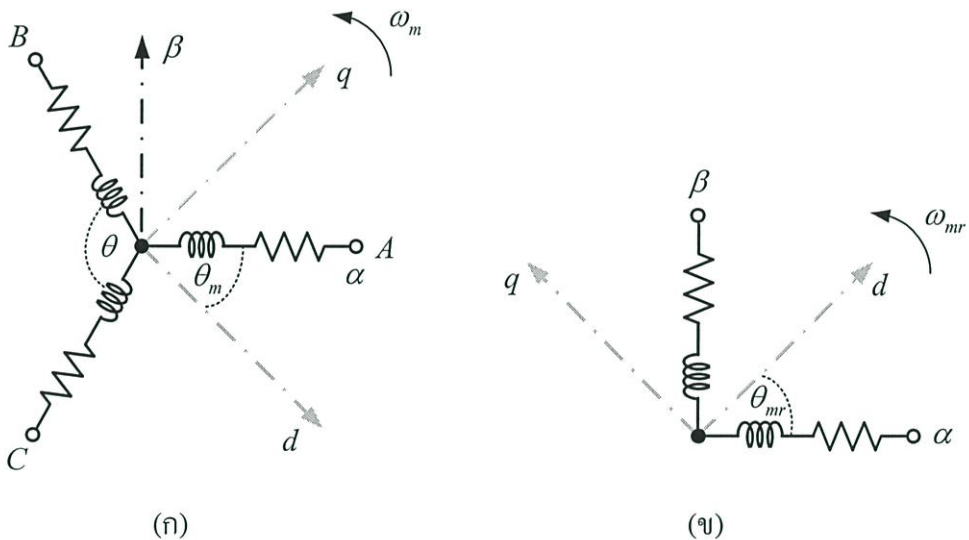
$\vec{\lambda}_r(t)$ คือ สเปซเวกเตอร์ของฟลักซ์ที่เกิดขึ้นในขดลวดโรเตอร์

และ θ คือ $\frac{2\pi}{3}$

ดังนั้นจากโครงสร้างของมอเตอร์ 3 เฟสแบบกรงกระรอก เราสามารถเขียนแทนได้ดังรูปที่ 2.2(ก) [16] โดยที่กระแสและแรงดันมีความสัมพันธ์กันดังสมการ (2.6)

$$\begin{aligned}v_{sA}(t) + v_{sB}(t) + v_{sC}(t) &= 0 \\i_{sA}(t) + i_{sB}(t) + i_{sC}(t) &= 0 \\i_{rA}(t) + i_{rB}(t) + i_{rC}(t) &= 0\end{aligned}\quad (2.6)$$

จากความสัมพันธ์ของแรงดัน กระแส และฟลักซ์ในรูปสเปซเวกเตอร์ดังสมการ (2.1) ถึง (2.5) เราเขียนโครงสร้างของมอเตอร์ใหม่เสมือนว่ามีขดลวดเพียงสองขด ($\alpha - \beta$) ที่ตั้งฉากกัน แสดงได้ดังรูปที่ 2.2(ข) [16]



รูปที่ 2.2 โครงสร้างของมอเตอร์เหนี่ยวนำแบบกรงกระรอก

(ก) แบบจำลองแบบ 3 เฟส

(ข) แบบจำลองแบบ 2 เฟส

จากรูปที่ 2.2(ก) เราทำการแปลงระบบจากสามแกน (ABC) ไปเป็นระบบสองแกน ($\alpha - \beta$) จะได้

$$f_\alpha = f_A \quad (2.7)$$

$$f_\beta = \frac{1}{\sqrt{3}}(f_A + 2f_B) \quad (2.8)$$

โดยที่ f สามารถแทนตัวแปรกระแส แรงดัน หรือฟลักซ์

และการแปลงระบบจากสองแกน ($\alpha - \beta$) ไปเป็นระบบสามแกน (ABC) จะได้

$$f_A = f_\alpha \quad (2.9)$$

$$f_B = \frac{1}{2}(-f_\alpha + \sqrt{3}f_\beta) \quad (2.10)$$

$$f_C = -\frac{1}{2}(f_\alpha + \sqrt{3}f_\beta) \quad (2.11)$$

2.1.2 การย้ายแกนอ้างอิงของสเปซเวกเตอร์ [16]

เนื่องจากเวกเตอร์ของแรงดันสเตเตอร์และกระแสสเตเตอร์อยู่บนแกนอ้างอิงแกนหนึ่ง ($\alpha - \beta$) ซึ่งเป็นแกนอ้างอิงที่ต่างไปจากแกนอ้างอิงของเวกเตอร์ของกระแสโรเตอร์ที่อยู่บนแกนหมุน ($d - q$) ซึ่งหมุนไปตามตำแหน่งของโรเตอร์ (θ_m) ดังนั้นในการดำเนินการทางคณิตศาสตร์ร่วมกันระหว่างสมการด้านสเตเตอร์และโรเตอร์ จึงจำเป็นต้องแสดงเวกเตอร์แรงดันและกระแสอ้างอิงอยู่บนแกนเดียวกัน ดังนั้นเราจะเลือกแกนอ้างอิงสเตเตอร์เป็นแกนอ้างอิงร่วมกันและสามารถแสดงการย้ายแกนอ้างอิงของเวกเตอร์กระแสโรเตอร์ไปยังแกนอ้างอิงสเตเตอร์ได้ตามสมการ(2.12)

$$\begin{bmatrix} i_{r\alpha} \\ i_{r\beta} \end{bmatrix} = \begin{bmatrix} \cos p\theta_m & -\sin p\theta_m \\ \sin p\theta_m & \cos p\theta_m \end{bmatrix} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} \quad (2.12)$$

2.1.3 แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำบนแกนอ้างอิงแกนหนึ่ง ($\alpha - \beta$)

งานวิจัยนี้ได้ใช้มอเตอร์เหนี่ยวนำชนิดกรงกระรอก ดังนั้นจากรูปที่ 2.1 จะเป็นการป้อนแรงดันเข้าด้านสเตเตอร์เพียงด้านเดียว ส่วนด้านโรเตอร์จะไม่มีป้อนแรงดัน ดังนั้นแรงดันด้านโรเตอร์ (v_{rd}, v_{rq}) มีค่าเท่ากับศูนย์ เราสามารถทำการแปลงจากแกนอ้างอิงบนแกนหมุนซิงโครนัส(Synchronously rotating reference frame)ไปสู่แกนอ้างอิงบนแกนนิ่ง (Stationary reference frame) โดยที่ค่า $\omega_{mr} = 0$ [2] และปริมาณดังกล่าวจะอยู่ในรูปของไฟสลับ ซึ่งวงจรสมมูลทางพลวัตแสดงได้ดังรูปที่ 2.3

พารามิเตอร์ที่ปรากฏดังรูปที่ 2.3 มีความหมายดังนี้[3]

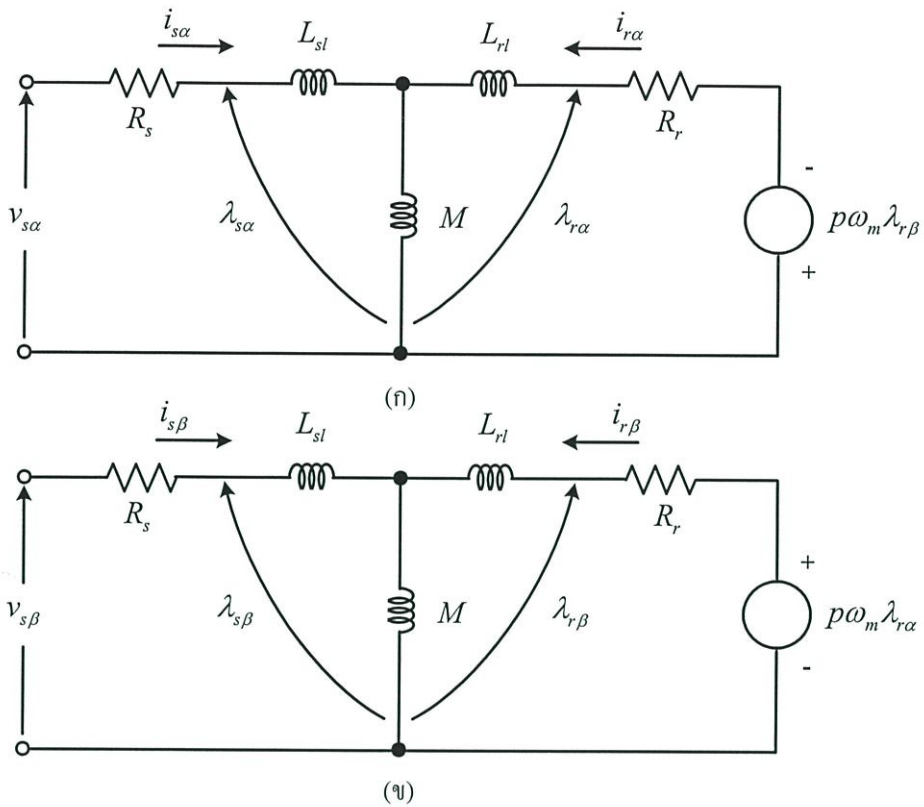
$v_{s\alpha}, v_{s\beta}$: แรงดันด้านสเตเตอร์บนแกน $\alpha - \beta$ ตามลำดับ

$i_{s\alpha}, i_{s\beta}$: กระแสด้านสเตเตอร์บนแกน $\alpha - \beta$ ตามลำดับ

$i_{r\alpha}, i_{r\beta}$: กระแสด้านโรเตอร์บนแกน $\alpha - \beta$ ตามลำดับ

$\lambda_{s\alpha}, \lambda_{s\beta}$: ฟลักซ์สเตเตอร์บนแกน $\alpha - \beta$ ตามลำดับ

$\lambda_{r\alpha}, \lambda_{r\beta}$: ฟลักซ์โรเตอร์บนแกน $\alpha - \beta$ ตามลำดับ



รูปที่ 2.3 วงจรสมมูลทางพลวัตที่มีแกนอ้างอิงบนแกนนิ่ง (Stationary reference frame) สำหรับมอเตอร์เหนี่ยวนำชนิดกรงกระรอก (ก) วงจรสมมูลแกน α (ข) วงจรสมมูลแกน β

รูปที่ 2.3(ก) เราเขียนสมการของแรงดันด้านสเตเตอร์[2] ให้อยู่ในรูปของสเปซเวกเตอร์ (α, β) ได้ดังนี้

$$v_{s\alpha} = i_{s\alpha} R_s + L_{sl} \frac{di_{s\alpha}}{dt} + \frac{d\lambda_{m\alpha}}{dt} \quad (2.13)$$

เมื่อฟลักซ์ในช่องว่างอากาศ (Air gap)[2] มีค่าดังสมการ (2.14) และ (2.15)

$$\lambda_{m\alpha} = \lambda_{s\alpha} - L_{sl} i_{s\alpha} = M (i_{s\alpha} + i_{r\alpha}) \quad (2.14)$$

$$\lambda_{m\beta} = \lambda_{s\beta} - L_{sl} i_{s\beta} = M (i_{s\beta} + i_{r\beta}) \quad (2.15)$$

ฟลักซ์โรเตอร์มีค่าดังสมการ (2.16) และ (2.17)

$$\lambda_{r\alpha} = M i_{s\alpha} + L_r i_{r\alpha} \quad (2.16)$$

$$\lambda_{r\beta} = M i_{s\beta} + L_r i_{r\beta} \quad (2.17)$$

ทำการกำจัดกระแส $i_{r\alpha}$ และ $i_{r\beta}$ จากสมการ(2.14) และ (2.15) กับสมการ(2.16) และ (2.17) ตามลำดับจะได้เป็นดังสมการ (2.18) และ (2.19)

$$\lambda_{r\alpha} = \frac{L_r}{M} \lambda_{m\alpha} - L_{rl} i_{s\alpha} \quad (2.18)$$

$$\lambda_{r\beta} = \frac{L_r}{M} \lambda_{m\beta} - L_{rl} i_{s\beta} \quad (2.19)$$

แทนสมการ (2.18) ใน (2.13) ซึ่งเขียนใหม่ได้เป็นดังสมการ (2.20)

$$v_{s\alpha} = \frac{M}{L_r} \cdot \frac{d\lambda_{r\alpha}}{dt} + \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{s\alpha} \quad (2.20)$$

เมื่อ $\sigma = 1 - M^2/L_r L_s$, $L_{sl} = L_s - M$ และ $L_{rl} = L_r - M$

โดยที่ L_s และ L_r เป็นค่าความเหนี่ยวนำในขดลวดสเตเตอร์และโรเตอร์ตามลำดับ

ดังนั้น

$$\frac{d\lambda_{r\alpha}}{dt} = \frac{L_r}{M} v_{s\alpha} - \frac{L_r}{M} \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{s\alpha} \quad (2.21)$$

และ

$$\frac{d\lambda_{r\beta}}{dt} = \frac{L_r}{M} v_{s\beta} - \frac{L_r}{M} \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{s\beta} \quad (2.22)$$

และจากรูปที่ 2.3(ก) และ 2.3(ข) เขียนสมการของแรงดันด้านโรเตอร์ได้เป็นดังสมการ (2.23) และ (2.24)

$$i_{r\alpha} R_r + \frac{d\lambda_{r\alpha}}{dt} + p\omega_m \lambda_{r\beta} = 0 \quad (2.23)$$

$$i_{r\beta} R_r + \frac{d\lambda_{r\beta}}{dt} - p\omega_m \lambda_{r\alpha} = 0 \quad (2.24)$$

ทำการกำจัดกระแส $i_{r\alpha}$ จากสมการ (2.23) กับ (2.16) จะได้ดังสมการ (2.25)

$$\frac{d\lambda_{r\alpha}}{dt} = -\frac{R_r}{L_r} \lambda_{r\alpha} - p\omega_m \lambda_{r\beta} + \frac{MR_r}{L_r} i_{s\alpha} \quad (2.25)$$

และกำจัดกระแส $i_{r\beta}$ จากสมการ (2.24) กับ (2.17) จะได้ดังสมการ (2.26)

$$\frac{d\lambda_{r\beta}}{dt} = -\frac{R_r}{L_r} \lambda_{r\beta} + p\omega_m \lambda_{r\alpha} + \frac{MR_r}{L_r} i_{s\beta} \quad (2.26)$$

แทนสมการ (2.25) และ (2.26) ลงในสมการ (2.21) และ (2.22) ตามลำดับจะได้ดังสมการ (2.27) และ (2.28)

$$\frac{di_{s\alpha}}{dt} = -\left(\frac{R_s}{\sigma L_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{s\alpha} + \frac{M}{\sigma L_s L_r \tau_r} \lambda_{r\alpha} + \frac{Mp\omega_m}{\sigma L_s L_r} \lambda_{r\beta} + \frac{1}{\sigma L_s} v_{s\alpha} \quad (2.27)$$

$$\frac{di_{s\beta}}{dt} = -\left(\frac{R_s}{\sigma L_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{s\beta} - \frac{Mp\omega_m}{\sigma L_s L_r} \lambda_{r\alpha} + \frac{M}{\sigma L_s L_r \tau_r} \lambda_{r\beta} + \frac{1}{\sigma L_s} v_{s\beta} \quad (2.28)$$

เมื่อ $\tau_r = \frac{L_r}{R_r}$

จากสมการ (2.25) และ (2.26) เขียนใหม่ได้เป็นดังสมการ (2.29) และ (2.30)

$$\frac{d\lambda_{r\alpha}}{dt} = \frac{M}{\tau_r} i_{s\alpha} - \frac{1}{\tau_r} \lambda_{r\alpha} - p\omega_m \lambda_{r\beta} \quad (2.29)$$

$$\frac{d\lambda_{r\beta}}{dt} = \frac{M}{\tau_r} i_{s\beta} + p\omega_m \lambda_{r\alpha} - \frac{1}{\tau_r} \lambda_{r\beta} \quad (2.30)$$

เราสามารถเขียนสมการ (2.27) ถึง (2.30) ให้ดูง่ายขึ้นดังนี้

สมการด้านสเตรเตอร์ :

$$\frac{d}{dt} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \frac{1}{\sigma L_s} \left\{ -\left(R_s + \frac{(1-\sigma)L_s}{\tau_r}\right) I \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \frac{M}{L_r} \left(\frac{1}{\tau_r} - p\omega_m J\right) \begin{bmatrix} \lambda_{r\alpha} \\ \lambda_{r\beta} \end{bmatrix} + \begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} \right\} \quad (2.31)$$

สมการด้านโรเตอร์ :

$$\frac{d}{dt} \begin{bmatrix} \lambda_{r\alpha} \\ \lambda_{r\beta} \end{bmatrix} = \frac{M}{\tau_r} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \left(-\frac{1}{\tau_r} I + p\omega_m J\right) \begin{bmatrix} \lambda_{r\alpha} \\ \lambda_{r\beta} \end{bmatrix} \quad (2.32)$$

สมการแรงบิด :

$$T_m = 1.5p \frac{M}{L_r} (\lambda_{r\alpha} i_{s\beta} - \lambda_{r\beta} i_{s\alpha}) \quad (2.33)$$

2.1.4 แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำบนแกนอ้างอิงแกนหมุน ($d-q$)

เนื่องจากการควบคุมแบบเวกเตอร์จะควบคุมกระแสสร้างฟลักซ์และแรงบิดบนแกนหมุน จึงต้องทำการแปลงแบบจำลองในสมการ (2.31) ถึง (2.33) ไปอยู่บนแกนหมุนโดยแปลงสเปซเวกเตอร์แรงดัน กระแส และฟลักซ์ไปอยู่บนแกนอ้างอิงแกนหมุนจะได้ [16]

กระแสสเตเตอร์บนแกนอ้างอิงแกนหมุน :

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} \cos \theta_{mr} & \sin \theta_{mr} \\ -\sin \theta_{mr} & \cos \theta_{mr} \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad (2.34)$$

แรงดันสเตเตอร์บนแกนอ้างอิงแกนหมุน :

$$\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} = \begin{bmatrix} \cos \theta_{mr} & \sin \theta_{mr} \\ -\sin \theta_{mr} & \cos \theta_{mr} \end{bmatrix} \begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} \quad (2.35)$$

ฟลักซ์โรเตอร์บนแกนอ้างอิงแกนหมุน :

$$\begin{bmatrix} \lambda_{rd} \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \theta_{mr} & \sin \theta_{mr} \\ -\sin \theta_{mr} & \cos \theta_{mr} \end{bmatrix} \begin{bmatrix} \lambda_{r\alpha} \\ \lambda_{r\beta} \end{bmatrix} \quad (2.36)$$

สมการด้านสเตเตอร์ :

$$\frac{di_{sd}}{dt} = \frac{1}{\sigma L_s} \left\{ -\left(R_s + \frac{(1-\sigma)L_s}{\tau_r}\right)i_{sd} + \sigma L_s \omega_{mr} i_{sq} + \frac{M}{L_r \tau_r} \lambda_{rd} + v_{sd} \right\} \quad (2.37)$$

$$\frac{di_{sq}}{dt} = \frac{1}{\sigma L_s} \left\{ -\left(R_s + \frac{(1-\sigma)L_s}{\tau_r}\right)i_{sq} - \sigma L_s \omega_{mr} i_{sd} - \frac{Mp\omega_m}{L_r} \lambda_{rd} + v_{sq} \right\} \quad (2.38)$$

สมการด้านโรเตอร์ :

$$\frac{d\lambda_{rd}}{dt} = \frac{M}{\tau_r} i_{sd} - \frac{1}{\tau_r} \lambda_{rd} \quad (2.39)$$

$$0 = \frac{M}{\tau_r} i_{sq} - \omega_{sl} \lambda_{rd} \quad (2.40)$$

$$\omega_{mr} - p\omega_m = \omega_{sl} = \frac{i_{sq}}{\tau_r i_{sd}} \quad (2.41)$$

สมการแรงบิด :

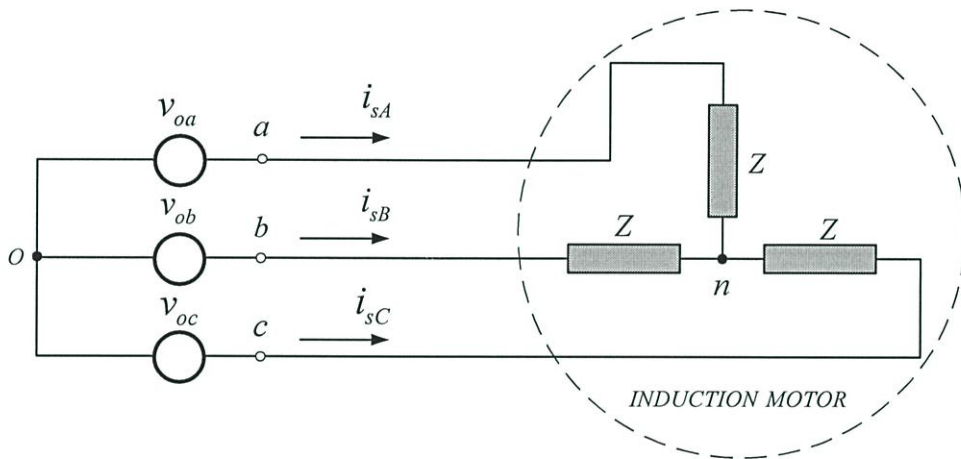
$$T_m = 1.5p \frac{M}{L_r} i_{sd} \lambda_{rd} \quad (2.42)$$

2.2 อินเวอร์เตอร์และสเปซเวกเตอร์แรงดัน [15]

สเปซเวกเตอร์แรงดันใช้สำหรับกำเนิดแรงดันที่จ่ายให้กับขดลวดแต่ละเฟสของมอเตอร์ทำได้โดยการสวิตช์อินเวอร์เตอร์ 3 เฟส ซึ่งเป็นอุปกรณ์ที่ทำหน้าที่เป็นสวิตช์เป็นอิเล็กทรอนิกส์กำลังรูปแบบการสวิตช์อินเวอร์เตอร์ขึ้นอยู่กับค่าที่คำนวณเพื่อได้ค่าที่เหมาะสม

2.2.1 ระบบแหล่งจ่ายแรงดัน 3 เฟส

การกำเนิดสนามแม่เหล็กหมุนจะใช้แหล่งจ่ายแรงดัน 3 เฟส ซึ่งเป็นแหล่งจ่ายไฟอิสระ 3 ชุด แต่ละชุดทำมุมต่างกัน 120 องศา จ่ายให้กับมอเตอร์ดังรูปที่ 2.4



รูปที่ 2.4 แหล่งจ่ายไฟอิสระ 3 ชุด

ในมาตรฐานระบบไฟ 3 เฟส แรงดันลักษณะเป็นไซน์ ถูกจ่ายให้กับมอเตอร์กำเนิดกระแสที่เป็นไซน์มีแรงดันเฟสแสดงได้ดังนี้

$$v_{oa} = \sqrt{2}V \cos(\omega_{mr}t) \quad (2.43)$$

$$v_{ob} = \sqrt{2}V \cos\left(\omega_{mr}t - \frac{2\pi}{3}\right) \quad (2.44)$$

$$v_{oc} = \sqrt{2}V \cos\left(\omega_{mr}t - \frac{4\pi}{3}\right) \quad (2.45)$$

สามารถคำนวณแรงดันเทียบเฟสกับนิวตรอล (v_{an}, v_{bn}, v_{cn}) จากแหล่งจ่ายแรงดันที่จ่ายให้กับมอเตอร์ (v_{oa}, v_{ob}, v_{oc}) โดยกำหนดให้เป็นระบบ 3 เฟสแบบสมมูล ได้ความสัมพันธ์ดังนี้

$$v_{on} = v_{oa} + Z \cdot i_{sA} \quad (2.46)$$

$$v_{on} = v_{ob} + Z \cdot i_{sB} \quad (2.47)$$

$$v_{on} = v_{oc} + Z \cdot i_{sC} \quad (2.48)$$

ดังนั้น

$$3v_{on} = v_{oa} + v_{ob} + v_{oc} = Z(i_{sA} + i_{sB} + i_{sC}) \quad (2.49)$$

และเมื่อ

$$i_{sA} + i_{sB} + i_{sC} = 0 \quad (2.50)$$

จากสมการ (2.49) และ (2.50) แรงดันเฟส $A(v_{an})$ มีค่า

$$v_{an} = v_{on} - v_{oa} = \left(\frac{1}{3}\right)(v_{oa} + v_{ob} + v_{oc}) - v_{oa} \quad (2.51)$$

$$v_{an} = \frac{-2}{3}v_{oa} + \frac{1}{3}v_{ob} + \frac{1}{3}v_{oc} = \frac{2}{3}v_{ao} - \frac{1}{3}v_{bo} - \frac{1}{3}v_{co} \quad (2.52)$$

ดังนั้นในระบบแหล่งจ่ายแรงดัน 3 เฟสแบบสมมูลแรงดันเฟสจะมีค่า

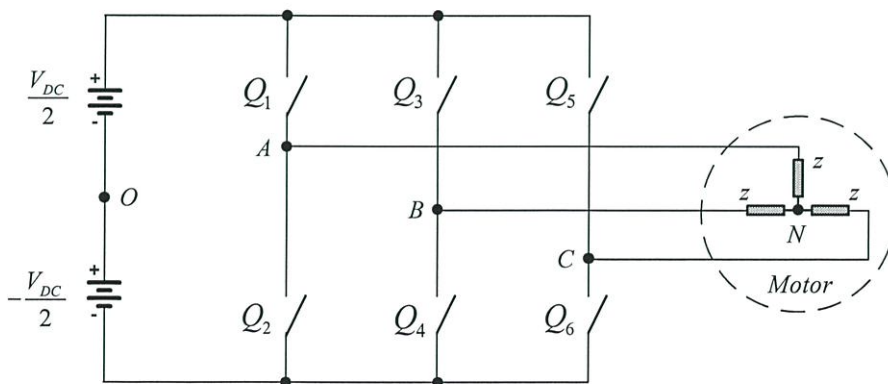
$$v_{an} = \left(\frac{1}{3}\right)(2v_{ao} - v_{bo} - v_{co}) \quad (2.53)$$

$$v_{bn} = \left(\frac{1}{3}\right)(2v_{bo} - v_{ao} - v_{co}) \quad (2.54)$$

$$v_{cn} = \left(\frac{1}{3}\right)(2v_{co} - v_{ao} - v_{bo}) \quad (2.55)$$

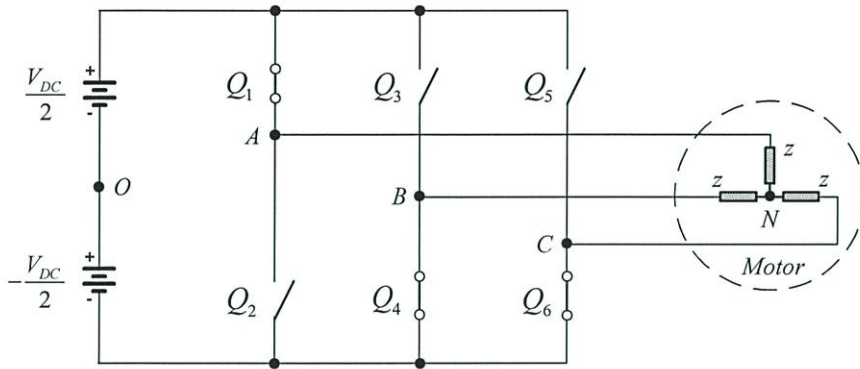
2.2.2 ระบบแหล่งจ่ายไฟแบบอินเวอร์เตอร์ 3 เฟส

อินเวอร์เตอร์ที่นิยมใช้กันเป็นอินเวอร์เตอร์แบบควบคุมแรงดัน (Voltage Source Inverter) โดยมีโครงสร้างประกอบด้วยแหล่งจ่ายแรงดันไฟฟ้ากระแสตรงและสวิตช์ต่อกันดังรูปที่ 2.5



รูปที่ 2.5 แหล่งจ่ายไฟแบบอินเวอร์เตอร์ 3 เฟส

จากรูปที่ 2.6 สวิตช์ที่อยู่ในกิ่งวงจรเดียวกันจะไม่ปิดวงจรพร้อมกัน เช่น เมื่อ Q_1 ปิด Q_2 จะเปิดเสมอ ดังนั้นสวิตช์สามารถปิดเปิดได้ทั้งหมด 8 รูปแบบ โดยกำหนดรูปแบบเรียงลำดับ สวิตช์ตัวบน (1,3,5) เรียงจากเฟส A ; B และ C ตามลำดับ โดยที่ 1 หมายถึงสวิตช์ตัวบนปิด วงจรและสวิตช์ตัวล่างเปิดวงจร และ 0 หมายถึงสวิตช์ตัวบนเปิดวงจรและสวิตช์ตัวล่างปิดวงจร ยกตัวอย่างดังรูปที่ 2.6 เช่น 100 หมายถึงสวิตช์ 1,4,6 ปิดวงจรและสวิตช์ 2,3,5 เปิดวงจร จากรูปแบบการสวิตช์ทั้ง 8 รูปแบบจะได้แรงดันจุดสะเทินเสมือนดังตารางที่ 2.1



รูปที่ 2.6 สถานะอินเวอร์เตอร์ขณะสวิตช์ 100

ตารางที่ 2.1 รูปแบบการสวิตช์และแรงดันเทียบจุดสะเทินเสมือนของอินเวอร์เตอร์

รูปแบบการสวิตช์			แรงดัน		
Q_1	Q_3	Q_5	v_{AO}	v_{BO}	v_{CO}
0	0	0	$-V_{DC}/2$	$-V_{DC}/2$	$-V_{DC}/2$
0	0	1	$-V_{DC}/2$	$-V_{DC}/2$	$V_{DC}/2$
0	1	0	$-V_{DC}/2$	$V_{DC}/2$	$-V_{DC}/2$
0	1	1	$-V_{DC}/2$	$V_{DC}/2$	$V_{DC}/2$
1	0	0	$V_{DC}/2$	$-V_{DC}/2$	$-V_{DC}/2$
1	0	1	$V_{DC}/2$	$-V_{DC}/2$	$V_{DC}/2$
1	1	0	$V_{DC}/2$	$V_{DC}/2$	$-V_{DC}/2$
1	1	1	$V_{DC}/2$	$V_{DC}/2$	$V_{DC}/2$

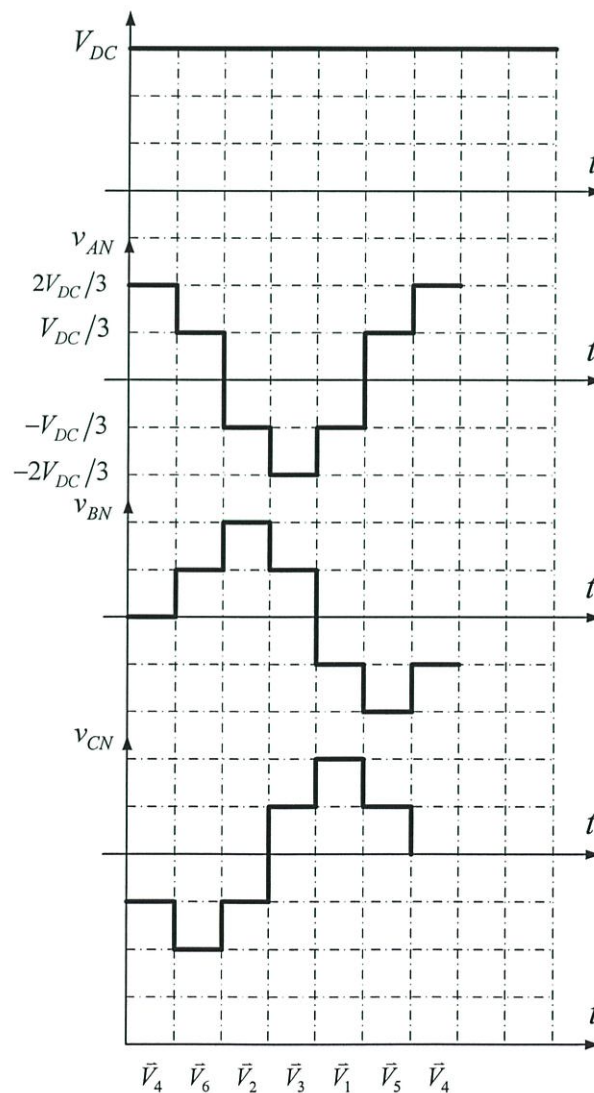
ตารางที่ 2.2 รูปแบบการสวิตช์และแรงดันเทียบจุดนิวตรอลของอินเวอร์เตอร์

Vector	Q_1	Q_3	Q_5	v_{AN}	v_{BN}	v_{CN}
\vec{V}_0	0	0	0	0	0	0
\vec{V}_1	0	0	1	$-V_{DC}/3$	$-V_{DC}/3$	$2V_{DC}/3$

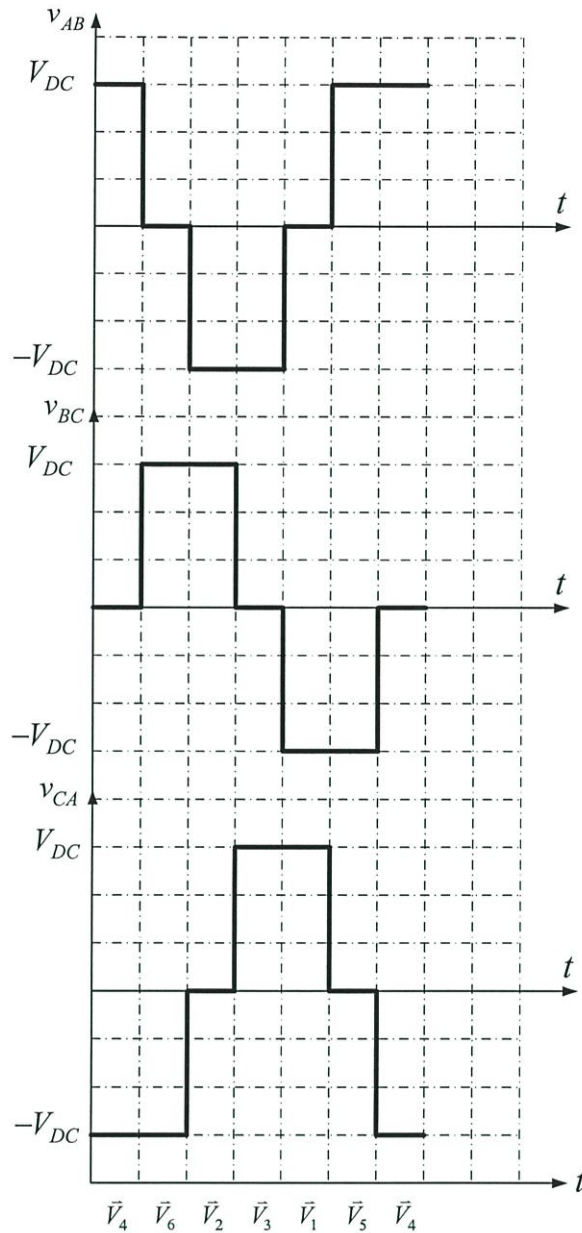
ตารางที่ 2.2 รูปแบบการสวิตช์และแรงดันเทียบจุดนิวตรอลของอินเวอร์เตอร์ (ต่อ)

\vec{V}_2	0	1	0	$-V_{DC}/3$	$2V_{DC}/3$	$-V_{DC}/3$
\vec{V}_3	0	1	1	$-2V_{DC}/3$	$V_{DC}/3$	$V_{DC}/3$
\vec{V}_4	1	0	0	$2V_{DC}/3$	$-V_{DC}/3$	$-V_{DC}/3$
\vec{V}_5	1	0	1	$V_{DC}/3$	$-2V_{DC}/3$	$V_{DC}/3$
\vec{V}_6	1	1	0	$V_{DC}/3$	$V_{DC}/3$	$-2V_{DC}/3$
\vec{V}_7	1	1	1	0	0	0

จากตารางที่ 2.2 ถ้าต้องการให้เกิดแรงดันสามเฟสรูปกึ่งสี่เหลี่ยม (Quasi square) ที่มีเฟสต่างกัน 120 องศา เราสามารถเขียนไดอะแกรมความสัมพันธ์ของแรงดันเฟสและแรงดันไลน์ให้มีลำดับการสวิตช์ $\vec{V}_4 \rightarrow \vec{V}_6 \rightarrow \vec{V}_2 \rightarrow \vec{V}_3 \rightarrow \vec{V}_1 \rightarrow \vec{V}_5 \rightarrow \vec{V}_4$ ได้ดังรูปที่ 2.7 และ 2.8



รูปที่ 2.7 ไดอะแกรมแสดงความสัมพันธ์ของแรงดันเฟส v_{AN}, v_{BN} และ v_{CN}



รูปที่ 2.8 โค้ดแอมแสดงความสัมพันธ์ของแรงดันไลน์ v_{AB} , v_{BC} และ v_{CA}

2.2.3 เวกเตอร์แรงดันในกรอบอ้างอิงสเตเตอร์ (α, β)

การแปลงแรงดัน 3 เฟส เป็นแบบ 2 เฟสในกรอบอ้างอิงสเตเตอร์ (α, β) จากสมการ (2.1) เราเขียนแสดงอีกครั้งในสมการ(2.56) ปริมาณทั้งหมดยังคงอยู่ในรูปไฟสลับ ซึ่งวิธีนี้เรียกกันโดยทั่วไปว่าการแปลงของคลาร์ก (Clarke's Transformation) ดังนั้นจากแรงดัน 3 เฟส (v_{AN}, v_{BN}, v_{CN}) รวมเป็นเวกเตอร์แยกเป็น 2 แกน (α, β) แสดงได้ดังสมการ (2.57)

$$\vec{v}_s(t) = \frac{2}{3}[v_{AN}(t) + av_{BN}(t) + a^2v_{CN}(t)] = v_{s\alpha} + jv_{s\beta} \quad (2.56)$$

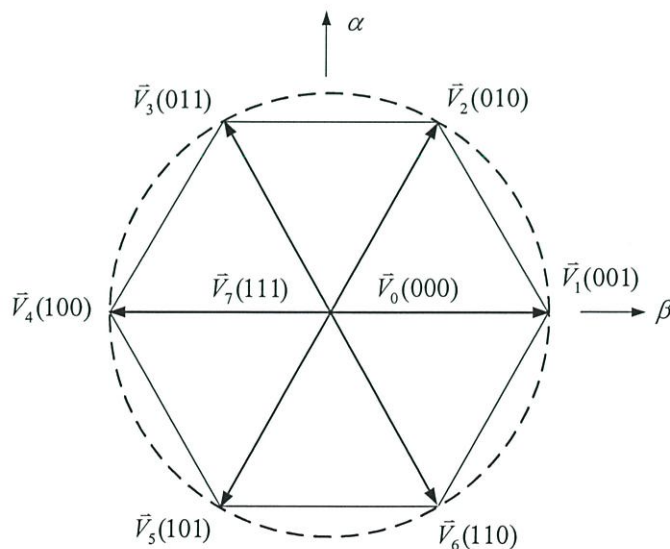
โดยที่ $a = e^{j\theta}$

$$\begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \\ v_o \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_{AN} \\ v_{BN} \\ v_{CN} \end{bmatrix} \quad (2.57)$$

ตารางที่ 2.3 รูปแบบการสวิตช์และค่าแรงดันของอินเวอร์เตอร์ในกรอบอ้างอิงสแตเตอร์

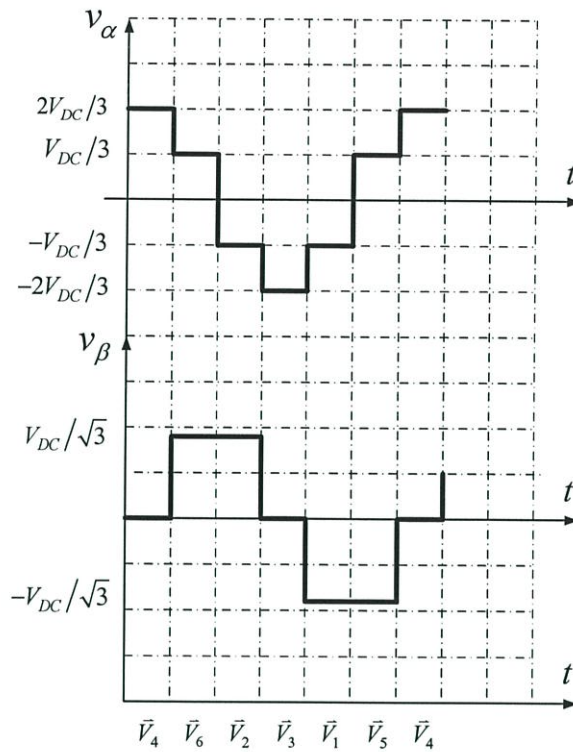
Vector	Q_1	Q_3	Q_5	v_α	v_β	\vec{V}_s
V_0	0	0	0	0	0	\vec{V}_0
V_1	0	0	1	$-V_{DC}/3$	$-V_{DC}/\sqrt{3}$	\vec{V}_1
V_2	0	1	0	$-V_{DC}/3$	$V_{DC}/\sqrt{3}$	\vec{V}_2
V_3	0	1	1	$-2V_{DC}/3$	0	\vec{V}_3
V_4	1	0	0	$2V_{DC}/3$	0	\vec{V}_4
V_5	1	0	1	$V_{DC}/3$	$-V_{DC}/\sqrt{3}$	\vec{V}_5
V_6	1	1	0	$V_{DC}/3$	$V_{DC}/\sqrt{3}$	\vec{V}_6
V_7	1	1	1	0	0	\vec{V}_7

จากสมการ (2.57) ค่าของแรงดัน ($v_{s\alpha}, v_{s\beta}$) จะได้รูปแบบการสวิตช์ (Q_1, Q_3, Q_5) ดังตารางที่ 2.3 และสามารถเขียนเวกเตอร์แรงดันได้ดังรูปที่ 2.9



รูปที่ 2.9 เวกเตอร์แรงดันทั้ง 8 เวกเตอร์

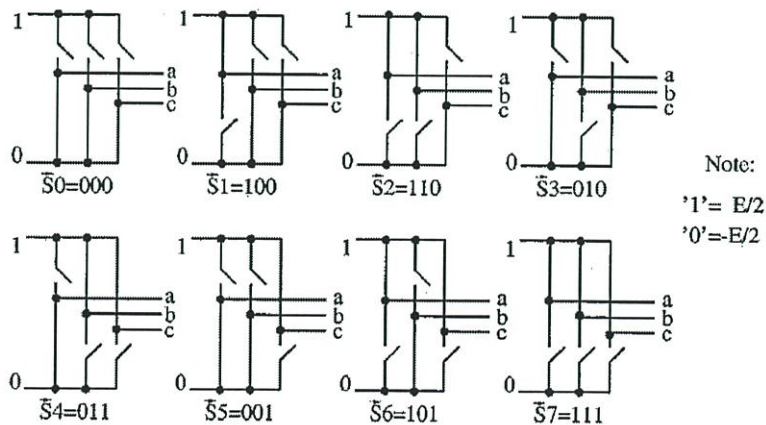
จากตารางที่ 2.3 เขียนไดอะแกรมแสดงความสัมพันธ์ของแรงดัน v_α และ v_β โดยให้มีลำดับการสวิตช์ $\vec{V}_4 \rightarrow \vec{V}_6 \rightarrow \vec{V}_2 \rightarrow \vec{V}_3 \rightarrow \vec{V}_1 \rightarrow \vec{V}_5 \rightarrow \vec{V}_4$ ได้ดังรูปที่ 2.10



รูปที่ 2.10 แสดงความสัมพันธ์ของแรงดัน v_α และ v_β

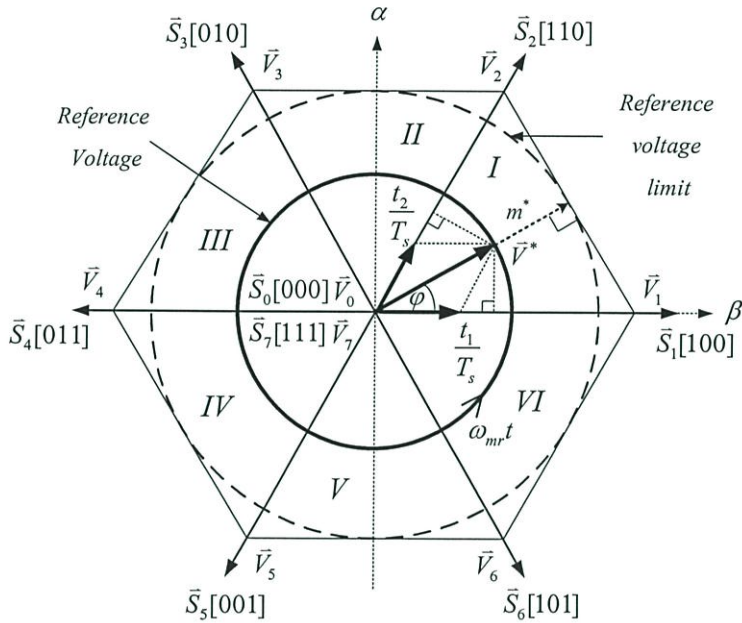
2.3 เทคนิคสเปซเวกเตอร์พีดับบีวเอ็ม [6]

เทคนิคนี้จะใช้การแบ่งการทำงานของสวิตช์กำลังออกเป็น 8 จังหวะ 6 เซกเตอร์ โดยสถานะของสวิตช์จะถูกแทนด้วยเวกเตอร์แรงดัน $\vec{V}_1 - \vec{V}_6$ ซึ่งอาจกล่าวได้ว่าเทคนิคนี้เป็นการนำรูปคลื่นสามเหลี่ยมที่มีแอมพลิจูดเท่ากับ 0.25 เท่าของสัญญาณไซน์ประกอบกับมีความถี่เป็น 3 เท่าของรูปคลื่นไซน์เข้าไปรวมกับรูปคลื่นไซน์ ซึ่งให้แรงดันองค์ประกอบมูลฐานทางด้านเอาต์พุตมีค่าสูงขึ้นอีกทั้งยังใช้ลดค่าการสูญเสียที่เกิดขึ้นเนื่องจากการสวิตช์ได้ด้วย



รูปที่ 2.11 การสวิตช์ 8 สถานะของอินเวอร์เตอร์

จากรูปที่ 2.11 แสดงอินเวอร์เตอร์พีดับบีวเอ็ม 3 เฟส แบบ 2 ชั้น $\bar{S}_1 - \bar{S}_6$ คือสวิตช์กำลัง เมื่อ $\bar{S}_1, \bar{S}_3, \bar{S}_5$ กำหนดสัญลักษณ์เป็น “1” มีค่าเท่ากับ $E/2$ เป็นสวิตช์กำลังกึ่งบน และ $\bar{S}_2, \bar{S}_4, \bar{S}_6$ เป็นสวิตช์กำลังกึ่งล่าง กำหนดสัญลักษณ์เป็น “0” มีค่าเท่ากับ $-E/2$ เวกเตอร์แรงดันเอาต์พุตกำหนดโดยสวิตช์ 8 สถานะ โดยมีเวกเตอร์แรงดันเป็น $\bar{V}_1 = [000], \dots, \bar{V}_7 = [111]$ ตามลำดับ โดยที่ความยาวของเวกเตอร์แรงดัน $\bar{V}_1 \dots \bar{V}_6$ เท่ากับ 1 และความยาวของเวกเตอร์แรงดัน \bar{V}_0 และ \bar{V}_7 เท่ากับ 0 และจาก 8 เวกเตอร์จากสเปซเวกเตอร์แรงดันแสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 สเปซเวกเตอร์แรงดัน

ในสเปซเวกเตอร์แรงดันถูกกำหนดเป็น

$$\bar{V}_1 = -\bar{V}_4, \bar{V}_2 = -\bar{V}_5, \bar{V}_3 = -\bar{V}_6 \quad (2.58)$$

$$\bar{V}_0 = \bar{V}_7 = \bar{0} \quad (2.59)$$

$$\bar{V}_1 + \bar{V}_3 + \bar{V}_5 = \bar{0} \quad (2.60)$$

ในช่วงเวลาการแซมปลิง (Sampling) เวกเตอร์แรงดันเอาต์พุตแสดงได้ดังสมการ (2.61)

$$\bar{V}(t) = \frac{t_0}{T_s} \bar{V}_0 + \frac{t_1}{T_s} \bar{V}_1 + \dots + \frac{t_7}{T_s} \bar{V}_7 \quad (2.61)$$

เมื่อ t_0, t_1, \dots, t_7 คือ ช่วงเวลาที่สวิตช์ทำงาน (Turn-on) ของเวกเตอร์แรงดัน $\bar{V}_1, \dots, \bar{V}_7$
 T_s คือ เวลาการแซมปลิง (Sampling)

ในที่นี้ขอยกตัวอย่างในเซกเตอร์ 1 ซึ่งสามารถเขียนเวกเตอร์แรงดันเอาต์พุตได้ดังนี้

$$\vec{V} = \frac{t_1}{T_s} \vec{V}_1 + \frac{t_2}{T_s} \vec{V}_2 + \frac{t_7}{T_s} \vec{V}_7 + \frac{t_0}{T_s} \vec{V}_0 \quad (2.62)$$

จากรูปที่ 2.12 เมื่อความยาวของ \vec{V}^* คือ $m^* E$ ดังนั้น

$$\frac{m^*}{\sin \frac{2\pi}{3}} = \frac{t_1}{T_s} \frac{1}{\sin \left(\frac{\pi}{3} - \omega_{mr} t \right)} = \frac{t_2}{T_s} \frac{1}{\sin \omega_{mr} t} \quad (2.63)$$

นั่นคือ

$$\frac{t_1}{T_s} = \frac{2}{\sqrt{3}} m^* \sin \left(\frac{\pi}{3} - \omega_{mr} t \right) = \frac{2}{\sqrt{3}} m^* \cos \left(\omega_{mr} t + \frac{\pi}{6} \right) \quad (2.64)$$

$$\frac{t_2}{T_s} = \frac{2}{\sqrt{3}} m^* \sin \omega_{mr} t = \frac{2}{\sqrt{3}} m^* \cos \left(\omega_{mr} t + \frac{3\pi}{2} \right) \quad (2.65)$$

จะได้

$$t_1 = \frac{2}{\sqrt{3}} m^* T_s \cos \left(\omega_{mr} t + \frac{\pi}{6} \right) \quad (2.66)$$

$$t_2 = \frac{2}{\sqrt{3}} m^* T_s \cos \left(\omega_{mr} t + \frac{3\pi}{2} \right) \quad (2.67)$$

$$t_0 = T_s - (t_1 + t_2) \quad (2.68)$$

โดยที่ $m^* = \frac{\hat{V}^*}{\hat{V}_{1sw}}$ และ $0 < m^* < 0.907$

เมื่อ \hat{V}^* คือ ขนาดสูงสุดของแรงดันเฟส

\hat{V}_{1sw} คือ แรงดันมูลฐานสูงสุด ($2V_{DC} / \pi$) ของแรงดันเฟสที่เหลื่อม

ดังนั้นเราสามารถคำนวณหาเวลาใดๆ ในเซกเตอร์ต่างๆ ทั้ง 6 เซกเตอร์ได้ดังตารางที่ 2.4 และแสดงแผนผังการสวิตช์ในแต่ละเซกเตอร์ได้ดังรูปที่ 2.13

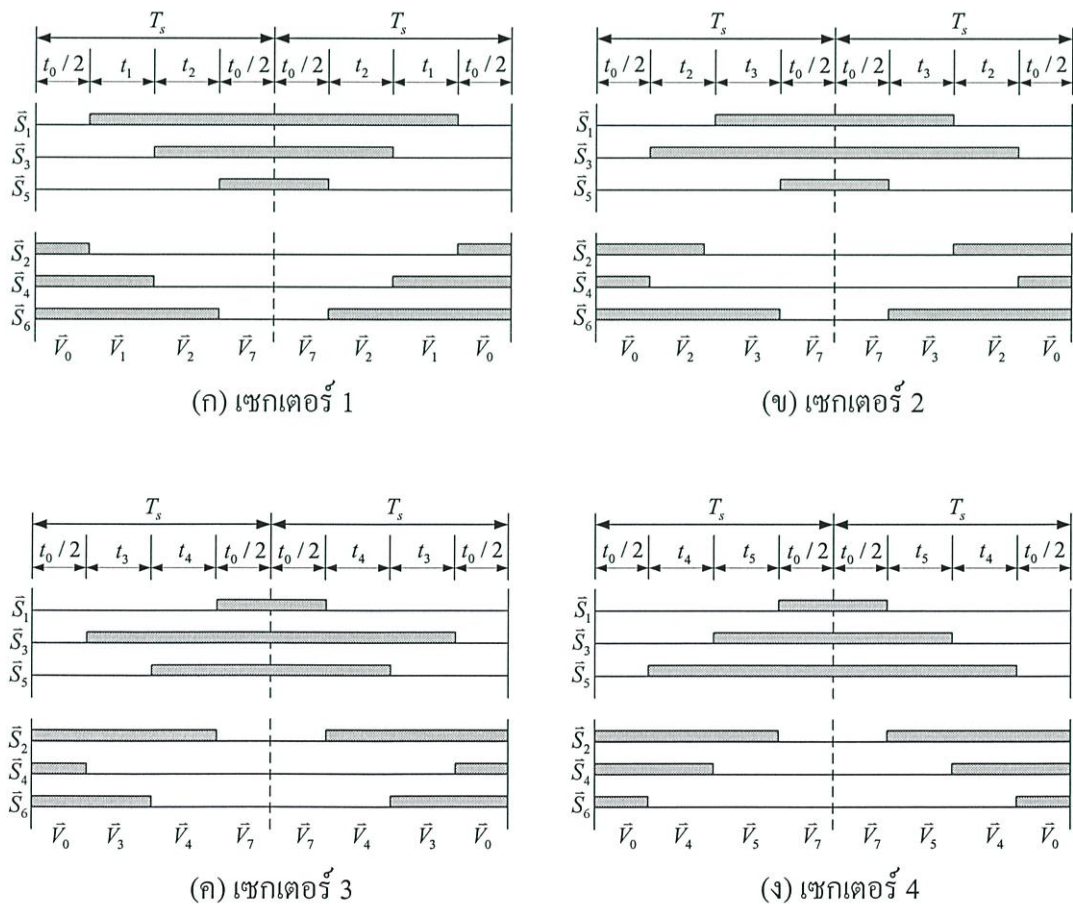
ตารางที่ 2.4 แสดงการคำนวณเวลาในแต่ละเซกเตอร์

SECTOR I ($0 \leq \omega_{mr} t \leq \pi/3$)	SECTOR II ($\pi/3 \leq \omega_{mr} t \leq 2\pi/3$)
$t_1 = \frac{2}{\sqrt{3}} m^* T_s \cos \left(\omega_{mr} t + \frac{\pi}{6} \right)$	$t_2 = \frac{2}{\sqrt{3}} m^* T_s \cos \left(\omega_{mr} t + \frac{11\pi}{6} \right)$
$t_2 = \frac{2}{\sqrt{3}} m^* T_s \cos \left(\omega_{mr} t + \frac{\pi}{6} \right)$	$t_3 = \frac{2}{\sqrt{3}} m^* T_s \cos \left(\omega_{mr} t + \frac{7\pi}{6} \right)$
$t_0 = T_s - (t_1 + t_2)$	$t_0 = T_s - (t_2 + t_3)$

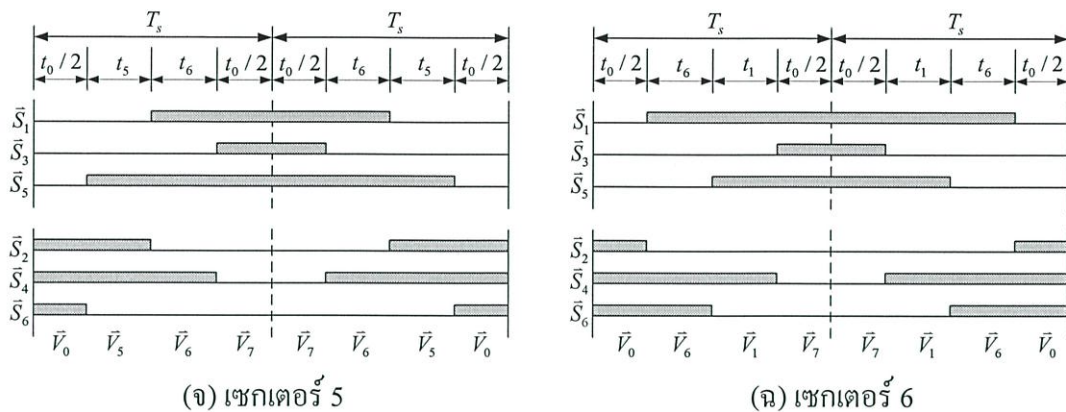
ตารางที่ 2.4 แสดงการคำนวณเวลาในแต่ละเซกเตอร์ (ต่อ)

SECTOR III ($2\pi/3 \leq \omega_{mr}t \leq \pi$)	SECTOR IV ($\pi \leq \omega_{mr}t \leq 4\pi/3$)
$t_3 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{3\pi}{2}\right)$	$t_4 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{7\pi}{6}\right)$
$t_4 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{5\pi}{6}\right)$	$t_5 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{\pi}{2}\right)$
$t_0 = T_s - (t_3 + t_4)$	$t_0 = T_s - (t_4 + t_5)$
SECTOR V ($4\pi/3 \leq \omega_{mr}t \leq 5\pi/3$)	SECTOR VI ($5\pi/3 \leq \omega_{mr}t \leq 2\pi$)
$t_5 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{5\pi}{6}\right)$	$t_6 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{\pi}{2}\right)$
$t_6 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{\pi}{6}\right)$	$t_1 = \frac{2}{\sqrt{3}} m^* T_s \cos\left(\omega_{mr}t + \frac{11\pi}{6}\right)$
$t_0 = T_s - (t_5 + t_6)$	$t_0 = T_s - (t_6 + t_1)$

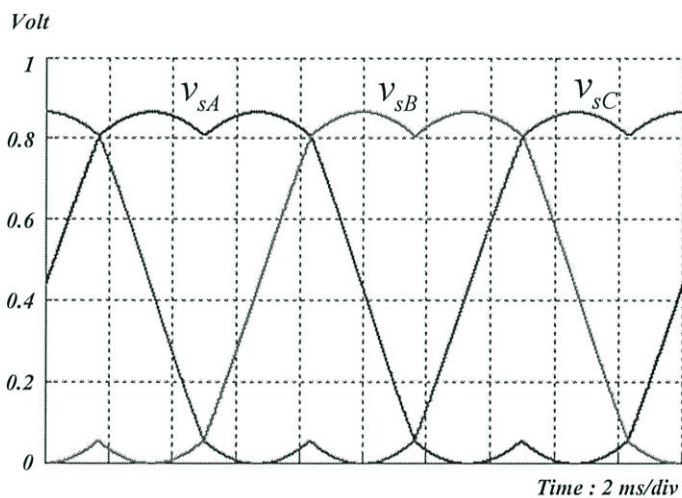
ดังนั้นจากแผนผังการสวิตช์ดังรูปที่ 2.12 แทนค่าในสมการ (2.62) จะได้แรงดันเฟส v_{sA}, v_{sB}, v_{sC} ดังรูปที่ 2.14



รูปที่ 2.13 แผนผังการสวิตช์จากการคำนวณเวลาในแต่ละเซกเตอร์



รูปที่ 2.13 แผนผังการสวิตซ์จากการคำนวณเวลาในแต่ละเซกเตอร์ (ต่อ)



รูปที่ 2.14 สเปซเวกเตอร์แรงดัน v_{sA}, v_{sB}, v_{sC}

2.4 แนวคิดของการควบคุมแบบเวกเตอร์ [1]

การควบคุมมอเตอร์เหนี่ยวนำโดยระบบสเกลาร์ของอินเวอร์เตอร์แบบแหล่งจ่ายแรงดัน (Voltage-fed) และแบบแหล่งจ่ายกระแส (Current-fed) ได้มีการวิเคราะห์กันมานานแล้ว ซึ่งมีข้อสรุปง่ายๆคือ แรงดัน กระแส และความถี่เป็นตัวแปรพื้นฐานในการควบคุมมอเตอร์เหนี่ยวนำ ยกตัวอย่างเช่น ในส่วนของอินเวอร์เตอร์แบบแหล่งจ่ายแรงดัน แรงบิดและฟลักซ์ที่อยู่ในช่องว่างอากาศ (Air gap) เป็นตัวที่ใช้ในการหาค่าของแรงดันและความถี่ ซึ่งแรงบิดและฟลักซ์นี้จะส่งผลให้มอเตอร์เหนี่ยวนำมีการตอบสนองที่ช้าลง ถ้าหากต้องการเพิ่มแรงบิดโดยการเพิ่มความถี่ ฟลักซ์จะพองตัวน้อยลงแต่จะถูกทดแทนโดยวงจรควบคุมฟลักซ์ที่เกิดขึ้นอย่างช้าๆไปพร้อมกับแรงดัน ค่าฟลักซ์ที่เกิดขึ้นช่วงนี้จะไปลดค่าแรงบิดด้วยสลิปและยึดค่าเวลาตอบสนองออกไป ค่าเวลาที่ยึดออกไปนี้จะเท่ากับค่าที่ใช้ในระบบอินเวอร์เตอร์แบบแหล่งจ่ายกระแส

การกำหนดค่าที่เกิดขึ้นก่อนนี้สามารถทดแทนได้โดยใช้วิธีควบคุมแบบเวกเตอร์ (Vector control method) หรือวิธีปรับสนามแม่เหล็ก (Field-oriented control method) ซึ่งสามารถนำไปใช้ได้ทั้งมอเตอร์เหนี่ยวนำและมอเตอร์ซิงโครนัส ในที่นี้เราจะกล่าวถึงวิธีควบคุมแบบเวกเตอร์ การควบคุมเครื่องกลไฟฟ้ากระแสสลับมีวิธีการคล้ายกับเครื่องกลไฟฟ้ากระแสตรงแบบกระตุ้นแยก (Separately excited dc machine) ซึ่งสามารถอธิบายได้ดังรูปที่ 2.15 การทำงานของเครื่องกลไฟฟ้ากระแสตรงแบบกระตุ้นแยก คือ ฟลักซ์คัลล์ของเครื่องกำเนิดชนิดนี้ จะถูกกระตุ้นให้มีอำนาจแม่เหล็กโดยการใช้แหล่งจ่ายไฟฟ้ากระแสตรงจากภายนอกเป็นตัวกระตุ้นในระบบเครื่องกลไฟฟ้ากระแสตรงที่มีการลดค่ากระแสอาร์เมเจอร์และการอิมตัวของสนามแม่เหล็ก เราหาค่าแรงบิดได้จาก

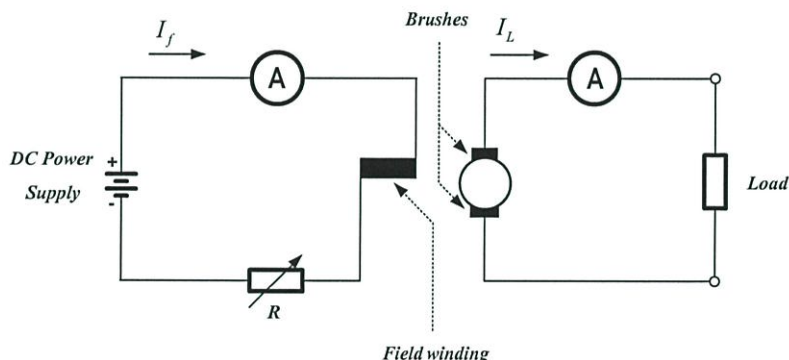
$$T_m = K I_a I_f \quad (2.69)$$

เมื่อ T_m คือ แรงบิดของมอเตอร์

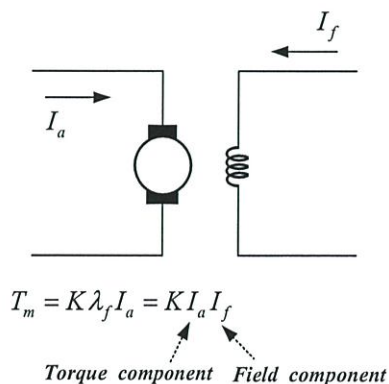
I_a คือ กระแสอาร์เมเจอร์หรือกระแสที่เกิดจากแรงบิด

I_f คือ กระแสฟลักซ์ หรือ กระแสที่เกิดจากฟลักซ์

และ K คือ ค่าคงที่

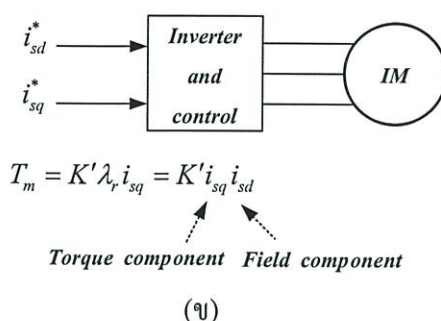


รูปที่ 2.15 เครื่องกลไฟฟ้ากระแสตรงแบบกระตุ้นแยก (Separately excited DC machine)



(ก)

รูปที่ 2.16 มอเตอร์เหนี่ยวนำและเครื่องกลไฟฟ้ากระแสตรงในรูปของ Vector Control



รูปที่ 2.16 มอเตอร์เหนี่ยวนำและเครื่องกลไฟฟ้ากระแสตรงในรูปของ Vector Control (ต่อ)

ในเครื่องกลไฟฟ้ากระแสตรงจะกำหนดค่า I_a และ I_f ให้ทำมุมตั้งฉากกันตามทิศทางของเวกเตอร์ โดยสภาวะปกติจะกำหนดให้กระแสฟิลด์ (I_f) มีค่าคงที่ ส่วนอัตราส่วนของฟลักซ์และแรงบิดจะเปลี่ยนแปลงตามค่ากระแสอาร์เมเจอร์ เมื่อกระแสฟิลด์หรือการตอบสนองของฟลักซ์ไม่ขึ้นกับค่ากระแสอาร์เมเจอร์แรงบิดก็จะยังคงมีค่ามากต่อไปทั้งในการทำงานแบบสถานะชั่วคราว (Transient) และสถานะอยู่ตัว (Steady state) โหมคของการควบคุมแบบนี้สามารถนำไปใช้กับมอเตอร์เหนี่ยวนำได้ในรูปที่ 2.16(จ) แสดงรูปของมอเตอร์เหนี่ยวนำต่อกับอินเวอร์เตอร์และวงจรควบคุมโดยมี i_{sd}^* และ i_{sq}^* ควบคุมทางด้านอินพุต กระแส i_{sd}^* คือส่วนประกอบทางด้านแกนตรง (Direct axis) และกระแส i_{sq}^* คือส่วนประกอบทางด้านแกนตั้งฉากของกระแสทางสเตเตอร์ ซึ่งกระแสทั้งคู่จะจัดให้อยู่บนแกนอ้างอิงบนแกนหมุนซิงโครนัส (Synchronously rotating reference frame) ในการควบคุมแบบเวกเตอร์นั้นกระแส i_{sd} จะมีลักษณะที่เหมือนกับกระแสฟิลด์ (I_f) และกระแส i_{sq} จะมีลักษณะที่เหมือนกระแสอาร์เมเจอร์ (I_a) ของเครื่องกลไฟฟ้ากระแสตรง ดังนั้นแรงบิดจะหาได้จาก

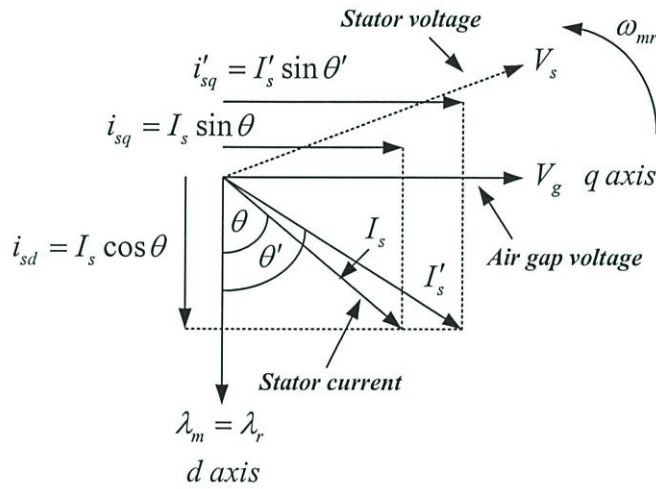
$$T_m = K' \lambda_r i_{sq} = K' i_{sd} i_{sq} \quad (2.70)$$

พื้นฐานในการควบคุมคือการนำกระแส i_{sd} และ i_{sq} มาสร้างให้อยู่ในรูปของเวกเตอร์ ซึ่งหลักการของการควบคุมเวกเตอร์จะอธิบายในรูปที่ 2.17 โดยนำเฟสเซอร์ไดอะแกรมมาช่วยในการหมุนแบบซิงโครนัส หลักการง่ายๆคือเราจะไม่สนใจค่าความเหนี่ยวนำรั่วไหล (Leakage inductance) ของโรเตอร์ เฟสเซอร์ไดอะแกรมจะวาดค่าแรงดันในช่องว่างอากาศ (Air gap voltage, V_g) ให้อยู่บนแกนเดียวกับแกน q กระแสสเตเตอร์จะล้าหลังแรงดัน V_g เท่ากับ $(90-\theta)^\circ$ โดยที่ i_{sq} จะมีเฟสตรงกันกับ V_g และ i_{sd} จะอยู่ในมุมตั้งฉากกับ V_g กระแส i_{sq} คือค่าแอกทิฟหรือส่วนประกอบของแรงบิดของกระแสสเตเตอร์และค่ากำลังแอกทิฟที่ตกคร่อมกับแรงดันช่องว่างอากาศ คือ $V_g i_{sq}$ ส่วนกระแส i_{sd} คือค่ารีแอกทิฟหรือส่วนประกอบของฟลักซ์มีผลต่อฟลักซ์ของช่องว่างอากาศ (λ_m) ของกระแสสเตเตอร์ และค่ากำลังรีแอกทิฟที่ตกคร่อมช่อง

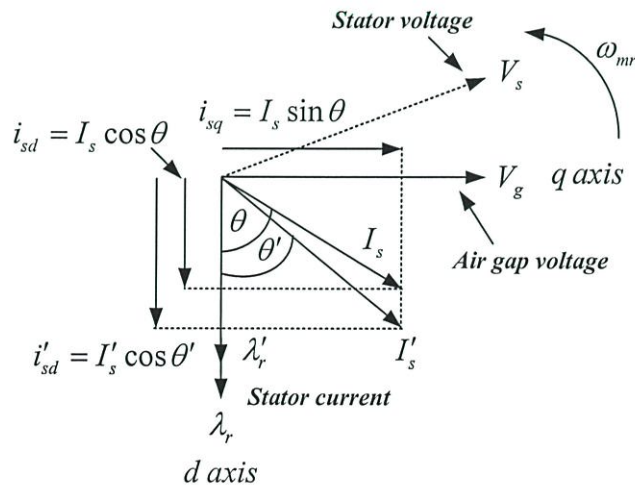
ว่างอากาศ คือ $V_g i_{sd}$ จากเฟสเซอร์ไดอะแกรมค่าแรงบิด (T_m) ตกคร่อมช่องว่างอากาศที่ปรับปรุงแล้วหาได้จาก

$$T_m = K' |\lambda_m| i_{sq} = K' i_{sd} i_{sq} \tag{2.71}$$

เมื่อกระแส i_{sd} และ i_{sq} แสดงดังรูปที่ 2.17 สมการของแรงบิดก็จะเหมือนกับในเครื่องกลไฟฟ้ากระแสตรง ในสภาวะปกติกระแส i_{sd} จะมีค่าคงที่และแรงบิดจะเปลี่ยนแปลงตามค่าของ i_{sq} ที่มีการเปลี่ยนแปลง



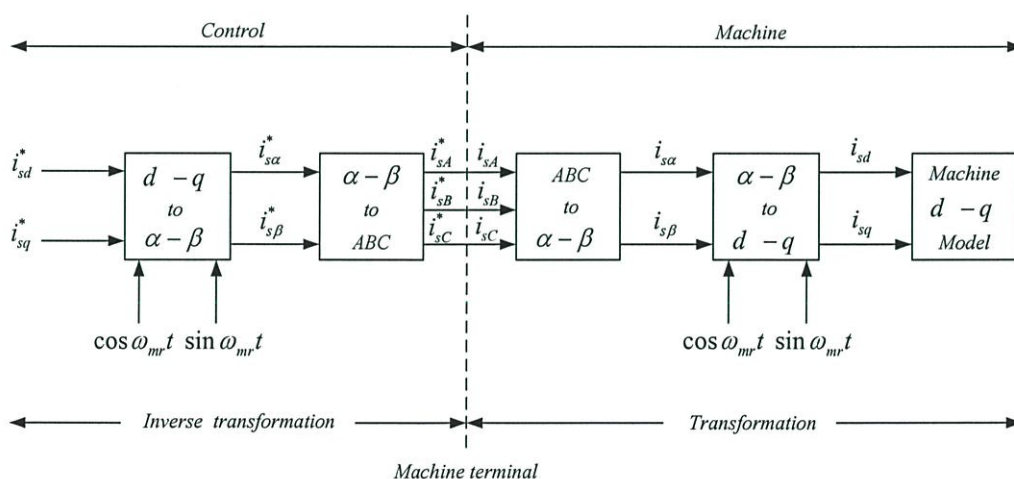
(ก) การเพิ่มของกระแสสร้างแรงบิด



(ข) การเพิ่มของกระแสสร้างสนามแม่เหล็ก

รูปที่ 2.17 เฟสเซอร์ไดอะแกรมในการควบคุมเวกเตอร์แบบทางตรง

การหาค่าต่างๆโดยวิธีเวกเตอร์คอนโทรลโดยใช้แบบจำลองทางเครื่องกลไฟฟ้าจะอธิบายโดยใช้รูปที่ 2.18 เราจะไม่พิจารณาถึงอินเวอร์เตอร์แต่จะสมมติว่ามันได้กำเนิดเฟสของกระแสในทางอุดมคติคือ i_{sA}, i_{sB} และ i_{sC} โดยมีตัวคอนโทรลเลอร์(Controller) สร้างสัญญาณอ้างอิงขึ้นมา แบบจำลองของเครื่องกลไฟฟ้าจะอยู่ทางขวามือ เฟสของกระแส i_{sA}, i_{sB} และ i_{sC} จะถูกแปลงเป็น $i_{s\alpha}$ และ $i_{s\beta}$ โดยการแปลงจาก 3 เฟสเป็น 2 เฟส จากนั้นจะถูกเปลี่ยนไปอยู่บนแกนอ้างอิงบนแกนหมุนซึ่งโคจรด้วยยูนิตเวกเตอร์ (Unit vector) คือ $\cos \omega_{mr}t$ และ $\sin \omega_{mr}t$ ก่อนที่จะถูกเปลี่ยนเป็นแบบจำลองทางเครื่องกลไฟฟ้า จากรูปที่ 2.18 ตัวควบคุมจะสร้าง stage ของการแปลงกลับขึ้นมาสองส่วน แล้ว i_{sd}^* และ i_{sq}^* จะถูกจัดให้อยู่ในรูปตัวแปรของทางด้านเครื่องกลไฟฟ้าที่ตอบสนองกระแส i_{sd} และ i_{sq} ค่าเวกเตอร์ขนาดหนึ่งหน่วยจะเป็นตัวบอกว่าอยู่บนแกนเดียวกันกับเฟสของ λ_m และ i_{sq} อยู่แกนเดียวกับเฟสของ V_g จะสังเกตเห็นว่าในส่วนของการแปลงไปและการแปลงกลับจะไม่รวมเข้าเป็นรูปเดียวกันกับพลวัตอื่น ดังนั้นการตอบสนองของกระแส i_{sd} และ i_{sq} จะเกิดขึ้นเพียงชั่วขณะ โดยทั่วไปในการควบคุมแบบเวกเตอร์มีอยู่ 2 วิธี วิธีแรกเรียกว่าวิธีโดยตรง(Direct method) ส่วนอีกวิธีเรียกว่าวิธีโดยอ้อม(Indirect method) ทั้งสองวิธีจะกล่าวถึงการเกิดสัญญาณ $\cos \omega_{mr}t$ และ $\sin \omega_{mr}t$ ซึ่งเป็นยูนิตเวกเตอร์



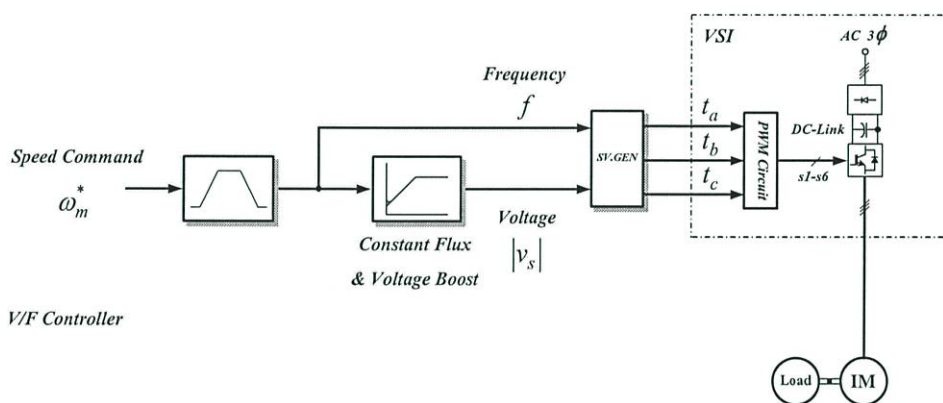
รูปที่ 2.18 การหาค่าต่างๆโดยวิธี Vector Control โดยใช้แบบจำลองทางเครื่องกลไฟฟ้า

2.5 การควบคุมอินเวอร์เตอร์ประเภทต่าง ๆ ในอุตสาหกรรม [16]

ปัจจุบันมีการนำระบบขับเคลื่อนมอเตอร์เหนี่ยวนำด้วยอินเวอร์เตอร์มาใช้อย่างกว้างขวางในระบบขับเคลื่อนในอุตสาหกรรม ทั้งนี้การพัฒนาทางด้านวิธีการควบคุมของอินเวอร์เตอร์มาอย่างต่อเนื่องและมีแนวโน้มการพัฒนาในสองทางใหญ่ ๆ คือการพัฒนาอินเวอร์เตอร์อย่างง่ายและอินเวอร์เตอร์ที่มีสมรรถนะสูง เราสามารถแยกชนิดของอินเวอร์เตอร์ตามลักษณะการควบคุมได้เป็น 3 ประเภทดังต่อไปนี้

2.5.1 การควบคุมอินเวอร์เตอร์แบบ V/F

อินเวอร์เตอร์แบบ V/F ถือเป็นอินเวอร์เตอร์ประเภทแรกที่ถูกนำไปประยุกต์ใช้ในอุตสาหกรรม เนื่องจากเป็นอินเวอร์เตอร์ที่ใช้งานได้ง่ายไม่ซับซ้อน หลักการพื้นฐานของอินเวอร์เตอร์ชนิดนี้คือจะทำการควบคุมแรงดันที่ปลายขั้วให้มีการเปลี่ยนแปลงตามความถี่ในลักษณะเชิงเส้น โดยเราเรียกการควบคุมแบบ V/F ดังแสดงในรูปที่ 2.19 ซึ่งเป็นที่ทราบกันดีว่าการควบคุมแบบนี้มีข้อดีคือ สามารถใช้กับมอเตอร์เหนี่ยวนำทั่วไปโดยไม่ต้องทราบข้อมูลพารามิเตอร์ภายในของมอเตอร์ แต่เนื่องจากการเป็นการควบคุมแบบง่ายๆจึงไม่สามารถให้คุณสมบัติแรงบิดและความเร็วที่ดีที่สุดได้ และยังไม่สามารถควบคุมแรงบิดได้โดยตรง นอกจากนี้ความเร็วที่ได้เปลี่ยนแปลงตามโหลดจึงเกิดความคลาดเคลื่อนไปจากค่าความเร็วที่ตั้งไว้ อย่างไรก็ตามการควบคุมแบบ V/F จะเหมาะกับงานทั่วไปที่ต้องการแค่เพียงปรับความเร็วของมอเตอร์อย่างง่ายๆ เช่น เครื่องสูบน้ำ สายพานลำเลียงในกระบวนการผลิต เป็นต้น

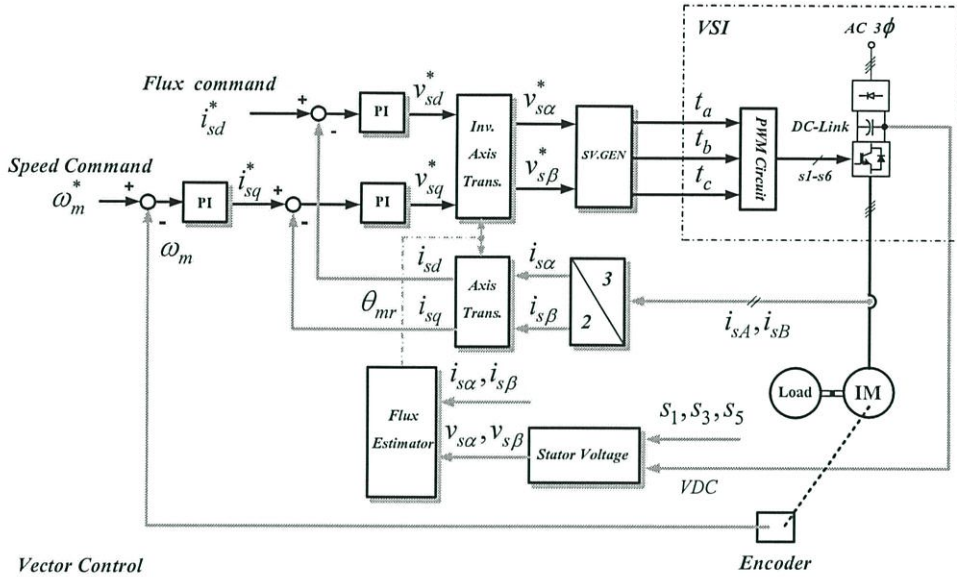


รูปที่ 2.19 โครงสร้างของระบบควบคุมแบบ V/F

2.5.2 การควบคุมอินเวอร์เตอร์แบบเวกเตอร์ (Vector Control : Field - oriented Control)

อินเวอร์เตอร์ชนิดนี้จัดอยู่ในอินเวอร์เตอร์ประเภทที่มีสมรรถนะสูง รูปที่ 2.20 แสดงถึงโครงสร้างของระบบควบคุมแบบเวกเตอร์ซึ่งเป็นวิธีการควบคุมมอเตอร์เหนี่ยวนำในลักษณะคล้ายคลึงกับมอเตอร์กระแสตรง โดยทั่วไประบบควบคุมจะทำการจ่ายกระแสสเตเตอร์ที่มีองค์ประกอบ 2 ส่วนคือ ส่วนสร้างฟลักซ์แม่เหล็กและส่วนสร้างแรงบิด ทั้งนี้การควบคุมทั้งหมดจะกระทำอยู่บนแกนอ้างอิงแกนที่หมุนไปพร้อมกับฟลักซ์เวกเตอร์ทางด้านโรเตอร์ ประเด็นสำคัญในการควบคุมชนิดนี้จึงอยู่ที่ว่า เราสามารถหาตำแหน่งของฟลักซ์เวกเตอร์ได้แม่นยำเพียงไร ซึ่งในทางปฏิบัติเราไม่สามารถวัดค่าฟลักซ์ได้โดยตรง เราจึงจำเป็นต้องใช้การคำนวณ โดยอาศัยแบบจำลองมอเตอร์ในการหาค่าฟลักซ์เวกเตอร์ ดังนั้นข้อดีของระบบนี้จึงอยู่ที่เราจำเป็นต้องทราบค่าพารามิเตอร์ของมอเตอร์อย่างถูกต้องจึงจะได้คุณสมบัติการควบคุมที่ดี ข้อจำกัดอีกประการหนึ่งของระบบก็คือเราจำเป็นต้องทราบข้อมูลความเร็วหรือตำแหน่งโรเตอร์ด้วย จึงทำให้เราไม่สามารถนำไปใช้กับระบบที่ไม่มีเซนเซอร์วัดความเร็วได้ ดังนั้นโดยทั่วไปเรามักจะใช้ระบบ

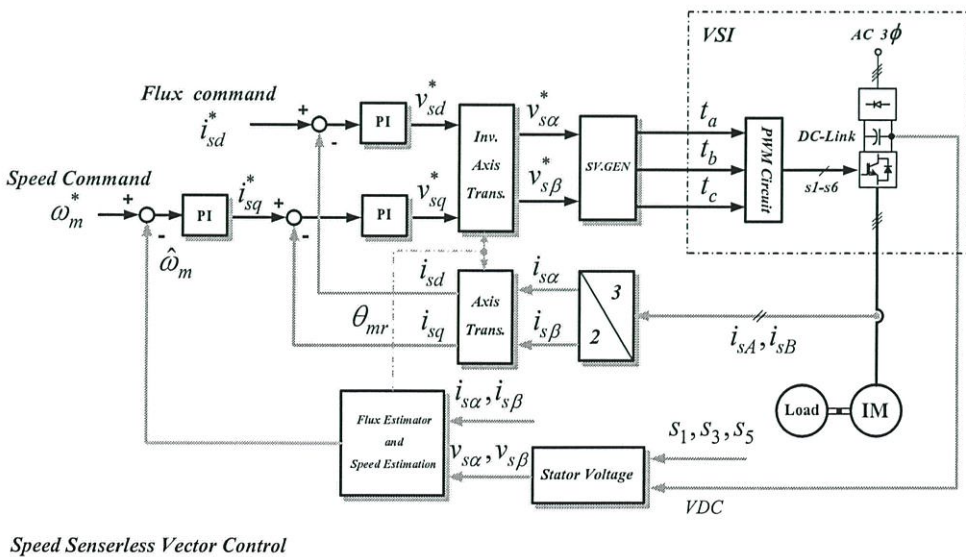
ควบคุมแบบเวกเตอร์นี้กับงานที่ต้องการความแม่นยำและความเร็วในการตอบสนองสูง เช่น ระบบขับเคลื่อนในลิฟท์ ระบบเซอร์โวในโรงงานพลาสติก หรือระบบควบคุมความตึงในโรงงานกระดาษ



รูปที่ 2.20 โครงสร้างของระบบควบคุมแบบ Vector Control

2.5.3 การควบคุมอินเวอร์เตอร์แบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็ว (Speed-Sensorless Vector Control)

ในระบบขับเคลื่อนมอเตอร์เหนี่ยวนำที่ต้องการผลตอบสนองต่อโหลดดีนั้น การควบคุมแบบเวกเตอร์เป็นวิธีหนึ่งที่น่าเชื่อถืออย่างแพร่หลาย โดยอาศัยการควบคุมฟลักซ์และแรงบิดให้มีอิสระต่อกันเสมือนการควบคุมมอเตอร์กระแสตรงแบบกระตุ้นแยกแต่การควบคุมแบบเวกเตอร์



รูปที่ 2.21 โครงสร้างของระบบควบคุมแบบ Sensorless Vector Control

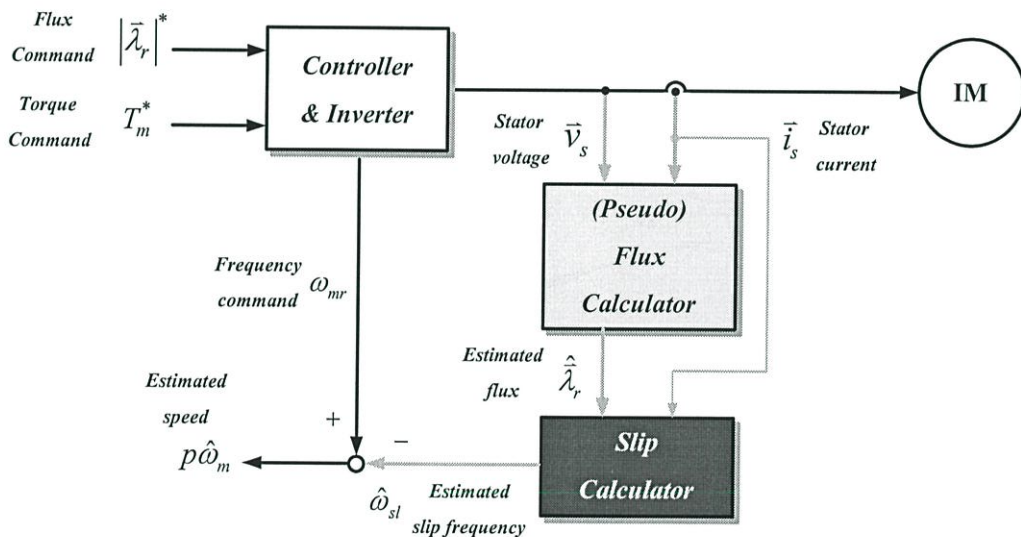
จำเป็นต้องมีการติดตั้งเซ็นเซอร์วัดความเร็ว ดังนั้นการควบคุมความเร็วแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วจึงได้ถูกนำมาใช้ในทางอุตสาหกรรมมากขึ้น รูปที่ 2.21 แสดงถึงโครงสร้างของระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็ว ดังนั้นทำให้มีการศึกษาและวิจัยอย่างแพร่หลาย พร้อมทั้งมีการนำเสนอวิธีการประมาณค่าความเร็วที่แตกต่างกันซึ่งสามารถแบ่งออกได้เป็น

2.5.3.1 วิธีคำนวณค่าโดยตรง (Direct Calculation Methode)

วิธีนี้จะเป็นการคำนวณค่าความเร็วโรเตอร์ ($\hat{\omega}_m$) ผ่านการคำนวณค่าความถี่สลลิปประมาณ ($\hat{\omega}_{sl}$) โดยอาศัยความสัมพันธ์ที่ว่า

$$p\hat{\omega}_m = \omega_{mr} - \hat{\omega}_{sl} \quad (2.72)$$

โครงสร้างโดยพื้นฐานของวิธีนี้สามารถแสดงได้ดังรูปที่ 2.22



รูปที่ 2.22 การประมาณค่าความเร็วโรเตอร์ด้วยการคำนวณความถี่สลลิปโดยตรง

จากสมการ (2.72) สามารถค่าความถี่สลลิป ($\hat{\omega}_{sl}$) โดยตรงจากความสัมพันธ์

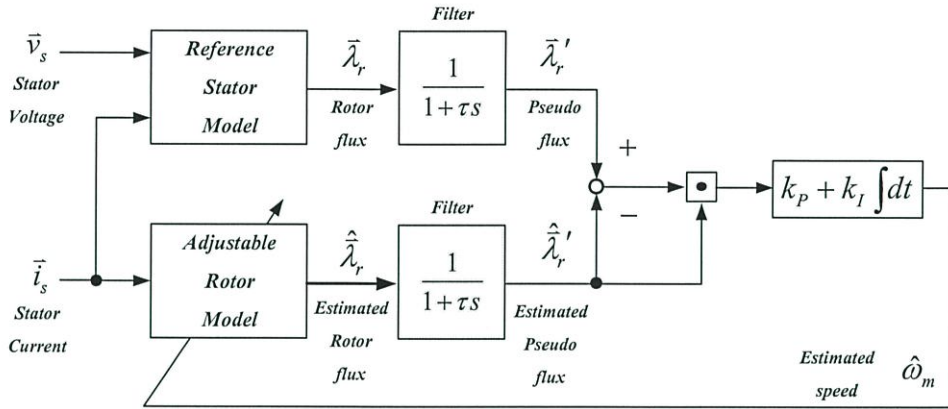
$$\hat{\omega}_{sl} = \frac{MR_r}{L_r} \frac{\vec{i}_s^T J \vec{\lambda}_r}{|\vec{\lambda}_r|^2} \quad (2.73)$$

2.5.3.2 วิธีประมาณค่าความเร็วโดยการปรับแบบจำลอง (Model Adjustment Methode)

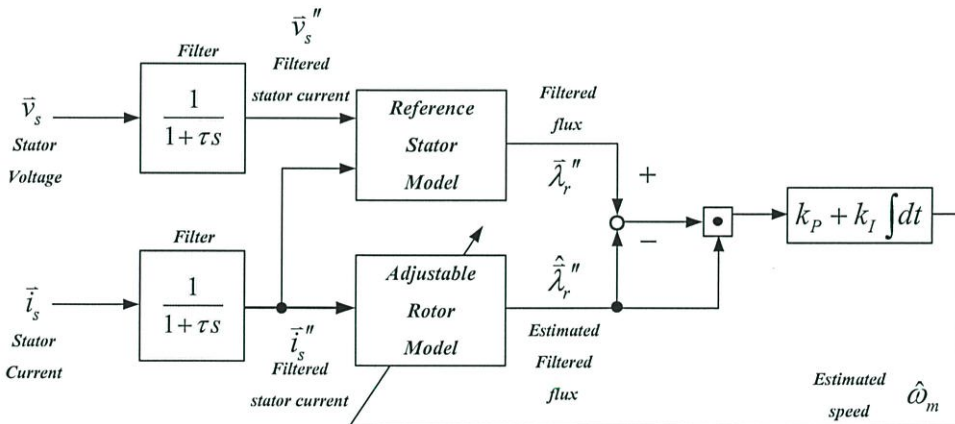
การประมาณค่าความเร็วโรเตอร์โดยอาศัยแบบจำลองทางคณิตศาสตร์ของมอเตอร์เป็นที่นิยมอย่างแพร่หลาย โดยสามารถแบ่งวิธีการประมาณออกเป็นแบบต่างๆ

2.5.3.2.1 ระบบจำลองอ้างอิงแบบปรับตัวได้ (Model Reference Adaptive System, MRAS)

วิธีการนี้ถูกนำเสนอโดย S.Tamai et al.(1997)[11] เป็นอันดับแรก และในลำดับถัดมา โดย C.Schauder (1992)[9] ซึ่งรูปที่ 2.23 แสดงบล็อกไดอะแกรมของการคำนวณค่าความเร็วที่นำเสนอในงานวิจัยทั้งสอง



(ก) ระบบประมาณค่าความเร็วด้วยสัญญาณผิดพลาดของฟลักซ์เทียม



(ข) ระบบประมาณค่าความเร็วที่นำเสนอโดย C.Schauder

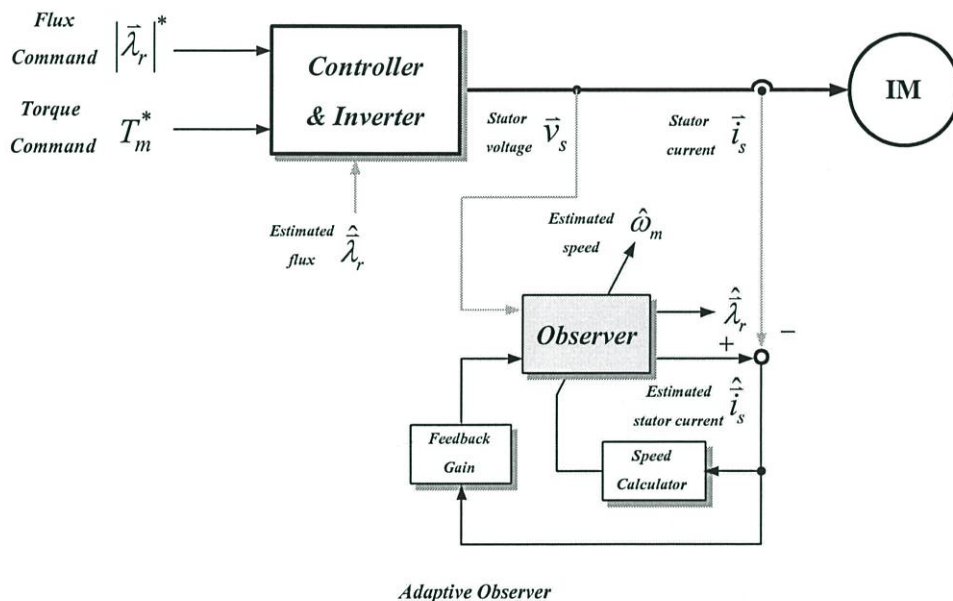
รูปที่ 2.23 การประมาณค่าความเร็วโรเตอร์ด้วยวิธี MRAS

วิธีการประมาณค่านี้จะให้ฟลักซ์โรเตอร์ที่คำนวณจากแบบจำลองด้านสเตเตอร์เป็นสัญญาณอ้างอิงและความเร็วโรเตอร์จะถูกปรับด้วยสัญญาณผิดพลาดระหว่างฟลักซ์โรเตอร์อ้างอิงและฟลักซ์โรเตอร์ที่คำนวณจากแบบจำลองด้านโรเตอร์ วงจรกรองที่เพิ่มเข้าไปทางด้านขาออกของแบบจำลองทั้งสองในรูปที่ 2.23(ก) มีวัตถุประสงค์เพื่อหลีกเลี่ยงปัญหาที่เกิดจากการอินทิเกรตสัญญาณออฟเซตในระบบจริง อย่างไรก็ตามวงจรกรองนี้จะทำให้ฟังก์ชันการโอนย้ายไปข้างหน้าของระบบประมาณขาดคุณสมบัติ Positive real ซึ่งทำให้ลักษณะสมบัติของระบบประมาณไม่สอดคล้องตามเงื่อนไขของทฤษฎี Hyperstability ยังผลให้ไม่สามารถสรุป

เสถียรภาพของระบบประมาณได้ C.Schauder ได้ชี้ให้เห็นถึงประเด็นปัญหาดังกล่าวและได้นำเสนอวิธีการแก้ปัญหาโดยย้ายวงจรรองมาไว้ด้านขาเข้าของแบบจำลองแสดงได้ดังรูปที่ 2.23(ข) ซึ่งทำให้ระบบประมาณมีคุณสมบัติที่สอดคล้องตามเงื่อนไขของทฤษฎี Hyperstability

2.5.3.2.2 ระบบประมาณค่าความเร็วด้วยตัวสังเกตแบบปรับตัว (Adaptive Observer Based Estimation)

หลักการของตัวสังเกตแบบปรับตัว (Adaptive observer) คือใช้แบบจำลองทางคณิตศาสตร์ของมอเตอร์ทำการประมาณค่าตัวแปรสถานะและความเร็วโรเตอร์ไปพร้อมๆ กันดังแสดงในรูปที่ 2.24



รูปที่ 2.24 การประมาณค่าความเร็วโรเตอร์ด้วย Adaptive Observer

ปัญหาที่สำคัญของระบบประมาณค่าความเร็วด้วยตัวสังเกตแบบปรับตัวก็คือการขาดเสถียรภาพของระบบในย่านคิณพลังงาน ทั้งนี้เนื่องจากลักษณะสมบัติของระบบประมาณที่เป็นระบบแบบวงรอบปิด นอกจากนี้ผลกระทบจากค่าความผิดพลาดของพารามิเตอร์ของมอเตอร์ก็เป็นอีกปัจจัยหนึ่งที่ทำให้สมรรถนะของระบบลดถอยลง

บทที่ 3

ระบบควบคุมแบบเวกเตอร์

ในบทนี้จะกล่าวถึงองค์ประกอบโดยรวมของการควบคุมแบบเวกเตอร์ ดังเช่น การควบคุมกระแสบนแกนหมุนสำหรับการควบคุมแบบเวกเตอร์ แนวทางการออกแบบอัตราขยายตัวควบคุม PI และสุดท้ายจะกล่าวถึงระบบประมาณค่าฟลักซ์โดยใช้ตัวสังเกตแบบวงรอบปิด ฟลักซ์สเตเตอร์ที่ใช้สำหรับคำนวณหามุมของฟลักซ์โรเตอร์ ในเบื้องต้นเราจะทำการวิเคราะห์การควบคุมแบบเวกเตอร์จากแบบจำลองทางพลวัตของมอเตอร์เป็นอันดับแรก

3.1 การควบคุมแบบเวกเตอร์ (Vector Control)

เนื่องจากการควบคุมแบบเวกเตอร์จะกระทำบนแกนหมุน จากรูปที่ 2.1 พิจารณาวงจรสมมูลทางพลวัตที่อ้างอิงบนแกนหมุนซิงโครนัส(Synchronously rotating reference frame) ของมอเตอร์เหนี่ยวนำชนิดกรงกระรอกโดยให้แรงดันด้านโรเตอร์ (v_{rd}, v_{rq}) เป็นศูนย์ ดังนั้นเราจึงเขียนสมการของมอเตอร์ทางด้านสเตเตอร์และโรเตอร์ได้ดังนี้

ด้านสเตเตอร์ :

$$v_{sd} = R_s i_{sd} + \frac{d\lambda_{sd}}{dt} - \omega_{mr} \lambda_{sq} \quad (3.1)$$

$$v_{sq} = R_s i_{sq} + \frac{d\lambda_{sq}}{dt} + \omega_{mr} \lambda_{sd} \quad (3.2)$$

ด้านโรเตอร์ :

$$0 = R_r i_{rd} + \frac{d\lambda_{rd}}{dt} - (\omega_{mr} - p\omega_m) \lambda_{rq} \quad (3.3)$$

$$0 = R_r i_{rq} + \frac{d\lambda_{rq}}{dt} + (\omega_{mr} - p\omega_m) \lambda_{rd} \quad (3.4)$$

เมื่อ $\lambda_{sd} = \frac{M}{L_r} \lambda_{rd} + \sigma L_s i_{sd}$ และ $\lambda_{sq} = \frac{M}{L_r} \lambda_{rq} + \sigma L_s i_{sq}$ แทนลงในสมการ (3.1) - (3.2) จะได้

$$v_{sd} = \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{sd} - \sigma L_s \omega_{mr} i_{sq} + \frac{M}{L_r} \cdot \frac{d\lambda_{rd}}{dt} - \omega_{mr} \frac{M}{L_r} \lambda_{rq} \quad (3.5)$$

$$v_{sq} = \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{sq} + \sigma L_s \omega_{mr} i_{sd} + \frac{M}{L_r} \cdot \frac{d\lambda_{rq}}{dt} + \omega_{mr} \frac{M}{L_r} \lambda_{rd} \quad (3.6)$$

และเมื่อ $i_{rd} = \frac{1}{L_r} \lambda_{rd} - \frac{M}{L_r} i_{sd}$ และ $i_{rq} = \frac{1}{L_r} \lambda_{rq} - \frac{M}{L_r} i_{sq}$ แทนลงในสมการ (3.3) - (3.4) จะได้

$$0 = \frac{R_r \lambda_{rd}}{L_r} - \frac{R_r M i_{sd}}{L_r} + \frac{d\lambda_{rd}}{dt} - \omega_{sl} \lambda_{rq} \quad (3.7)$$

$$0 = \frac{R_r \lambda_{rq}}{L_r} - \frac{R_r M i_{sq}}{L_r} + \frac{d\lambda_{rq}}{dt} + \omega_{sl} \lambda_{rd} \quad (3.8)$$

เมื่อ $\omega_{sl} = \omega_{mr} - p\omega_m$

ในขณะที่เกิดฟิวโอเรียนเตชัน (Field Orientation) $\lambda_{rq} = 0$ และ $\frac{d\lambda_{rq}}{dt} = 0$ ดังนั้นจากสมการ(3.5) - (3.8) จะได้

$$v_{sd} = \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{sd} - \sigma L_s \omega_{mr} i_{sq} + \frac{M}{L_r} \cdot \frac{d\lambda_{rd}}{dt} \quad (3.9)$$

$$v_{sq} = \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{sq} + \sigma L_s \omega_{mr} i_{sd} + \omega_{mr} \frac{M}{L_r} \lambda_{rd} \quad (3.10)$$

$$0 = \frac{R_r \lambda_{rd}}{L_r} - \frac{R_r M i_{sd}}{L_r} + \frac{d\lambda_{rd}}{dt} \quad (3.11)$$

$$0 = -\frac{R_r M i_{sq}}{L_r} + \omega_{sl} \lambda_{rd} \quad (3.12)$$

ถ้า λ_{rd} มีค่าคงที่จะทำให้ $\frac{d\lambda_{rd}}{dt} = 0$ ดังนั้นสมการ (3.9) - (3.11) จะได้

$$v_{sd} = \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{sd} - \sigma L_s \omega_{mr} i_{sq} \quad (3.13)$$

$$v_{sq} = \left(R_s + \sigma L_s \cdot \frac{d}{dt} \right) i_{sq} + \omega_{mr} L_s i_{sd} \quad (3.14)$$

$$\lambda_{rd} = M i_{sd} \quad (3.15)$$

นำสมการ (3.15) แทนใน (3.12) จะได้

$$\omega_{sl} = \frac{i_{sq}}{\tau_r i_{sd}} \quad (3.16)$$

เมื่อ $\tau_r = \frac{L_r}{R_r}$

สมการแรงบิดของมอเตอร์ในขณะที่เกิดฟิวโอเรียนเตชัน จะได้

$$T_m = 1.5 p \frac{M}{L_r} i_{sq} \lambda_{rd} \quad (3.17)$$

จากสมการ (3.13) และ (3.14) แสดงความสัมพันธ์ระหว่างแรงดันและกระแสด้านสเตเตอร์ ซึ่งจะเห็นได้ว่าผลการตอบสนองทางพลวัตของกระแสสเตเตอร์นั้นขึ้นอยู่กับขนาดของแรงดันที่ป้อน (v_{sd}, v_{sq}) และแรงเคลื่อนไฟฟ้าเหนี่ยวนำเชื่อมโยงที่เกิดจากกระแสต่างแกนกัน จากสมการ (3.15) และ (3.16) แสดงความสัมพันธ์ของปริมาณต่างๆด้านโรเตอร์ โดยสมการ (3.15) แสดงขนาดของฟลักซ์โรเตอร์ที่ขึ้นอยู่กับกระแสสเตเตอร์บนแกน d ส่วนสมการ (3.16) แสดงถึงความถี่สลิปที่แปรไปตามขนาดของกระแสสเตเตอร์บนแกน $d-q$ และสมการ (3.17) แสดงถึงแรงบิดของมอเตอร์ที่ขึ้นอยู่กับขนาดของฟลักซ์โรเตอร์ที่มีค่าคงที่และกระแสสเตเตอร์บนแกน q

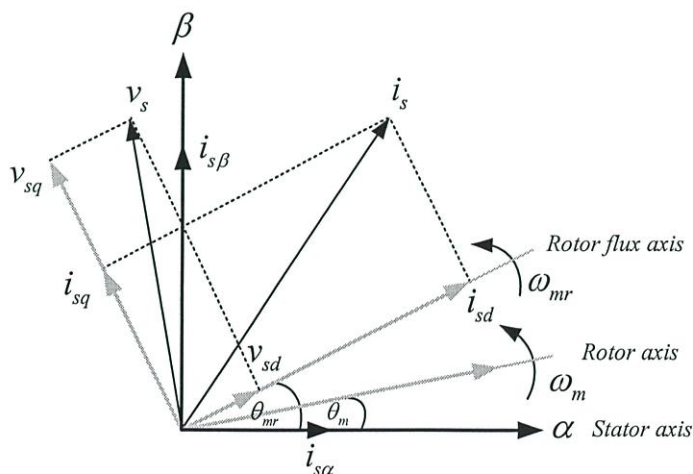
การควบคุมแบบเวกเตอร์จะอาศัยสมการด้านโรเตอร์ในการคำนวณหาฟลักซ์โรเตอร์หรือกระแสโรเตอร์และควบคุมกระแสสเตเตอร์ตามตำแหน่งของฟลักซ์โรเตอร์ โดยการควบคุมขนาดของฟลักซ์โรเตอร์ผ่านกระแสสเตเตอร์บนแกน d และขนาดของแรงบิดผ่านแกน q มุมของฟลักซ์โรเตอร์เป็นค่าอ้างอิงที่สำคัญในการควบคุมแบบเวกเตอร์ ซึ่งเราสามารถหามุมของฟลักซ์โรเตอร์ได้จาก

$$\frac{d\theta_{mr}}{dt} = \omega_{mr} = p\omega_m + \frac{i_{sq}}{\tau_r i_{sd}} = p\omega_m + \omega_{sl} \quad (3.18)$$

และมุมของโรเตอร์คำนวณได้จาก

$$\frac{d\theta_m}{dt} = \omega_m \quad (3.19)$$

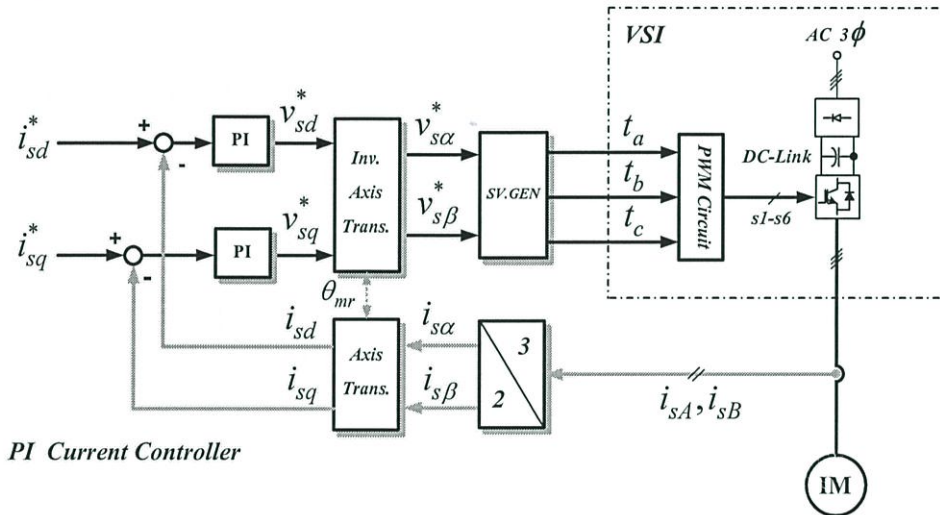
จากสมการ (3.18) มุมของฟลักซ์โรเตอร์ที่ได้นี้เป็นการคำนวณโดยอ้อม สำหรับการคำนวณโดยตรงนั้นจะได้แสดงหัวข้อต่อไป จากรูปที่ 3.1 จะแสดงถึงหลักการควบคุมดังกล่าวข้างต้น เราสามารถสรุปได้ว่าสมรรถนะของการควบคุมแบบเวกเตอร์นั้นขึ้นอยู่กับ การตอบสนองของการควบคุมกระแสเป็นสำคัญ[18]



รูปที่ 3.1 การอ้างอิงสเปซเวกเตอร์กระแสและแรงดันบนแกนนิ่งและฟลักซ์โรเตอร์

3.2 การควบคุมกระแสบนแกนหมุนสำหรับการควบคุมแบบเวกเตอร์

ในการควบคุมกระแสบนแกนหมุนดังรูปที่ 3.2 จะทำการตรวจจับกระแสด้านสเตเตอร์มาทำการแปลงเป็นให้อยู่บนแกนหนึ่ง ($i_{s\beta}, i_{s\alpha}$) จากนั้นแปลงให้อยู่บนแกนหมุน (i_{sd}, i_{sq}) โดยใช้มุมของฟลักซ์โรเตอร์ (θ_{mr}) จะถูกนำไปเปรียบเทียบกับกระแสคำสั่ง (i_{sd}^*, i_{sq}^*) สัญญาณความผิดพลาดนี้จะผ่านตัวควบคุมแบบ PI จะได้แรงดันคำสั่งด้านสเตเตอร์ (v_{sd}^*, v_{sq}^*) จากนั้นแปลงให้อยู่บนแกนหนึ่ง ($v_{s\alpha}^*, v_{s\beta}^*$) แล้วจึงนำไปสร้างสัญญาณ PWM โดยวิธีสเปซเวกเตอร์เพื่อใช้ในการขับนำสวิทช์ต่อไป[17]



รูปที่ 3.2 การควบคุมกระแสด้วย PI บนแกนหมุน

เมื่อพิจารณาฟังก์ชันการโอนย้ายระหว่างกระแสและแรงดันในแต่ละแกนของมอเตอร์นั้น จากสมการ (3.13) และ (3.14) พบว่าเทอมของแรงเคลื่อนไฟฟ้าเหนี่ยวนำย้อนกลับเกิดจากความถี่ของฟลักซ์โรเตอร์บนแกนหมุนและกระแสที่ต่างแกนกันจึงทำให้ไม่สามารถเขียนความสัมพันธ์ได้อย่างอิสระ ดังนั้นจากสมการ (3.20) และ (3.21) จึงเขียนสมการเป็นอีกลักษณะหนึ่งโดยแยกเทอมแรงดันตกคร่อมความต้านทานสเตเตอร์และเทอมแรงเคลื่อนไฟฟ้าเหนี่ยวนำย้อนกลับออกโดยมองว่าเป็นสัญญาณที่เข้ามารบกวนระบบ เพราะฉะนั้นฟังก์ชันการโอนย้ายระหว่างกระแสและแรงดันของมอเตอร์จะเหลือเพียงเทอมความเหนี่ยวนำรั่วไหล[17]

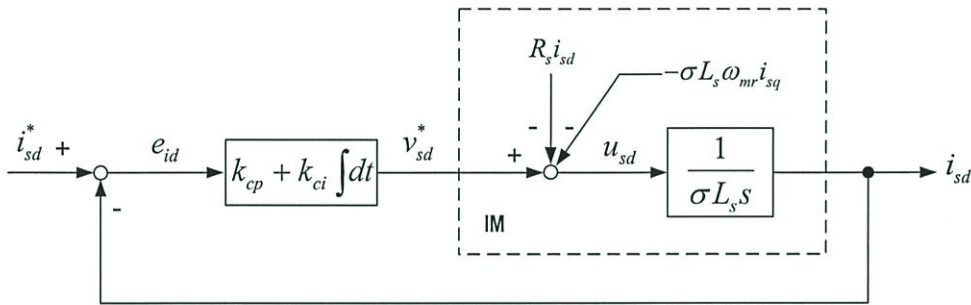
$$u_{sd} = \sigma L_s \frac{di_{sd}}{dt} = [v_{sd} - R_s i_{sd} + \sigma L_s \omega_{mr} i_{sq}] \quad (3.20)$$

$$u_{sq} = \sigma L_s \frac{di_{sq}}{dt} = [v_{sq} - R_s i_{sq} - \omega_{mr} L_s i_{sd}] \quad (3.21)$$

จากสมการ (3.20) และ (3.21) ทำการแปลงลาปลาซ และสามารถแสดงระบบควบคุมกระแสในแกน d และแกน q ได้ดังรูปที่ 3.3 และ 3.4 ตามลำดับ ในที่นี้ถือว่าฟังก์ชันการโอน

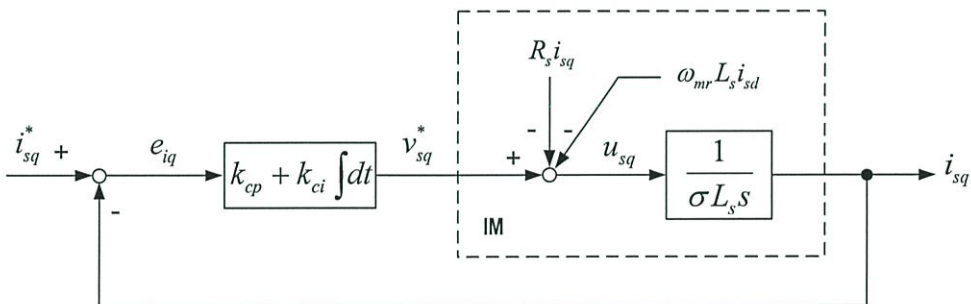
ย้ายของส่วนการสร้างสัญญาณ PWM และอินเวอร์เตอร์มีค่าเท่ากับ 1 สมการ (3.22) และ (3.23) แสดงความสัมพันธ์ระหว่างความผิดพลาดของกระแสกับแรงเคลื่อนไฟฟ้าเหนี่ยวนำย้อนกลับ

$$e_{id} = \left(\frac{s}{\sigma L_s s^2 + s k_{cp} + k_{ci}} \right) \text{Laplace}[-\sigma L_s \omega_{mr} i_{sq}] \quad (3.22)$$



รูปที่ 3.3 บล็อกไดอะแกรมระบบควบคุมกระแสบนแกน d

$$e_{iq} = \left(\frac{s}{\sigma L_s s^2 + s k_{cp} + k_{ci}} \right) \text{Laplace}[\omega_{mr} L_s i_{sd}] \quad (3.23)$$



รูปที่ 3.4 บล็อกไดอะแกรมระบบควบคุมกระแสบนแกน q

ในขณะที่เปลี่ยนแปลงความเร็วจะทำให้ทอมแรงเคลื่อนไฟฟ้าเหนี่ยวนำย้อนกลับมีลักษณะเป็นแรมป์ซึ่งจะทำให้ค่าความผิดพลาดของกระแสในขณะอยู่ตัวไม่เท่ากับศูนย์จึงส่งผลให้ไม่สามารถควบคุมแรงบิดได้ ดังนั้นจากสมการ (3.22) และ (3.23) ใช้ทฤษฎีบทค่าสุดท้าย (Final value theorem) หากความผิดพลาดจะได้อ

แกน d :

$$e_{ss} = \lim_{s \rightarrow 0} \left[s \cdot \left(\frac{s}{\sigma L_s s^2 + s k_{cp} + k_{ci}} \right) \cdot \frac{K_d}{s^2} \right]$$

$$e_{ss} = \frac{K_d}{k_{ci}} \quad (3.24)$$

แกน q :

$$e_{ss} = \lim_{s \rightarrow 0} \left[s \cdot \left(\frac{s}{\sigma L_s s^2 + s k_{cp} + k_{ci}} \right) \cdot \frac{K_q}{s^2} \right]$$

$$e_{ss} = \frac{K_q}{k_{ci}} \quad (3.25)$$

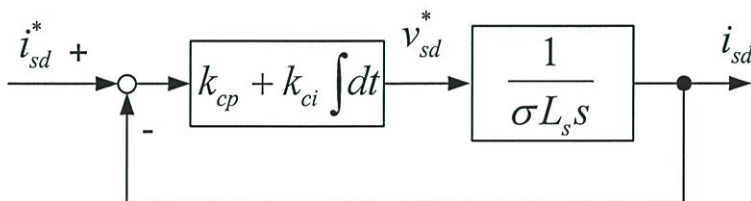
โดยที่ K_d และ K_q เป็นค่าคงที่ของแกน d และ q และแรงเคลื่อนไฟฟ้าเหนี่ยวนำย้อนกลับคือ $\frac{K_d}{s^2}$ และ $\frac{K_q}{s^2}$ จากสมการ (3.24) และ (3.25) จะสังเกตเห็นว่าถ้าใช้อัตราขยาย k_{ci} มีค่าสูงจะช่วยลดค่าความผิดพลาดของกระแสในสภาวะอยู่ตัวได้

3.3 แนวทางการออกแบบอัตราขยายตัวควบคุม PI

โครงสร้างของระบบควบคุมแบบเวกเตอร์ในงานวิจัยนี้ดังแสดงได้ดังรูปที่ 2.21 ซึ่งเป็นระบบควบคุมแบบเวกเตอร์โดยอ้อม (Indirect vector control) และเพื่อให้ระบบควบคุมแบบเวกเตอร์ทำงานได้อย่างสมบูรณ์และมีเสถียรภาพจึงได้นำเสนอแนวทางการออกแบบระบบควบคุมในแต่ละส่วนได้ดังนี้

3.3.1 การออกแบบระบบควบคุมกระแส

จากสมการ (3.13) และ (3.14) ได้ละเอียดแรงดันเชื่อมโยงระหว่างแกนและแรงดันตกคร่อมความต้านทานสเตเตอร์ซึ่งจะได้ระบบควบคุมกระแสที่ใช้ในการออกแบบดังรูปที่ 3.5 และฟังก์ชันการโอนย้ายวงรอบเปิดของระบบควบคุมกระแสเป็นดังสมการ (3.26)



รูปที่ 3.5 บล็อกไดอะแกรมของระบบควบคุมกระแส

$$G_c(s) = \frac{1}{\sigma L_s s} \left(k_{cp} + \frac{k_{ci}}{s} \right) \quad (3.26)$$

จากสมการ (3.27) แสดงเงื่อนไขอัตราขยายวงรอบเปิดที่ความถี่ตัดข้าม ω_o (Crossover frequency) ฟังก์ชันการโอนย้ายของกระแสถือว่าเป็นส่วนในการสร้างสัญญาณ PWM โดยจะกำหนดอัตราขยายเท่ากับ 1 และสมการ (3.28) แสดงค่าความถี่หักมุม ω_c (Corner frequency) ของตัวควบคุม PI ต้องน้อยกว่าความถี่ตัดข้าม

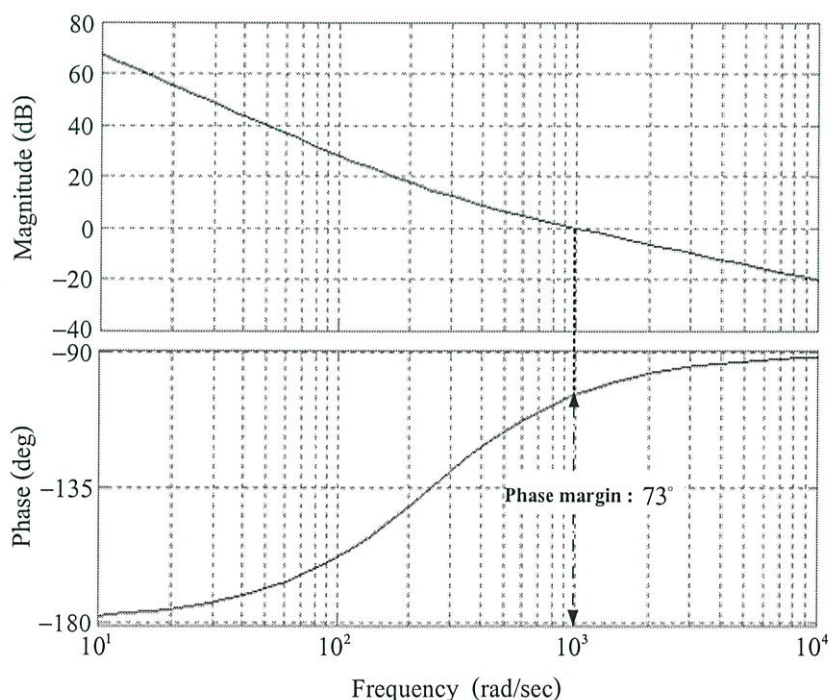
$$|G_C(s)|_{s=j\omega_o} = 1 \quad (3.27)$$

$$\omega_c = \frac{k_{ci}}{k_{cp}} < \omega_o \quad (3.28)$$

ในระบบควบคุมกระแสจะเป็นการควบคุมวงรอบในซึ่งต้องมีการตอบสนองที่เร็วจึงต้องเลือกความถี่ตัดข้ามที่มีค่ามากแต่ไม่ควรมากเกินไปเนื่องจากจะส่งผลกระทบต่อกรขยายสัญญาณรบกวนในระบบได้ งานวิจัยนี้ได้เลือกความถี่ตัดข้าม $\omega_o = 1000$ rad/s และการเลือกค่าความถี่หักมุมจะพิจารณาถึงเสถียรภาพเป็นหลักซึ่งต้องเพื่อเฟสให้เพียงพอกับระบบ จึงได้เลือก $\omega_c = k_{ci}/k_{cp} = 250$ rad/s [17] ดังนั้นเราแทนค่าพารามิเตอร์ตามตารางที่ 4.1 จะได้อัตราขยายของตัวควบคุม PI บนแกน d และ q คือ

$$k_{cp} = 20 \quad \text{และ} \quad k_{ci} = 5000$$

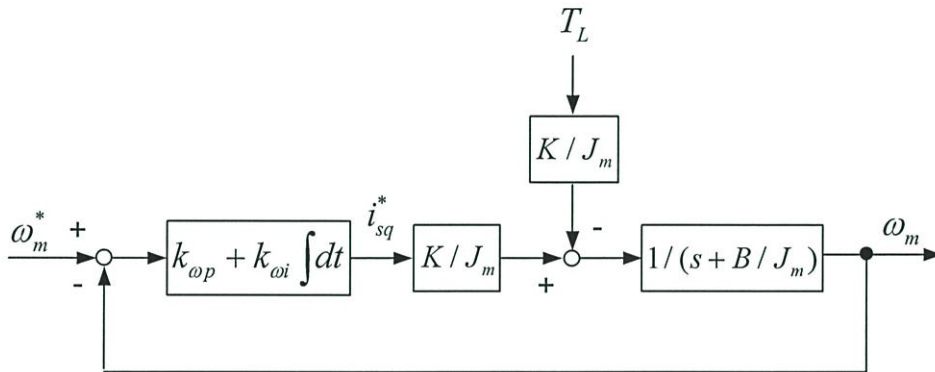
เราแสดงแผนภาพโบเดจากการออกแบบได้ดังรูปที่ 3.6 ซึ่งพบว่ามีส่วนเฟสเพื่อเฟส 73 องศาซึ่งมากเพียงพอสำหรับเสถียรภาพการควบคุม



รูปที่ 3.6 แผนภาพโบเดของระบบควบคุมกระแสจากการออกแบบ

3.3.1 การออกแบบระบบควบคุมความเร็ว

จากรูปที่ 3.7 เราสามารถหาฟังก์ชันการโอนย้ายวงรอบเปิดของระบบควบคุมความเร็วได้ ดังสมการ (3.29) โดยแสดงเงื่อนไขอัตราขยายวงรอบเปิดที่ความถี่ตัดข้ามมีค่าเท่ากับ 1 และ สมการ (3.30) แสดงค่าความถี่หักมุมของตัวควบคุม PI ต้องน้อยกว่าความถี่ตัดข้าม ในระบบ ควบคุมความเร็วจะเป็นการควบคุมวงรอบนอก ดังนั้นจึงไม่จำเป็นต้องเลือกความถี่ตัดข้ามที่มีค่า สูงมากนัก งานวิจัยนี้จึงเลือกความถี่ตัดข้าม $\omega_o = 50 \text{ rad/s}$ ซึ่งใช้งานกันอยู่ในอุตสาหกรรม[14]



รูปที่ 3.7 บล็อกไดอะแกรมของระบบควบคุมความเร็ว

$$\left| \left(k_{\omega p} + \frac{k_{\omega i}}{s} \right) \frac{K}{J_m s} \right|_{s=j\omega_o} = 1 \quad (3.29)$$

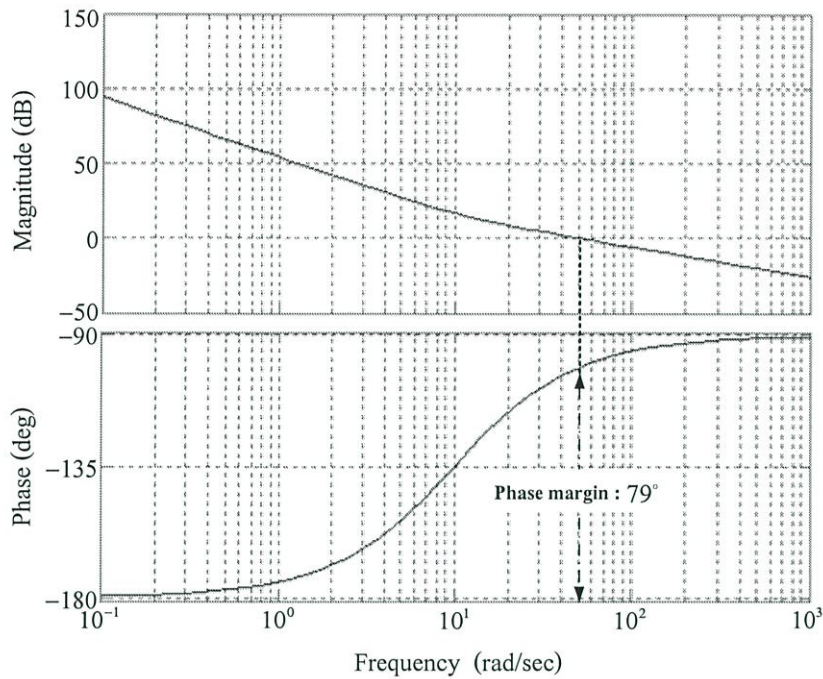
เมื่อ K เป็นค่าคงที่

$$\omega_c = \frac{k_{\omega i}}{k_{\omega p}} < \omega_o \quad (3.30)$$

ในการเลือกความถี่หักมุม (ω_c) ของตัวควบคุม PI ต้องมีพิจารณาการเฟื่อเฟสให้เพียงพอ กับระบบเหมือนกับการควบคุมกระแส ดังนั้นจึงได้เลือก $\omega_c = k_{\omega i}/k_{\omega p} = 10 \text{ rad/s}$ แทน ค่าพารามิเตอร์ตามตารางที่ 4.1 จะได้

$$k_{\omega p} = 0.6 \quad \text{และ} \quad k_{\omega i} = 6$$

เราแสดงแผนภาพโบเดจากการออกแบบได้ดังรูปที่ 3.8 ซึ่งพบว่ามีส่วนเฟื่อเฟส 79 องศา ซึ่งมากเพียงพอสำหรับเสถียรภาพการควบคุม



รูปที่ 3.8 แผนภาพโบเดของระบบควบคุมความเร็วจากการออกแบบ

3.4 ระบบประมาณค่าฟลักซ์โดยใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์

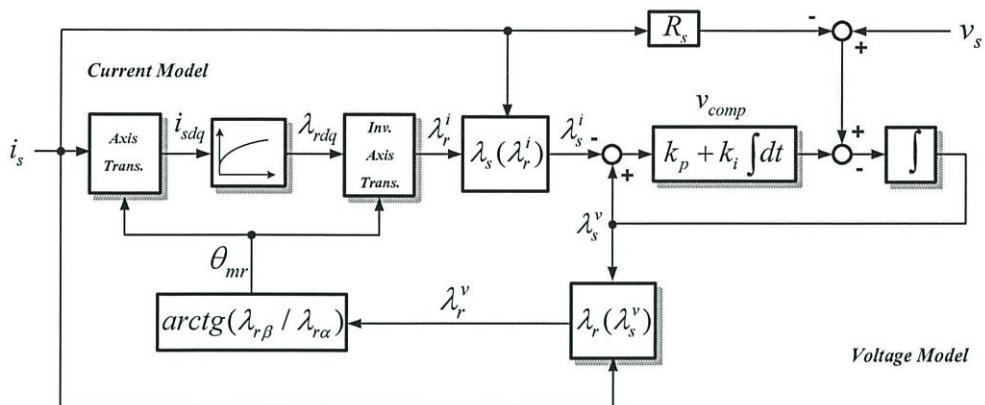
ในการคำนวณหาค่าฟลักซ์สเตเตอร์ จะใช้แบบจำลองแรงดัน (Voltage model) และแบบจำลอง (Current model) ที่ใช้คำนวณหาฟลักซ์สเตเตอร์ตามสมการของมอเตอร์เหนี่ยวนำ ซึ่งได้มีการชดเชยค่าความต้านทานสเตเตอร์และปัญหาอินทิเกรต (Pure integrate) ที่ความเร็วรอบต่ำ โดยผ่านตัวควบคุมแบบ PI ที่เป็นตัวชดเชยค่าความต้านทานสเตเตอร์ที่เปลี่ยนแปลงไป ซึ่งทำให้การคำนวณฟลักซ์สเตเตอร์ที่ความเร็วรอบต่ำถูกต้องมากขึ้น[15]

รูปแบบสมการเดิมที่ใช้สำหรับคำนวณหาฟลักซ์สเตเตอร์แสดงได้ดังสมการ (3.31) และ (3.32) ดังนี้

$$\lambda_{s\alpha} = \int (v_{s\alpha} - R_s i_{s\alpha}) dt \quad (3.31)$$

$$\lambda_{s\beta} = \int (v_{s\beta} - R_s i_{s\beta}) dt \quad (3.32)$$

ในงานวิจัยนี้เราจะใช้แบบจำลองแรงดัน (Voltage model) และแบบจำลองกระแส (Current model) มาใช้ในการคำนวณหาฟลักซ์สเตเตอร์แทนสมการ (3.31) และ (3.32) ซึ่งสามารถแสดงบล็อกไดอะแกรมได้ดังรูปที่ 3.9



รูปที่ 3.9 บล็อกไดอะแกรมคำนวณฟลักซ์สเตเตอร์ในรูปแบบจำลองแรงดัน (Voltage model) และรูปแบบจำลองกระแส (Current model)

เราสามารถเขียนสมการของฟลักซ์สเตเตอร์และฟลักซ์โรเตอร์ที่อยู่บนแกนหนึ่งได้ดังสมการ (3.33) - (3.36)

$$\lambda_{s\alpha} = L_s i_{s\alpha} + M i_{r\alpha} \quad (3.33)$$

$$\lambda_{s\beta} = L_s i_{s\beta} + M i_{r\beta} \quad (3.34)$$

$$\lambda_{r\alpha} = L_r i_{r\alpha} + M i_{s\alpha} \quad (3.35)$$

$$\lambda_{r\beta} = L_r i_{r\beta} + M i_{s\beta} \quad (3.36)$$

นำ $L_s = L_{sl} + M$ แทนในสมการของฟลักซ์สเตเตอร์ (3.33) และ (3.34) จะได้

$$\lambda_{s\alpha} = (L_{sl} + M) i_{s\alpha} + M i_{r\alpha} = L_{sl} i_{s\alpha} + M (i_{s\alpha} + i_{r\alpha}) \quad (3.37)$$

$$\lambda_{s\beta} = (L_{sl} + M) i_{s\beta} + M i_{r\beta} = L_{sl} i_{s\beta} + M (i_{s\beta} + i_{r\beta}) \quad (3.38)$$

จากสมการ (3.37) และ (3.38) จัดรูปได้ใหม่ดังนี้

$$i_{r\alpha} = \left(\frac{\lambda_{s\alpha} - L_{sl} i_{s\alpha}}{M} \right) - i_{s\alpha} \quad (3.39)$$

$$i_{r\beta} = \left(\frac{\lambda_{s\beta} - L_{sl} i_{s\beta}}{M} \right) - i_{s\beta} \quad (3.40)$$

แทนสมการ (3.39) และ (3.40) ในสมการของฟลักซ์โรเตอร์ (3.35) และ (3.36) ตามลำดับดังนี้

$$\lambda_{r\alpha} = L_r \left(\frac{\lambda_{s\alpha} - L_{sl} i_{s\alpha}}{M} - i_{s\alpha} \right) + M i_{s\alpha} = \frac{L_r}{M} \lambda_{s\alpha} - \frac{L_r L_{sl}}{M} i_{s\alpha} - (L_r - M) i_{s\alpha} \quad (3.41)$$

จาก $L_r - M = L_{rl}$ และ $\frac{L_r L_{sl}}{M} + L_{rl} + \sigma L_s \frac{L_r}{M}$ นำไปแทนในสมการ (3.41) จะได้ดังสมการ (3.42)

$$\lambda_{r\alpha} = \frac{L_r}{M} \lambda_{s\alpha} - \left(\frac{L_r L_{sl}}{M} + L_{rl} \right) i_{s\alpha} = \frac{L_r}{M} \lambda_{s\alpha} - \frac{L_r}{M} \sigma L_s i_{s\alpha} \quad (3.42)$$

ดังนั้นสมการของฟลักซ์โรเตอร์ในรูปแบบจำลองแรงดัน (Voltage model) ในตัวแปรของฟลักซ์และกระแสเตเตอร์จะได้ดังสมการ (3.43)

$$\lambda_{r\alpha}^{s,v} = \frac{L_r}{M} (\lambda_{s\alpha}^{s,v} - \sigma L_s i_{s\alpha}^s) = \frac{L_r}{M} \lambda_{s\alpha}^{s,v} - \left(\frac{L_s L_r - M^2}{M} \right) i_{s\alpha}^s \quad (3.43)$$

ดัชนี s, v คือ แกนอ้างอิงแกนนิ่ง (Stationary reference frame) และรูปแบบจำลองแรงดัน (Voltage model) ในทำนองเดียวกันก็จะหา $\lambda_{r\beta}^{s,v}$ ได้โดยการแทนสมการเหมือนกันข้างบนซึ่งจะได้

$$\lambda_{r\beta}^{s,v} = \frac{L_r}{M} (\lambda_{s\beta}^{s,v} - \sigma L_s i_{s\beta}^s) = \frac{L_r}{M} \lambda_{s\beta}^{s,v} - \left(\frac{L_s L_r - M^2}{M} \right) i_{s\beta}^s \quad (3.44)$$

ดังนั้นมุมของฟลักซ์โรเตอร์ที่หาได้จากแบบจำลองแรงดันได้ดังนี้

$$\theta_{mr} = \tan^{-1} \frac{\lambda_{r\beta}^{s,v}}{\lambda_{r\alpha}^{s,v}} \quad (3.45)$$

ถ้าพิจารณาแบบจำลองกระแสจากสมการ (3.3) และ (3.4) เขียนได้ใหม่ดังสมการ (3.46) และ (3.47)

$$\rho \lambda_{rd} = \frac{M}{\tau_r} i_{sd} - \frac{1}{\tau_r} \lambda_{rd} + (\omega_{mr} - p\omega_m) \lambda_{rq} \quad (3.46)$$

$$\rho \lambda_{rq} = \frac{M}{\tau_r} i_{sq} - \frac{1}{\tau_r} \lambda_{rq} - (\omega_{mr} - p\omega_m) \lambda_{rd} \quad (3.47)$$

ในแบบจำลองกระแส พิจารณาสมการ (3.46) และ (3.47) ในขณะเกิดฟิวโอเรียนเตชั่นจะได้

$$\lambda_{rq} = 0 \quad (3.48)$$

แทน (3.48) ลงใน (3.46) และ (3.47) จะได้

$$\rho\lambda_{rd} = \frac{M}{\tau_r} i_{sd} - \frac{1}{\tau_r} \lambda_{rd} \quad (3.49)$$

$$\lambda_{rq} = 0 \quad (3.50)$$

จากสมการ (3.49) และ (3.50) แปลงไปอยู่บนแกนนิ่งจะได้

$$\lambda_{r\alpha}^{s,i} = \lambda_{rd} \cos(\theta_{mr}) - \lambda_{rq} \sin(\theta_{mr}) = \lambda_{rd} \cos(\theta_{mr}) \quad (3.51)$$

$$\lambda_{r\beta}^{s,i} = \lambda_{rd} \sin(\theta_{mr}) + \lambda_{rq} \cos(\theta_{mr}) = \lambda_{rd} \sin(\theta_{mr}) \quad (3.52)$$

จากสมการ (3.35) จะได้ $i_{r\alpha} = \frac{\lambda_{r\alpha} - Mi_{s\alpha}}{L_r}$ แทนในสมการ (3.33) จะได้

$$\lambda_{s\alpha}^{s,i} = \left(\frac{L_s L_r - M^2}{L_r} \right) i_{s\alpha} + \frac{M}{L_r} \lambda_{r\alpha}^{s,i} \quad (3.53)$$

และในทำนองเดียวกันจากสมการ (3.36) จะได้ $i_{r\beta} = \frac{\lambda_{r\beta} - Mi_{s\beta}}{L_r}$ แทนในสมการ (3.34) จะได้

$$\lambda_{s\beta}^{s,i} = \left(\frac{L_s L_r - M^2}{L_r} \right) i_{s\beta} + \frac{M}{L_r} \lambda_{r\beta}^{s,i} \quad (3.54)$$

เราคำนวณหาฟลักซ์สเตเตอร์ในแบบจำลองแรงดัน (Voltage model) โดยการวัดกระแสและแรงดันด้านสเตเตอร์ สำหรับการคำนวณฟลักซ์สเตเตอร์บนแกนนิ่งจะได้

$$\lambda_{s\alpha}^v = \int (v_{s\alpha} - i_{s\alpha} R_s - v_{comp,\alpha}) dt \quad (3.55)$$

$$\lambda_{s\beta}^v = \int (v_{s\beta} - i_{s\beta} R_s - v_{comp,\beta}) dt \quad (3.56)$$

ในแบบจำลองแรงดัน ได้ทำการชดเชยค่าความผิดพลาดของฟลักซ์สเตเตอร์อันผล
เนื่องมาจากความต้านทานสเตเตอร์เปลี่ยนแปลงไป โดยผ่านกระบวนการควบคุมแบบ PI ดัง
สมการ (3.57) และ (3.58)

$$v_{comp.\alpha} = k_p (\lambda_{s\alpha}^v - \lambda_{s\alpha}^{s,i}) + k_i \int (\lambda_{s\alpha}^v - \lambda_{s\alpha}^{s,i}) dt \quad (3.57)$$

$$v_{comp.\beta} = k_p (\lambda_{s\beta}^v - \lambda_{s\beta}^{s,i}) + k_i \int (\lambda_{s\beta}^v - \lambda_{s\beta}^{s,i}) dt \quad (3.58)$$

จากสมการ (3.57) และ (3.58) นั้นที่ความเร็วรอบต่ำแบบจำลองกระแสจะมีบทบาทมาก
ในการคำนวณค่าฟลักซ์สเตเตอร์เพราะว่าแรงเคลื่อนไฟฟ้าเหนี่ยวนำที่คำนวณได้จากแบบจำลอง
แรงดันมีค่าน้อยมากทำให้การคำนวณเกิดความผิดพลาดได้ และในขณะที่ความเร็วรอบสูง
แบบจำลองแรงดันจะมีบทบาทสำคัญในการคำนวณค่าฟลักซ์สเตเตอร์เช่นกัน รูปแบบนี้ในทาง
ปฏิบัติจะมีการส่งผ่านที่ราบเรียบทั้งสองรูปแบบ ทั้งแบบจำลองกระแสและแบบจำลองแรงดัน ซึ่ง
จะมีความถี่ที่สองโพล (ω_1, ω_2) ของฟังก์ชันการ โอนย้ายของตัวประมาณฟลักซ์แบบวงรอบปิด
ซึ่งจะเป็นตัวกำหนดค่าอัตราขยายตัวควบคุมแบบ PI ที่ทำการชดเชย [15]

บทที่ 4

ระบบควบคุมแบบเวกเตอร์ไ้เซนเซอร์วัดความเร็ว

ในบทนี้จะกล่าวถึงการวิเคราะห์ระบบจำลองอ้างอิงแบบปรับตัวพร้อมทั้งบูรณาการตัวสังเกตวงรอบปิดฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัวเข้าด้วยกัน การวิเคราะห์ระบบประมาณค่าความเร็วบนแกนหมุน การออกแบบอัตราขยายปรับตัว และสุดท้ายจะแสดงการจำลองระบบควบคุมแบบเวกเตอร์ไ้เซนเซอร์วัดความเร็วเพื่อดูผลการตอบสนองทางพลวัตโดยใช้โปรแกรมคอมพิวเตอร์

4.1 ระบบจำลองอ้างอิงแบบปรับตัว (Model Reference Adaptive System)

เนื่องจากระบบจำลองอ้างอิงแบบปรับตัวจะกระทำอยู่บนแกนนิ่ง ดังนั้นพิจารณาสมการทางพลวัตของมอเตอร์เหนี่ยวนำชนิดกรงกระรอกโดยการต่อขดลวดแบบสตาร์ (Y) ที่อยู่บนแกนนิ่ง[13] ซึ่งได้นำเสนอไว้แล้วในบทที่ 2 เราเขียนสมการให้มีความง่ายขึ้นได้ดังสมการ (4.1) และ (4.2)

$$\frac{di_s}{dt} = A_{11}i_s + A_{12}\lambda_r + B_1v_s \quad (4.1)$$

$$\frac{d\lambda_r}{dt} = A_{21}i_s + A_{22}\lambda_r \quad (4.2)$$

โดยที่

$$\begin{aligned} A_{11} &= -\left(\frac{R_s}{\sigma L_s} + \frac{1-\sigma}{\sigma\tau_r}\right)I & A_{12} &= k\left(\frac{1}{\tau_r} - p\omega_m J\right) & A_{21} &= \frac{M}{\tau_r}I \\ A_{22} &= -\frac{1}{\tau_r}I + p\omega_m J & B_1 &= \frac{1}{\sigma L_s}I & \sigma &= 1 - \frac{M^2}{L_s L_r} & \tau_r &= \frac{L_r}{R_r} \\ k &= \frac{M}{\sigma L_s L_r} & I &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & J &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \end{aligned}$$

v_s : สเปซเวกเตอร์ของแรงดันสเตเตอร์บนแกน $\alpha - \beta$

i_s : สเปซเวกเตอร์ของกระแสสเตเตอร์บนแกน $\alpha - \beta$

λ_r : สเปซเวกเตอร์ของฟลักซ์โรเตอร์บนแกน $\alpha - \beta$

R_s, R_r : ความต้านทานของขดลวดสเตเตอร์และโรเตอร์ตามลำดับ

L_s, L_r : ความเหนี่ยวนำของขดลวดสเตเตอร์และโรเตอร์ตามลำดับ

M : ความเหนี่ยวนำร่วมระหว่างขดลวดสเตเตอร์และโรเตอร์

σ : สัมประสิทธิ์การรั่วไหลรวม (Total leakage coefficient)

ω_m : ความเร็วโรเตอร์ (Rotor speed)

p : จำนวนคู่ของขั้วแม่เหล็ก (Poles pair)

การประมาณค่าความเร็วด้วยระบบจำลองอ้างอิงแบบปรับตัวจะใช้สถานะของฟลักซ์โรเตอร์ของมอเตอร์เหนี่ยวนำโดยการวัดปริมาณแรงดันและกระแสด้านสเตเตอร์ซึ่งสถานะของฟลักซ์โรเตอร์ ($\lambda_{r\alpha}, \lambda_{r\beta}$) ที่ได้จากแบบจำลองแรงดัน (Voltage model) และสถานะของฟลักซ์โรเตอร์ ($\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$) ที่ได้จากแบบจำลองกระแส (Current model) ซึ่งความแตกต่างของสถานะฟลักซ์โรเตอร์ดังกล่าวจะนำไปปรับเอาต์พุตของค่าประมาณความเร็วโดยผ่านกระบวนการควบคุม PI ซึ่งเอาต์พุตของตัวควบคุมแบบ PI ก็คือความเร็วที่ประมาณได้

จากสมการ (4.1) ซึ่งเป็นสมการของมอเตอร์ด้านสเตเตอร์เราสามารถเขียนสมการได้ใหม่ในแบบจำลองแรงดัน (Voltage model) ได้ดังสมการ (4.3) และ (4.4)

$$\lambda_{r\alpha} = \frac{L_r}{M} \left[\int (v_{s\alpha} - R_s i_{s\alpha}) dt - \sigma L_s i_{s\alpha} \right] \quad (4.3)$$

$$\lambda_{r\beta} = \frac{L_r}{M} \left[\int (v_{s\beta} - R_s i_{s\beta}) dt - \sigma L_s i_{s\beta} \right] \quad (4.4)$$

และจากสมการ (4.2) เป็นสมการของมอเตอร์ด้านโรเตอร์เราสามารถเขียนสมการได้ใหม่ในแบบจำลองกระแส (Current model) ได้ดังสมการ (4.5) และ (4.6)

$$\hat{\lambda}_{r\alpha} = \frac{1}{\tau_r} \int [M i_{s\alpha} - \hat{\lambda}_{r\alpha} - \tau_r p \hat{\omega}_m \hat{\lambda}_{r\beta}] dt \quad (4.5)$$

$$\hat{\lambda}_{r\beta} = \frac{1}{\tau_r} \int [M i_{s\beta} - \hat{\lambda}_{r\beta} + \tau_r p \hat{\omega}_m \hat{\lambda}_{r\alpha}] dt \quad (4.6)$$

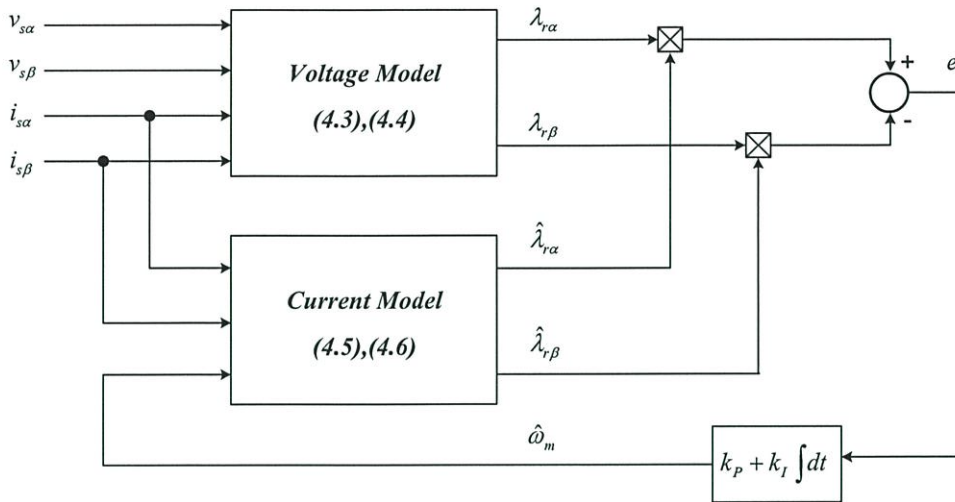
ดังนั้นสัญญาณที่ใช้ในการปรับความเร็ว (Speed tuning signal) ดังสมการ (4.7)

$$e = \lambda_{r\beta} \hat{\lambda}_{r\alpha} - \lambda_{r\alpha} \hat{\lambda}_{r\beta} \quad (4.7)$$

$$\hat{\omega}_m = (k_p + k_I \int dt) e \quad (4.8)$$

สำหรับสัญญาณที่ใช้ในการปรับความเร็วนั้นก็คืออินพุตที่ไปผ่านกระบวนการควบคุมแบบ PI ดังสมการ (4.8) เอาต์พุตของของตัวควบคุม PI ก็คือความเร็วที่ได้จากการประมาณนั่นเอง ซึ่งตัวควบคุมแบบ PI นั้นจะทำการปรับค่าความเร็วเมื่อเกิดความคลาดเคลื่อนระหว่างฟลักซ์โรเตอร์ทั้งสองสถานะที่ไม่เท่ากับศูนย์ และเมื่อความคลาดเคลื่อนเป็นศูนย์แล้วค่าความเร็ว

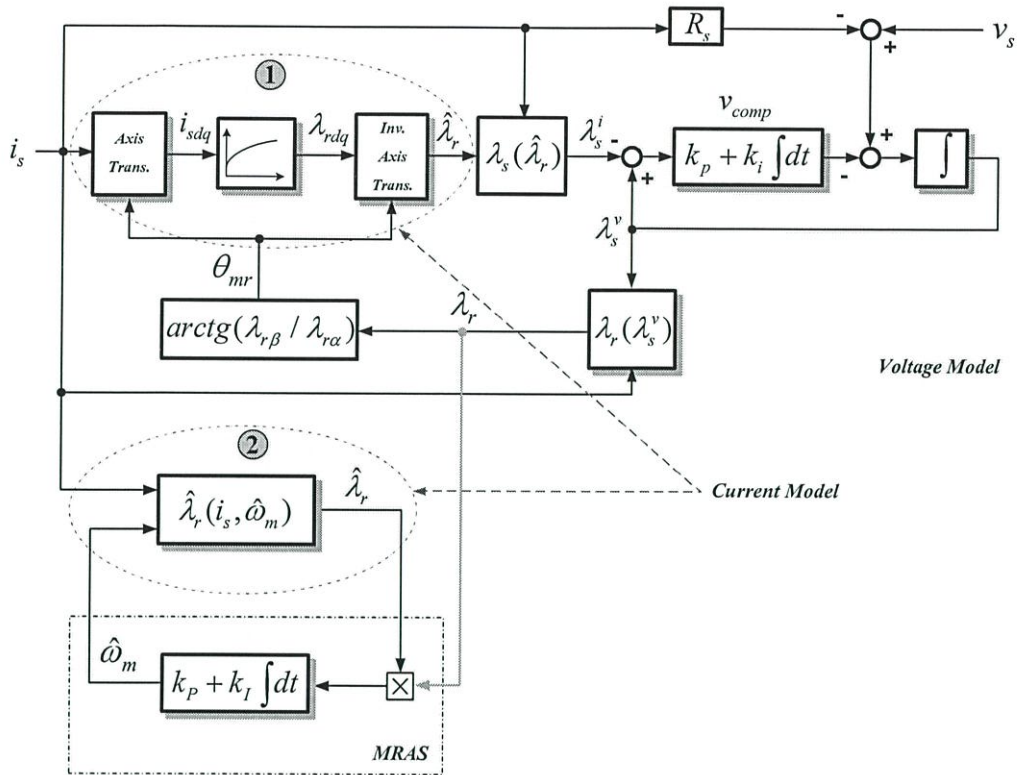
ของโรเตอร์จากการประมาณจะเท่ากับความเร็วจริงของโรเตอร์ ดังนั้นจากสมการ (4.3), (4.4), (4.5), (4.6), (4.7) และ (4.8) นำมาเขียนเป็นแบบจำลองได้ดังรูปที่ 4.1



รูปที่ 4.1 ระบบจำลองอ้างอิงแบบปรับตัวโดยใช้วิธีฟลักซ์โรเตอร์

4.2 การบูรณาการตัวสังเกตวงรอบปิดฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัว

เนื่องจากระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์ประกอบไปด้วยฟลักซ์โรเตอร์จากแบบจำลองกระแสและฟลักซ์โรเตอร์จากแบบจำลองแรงดัน แต่สามารถใช้ฟลักซ์โรเตอร์จากแบบจำลองแรงดันเท่านั้นในการประมาณค่าความเร็ว เนื่องจากระบบจำลองอ้างอิงแบบปรับตัว (MRAS) เป็นระบบวงรอบปิดซึ่งต้องมีการป้อนกลับค่าความเร็วจากการประมาณมายังแบบจำลองกระแส แต่แบบจำลองกระแสในระบบประมาณค่าฟลักซ์นั้นไม่ได้ใช้ความเร็วในการคำนวณ จึงไม่สามารถป้อนกลับความเร็วได้ ดังนั้นจึงต้องใช้สมการฟลักซ์โรเตอร์ที่อยู่บนแกนนิ่งเป็นแบบจำลองกระแสอีกครั้ง ด้วยเหตุนี้ระบบจึงเกิดความซ้ำซ้อนเนื่องจากต้องมีแบบจำลองกระแส 2 ชุด แสดงได้ดังรูปที่ 4.2



รูปที่ 4.2 แสดงความซ้ำซ้อนของระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัว

ดังนั้นจากสมการ (4.2) ถ้าพิจารณาฟลักซ์โรเตอร์บนแกนหมุนจะได้ดังสมการ (4.9)

$$\lambda_{rdq} = \frac{M}{1+s\tau_r} i_{sdq} - J \frac{\omega_{mr} - p\omega_m}{1+s\tau_r} \tau_r \lambda_{rdq} \tag{4.9}$$

รูปแบบจำลองกระแส : จากสมการ (4.9) เมื่อทำการแปลงให้อยู่บนแกนนิ่ง ($\omega_{mr} = 0$) จะได้ฟลักซ์โรเตอร์ดังสมการ (4.10)

$$\hat{\lambda}_r = \frac{M}{1+s\tau_r} i_s + J \frac{p\omega_m}{1+s\tau_r} \tau_r \hat{\lambda}_r \tag{4.10}$$

จากแบบจำลองกระแสนี้ ฟลักซ์สเตเตอร์ที่อยู่บนแกนนิ่งสามารถคำนวณได้จากสมการ (4.11)

$$\lambda_s^i = \frac{M}{L_r} \hat{\lambda}_r + \frac{L_s L_r - M^2}{L_r} i_s \tag{4.11}$$

รูปแบบจำลองแรงดัน : รูปแบบจำลองแรงดันได้จากการวัดแรงดันและกระแสด้านสเตเตอร์ สำหรับบนแกนนิ่งฟลักซ์สเตเตอร์คำนวณได้จากสมการ (4.12)

$$\lambda_s^v = \frac{1}{s} (v_s - R_s i_s - v_{comp}) \quad (4.12)$$

ในแบบจำลองแรงดันได้ทำการชดเชยค่าความผิดพลาดของฟลักซ์สเตเตอร์อันผล
เนื่องมาจากความต้านทานสเตเตอร์เปลี่ยนแปลงไปโดยผ่านตัวควบคุมแบบ PI ดังสมการ (4.13)

$$v_{comp} = (k_p + k_i/s) (\lambda_s^v - \lambda_s^i) \quad (4.13)$$

จากแบบจำลองแรงดันนี้สามารถคำนวณหาฟลักซ์โรเตอร์ที่อยู่บนแกนหนึ่ง ได้ดังสมการ (4.14)

$$\lambda_r = \frac{L_r}{M} \lambda_s^v - \frac{L_s L_r - M^2}{M} i_s \quad (4.14)$$

และมุมหรือตำแหน่งของฟลักซ์โรเตอร์จะคำนวณได้จากสมการ (4.15)

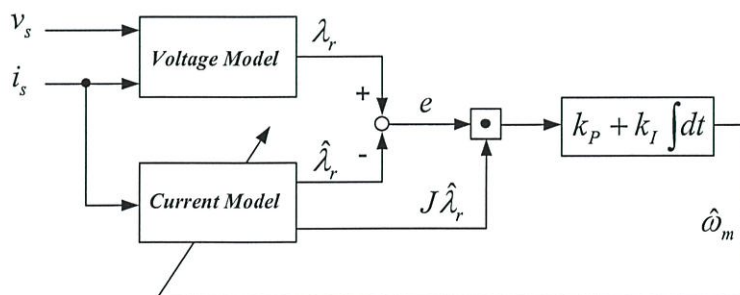
$$\theta_{mr} = \tan^{-1} (\lambda_{r\beta} / \lambda_{r\alpha}) \quad (4.15)$$

ดังนั้นสัญญาณที่ใช้ในการปรับความเร็ว (Speed tuning signal) ของระบบจำลองอ้างอิง
แบบปรับตัวเป็นดังสมการ (4.16) จะไปผ่านตัวควบคุมแบบ PI ดังรูปที่ 4.3 เพื่อประมาณค่า
ความเร็วดังสมการ (4.17)

$$\varepsilon_\omega = e^T J \hat{\lambda}_r \quad (4.16)$$

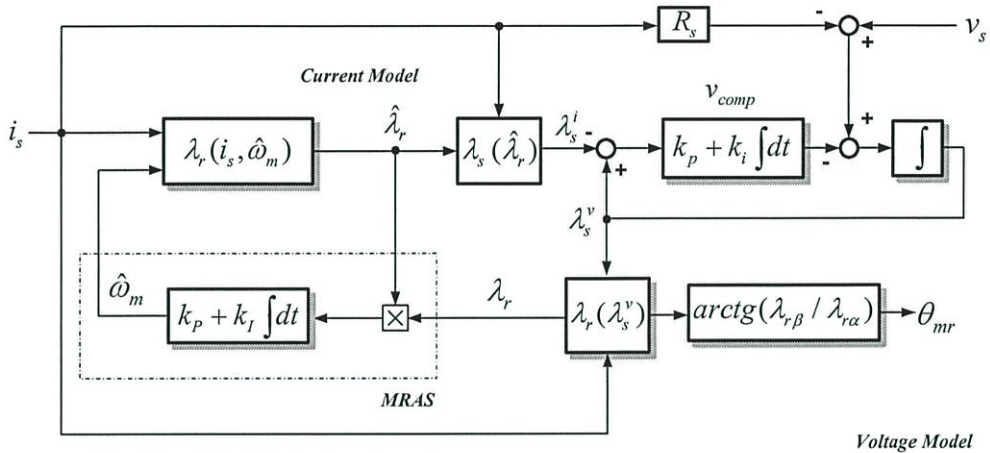
โดยที่ $e = \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix}$

$$\hat{\omega}_m = (k_p + k_i \int dt) \varepsilon_\omega \quad (4.17)$$



รูปที่ 4.3 ระบบจำลองอ้างอิงแบบปรับตัว (MRAS)

จากรูปที่ 4.2 จะเห็นว่าแบบจำลองกระแสทั้ง 2 ชุด สามารถคำนวณค่าฟลักซ์โรเตอร์ได้เท่ากัน เราจึงทำการยกเลิกแบบจำลองกระแส (แบบจำลองที่ 1) ของระบบประมาณค่าฟลักซ์แล้วแทนที่ด้วยแบบจำลองกระแส (แบบจำลองที่ 2) ของระบบประมาณค่าความเร็วเพื่อให้สามารถป้อนกลับความเร็วประมาณ ($\hat{\omega}_m$) ไปยังแบบจำลองกระแสได้ ดังนั้นเราสามารถบูรณาการระบบทั้งสองเข้าด้วยกันเพื่อลดความซ้ำซ้อนได้ดังรูปที่ 4.4



รูปที่ 4.4 การบูรณาการระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์และระบบจำลองอ้างอิงแบบปรับตัว

4.3 ระบบการประมาณค่าความเร็วบนแกนหมุน

เนื่องจากสมการแบบจำลองแรงดันเป็นสมการจริงของมอเตอร์ ดังนั้นสมการจากแบบจำลองแรงดันจึงสมมูลกับสมการแบบจำลองกระแสที่ใช้ความเร็วจริง เราจึงสามารถใช้สมการแบบจำลองกระแสของมอเตอร์จริงเป็นแบบจำลองแรงดันในการพิจารณาได้[18]

แบบจำลองแรงดัน (Voltage model) :

$$\frac{d\lambda_r}{dt} = \frac{R_r}{L_r} I (Mi_s - \lambda_r) + J\lambda_r p\omega_m \quad (4.18)$$

แบบจำลองกระแส (Current model) :

$$\frac{d\hat{\lambda}_r}{dt} = \frac{R_r}{L_r} I (Mi_s - \hat{\lambda}_r) + J\hat{\lambda}_r p\hat{\omega}_m \quad (4.19)$$

จากสมการ (4.18) และ(4.19) จะได้ความผิดพลาดของแรงเคลื่อนไฟฟ้าเหนี่ยวนำดังสมการ (4.20)

$$s(\lambda_r - \hat{\lambda}_r) = -\frac{R_r}{L_r} I (\lambda_r - \hat{\lambda}_r) + pJ\omega_m (\lambda_r - \hat{\lambda}_r) + pJ\hat{\lambda}_r (\omega_m - \hat{\omega}_m) \quad (4.20)$$

เมื่อ $se = s(\lambda_r - \hat{\lambda}_r)$ และ s แทนตัวนำเนินการอนุพันธ์ (d/dt) จะได้ดังสมการ (4.21)

$$se = \left(-\frac{R_r}{L_r} I + Jp\omega_m \right) e + pJ\hat{\lambda}_r (\omega_m - \hat{\omega}_m) \quad (4.21)$$

ดังนั้นจะได้ความผิดพลาดของฟลักซ์โรเตอร์ดังสมการ (4.22)

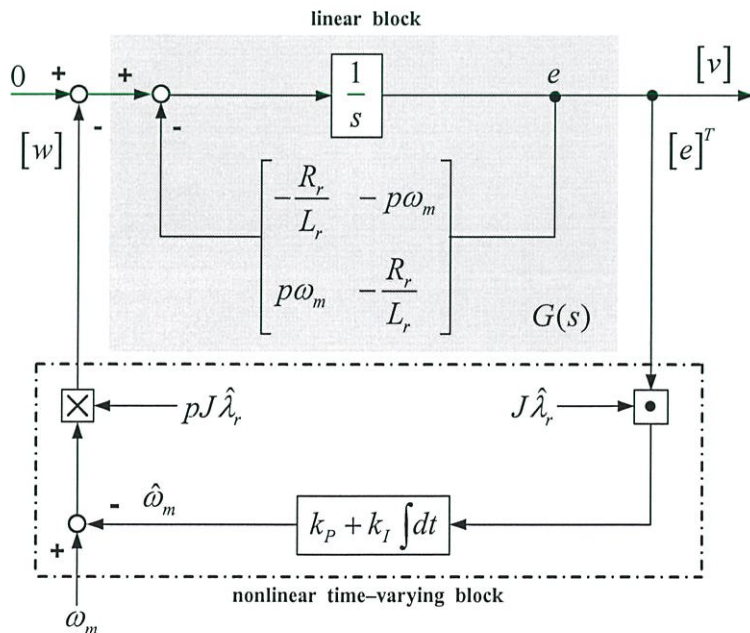
$$e = \left[\left(s + \frac{R_r}{L_r} \right) I - p\omega_m J \right]^{-1} pJ\hat{\lambda}_r (\omega_m - \hat{\omega}_m) \quad (4.22)$$

$$e = G(s) pJ\hat{\lambda}_r (\omega_m - \hat{\omega}_m) \quad (4.23)$$

โดยที่
$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} = [(s+x)I - yJ]^{-1} \quad (4.24)$$

และ
$$x = \frac{R_r}{L_r}, \quad y = p\omega_m$$

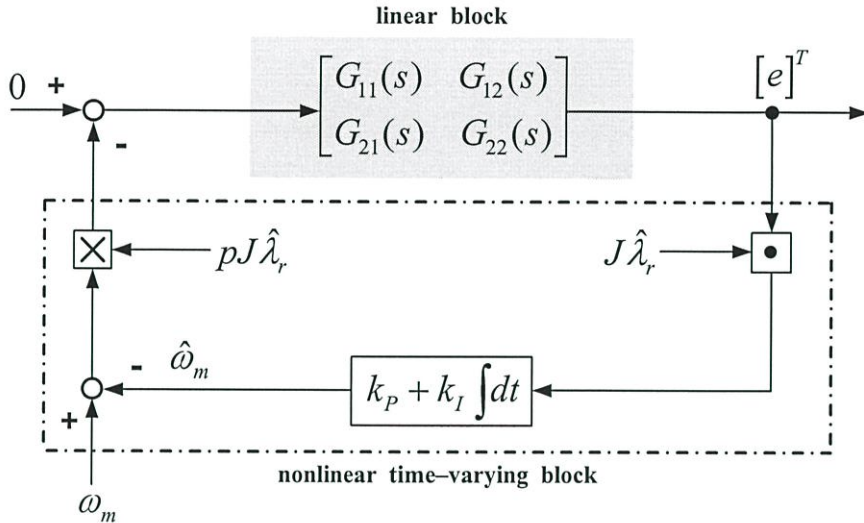
จากสมการ (4.22) และ (4.23) เราแสดงบล็อกไดอะแกรมวงรอบปิดซึ่งจะประกอบไปด้วยฟังก์ชันการโอนย้ายของค่าผิดพลาดฟลักซ์โรเตอร์และตัวควบคุมแบบ PI ได้ดังรูปที่ 4.5 จากบล็อกไดอะแกรมจะมองเห็นประกอบ 2 ส่วน คือ ส่วนเป็นเชิงเส้น (linear block) และส่วนไม่เป็นเชิงเส้นที่เปลี่ยนแปลงตามเวลา (nonlinear time-varying block) และเราแสดงบล็อกไดอะแกรมของระบบประมาณค่าความเร็วได้ใหม่ดังรูปที่ 4.6



รูปที่ 4.5 บล็อกไดอะแกรมความผิดพลาดของระบบประมาณค่าความเร็ว

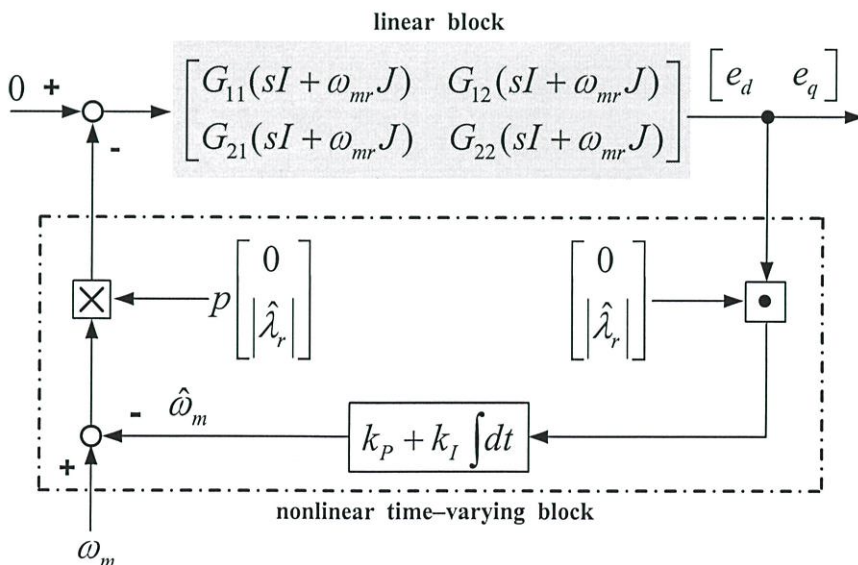
พิจารณาจากระบบจำลองอ้างอิงแบบปรับตัวดังรูปที่ 4.5 จะได้ความเร็วจากการประมาณ
 ดังสมการ (4.25)

$$\hat{\omega}_m = (k_p + k_I \int dt) e^T J \hat{\lambda}_r \quad (4.25)$$



รูปที่ 4.6 ระบบประมาณค่าความเร็วบนแกนนิ่ง

จากรูปที่ 4.6 การวิเคราะห์ระบบประมาณค่าความเร็วบนแกนนิ่งนั้นค่อนข้างยุ่งยาก
 เนื่องจากมีสัญญาณเข้าออกหลายสัญญาณ ดังนั้นเราจึงแปลงระบบประมาณค่าความเร็วจากแกน
 นิ่งไปอยู่บนแกนหมุนซึ่งเป็นแกนเดียวกับการควบคุมแบบเวกเตอร์ ดังนั้นฟังก์ชันการโอนย้าย
 $G(s)$ จะถูกแปลงเป็น $G(sI + \omega_{mr}J)$ แสดงได้ดังรูปที่ 4.7



รูปที่ 4.7 ระบบประมาณค่าความเร็วบนแกนหมุน

เนื่องจากสัญญาณขาเข้า $G(sI + \omega_{mr}J)$ มีเฉพาะองค์ประกอบในแกน q เท่านั้น ดังนั้นสัญญาณที่ตอบสนองสัญญาณเข้านี้คือ $G_{12}(sI + \omega_{mr}J)$ และ $G_{22}(sI + \omega_{mr}J)$ และองค์ประกอบในแกน q ของสัญญาณผิดพลาด e เท่านั้นที่ใช้ประมาณค่าความเร็วเชิงสมการ (4.26)

$$e = \begin{bmatrix} e_d \\ e_q \end{bmatrix} = G(sI + \omega_{mr}J) \begin{bmatrix} 0 \\ |\hat{\lambda}_r| \end{bmatrix} p(\omega_m - \hat{\omega}_m) \quad (4.26)$$

$$\text{และ } G'(s) = \begin{bmatrix} G_{11}(sI + \omega_{mr}J) & G_{12}(sI + \omega_{mr}J) \\ G_{21}(sI + \omega_{mr}J) & G_{22}(sI + \omega_{mr}J) \end{bmatrix} \quad (4.27)$$

$$= \begin{bmatrix} G'_{11}(s) & G'_{12}(s) \\ G'_{21}(s) & G'_{22}(s) \end{bmatrix} \quad (4.28)$$

นั่นคือฟังก์ชันการโอนย้าย $G'_{22}(s)$ เท่านั้นที่ตอบสนองสัญญาณขาเข้าและถูกใช้ประมาณค่าความเร็ว โดยที่

$$G'_{22}(s) = [(s+x)I - (\omega_{mr} - y)J]^{-1} \quad (4.29)$$

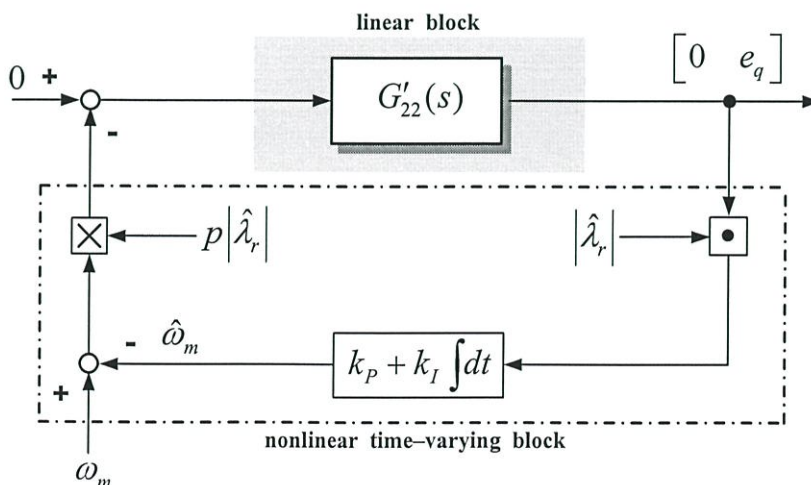
เมื่อ $\omega_{sl} = \omega_{mr} - y$ ดังนั้นจะได้

$$G'_{22}(s) = \frac{1}{(s+x)^2 + \omega_{sl}^2} \begin{bmatrix} s+x & -\omega_{sl} \\ \omega_{sl} & s+x \end{bmatrix} \quad (4.30)$$

ดังนั้นเราจะได้

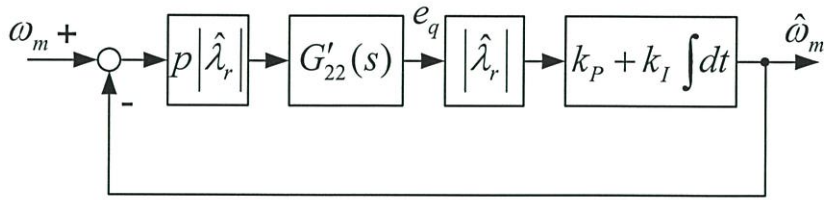
$$G'_{22}(s) = \frac{s+x}{(s+x)^2 + \omega_{sl}^2} \quad (4.31)$$

แสดงบล็อกไดอะแกรมระบบประมาณค่าความเร็วบนแกนหมุนได้ดังรูปที่ 4.8



รูปที่ 4.8 ระบบประมาณค่าความเร็วบนแกนหมุนที่ใช้ฟังก์ชันการโอนย้าย $G'_{22}(s)$

สามารถแสดงบล็อกไดอะแกรมของระบบประมาณค่าความเร็วได้ดังรูปที่ 4.9



รูปที่ 4.9 ระบบประมาณค่าความเร็วบนแกนหมุนในลักษณะสัญญาณเข้าออกเดียว (SISO)

4.4 การออกแบบอัตราขยายปรับตัว [13]

เนื่องจากในทางปฏิบัติการเปลี่ยนแปลงความเร็วจริงของมอเตอร์มีลักษณะเป็นแรมปี่อยู่เสมอ ซึ่งในขณะที่มอเตอร์เปลี่ยนแปลงความเร็วนั้นแรงบิดที่มอเตอร์สร้างขึ้น (T_m) จะมีค่าคงที่ และจะทำให้ความเร็วของมอเตอร์เพิ่มขึ้นหรือลดลงเป็นเชิงเส้น โดยมีค่าความชันเท่ากับ $R = \frac{d\omega_m}{dt} = \frac{T_m - T_L}{J_m}$ ซึ่งตัวควบคุมแบบ PI ของระบบประมาณค่าความเร็วไม่สามารถจัดค่าความเร็วผิดพลาดให้มีค่าเป็นศูนย์ได้ ดังนั้นจึงออกแบบโดยพิจารณาถึงผลตอบสนองของระบบประมาณค่าความเร็วเมื่อความเร็วจริงเป็นฟังก์ชันแรมปี่ จากรูปที่ 4.9 สามารถแสดงฟังก์ชันการโอนย้ายระหว่างความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) เทียบกับความเร็วจริง (ω_m) ได้ดังสมการ (4.32)

$$\frac{\omega_m(s) - \hat{\omega}_m(s)}{\omega_m(s)} = \frac{1}{1 + CG'_{22}(s) \cdot (k_p + k_I/s)} \quad (4.32)$$

เมื่อ $C = p|\hat{\lambda}_r|^2$

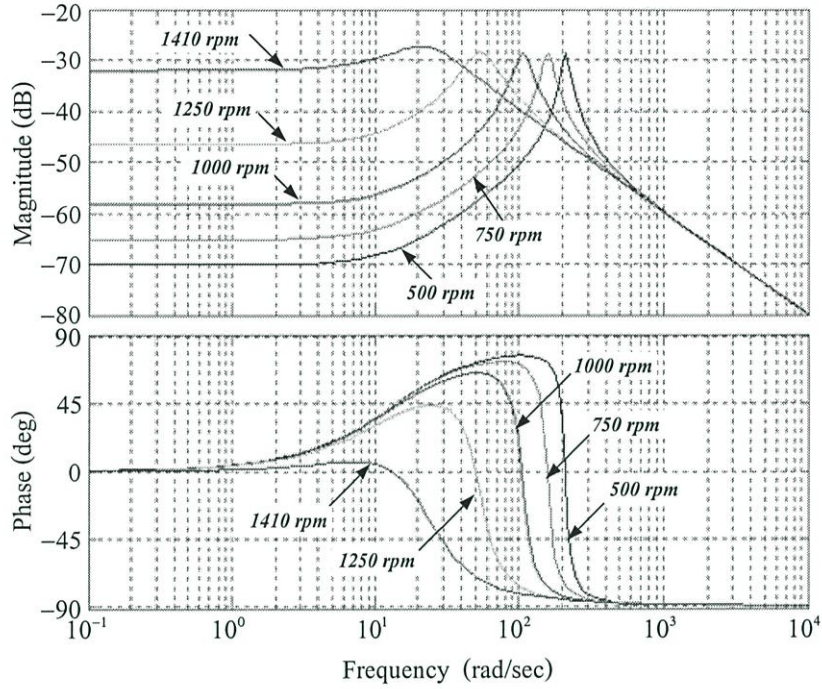
จากทฤษฎีบทค่าสุดท้าย (Final value theorem) สามารถหาค่าความเร็วผิดพลาด (e_{ss}) ในขณะเปลี่ยนแปลงความเร็วจะได้ดังสมการ (4.33)

$$e_{ss} = \lim_{s \rightarrow 0} \left[s \cdot \frac{s}{s + CG'_{22}(s) \cdot (sk_p + k_I)} \right] \left[\frac{T_m - T_L}{J_m s^2} \right] \quad (4.33)$$

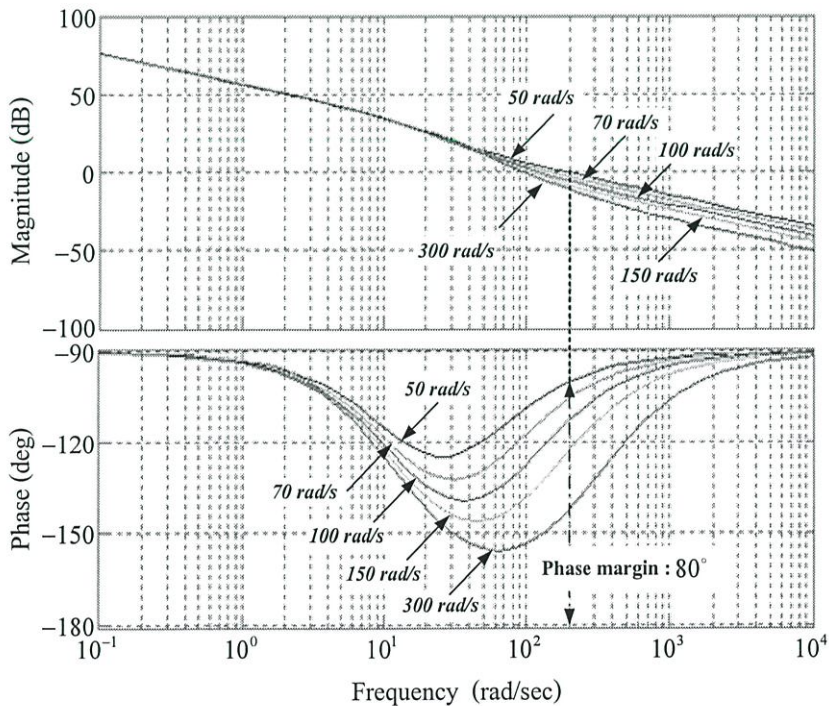
$$= \frac{T_m - T_L}{J_m CG'_{22}(s)|_{s=0} \cdot k_I} \quad (4.34)$$

จากสมการ (4.34) สังเกตจะเห็นว่าค่า k_I มากจะลดค่าความผิดพลาดของความเร็วได้ งานวิจัยนี้ได้เลือกค่า $k_I = 30000$ ในขณะไม่มีโหลด ($T_L = 0$) แทนค่าพารามิเตอร์ตามตารางที่ 4.1 จะได้ค่า $e_{ss} = 5$ rpm การเลือกค่า k_p จะพิจารณาส่วนเฟส (Phase margin) ของระบบประมาณค่าความเร็ว โดยเลือกค่าความถี่หักมุม ($\omega_c = k_I/k_p$) ของตัวควบคุม PI ไม่ควร

เลือกให้มีค่าสูงจนเกินไปเพราะอาจจะมีผลต่อสัญญาณรบกวนส่งผลต่อเสถียรภาพของระบบได้ จากรูปที่ 4.10 เลือกความถี่หักมุม $\omega_c = 50$ rad/s จะได้ $k_p = 600$ แสดงแผนภาพโบเดของระบบประมาณค่าความเร็วรอบเปิดที่ได้ทำการออกแบบไว้ดังรูปที่ 4.11 ซึ่งมีส่วนเฟส 80 องศา ซึ่งมีค่ามากเพียงพอสำหรับระบบควบคุม



รูปที่ 4.10 แผนภาพโบเดของ $G'_{22}(s)$ ที่จุดทำงานใดๆ



รูปที่ 4.11 แผนภาพโบเดของระบบประมาณค่าความเร็วจากการออกแบบ

4.5 การจำลองระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วบนระบบจำลองอ้างอิงแบบปรับตัวด้วยโปรแกรมคอมพิวเตอร์

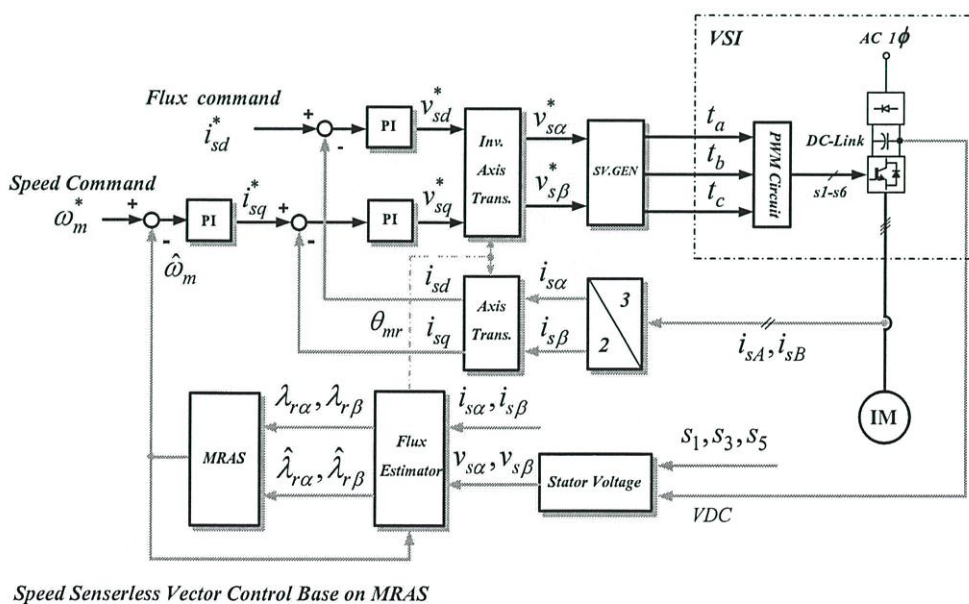
เพื่อเป็นการตรวจสอบความถูกต้องของการออกแบบระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วบนระบบจำลองอ้างอิงแบบปรับตัว งานวิจัยนี้ได้ใช้โปรแกรม MATLAB/SIMULINK จำลองทดสอบเพื่อผลการตอบสนองด้านพลวัต โดยจำลองโครงสร้างของระบบดังรูปที่ 4.12 พารามิเตอร์ของมอเตอร์ได้แสดงไว้ตามตารางที่ 4.1 และอัตราขยายที่ได้ทำการออกแบบเพื่อใช้ในการทดสอบแสดงไว้ตามตารางที่ 4.2

ตารางที่ 4.1 พิกัดและพารามิเตอร์ของมอเตอร์เหนี่ยวนำ

3HP, 220 / 380V, 8.9 / 5.16A, 1410 rpm, 4 Pole	
$i_{sd} = 1.8 \text{ A}$	$i_{sq} = 3.56 \text{ A}$
$R_s = 3.125 \Omega$	$R_r = 3.115 \Omega$
$L_s = 0.224 \text{ H}$	$L_r = 0.228 \text{ H}$
$M = 0.215 \text{ H}$	$J_m = 0.012 \text{ kg} - \text{m}^2$

ตารางที่ 4.2 อัตราขยายระบบควบคุมที่ได้ออกแบบ

อัตราขยายระบบควบคุมกระแส	$k_{cp} = 20, k_{ci} = 5000$
อัตราขยายระบบควบคุมความเร็ว	$k_{\omega p} = 0.6, k_{\omega i} = 6$
อัตราขยายระบบประมาณค่าฟลักซ์ (ทดลอง)	$k_p = 35, k_i = 150$
อัตราขยายระบบประมาณค่าความเร็ว	$k_p = 600, k_I = 30000$

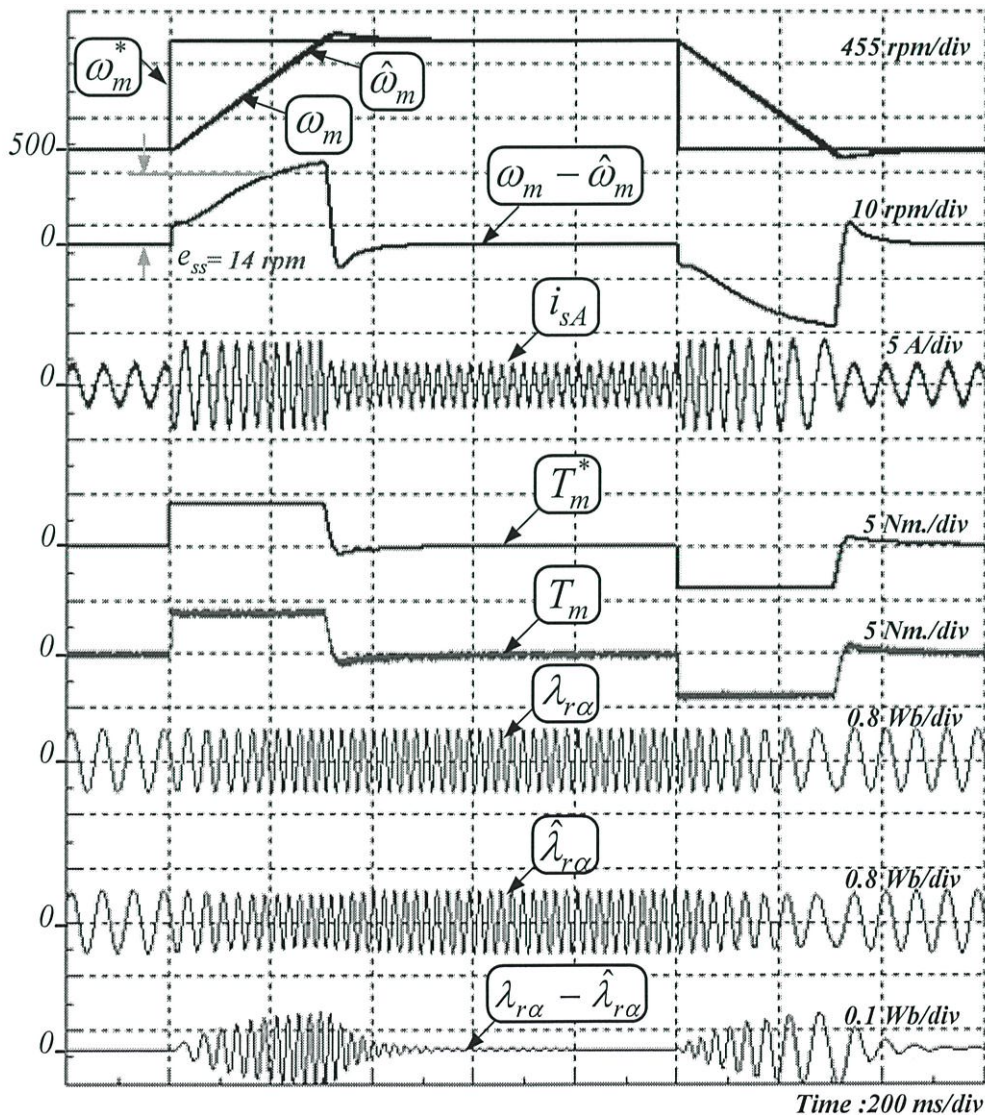


รูปที่ 4.12 โครงสร้างของระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็ว

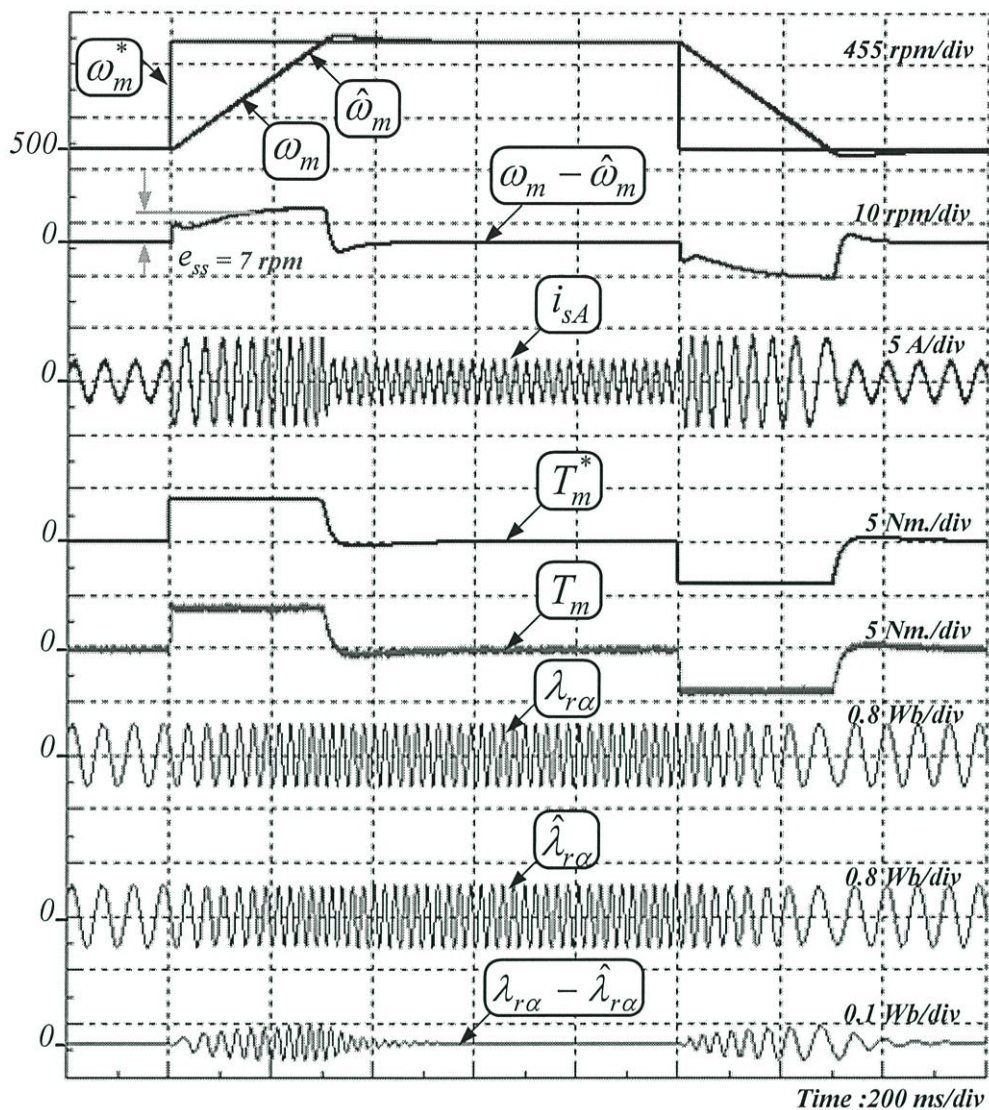
4.5.1 ทดสอบขณะเปลี่ยนแปลงความเร็ว จาก 500 ถึง 1410 rpm และ จาก 1410 ถึง 500 rpm

โดยการเปลี่ยนแปลงอัตราขยายปรับตัวแบบอินทิเกรต (k_I)

ทดสอบในขณะที่เปลี่ยนแปลงความเร็วรอบจาก 500 ถึง 1410 rpm เรททดลองเลือกค่า $k_I=15000$ ซึ่งจะได้ค่า $e_{ss} = 14$ rpm (ค่าเฉลี่ย) ดังรูปที่ 4.13 จะเห็นว่าฟลักซ์โรเตอร์จากการคำนวณในแบบจำลองแรงคั้นและแบบจำลองกระแสมีความผิดพลาด จึงทำให้ระบบจำลองอ้างอิงแบบปรับตัวประมาณค่าความเร็วได้ผิดพลาดเช่นกัน แต่ในขณะที่เดียวกันเรากำหนดค่า $k_I=30000$ (จากการออกแบบ) ซึ่งจะได้ค่า $e_{ss} = 7$ rpm (ค่าเฉลี่ย) ดังรูปที่ 4.14 ซึ่งค่า e_{ss} ลดลง 2 เท่าของค่า k_I ที่เพิ่มขึ้น 2 เท่า ซึ่งจะสังเกตเห็นว่าฟลักซ์โรเตอร์จากการคำนวณในแบบจำลองแรงคั้นและแบบจำลองกระแสมีค่าผิดพลาดลดลงยังผลทำให้ระบบจำลองอ้างอิงแบบปรับตัวประมาณค่าความเร็วได้ใกล้เคียงกับความเร็วจริงมากขึ้น ซึ่งผลการทดสอบนี้มีแนวโน้มเป็นไปตามสมการ (4.34)



รูปที่ 4.13 ทดสอบขณะเปลี่ยนแปลงความเร็วของมอเตอร์ เมื่อ $k_p = 600$, $k_I = 15000$



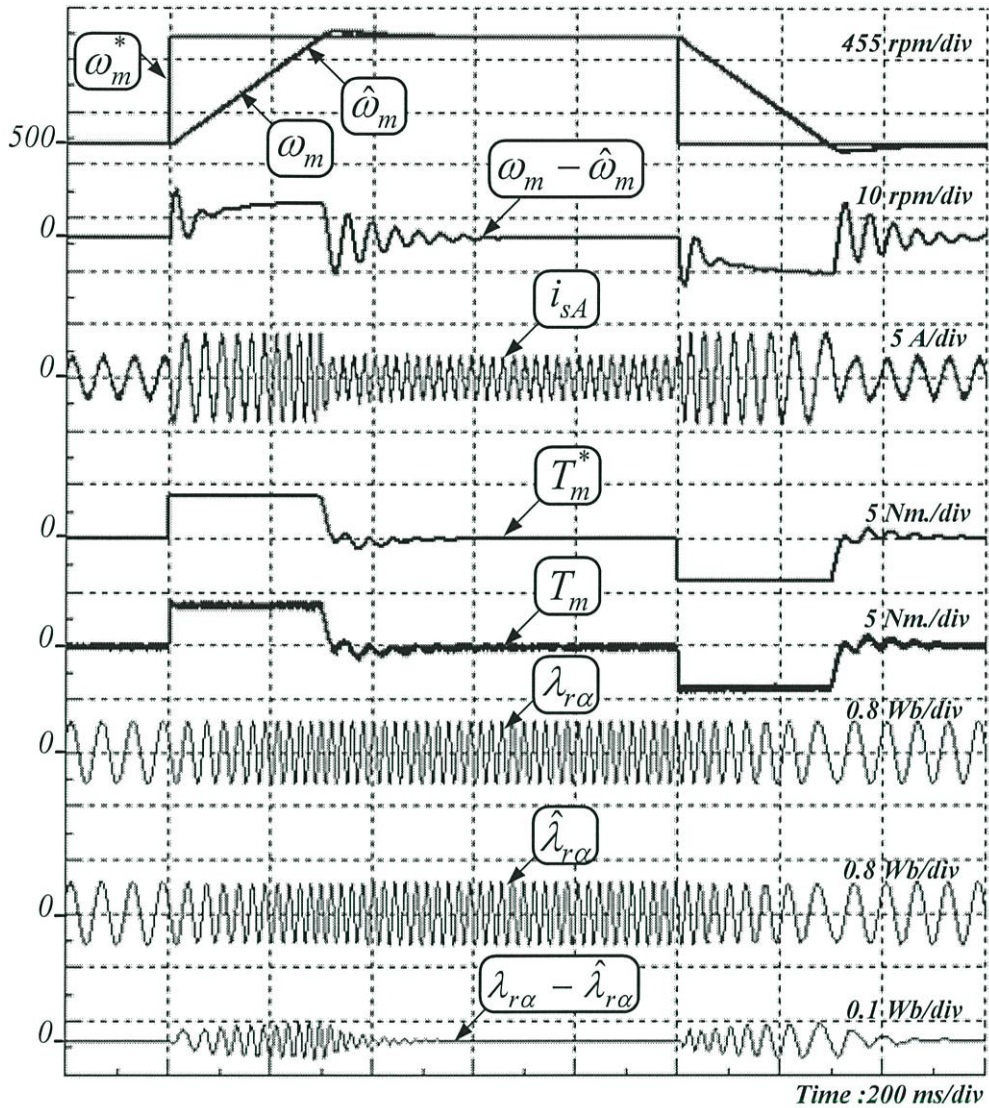
รูปที่ 4.14 ทดสอบขณะเปลี่ยนแปลงความเร็วของมอเตอร์ เมื่อ $k_p = 600$, $k_I = 30000$

4.5.2 ทดสอบขณะเปลี่ยนแปลงความเร็ว จาก 500 ถึง 1410 rpm และ จาก 1410 ถึง 500 rpm

โดยการเปลี่ยนแปลงอัตราขยายปรับตัวแบบสัดส่วน (k_p)

จากการทดสอบที่ผ่านมา ดังสมการ (4.34) ถ้าเราเพิ่มค่า k_I มากก็จะทำให้ระบบประมาณค่าความเร็วประมาณค่าได้ถูกต้องมากขึ้น และเราควรเลือกค่า k_p มีค่าต่ำเพื่อลดผลกระทบจากสัญญาณรบกวนในทางปฏิบัติ แต่การเลือก k_p น้อยเกินไปอาจจะทำให้ระบบประมาณค่าความเร็วเกิดการแกว่งเนื่องจากเฟส (Phase margin) น้อยเกินไป การทดสอบนี้จะพิจารณาความถี่ของพลาซมีโรเตอร์ ($\omega_{mr} = 295 \text{ rad/s}$) ถ้าเลือกความถี่หักมุมใกล้เคียงหรือมากกว่านี้จะทำให้ระบบประมาณค่าความเร็วแกว่งได้ เช่น เลือก $\omega_c = 300 \text{ rad/s}$ จะทำให้ได้ค่า $k_p = 100$ และแสดงแผนภาพโพลดังรูปที่ 4.11 พบว่าจะมีส่วนเฟสเพียง 15 องศา ซึ่งแสดงผลการทดสอบได้ดังรูปที่ 4.15 จะสังเกตเห็นว่าความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) มีค่ามากในขณะที่มอเตอร์

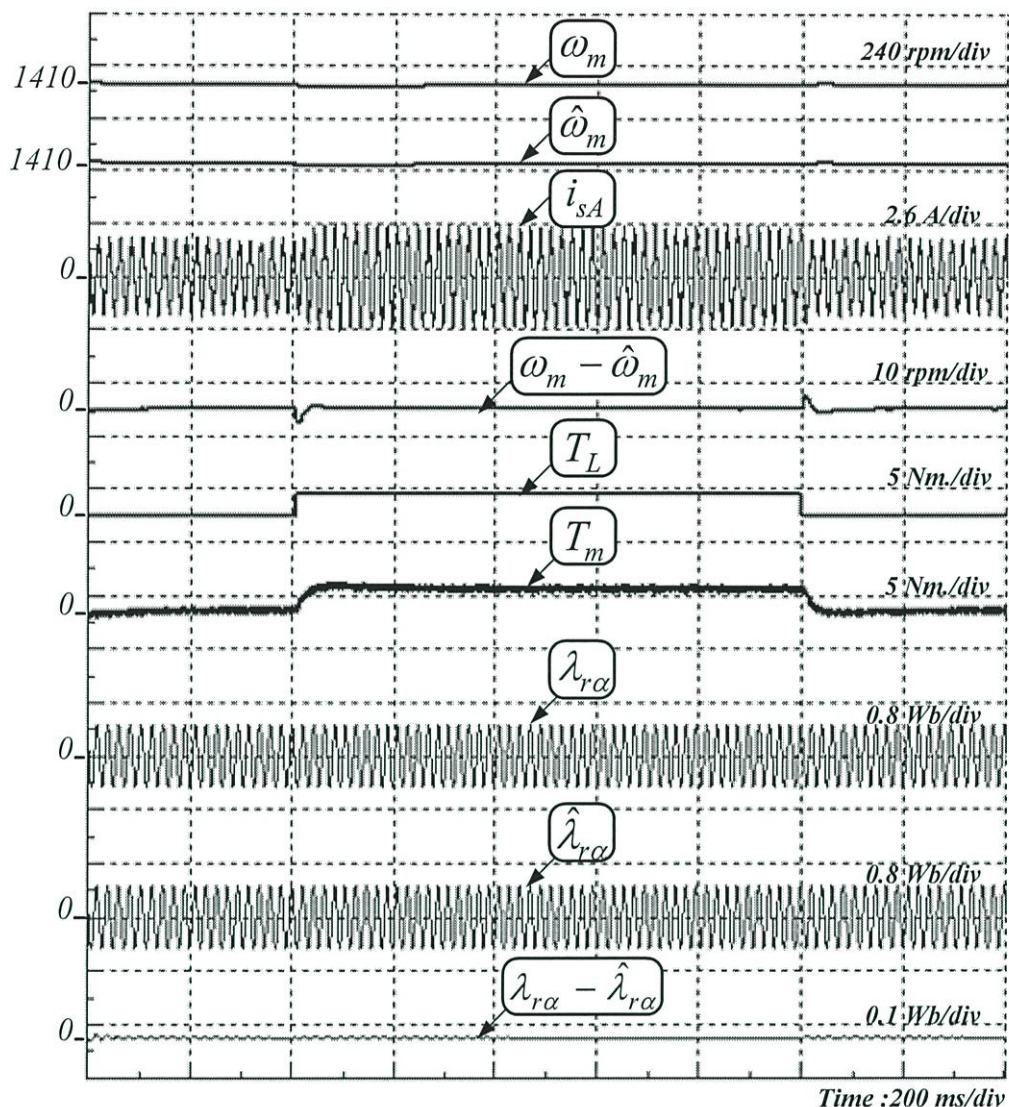
เริ่มเปลี่ยนแปลงความเร็วและแรงบิดกระเพื่อมในขณะมอเตอร์กำลังเข้าสู่สถานะอยู่ตัวซึ่งจะเห็นความแตกต่างอย่างชัดเจนเมื่อเปรียบเทียบกับรูปที่ 4.14 เมื่อ k_f มีค่าเท่ากัน



รูปที่ 4.15 ทดสอบขณะเปลี่ยนแปลงความเร็วของมอเตอร์เมื่อ $\omega_c > \omega_{mr}$ ($k_p = 100$, $k_I = 30000$)

4.5.3 ทดสอบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ที่ความเร็ว 1410 rpm

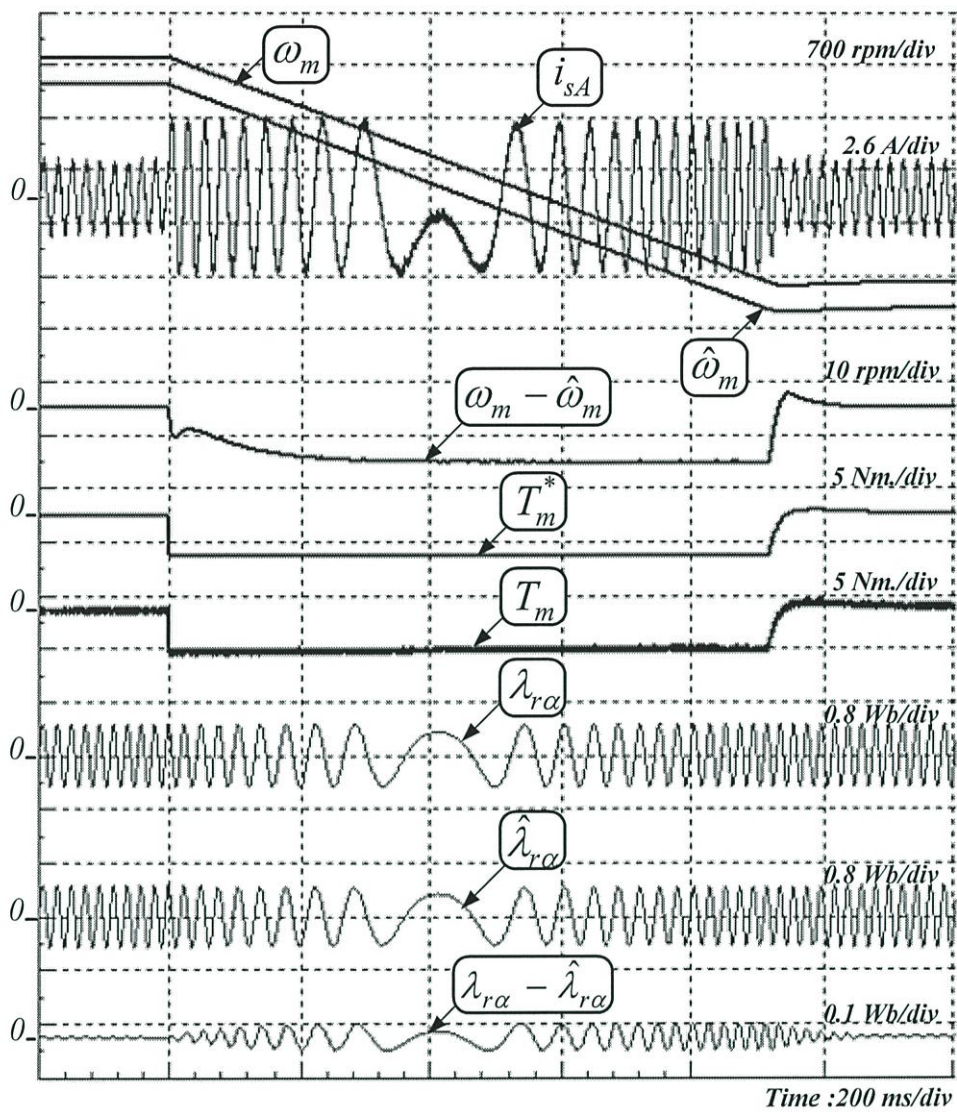
จากรูปที่ 4.16 ทดสอบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ซึ่งในขณะป้อนโหลดทันทีทันใดค่าความเร็วจะมีค่าลดลงเล็กน้อย ส่งผลทำให้เกิดค่าผิดพลาด (Error) ที่เกิดจากการลบกั้ระหว่างความเร็วคำสั่ง (ω_m^*) ที่ใช้ในการควบคุมกับค่าความเร็วของมอเตอร์ภายในวงรอบความเร็ว ซึ่งค่าผิดพลาดดังกล่าวจะทำให้ค่าของแรงบิดควบคุมมีการเปลี่ยนแปลง ผลทำให้มอเตอร์จะดึงกระแสสูงเพื่อสร้างสนามแม่เหล็กและแรงบิดให้เพียงพอกับโหลดที่เพิ่มขึ้นและเมื่อปลดโหลดออก มอเตอร์จะเข้าสู่สถานะอยู่ตัวตามเดิม ค่าความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) เกิดขึ้นในขณะป้อนโหลดและปลดโหลดออกอย่างทันทีทันใดจากนั้นความเร็วผิดพลาดจะเป็นศูนย์อย่างรวดเร็ว



รูปที่ 4.16 ทดสอบขณะป้อนโหลดและปลดโหลดขนาด 2 Nm. อย่างทันทีทันใดที่ความเร็วคงที่ 1410 rpm

4.5.4 ทดสอบขณะกลับทิศความเร็วของมอเตอร์ 1410 ถึง -1410 rpm

จากรูปที่ 4.17 ทดสอบการกลับทิศความเร็วของมอเตอร์ ในช่วงที่มีการกลับทิศความเร็ว ค่าของแรงบิดควบคุมจะมีค่าติดลบ เนื่องจากค่าผิดพลาด (Error) ที่เกิดจากการลบกัณฑ์ภายในวงรอบของความเร็ว มอเตอร์จะดึงกระแสมากเนื่องจากผลต่างในวงรอบความเร็วมีค่าเป็นลบ ส่งผลให้แรงบิดมีค่าเป็นลบไปด้วยมอเตอร์จึงหมุนกลับทิศทาง ในที่นี้ใช้ความเร็วคำสั่งเป็น 1410 ถึง -1410 rpm ค่าความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) เกิดขึ้นในช่วงมอเตอร์กำลังกลับทิศความเร็ว เนื่องจากระบบประมาณค่าความเร็วไม่สามารถจะจัดความเร็วผิดพลาดให้มีค่าเป็นศูนย์ได้ และในช่วงกลับทิศความเร็ว ความเร็วจริงมีลักษณะเป็นแรมป์ ซึ่งงานวิจัยนี้ได้ออกแบบให้รองรับปัญหานี้อยู่แล้ว ซึ่งค่าความเร็วผิดพลาดขณะกลับทิศความเร็วของมอเตอร์จะมีค่าไม่เกินที่ออกแบบไว้ข้างต้น

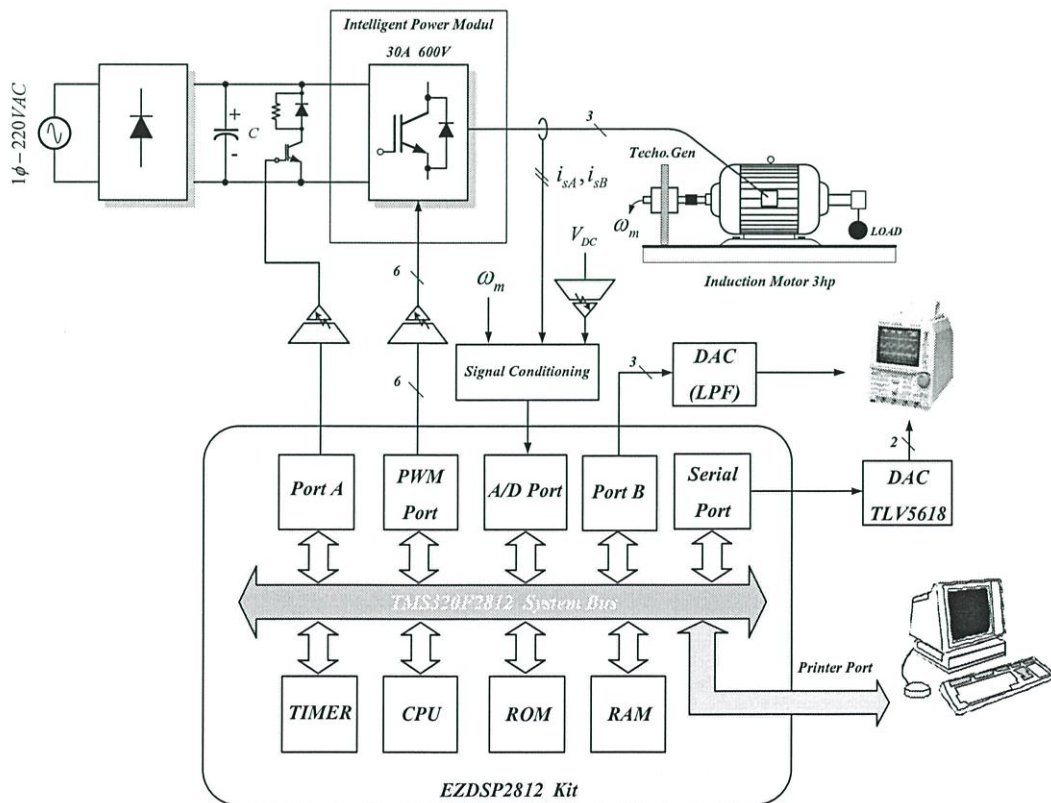


รูปที่ 4.17 ทดสอบขณะกลับทิศความเร็วของมอเตอร์ 1410 ถึง -1410 rpm

บทที่ 5

การสร้างระบบทดสอบจริง

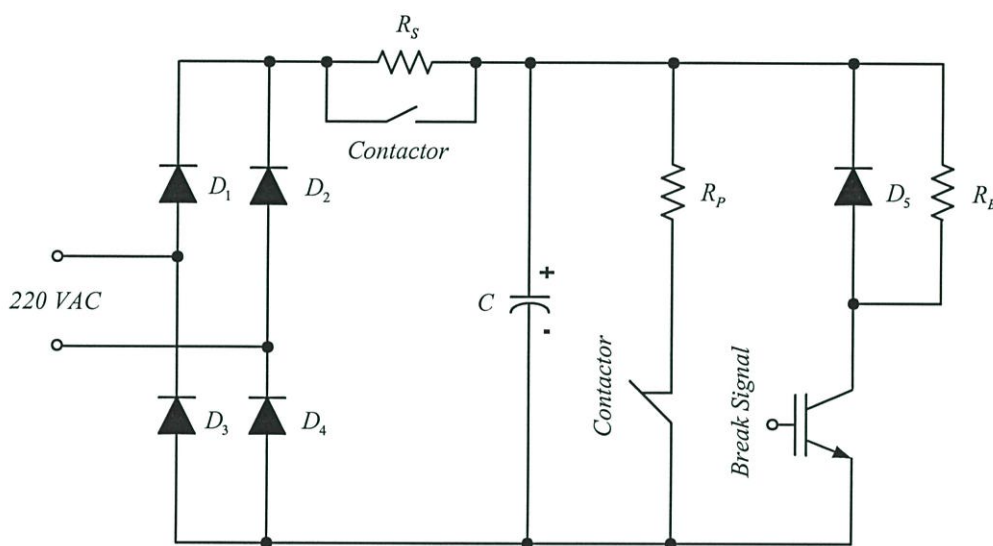
ในบทนี้จะกล่าวถึงการสร้างระบบสำหรับทดสอบจริงที่ใช้ในการทดสอบระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็ว การออกแบบและสร้างชิ้นนี้ประกอบไปด้วยภาคจ่ายกำลังคือ ชุดวงจรคอนเวอร์เตอร์ วงจรกรอง และชุดวงจรอินเวอร์เตอร์ ซึ่งจะนำมาต่อรวมเข้ากับวงจรควบคุมและระบบป้องกันวงจร จากรูปที่ 5.1 จะเป็นการแสดงโครงสร้างของระบบทดสอบจริง โดยวงจรด้านขาเข้าจะต่อกับระบบไฟฟ้ากระแสสลับ 1 เฟส 220 V ผ่านวงจรเรียงกระแสเพื่อแปลงเป็นไฟฟ้ากระแสตรงแล้วผ่านไปยังตัวเก็บประจุเพื่อทำให้ไฟฟ้ากระแสตรงที่ได้มีขนาดที่เรียบ (Voltage source) จากนั้นนำไฟฟ้ากระแสตรงที่ได้ส่งต่อไปยังวงจรอินเวอร์เตอร์เพื่อแปลงจากไฟฟ้ากระแสตรงเป็นไฟฟ้ากระแสสลับที่มีความถี่ต่าง ๆ เพื่อป้อนให้กับมอเตอร์เหนี่ยวนำ โดยกระบวนการทั้งหมดจะควบคุมโดยชุดประมวลผลสัญญาณดิจิทัล (TMS320F2812) ซึ่งตัวประมวลผลดังกล่าวจะมีการทำงานที่เชื่อมต่อกับระบบคอมพิวเตอร์เพื่อใช้ตรวจสอบสถานะการทำงานของบอร์ดที่ใช้ในการควบคุม ซึ่งรายละเอียดของวงจรต่างๆจะกล่าวถึงในหัวข้อต่อไป



รูปที่ 5.1 โครงสร้างของระบบทดสอบจริง

5.1 ชุดวงจรคอนเวอร์เตอร์

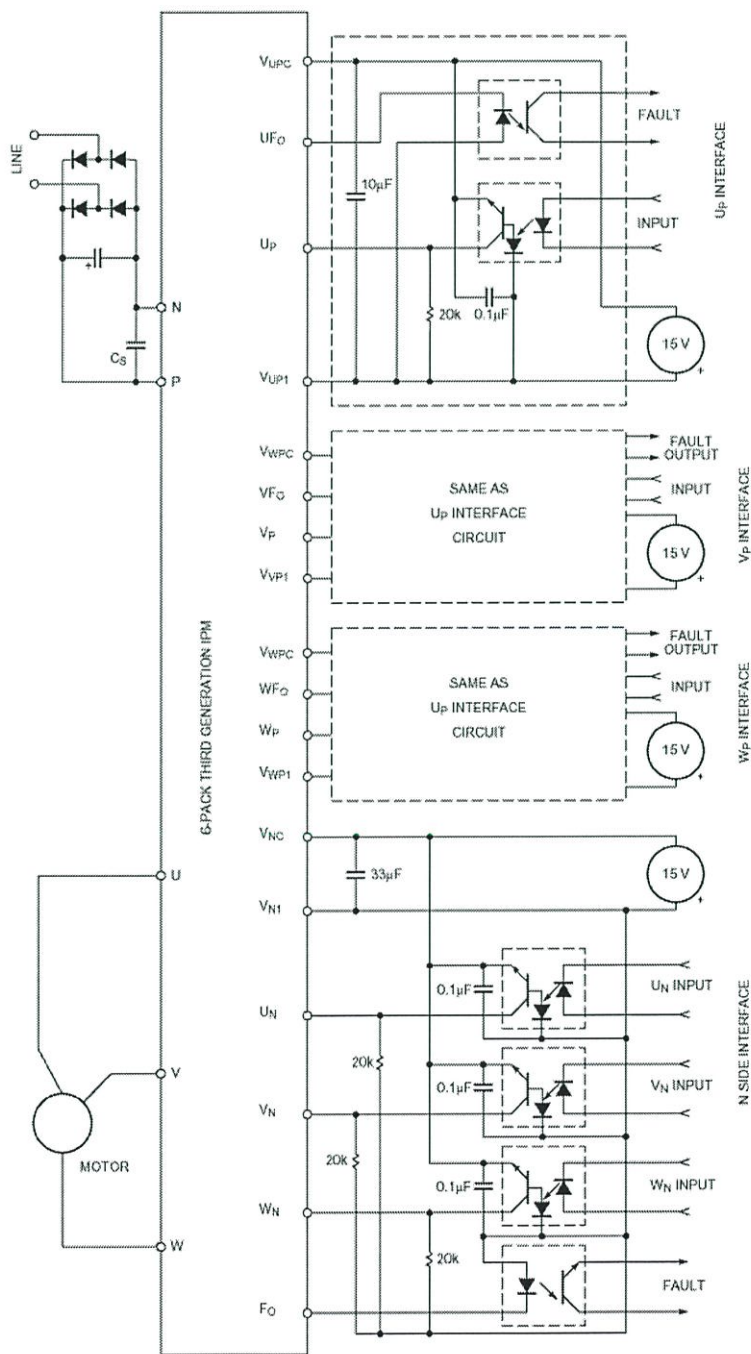
การสร้างชุดคอนเวอร์เตอร์ที่ใช้เป็นแหล่งจ่ายแรงดันให้กับชุดอินเวอร์เตอร์เพื่อใช้ในการควบคุมมอเตอร์ ซึ่งชุดดังกล่าวจะประกอบไปด้วยวงจรเรียงกระแสแบบบริดจ์(400V 25A, Silicon Bridge Rectifier) และวงจรกรองที่ใช้ตัวเก็บประจุโดยรับแรงดันทางด้านอินพุตเป็นไฟฟ้ากระแสสลับ 1 เฟส ขนาดแรงดัน 220 V ความถี่ 50 Hz ซึ่งจะทำให้ได้ค่าแรงดันไฟฟ้ากระแสตรงทางด้านเอาต์พุตประมาณ 310 V ดังแสดงรายละเอียดวงจรไว้ในรูปที่ 5.2



รูปที่ 5.2 วงจรเรียงกระแสแบบบริดจ์และวงจรกรอง

จากรูปที่ 5.2 ความต้านทานที่ต่ออนุกรม (R_s) กับตัวเก็บประจุ จะมีหน้าที่จำกัดกระแสเมื่อเริ่มทำงาน เพราะในช่วงเวลานี้ แรงดันที่ตัวเก็บประจุยังมีค่าประมาณ 0 V ทำให้กระแสที่ไหลอัดประจุให้แก่ตัวเก็บประจุมีค่าค่อนข้างสูง ซึ่งกระแสจำนวนนี้จะไหลผ่านไดโอดของวงจรเรียงกระแสซึ่งอาจจะทำให้ไดโอดเสียหายได้ เมื่อแรงดันคร่อมตัวเก็บประจุมีค่าเพิ่มขึ้นถึงระดับที่เหมาะสมแล้วตัวต้านทานนี้จะถูกตัดวงจรด้วยแมกเนติกคอนแทคเตอร์เพื่อไม่ให้เกิดกำลังสูญเสียที่ค่าความต้านทานที่ต่ออนุกรม (R_s) กับตัวเก็บประจุขณะที่วงจรทำงานและวงจรเรียงกระแสจะได้ส่งผ่านกำลังได้เต็มที่ ส่วนความต้านทานที่ต่อขนาน (R_p) กับตัวเก็บประจุมีหน้าที่ในการคายประจุทิ้งไปเมื่อวงจรหยุดทำงานเพื่อป้องกันมิให้เกิดอันตรายเนื่องจากประจุที่ค้างอยู่ ทั้งนี้เพราะประจุที่ค้างอยู่มีปริมาณมากและมีแรงดันสูง นอกจากนี้ยังประกอบด้วยชุดไดนามิกเบรกกิ้ง (Dynamic braking) เพื่อป้องกันแรงดันตกคร่อมสวิตช์กำลังมีค่าสูงในขณะที่มอเตอร์เปลี่ยนแปลงความเร็วจากความเร็วสูงไปสู่ความเร็วต่ำโดยควบคุมผ่านชุดประมวลผลสัญญาณดิจิทัล

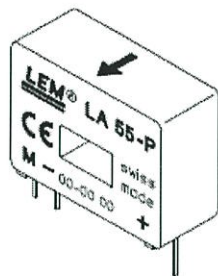
จำนวน 4 ตัว โดยสัญญาณที่มาจากอปได้ 3 ตัวแรกนี้จะมาจากการตรวจจับบริเวณสวิทช์ 3 กิ่งด้านบน ส่วนสัญญาณอีกตัวจะได้จากการตรวจจับจากบริเวณสวิทช์ 3 กิ่งล่าง การทำงานของ วงจรส่งสัญญาณฟอลต์นี้ จะมีการทำงานโดยส่งสัญญาณที่มีคาบความกว้างสัญญาณขนาด 10 ไมโครวินาที ออกมาทางขาสัญญาณฟอลต์ ซึ่งการใช้งานสัญญาณฟอลต์นี้จะต้องอาศัยวงจร ภายนอกในการช่วยล้างสถานะการเกิดเพื่อหยุดสัญญาณที่ใช้ในการควบคุมอินเวอร์เตอร์ต่อไป โดยวงจรภายนอกนี้จะกล่าวถึงในหัวข้อวงจรขั้วนำสวิทช์ต่อไป



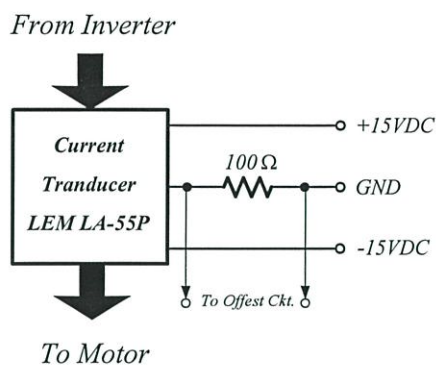
รูปที่ 5.4 วงจรแยกโคดและชุดขั้วนำทรานซิสเตอร์โมดูลกำลังแบบ IGBT

5.3 ชุดวงจรขั้วนำสวิตช์

การควบคุมพัลส์สัญญาณในงานวิจัยนี้จะใช้อุปกรณ์ชุดประมวลผลสัญญาณดิจิทัล (TMS320F2812) ในการควบคุมและสั่งการ ซึ่งสัญญาณขั้วนำที่ได้จาก DSP จะถูกใช้ในการควบคุมการทำงานของอินเวอร์เตอร์ สัญญาณที่ออกจาก DSP ก่อนที่จะนำไปใช้งานจะต้องมีการผ่านวงจรเกตเนื่องจากสัญญาณที่ออกจาก DSP ที่ระดับสัญญาณสูง (High) จะมีค่าอยู่ที่ 2.4 – 3.3 V และระดับต่ำ (Low) จะมีค่าอยู่ 0.1 – 0.4 V หากนำไปใช้ในการควบคุมเกตตระกูล CMOS จะทำให้ประสบปัญหาในเรื่องของระดับสัญญาณได้ ดังนั้นจำเป็นต้องทำให้สัญญาณดังกล่าวมีระดับสัญญาณที่มีความชัดเจนมากขึ้นโดยนำมาผ่านวงจรเกต ซึ่งเกตที่จะใช้จะต้องใช้ตระกูล HCT เท่านั้น เนื่องจากสามารถรับสัญญาณในระดับ TTL แต่ให้สัญญาณทางด้านเอาต์พุตแบบ CMOS (ระดับต่ำ : 0 – 0.1 V และระดับสูง : 4.9 – 5 V) จึงทำให้ระดับสัญญาณที่ใช้ในการขั้วนำสวิตช์มีสถานะที่แน่นอนมากขึ้น สัญญาณที่ได้จะถูกควบคุมให้มีการทำงานที่มีความสัมพันธ์กับวงจรชุดโมดูลกำลังซึ่งตัวอุปกรณ์ดังกล่าวจะมีความสามารถในการส่งสัญญาณพัลส์ฟรอนต์ขนาด 10 ไมโครวินาทีออกมาที่ขาฟรอนต์ได้ในกรณีที่เกิดการลัดวงจรภายในกึ่ง , กระแสเกินหรือมีอุณหภูมิสูง โดยสัญญาณดังกล่าวนี้จะนำไปเข้าออปโตไดโอด TLP521 เพื่อทำการแยกโคดของสัญญาณควบคุม ปกติทางด้านเอาต์พุตของออปโตไดโอด TLP521 จะเปิดวงจรจึงทำให้ขา PRESET ของไอซี 74HCT74 มีสถานะสูง ซึ่งในสภาวะนี้ขาสัญญาณ 1Q จะมีสถานะต่ำ ทำให้ LED ที่แสดงสภาวะปกติ (Normal) สว่าง และขาสัญญาณ 2Q จะมีสถานะสูง ซึ่งขานี้จะต่ออยู่กับเกต 74HCT00 เพื่อใช้ในการควบคุมสัญญาณที่ใช้ในการขั้วนำสวิตช์ชุดอินเวอร์เตอร์ โดยจะสังเกตได้ว่าหากสถานะของขา 2Q ของเกต 74HCT00 มีสถานะสูง มีสัญญาณทริกซ์หรือสัญญาณขั้วนำสวิตช์อินเวอร์เตอร์ที่อยู่ต่อกับอีกขาหนึ่งของเกต 74HCT00 จะสามารถควบคุมการทำงานของออปโตไดโอด TLP550 ได้ แต่ถ้าสถานะของขา 2Q ของเกต 74HCT00 มีสถานะต่ำ จะทำให้สัญญาณทริกซ์หรือสัญญาณขั้วนำสวิตช์อินเวอร์เตอร์ที่อยู่ต่อกับอีกขาหนึ่งของเกต 74HCT00 ไม่สามารถควบคุมการทำงานของออปโตไดโอด TLP550 ได้ ซึ่งขั้นตอนการทำงานในสภาวะที่มีการลัดวงจรสามารถอธิบายได้ดังนี้ หากมีการลัดวงจรในกึ่งของชุดโมดูลกำลังหรือมีการนำกระแสสูงเกินพิกัดของตัวชุดโมดูลกำลังจะมีการส่งสัญญาณฟรอนต์ออกมาทางขาสัญญาณฟรอนต์ของตัวโมดูลทำให้ขา PRESET ของไอซี 74HCT00 มีสถานะต่ำ ส่งผลทำให้ขาสัญญาณ 1Q จะมีสถานะสูง ทำให้ LED ที่แสดงสภาวะปกติ (Normal) ดับลง ในขณะที่ขาสัญญาณ 1Q จะมีสถานะต่ำ ทำให้ LED ที่แสดงสภาวะลัดวงจร (Fault) สว่างขึ้น และช่วงเวลาเดียวกันนี้ขาสัญญาณ 2Q จะมีสถานะต่ำ จึงทำให้สัญญาณที่ไปขั้วชุดอินเวอร์เตอร์หยุดทำงานเนื่องจากคุณสมบัติของแนนเกต (NAND Gate) และหากต้องการเริ่มวงจรใหม่สามารถทำได้โดยการกด RESET เพื่อเป็นการสร้างสัญญาณกระตุ้นเข้าไปเพื่อเปลี่ยนสถานะการทำงานให้กลับเป็นสภาวะปกติต่อไป จากที่กล่าวมาข้างต้นแสดงวงจรที่ใช้งานไว้ดังรูปที่ 5.5



(ก)



(ข)

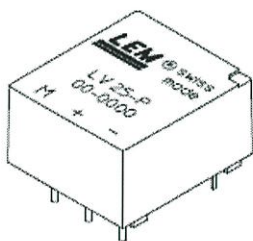
รูปที่ 5.6 วงจรตรวจจับกระแส

(ก) ลักษณะของอุปกรณ์ตรวจจับกระแส LEM รุ่น LA55-P

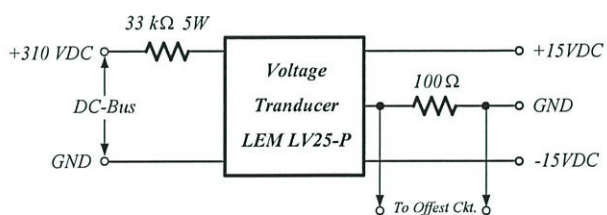
(ข) วงจรตรวจจับกระแสที่ใช้ในงานวิจัย

5.5 ชุดวงจรตรวจจับแรงดัน

วงจรตรวจจับแรงดันนี้ได้ใช้อุปกรณ์ตรวจจับแรงดัน (LEM รุ่น LV25-P) ดังรูปที่ 5.7 ซึ่งสัญญาณทางด้านเอาต์พุตจะอยู่ในรูปของกระแสที่มีค่าอัตราส่วน 2500:1000 ความแม่นยำของการตรวจจับแรงดันขึ้นอยู่กับกำหนดกระแสอินพุต (I_{PN}) ควรออกแบบให้กระแสด้านอินพุตมีค่าเท่ากับ 10 mA เนื่องจากงานวิจัยนี้ใช้แรงดันสูงสุดประมาณ 310 V ดังนั้นจึงเลือกใช้ความต้านทานอินพุต (R_N) เท่ากับ 33 kΩ ขนาด 5 W ดังนั้นสัญญาณทางด้านเอาต์พุตจึงต้องมาผ่านความต้านทานขนาด 100 Ω ซึ่งจะได้เป็นค่าแรงดันเพื่อป้อนเข้าวงจรขยายและยกระดับแรงดันต่อไป



(ก)



(ข)

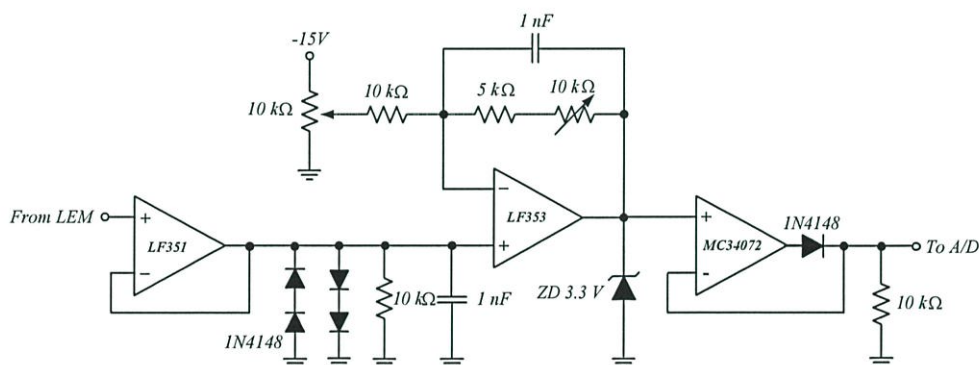
รูปที่ 5.7 วงจรตรวจจับแรงดัน

(ก) ลักษณะของอุปกรณ์ตรวจจับแรงดัน LEM รุ่น LV25-P

(ข) วงจรตรวจจับแรงดันที่ใช้ในงานวิจัย

5.6 ชุดวงจรขยายและยกระดับแรงดัน

วงจรถ่ายและยกระดับแรงดันนี้เป็นวงจรที่ใช้เพื่อทำการยกระดับสัญญาณจากอุปกรณ์ตัวจับกระแส (LEM รุ่น LA55-P) และอุปกรณ์ตรวจจับแรงดัน (LEM รุ่น LV25-P) จากรูปที่ 5.8 ภายในวงจรจะประกอบด้วยส่วนของการขลิบสัญญาณโดยใช้ไดโอด 1N4148 ต่อไขว้กัน เพื่อป้องกันสัญญาณกระชาก (Transient) ที่ส่งไปยังวงจรถ่ายระดับแรงดัน ซึ่งวงจรถ่ายระดับนี้จะทำหน้าที่ยกระดับแรงดันและขยายสัญญาณ 1.5 เท่า เมื่อออกจากวงจรนี้แล้วจะไปเข้าวงจรขลิบสัญญาณทางด้านซีกลับ โดยวงจรนี้จะมีการทำงาน 2 ชั้น คือผ่านวงจรออฟแอมป์ LF353 ซึ่งใช้แหล่งจ่าย $\pm 15\text{ V}$ สัญญาณเอาต์พุตที่ได้ก่อนข้างอินทราัยกับ DSP จึงได้มีการนำสัญญาณดังกล่าวมาผ่านเข้าวงจรออฟแอมป์เบอร์ MC34072 ซึ่งใช้ไฟเลี้ยงขนาด 5 V โดยข้อดีของวงจรที่ใช้ออฟแอมป์เบอร์ MC34072 คือสัญญาณที่ออกทางด้านเอาต์พุตจะมีขนาดไม่เกิน 3.3 V และมีความสามารถในการทำงานดีเท่ากับออฟแอมป์เบอร์ LF351 จึงทำให้วงจรที่ได้รับการออกแบบนี้ค่อนข้างสมบูรณ์และปลอดภัยสำหรับ DSP



รูปที่ 5.8 วงจรถ่ายและยกระดับแรงดัน

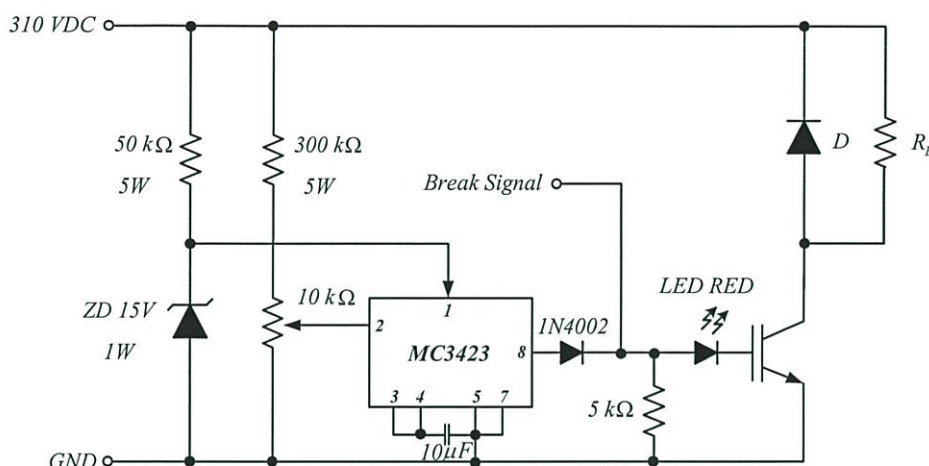
5.7 ชุดวงจรป้องกันต่าง ๆ

ระบบป้องกันวงจรมักเป็นส่วนสำคัญส่วนหนึ่งของเครื่องควบคุมความเร็วมอเตอร์เหนี่ยวนำ โดยระบบป้องกันวงจรจะประกอบไปด้วย วงจรป้องกันกระแสเกิน (Over Current), วงจรป้องกันแรงดันเกิน (Over voltage) โดยวงจรป้องกันที่ออกแบบและสร้างขึ้นนี้สามารถสรุปการทำงานในแต่ละส่วนได้ดังนี้

5.7.1 ชุดวงจรป้องกันแรงดันเกิน

การเกิดสถานะแรงดันเกินนั้นมักจะเกิดเมื่อมอเตอร์มีการลดความเร็วซึ่งการทำงานของวงจรป้องกันแรงดันเกินนี้จะทำหน้าที่ตรวจจับแรงดันจากแหล่งจ่ายไฟกระแสตรง ซึ่งในสภาวะปกติจะมีค่าแรงดันประมาณ 310 V โดยจะให้แรงดันขาออกประมาณ 2.6 V ขึ้นอยู่กับแรงดัน

ของแหล่งจ่ายไฟตรง หากระดับแรงดันทางค่านไฟตรงมีการเปลี่ยนแปลงมากกว่า 315 V ส่วนของวงจรเบรก (Break) จะมีการทำงานโดยวงจรสามารถแสดงได้ดังรูปที่ 5.9



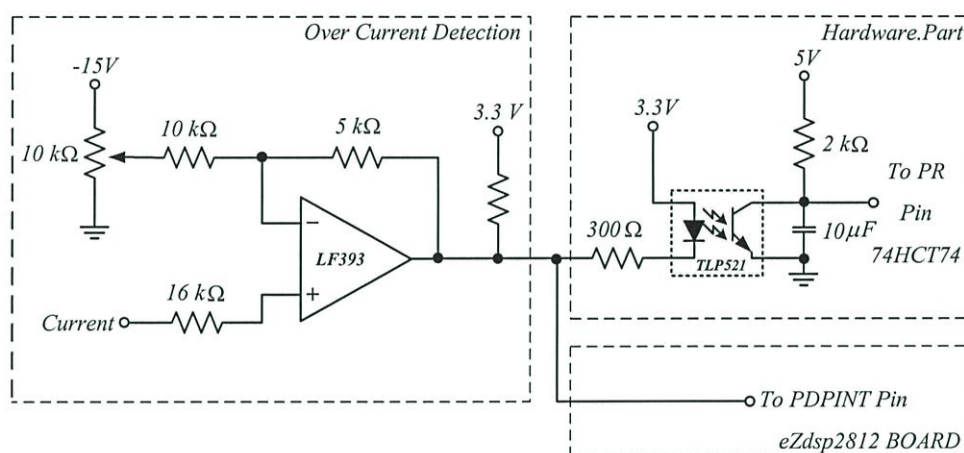
รูปที่ 5.9 วงจรป้องกันแรงดันเกิน

การทำงานของวงจรจะใช้ไอซีเบอร์ MC3423 ซึ่งทำหน้าที่ตรวจจับระดับของสัญญาณที่มาจากภาคการตรวจจับแรงดัน ซึ่งในขณะที่แรงดันเกิน 315 V จะทำให้สัญญาณไฟตรงจากภาคการตรวจจับแรงดันมีการเปลี่ยนแปลงส่งผลทำให้ขาสัญญาณทางค่านเอาต์พุตของไอซีมีการเปลี่ยนสถานะจาก Low เป็น High ซึ่งสัญญาณดังกล่าวจะถูกส่งไปทริกซ์การทำงานของทรานซิสเตอร์ที่ใช้ในการเบรกหรือระบายกระแส ส่วนสัญญาณเบรก (Break Signal) จากรูปที่ 5.9 จะใช้เป็นสัญญาณอินพุตจากอุปกรณ์ภายนอก (ถ้ามีการต่อใช้งาน)

5.7.2 ชุดวงจรป้องกันกระแสเกิน

กรณีกระแสเกินส่วนใหญ่มักจะเกิดขึ้นเมื่อเริ่มเดินเครื่องหรือเพิ่มความถี่ของเครื่องควบคุมมอเตอร์เร็วเกินไปหรือการใส่ภาระให้กับมอเตอร์ที่ไม่เหมาะสมหรืออีกกรณีหนึ่งคือการเกิดการลัดวงจรมอเตอร์ ซึ่งในการตรวจจับกระแสในงานวิจัยนี้จะใช้ Hall Effect Current Sensor (LA55-P) ตรวจจับกระแสในวงจร ซึ่งข้อดีของ Hall Effect นี้คือมีการแยกส่วน (Isolate) ของกระแสที่ใช้ในการตรวจจับสามารถวัดขนาดกระแสได้ถึง 50 A โดยมีการเบี่ยงเบนเชิงเส้นต่ำกว่า 0.15 % และสามารถตอบสนองความถี่ในการทำงานได้ถึง 200 kHz โดยให้มีค่าผิดพลาด ± 0.65 % ที่ระดับการวัดกระแส 50 A อีกทั้งยังสามารถรองรับอัตราการเปลี่ยนแปลงค่าของกระแสต่อเวลา (di/dt) ที่มีค่ามากกว่า 200 A ต่อไมโครวินาทีได้ ซึ่งสัญญาณที่ออกจาก LEM นี้ จะต้องนำไปผ่านวงจรระดับและจำกัดแรงดันดังที่ได้อธิบายมาแล้วในหัวข้อที่ 5.4 สัญญาณที่ผ่านวงจรระดับแรงดันนี้จะถูกนำมาเข้าวงจรตรวจจับกระแสเกินซึ่งอาศัยหลักการเปรียบเทียบกับแรงดันอ้างอิงโดยใช้ออฟแอมป์ LM393 ในการทำงาน จากวงจรในรูปที่ 5.10

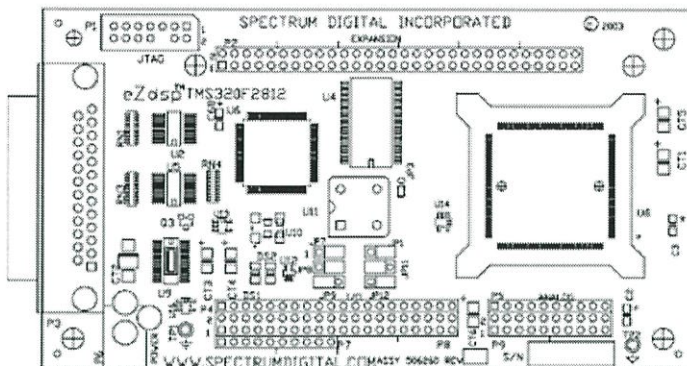
จะสังเกตได้ว่าแรงดันทางด้านเอาต์พุตจะมีขนาดแรงดันการทำงานเพียง 3.3 V เนื่องจากวงจรดังกล่าวนี้ถูกออกแบบมาเพื่อการควบคุมการตัดการทำงานในขณะกระแสเกิน (Over current) หรือลัดวงจร (Fault) ได้ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ โดยทางด้านฮาร์ดแวร์จะเป็นการหยุดพัลส์สัญญาณขับนำของซดอินเวอร์เตอร์โดยอาศัยการควบคุมไอซี 74HCT74 ที่ใช้ในการควบคุม PWM ดังที่ได้กล่าวมาแล้วในหัวข้อวงจรขับนำสวิตช์ ส่วนทางด้านซอฟต์แวร์จะอาศัยขาพอลต์ PDPINT ของ eZdsp2812 ในการทำงาน ซึ่งถ้าขาดังกล่าวนี้มีสถานะต่ำ (Low) เมื่อใด จะทำให้เกิดการอินเทอร์รัพต์ในส่วนของโปรเทคชั่น (Protection) ขึ้น ส่งผลทำให้มีการหยุดการทำงานของซดสัญญาณ PWM อย่างฉับพลันโดยระดับสัญญาณที่ได้จะอยู่ในสถานะ High Impedance ทั้งหมด



รูปที่ 5.10 วงจรป้องกันกระแสเกิน

5.8 ชุดประมวลผลสัญญาณดิจิทัล

เนื่องจากระบบควบคุมมอเตอร์ไฟฟ้ากระแสสลับแบบไร้เซนเซอร์วัดความเร็วมีสมการทางคณิตศาสตร์สลับซับซ้อน ดังนั้นในการคำนวณและประมวลผลจึงต้องใช้หน่วยประมวลผลที่มีประสิทธิภาพและความเร็วสูง ซึ่งในงานวิจัยนี้ใช้บอร์ดสำเร็จรูป DSP Controllers รุ่น eZdsp2812 ของบริษัท Texas Instruments ดังรูปที่ 5.11 ซึ่งมีฟังก์ชันสนับสนุนการทำงานในด้านอิเล็กทรอนิกส์กำลัง (Power electronics) หลายอย่างและเป็นที่แพร่หลายอย่างมาก โดยโครงสร้างของ DSP ประกอบด้วย สัญญาณนาฬิกาที่มีความถี่ เท่ากับ 150 MHz, ROM ขนาด 128K Words 16 bits, RAM ขนาด 128K Words 16 bits, ADC ขนาด 12 bit 16 Channel, I/O-EVA (พอร์ต A) และ I/O-EVB (พอร์ต B) และยังมีพอร์ตอนุกรม(SPI) รวมทั้งยังมีระบบ Watch dog และ Timer ด้วย

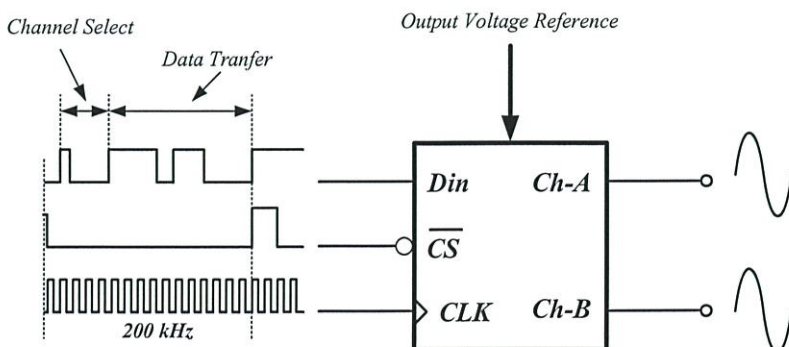


รูปที่ 5.11 ชุดประมวลผลสัญญาณดิจิทัล รุ่น eZdsp2812 ของบริษัท Texas Instruments

สำหรับซอฟต์แวร์ใช้โปรแกรม Code Composer ที่ทำการพัฒนาบนเครื่องคอมพิวเตอร์ จะถูกถ่ายโอนมาที่ RAM บนบอร์ดสำเร็จรูปผ่านพอร์ตขนาน (Parallel port) ซึ่งเป็นตัวเชื่อมระหว่างเครื่องคอมพิวเตอร์กับบอร์ดสำเร็จรูป (DSP Board) และข้อมูลของแรงดันบัสไฟตรง กระแสมอเตอร์และค่าความเร็วของมอเตอร์ที่ตรวจจับมา จะถูกส่งมาที่บอร์ดสำเร็จรูปผ่านทางพอร์ตแบบขนาน (Parallel ADC Port) เพื่อแปลงเป็นสัญญาณดิจิทัลและทำการประมวลผลต่อไปดังรูปที่ 5.1 สำหรับสัญญาณ PWM (PWM1-PWM6) ที่ได้จากตัวประมวลผลจะถูกส่งไปยังอินเวอร์เตอร์ผ่านทาง I/O-EVA โดยสัญญาณทั้งหมดนี้จะต้องผ่านวงจรขั้วนำสวิตช์ก่อนนำไป สวิตช์โมดูลกำลัง (Power Modul)

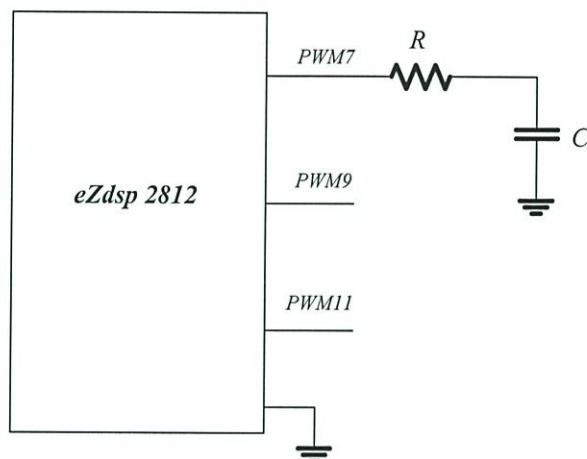
5.9 ชุดการแสดงผลของค่าต่างๆ

ในการแสดงผลจะส่งข้อมูลไปที่ตัวแปลงสัญญาณดิจิทัลเป็นอนาล็อกเบอร์ TLV5618 ของ Texas Instruments ดังรูปที่ 5.12 ซึ่งจะมีเอาต์พุต 2 พอร์ตและใช้การติดต่อที่เป็นสัญญาณอนุกรมแบบซิงโครนัสขนาด 16 บิต โดยที่ 4 บิตหน้าเป็นบิตที่ใช้ในการเลือกช่องสัญญาณออก และ 12 บิตหลังเป็นบิตข้อมูลที่ใช้ส่งออกโดยสามารถตั้งแรงดันเอาต์พุตได้ ในที่นี้จะตั้งให้เอาต์พุตอยู่ระหว่าง 0-4 V



รูปที่ 5.12 การแสดงผลออกของ TLV5618 ของ Texas Instruments

นอกจากนี้บอร์ด eZdsp2812 ยังมีโมดูล PWM_DAC_DRV สำหรับแสดงผลผ่าน I/O-EVB จำนวน 3 ช่อง คือ EPWM4A(PWM7) , EPWM5A(PWM9) และ EPWM6A(PWM11) โดยใช้ความถี่ในการส่งออก 30 kHz สัญญาณที่ได้เป็นแบบดิจิตอล ดังนั้นจึงต้องผ่านวงจรกรองความถี่ต่ำ (Low-pass filter) เพื่อแสดงสัญญาณแอนะล็อกออกมาดังแสดงในรูปที่ 5.13



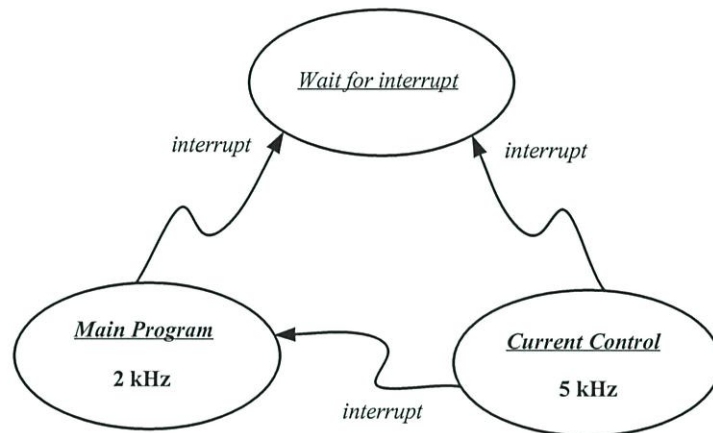
รูปที่ 5.13 การแสดงผลออกโมดูล PWM_DAC_DRV ของ eZdsp2812

5.10 โครงสร้างทางซอฟต์แวร์ที่ใช้ในการควบคุมมอเตอร์

จากโครงสร้างของระบบควบคุมในรูปที่ 4.12 ชุดประมวลผลสัญญาณดิจิตอลจะทำการคำนวณกระแสคำสั่งสร้างแรงบิด (i_{sq}^*) จากผลต่างระหว่างความเร็วคำสั่ง (ω_m^*) และความเร็วประมาณ ($\hat{\omega}_m$) ผ่านตัวควบคุม PI ที่มีการจำกัดขนาดของสัญญาณออกไว้เพื่อไม่ให้เกินกระแสพิสัย กระแสคำสั่ง i_{sq}^* ที่คำนวณได้และกระแสสร้างฟลักซ์ (i_{sd}^*) จะถูกส่งไปคำนวณเพื่อสร้างแรงดันคำสั่ง (v_{sd}^*, v_{sq}^*) โดยใช้ผลต่างของกระแสคำสั่ง (i_{sd}^*, i_{sq}^*) และกระแสจริง (i_{sd}, i_{sq}) ที่ป้อนกลับมาซึ่งถูกแปลงเข้าสู่แกนหมุนโดยใช้มุมของฟลักซ์โรเตอร์ (θ_{mr}) แล้วผ่านตัวควบคุม PI จากนั้นทำการแปลงค่าแรงดันที่ได้ไปอยู่บนแกนนิ่ง ($v_{s\alpha}^*, v_{s\beta}^*$) เพื่อใช้ในการสร้างสัญญาณ PWM ต่อไป ลักษณะการควบคุมเป็นทอดเนื่อง (cascade) โดยการควบคุมความเร็วจะอยู่วงรอบนอก ส่วนการควบคุมกระแสจะอยู่วงรอบใน การควบคุมกระแสนั้นจะต้องมีผลการตอบสนองที่เร็วกว่าการควบคุมความเร็วที่วงรอบนอก ดังนั้นจึงทำการแยกซอฟต์แวร์ในส่วนของ การควบคุมกระแสและสร้างสัญญาณ PWM ออกจากซอฟต์แวร์หลัก โดยให้มีความถี่สวิตซ์เท่ากับ 5 kHz ส่วนซอฟต์แวร์หลักให้มีความถี่เท่ากับ 2 kHz รูปที่ 5.14 แสดงไดอะแกรมสถานะของระบบและการเรียกใช้อินเตอร์รัปต์ โดยทิศทางของลูกศรแสดงความสามารถในการเรียกใช้อินเตอร์รัปต์

ในที่นี้เราจะกำหนดให้ในส่วนของ การควบคุมกระแสมีความสำคัญ (Priority) สูงกว่าตัวโปรแกรมหลัก ซึ่งหมายความว่าถ้าส่วนของการควบคุมกระแสเรียกใช้อินเตอร์รัปต์นั้น ไม่ว่าจะโปรแกรมหลักจะทำงานอยู่หรือไม่ โปรแกรมส่วนการควบคุมกระแสสามารถเรียกใช้บริการ

อินเทอร์รัปต์ได้ และเมื่อทำงานเสร็จก็จะกลับมาทำงานเดิมต่อ ในทางกลับกันถ้าโปรแกรมใน ส่วนของควบคุมกระแสทำงานอยู่นั้น โปรแกรมหลักไม่สามารถเรียกใช้บริการอินเทอร์รัปต์ได้ จนกว่าโปรแกรมควบคุมกระแสจะทำงานเสร็จสิ้น และเมื่อระบบทำงานในส่วนของ โปรแกรมหลักและในส่วนการควบคุมกระแสเสร็จแล้ว โปรแกรมจะไปอยู่ที่สถานะคอยอินเทอร์รัปต์ต่อไป ซอฟต์แวร์ทั้งหมดสามารถเขียนแสดงด้วย PDL (Program Development Language) ในส่วนของไคอะแกรมเวลาแสดงได้ดังรูปที่ 5.15 และ 5.16



รูปที่ 5.14 ไคอะแกรมสถานะและการเรียกใช้อินเทอร์รัปต์

ซอฟต์แวร์ในส่วน of โปรแกรมหลัก

ซอฟต์แวร์โมดูลจะใช้การอินเทอร์รัปต์ทุกๆ 500 ไมโครวินาที และโปรแกรมในการ บริการอินเทอร์รัปต์จะใช้เวลาทั้งหมด 8 ไมโครวินาที

MODULE : Main Program

Initialize

- Initialize all variables
- Initialize all timers
- Clear all variables
- Enable time interrupt

Loop here and wait for interrupt only

Main Program (2k) Interrupt Service Routine

- Clear interrupt flag

Interrupt exception 1

Receive $\lambda_{r\alpha}$, $\lambda_{r\beta}$ and $\hat{\lambda}_{r\alpha}$, $\hat{\lambda}_{r\beta}$ from Current Control Interrupt

Get speed command (ω_m^*)

Get estimate speed ($\hat{\omega}_m$) from previous Main program Interrupt

Speed regulator (PI)

Calculate speed error ($\omega_m^* - \hat{\omega}_m$)

Calculate PI output (i_{sq}^*)

Speed estimation (MRAS)

Calculate flux error ($\lambda_r - \hat{\lambda}_r$)

Calculate PI output ($\hat{\omega}_m$)

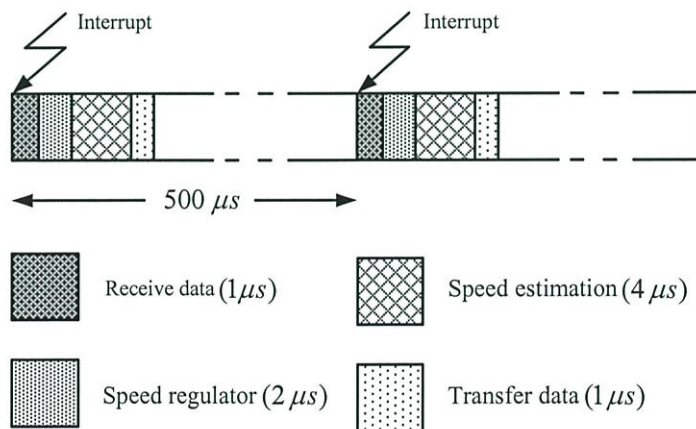
Interrupt exception 2

Transfer i_{sq}^* , $\hat{\omega}_m$ to Current Control Interrupt

Return Interrupt

End of Main Program

.....



รูปที่ 5.15 ไคอะแกรมเวลาในส่วนของโปรแกรมหลัก

ซอฟต์แวร์ในส่วนของการควบคุมกระแส

ซอฟต์แวร์โมดูลจะใช้การอินเทอร์รัปต์ทุกๆ 200 ไมโครวินาที และโปรแกรมในการบริการอินเทอร์รัปต์จะใช้เวลาทั้งหมด 57 ไมโครวินาที

MODULE : Current Control Interrupt

Switching frequency (5k) Interrupt Service Routine

Clear interrupt flag

Read motor current

Input i_{sA}, i_{sB} and V_{DC} from A/D

Get rotor flux angle (θ_{mr}) from previous Current Control Interrupt

Transform to rotating axis

Convert i_{sA}, i_{sB} to i_{sd}, i_{sq}

Get current command (i_{sd}^*, i_{sq}^*)

Current control (PI) d-axis

Calculate current error ($i_{sd}^* - i_{sd}$)

Calculate current output (v_{sd}^*)

Current control (PI) q-axis

Calculate current error ($i_{sq}^* - i_{sq}$)

Calculate current output (v_{sq}^*)

Transform to stationary axis

Convert v_{sd}^*, v_{sq}^* to $v_{s\alpha}^*, v_{s\beta}^*$

Generate PWM signal

Find sector of voltage command

Calculate timing of switching pattern

Calculate stationary voltage

Get estimate speed ($\hat{\omega}_m$) from Main program Interrupt

Calculate rotor flux

Rotor flux ($\lambda_{r\alpha}, \lambda_{r\beta}$) from voltage model

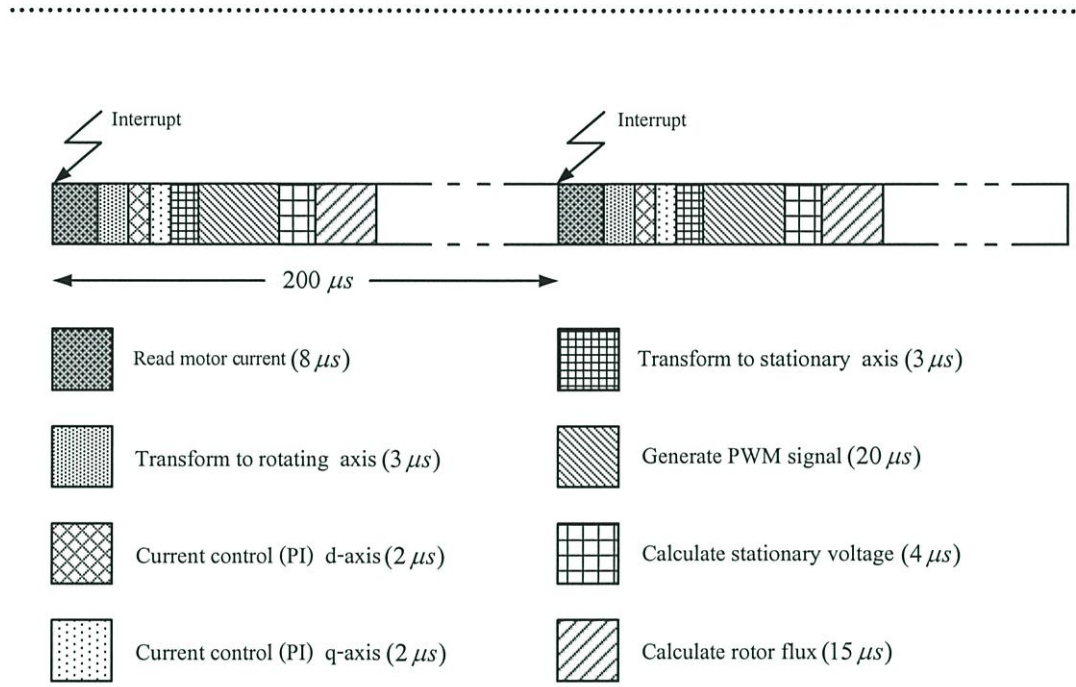
Rotor flux ($\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$) from current model

Compensate rotor flux (PI) in voltage model

Rotor flux angle (θ_{mr})

Return Interrupt

End of Current Control Interrupt



รูปที่ 5.16 ไตอะแกรมเวลาในส่วนของโปรแกรมควบคุมกระแส

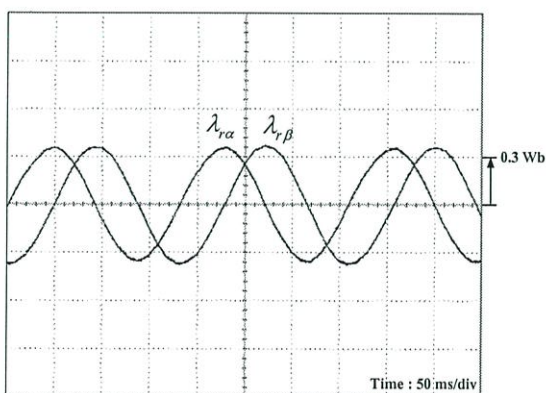
บทที่ 6

ผลการทดสอบการทำงานของระบบจริง

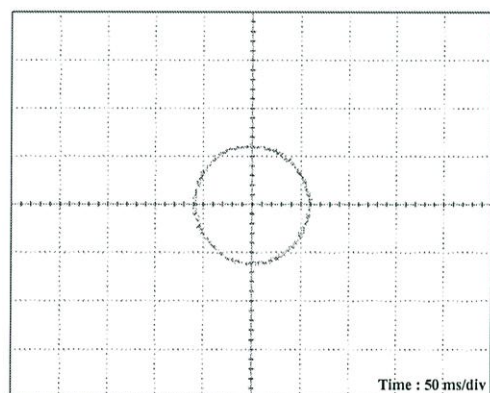
ในบทนี้จะนำวิธีและหลักการที่ได้นำเสนอมาทำการทดสอบกับระบบจริง ด้วยเงื่อนไขการทำงานของระบบในลักษณะต่าง ๆ เพื่อดูผลการตอบสนองของระบบในสถานะอยู่ตัวและสถานะชั่วคราว โดยพารามิเตอร์ของมอเตอร์แสดงไว้ในตารางที่ 4.1 ส่วนอัตราขยายที่ได้ทำการออกแบบไว้สำหรับทดสอบกับระบบแสดงไว้ในตารางที่ 4.2

6.1 ผลการทดสอบของฟลักซ์โรเตอร์จากแบบจำลองแรงดันและแบบจำลองกระแส

รูปที่ 6.1–6.4 แสดงผลการทดสอบของฟลักซ์โรเตอร์ที่ได้จากแบบจำลองแรงดัน (Voltage model) และแบบจำลองกระแส (Current model) โดยทดสอบที่ความถี่ต่ำ (5 Hz) และที่ความถี่สูง (50 Hz) จะเห็นว่าที่ความถี่ต่ำฟลักซ์โรเตอร์ที่ได้จากแบบจำลองแรงดันคำนวณได้ไม่ถูกต้องนัก แต่ที่ความถี่สูงสามารถคำนวณฟลักซ์โรเตอร์ได้อย่างถูกต้อง ในขณะที่ความถี่ต่ำฟลักซ์โรเตอร์ที่ได้จากแบบจำลองกระแสสามารถคำนวณได้ถูกต้องกว่าที่ความถี่สูง เนื่องจากที่ความถี่ต่ำแรงเคลื่อนไฟฟ้าเหนี่ยวนำที่คำนวณโดยแบบจำลองแรงดันมีค่าต่ำมากทำให้เกิดการคำนวณฟลักซ์โรเตอร์ผิดพลาดได้ ดังนั้นที่ความถี่ต่ำแบบจำลองกระแสจึงมีบทบาทมากในการคำนวณฟลักซ์โรเตอร์ ส่วนที่ความถี่สูงแบบจำลองแรงดันจะมีบทบาทสำคัญในการคำนวณฟลักซ์โรเตอร์

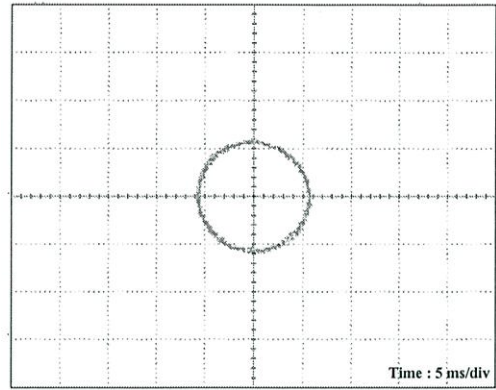
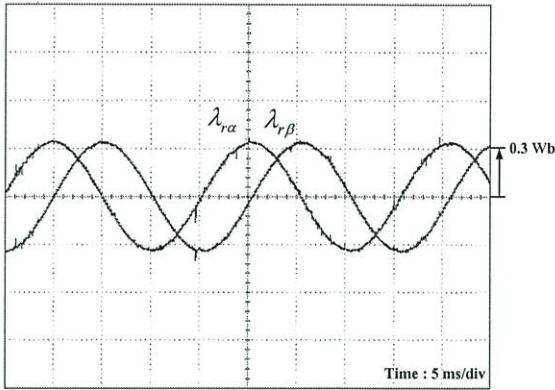


(ก) รูปคลื่นฟลักซ์ $\lambda_{ra}, \lambda_{rb}$ ที่ความถี่ 5 Hz



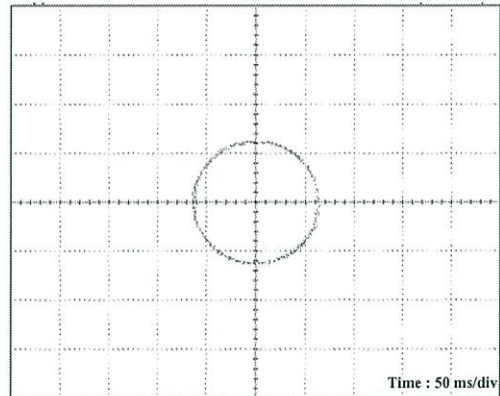
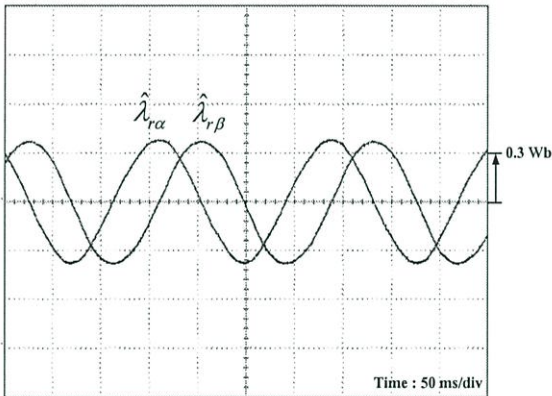
(ข) ทางเดินฟลักซ์ $\lambda_{ra}, \lambda_{rb}$ ที่ความถี่ 5 Hz

รูปที่ 6.1 ฟลักซ์โรเตอร์จากแบบจำลองแรงดัน (Voltage model) ที่ความถี่ 5 Hz



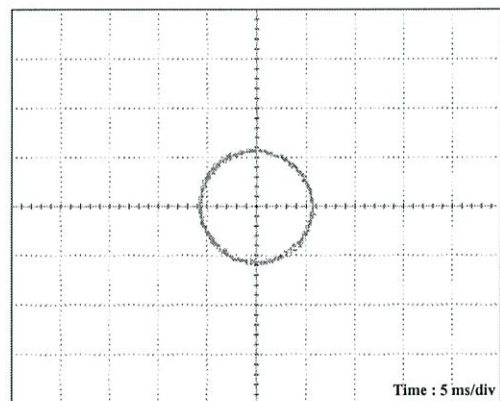
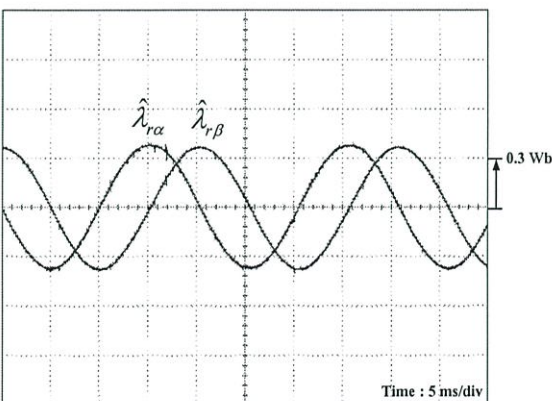
(ก) รูปคลื่นฟลักซ์ $\lambda_{r\alpha}, \lambda_{r\beta}$ ที่ความถี่ 50 Hz (ข) ทางเดินฟลักซ์ $\lambda_{r\alpha}, \lambda_{r\beta}$ ที่ความถี่ 50 Hz

รูปที่ 6.2 ฟลักซ์โรเตอร์จากแบบจำลองแรงดัน (Voltage model) ที่ความถี่ 50 Hz



(ก) รูปคลื่นฟลักซ์ $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ ที่ความถี่ 5 Hz (ข) ทางเดินฟลักซ์ $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ ที่ความถี่ 5 Hz

รูปที่ 6.3 ฟลักซ์โรเตอร์จากแบบจำลองกระแส (Current model) ที่ความถี่ 5 Hz

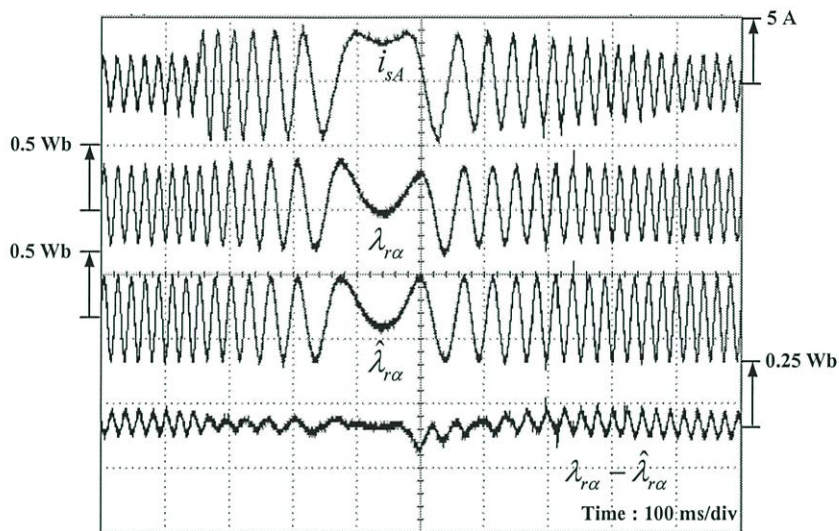


(ก) รูปคลื่นฟลักซ์ $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ ที่ความถี่ 50 Hz (ข) ทางเดินฟลักซ์ $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ ที่ความถี่ 50 Hz

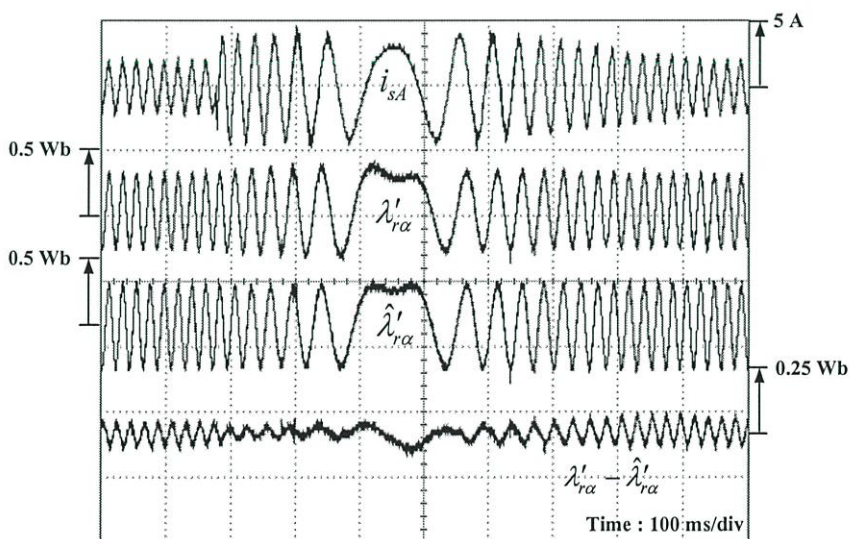
รูปที่ 6.4 ฟลักซ์โรเตอร์จากแบบจำลองกระแส (Current model) ที่ความถี่ 50 Hz

6.2 ผลการทดสอบของฟลักซ์โรเตอร์จากแบบจำลองกระแสก่อนและหลังบูรณาการ

รูปที่ 6.5 แสดงผลการทดสอบฟลักซ์โรเตอร์ ($\lambda_{r\alpha}$) ที่ได้จากแบบจำลองกระแสก่อนบูรณาการร่วมกับระบบประมาณค่าความเร็ว และรูปที่ 6.6 เป็นฟลักซ์โรเตอร์ ($\lambda'_{r\alpha}$) ที่ได้จากแบบจำลองกระแสหลังจากการบูรณาการร่วมกับระบบประมาณค่าความเร็ว จะเห็นว่าฟลักซ์โรเตอร์ที่ได้หลังจากบูรณาการแล้วระบบประมาณค่าฟลักซ์โรเตอร์ยังคำนวณได้ถูกต้องเหมือนเดิม แสดงให้เห็นว่าเราสามารถบูรณาการระบบทั้งสองเข้าด้วยกันได้และไม่เกิดการคำนวณค่าฟลักซ์โรเตอร์ผิดพลาดแต่อย่างใด ซึ่งสามารถลดความซ้ำซ้อนของแบบจำลองได้



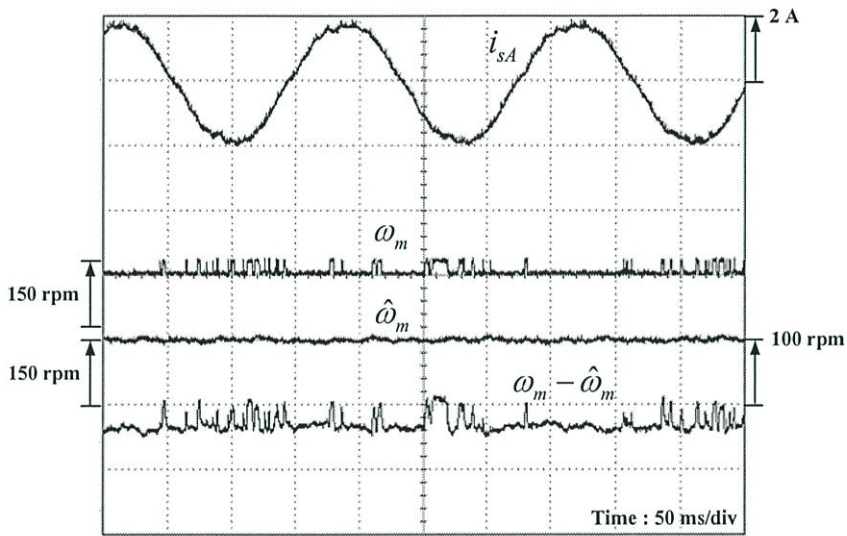
รูปที่ 6.5 รูปคลื่นของฟลักซ์โรเตอร์จากแบบจำลองกระแส(Current model) ก่อนบูรณาการเข้ากับระบบประมาณค่าความเร็ว



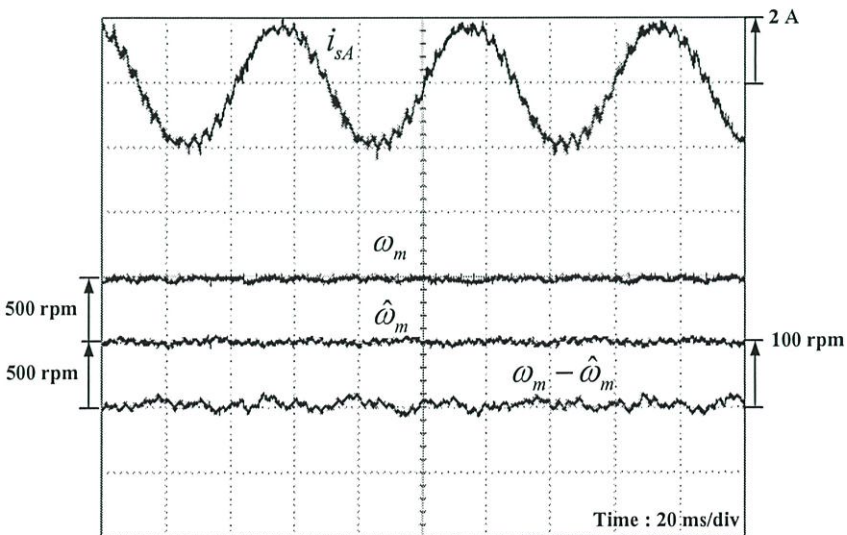
รูปที่ 6.6 รูปคลื่นของฟลักซ์โรเตอร์จากแบบจำลองกระแส(Current model) หลังบูรณาการเข้ากับระบบประมาณค่าความเร็ว

6.3 ผลการทดสอบในสถานะอยู่ตัว

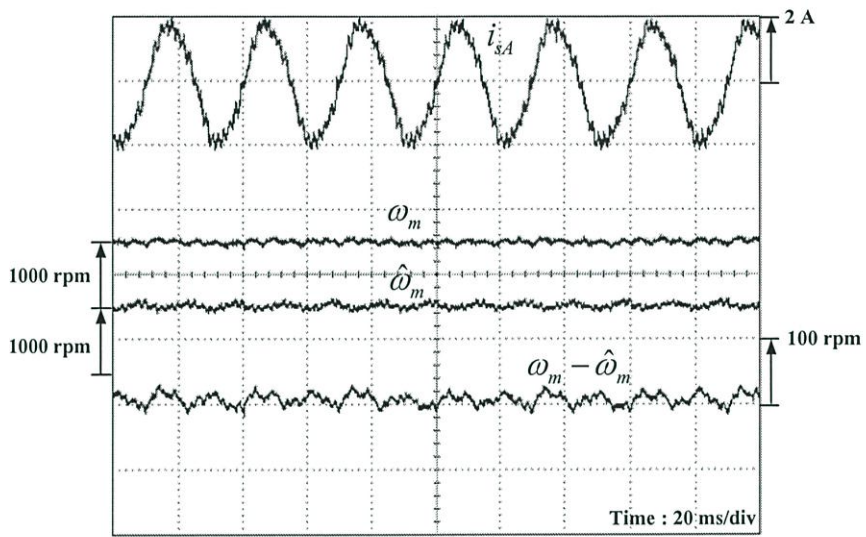
รูปที่ 6.7 - 6.12 แสดงผลการทำงานของระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วที่ความเร็วรอบต่างๆ คือ 150, 500, 1000, 1200, 1410, และ 1500 rpm ตามลำดับ ระบบสามารถประมาณความเร็วได้อย่างถูกต้องและมีระลอกของค่าความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) ที่ความเร็วสูง (1410, 1500 rpm) เพียงเล็กน้อย ส่วนที่ความเร็วต่ำ (150 rpm) ระลอกของค่าความเร็วผิดพลาดมีค่ามาก เนื่องจากงานวิจัยนี้ได้ใช้ Techogenerator ในการป้อนกลับความเร็วจริง (ω_m) มาเปรียบเทียบกับความเร็วประมาณ ($\hat{\omega}_m$) ซึ่งที่ความเร็วต่ำ Techogenerator ไม่สามารถให้กำเนิดแรงดันได้ถูกต้องเท่ากับที่ความเร็วสูงได้



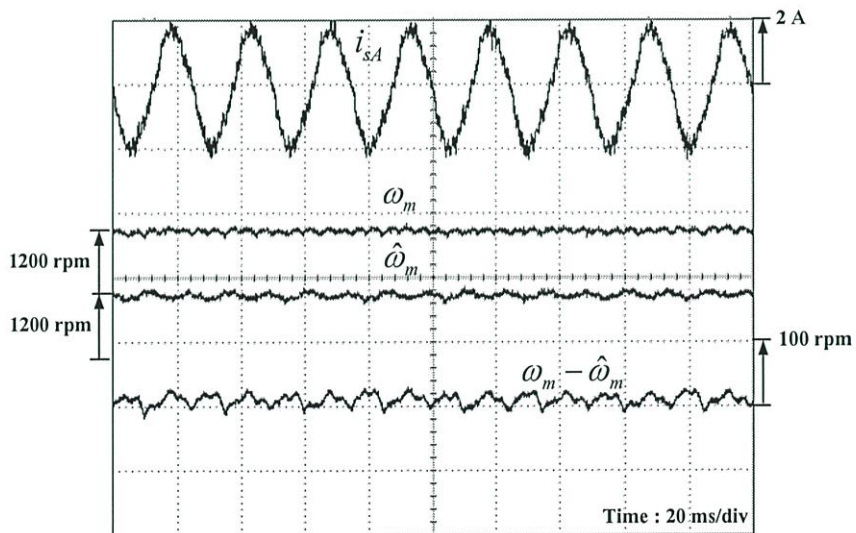
รูปที่ 6.7 ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 150 rpm



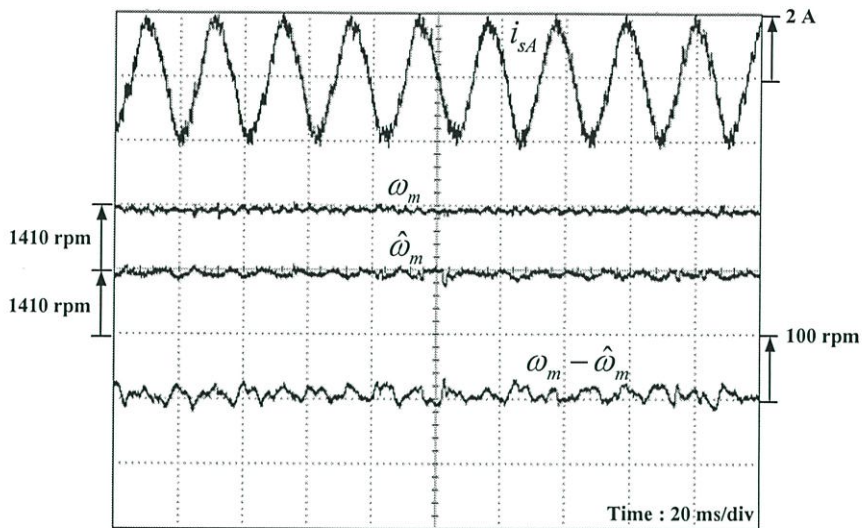
รูปที่ 6.8 ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 500 rpm



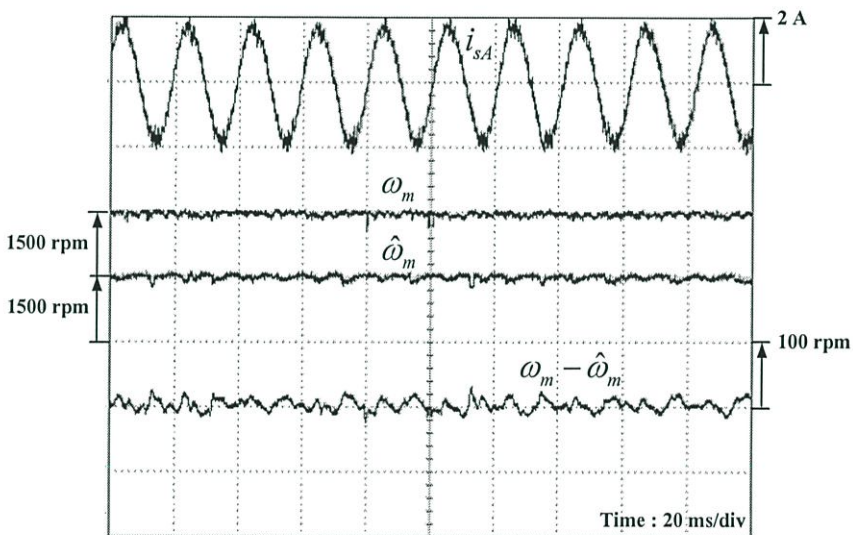
รูปที่ 6.9 ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1000 rpm



รูปที่ 6.10 ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1200 rpm



รูปที่ 6.11 ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1410 rpm

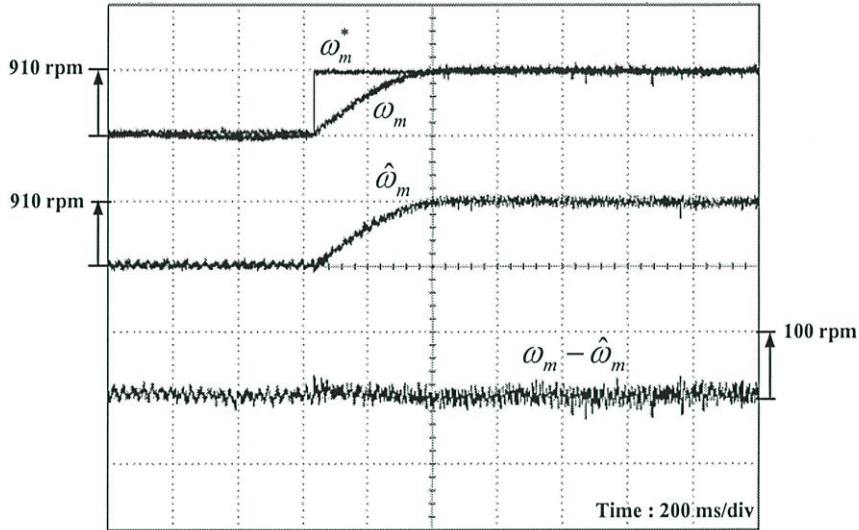


รูปที่ 6.12 ผลการทำงานของระบบที่สถานะอยู่ตัวที่ความเร็วคำสั่ง 1500 rpm

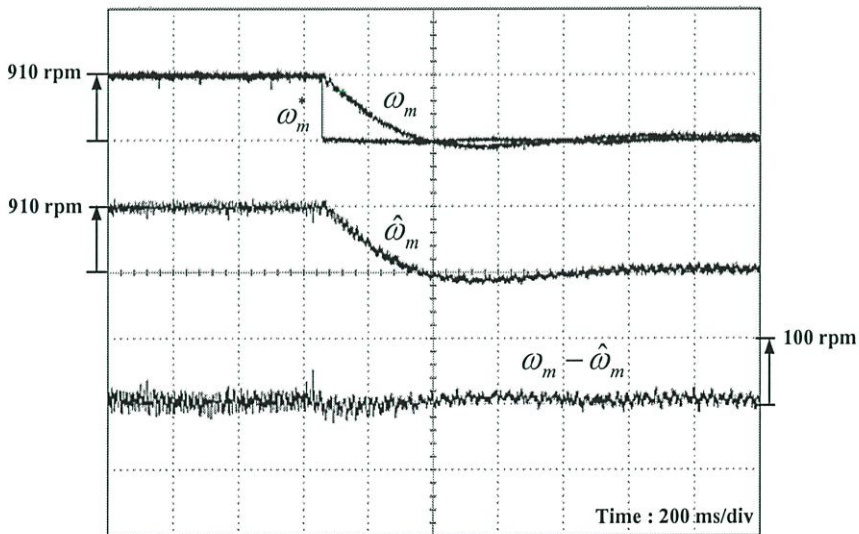
6.4 ผลการทดสอบในสถานะชั่วคราว

จากรูปที่ 6.13 และ 6.14 เราทดสอบการทำงานของระบบควบคุมความเร็วโดยทำการเปลี่ยนความเร็วจาก 500 rpm ไป 1410 rpm และเปลี่ยนความเร็วจาก 1410 rpm ไป 500 rpm โดยใช้ความเร็วคำสั่ง (ω_m^*) เป็นแบบขั้น ส่วนรูปที่ 6.15 และ 6.16 แสดงการกลับทิศความเร็วของมอเตอร์ จะเห็นว่าความเร็วประมาณ ($\hat{\omega}_m$) ติดตามความเร็วจริง (ω_m) ได้อย่างถูกต้องและมีค่าความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) ของความเร็วในสถานะชั่วคราวเล็กน้อย จากรูปที่ 6.13 และ 6.14 พบ

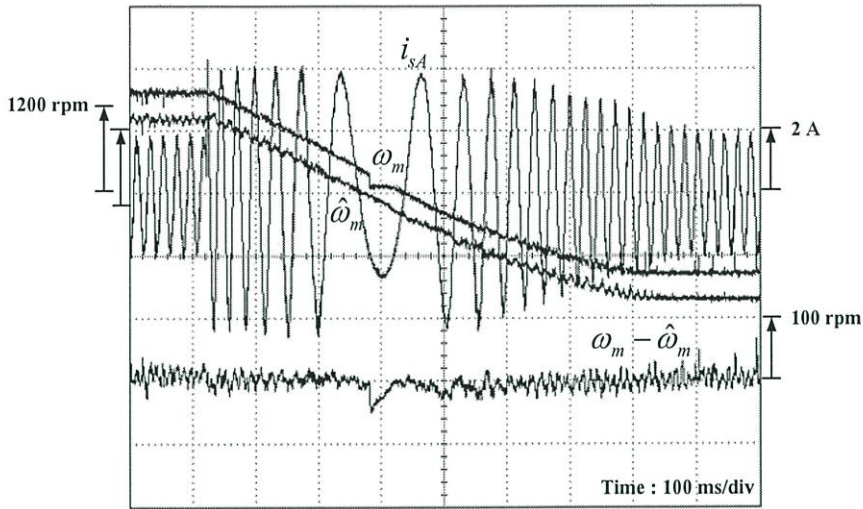
ว่าผลการตอบสนองต่อความเร็วคำสั่งหรือเวลาขึ้น (Rise time) ประมาณ 300 ms และใช้เวลาเข้าสู่ค่าความเร็วคำสั่ง (Settling time) ประมาณ 500 ms ซึ่งมีการตอบสนองใกล้เคียงกับการจำลองโดยใช้โปรแกรมคอมพิวเตอร์ดังแสดงในรูปที่ 4.14 ส่วนรูปที่ 6.15 ใช้เวลาในการกลับทิศความเร็วประมาณ 800 ms ซึ่งมีการตอบสนองใกล้เคียงกับการจำลองในรูปที่ 4.17 เช่นกัน



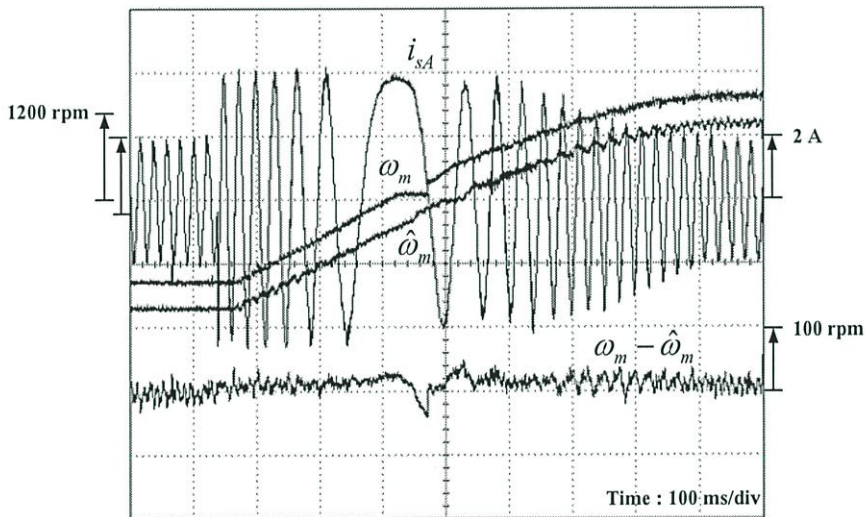
รูปที่ 6.13 ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 500 rpm → 1410 rpm



รูปที่ 6.14 ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 1410 rpm → 500 rpm



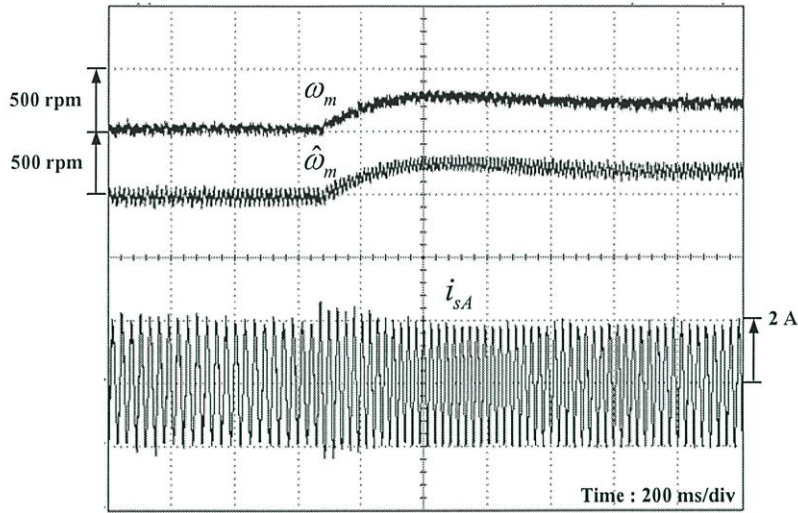
รูปที่ 6.15 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก 1410 rpm \rightarrow -1410 rpm
เมื่อใช้อัตราขยาย $k_p = 600$, $k_I = 30000$



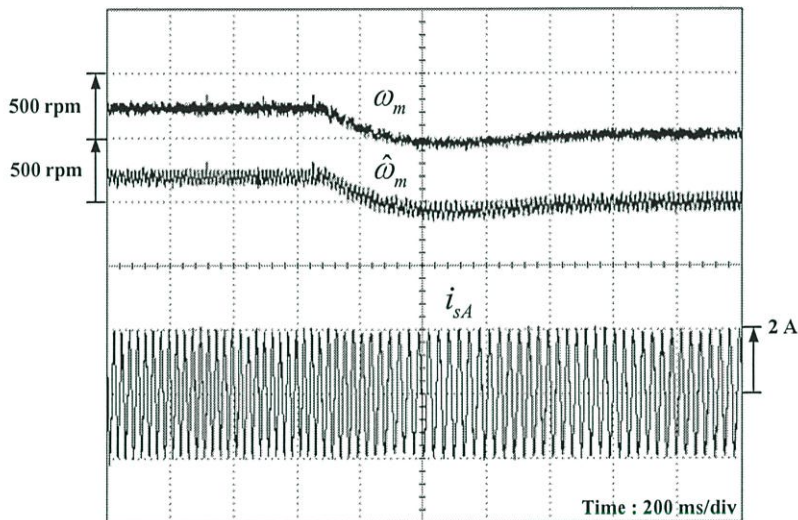
รูปที่ 6.16 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก -1410 rpm \rightarrow 1410 rpm
เมื่อใช้อัตราขยาย $k_p = 600$, $k_I = 30000$

6.5 ผลการทดสอบการตอบสนองต่อคำสั่งความเร็วในช่วงแคบ

รูปที่ 6.17 และ 6.18 เป็นผลการทดลองการทำงานของระบบในกรณีเปลี่ยนแปลงความเร็วคำสั่ง (ω_m^*) ในช่วงแคบ จาก 1000 rpm ไป 1200 rpm และจาก 1200 rpm ไป 1000 rpm ตามลำดับในขณะที่ไม่มีโหลด พบว่าผลการตอบสนองต่อความเร็วคำสั่งหรือเวลาขึ้น (Rise time) ประมาณ 300 ms และใช้เวลาเข้าสู่ค่าความเร็วคำสั่ง (Settling time) ประมาณ 500 ms



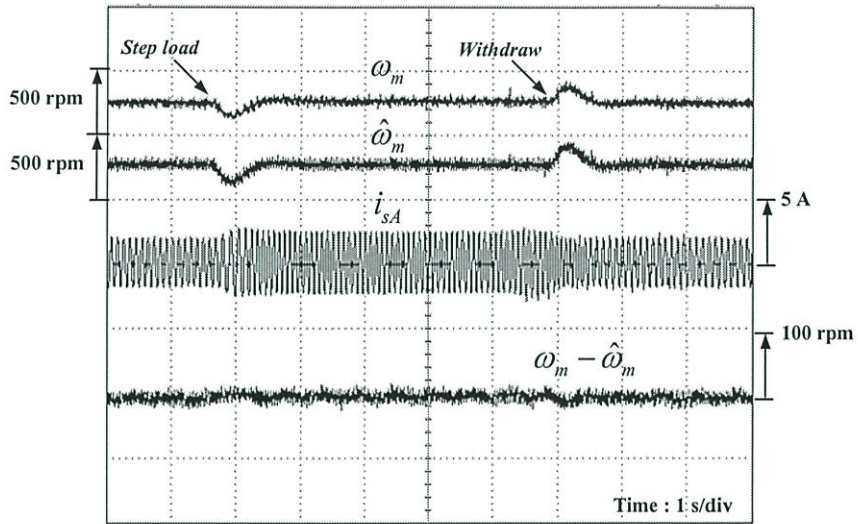
รูปที่ 6.17 ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 1000 rpm → 1200 rpm



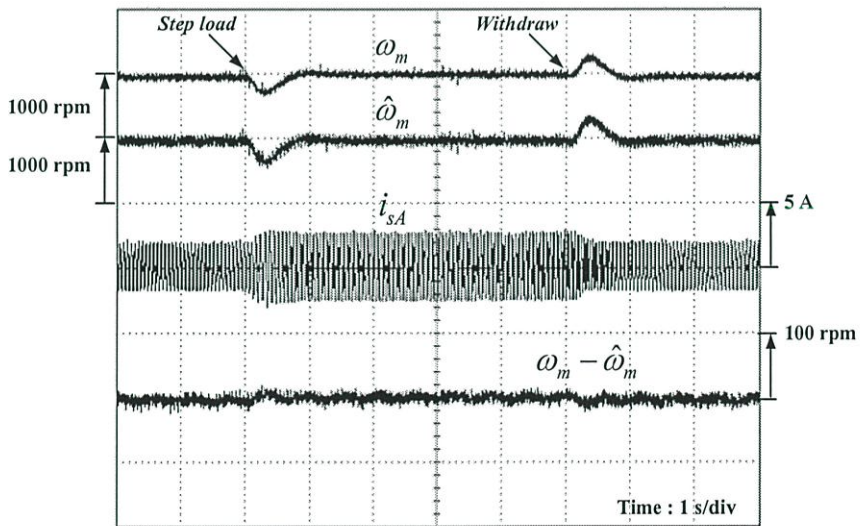
รูปที่ 6.18 ผลการทำงานของระบบขณะเปลี่ยนความเร็วจาก 1200 rpm → 1000 rpm

6.6 ผลการทดสอบการตอบสนองต่อโหลดแบบขั้น

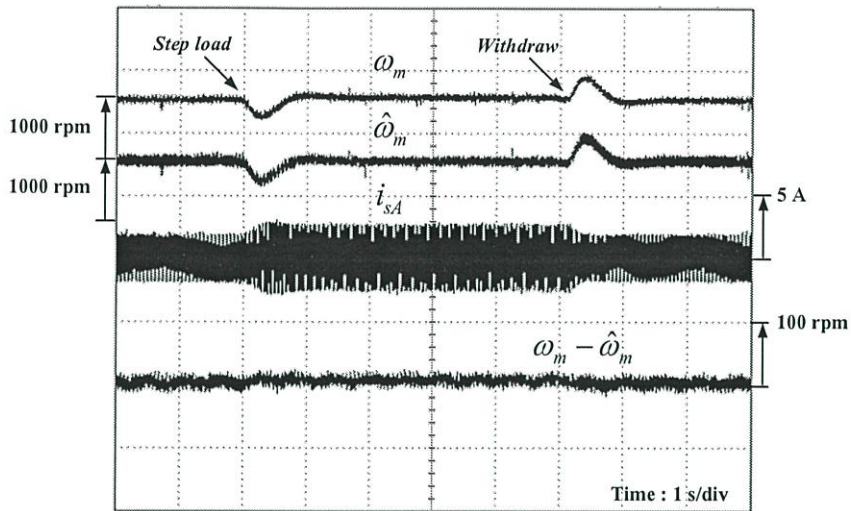
รูปที่ 6.19 - 6.22 เป็นผลการทดลองการทำงานของระบบในกรณีเปลี่ยนแปลงโหลดแบบขั้นที่ความเร็ว 300, 500, 1000 และ 1410 rpm ตามลำดับ โดยป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ให้กับมอเตอร์ ระบบสามารถควบคุมความเร็วได้อย่างถูกต้องและมีเสถียรภาพ ความเร็วประมาณ ($\hat{\omega}_m$) ติดตามความเร็วจริง (ω_m) ได้ถูกต้องเช่นกัน และมีเวลาในการตอบสนองของความเร็วที่ช้าที่สุดประมาณ 800 ms ในการกลับเข้าสู่ภาวะอยู่ตัว



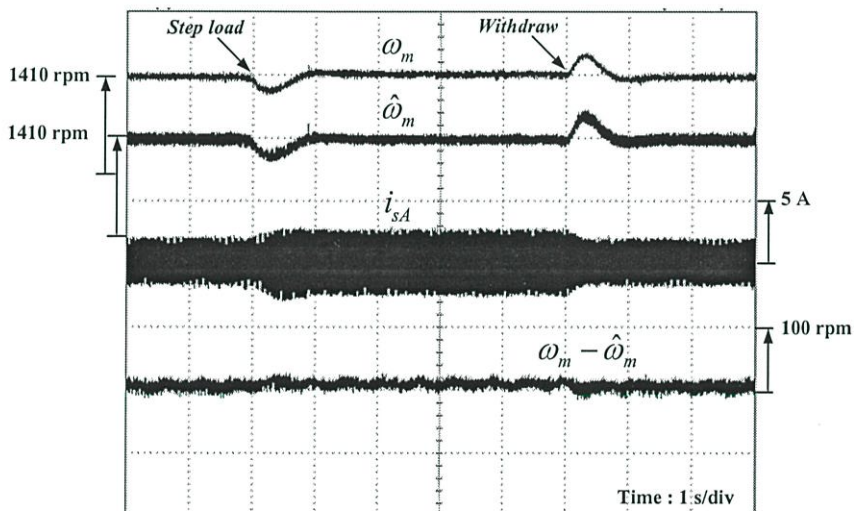
รูปที่ 6.19 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีที่ทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 300 rpm



รูปที่ 6.20 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีที่ทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 500 rpm



รูปที่ 6.21 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 1000 rpm

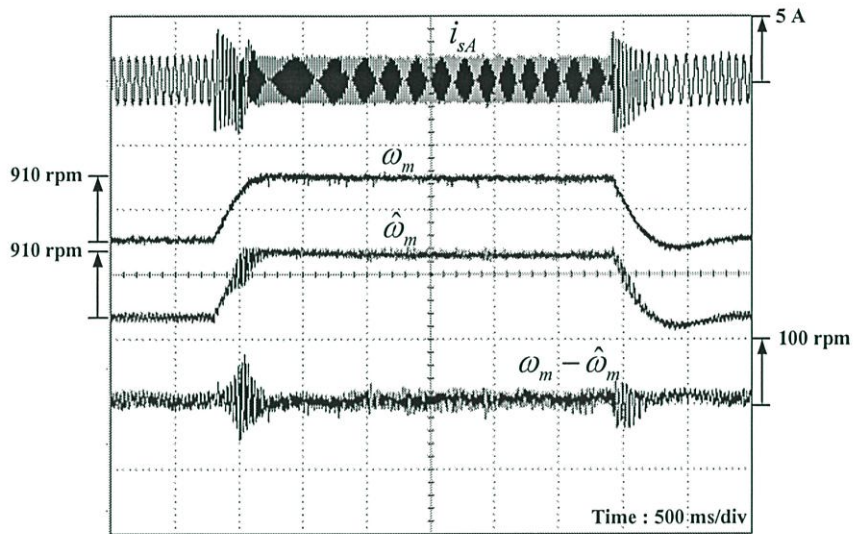


รูปที่ 6.22 ผลการทำงานของระบบขณะป้อนโหลดและปลดโหลดอย่างทันทีทันใดขนาด 2 Nm. ที่ความเร็วคำสั่ง 1410 rpm

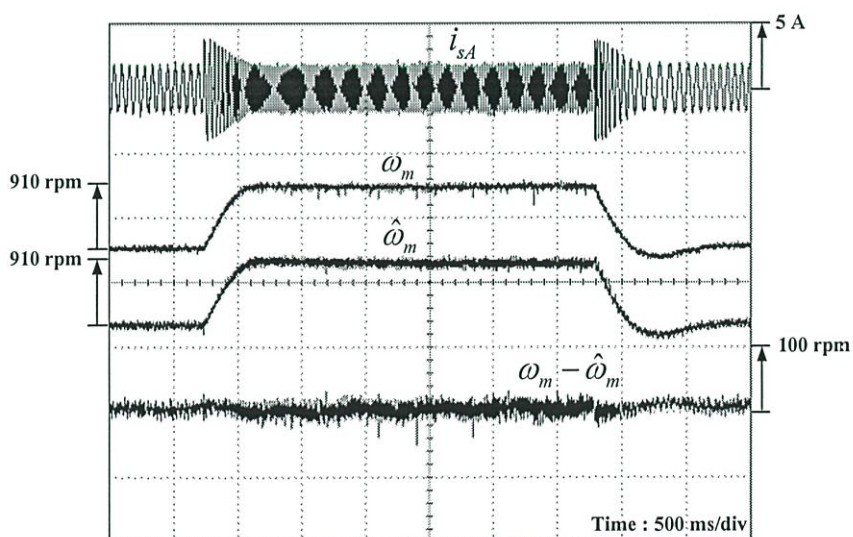
6.7 ผลการทดสอบของอัตราขยาย k_p

ทำการทดสอบความมีเสถียรภาพการทำงานของระบบประมาณค่าความเร็วสำหรับการเลือกอัตราขยาย k_p โดยให้มอเตอร์ทำงานขณะไม่มีโหลด โดยการเปลี่ยนความเร็วจาก 500 rpm ไป 1410 rpm และเปลี่ยนความเร็วจาก 1410 rpm ไป 500 rpm ความเร็วคำสั่งเป็นแบบขั้น โดย

ใช้อัตราขยาย $k_p = 100$, $k_I = 30000$ จากการทดสอบในรูปที่ 6.23 จะเห็นว่าความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) มีค่ามากในขณะที่มอเตอร์เริ่มเปลี่ยนแปลงความเร็วและขณะมอเตอร์กำลังเข้าสู่สภาวะอยู่ตัว ส่วนรูปที่ 6.24 เราเลือก $k_p = 600$, $k_I = 30000$ (ตามที่ได้ออกแบบไว้) จะเห็นว่าความเร็วผิดพลาดมีค่าลดลง แสดงให้เห็นว่าถ้าเราเลือกความถี่หักมุม (ω_c) ใกล้เคียงหรือมากกว่าความถี่ของฟลักซ์โรเตอร์ (ω_{mr}) จะทำให้อัตราขยาย k_p ลดลงซึ่งทำให้ระบบแกว่งในที่สุด



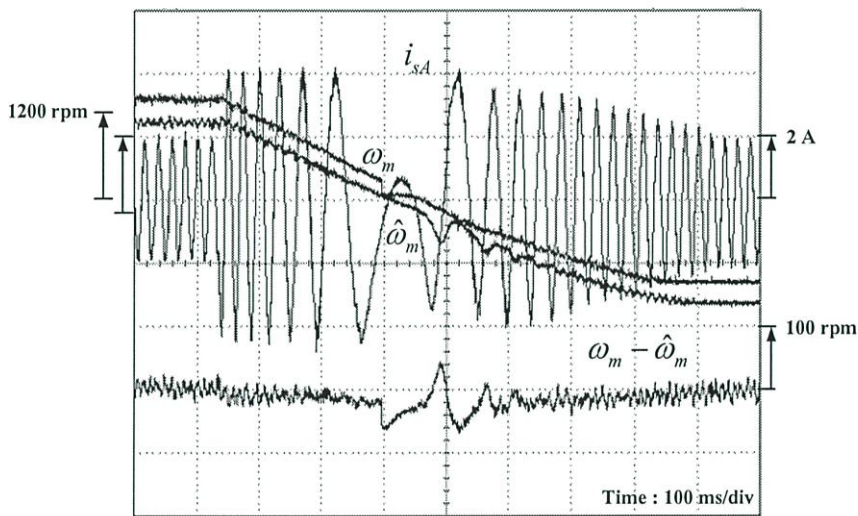
รูปที่ 6.23 ผลการทำงานของระบบ เมื่อใช้อัตราขยาย $k_p = 100$, $k_I = 30000$



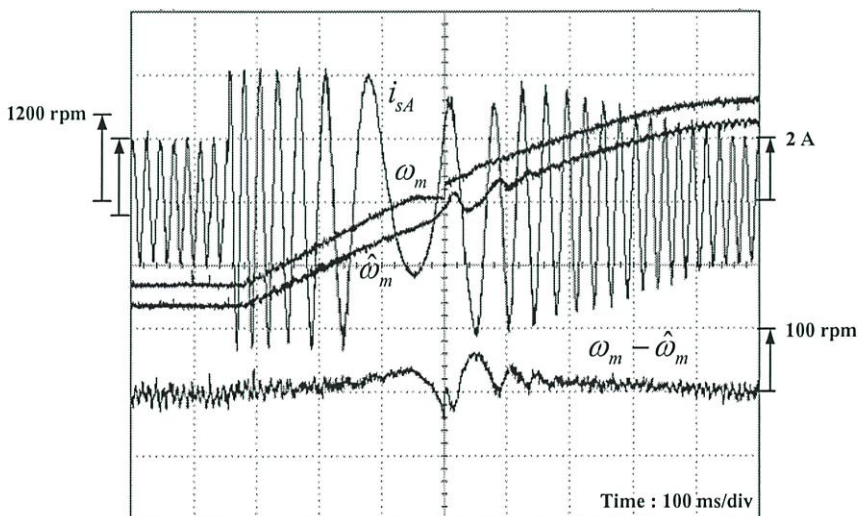
รูปที่ 6.24 ผลการทำงานของระบบ เมื่อใช้อัตราขยาย $k_p = 600$, $k_I = 30000$

6.8 ผลการทดสอบของอัตราขยาย k_I

ทำการทดสอบความมีเสถียรภาพการทำงานของระบบประมาณค่าความเร็วสำหรับการเลือกอัตราขยาย k_I โดยให้มอเตอร์ทำงานขณะไม่มีโหลด โดยกลับทิศความเร็วของมอเตอร์จาก 1410 rpm ไป -1410 rpm และ -1410 rpm ไป 1410 rpm โดยใช้อัตราขยาย $k_p = 600$, $k_I = 15000$ จากการทดสอบในรูปที่ 6.25 และ 6.26 จะเห็นว่าความเร็วผิดพลาด ($\omega_m - \hat{\omega}_m$) มีค่ามากในขณะที่มอเตอร์กลับทิศความเร็ว และเราสามารถเปรียบจากรูปที่ 6.15 และ 6.16 โดยเลือก k_I เพิ่มขึ้นสองเท่า ตามที่ได้ออกแบบไว้ โดยให้ k_p เท่าเดิม เราจะพบว่าค่าความเร็วผิดพลาดลดลงประมาณสองเท่าเช่นกัน การทดสอบนี้แสดงให้เห็นว่าสมการ (4.34) เป็นจริง นั่นคือเมื่ออัตราขยาย k_I มีค่ามากค่าความเร็วผิดพลาดจะมีค่าลดลง



รูปที่ 6.25 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก 1410 rpm \rightarrow -1410 rpm เมื่อใช้อัตราขยาย $k_p = 600$, $k_I = 15000$



รูปที่ 6.26 ผลการทำงานของระบบขณะกลับทิศความเร็วจาก -1410 rpm \rightarrow 1410 rpm เมื่อใช้อัตราขยาย $k_p = 600$, $k_I = 15000$

สรุปผลการวิจัยและข้อเสนอแนะ

7.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ผู้วิจัยได้พัฒนาระบบควบคุมแบบเวกเตอร์ไว้เช่นเซอร์วัดความเร็วสำหรับมอเตอร์เหนี่ยวนำ โดยใช้การประมาณค่าความเร็วจากระบบจำลองอ้างอิงแบบปรับตัวและได้ทำการศึกษาวิจัยในประเด็นสำคัญอันได้แก่ ปัญหาความช้าช้อนของระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์และระบบประมาณค่าความเร็ว และปัญหาในการออกแบบอัตรายายปรับตัว

1. นูรณาการระบบประมาณค่าฟลักซ์ที่ใช้ตัวสังเกตแบบวงรอบปิดฟลักซ์สเตเตอร์และระบบประมาณค่าความเร็ว สามารถลดความช้าช้อนของแบบจำลองได้ ซึ่งฟลักซ์โรเตอร์จากแบบจำลองกระแสหลังจากนูรณาการแล้วยังสามารถคำนวณได้ถูกต้องเหมือนเดิม และสามารถลดเวลาในการคำนวณในตัวไมโครคอนโทรลเลอร์ (Microcontroller) ที่ใช้ในการควบคุมระบบได้

2. การออกแบบอัตรายายปรับตัว มีเงื่อนไขการออกแบบที่ชัดเจนมากขึ้น สรุปได้คือ อัตรายายปรับตัวแบบอินทิเกรต (k_I) จะเป็นตัวกำหนดค่าความเร็วผิดพลาดของระบบประมาณค่าความเร็วและอัตรายายปรับตัวแบบสัดส่วน (k_p) จะเป็นตัวกำหนดขนาดของสัญญาณรบกวน การเลือกค่าความถี่หักมุม (ω_c) เป็นสิ่งสำคัญเนื่องจากเป็นตัวกำหนดอัตรายายปรับตัวแบบสัดส่วน เพื่อไม่ให้ส่วนเฟสของระบบประมาณค่าความเร็วมีค่าน้อยเกินไปซึ่งจะเป็นผลทำให้ระบบประมาณค่าความเร็วแกว่งในที่สุด นั่นคือต้องเลือก $\omega_c < \omega_{mr}$ แต่การเลือก ω_c น้อยเกินไปก็ไม่เป็นผลดีเนื่องจากจะทำให้อัตรายายปรับตัวแบบสัดส่วนมีค่ามากส่งผลต่อการขยายสัญญาณรบกวนได้ในทางปฏิบัติ ดังนั้นจึงต้องพิจารณาจากแผนภาพ โบเด (Bode) ของฟังก์ชันการโอนย้ายของ $G'_{22}(s)$

7.2 ข้อเสนอแนะ

จากการทำงานวิจัยในครั้งนี้ มีประเด็นที่สำคัญบางประการที่ควรพิจารณาศึกษาและวิจัยเพิ่มเติมเพื่อพัฒนาขีดความสามารถของระบบให้ดียิ่งขึ้นดังนี้

1. ความถูกต้องของค่าพารามิเตอร์ของมอเตอร์ที่ใช้ในการทดสอบเนื่องจากค่าพารามิเตอร์ต่างๆ ของมอเตอร์จะเปลี่ยนแปลงไปตามสถานะและเงื่อนไขในการทำงาน ตัวอย่างเช่น ค่าความต้านทานสเตเตอร์จะแปรเปลี่ยนไปตามอุณหภูมิแวดล้อม ดังนั้นเราควรศึกษาและวิเคราะห์ถึงผลกระทบอันเกิดจากค่าพารามิเตอร์ที่เปลี่ยนไปว่ามีผลต่อระบบอย่างไร และควรพัฒนาวิธีการควบคุมเพื่อให้ระบบมีความคงทนต่อการเปลี่ยนแปลงของค่าพารามิเตอร์ ซึ่งในงาน

วิจัยนี้ยังไม่ได้ทำการออกแบบอัตราขยายตัวควบคุม PI ในระบบประมาณค่าฟลักซ์ ซึ่งการพัฒนาในครั้งต่อไปควรพิจารณาถึงการออกแบบอัตราขยายให้เหมาะสมเพื่อให้ระบบประมาณค่าฟลักซ์มีความคงทน (Robust) ต่อการเปลี่ยนแปลงของค่าความต้านทานสเตเตอร์ที่เปลี่ยนไป

2. เนื่องจากงานวิจัยนี้ได้สร้างชุดต้นแบบเพื่อใช้ในการทดสอบการทำงาน ซึ่งมีวงจรอิเล็กทรอนิกส์หลายชุด โดยทำการเชื่อมต่อกันแต่ละชุดโดยสายไฟ ชุดทดสอบนี้จึงมีสายไฟหลายเส้นทำให้เกิดสัญญาณรบกวนได้ ดังนั้นจึงควรออกแบบให้วงจรอิเล็กทรอนิกส์แต่ละชุดอยู่บนบอร์ด (Board) เดียวกันและขนาดลายวงจรที่เป็นสายสัญญาณควรมีขนาดเล็ก เพื่อลดผลของสัญญาณรบกวน

3. เนื่องจากการทดสอบในระบบจริงจะใช้ Techogenerator เป็นตัวตรวจจับความเร็วจริง (ω_m) ป้อนกลับให้กับชุดประมวลผลสัญญาณดิจิทัล (DSP) เพื่อนำมาเปรียบเทียบกับความเร็วประมาณ ($\hat{\omega}_m$) ซึ่งตัวตรวจจับความเร็วดังกล่าวจะเกิดความผิดพลาดที่ความเร็วต่ำ ดังนั้นจึงควรใช้เอนโคเดอร์ (Encoder) เป็นตัวตรวจจับความเร็ว โดยใช้ร่วมกับชุดประมวลผลสัญญาณดิจิทัลที่สามารถนับจำนวนพัลส์ (Pulse) จากเอนโคเดอร์ได้โดยตรง

เอกสารอ้างอิง

- [1] Bimal K. Bose. **Power Electronics and AC Drive**. Prentice - Hall, 1986.
- [2] Bimal K. Bose. **Modern Power Electronics and AC Drive**. Prentice - Hall, 2002.
- [3] Peter Vas. **Vector Control of AC Machine**. Clarendon Press, Oxford, 1993.
- [4] Andrzej M.Trzynadlowski. **The Field Oriented Principle in Control of Induction Motors**. Academic Press, 2000.
- [5] Marian P. Kazmierkowski, R. Krishnan and Frede Blaabjerg. **Control in Power Electronics**. Kluwer Academic Publishers, 1994.
- [6] K.Zhou and D.Wang. “ **Relationship Between Space-Vector Modulation and Three Phase Carrier-Based PWM:A Comprehensive Analysis**”, IEEE Trans. on Ind. Elect., Vol. 49, No. 1, 2002, pp. 186-196.
- [7] H. Tajima , Y. Hori. “ **Speed Sensorless Field Orientation Control of the Induction Machine** ”, IEEE Trans. on Ind. Appl., Vol. 29, No. 1, 1993, pp. 175-180.
- [8] C. Lascu et al. “ **A Modified Direct Torque Control for Induction Motor Sensorless Drive** ”, IEEE Trans. on Ind. Appl., Vol. 36, No. 1, 2000, pp. 122-130.
- [9] C. Schauder. “ **Adaptive Speed Identification For Vector Control Of Induction Motors Without Rotational Transducers**”, IEEE Trans. on Ind. Appl., Vol. AL-28, No.5,1992, pp.1054-1061.
- [10] Z. Peng et al. “ **Robust Speed Identification for Speed Sensorless Vector Control of Induction Motors**”, IEEE Trans. on Ind. Appl., Vol. IA-30, No.5, 1994, pp. 1234–1240.
- [11] S. Tamai et al. “ **Speed Sensorless Vector Control of Induction Motor With Model Reference Adaptive System**”, Proc. of IEEE /IAS, 1987, pp. 189-195.
- [12] S.Sangwongwanich and S.Suwankawin , “ **A Speed-Sensorless IM Drive with Modified Decoupling Control**”, In Proc.PCC , Vol.1, Nagaoka, Japan, 1997 ,pp. 85-90.
- [13] ณัฐพงศ์ เมืองจันทร์ และ วิจิตร กิณเรศ, “การออกแบบอัตราขยายปรับตัวระบบควบคุมแบบเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วสำหรับขับเคลื่อนมอเตอร์เหนี่ยวนำ.” วิศวกรรมลาดกระบัง ปีที่ 25, ฉบับที่ 4, เดือนธันวาคม, 2551.
- [14] จริญญา ตั้งวงศ์ชูเกตุ. “ การควบคุมความเร็วมอเตอร์เหนี่ยวนำ 3 เฟส แบบเฟสเซอร์โดยใช้ตัวประมวลผลสัญญาณดิจิทัลในการควบคุมการทำงาน.” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546.

เอกสารอ้างอิง (ต่อ)

- [15] จรูญ สุริยะวิจิตรวงศ์, “ การควบคุมมอเตอร์เหนี่ยวนำแบบควบคุมแรงบิดโดยตรงไร้เซ็นเซอร์วัดความเร็วรอบ.” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า, สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2546.
- [16] สุรพงศ์ สุวรรณกวิน. “ เทคนิคใหม่ในการวิเคราะห์เสถียรภาพและออกแบบระบบขับเคลื่อนมอเตอร์เหนี่ยวนำไร้เซ็นเซอร์วัดความเร็วที่ใช้การควบคุมแบบการเชื่อมร่วม.” วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า, จุฬาลงกรณ์มหาวิทยาลัย, 2544.
- [17] ชูเกียรติ นิธิโยธาน. “ ระบบควบคุมเวกเตอร์แบบควบคุมกระแสไร้เซ็นเซอร์วัดความเร็วสำหรับมอเตอร์เหนี่ยวนำ.”, วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า, จุฬาลงกรณ์มหาวิทยาลัย , 2542.
- [18] อุเทน นิตยารักษ์กุล. “ ระบบควบคุมเวกเตอร์ไร้เซ็นเซอร์วัดความเร็วที่อาศัยการประมาณค่าความเร็วจากแรงดันไฟฟ้าเหนี่ยวนำ.”, วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า, จุฬาลงกรณ์มหาวิทยาลัย, 2542.

ภาคผนวก ก

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่



วิศวกรรมลาดกระบัง

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang

ปีที่ 25 ฉบับที่ 4

ธันวาคม 2551

วิศวกรรมโทรคมนาคม

1. การออกแบบวงจรเทอร์นาทีลจิก
สมปอง วิเศษพานิชกิจ พิชัย สุวรรณลอยล่อง กอบชัย เดชหาญ 1
2. โครงข่ายผ่านทุกความถี่อันดับที่หนึ่งโดยใช้วงจร DDCC
มนตรี คำเงิน กอบชัย เดชหาญ 7
3. วงจรกำเนิดสัญญาณแบบควอดราเจอร์โดยใช้วงจร DDCC
มนตรี คำเงิน กอบชัย เดชหาญ 12
4. วงจรรองความถี่หลายหน้าที่และวงจรมอดูเลชันที่ใช้โครงสร้างของอินทิเกรตเตอร์แบบสูญเสียและแบบไม่สูญเสีย
มนตรี สมดลยกันก พิพัฒน์ พรหมมี กอบชัย เดชหาญ 17

วิศวกรรมไฟฟ้ากำลัง

5. การออกแบบอัตราขยายปรับตัวระบบควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็วสำหรับขับเคลื่อนมอเตอร์เหนี่ยวนำ
ณัฐพงศ์ เมืองจันทร์ วิจิตร กิมเรศ 23
6. รีเลย์ระยะทางแบบโมห์ที่ปรับค่าได้ด้วยวิธี KU เพื่อขจัดเซชความต้านทานฟอล์ทขณะเกิดฟอล์ทแบบเฟส-เฟส
สุระเชษฐ เดชฟุ้ง ตฤณ แสงสุวรรณ 29
7. การปรับปรุงค่าความเพี้ยนเชิงฮาร์มอนิกของวงจรมัลติเซลล์อินเวอร์เตอร์หนึ่งเฟสโดยวิธีปรับมุมและระดับแรงดันไฟฟ้ากระแสตรง
มัจฉิ ได้ตะดาหยง อนุวัตร ประเสริฐสิทธิ์ กุสุมาลย์ เฉลิมยานนท์ เกริกชัย ทองหนูสุระพล เลี้ยวมนตรี 35

อิเล็กทรอนิกส์

8. ลักษณะสมบัติของซีดทีดีแบริเออร์โฟโตไดโอดแบบเชิงราบบนซิลิคอนชนิดเอ็นที่ขึ้นกับความเข้มข้นสารเจือของฐานรอง
สุรศักดิ์ เนียมเจริญ 41

วิศวกรรมอุตสาหการ

9. การศึกษาลำดับความสำคัญของตัวบ่งชี้ประเมินการจัดการโซ่อุปทานอุตสาหกรรมยานยนต์ในมิติเชิงคุณภาพ
ตรีทศ เหล่าศิริพงษ์ทอง 47

วิศวกรรมเคมี

10. การพัฒนาระบบสำหรับการวัดค่าคงที่ไดอิเล็กตริกของเซรามิกเลดเซอร์โคเนตไททานเนต
อดิศักดิ์ ร่มพุดมताल 53

การออกแบบอัตราขยายปรับตัวระบบควบคุมแบบเวกเตอร์ ไร้เซนเซอร์วัดความเร็วสำหรับขับเคลื่อนมอเตอร์เหนี่ยวนำ

Design of an Adaptive Gain Speed Sensorless

Vector Control System for an Induction Motor Drive

ณัฐพงศ์ เมืองจันทร์ วิจิตร กิณเรศ

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

บทความนี้นำเสนอการออกแบบอัตราขยายปรับตัวของเทคนิคควบคุมแบบเวกเตอร์ไร้เซนเซอร์วัดความเร็วบนระบบจำลองอ้างอิงแบบปรับตัว โดยนำความแตกต่างของสถานะฟลักซ์โรเตอร์ที่ได้จากแบบจำลองแรงดันและที่ได้จากแบบจำลองกระแสเข้าไปปรับเอาต์พุตของค่าประมาณความเร็วรอบโดยผ่านกระบวนการควบคุมแบบพีไอ โดยแนวทางการออกแบบอัตราขยายปรับตัวจะพิจารณาถึงความเร็วรอบจากการประมาณติดตามความเร็วรอบจริงในช่วงมอเตอร์เปลี่ยนแปลงความเร็วรอบและเพื่อให้มั่นใจว่าระบบประมาณค่าความเร็วมีเสถียรภาพตลอดย่านการทำงานและเหมาะสมกับการนำไปใช้ในทางปฏิบัติจริง ผลจากการทดสอบโดยใช้โปรแกรม MATLAB แสดงถึงสมรรถนะการตอบสนองทางพลวัต

คำสำคัญ : ระบบจำลองอ้างอิงแบบปรับตัว, แบบจำลองแรงดัน, แบบจำลองกระแส, การออกแบบอัตราขยายปรับตัว

Abstract

This paper describes the adaptive gain design of a speed sensorless vector control technique based on a model reference adaptive system (MRAS). The estimated speed is derived from the difference of the rotor flux state resulting from voltage and current models which is passed through the PI controller. The design guideline of the adaptive gain considers the estimated speed for a motor speed change in order to ensure that the proposed speed estimation system has stability over the entire operating range. The simulation results show that the estimated rotor speed is close to the actual speed. MATLAB is used for the simulation. The dynamic performance is demonstrated.

Keywords : Model Reference Adaptive System, Current Model, Adaptive Gain Design, PI Controller

1. บทนำ

ระบบจำลองอ้างอิงแบบปรับตัว (MRAS) โดยวิธีใช้ฟลักซ์โรเตอร์[1] เป็นการประมาณค่าความเร็วแบบวงรอบปิดและเป็นวิธีการที่ง่ายมีความคงทนต่อการเปลี่ยนแปลงของค่าพารามิเตอร์ของมอเตอร์ดีพอสมควร โดยหลักการจะนำค่าแรงดันและกระแสด้านสเตเตอร์ของ

มอเตอร์มาทำการคำนวณหาค่าฟลักซ์โรเตอร์ที่ได้จากแบบจำลองแรงดันและฟลักซ์โรเตอร์ที่ได้จากแบบจำลองกระแส[2] ซึ่งความแตกต่างของฟลักซ์โรเตอร์ทั้งสองนี้จะนำไปปรับเอาต์พุตของค่าประมาณความเร็วรอบโดยผ่านตัวควบคุมแบบพีไอ (PI controller) สิ่งสำคัญคือการออกแบบอัตราขยายปรับตัวให้เหมาะสมเพื่อให้ระบบ

ประมาณค่าความเร็วประมาณความเร็วรอบได้ใกล้เคียงกับความเร็วยุติและมีเสถียรภาพตลอดการทำงานจากบทความ [4,5] ได้วิเคราะห์การออกแบบอัตราขยายปรับตัวโดยทางเดินรากของระบบประมาณค่าความเร็วซึ่งการวิเคราะห์สรุปได้เพียงว่าสามารถใช้อัตราขยายปรับตัวที่มีค่ามากได้ โดยแบนด์วิดธ์ของระบบประมาณค่าความเร็วถูกจำกัดด้วยสัญญาณรบกวนซึ่งยังขาดความชัดเจนสำหรับนำไปปฏิบัติจริง เนื่องจากไม่ทราบเงื่อนไขในการออกแบบอัตราขยายปรับตัวที่แน่นอน และบทความ[6] ได้นำเสนอการออกแบบอัตราขยายปรับตัวเฉพาะบนระบบประมาณค่าความเร็วแบบตัวตั้งเกดเต็มอันดับ(Full-order observer) เท่านั้นและยังไม่ได้แสดงถึงการเลือกค่าอัตราขยายปรับตัวที่ชัดเจนและเหมาะสม

บทความนี้ได้นำเสนอการออกแบบอัตราขยายปรับตัวบนระบบจำลองอ้างอิงแบบปรับตัว(MRAS)โดยพิจารณาถึงความเร็วรอบจากการประมาณติดตามความเร็วรอบจริงในช่วงมอเตอร์เปลี่ยนแปลงความเร็วรอบโดยเงื่อนไขการออกแบบนี้จะถูกกำหนดด้วยค่าความเร็วรอบผิดพลาดและนำไปสู่การหาค่าอัตราขยายปรับตัวแบบอินทิเกรตของระบบประมาณค่าความเร็ว โดยที่อัตราขยายปรับตัวแบบสัดส่วนจะถูกพิจารณาจากกรณีที่มอเตอร์ทำงาน ซึ่งเป็นการวิเคราะห์ที่มีเงื่อนไขชัดเจนขึ้นและเลือกค่าอัตราขยายปรับตัวที่เหมาะสมกับระบบประมาณค่าความเร็วได้ซึ่งการออกแบบจะแสดงได้ในหัวข้อต่อไป

2. แบบจำลองทางพลวัตของมอเตอร์เหนี่ยวนำ

แบบจำลองทางพลวัตของมอเตอร์เหนี่ยวนำชนิดกรงกระรอกโดยการต่อขดลวดแบบสตาร์(Y) ที่อยู่บนแกนนิ่งเป็นดังสมการ (1) และ (2) [3]

$$\frac{di_s}{dt} = A_{11}i_s + A_{12}\lambda_r + B_1v_s \quad (1)$$

$$\frac{d\lambda_r}{dt} = A_{21}i_s + A_{22}\lambda_r \quad (2)$$

เมื่อ

$$A_{11} = -\left(\frac{R_s}{\sigma L_s} + \frac{1-\sigma}{\sigma\tau_r}\right)I$$

$$A_{12} = k\left(\frac{1}{\tau_r} - p\omega_m J\right) \quad A_{21} = \frac{M}{\tau_r}I$$

$$A_{22} = -\frac{1}{\tau_r}I + p\omega_m J \quad B_1 = \frac{1}{\sigma L_s}I$$

$$\sigma = 1 - \frac{M^2}{L_s L_r} \quad \tau_r = \frac{L_r}{R_r} \quad k = \frac{M}{\sigma L_s L_r}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

v_s : สเปซเวกเตอร์ของแรงดันสเตเตอร์บนแกน $\alpha - \beta$

i_s : สเปซเวกเตอร์ของกระแสสเตเตอร์บนแกน $\alpha - \beta$

λ_r : สเปซเวกเตอร์ของฟลักซ์โรเตอร์บนแกน $\alpha - \beta$

R_s, R_r : ความต้านทานของขดลวดสเตเตอร์และโรเตอร์

L_s, L_r : ความเหนี่ยวนำของขดลวดสเตเตอร์และโรเตอร์

M : ความเหนี่ยวนำร่วมระหว่างขดลวดสเตเตอร์และโรเตอร์

σ : สัมประสิทธิ์การรั่วไหล (Leakage coefficient)

ω_m : ความเร็วโรเตอร์(Rotor speed)

p : จำนวนคู่ของขั้วแม่เหล็ก (Number of pole pairs)

3. การประมาณค่าความเร็วรอบ

3.1 โครงสร้างของระบบประมาณค่าความเร็ว

จากสมการ (1) และ (2) สามารถคำนวณหาค่าฟลักซ์โรเตอร์ที่ได้จากแบบจำลองแรงดันและที่ได้จากแบบจำลองกระแสได้ดังสมการ (3) และ (4)

แบบจำลองแรงดัน(Voltage Model) :

$$\lambda_r = \frac{L_r}{M} \left[\frac{1}{s} (v_s - R_s i_s) - \sigma L_s i_s \right] \quad (3)$$

แบบจำลองกระแส(Current Model) :

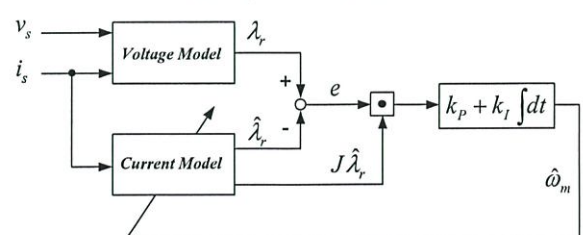
$$\hat{\lambda}_r = \frac{M}{1 + s\tau_r} i_s + J \frac{p\omega_m}{1 + s\tau_r} \tau_r \hat{\lambda}_r \quad (4)$$

เมื่อ s แทนตัวดำเนินการอนุพันธ์ (d/dt)

ผลต่างระหว่างฟลักซ์โรเตอร์ทั้งสองจะผ่านไปผ่านตัวควบคุม

แบบพีไอดังรูปที่ 1 เพื่อประมาณความเร็วรอบดังสมการ (5)

$$\hat{\omega}_m = (k_p + k_I/s) e^T J \hat{\lambda}_r \quad (5)$$



รูปที่ 1 ระบบจำลองอ้างอิงแบบปรับตัว (MRAS)

3.2 ระบบประมาณค่าความเร็วบนแกนหมุน

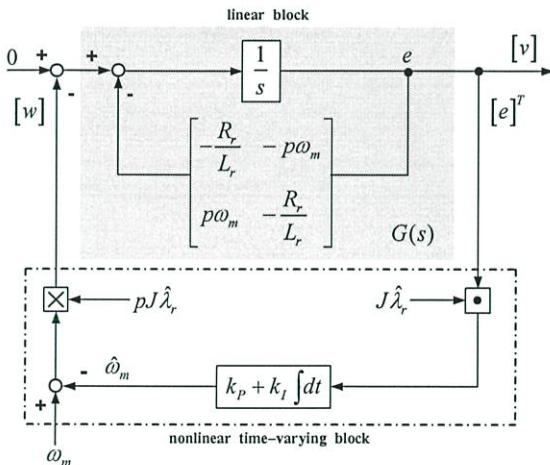
เนื่องจากสมการแบบจำลองแรงดันเป็นสมการจริงของมอเตอร์ ดังนั้นสมการจากแบบจำลองแรงดันจึงสมมูลกับสมการแบบจำลองกระแสที่ใช้ความเร็วรอบจริงจึงใช้สมการแบบจำลองกระแสของมอเตอร์จริงเป็นแบบจำลองแรงดันในการพิจารณาได้ ดังนั้นสามารถหาค่าความผิดพลาดของฟลักซ์โรเตอร์ได้ดังสมการ(6) และแสดงบล็อกไดอะแกรมวงรอบปิดซึ่งจะประกอบไปด้วยฟังก์ชันการอินทรีย์ของค่าผิดพลาดฟลักซ์โรเตอร์และตัวควบคุมแบบพีไอได้ดังรูปที่ 2

$$e = \lambda_r - \hat{\lambda}_r = G(s)pJ\hat{\lambda}_r(\omega_m - \hat{\omega}_m) \quad (6)$$

เมื่อ

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} = [(s+x)I - yJ]^{-1} \quad (7)$$

และ $x = \frac{R_r}{L_r}, \quad y = p\omega_m$



รูปที่ 2 บล็อกไดอะแกรมความผิดพลาดระบบประมาณค่าความเร็วบนแกนนี้

จากรูปที่ 2 แปลงระบบประมาณค่าความเร็วไปอยู่บนแกนหมุน ดังนั้นฟังก์ชันการอินทรีย์ $G(s)$ จะถูกแปลงเป็น $G(sI + \omega_e J)$ เนื่องจากสัญญาณขาเข้า $G'(s)$ มีเฉพาะองค์ประกอบในแกน q เท่านั้น ดังนั้นสัญญาณที่ตอบสนองสัญญาณเข้านี้คือ $G'_{12}(s)$ และ $G'_{22}(s)$ และองค์ประกอบในแกน q ของสัญญาณผิดพลาด e เท่านั้นที่ใช้ประมาณค่าความเร็วดังสมการ (8)

$$\begin{bmatrix} e_d \\ e_q \end{bmatrix} = G(sI + \omega_e J) \begin{bmatrix} 0 \\ |\hat{\lambda}_r| \end{bmatrix} p(\omega_m - \hat{\omega}_m) \quad (8)$$

เมื่อ

$$G(sI + \omega_e J) = G'(s) = \begin{bmatrix} G'_{11}(s) & G'_{12}(s) \\ G'_{21}(s) & G'_{22}(s) \end{bmatrix} \quad (9)$$

และ $|\hat{\lambda}_r|$ คือขนาดของฟลักซ์โรเตอร์

ดังนั้น Inverse Matrix ของ $G'(s)$ จะได้

$$[G'(s)]^{-1} = \frac{1}{(s+x)^2 + \omega_{sl}^2} \begin{bmatrix} s+x & -\omega_{sl} \\ \omega_{sl} & s+x \end{bmatrix} \quad (10)$$

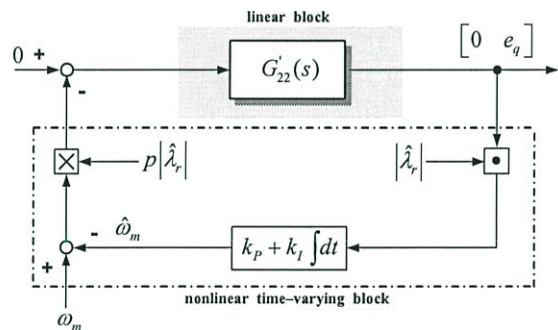
โดยที่ $\omega_{sl} = \omega_e - y$

และ ω_e คือความเร็วเชิงมุมทางไฟฟ้า (rad/s)

จากสมการ (10) ฟังก์ชันการอินทรีย์ $G'_{22}(s)$ เท่านั้นที่ถูกใช้ประมาณค่าความเร็วจะได้

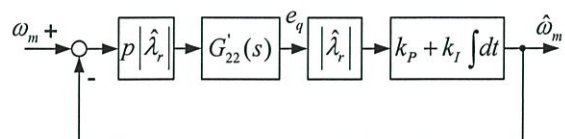
$$G'_{22}(s) = \frac{s+x}{(s+x)^2 + \omega_{sl}^2} \quad (11)$$

เราสามารถแสดงบล็อกไดอะแกรมความผิดพลาดระบบประมาณค่าความเร็วบนแกนหมุนของฟังก์ชันการอินทรีย์ $G'_{22}(s)$ ได้ดังรูปที่ 3



รูปที่ 3 บล็อกไดอะแกรมความผิดพลาดระบบประมาณค่าความเร็วบนแกนหมุนที่ใช้ฟังก์ชันการอินทรีย์ $G'_{22}(s)$

จากรูปที่ 3 ทำการลดรูปและจัดระบบวงรอบปิดของค่าผิดพลาดระบบประมาณค่าความเร็วบนแกนหมุนในลักษณะสัญญาณเข้าออกทางเดียวได้ดังรูปที่ 4



รูปที่ 4 วงรอบปิดของระบบประมาณค่าความเร็วบนแกนหมุนในลักษณะสัญญาณเข้าออกทางเดียว (SISO)

3.3 การออกแบบอัตราขยายปรับตัว (k_p, k_I)

เนื่องจากในทางปฏิบัติการเปลี่ยนแปลงความเร็วรอบจริงของมอเตอร์มีลักษณะเป็นแรมปป์อยู่เสมอ ซึ่งในขณะที่มอเตอร์เปลี่ยนแปลงความเร็วรอบนั้นแรงบิดที่มอเตอร์สร้างขึ้น (T_e) จะมีค่าคงที่และจะทำให้ความเร็วรอบของมอเตอร์เพิ่มขึ้นหรือลดลงเป็นเชิงเส้นโดยมีค่า

$$R = \frac{d\omega_m}{dt} = \frac{T_e - T_L}{J_m}$$

แบบฟีดโอบของระบบประมาณค่าความเร็วไม่สามารถจัดค่าความเร็วรอบผิดพลาดให้มีค่าเป็นศูนย์ได้ ดังนั้นจึงออกแบบโดยพิจารณาถึงผลตอบสนองของระบบประมาณค่าความเร็วเมื่อความเร็วรอบจริงเป็นฟังก์ชันแรมปป์[6] จากรูปที่ 4 สามารถแสดงฟังก์ชันการโอนย้ายระหว่างความเร็วรอบผิดพลาด ($\omega_m - \hat{\omega}_m$) เทียบกับความเร็วรอบจริง (ω_m) ได้ดังสมการ (12)

$$\frac{\omega_m(s) - \hat{\omega}_m(s)}{\omega_m(s)} = \frac{1}{1 + CG'_{22}(s) \cdot (k_p + k_I / s)} \quad (12)$$

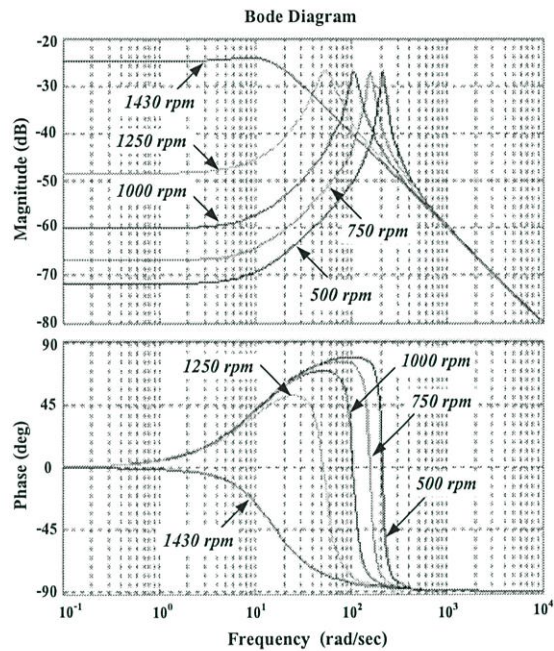
เมื่อ $C = p|\hat{\lambda}_r|^2$

จากทฤษฎีบทค่าสุดท้าย (Final value Theorem) สามารถหาค่าความเร็วรอบผิดพลาด (e_{ss}) ในขณะที่เปลี่ยนแปลงความเร็วรอบจะได้ดังสมการ (13)

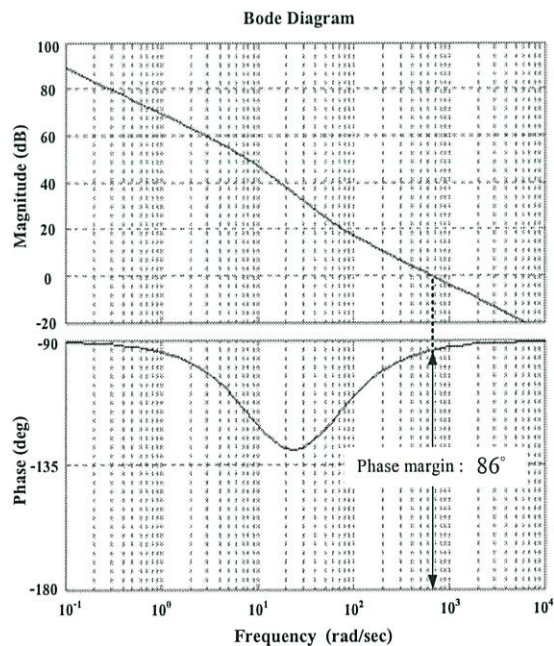
$$e_{ss} = \lim_{s \rightarrow 0} \left[s \cdot \frac{s}{s + CG'_{22}(s) \cdot (sk_p + k_I)} \right] \left[\frac{T_e - T_L}{J_m s^2} \right] = \frac{T_e - T_L}{J_m CG'_{22}(s)|_{s=0} \cdot k_I} \quad (13)$$

จากสมการ (13) สังเกตจะเห็นว่าค่า k_I มากจะลดค่าความผิดพลาดของความเร็วรอบได้ บทความนี้ได้เลือกค่า $e_{ss} = 2$ rpm ในขณะที่ไม่มีโหลด ($T_L = 0$) แทนค่าพารามิเตอร์จากภาคผนวก จะได้ค่า $k_I = 25000$ และการเลือกค่า k_p จะพิจารณาส่วนเพื่อเฟส (Phase margin) ของระบบประมาณค่าความเร็ว โดยการเลือกค่าความถี่หักมุม ($\omega_c = k_I/k_p$) ของตัวควบคุมฟีดโอบ จากรูปที่ 5 เลือก $\omega_c = 50$ rad/s จะได้ $k_p = 500$ แสดงแผนภาพโบเดของระบบประมาณค่าความเร็ววงรอบเปิดจากการออกแบบได้ ดังรูปที่ 6 ซึ่งพบว่ามีส่วนเพื่อเฟสอยู่ 86 องศา ซึ่งมากเพียงพอสำหรับเสถียรภาพการควบคุม การเลือก

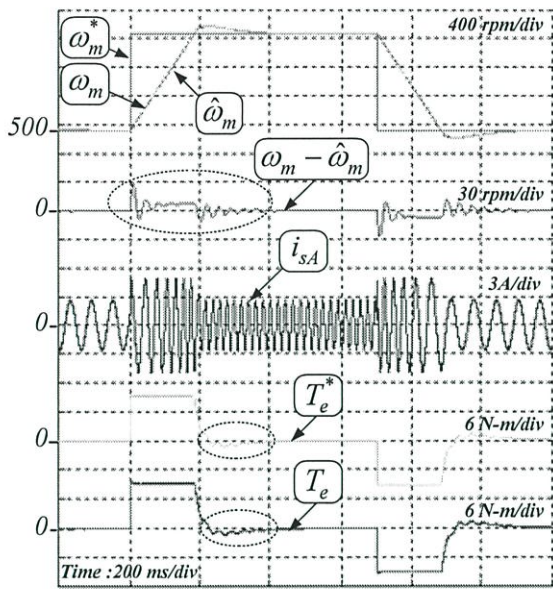
ค่าความถี่หักมุม (ω_c) ไม่ควรเลือกให้มีค่ามากกว่าหรือเท่ากับความถี่ที่มอเตอร์ทำงาน ($\omega_o = 300$ rad/s) เพราะจะทำให้ส่วนเพื่อเฟสตกลงส่งผลต่อเสถียรภาพ ได้ทำให้ระบบประมาณค่าความเร็วแกว่งในที่สุด ดังแสดงแผนภาพโบเดของระบบประมาณค่าความเร็วที่ค่าความถี่หักมุมใด ๆ ได้ ดังรูปที่ 7



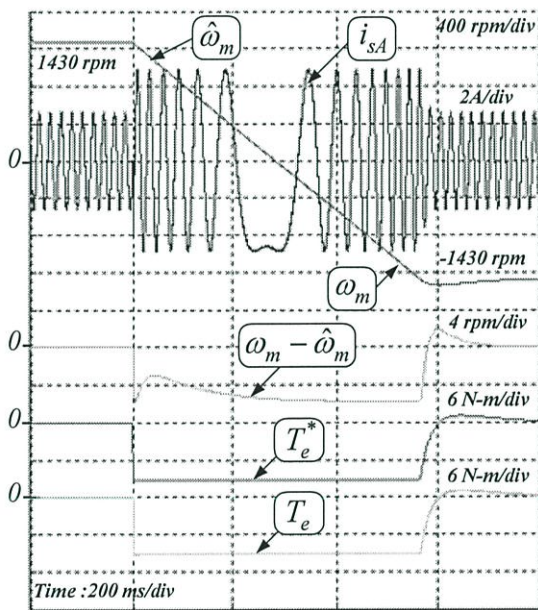
รูปที่ 5 แผนภาพโบเดของ $G'_{22}(s)$ ที่จุดทำงานใด ๆ



รูปที่ 6 แผนภาพโบเดของระบบประมาณค่าความเร็วจากการออกแบบ เมื่อ $k_p = 500, k_I = 25000$



รูปที่ 11 ทดสอบขณะเปลี่ยนแปลงความเร็วรอบของมอเตอร์เมื่อ $\omega_c > \omega_o$ ($k_p = 50, k_I = 25000$)



รูปที่ 12 ทดสอบการกลับทิศทางการหมุนของมอเตอร์ที่ความเร็ว 1430 ถึง -1430 rpm เมื่อ $k_p = 500, k_I = 25000$

5. สรุป

การออกแบบอัตรายายปรับตัวของระบบประมาณค่าความเร็วบนระบบจำลองอ้างอิงแบบปรับตัวโดยพิจารณาผลการตอบสนองเมื่อความเร็วจริงมีลักษณะเป็นแรมป์ จากการทดสอบโดยใช้โปรแกรม MATLAB พบว่าอัตรายายปรับตัวแบบอินทิเกรทจะเป็นตัวกำหนดค่าความเร็วรอบผิดพลาดของระบบประมาณค่าความเร็วและอัตรายายปรับตัวแบบสัดส่วนจะเป็นตัวกำหนดขนาดของ

สัญญาณรบกวน การเลือกค่าความถี่หักมุมเป็นสิ่งสำคัญเนื่องจากเป็นตัวกำหนดอัตรายายปรับตัวแบบสัดส่วนเพื่อไม่ให้ส่วนเฟสของระบบประมาณค่าความเร็วมีค่าน้อยเกินไปซึ่งจะเป็นผลทำให้ระบบประมาณค่าความเร็วแกว่งในที่สุด นั่นคือต้องเลือก $\omega_c < \omega_o$ แต่การเลือก ω_c น้อยเกินไปก็ไม่เป็นผลดีเนื่องจากจะทำให้อัตรายายปรับตัวแบบสัดส่วนมีค่ามากส่งผลต่อการขยายสัญญาณรบกวนได้ในทางปฏิบัติ ซึ่งเราจะพิจารณาได้จากรูปที่ 5 บทความนี้จึงเลือกความถี่หักมุมอยู่ในช่วง $50 \leq \omega_c < 300$ rad/s ดังนั้นการออกแบบอัตรายายปรับตัวที่พิจารณาถึงความเร็วรอบจากการประมาณติดตามความเร็วรอบจริงในช่วงมอเตอร์เปลี่ยนแปลงความเร็วรอบบนระบบจำลองอ้างอิงแบบปรับตัวนี้มีเงื่อนไขการออกแบบอัตรายายปรับตัวที่ชัดเจนและสามารถนำไปใช้ในทางปฏิบัติจริงได้

6. เอกสารอ้างอิง

- [1] Bimal K. Bose. Modern Power Electronics and AC Drive. Prentice - Hall, 2002.
- [2] C. Lascu et al. "A Modified Direct Torque Control for Induction Motor Sensorless Drive", IEEE Trans. on Ind. Appl., Vol. 36, No. 1, 2000, pp. 122-130.
- [3] H. Tajima and Y. Hori, "Speed Sensorless Field Orientation Control of the Induction Machine", IEEE Trans. on Ind. Appl., Vol. 29, No. 1, 1993, pp. 175-180.
- [4] C. Schauder. "Adaptive Speed Identification For Vector Control of Induction Motors Without Rotational Transducers", Conf.Record of IEEE/IAS Annual Meeting 1989 , pp. 493-499.
- [5] F.Z. Peng and T. Fukao, "Robust Speed Identification for Speed-Sensorless Vector Control of Induction Motors", IEEE Trans. on Ind. Appl., Vol. 30, No. 5, 1994, pp. 1234-1240.
- [6] S.Sangwongwanich and S.Suwankawin , "A Speed-Sensorless IM Drive with Modified Decoupling Control" ,In Proc.PCC , Vol.1, Nagaoka, Japan, 1997 ,pp. 85-90.

7. ภาคผนวก

ก. พิกัดและพารามิเตอร์ของมอเตอร์เหนี่ยวนำ

2HP, 220/380V, 6.3/3.6 A, 1430 rpm, 4 Pole	
$i_{sd} = 2.7 A$	$i_{sq} = 4.3 A$
$R_s = 3.965 \Omega$	$R_r = 3.477 \Omega$
$L_s = 0.311 H$	$L_r = 0.311 H$
$M = 0.292 H$	$J_m = 0.018 kg - m^2$

ภาคผนวก ข

โปรแกรมที่ใช้ในการควบคุมระบบ

```

=====
Speed Sensorless Vector Control Base on MRAS
=====

```

```

// Include header files used in the main function
#include "target.h"
#include "DSP281x_Device.h"
#include "f281xadc04b.h"
#include "IQmathLib.h"
#include "aci_fe.h"
#include "aci_se.h"
#include "park.h"
#include "ipark.h"
#include "pid_reg3.h"
#include "clarke.h"
#include "svgen_dq.h"
#include "f281xpwm.h"
#include "f281xpwm dac.h"
#include "volt_calc.h"

// Prototype statements for functions found within this file.
interrupt void MainISR(void);

// Global variables used in this system
float32 SpeedCom=0;           // Speed Command (pu)
float32 Speedfb;             // Speed feedback (pu)
float32 IdRef = 0.45;        // Id reference (pu)
float32 IqLim = 0.89;        // Iq Limited (pu)
float32 T = 0.001/ISR_FREQUENCY; // Samping period
_iq DatQ, DatQ1, DatQ2, Tmp, Tmp1, Tmp2, ChOut, ChOut1,
    ChOut2, DcBus, spd;
Uint16 IsrTicker = 0;
Uint16 BackTicker = 0;
float32 Direct;
int16 PwmDacCh1 = 0;
int16 PwmDacCh2 = 0;
int16 PwmDacCh3 = 0;
Uint16 sdata1; // send data
Uint16 sdata2;
Uint16 rdata; // received data
Uint16 rdata1;
Uint16 sum=0;
Uint16 sum1=0;
Uint16 a=1;
Uint16 b=1;
Uint16 SpeedLoopPrescaler = 3; // Real speed loop prescaler
Uint16 SpeedLoopCount = 1; // Real speed loop counter
Uint16 SpeedLoopPrescaler01 = 3; // Estimate speed loop prescaler
Uint16 SpeedLoopCount01 = 1; // Estimate speed loop counter

// Instance rotor flux and speed estimations
ACIFE fe1 = ACIFE_DEFAULTS;
ACISE se1 = ACISE_DEFAULTS;

// Instance a few transform objects
CLARKE clarke1 = CLARKE_DEFAULTS;
PARK park1 = PARK_DEFAULTS;
IPARK ipark1 = IPARK_DEFAULTS;

// Instance PID regulators to regulate the d and q
// synchronous axis currents and speed
PIDREG3 pid1_id = PIDREG3_DEFAULTS;
PIDREG3 pid1_iq = PIDREG3_DEFAULTS;

```

```

PIDREG3 pid1_spd = PIDREG3_DEFAULTS;

// Instance a PWM DAC driver instance
PWMDAC pwmdac1 = PWMDAC_DEFAULTS;

// Instance a Space Vector PWM modulator. This modulator
// generates a, b and c
// phases based on the d and q stationery reference frame inputs
SVGENDQ svgen_dq1 = SVGENDQ_DEFAULTS;

// Instance a phase voltage calculation
PHASEVOLTAGE volt1 = PHASEVOLTAGE_DEFAULTS;

// ADC chanel4
ADCVALSB adc04b_1 = ADCVALSB_DEFAULTS;

// Instance the constant calculations for rotor flux and
// speed estimations
ACIFE_CONST fel_const = ACIFE_CONST_DEFAULTS;
ACISE_CONST sel_const = ACISE_CONST_DEFAULTS;

void main(void)
{
// *****
// Initialization code for DSP_TARGET = F2812
// *****
// Initialize System Control registers, PLL, WatchDog Clocks to
// Alphafault state:
// This function is found in the DSP281x_SysCtrl.c file.
    InitSysCtrl();

// HISPCP prescale register settings, normally it will be set to
// Alphafault values
    EALLOW; // This is neeAlphad to write to EALLOW
// protected registers
    SysCtrlRegs.HISPCP.all = 0x0000; // SYSCLKOUT/1
    EDIS; // This is neeAlphad to disable write to
// EALLOW protected registers

// Disable and clear all CPU interrupts:
    DINT;
    IER = 0x0000;
    IFR = 0x0000;

// Initialize Pie Control Registers To default State:
// This function is found in the DSP281x_PieCtrl.c file.
    InitPieCtrl();

// Initialize the PIE Vector Table To a Known State:
// This function is found in DSP281x_PieVect.c.
// This function populates the PIE vector table with
// pointers to the shell ISR functions found in
// DSP281x_DefaultIsr.c.
    InitPieVectTable();

// User specific functions, Reassign vectors (optional)
// Enable Interrupts:

// Initialize EVA Timer 1:
// Setup Timer 1 Registers (EV A)
    EvaRegs.GPTCONA.all = 0;

```

```

// Enable Underflow interrupt bits for GP timer 1
  EvaRegs.EVAIMRA.bit.T1UFINT = 1;
  EvaRegs.EVAIFRA.bit.T1UFINT = 1;

// Reassign ISRs.
// Reassign the PIE vector for T1UFINT to point to a different
// ISR then the shell routine found in DSP281x_DefaultIsr.c.
// This is done if the user does not want to use the shell ISR
// routine but instead wants to use their own ISR.

    EALLOW;    // This is neeAlphad to write to EALLOW
// protected registers
    PieVectTable.T1UFINT = &MainISR;
    EDIS;    // This is neeAlphad to disable write to
// EALLOW protected registers

// Enable PIE group 2 interrupt 6 for T1UFINT
    PieCtrlRegs.PIEIER2.all = M_INT6;

// Enable CPU INT2 for T1UFINT:
    IER |= M_INT2;

// Initialize PWM module
    pwm1.PeriodMax = SYSTEM_FREQUENCY*1000000*T/2;
// Perscaler X1 (T1), ISR period = T x 1
    pwm1.init(&pwm1);

// Initialize PWMDAC module
// PWMDAC Frequency = 30 kHz
    pwmdac1.PeriodMax = (SYSTEM_FREQUENCY*200/(30*2))*5;
    pwmdac1.PwmDacInPointer0 = &PwmDacCh1;
    pwmdac1.PwmDacInPointer1 = &PwmDacCh2;
    pwmdac1.PwmDacInPointer2 = &PwmDacCh3;
    pwmdac1.init(&pwmdac1);

// Initialize ADC 4 ch
    adc04b_1.init(&adc04b_1);

// Initialize Flux Estimation constant
    fel_const.Rs = RS;
    fel_const.Rr = RR;
    fel_const.Ls = LS;
    fel_const.Lr = LR;
    fel_const.Lm = LM;
    fel_const.Ib = BASE_CURRENT;
    fel_const.Vb = BASE_VOLTAGE;
    fel_const.Ts = T;
    fel_const.calc(&fel_const);

// Initialize MRAS Estimation constant
    sel_const.Rr = RR;
    sel_const.Lr = LR;
    sel_const.fb = BASE_FREQ;
    sel_const.fc = 10;
    sel_const.Ts = T;
    sel_const.calc(&sel_const);

// Initialize Flux Estimation
    fel.K1 = _IQ(fel_const.K1);
    fel.K2 = _IQ(fel_const.K2);
    fel.K3 = _IQ(fel_const.K3);
    fel.K4 = _IQ(fel_const.K4);
    fel.K5 = _IQ(fel_const.K5);

```

```

    fe1.K6 = _IQ(fe1_const.K6);
    fe1.K7 = _IQ(fe1_const.K7);
    fe1.K8 = _IQ(fe1_const.K8);
    fe1.K9 = _IQ(fe1_const.K9);
    fe1.K10 = _IQ(sel_const.K3);
    fe1.K11 = _IQ(sel_const.K4);
    fe1.Kp = _IQ(0.168);
    fe1.Ki = _IQ(0.72*T);

// Initialize MRAS Estimation
    sel.KP = _IQ(120);
    sel.KI = _IQ(T*SpeedLoopPrescaler01*6622);
    sel.KC = _IQ(0.5);

// Initialize the PID module for Id
    pid1_id.Kp = _IQ(0.446);
    pid1_id.Ki = _IQ(T*112);
    pid1_id.Kd = _IQ(0/T);
    pid1_id.Kc = _IQ(0.5);
    pid1_id.OutMax = _IQ(0.3);
    pid1_id.OutMin = _IQ(-0.3);

// Initialize the PID module for Iq
    pid1_iq.Kp = _IQ(0.446);
    pid1_iq.Ki = _IQ(T*112);
    pid1_iq.Kd = _IQ(0/T);
    pid1_iq.Kc = _IQ(0.5);
    pid1_iq.OutMax = _IQ(0.95);
    pid1_iq.OutMin = _IQ(-0.95);

// Initialize the PID module for speed
    pid1_spd.Kp = _IQ(2);
    pid1_spd.Ki = _IQ(T*SpeedLoopPrescaler*3.2);
    pid1_spd.Kd = _IQ(0/T);
    pid1_spd.Kc = _IQ(0.5);
    pid1_spd.OutMax = _IQ(IqLim);
    pid1_spd.OutMin = _IQ(-IqLim);

// Enable global Interrupts and higher priority real-
// time debug events:
    EINT; // Enable Global interrupt INTM
    ERTM; // Enable Global realtime interrupt DBGM

// IDLE loop. Just sit and loop forever:
    for(;;) BackTicker++;
}

interrupt void MainISR(void)
{
// Verifying the ISR
    IsrTicker++;

////////////////////////////////////// SPI to TLV5618AI ////////////////////////////////////////

    EALLOW;

    GpioMuxRegs.GPFMUX.all=0x000F;

    EDIS;

```

```

// Initialize SPI FIFO registers
  SpiaRegs.SPIFFTX.all=0xE040;
  SpiaRegs.SPIFFRX.all=0x204F;
  SpiaRegs.SPIFFCT.all=0x0;
  SpiaRegs.SPICTL.all =0x000E;
  SpiaRegs.SPIBRR =0x007F;
  SpiaRegs.SPICCR.all =0x00DF;
  SpiaRegs.SPIPRI.bit.FREE = 1;

//----- Channel # 1 -----//

// Transmit data
  SpiaRegs.SPITXBUF=0xC000+sdata1;
// Wait until data is received
  while(SpiaRegs.SPIFFRX.bit.RXFFST !=1) { }
// Check against sent data
  rdata = SpiaRegs.SPIRXBUF;

//----- Channel # 2 -----//

// Transmit data
  SpiaRegs.SPITXBUF=0x4000+sdata2;
// Wait until data is received
  while(SpiaRegs.SPIFFRX.bit.RXFFST !=1) { }
// Check against sent data
  rdata1 = SpiaRegs.SPIRXBUF;

  sdata1 = (Speedfb*1000)+1000;
  sdata2 = ((se1.WmEst)*1000)+1000;

//////////////////////////////////// End DAC Output //////////////////////////////////////

//////////////////////////////////// Speed Step //////////////////////////////////////

  EALLOW;

  GpioMuxRegs.GPBMUX.all = 0x0FFF;
  GpioMuxRegs.GPBDIR.all = 0x0FFF;
  GpioMuxRegs.GPBQUAL.all= 0x0000;

  EDIS;

if(GpioDataRegs.GPBDAT.bit.GPIOB13 == 0){
    SpeedCom = 0.333;
}

if(GpioDataRegs.GPBDAT.bit.GPIOB14 == 0){
    SpeedCom = 0.94;
}

if(GpioDataRegs.GPBDAT.bit.GPIOB15 == 0){
    SpeedCom = -0.94;
}

```

```

// -----
//   Read ADC.
// -----
    adc04b_1.read(&adc04b_1);

//-----
//   DC-bus
//-----
// Convert raw result to Q15 (unipolar signal)
    DatQ1 = (AdcRegs.ADCRESULT0>>1)&0x7FFF;
// Tmp = gain*dat => Q28 = Q13*Q15
    Tmp1 = (int32)0x1FFF*(int32)DatQ1;
// Convert Q28 to Q15
    DcBus = (int16)(Tmp1>>13);

// -----
//   Connect inputs of the CLARKE module and call the
//   clarke transformation calculation function.
// -----
clarke1.As = _IQ15toIQ((int32)adc04b_1.Ch2Out); // IsA
clarke1.Bs = _IQ15toIQ((int32)adc04b_1.Ch3Out); // IsB
clarke1.calc(&clarke1);

//-----
//   Connect inputs of the PARK module and call the park
//   transformation calculation function.
// -----
park1.Alpha = clarke1.Alpha;
park1.Beta = clarke1.Beta;
park1.Angle = fe1.ThetaFlux;
park1.calc(&park1);

// -----
//   Connect inputs of the PID_REG3 module and call the
//   PI speed controller calculation function.
// -----
if (SpeedLoopCount==SpeedLoopPrescaler)
    {
        pid1_spd.Ref = _IQ(SpeedCom);
        pid1_spd.Fdb = _IQ(se1.WmEst);
        pid1_spd.calc(&pid1_spd);
        SpeedLoopCount=1;
    }
else SpeedLoopCount++;

// -----
//   Connect inputs of the PID_REG3 module and call the
//   PI Id controller calculation function.
// -----
    pid1_id.Ref = _IQ(IdRef);
    pid1_id.Fdb = park1.Ds;
    pid1_id.calc(&pid1_id);

// -----
//   Connect inputs of the PID_REG3 module and call the
//   PI Iq controller calculation function.
// -----
    pid1_iq.Ref = pid1_spd.Out;
    pid1_iq.Fdb = park1.Qs;
    pid1_iq.calc(&pid1_iq);

```

```

// -----
//   Connect inputs of the INV_PARK module and call the
//   inverse park transformation calculation function.
// -----
    ipark1.Ds = pid1_id.Out;
    ipark1.Qs = pid1_iq.Out;
    ipark1.Angle = fel.ThetaFlux;
    ipark1.calc(&ipark1);

// -----
//   Connect inputs of the SVGEN_DQ module and call the
//   space-vector gen. calculation function.
// -----
    svgen_dq1.Ualpha = ipark1.Alpha;
    svgen_dq1.Ubeta = ipark1.Beta;
    svgen_dq1.calc(&svgen_dq1);

// -----
//   Connect inputs of the PWM_DRV module and call the
//   PWM signal generation update function.
// -----
    pwm1.MfuncC1 = (int16)_IQtoIQ15(svgen_dq1.Ta); // MfuncC1 is
in Q15
    pwm1.MfuncC2 = (int16)_IQtoIQ15(svgen_dq1.Tb); // MfuncC2 is
in Q15
    pwm1.MfuncC3 = (int16)_IQtoIQ15(svgen_dq1.Tc); // MfuncC3 is
in Q15
    pwm1.update(&pwm1);

// -----
//   Connect inputs of the VOLT_CALC module and call the
//   phase voltage calculation function.
// -----
    volt1.DcBusVolt = _IQ15toIQ((int32)DcBus);
    volt1.MfuncV1 = svgen_dq1.Ta;
    volt1.MfuncV2 = svgen_dq1.Tb;
    volt1.MfuncV3 = svgen_dq1.Tc;
    volt1.calc(&volt1);

// -----
//   Connect inputs of the ACI module and call the flux
//   estimation calculation function.
// -----
    fel.UDsS = volt1.Valpha;
    fel.UQsS = volt1.Vbeta;
    fel.IDsS = clarkel.Alpha;
    fel.IQsS = clarkel.Beta;
    fel.Wm = sel.Wm;
    fel.calc(&fel);

// -----
//   Connect inputs of the ACI module and call the speed
//   estimation (MRAS) calculation function.
// -----
    if (SpeedLoopCount01==SpeedLoopPrescaler01)
    {
        sel.FrAlpha_v = fel.PsiDrS;
        sel.FrBeta_v = fel.PsiQrS;
        sel.FrAlpha_i = fel.FluxDrSs;
        sel.FrBeta_i = fel.FluxQrSs;
        sel.calc(&sel);
        SpeedLoopCount01=1;
    }

```

```

else SpeedLoopCount01++;

//-----
//   Speed Feedback
//-----
if(GpioDataRegs.GPBDAT.bit.GPIOB12 == 0){

    Direct = 1;
    DatQ2 = (AdcRegs.ADCRESULT3>>1)&0x7FFF;
    Tmp2 = (int32)0x1FFF*(int32)DatQ2;
    ChOut2 = (int16)(Tmp2>>13);
    Speedfb = (_IQ15toF(ChOut2))*Direct;
}

if(GpioDataRegs.GPBDAT.bit.GPIOB12 == 1){

    Direct = -1;
    DatQ2 = (AdcRegs.ADCRESULT4>>1)&0x7FFF;
    Tmp2 = (int32)0x1FFF*(int32)DatQ2;
    ChOut2 = (int16)(Tmp2>>13);
    Speedfb = (_IQ15toF(ChOut2))*Direct;
}

    spd = _IQ(Speedfb);

// -----
//   Connect inputs of the PWMDAC module
// -----
    PwmDacCh1 = (int16)_IQtoIQ15(spd-se1.Est);
    PwmDacCh2 = (int16)_IQtoIQ15(fe1.FluxDrSs);
    PwmDacCh3 = (int16)_IQtoIQ15(fe1.PsiDrS);

// -----
//   Call the PWMDAC update function.
// -----
    pwmdac1.update(&pwmdac1);

// -----
//   Call the DATALOG update function.
// -----
    dlog.update(&dlog);

// Enable more interrupts from this timer
    EvaRegs.EVAIMRA.bit.T1UFINT = 1;

// Note: To be safe, use a mask value to write to the entire
// EVAIFRA register. Writing to one bit will cause a
// read-modify-write
// operation that may have the result of writing 1's to clear
// bits other than those intended.
    EvaRegs.EVAIFRA.all = BIT9;

// Acknowledge interrupt to receive more interrupts from
// PIE group 2
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP2;
}

```

clarke.h

```

#ifndef __CLARKE_H__
#define __CLARKE_H__

```

```

typedef struct {
    _iq As;
    _iq Bs;
    _iq Alpha;
    _iq Beta;
} CLARKE;

void (*calc)(); // Pointer to calculation function
} CLARKE;

typedef CLARKE *CLARKE_handle;
/*-----
Default initializer for the CLARKE object.
-----*/
#define CLARKE_DEFAULTS { 0, \
                          0, \
                          0, \
                          0, \
void (*)(Uint32))clarke_calc }
/*-----
Prototypes for the functions in CLARKE.C
-----*/
void clarke_calc(CLARKE_handle);

#endif // __CLARKE_H__

```

clarke.c

```

#include "clarke.h"
/* 1/sqrt(3) = 0.57735026918963 */

void clarke_calc(CLARKE *v)
{
    v->ds = v->as;
    v->qs = ((v->as + 2 * v->bs) * 0.57735026918963);
}

```

park.h

```

#ifndef __PARK_H__
#define __PARK_H__

typedef struct {
    _iq Alpha;
    _iq Beta;
    _iq Angle;
    _iq Ds;
    _iq Qs;
} PARK;

void (*calc)(); // Pointer to calculation function
} PARK;

typedef PARK *PARK_handle;
/*-----
Default initializer for the PARK object.
-----*/
#define PARK_DEFAULTS { 0, \
                       0, \
                       0, \
                       0, \
                       0, \
                       (void (*)(Uint32))park_calc }
/*-----
Prototypes for the functions in PARK.C
-----*/
void park_calc(PARK_handle);

```

```
#endif // __PARK_H__
```

park.c

```
#include "park.h"
void park_calc(PARK *v) {
double cos_ang, sin_ang;
sin_ang = v->sinAng;
cos_ang = v->cosAng;
v->de = v->ds * cos_ang + v->qS * sin_ang;
v->qe = v->qS * cos_ang - v->ds * sin_ang;
}
```

ipark.h

```
#ifndef __IPARK_H__
#define __IPARK_H__

typedef struct {
    _iq Alpha;
    _iq Beta;
    _iq Angle;
    _iq Ds;
    _iq Qs;
} IPARK;

void (*calc)(); // Pointer to calculation function
} IPARK;

typedef IPARK *IPARK_handle;
/*-----
Default initializer for the IPARK object.
-----*/
#define IPARK_DEFAULTS { 0, \
                        0, \
                        0, \
                        0, \
                        0, \
}
(void (*)(Uint32))ipark_calc }
/*-----
Prototypes for the functions in IPARK.C
-----*/
void ipark_calc(IPARK_handle);

#endif // __IPARK_H__
```

ipark.c

```
#include "ipark.h"
void ipark_calc(IPARK *v) {
double cos_ang, sin_ang;

sin_ang = v->sinAng;
cos_ang = v->cosAng;
v->ds = v->de * cos_ang - v->qe * sin_ang;
v->qS = v->qe * cos_ang + v->de * sin_ang;
}
```

svgen_dq.h

```
#ifndef __SVGEN_DQ_H__
#define __SVGEN_DQ_H__

typedef struct {
    _iq Ualpha;
    _iq Ubeta;
    _iq Ta;
    _iq Tb;
}
```

```

        _iq Tc;
void (*calc)(); // Pointer to calculation function
} SVGENDQ;

typedef SVGENDQ *SVGENDQ_handle;
/*-----
Default initializer for the SVGENDQ object.
-----*/
#define SVGENDQ_DEFAULTS { 0,0,0,0,0, \
                          (void (*)(Uint32))svgendq_calc }

/*-----
Prototypes for the functions in SVGEN_DQ.C
-----*/
void svgendq_calc(SVGENDQ_handle);

#endif // __SVGEN_DQ_H__

```

svgen_dq.c

```

#include "svgen_dq.h"
void svgendq_calc(SVGENDQ *v){
double Va,Vb,Vc,t1,t2;
int sector = 0;
/* 0.8660254 = sqrt(3)/2 */
/* Inverse clarke transformation */
Va = v->Ubeta;
Vb = -0.5 * v->Ubeta + 0.8660254 * v->Ualfa;
Vc = -0.5 * v->Ubeta - 0.8660254 * v->Ualfa;
/* 60 degree sector determination */
if (Va>0)
sector = 1;
if (Vb>0)
sector = sector + 2;
if (Vc>0)
sector = sector + 4;
/* X,Y,Z (Va,Vb,Vc) calculations */
//sqrt(3)*50us/300V
Va = v->Ubeta * 0.000000288675134;
Vb = (0.000000144337567 * v->Ubeta + 0.00000025 * v->Ualfa);
Vc = (0.000000144337567 * v->Ubeta - 0.00000025 * v->Ualfa);

if (sector==1){ /* sector 1: t1=Z and t2=Y (abc ---> Tb,Ta,Tc) */
t1 = Vc;
t2 = Vb;
v->Tb = 0.5 * (0.00005-t1-t2);/* tbon = (1-t1-t2)/2 */
v->Ta = v->Tb+t1; /* taon = tbon+t1 */
v->Tc = v->Ta+t2; /* tcon = taon+t2 */
}

else if (sector==2){
t1 = Vb;
t2 = -Va;
v->Ta = 0.5*(0.00005-t1-t2);/* taon = (1-t1-t2)/2 */
v->Tc = v->Ta+t1; /* tcon = taon+t1 */
v->Tb = v->Tc+t2; /* tbon = tcon+t2 */
}

else if (sector==3){/* sector 3: t1=-Z and t2=X (Ta,Tb,Tc) */
t1 = -Vc;
t2 = Va;
v->Ta = 0.5*(0.00005-t1-t2);/* taon = (1-t1-t2)/2 */

```

```

v->Tb = v->Ta+t1; /* tbon = taon+t1 */
v->Tc = v->Tb+t2; /* tcon = tbon+t2 */
}

else if (sector==4) /* sector 4: t1=-X and t2=Z (Tc,Tb,Ta) */
t1 = -Va;
t2 = Vc;
v->Tc = 0.5*(0.00005-t1-t2);/* tcon = (1-t1-t2)/2 */
v->Tb = v->Tc+t1; /* tbon = tcon+t1 */
v->Ta = v->Tb+t2; /* taon = tbon+t2 */
}
else if (sector==5){ /* sector 5: t1=X and t2=-Y (Tb,Tc,Ta) */
t1 = Va;
t2 = -Vb;
v->Tb = 0.5*(0.00005-t1-t2);/* tbon = (1-t1-t2)/2 */
v->Tc = v->Tb+t1; /* tcon = tbon+t1 */
v->Ta = v->Tc+t2; /* taon = tcon+t2 */
}

else if (sector==6){ /* sector 6: t1=-Y and t2=-Z (Tc,Ta,Tb) */
t1 = -Vb;
t2 = -Vc;
v->Tc = 0.5*(0.00005-t1-t2);/* tcon = (1-t1-t2)/2 */
v->Ta = v->Tc+t1; /* taon = tcon+t1 */
v->Tb = v->Ta+t2; /* tbon = taon+t2 */
}
v->Ta = v->Ta * 20000;
v->Tb = v->Tb * 20000;
v->Tc = v->Tc * 20000;
}

```

f281xpwm.h

```

#ifndef __F281X_PWM_H__
#define __F281X_PWM_H__
#include "f281xbmsk.h"

#define PWM_INIT_STATE (FREE_RUN_FLAG + \
TIMER_CONT_UPDN + \
TIMER_CLK_PRESCALE_X_1 + \
TIMER_ENABLE_BY_OWN + \
TIMER_ENABLE)

#define ACTR_INIT_STATE ( COMPARE1_AH + \
COMPARE2_AL + \
COMPARE3_AH + \
COMPARE4_AL + \
COMPARE5_AH + \
COMPARE6_AL )

#define DBTCON_INIT_STATE ( DBT_VAL_10 + \
EDBT3_EN + \
EDBT2_EN + \
EDBT1_EN + \
DBTPS_X8 )

typedef struct {
  Uint16 PeriodMax;
  double MfuncC1; // Input: PWM 1&2 Duty cycle ratio (Q15)
  double MfuncC2; // Input: PWM 3&4 Duty cycle ratio (Q15)
  double MfuncC3; // Input: PWM 5&6 Duty cycle ratio (Q15)
  void (*init)(); // Pointer to the init function
  void (*update)(); // Pointer to the update function
}

```

```

} PWMGEN ;
typedef PWMGEN *PWMGEN_handle;
#define F281X_EV1_FC_PWM_GEN {1000, \
0x4000, \
0x4000, \
0x4000, \
(void (*)(Uint32))F281X_EV1_PWM_Init, \
(void (*)(Uint32))F281X_EV1_PWM_Update \
}
#define PWMGEN_DEFAULTS F281X_EV1_FC_PWM_GEN
void F281X_EV1_PWM_Init(PWMGEN_handle);
void F281X_EV1_PWM_Update(PWMGEN_handle);
#endif // __F281X_PWM_H__

```

f281xpwm.c

```

#include "DSP281x_Device.h"
#include "f281xpwm.h"

```

```

void F281X_EV1_PWM_Init(PWMGEN *p){

EvaRegs.T1PR = p->PeriodMax; // Init Timer 1 period Register
EvaRegs.T1CON.all = 0x8840; // Symmetrical Operation
EvaRegs.DBTCONA.all = DBTCON_INIT_STATE; // Init DBTCONA
EvaRegs.ACTRA.all = ACTR_INIT_STATE; // Init ACTRA
EvaRegs.COMCONA.all = 0xA200; // Init COMCONA
EvaRegs.CMPR1 = p->PeriodMax; // Init CMPR1
EvaRegs.CMPR2 = p->PeriodMax; // Init CMPR2
EvaRegs.CMPR3 = p->PeriodMax; // Init CMPR3

EALLOW; // Enable EALLOW
GpioMuxRegs.GPAMUX.all |= 0x003F;
EDIS; // Disable EALLOW
}

void F281X_EV1_PWM_Update(PWMGEN *p){

EvaRegs.CMPR1 = p->MfuncC1 * p->PeriodMax;
EvaRegs.CMPR2 = p->MfuncC2 * p->PeriodMax;
EvaRegs.CMPR3 = p->MfuncC3 * p->PeriodMax;
}

```

volt_calc.h

```

#ifndef __VOLT_CAL_H__
#define __VOLT_CAL_H__

```

```

typedef struct {
    _iq DcBusVolt;
    _iq MfuncV1;
    _iq MfuncV2;
    _iq MfuncV3;
    Uint32 OutOfPhase;
    _iq VphaseA;
    _iq VphaseB;
    _iq VphaseC;
    _iq Valpha;
    _iq Vbeta;
} PHASEVOLTAGE;

void (*calc)();
typedef PHASEVOLTAGE *PHASEVOLTAGE_handle;

```

```

/*

```

```

OutOfPhase = 1 for the out of phase correction if
* MfuncV1 is out of phase with PWM1,

```



```

// Phase C
v->VphaseC = _IQmpy(TWO_THIRD,v->MfuncV3);
v->VphaseC -= _IQmpy(ONE_THIRD,v->MfuncV1);
v->VphaseC -= _IQmpy(ONE_THIRD,v->MfuncV2);

// Voltage transformation (a,b,c) -> (Alpha,Beta) Alpha-axis
v->Valpha = v->VphaseA;

// Beta-axis
v->Vbeta = _IQmpy((v->VphaseA + _IQmpy(_IQ(2),v->VphaseB)), INV_SQRT3);
}

```

pid_reg3.h

```

#ifndef __PIDREG3_H__
#define __PIDREG3_H__

typedef struct {
    _iq Ref;
    _iq Fdb;
    _iq Err;
    _iq Kp;
    _iq Up;
    _iq Ui;
    _iq OutPreSat;
    _iq OutMax;
    _iq OutMin;
    _iq Out;
    _iq SatErr;
    _iq Ki;
    _iq Kc;

    void (*calc)(); // Pointer to calculation function
} PIDREG3;

typedef PIDREG3 *PIDREG3_handle;
/*-----
Default initializer for the PIDREG3 object.
-----*/
#define PIDREG3_DEFAULTS { 0, \
    0, \
    0, \
    _IQ(1.3), \
    0, \
    0, \
    0, \
    _IQ(1), \
    _IQ(-1), \
    0, \
    0, \
    _IQ(0.02), \
    _IQ(0.5), \
    _IQ(1.05), \
    0, \
(void (*)(Uint32))pid_reg3_calc }

/*-----
Prototypes for the functions in PIDREG3.C
-----*/
void pid_reg3_calc(PIDREG3_handle);

```

```
#endif // __PIDREG3_H__
```

pid_reg3.c

```
#include "IQmathLib.h"           // Include header for IQmath
library
// Don't forget to set a proper GLOBAL_Q in "IQmathLib.h" file
#include "dmctype.h"
#include "pid_reg3.h"

void pid_reg3_calc(PIDREG3 *v)
{
    // Compute the error
    v->Err = v->Ref - v->Fdb;

    // Compute the proportional output
    v->Up = _IQmpy(v->Kp,v->Err);

    // Compute the pre-saturated output
    v->OutPreSat = v->Up + v->Ui;

    // Saturate the output
    if (v->OutPreSat > v->OutMax)
        v->Out = v->OutMax;
    else if (v->OutPreSat < v->OutMin)
        v->Out = v->OutMin;
    else
        v->Out = v->OutPreSat;

    // Compute the saturate difference
    v->SatErr = v->Out - v->OutPreSat;

    // Compute the integral output
    v->Ui = v->Ui + _IQmpy(v->Ki,v->Err) + _IQmpy(v->Kc,v->
>SatErr);
}

```

aci_fe.h

```
#ifndef __ACI_FE_H__
#define __ACI_FE_H__

typedef struct {
    _iq  ThetaFlux;
    _iq  IDsS;
    _iq  IQsS;
    _iq  K1;
    _iq  FluxDrE;
    _iq  K2;
    _iq  FluxDrS;
    _iq  FluxQrS;
    _iq  K3;
    _iq  K4;
    _iq  FluxDsS;
    _iq  FluxQsS;
    _iq  PsiDsS;
    _iq  Kp;
    _iq  UiDsS;
    _iq  UCompDsS;
    _iq  Ki;
    _iq  PsiQsS;
    _iq  UiQsS;
    _iq  UCompQsS;
}

```



```

/*-----
Prototypes for the functions in ACI_FE.C
-----*/
void aci_fe_calc(ACIFE_handle);

#endif // __ACI_FE_H__

aci_fe.c
#include "IQmathLib.h" // Include header for IQmath
library
// Don't forget to set a proper GLOBAL_Q in "IQmathLib.h" file
#include "dmctype.h"
#include "aci_fe.h"

void aci_fe_calc(ACIFE *v)
{
    _iq IDSE, Error, OldEmf;

    // Lowpass Filter
    v->IDSS = _IQmpy(v->K10,v->IDSS) + v->K11;
    v->IQSS = _IQmpy(v->K10,v->IQSS) + v->K11;
    v->UDSS = _IQmpy(v->K10,v->UDSS) + v->K11;
    v->UQSS = _IQmpy(v->K10,v->UQSS) + v->K11;

    // Park transformation on the measured stator current
    IDSE = _IQmpy(v->IQSS,_IQsinPU(v->ThetaFlux));
    IDSE += _IQmpy(v->IDSS,_IQcosPU(v->ThetaFlux));

    // The current model section (Classical Rotor Flux Vector Control
    // Equation)
    v->FluxDrE = _IQmpy(v->K1,v->FluxDrE) + _IQmpy(v->K2,IDSE);

    // Inverse park transformation on the rotor flux from the current
    // model
    v->FluxDrS = _IQmpy(v->FluxDrE,_IQcosPU(v->ThetaFlux));
    v->FluxQrS = _IQmpy(v->FluxDrE,_IQsinPU(v->ThetaFlux));

    // Compute the Rotor flux based on the rotor flux from current
    // model(Using for MRAS System)
    v->FluxDrSs = _IQmpy(v->K1,v->FluxDrSs) + _IQmpy(v->K2,v-
>IDSS) - _IQmpy(_IQ(31.4),_IQmpy(v->Wm,_IQmpy(v->K9,v-
>FluxQrSs)));
    v->FluxQrSs = _IQmpy(v->K1,v->FluxQrSs) + _IQmpy(v->K2,v-
>IQSS) + _IQmpy(_IQ(31.4),_IQmpy(v->Wm,_IQmpy(v->K9,v-
>FluxDrSs)));

    // Compute the stator flux based on the rotor flux from current
    // model
    v->FluxDsS = _IQmpy(v->K3,v->FluxDrSs) + _IQmpy(v->K4,v-
>IDSS);
    v->FluxQsS = _IQmpy(v->K3,v->FluxQrSs) + _IQmpy(v->K4,v-
>IQSS);

    // Conventional PI controller section
    Error = v->PsiDsS - v->FluxDsS;
    v->UCompDsS = _IQmpy(v->Kp,Error) + v->UiDsS;
    v->UiDsS = _IQmpy(v->Ki,Error) + v->UiDsS;
    Error = v->PsiQsS - v->FluxQsS;
    v->UCompQsS = _IQmpy(v->Kp,Error) + v->UiQsS;
    v->UiQsS = _IQmpy(v->Ki,Error) + v->UiQsS;

    // Compute the estimated stator flux based on the integral of

```


ประวัติผู้วิจัย

นาย ฐฐพงษ์ เมืองจันทร์ เกิดเมื่อวันที่ 13 สิงหาคม พ.ศ. 2522 ที่อำเภอเมือง จ.พิจิตร

ประวัติการศึกษา

- ปี 2537 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น จากโรงเรียนวัดคงกลาง อ.เมือง จ.พิจิตร
- ปี 2541 สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพ (ปวช.) สาขาวิชาช่างไฟฟ้า จากวิทยาลัยเทคนิคพิจิตร
- ปี 2545 สำเร็จการศึกษาระดับปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขา วิศวกรรมไฟฟ้า เกียรตินิยมอันดับ 1 จากมหาวิทยาลัยอีสเทิร์นเอเซีย
- ปี 2546 - ปัจจุบัน เป็นอาจารย์ประจำ คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า มหาวิทยาลัยอีสเทิร์นเอเซีย