ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบ
โดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI

โดย นายเอกชัย พรรณวัลย์

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

_____ คณบดีบัณฑิตวิทยาลัย

(อาจารย์ ดร.มงคล หวังสถิตย์วงษ์)

21 พฤษภาคม 2550

คณะกรรมการสอบวิทยานิพนธ์

_____ ประธานกรรมการ

(รองศาสตราจารย์ ดร.สุขสันติ์ นุ่นงาม)

_____ กรรมการ

(รองศาสตราจารย์ ดร.วิบูลย์ ชื่นเขก)

_____ กรรมการ

(รองศาสตราจารย์สุมาลี อุณหวณิชย์)

การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบ โดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI

นายเอกชัย พรรณวัลย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2549
ลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ชื่อ : นายเอกชัย พรรณวัลย์
ชื่อวิทยานิพนธ์ : การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรง
ชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิง
ร่วมกับตัวชดเชยแบบ PI
สาขาวิชา : วิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ที่ปรึกษาวิทยานิพนธ์ : รองศาสตราจารย์ ดร.สุชสันติ์ นุ่นงาม
ปีการศึกษา : 2549

บทคัดย่อ

อุตสาหกรรมในปัจจุบันได้ให้ความสนใจกับวิธีการควบคุมแบบแรงบิดโดยตรง ซึ่งเป็นวิธีการปรับความเร็วรอบในมอเตอร์ไฟฟ้ากระแสสลับที่ให้การตอบสนองที่เร็วและมีความคงทนต่อการเบี่ยงเบนของค่าพารามิเตอร์ การควบคุมแบบแรงบิดโดยตรงจะเป็นการควบคุมโดยอ้างอิงพารามิเตอร์ทางด้านสเตเตอร์เป็นหลัก จึงเหมาะสมกับการควบคุมความเร็วโดยไม่ใช้เอ็นโคดเดอร์ ในงานวิจัยนี้นำเสนอการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงไร้ตัวตรวจจับความเร็วรอบ และได้นำ ตัวประมาณแบบสไลดิงร่วมกับตัวควบคุมแบบ PI โดยพิจารณาทั้งฟลักซ์ที่สเตเตอร์และฟลักซ์ที่โรเตอร์ โดยทำการทดลองบนตัวประมวลผลดิจิตอลเบอร์ TMS320F243 กับมอเตอร์เหนี่ยวนำขนาด 1.1 kW จากการทดลองให้ผลเป็นที่น่าพอใจ

(วิทยานิพนธ์มีจำนวนทั้งสิ้น 93 หน้า)

คำสำคัญ : การควบคุมแบบแรงบิดโดยตรง,ตัวสังเกตการณ์แบบสไลดิง

_____อาจารย์ที่ปรึกษาวิทยานิพนธ์

Name : Mr.Akekachai Pannawan

Thesis Title : A Speed-Sensorless Direct-Torque Control of an Induction Motor using a Sliding-Mode Observer and PI Compensators

Major Field : Electrical Engineering

King Mongkut's Institute of Technology North Bangkok

Thesis Advisor : Associate Professor Dr.Suksan Numngam

Academic Year : 2006

Abstract

Industries have paid attention to Direct Torque Control (DTC) techniques in variable-speed AC-drives. The DTC technique gives fast response and robustness due to parameter deviation. The control is based on the stator parameters, therefore it is suitable for the speed control without an encoder. This thesis presents of a sensorless DTC for an induction-motor drive. A sliding-mode observer is applied in conjunction with PI controllers. This technique considers both stator and rotor fluxes. The control algorithm is implemented on a TMS320F243-DSP. The experiment is carried on a 1.1 kW indution motor and good experimental results are obtained.

(Total 93 pages)

Keywords : Direct Torque Control,Sliding-Mode Observer

_____ Advisor

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงด้วยดี โดยได้รับความช่วยเหลือจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ รองศาสตราจารย์ ดร. สุขสันติ์ นุ่นงาม ที่ได้กรุณาให้คำปรึกษาในการแก้ปัญหาต่าง ๆและให้โอกาสถึงแม้ผู้วิจัยจะหายไปบ้างแต่ท่านก็ยังให้โอกาสอยู่เสมอมาตลอดระยะเวลาการทำวิจัย รวมทั้งการตรวจสอบแก้ไขข้อบกพร่องและสนับสนุนอุปกรณ์ต่างๆ จนวิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี

ผู้วิจัยขอขอบคุณเพื่อนๆ และพี่ๆ ทุกท่านที่ประจำศูนย์วิจัยและพัฒนาระบบอัจฉริยะ (ศ.ว.พ.อ) (Research and Development Center for Intelligent Systems : RCIS) คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่ให้คำแนะนำที่ดีและเป็นประโยชน์ต่อการดำเนินงานวิจัยจนวิทยานิพนธ์นี้สำเร็จได้ด้วยดี โดยเฉพาะพี่ ปาลรัฐ ไกลนรา(พี่เน) พี่โรจน์ พี่หนุ่ม ที่คอยช่วยเหลือและให้คำปรึกษาตลอดเวลาแม้ดึกดื่น ขอบคุณพี่ดุสิต(พี่โอ๋) ที่คอยช่วยเตือนเรื่องเอกสารต่าง ๆ ขอบคุณเพื่อน MEE โอเล่ มด โอ๊ต เอ๊ด หนึ่ง และอีกหลายท่าน ที่เป็นเพื่อนร่วมทุกข์ร่วมสุขตลอดระยะเวลาสี่ปีที่ผ่านมา

ท้ายสุดนี้ขอขอบพระคุณ บิดา-มารดา ลุงป้าน้าอา และ คุณย่า ที่คอยช่วยเหลือและเป็นกำลังใจตลอดระยะเวลาการศึกษาและให้ความสนับสนุนในด้านปัจจัยในการดำเนินการทำวิทยานิพนธ์ฉบับนี้

เอกชัย พรรณวัลย์

สารบัญ

		หน้า
บทคัดย่อภาษาไทย		ข
บทคัดย่อภาษาอังกฤษ		ค
กิตติกรรมประกาศ		ง
สารบัญตาราง		ง
สารบัญภาพ		ช
คำอธิบายสัญลักษณ์และคำย่อ		ฎ
บทที่ 1	บทนำ	1
1.1	ความเป็นมาและความสำคัญของปัญหา	3
1.2	วัตถุประสงค์	3
1.3	ขอบเขตของการวิจัย	3
1.4	ขั้นตอนและวิธีการวิจัย	3
1.5	ประโยชน์ของการวิจัย	4
บทที่ 2	ทฤษฎีเบื้องต้น	5
2.1	แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำ	5
2.2	อินเวอร์เตอร์และสเปซเวกเตอร์แรงดัน	14
2.3	สรุป	17
บทที่ 3	การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงและการประมาณความเร็วรอบ	19
3.1	หลักการควบคุมมอเตอร์เหนี่ยวนำแบบสไลดิงแรงบิดตรง	20
3.2	ระบบควบคุมโครงสร้างผันแปร	21
3.3	ตัวสังเกตการณ์แบบสไลดิงและตัวชดเชยแบบ PI	23
3.4	การประมาณความเร็วของมอเตอร์เหนี่ยวนำแบบวงรอบเปิด	26
3.5	สรุป	29
บทที่ 4	การจำลองระบบด้วยคอมพิวเตอร์	31
4.1	ผลการทดสอบแบบจำลอง	31
4.2	สรุป	34
บทที่ 5	การทดลอง	35
5.1	อุปกรณ์ที่ใช้ในการทดลอง	35
5.2	การทดลอง	39
5.3	วิธีทดลอง	40

สารบัญ (ต่อ)

		หน้า
	5.4 สรุป	40
บทที่ 6	การทดสอบระบบควบคุมมอเตอร์เหนี่ยวนำ	41
	6.1 ผลการทดสอบแบบจำลองการทดสอบโปรแกรมการทำงานของตัวสังเกตการสไลดิงแบบเปิดวงรอบ	42
	6.2 ผลการทดสอบระบบควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ไม่ใส่ภาระ	46
	6.3 ผลการทดสอบระบบควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ใส่ภาระ	46
	6.4 วิเคราะห์และสรุปผลการทดลอง	50
บทที่ 7	สรุปผลและข้อเสนอแนะ	51
	7.1 สรุปผลการวิจัย	51
	7.2 ข้อเสนอแนะ	52
เอกสารอ้างอิง		53
ภาคผนวก ก	ข้อมูลมอเตอร์เหนี่ยวนำ และค่าฐานต่างๆ	54
ภาคผนวก ข	ลำดับการทำงานของโปรแกรมที่ใช้ในงานวิจัย	57
ภาคผนวก ค	โปรแกรมที่ใช้ในงานวิจัย	61
ประวัติผู้วิจัย		93

สารบัญตาราง

ตารางที่		หน้า
4-1	พารามิเตอร์ของมอเตอร์เหนี่ยวนำ	32
5-1	ค่าพารามิเตอร์ของมอเตอร์เหนี่ยวนำที่ใช้ทดลอง	36

สารบัญภาพ

ภาพที่		หน้า
2-1	ภาพตัดขวางของมอเตอร์เหนี่ยวนำ	5
2-2	วงจรสมมูลของมอเตอร์เหนี่ยวนำ	6
2-3	สเปซเวกเตอร์ของกระแสสเตเตอร์	8
2-4	ความสัมพันธ์ระหว่างแกนอ้างอิงคงที่และแกนอ้างอิงที่หมุน	9
2-5	แหล่งจ่ายไฟอิสระ 3 ชุด	15
2-6	สเปซเวกเตอร์ของสเตเตอร์ฟลักซ์และสเปซเวกเตอร์ของโรเตอร์ฟลักซ์	17
3-1	โครงสร้างภายในของการควบคุมฟลักซ์และแรงบิดแบบแรงบิดโดยตรง	22
3-2	การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบ	22
3-3	โครงสร้างของตัวสังเกตการณ์แบบสไลดิ้งและตัวชดเชยแบบ PI	23
3-4	โครงสร้างของการควบคุมแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็ว	29
4-1	แบบจำลองระบบ	32
4-2	กระแสสเตเตอร์	32
4-3	ฟลักซ์ที่สเตเตอร์	33
4-4	สัญญาณแรงบิด	33
4-5	สัญญาณความเร็วรอบ	33
4-6	สัญญาณระนาบการสไลดิง	34
5-1	ชุดประมวลผลสัญญาณดิจิตอล TMS320F243 Evaluation Module	35
5-2	วงจรวัดกระแส	37
5-3	วงจรภายในชุดเชื่อมต่อระหว่างตัวประมวลผลกับชุดขับเคลื่อนมอเตอร์	37
5-4	การเชื่อมต่อระหว่างตัวประมวลผลกับชุดขับเคลื่อนมอเตอร์	38
5-5	มอเตอร์เหนี่ยวนำติดตั้งชุดภาระและตัวตรรวจจับความเร็วเอนโคดเดอร์	38
5-6	โครงสร้างฮาร์ดแวร์ของการทดลอง	39
6-1	กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ	41
6-2	ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ	42
6-3	แรงบิดประมาณของมอเตอร์เหนี่ยวนำ	42
6-4	ขนาดของสเตเตอร์ฟลักซ์ของมอเตอร์เหนี่ยวนำ	43
6-5	ทางเดินของสเตเตอร์ฟลักซ์ของแกนอัลฟ่าและเบต้า	43
6-6	ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ	44

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
6-7	กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ	44
6-8	ทางเดินของสเตเตอร์ฟลักซ์ของแกนอัลฟ่าและเบต้า	45
6-9	ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ	46
6-10	แรงบิดประมาณของมอเตอร์เหนี่ยวนำ	46
6-11	สเตเตอร์ฟลักซ์ของมอเตอร์เหนี่ยวนำ	47
6-12	กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ	47
6-13	ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ	48
6-14	ทางเดินของสเตเตอร์ฟลักซ์ของแกนอัลฟ่าและเบต้า	48
6-15	แรงบิดประมาณของมอเตอร์เหนี่ยวนำ	49
6-16	กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ	49

คำอธิบายสัญลักษณ์และคำย่อ

DSP	Digital Signal Possessor (ตัวประมวลผลดิจิตอล)
DTC	Direct Torque Control (การควบคุมแรงบิดตรง)
SMO	Sliding-Mode Observer ตัวสังเกตการณ์สไลดิง
u_{sA}, u_{sB}, u_{sC}	แรงดันแต่ละเฟสที่ขดสเตเตอร์
i_{sA}, i_{sB}, i_{sC}	กระแสแต่ละเฟสที่ขดสเตเตอร์
i_{ra}, i_{rb}, i_{rc}	กระแสแต่ละเฟสที่ขดโรเตอร์
$i_{s\alpha}\ i_{s\beta}$	กระแสสเตเตอร์ที่อ้างอิงบนแกนสเตเตอร์, A
$\psi_{sA}, \psi_{sB}, \psi_{sC}$	ฟลักซ์เหนี่ยวคล้องที่ตัดขดลวดแต่ละเฟสทางด้านสเตเตอร์
$\psi_{ra}, \psi_{rb}, \psi_{rc}$	ฟลักซ์เหนี่ยวคล้องที่ตัดขดลวดแต่ละเฟสทางด้านโรเตอร์
$\bar{\mathbf{u}}_s(t)$	สเปซเวกเตอร์ของแรงดันที่ขดลวดสเตเตอร์
$\bar{\mathbf{i}}_s(t)$	สเปซเวกเตอร์ของกระแสในขดลวดสเตเตอร์
$\bar{\mathbf{i}}_r(t)$	สเปซเวกเตอร์ของกระแสในขดลวดโรเตอร์
$\bar{\boldsymbol{\psi}}_s(t)$	สเปซเวกเตอร์ของฟลักซ์ที่เกิดขึ้นในขดลวดสเตเตอร์
$\bar{\boldsymbol{\psi}}_r(t)$	สเปซเวกเตอร์ของฟลักซ์ที่เกิดขึ้นในขดลวดโรเตอร์
R_s	ความต้านทานของขดลวดสเตเตอร์แต่ละเฟส
R_r	ความต้านทานของขดลวดโรเตอร์แต่ละเฟส
L_s	ค่าความเหนี่ยวนำตัวเองด้านสเตเตอร์
L_r	ค่าความเหนี่ยวนำตัวเองด้านโรเตอร์
L_m	ค่าความเหนี่ยวนำของสนามแม่เหล็ก
θ_r	มุมของโรเตอร์ที่วัดระหว่างขดลวดของสเตเตอร์เฟสเอและขดลวดของโรเตอร์เฟสเอ
ω	ความเร็วเชิงมุมของแกนอ้างอิงที่หมุนที่ใดๆ
ω_r	ความเร็วเชิงมุมของโรเตอร์
J_m	โมเมนต์ความเฉื่อย (กิโลกรัม-เมตร^2)
ω_m	ความเร็วเชิงมุมทางกลของโรเตอร์ (เรเดียนต่อวินาที)
T_e	แรงบิดของมอเตอร์ (นิวตัน-เมตร)
T_L	แรงบิดของโหลด (นิวตัน-เมตร)
σ	ค่าตัวประกอบของการรั่วไหล (Leakage Factor)
L_{ls}	ค่าความเหนี่ยวนำของการรั่วไหลของขดลวดสเตเตอร์
L_{lr}	ค่าความเหนี่ยวนำของการรั่วไหลของขดลวดโรเตอร์

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

$\theta_{\psi r}$	มุมของโรเตอร์ฟลักซ์บนแกนอ้างอิงคงที่
rpm	รอบต่อนาที
$u_{s\alpha}\, u_{s\beta}$	แรงดันสเตเตอร์ที่อ้างอิงบนแกนสเตเตอร์, V
$\theta_{\psi s}$	มุมของสเตเตอร์ฟลักซ์บนแกนอ้างอิงคงที่
P	จำนวนคู่ของขั้วแม่เหล็ก (Pole Pairs)
$\bar{\boldsymbol{\psi}}_s$	สเปซเวกเตอร์ของสเตเตอร์ฟลักซ์
$\bar{\mathbf{i}}_s$	สเปซเวกเตอร์ของกระแสสเตเตอร์
Ψ_s	ขนาดของสเตเตอร์ฟลักซ์เวกเตอร์
Ψ_r	ขนาดของสเตเตอร์ฟลักซ์เวกเตอร์
θ	มุมของของฟลักซ์คล้องสเตเตอร์ มื่อเทียบกับฟลักซ์คล้องโรเตอร์
ω_d	ความเร็วเชิงมุมของของฟลักซ์คล้องสเตเตอร์ มื่อเทียบกับฟลักซ์คล้องโรเตอร์
$\bar{\boldsymbol{\psi}}_s^{\psi_s}$	เวกเตอร์ของสเตเตอร์ฟลักซ์บนแกนสเตเตอร์ฟลักซ์
$S_{\Psi s}$	ระนาบการสไลด์ของสเตเตอร์ฟลักซ์
S_{Te}	ระนาบการสไลด์ของแรงบิด
$C_{\Psi s}$	ค่าคงตัวอัตราเร็วของการสไลด์ฟลักซ์
C_{Te}	ค่าคงตัวอัตราเร็วของการสไลด์แรงบิด
$\bar{\boldsymbol{\psi}}_r^e$	เวกเตอร์ของโรเตอร์ฟลักซ์บนแกนโรเตอร์ฟลักซ์
$\bar{\mathbf{i}}_s^{\psi_s}$	เวกเตอร์ของกระแสสเตเตอร์บนแกนสเตเตอร์ฟลักซ์
$\bar{\mathbf{u}}_s^{\psi_s}$	เวกเตอร์ของแรงดันสเตเตอร์บนแกนสเตเตอร์ฟลักซ์
$\hat{i}_{s\alpha}$	กระแสสเตเตอร์ประมาณบนแกนอัลฟ่า
$\hat{i}_{s\beta}$	กระแสสเตเตอร์ประมาณบนแกนเบต้า
$\varepsilon\bar{\mathbf{i}}$	เวกเตอร์ของความผิดพลาดกระแสสเตเตอร์บนแกนอ้างอิงคงที่
$\varepsilon\bar{\mathbf{i}}_s^e$	เวกเตอร์ของความผิดพลาดกระแสสเตเตอร์บนแกนอ้างอิงโรเตอร์ฟลักซ์
K_1	ค่าคงที่การปรับของตัวสังเกตการณ์สไลดิง
K_2	ค่าคงที่การปรับของตัวสังเกตการณ์สไลดิง
$K_{I\Psi}$, K_{ITe}	ค่าคงที่การอินทีเกรตของฟลักซ์และแรงบิดตามลำดับ
K_{PTe} , $K_{P\Psi}$	ค่าคงที่การปรับของแรงบิดและฟลักซ์ตามลำดับ

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

U_{DC}	แรงดันดีซีบัส
T_r	ค่าคงตัวทางเวลาของโรเตอร์
P.U	ต่อหนึ่งหน่วย
$\omega_r ref$	ความเร็วอ้างอิง
$e_{\psi s}$	ความผิดพลาดขนาดสเตเตอร์ฟลักซ์
e_{Te}	ความผิดพลาดของทอร์ก
$\lvert\Psi_s\rvert$	ขนาดของสเตเตอร์ฟลักซ์
Ψ_{sref}	ขนาดสเตเตอร์ฟลักซ์อ้างอิง
$\lvert\Psi_r\rvert$	ขนาดของโรเตอร์ฟลักซ์
Te_{ref}	แรงบิดอ้างอิง
u_d^*	แรงดันอ้างอิงอัลฟ่าบนแกนอ้างอิงฟลักซ์สเตเตอร์
u_q^*	แรงดันอ้างอิงเบต้าบนแกนอ้างอิงฟลักซ์สเตเตอร์
t	เวลา (วินาที)
rpm	รอบต่อนาที
ω_r	ความเร็วของโรเตอร์, rpm
ω_{ψ_s}	ความเร็วเชิงมุมของสเตเตอร์ฟลักซ์

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันมอเตอร์แบบเหนี่ยวนำ (Induction Motor) มีบทบาทสำคัญในอุตสาหกรรมอย่างกว้างขวาง เนื่องจากมอเตอร์ชนิดนี้มีราคาถูก สร้างได้ง่าย ประหยัดค่าซ่อมบำรุง เมื่อเทียบกับมอเตอร์กระแสตรง (DC Machine) อีกทั้งยังมีความน่าเชื่อถือในด้านความปลอดภัยในสภาวะแวดล้อมที่ต้องการความปลอดภัยสูง แต่มอเตอร์เหนี่ยวนำนั้น ก็มีข้อเสียอยู่เช่นกัน คือมีแบบจำลองทางคณิตศาสตร์ที่มีความไม่เป็นเชิงเส้น (Non-Linearity)สูง และยังมีตัวแปรค่อนข้างซับซ้อนตรวจวัดหาค่าที่ถูกต้องลำบาก แต่เนื่องจากปัจจุบันความสามารถในการคำนวณของตัวประมวลผลสัญญาณดิจิตอล (DSP) ซึ่งมีความรวดเร็วในการคำนวณและมีความละเอียดสูง จึงถูกนำมาใช้ควบคู่กับการควบคุมมอเตอร์เหนี่ยวนำ และทำให้ลดปัญหาดังกล่าวลงได้ ในส่วนการควบคุมการขับเคลื่อนมอเตอร์เหนี่ยวนำมีด้วยกันอยู่ 3 รูปแบบหลัก ๆ คือ การควบคุมแบบการเปลี่ยนอัตราส่วนแรงดันต่อความถี่คงที่ (Scalar Control :V/F) ซึ่งเหมาะกับงานที่ไม่ต้องการผลตอบสนองของแรงบิดที่รวดเร็วมากนัก เมื่อเทียบกับการควบคุมแบบปรับตามสนามแม่เหล็ก (Vector Control or Field Orientated Control, FOC) และแบบควบคุมแบบแรงบิดโดยตรง (Direct Torque Control, DTC) ซึ่ง DTC และ FOC จะให้การตอบสนองดีกว่า แบบ V/F การควบคุมแบบปรับตามสนามแม่เหล็ก(Vector Control or Field Orientated Control, FOC นั้นมีความละเอียดสูงตอบสนองได้ดี แต่มีส่วนประกอบของการคำนวณค่อนข้างสูง ส่วนแบบควบคุมแบบแรงบิดโดยตรง (Direct Torque Control, DTC) ให้ผลตอบสนองได้รวดเร็วกว่าเนื่องจากมีการคำนวณน้อยกว่าแบบเวกเตอร์แต่มีการกระเพื่อมของฟลักซ์และแรงบิดสูงกว่า

ทฤษฎีการควบคุมแรงบิดโดยตรง (DTC) ได้ถูกนำเสนอในปี คศ.1985 ซึ่งให้ผลตอบสนองของทอร์กและฟลักซ์ได้เร็ว และได้มีการพัฒนาประสิทธิภาพของการทำงานชนิดไร้ตัวตรวจจับความเร็วมารวมเข้าด้วยแล้ว ก็จะน่าสนใจมากยิ่งขึ้นซึ่งสังเกตุได้จากงานวิจัยในปัจจุบัน ซึ่งมีความพยายามที่จะพัฒนาประสิทธิภาพของการทำงานชนิดไร้ตัวตรวจจับความเร็ว เพื่อนำมาแทนเอ็นโคดเดอร์ในงานควบคุมมอเตอร์ที่ไม่ต้องการความละเอียดของความเร็วรอบจำนวนรอบหรือตำแหน่งมากนัก เพื่อลดค่าใช้จ่ายในส่วนของเอ็นโคดเดอร์ซึ่งมีราคาค่อนข้างสูงประกอบกับการควบคุมส่วนใหญ่ที่มีตัวประมวลผลสัญญาณดิจิตอลอยู่แล้วสามารถเพิ่มส่วนทดแทนขึ้นมาได้ดังจะเห็นจากงานวิจัยประเภท Sensorless ต่าง ๆ มีการใช้ตัวตัวประมวลผลสัญญาณดิจิตอลมาคำนวณหาจำนวนรอบและแรงบิดได้ซึ่งวิธีการคำนวณดังกล่าวสามารถหาได้ดังเช่น วิธีการประมาณความเร็วรอบสามารถแบ่งได้หลัก ๆ สองวิธี Open Loop Speed Estimator ประมาณจากความเร็วรอบของโรเตอร์และความเร็วสลิป และแบบ Closed- Loop Speed Estimator เช่นวิธี

Model Reference Adaptive System,MRAS[1] ประมาณโดยการนำฟลักซ์คล้องโรเตอร์ที่คำนวณได้จากการวัดแรงดันที่สเตเตอร์มาเทียบกับฟลักคล้องโรเตอร์ที่ถูกประมาณแล้วทำการปรับโดยตัวควบคุม PI หรือวิธี Reactive Power MRAS ซึ่งวิธีนี้มีข้อดีคือจะไม่มีความต้านทานสเตเตอร์ในสมการแบบจำลอง และไม่มีตัวอินทริเกรต ซึ่งที่กล่าวมาถ้าสามารถประมาณค่า ทอร์ก และฟลักซ์ได้ถูกต้องก็จะประมาณความเร็วรอบได้ถูกต้องด้วยระบบก็จะ คงทนต่อการเบี่ยงเบนค่าพารามิเตอร์ของมอเตอร์ วิธีการประมาณค่าทอร์กและฟลักซ์ สำหรับ การควบคุมแบบแรงบิดตรงนั้น มีด้วยกันหลายวิธี วิธีหนึ่งคือ Adaptive Flux Observer โดย [1] วิธีการนี้จะสนใจ ฟลักซ์บนสเตเตอร์ และ กระแสสเตเตอร์เป็นหลัก ต่อมาได้มีการนำ Adaptive Flux Observer มาร่วมกับตัวสังเกตการณ์แบบสไลดิ้ง[2] ซึ่งทำให้มีความแม่นยำและคงทนต่อการเบี่ยงเบนต่อค่าพารามิเตอร์ แต่ก็ยังสนใจที่แบบจำลองส่วนสเตเตอร์เป็นหลัก ต่อมาได้มีข้อสันนิษฐานว่าตัวสังเกตการณ์ยังมีการมองข้ามในส่วนของการกระจายตัวของฟลักซ์ในช่องว่างระหว่าง สเตเตอร์และโรเตอร์เพื่อให้ได้ค่าฟลักซ์และทอร์กที่ถูกต้องขึ้นต้องนำส่วนนี้มาพิจารณา จึงได้มีการนำการประมาณแบบ Full Order Observer เข้ามาใช้เพื่อให้ได้ค่าถูกต้องมากยิ่งขึ้น

ในงานวิจัยนี้ได้นำเสนอในส่วนที่จะพัฒนาในเรื่องตัวสังเกตการณ์เพื่อให้เหมาะสมกับการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงที่ให้ผลตอบสนองเร็ว จึงต้องการตัวสังเกตการณ์ที่มีทั้งความเร็วและความแม่นยำในการประมาณค่าเป็นหลัก โดยนำเสนอตัวสังเกตการณ์แบบสไลดิ้งร่วมกับตัวชดเชยแบบ PI โดยวิธีนี้พิจารณาทั้งฟลักซ์สเตเตอร์และฟลักซ์โรเตอร์ ซึ่งจะทำให้เกิดตัวแปรของเวลาใกล้เคียงกับตัวมอเตอร์จริงมากกว่า และนำตัวสังเกตการ์ณแบบสไลดิ้ง [3] เข้ามาใช้เพื่อให้ได้มาซึ่งความคงทนต่อการเบี่ยงเบนพารามิเตอร์และความแม่นยำ ซึ่งอีกทั้งมีการชดเชยด้วยตัวชดเชยแบบ PI [4] ในส่วนของแบบจำลองสเตเตอร์เพื่อชดเชยผลของเรื่องเวลาในแบบจำลองของตัวมอเตอร์ และอีกทั้งยังเพื่อแก้ปัญหา chattering ของ ตัวสไลดิงลง และยังมีไว้เพื่อลดความไวของตัวสไลดิงด้วย ซึ่งผลการจำลองระบบด้วยคอมพิวเตอร์เมื่อเทียบตัวควบคุมเดียวกันแต่เปลี่ยนตัวสังเกตุการณ์สามารถ ปรับให้สัญญาณความเร็วรอบและแรงบิดของระบบดีขึ้น และแสดงให้เห็นว่าตัวสังเกตการณ์ส่งผลต่อเสถียรภาพของระบบ

เนื้อหาในงานวิจัยนี้แบ่งออกเป็นเจ็ดบทบทแรกเป็นบทนำบทที่สองกล่าวถึงแบบจำลองมอเตอร์เหนี่ยวนำ ในบทที่สามกล่าว ระบบควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงและตัวสังเกตการณ์รวมไปถึงการประมาณความเร็วรอบ บทที่สี่เป็นผลการทดสอบแบบจำลองด้วยโปรแกรม Matlab Simulink บทที่ห้าเป็นเรื่องของอุปกรณ์ที่ใช้ในการทดลองและขั้นตอนการทดลอง ส่วนบทที่หกเป็นผลจากการทดลองทั้งหมดและบทสุดท้ายเป็นบทสรุปงานวิจัย และข้อเสนอแนะ

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาพฤติกรรมของมอเตอร์เหนี่ยวนำในการควบคุมมอเตอร์แบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI จากแบบจำลองระบบด้วยโปรแกรม MATLAB

1.2.2 เพื่อศึกษาทฤษฎีและพัฒนาการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงให้มีประสิทธิภาพเพื่อการปรับปรุงต่อไป

1.2.3 เพื่อให้เข้าใจถึงการออกแบบและศึกษาการควบคุมของมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็ว โดยการใช้ตัวประมวลผลสัญญาณดิจิตอล (DSP) เบอร์ TMS320F243 ทำหน้าที่ในการควบคุมระบบและประมวลผลทางคณิตศาสตร์ ซึ่งสามารถที่จะนำไปใช้ในการพัฒนาระบบควบคุมมอเตอร์แบบอื่นได้และในวิธีการควบคุมชนิดอื่นๆ

1.3 ขอบเขตของการวิจัย

1.3.1 ศึกษาและประยุกต์สร้างระบบควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ด้วยแบบจำลอง

1.3.2 ระบบควบคุมมอเตอร์เหนี่ยนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI สามารถใช้งานได้

1.3.3 ประยุกต์ใช้ระบบควบคุมแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI กับมอเตอร์เหนี่ยวนำโดยไม่ใช้ตัวตรวจจับความเร็วรอบ โดยเขียนโปรแกรมทำงานบนชุดควบคุม DSP TMS320F243 พร้อมทั้งวิเคราะห์ผลการทดลอง สรุปและเสนอแนะการปรับปรุง

1.4 ขั้นตอนและวิธีการวิจัย

1.4.1 ค้นคว้าหาข้อมูล และสำรวจงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ ศึกษาทฤษฎีของระบบควบคุมการขับเคลื่อนมอเตอร์เหนี่ยวนำแบบสไลดิงแรงบิดตรงและการประมาณหาค่าความเร็วรอบมอเตอร์ รวมถึงการจำลอง (Simulation) ด้วยโปรแกรม MATLAB

1.4.2 ศึกษาแบบจำลองของมอเตอร์เหนี่ยวนำและศึกษาทฤษฎีของการควบคุมแบบแรงบิดโดยตรง

1.4.3 ศึกษาการทำงานของตัวสังเกตการณ์แบบสไลดิงและการประมาณความเร็ว

1.4.4 ศึกษาการทำงานของตัวประมวลผลสัญญาณดิจิตอล (DSP) รุ่น TMS320F243

1.4.5 ศึกษาชุดขับเคลื่อนมอเตอร์ และวงจรเชื่อมต่อระหว่างชุดควบคุมกับชุดขับเคลื่อน

1.4.6 ทำการหมุนมอเตอร์ด้วยวิธี V/F เก็บกระแสและแรงดันอ้างอิงเพื่อทดสอบการประมาณความเร็ว

1.4.7 เขียนโปรแกรมควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI

1.4.8 ทำการทดสอบกับระบบจริงที่สภาวะการทำงานต่างๆ โดยใช้มอเตอร์เหนี่ยวนำขนาด 1.1 kW เก็บผลการทดลอง

1.4.9 เปรียบเทียบผลที่ได้จากระบบจริงกับการจำลองระบบด้วยคอมพิวเตอร์และวิเคราะห์ผลหาสาเหตุและแก้ไขข้อผิดพลาดต่างๆ

1.4.10 สรุปทำรายงานการวิจัย และนำเสนอผลที่ได้จากงานวิจัย

1.5 ประโยชน์ของการวิจัย

1.5.1 ทำให้เข้าใจถึงหลักการทำงานของระบบควบคุมแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิง ร่วมกับตัวชดเชยแบบ PI ที่ประยุกต์ใช้กับมอเตอร์เหนี่ยวนำ

1.5.2 ส่งเสริมให้เกิดการประยุกต์พัฒนาใช้งานตัวประมวลผลสัญญาณดิจิตอลในระบบการควบคุมการขับเคลื่อนมอเตอร์แบบต่างๆ

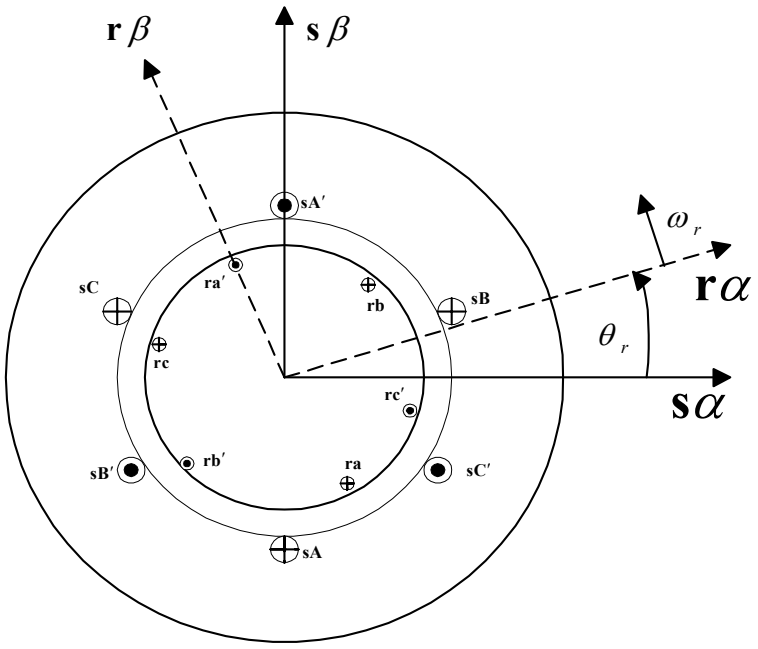
1.5.3 ประหยัดค่าใช้จ่ายของมอเตอร์ โดยไม่ต้องสิ้นเปลืองกับตัววัดตำแหน่งของมอเตอร์

บทที่ 2
ทฤษฎีเบื้องต้น แบบจำลองของมอเตอร์เหนี่ยวนำ

ในบทนี้จะกล่าวถึงทฤษฎีและหลักการของแบบจำลองทางคณิตศาสตร์ (Mathematical-Model) ของมอเตอร์เหนี่ยวนำและ การมอดูเลทแรงดันแบบสเปซเวกเตอร์เบื้องต้น

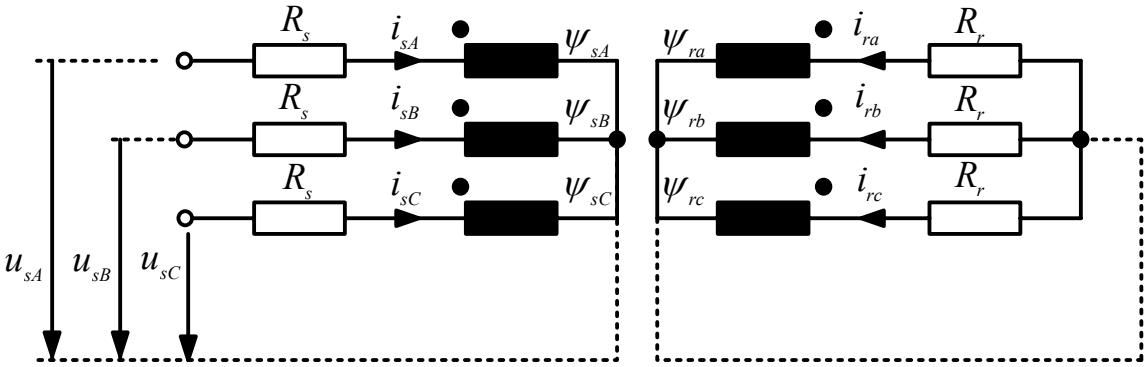
2.1 แบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำ

ถ้าพิจารณาที่แบบจำลองสมการของแรงดันทั้งด้านสเตเตอร์ (Stator) และฝั่งโรเตอร์ (Rotor) ซึ่งตามภาพที่ 2-1 และภาพที่ 2-2 เพื่อให้เป็นการง่ายในการพิจารณาสามารถแปลงระบบ 3 แกน ไปเป็นระบบ 2 แกนดังรูป



ภาพที่ 2-1 ภาพตัดขวางของมอเตอร์เหนี่ยวนำ

ภาพที่ 2-1 ได้แสดงขดลวดของสเตเตอร์และโรเตอร์ของเฟส เอ บี และ ซี ซึ่งวางห่างกัน 120 องศาทางไฟฟ้า θ_r คือ มุมของโรเตอร์ ที่วัดระหว่างขดลวดของสเตเตอร์เฟสเอและขดลวดของโรเตอร์เฟสเอ และ $\omega_r = \frac{d\theta_r}{dt}$ คือ ความเร็วของโรเตอร์



ภาพที่ 2-2 วงจรสมมูลของมอเตอร์เหนี่ยวนำ

สมการแรงดันด้านสเตเตอร์

$$R_s i_{sA} + \frac{d\psi_{sA}}{dt} = u_{sA}(t) \tag{2-1}$$

$$R_s i_{sB} + \frac{d\psi_{sB}}{dt} = u_{sB}(t) \tag{2-2}$$

$$R_s i_{sC} + \frac{d\psi_{sC}}{dt} = u_{sC}(t) \tag{2-3}$$

สมการแรงดันด้านโรเตอร์

$$R_r i_{ra} + \frac{d\psi_{ra}}{dt} = 0 \tag{2-4}$$

$$R_r i_{rb} + \frac{d\psi_{rb}}{dt} = 0 \tag{2-5}$$

$$R_r i_{rc} + \frac{d\psi_{rc}}{dt} = 0 \tag{2-6}$$

โดยที่ R_s คือ ความต้านทานแต่ละเฟสของขดลวดสเตเตอร์

R_r คือ ความต้านทานแต่ละเฟสของขดลวดโรเตอร์

u_{sA}, u_{sB}, u_{sC} คือ แรงดันแต่ละเฟสที่ขดสเตเตอร์

i_{sA}, i_{sB}, i_{sC} คือ กระแสแต่ละเฟสที่ขดสเตเตอร์

i_{ra}, i_{rb}, i_{rc} คือ กระแสแต่ละเฟสที่ขดโรเตอร์

$\psi_{sA}, \psi_{sB}, \psi_{sC}$ คือ ฟลักซ์คล้องที่ตัดขดลวดแต่ละเฟสทางด้านสเตเตอร์

$\psi_{ra}, \psi_{rb}, \psi_{rc}$ คือ ฟลักซ์คล้องที่ตัดขดลวดแต่ละเฟสทางด้านโรเตอร์

สำหรับแบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำส่วนใหญ่นิยมใช้ทฤษฎีของสเปซเวกเตอร์เพื่ออธิบายปริมาณต่าง ๆ ดังนี้

$$\bar{\mathbf{u}}_s(t) = \frac{2}{3}\left[u_{sA}(t) + u_{sB}(t)e^{j\gamma} + u_{sC}(t)e^{j2\gamma}\right] = u_{s\alpha} + ju_{s\beta} \tag{2-7}$$

$$\bar{\mathbf{i}}_s(t) = \frac{2}{3}\left[i_{sA}(t) + i_{sB}(t)e^{j\gamma} + i_{sC}(t)e^{j2\gamma}\right] = i_{s\alpha} + ji_{s\beta} \tag{2-8}$$

$$\bar{\mathbf{i}}_r(t) = \frac{2}{3}\left[i_{ra}(t) + i_{rb}(t)e^{j\gamma} + i_{rc}(t)e^{j2\gamma}\right] = i_{r\alpha} + ji_{r\beta} \tag{2-9}$$

$$\bar{\boldsymbol{\psi}}_s(t) = \frac{2}{3}\left[\psi_{sA}(t) + \psi_{sB}(t)e^{j\gamma} + \psi_{sC}(t)e^{j2\gamma}\right] = \psi_{s\alpha} + j\psi_{s\beta} \tag{2-10}$$

$$\bar{\boldsymbol{\psi}}_r(t) = \frac{2}{3}\left[\psi_{ra}(t) + \psi_{rb}(t)e^{j\gamma} + \psi_{rc}(t)e^{j2\gamma}\right] = \psi_{r\alpha} + j\psi_{r\beta} \tag{2-11}$$

โดยที่ $\bar{\mathbf{u}}_s(t)$ คือ สเปซเวกเตอร์ของแรงดันทางด้านสเตเตอร์

$\bar{\mathbf{i}}_s(t)$ คือ สเปซเวกเตอร์ของกระแสทางด้านสเตเตอร์

$\bar{\mathbf{i}}_r(t)$ คือ สเปซเวกเตอร์ของกระแสทางด้านโรเตอร์

$\bar{\boldsymbol{\psi}}_s(t)$ คือ สเปซเวกเตอร์ของฟลักซ์ทางด้านสเตเตอร์

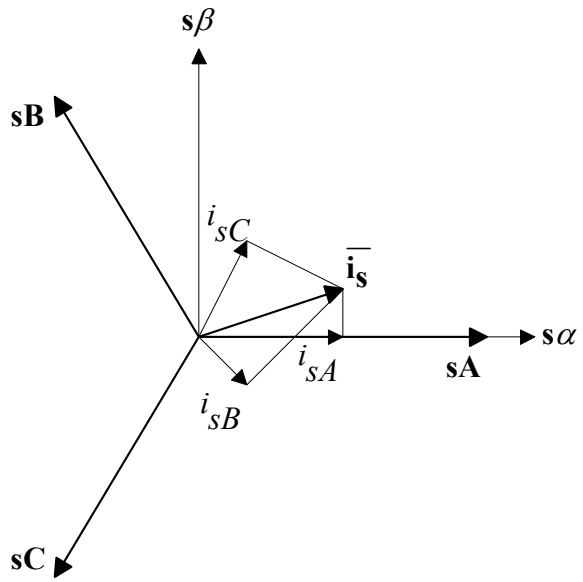
$\bar{\boldsymbol{\psi}}_r(t)$ คือ สเปซเวกเตอร์ของฟลักซ์ทางด้านโรเตอร์

γ มีค่าเท่ากับ $\frac{2\pi}{3}$

ที่จุดนิวตรอล จะได้สมการกระแสสเตเตอร์ดังนี้

$$i_{sA}(t) + i_{sB}(t) + i_{sC}(t) = 0 \tag{2-12}$$

ภาพที่ 2-3 ได้แสดงปริมาณของสเปซเวกเตอร์ของกระแสสเตเตอร์ ที่วางบนแกนของอัลฟ่าเบต้า $(\alpha\beta)$ ส่วนของปริมาณอื่นก็ทำนองเดียวกันที่เปลี่ยนรูปมาที่แกนของอัลฟ่าเบต้า $(\alpha\beta)$



ภาพที่ 2-3 สเปซเวกเตอร์ของกระแสสเตเตอร์บนแกนอ้างอิงคงที่

การเปลี่ยนรูปจากสามเฟสไปที่สองเฟสแสดงได้ตามสมการดังนี้ (ABC/$\alpha\beta$)

$$f_\alpha = f_A \tag{2-13}$$

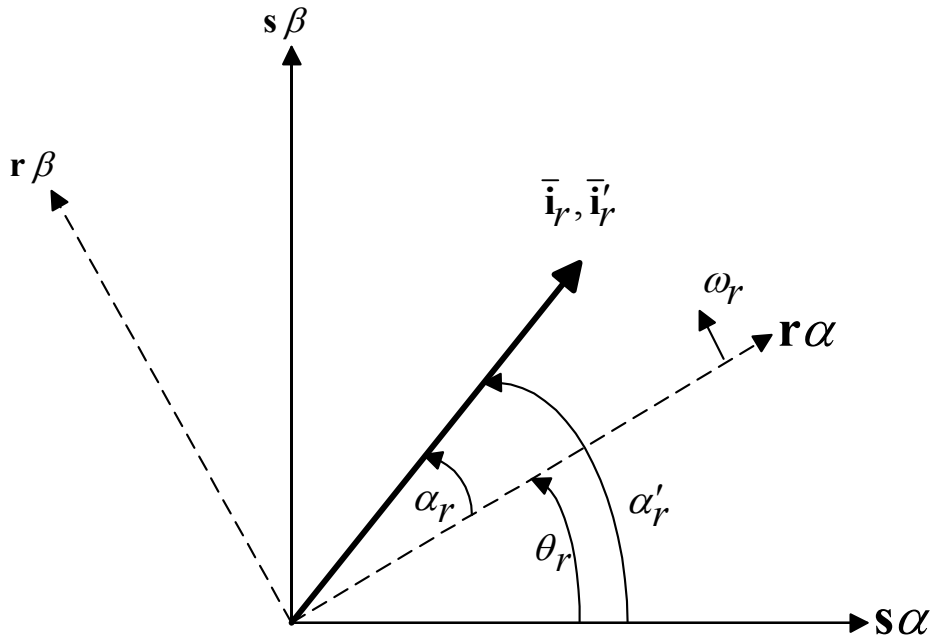
$$f_\beta = \frac{1}{\sqrt{3}}\left(f_A + 2f_B\right) \tag{2-14}$$

โดยที่ f สามารถแทนตัวแปรของกระแส แรงดัน หรือ ฟลักซ์อย่างใดอย่างหนึ่ง
การเปลี่ยนรูปจากสองเฟสไปที่สามเฟสแสดงได้ตามสมการดังนี้ ($\alpha\beta$/ABC)

$$f_A = f_\alpha \tag{2-15}$$

$$f_B = \frac{1}{2}\left(-f_\alpha + \sqrt{3}f_\beta\right) \tag{2-16}$$

$$f_C = -\frac{1}{2}\left(f_\alpha + \sqrt{3}f_\beta\right) \tag{2-17}$$



ภาพที่ 2-4 ความสัมพันธ์ระหว่างแกนอ้างอิงคงที่ (Stationary Reference Frame) และแกนอ้างอิงโรเตอร์ (Rotating Reference Frame)

จากภาพที่ 2-4 α_r คือ มุมระหว่างเวกเตอร์กระแสโรเตอร์กับแกนอ้างอิงโรเตอร์ θ_r คือ มุมระหว่างแกนอ้างอิงของสเตเตอร์และแกนอ้างอิงโรเตอร์ α'_r คือ มุมระหว่างแกนอ้างอิงของสเตเตอร์และเวกเตอร์กระแสโรเตอร์และ $\bar{\mathbf{i}}_r$ คือสเปซเวกเตอร์ของกระแสโรเตอร์ สมการของแรงดันและฟลักซ์ได้เขียนในรูปแบบของสเปซเวกเตอร์จะแสดงได้ดังนี้ [5, 6]

$$\bar{\mathbf{u}}_s = \bar{\mathbf{i}}_s R_S + \frac{d\bar{\boldsymbol{\psi}}_s}{dt} \tag{2-18}$$

$$\bar{\mathbf{u}}_r = \bar{\mathbf{i}}_r R_r + \frac{d\bar{\boldsymbol{\psi}}_r}{dt} \tag{2-19}$$

$$\bar{\boldsymbol{\psi}}_s = L_s \bar{\mathbf{i}}_s + L_m e^{j\theta_r} \bar{\mathbf{i}}_r \tag{2-20}$$

$$\bar{\boldsymbol{\psi}}_r = L_r \bar{\mathbf{i}}_r + L_m e^{-j\theta_r} \bar{\mathbf{i}}_s \tag{2-21}$$

โดยที่ L_s คือ ค่าความเหนี่ยวนำตัวเองด้านสเตเตอร์

L_r คือ ค่าความเหนี่ยวนำตัวเองด้านโรเตอร์

L_m คือ ค่าความเหนี่ยวนำของสนามแม่เหล็ก

θ_r คือ มุมของโรเตอร์ที่วัดระหว่างขดลวดของสเตเตอร์เฟสเอและขดลวดของโรเตอร์เฟสเอ

จากสมการ (2-18)-(2-21) เราได้ทำการเปลี่ยนรูปแบบของสมการไปที่แกนอ้างอิงหมุนใดๆ (Arbitrary Reference Frame) ซึ่งจะได้สมการดังนี้

$$\bar{\mathbf{u}}_s = R_S \bar{\mathbf{i}}_s + j\omega\bar{\boldsymbol{\psi}}_s + \frac{d\bar{\boldsymbol{\psi}}_s}{dt} \tag{2-22}$$

$$\bar{\mathbf{u}}_r = R_r \bar{\mathbf{i}}_r + j\left(\omega - \omega_r\right)\bar{\boldsymbol{\psi}}_r + \frac{d\bar{\boldsymbol{\psi}}_r}{dt} \tag{2-23}$$

$$\bar{\boldsymbol{\psi}}_s = L_s \bar{\mathbf{i}}_s + L_m \bar{\mathbf{i}}_r \tag{2-24}$$

$$\bar{\boldsymbol{\psi}}_r = L_s \bar{\mathbf{i}}_r + L_m \bar{\mathbf{i}}_s \tag{2-25}$$

โดยที่ ω คือ ความเร็วเชิงมุมของแกนอ้างอิงหมุนใดๆ

ω_r คือ ความเร็วเชิงมุมของโรเตอร์

จากสมการ (2-22)-(2-25) เมื่อเราให้ความเร็วเชิงมุมของแกนอ้างอิงที่หมุนใดๆ ไว้ที่สเตเตอร์ เราก็จะได้ความเร็วเชิงมุมของแกนอ้างอิงที่หมุนเท่ากับศูนย์ $\left(\omega = 0\right)$ ซึ่งก็จะได้สมการดังนี้

$$\bar{\mathbf{u}}_s = R_s \bar{\mathbf{i}}_s + \frac{d\bar{\boldsymbol{\psi}}_s}{dt} \tag{2-26}$$

$$\bar{\mathbf{u}}_r = R_r \bar{\mathbf{i}}_r - j\omega_r\bar{\boldsymbol{\psi}}_r + \frac{d\bar{\boldsymbol{\psi}}_r}{dt} \tag{2-27}$$

$$\bar{\boldsymbol{\psi}}_s = L_s \bar{\mathbf{i}}_s + L_m \bar{\mathbf{i}}_r \tag{2-28}$$

$$\bar{\boldsymbol{\psi}}_r = L_s \bar{\mathbf{i}}_r + L_m \bar{\mathbf{i}}_s \tag{2-29}$$

สมการการเคลื่อนที่เชิงมุมของมอเตอร์คือ

$$T_e - T_L = J_m \frac{d\omega_m}{dt} \tag{2-30}$$

โดยที่ J_m คือ โมเมนต์ความเฉื่อย (กิโลกรัม-เมตร^2)

ω_m คือ ความเร็วเชิงมุมทางกลของโรเตอร์ (เรเดียนต่อวินาที)

T_e คือ แรงบิดของมอเตอร์ (นิวตัน-เมตร)

T_L คือ แรงบิดของภาระ (นิวตัน-เมตร)

จากสมการ (2-26)-(2-29) เราจัดรูปใหม่ของสมการตามแกนอัลฟ่าเบต้า $(\alpha\beta)$ จะได้ดังนี้
สมการของแรงดันสเตเตอร์และโรเตอร์

$$u_{s\alpha} = R_s i_{s\alpha} + \frac{d\psi_{s\alpha}}{dt} \tag{2-31}$$

$$u_{s\beta} = R_s i_{s\beta} + \frac{d\psi_{s\beta}}{dt} \tag{2-32}$$

$$u_{r\alpha} = R_r i_{r\alpha} + \frac{d\psi_{r\alpha}}{dt} + \omega_r \psi_{r\beta} \tag{2-33}$$

$$u_{r\beta} = R_r i_{r\beta} + \frac{d\psi_{r\beta}}{dt} - \omega_r \psi_{r\alpha} \tag{2-34}$$

สมการของฟลักซ์สเตเตอร์และโรเตอร์

$$\psi_{s\alpha} = L_s i_{s\alpha} + L_m i_{r\alpha} \tag{2-35}$$

$$\psi_{s\beta} = L_s i_{s\beta} + L_m i_{r\beta} \tag{2-36}$$

$$\psi_{r\alpha} = L_r i_{r\alpha} + L_m i_{s\alpha} \tag{2-37}$$

$$\psi_{r\beta} = L_r i_{r\beta} + L_m i_{s\beta} \tag{2-38}$$

$$\sigma = 1 - \left(\frac{L_m^2}{L_s L_r} \right) \tag{2-39}$$

$$L_s = L_{ls} + L_m \tag{2-40}$$

$$L_r = L_{lr} + L_m \tag{2-41}$$

โดยที่ σ คือ ค่าตัวประกอบของการรั่วไหล (Leakage Factor)
L_{ls} คือ ค่าความเหนี่ยวนำของการรั่วไหลของขดลวดสเตเตอร์
L_{lr} คือ ค่าความเหนี่ยวนำของการรั่วไหลของขดลวดโรเตอร์
ความสัมพันธ์ของกระแสและฟลักซ์ในแกนอ้างอิงคงที่ในสมการ (2-37) และ (2-38)

$$i_{r\alpha} = \left(\frac{\psi_{s\alpha} - L_{ls} i_{s\alpha}}{L_m} \right) - i_{s\alpha} \tag{2-42}$$

$$i_{r\beta} = \left(\frac{\psi_{s\beta} - L_{ls} i_{s\beta}}{L_m} \right) - i_{s\beta} \tag{2-43}$$

แทนสมการ (2-42) และ (2-43) ในสมการของโรเตอร์ฟลักซ์ (2-37) และ (2-38) ตามลำดับจะได้ดังนี้

$$\psi_{r\alpha} = L_r\left(\frac{\psi_{s\alpha} - L_{ls}i_{s\alpha}}{L_m} - i_{s\alpha}\right) + L_m i_{s\alpha} = \frac{L_r}{L_m}\psi_{s\alpha} - \frac{L_r L_{ls}}{L_m}i_{s\alpha} - \left(L_r - L_m\right)i_{s\alpha} \qquad (2\text{-}44)$$

จาก $L_r - L_m = L_{lr}$ และ $\frac{L_r L_{ls}}{L_m} + L_{lr} = \sigma L_s \frac{L_r}{L_m}$ นำไปแทนในสมการ (2-44) จะได้ดังสมการ (2-45)

$$\psi_{r\alpha} = \frac{L_r}{L_m}\psi_{s\alpha} - \left(\frac{L_r L_{ls}}{L_m} + L_{lr}\right)i_{s\alpha} = \frac{L_r}{L_m}\psi_{s\alpha} - \frac{L_r}{L_m}\sigma L_s i_{s\alpha} \qquad (2\text{-}45)$$

ดังนั้นสมการของโรเตอร์ฟลักซ์ในรูปแบบจำลองแรงดัน (Voltage Model) ในตัวแปรของฟลักซ์และกระแสสเตเตอร์จะได้ดังนี้

$$\psi_{r\alpha} = \frac{L_r}{L_m}\left(\psi_{s\alpha} - \sigma L_s i_{s\alpha}\right) = \frac{L_r}{L_m}\psi_{s\alpha} - \frac{L_\sigma}{L_m}i_{s\alpha} \qquad (2\text{-}46)$$

โดยที่ $L_\sigma = \sigma L_s L_r = L_s L_r - L_m^2$ สมการที่(2-46) คือ ความสัมพันธ์ของโรเตอร์ฟลักซ์ สเตเตอร์ฟลักซ์และกระแสสเตเตอร์ที่อยู่บนแกนอ้างอิงคงที่ (Stationary Reference Frame) ทำนองเดียวกันก็จะหา $\psi_{r\beta}$ ได้โดยการแทนสมการเหมือนข้างบนซึ่งก็จะได้ดังนี้

$$\psi_{r\beta} = \frac{L_r}{L_m}\left(\psi_{s\beta} - \sigma L_s i_{s\beta}\right) = \frac{L_r}{L_m}\psi_{s\beta} - \frac{L_\sigma}{L_m}i_{s\beta} \qquad (2\text{-}47)$$

ดังนั้นมุมของโรเตอร์ฟลักซ์ที่หาได้จากรูปแบบจำลองแรงดันได้ดังนี้

$$\theta_{\psi r} = \tan^{-1}\left(\frac{\psi_{r\beta}}{\psi_{r\alpha}}\right) \qquad (2\text{-}48)$$

สมการของแกนอ้างอิงโรเตอร์ฟลักซ์ของโรเตอร์ฟลักซ์ (Rotor Flux Reference Frame ,dq) แทนค่าจากสมการ (2-23) โดยให้ $\bar{\mathbf{u}}_r = 0$ และ $\omega = \omega_{\psi r} = \omega_e$ จะได้ดังนี้

$$0 = R_r\,\bar{\mathbf{i}}_r^{\,e} + j\left(\omega_e - \omega_r\right)\bar{\boldsymbol{\psi}}_r^{\,e} + \frac{d\bar{\boldsymbol{\psi}}_r^{\,e}}{dt} \qquad (2\text{-}49)$$

ดัชนี e คือ แกนอ้างอิงโรเตอร์ฟลักซ์ (Rotor Flux Reference Frame) และรูปแบบจำลองกระแส

ให้

$$\bar{\boldsymbol{\psi}}_r^e = \psi_{rd} + j\psi_{rq} = \psi_{r\alpha}^e + j\psi_{r\beta}^e$$
$$\bar{\mathbf{i}}_r^e = i_{rd} + ji_{rq} = i_{r\alpha}^e + ji_{r\beta}^e$$
$$\bar{\mathbf{i}}_s^e = i_{sd} + ji_{sq} = i_{s\alpha}^e + ji_{s\beta}^e$$
$$\bar{\boldsymbol{\psi}}_s^e = \psi_{sd} + j\psi_{sq} = \psi_{s\alpha}^e + j\psi_{s\beta}^e$$

นำความสัมพันธ์ข้างบนลงไปแทนลงในสมการ (2-49) จะได้ดังนี้

$$\frac{d\psi_{r\alpha}^e}{dt} = \frac{d\psi_{rd}}{dt} = \frac{L_m}{T_r} i_{s\alpha}^e - \frac{1}{T_r}\psi_{r\alpha}^e + \left(\omega_e - \omega_r\right)\psi_{r\beta}^e \tag{2-50}$$

$$\frac{d\psi_{r\beta}^e}{dt} = \frac{d\psi_{rq}}{dt} = \frac{L_m}{T_r} i_{s\beta}^e - \frac{1}{T_r}\psi_{r\beta}^e - \left(\omega_e - \omega_r\right)\psi_{r\alpha}^e \tag{2-51}$$

ให้ $T_r = \dfrac{L_r}{R_r}$,สมการ (2-50) และ (2-51) นำมาเขียนในรูปแบบสเปซเวกเตอร์ได้เป็นดังนี้

$$\frac{d\bar{\boldsymbol{\psi}}_r^e}{dt} = \frac{L_m}{T_r}\bar{\mathbf{i}}_s^e + \left(-\frac{1}{T_r} + j\left(\omega_e - \omega_r\right)\right)\bar{\boldsymbol{\psi}}_r^e \tag{2-52}$$

และจากความสัมพันธ์ของสมการ (2-46) และ (2-47) นำมาแปลงให้อยู่บนแกนอ้างอิงโรเตอร์ฟลักซ์จะได้

$$\bar{\boldsymbol{\psi}}_r^e = \frac{L_r}{L_m}\bar{\boldsymbol{\psi}}_s^e - \frac{L_\sigma}{L_m}\bar{\mathbf{i}}_s^e \tag{2-53}$$

ฟลักซ์คล้องโรเตอร์ในสมการ (2-50) และ (2-51) ถูกเปลี่ยนรูปไปอยู่ในแกนอ้างอิงคงที่ (Stationary Reference Frame) โดยการใช้การแปลงรูปจากดีคิวไปที่อัลฟ่าเบต้า (Inverse Park-transformation : dq to $\alpha\beta$, $e^{j\theta_{\psi r}}$)ดังนี้

$$\begin{bmatrix} f_{r\alpha} \\ f_{r\beta} \end{bmatrix} = \begin{bmatrix} \cos\theta_{\psi r} & -\sin\theta_{\psi r} \\ \sin\theta_{\psi r} & \cos\theta_{\psi r} \end{bmatrix} \begin{bmatrix} f_{rd} \\ f_{rq} \end{bmatrix} \tag{2-54}$$

การแปลงรูปจากอัลฟ่าเบต้าไปที่ดีคิว (Park-transformation : $\alpha\beta$ to dq, $e^{-j\theta_{\psi r}}$)

$$\begin{bmatrix} f_{rd} \\ f_{rq} \end{bmatrix} = \begin{bmatrix} \cos\theta_{\psi r} & \sin\theta_{\psi r} \\ -\sin\theta_{\psi r} & \cos\theta_{\psi r} \end{bmatrix} \begin{bmatrix} f_{r\alpha} \\ f_{r\beta} \end{bmatrix} \tag{2-55}$$

ดังนั้นจากสมการ (2-50) จะได้ว่า $\psi_{rq} = 0$ แทนในสมการ (2-51) จะได้ดังนี้

$$\psi_{r\alpha}^{s} = \psi_{rd}\cos\left(\theta_{\psi r}\right) - \psi_{rq}\sin\left(\theta_{\psi r}\right) = \psi_{rd}\cos\left(\theta_{\psi r}\right) \tag{2-56}$$

$$\psi_{r\beta}^{s} = \psi_{rd}\sin\left(\theta_{\psi r}\right) + \psi_{rq}\cos\left(\theta_{\psi r}\right) = \psi_{rd}\sin\left(\theta_{\psi r}\right) \tag{2-57}$$

โดยที่ $\theta_{\psi r}$ คือ มุมของโรเตอร์ฟลักซ์

สมการแรงบิดของมอเตอร์

$$T_e = \frac{3}{2}P\bar{\boldsymbol{\psi}}_s \times \bar{\mathbf{i}}_s = \frac{3}{2}P\,\mathrm{Im}\left(\overline{\boldsymbol{\psi}_\mathbf{s}^*}\bar{\mathbf{i}}_\mathbf{s}\right) \tag{2-58}$$

$$T_e = \frac{3}{2}P\left(\psi_{s\alpha}i_{s\beta} - \psi_{s\beta}i_{s\alpha}\right) \tag{2-59}$$

จากการแทนสมการฟลักซ์สเตเตอร์และฟลักซ์โรเตอร์ในสมการ (2-59) สามารถจัดรูปสมการแรงบิดของมอเตอร์ใหม่ได้ดังนี้

$$T_e = \frac{3}{2}P\frac{L_m}{\sigma L_s L_r}\left(\psi_{s\alpha}\psi_{r\beta} - \psi_{s\beta}\psi_{r\alpha}\right) \tag{2-60}$$

$$T_e = \frac{3}{2}P\frac{L_m}{L_r}\left(\psi_{r\alpha}i_{s\beta} - \psi_{r\beta}i_{s\alpha}\right) \tag{2-62}$$

โดยที่ P คือ จำนวนคู่ของขั้วแม่เหล็ก (Pole Pairs)

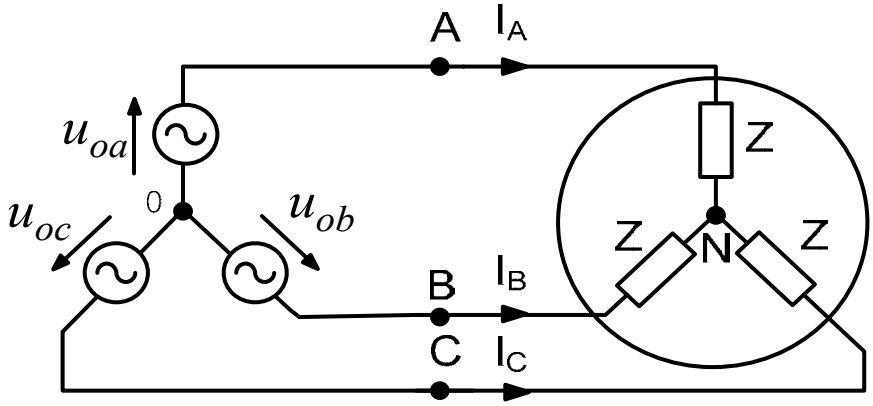
ในรูปแบบของมอเตอร์เหนี่ยวนำแบบทรงกระรอก ก็จะได้ว่า $\overline{\mathbf{u}_\mathbf{r}} = 0$ ก็จะได้ว่า $u_{r\alpha} = u_{r\beta} = 0$

2.2 อินเวอร์เตอร์และสเปซเวกเตอร์แรงดัน (Inverter and Voltage Space Vector) [7, 8]

สเปซเวกเตอร์แรงดันใช้สำหรับกำเนิดแรงดันที่จ่ายให้กับขดลวดสเตเตอร์แต่ละเฟสทำได้โดยการสวิตช์อินเวอร์เตอร์ 3 เฟส ซึ่งอุปกรณ์ที่ทำหน้าที่เป็นสวิตช์เป็นอิเล็กทรอนิกส์กำลังรูปแบบการสวิตช์อินเวอร์เตอร์ขึ้นอยู่กับค่าที่คำนวณเพื่อได้ค่าที่เหมาะสม

2.2.1 ระบบแหล่งจ่ายแรงดัน 3 เฟสทั่วไป

วิธีกำเนิดสนามแม่เหล็กหมุนโดยใช้แหล่งจ่ายแรงดันสามเฟสที่มีลักษณะเป็นแหล่งจ่ายไฟอิสระสามชุด โดยมีมุมต่างเฟส 120 องศาจ่ายให้กับมอเตอร์แสดงดังภาพที่ 2-5



ภาพที่ 2-5 แหล่งจ่ายไฟอิสระ 3 ชุด

ในมาตรฐานระบบไฟ 3 เฟส แรงดันไซน์ถูกจ่ายให้กับมอเตอร์กำเนิดกระแสที่เป็นไซน์มีแรงดันเฟสแสดงได้ดังนี้

$$u_{oa}(t) = \sqrt{2}V\cos(\omega t) \tag{2-63}$$

$$u_{ob}(t) = \sqrt{2}V\cos\left(\omega t - \frac{2\pi}{3}\right) \tag{2-64}$$

$$u_{oc}(t) = \sqrt{2}V\cos\left(\omega t - \frac{4\pi}{3}\right) \tag{2-65}$$

สามารถคำนวณแรงดันเฟสเทียบกับนิวทรัล (u_{an}, u_{bn}, u_{cn}) จากแหล่งจ่ายแรงดันที่จ่ายให้มอเตอร์ (u_{oa}, u_{ob}, u_{oc}) โดยกำหนดให้เป็นระบบ 3 เฟสแบบสมดุล ได้ความสัมพันธ์ดังนี้

$$u_{on} = u_{oa} + Z * I_1 \tag{2-66}$$

$$u_{on} = u_{ob} + Z * I_2 \tag{2-67}$$

$$u_{on} = u_{oc} + Z * I_3 \tag{2-68}$$

ดังนั้น

$$3u_{on} = u_{oa} + u_{ob} + u_{oc} + Z(I_1 + I_2 + I_3) \tag{2-69}$$

ที่

$$(I_1 + I_2 + I_3) = 0 \tag{2-70}$$

จากสมการที่ (2-69) และ (2-70) แรงดันเฟส A $\left(u_{an}\right)$ มีค่า

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

$$u_{an} = \frac{-2}{3}u_{oa} + \frac{1}{3}u_{ob} + \frac{1}{3}u_{oc} = \frac{2}{3}u_{ao} - \frac{1}{3}u_{bo} - \frac{1}{3}u_{co}$$

ดั้งนั้นในระบบแหล่งจ่ายแรงดัน 3 เฟสแบบสมดุลแรงดันเฟสมีค่า

$$u_{an} = \left(\frac{1}{3}\right)\left(2 * u_{ao} - u_{bo} - u_{co}\right) \tag{2-71}$$

$$u_{bn} = \left(\frac{1}{3}\right)\left(2 * u_{bo} - u_{ao} - u_{co}\right) \tag{2-72}$$

$$u_{cn} = \left(\frac{1}{3}\right)\left(2 * u_{co} - u_{ao} - u_{bo}\right) \tag{2-73}$$

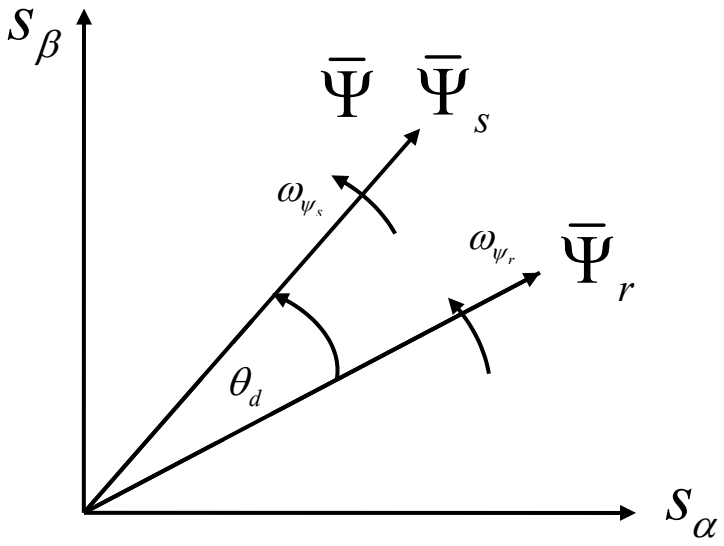
ในส่วนนี้จะกล่าวถึงการควบคุมแรงบิดซึ่งใช้การควบคุมของสเตเตอร์ฟลักซ์ ซึ่งการควบคุมแรงบิดสามารถใช้วิธีการควบคุมของโรเตอร์ฟลักซ์ หรือแมกเนไตน์ซิงฟลักซ์ (Magnetizing Flux) ก็ได้ โดยทั่วไปมอเตอร์เหนี่ยวนำ 3 เฟส แรงบิดชั่วขณะหาได้จากผลของการครอสเวกเตอร์ (Cross-vectorial Product) สเปซเวกเตอร์ของสเตเตอร์ฟลักซ์และสเปซเวกเตอร์ของกระแสสเตเตอร์ โดยที่ทั้งคู่ถูกพิจารณาในแกนอ้างอิงคงที่ (Stationary Reference Frame)

$$T_e = \frac{3}{2}P\bar{\boldsymbol{\psi}}_s \times \bar{\mathbf{i}}_s \tag{2-74}$$

โดยที่ P คือ จำนวนคู่ของขั้วแม่เหล็ก (Pole Pairs)

$\bar{\boldsymbol{\psi}}_s$ คือ สเปซเวกเตอร์ของสเตเตอร์ฟลักซ์

$\bar{\mathbf{i}}_s$ คือ สเปซเวกเตอร์ของกระแสสเตเตอร์



ภาพที่ 2-6 สเปซเวกเตอร์ของสเตเตอร์ฟลักซ์และสเปซเวกเตอร์ของโรเตอร์ฟลักซ์

และแรงบิดของมอเตอร์เหนี่ยวนำสามารถพิจารณาได้อีกรูปแบบคือจากครอสโปรดักของสเปซเวกเตอร์ของสเตเตอร์ฟลักซ์และสเปซเวกเตอร์ของโรเตอร์ฟลักซ์ θ_d คือมุมของสเปซเวกเตอร์ของสเตเตอร์ฟลักซ์เทียบกับสเปซเวกเตอร์ของโรเตอร์ฟลักซ์ จากภาพที่ 2-13 จะได้ว่า

$$T_e = \frac{3pL_m}{2L_\sigma}\bar{\Psi}_s \times \bar{\Psi}_r = \frac{3pL_m}{2L_\sigma}\left|\bar{\Psi}_s\right|\left|\bar{\Psi}_r\right|\sin\theta_d \tag{2-75}$$

$$\theta_d = \frac{d}{dt}\theta_d = \omega_{\psi_s} - \omega_{\psi_r} \tag{2-76}$$

โดยที่ ω_{ψ_s} และ ω_{ψ_r} คือ ความเร็วเชิงมุมของสเปซเวกเตอร์ของสเตเตอร์ฟลักซ์ และสเปซเวกเตอร์ของโรเตอร์ฟลักซ์ตามลำดับ

2.3 สรุป

ในงานวิจัยนี้จะใช้แบบจำลองบนแกนอ้างอิงคงที่ (Stationary Reference Frame) เป็นหลักทั้งในส่วนของแบบจำลองมอเตอร์และตัวสังเกตการณ์ แต่จะถูกทำให้อยู่ในรูปต่อหนึ่งหน่วยเพื่อให้ง่ายต่อการนำไปใช้งานในระบบควบคุมแบบดิจิตอลสำหรับตัวประมวลผล

บทที่ 3
การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงและการประมาณความเร็วรอบ

ในบทนี้จะกล่าวถึงหลักการและวิธีการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงและการประมาณความเร็วรอบของมอเตอร์เหนี่ยวนำ โดยวิธีการประมาณความเร็วรอบแบบลูปเปิด (Open-loop Speed Estimator)

3.1 หลักการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรง

การกำเนิดของแรงบิดในมอเตอร์เหนี่ยวนำสามารถเป็นสมการได้ดังนี้

$$T_e = \frac{3pL_m}{2L^2_\sigma}\psi_s\psi_r \sin(\theta) \tag{3-1}$$

$$\frac{dTe}{dt} = \frac{3pL_m}{2L^2_\sigma}\psi_s\psi_r\omega_d \cos(\theta) \tag{3-2}$$

$$\omega_d = \frac{d\theta}{dt} \tag{3-3}$$

Te คือ แรงบิดชั่วขณะที่เกิดขึ้น

θ คือมุมของของฟลักซ์คล้องสเตเตอร์ มื่อเทียบกับฟลักซ์คล้องโรเตอร์

ψ_s คือขนาดของฟลักซ์คล้องสเตเตอร์

ψ_r คือขนาดของฟลักซ์คล้องโรเตอร์

และ $L^2_\sigma = L_sL_r - L^2_m$ พิจารณาสมการ ถ้าเราทำให้ ขนาดของฟลักซ์ทางสเตเตอร์และขนาดฟลักซ์ทางโรเตอร์มีขนาดคงที่ดังนั้น เราสามารถควบคุมอัตราเร่งของ แรงบิดที่เกิดขึ้นได้โดยทำการ หมุนฟลักซ์สเตเตอร์ให้เร็วที่สุด เมื่อเทียบกับ ฟลักซ์โรเตอร์

การควบคุมแรงบิดสามารถใช้วิธีการควบคุมสเตเตอร์ฟลักซ์ ทางโรเตอร์ฟลักซ์ หรือแมกเนไตน์ซิงฟลักซ์ (Magnetizing Flux) ก็ได้ ในวิทยานิพนธ์นี้จะทำการควบคุมแรงบิดซึ่งใช้การควบคุมของสเตเตอร์ฟลักซ์ พิจารณาสมการที่ (2-22) ซึ่งเป็นสมการของสเปซเวกเตอร์ของสเตเตอร์ฟลักซ์ที่แกนอ้างอิงหมุนใดๆ (Arbitrary Reference Frame)

$$\bar{\mathbf{u}}_s = R_S\bar{\mathbf{i}}_s + j\omega\bar{\boldsymbol{\psi}}_s + \frac{d\bar{\boldsymbol{\psi}}_s}{dt} \tag{3-4}$$

การควบคุม ฟลักซ์สเตเตอร์ และ แรงบิดสามารถทำได้โดย พิจารณาจากสมการของมอเตอร์เหนี่ยวนำบนแกนที่วางอยู่ในแนวเดียวกับ ฟลักซ์สเตเตอร์ (Stator Flux Reference Frame)

ถ้านำสมการ (2-22) มาแปลงให้อยู่บนแกนอ้างอิงสเตเตอร์ฟลักซ์(Stator Flux Reference Frame) โดยที่ $\omega = \omega_{\psi s} = \dfrac{d\theta_{\psi s}}{dt}$ จะได้ดังนี้

$$\bar{\mathbf{u}}_s^{\psi_s} = R_S \bar{\mathbf{i}}_s^{\psi_s} + j\omega_{\psi_s} \bar{\boldsymbol{\psi}}_s^{\psi_s} + \frac{d\bar{\boldsymbol{\psi}}_s^{\psi_s}}{dt} \tag{3-5}$$

ให้

$$\bar{\boldsymbol{\psi}}_s^{\psi_s} = \psi_{sd} + j\psi_{sq} = \psi_{s\alpha}^{s} + j\psi_{s\beta}^{s}$$
$$\bar{\mathbf{i}}_s^{\psi_s} = i_{sd} + ji_{sq} = i_{s\alpha}^{s} + ji_{s\beta}^{s}$$
$$\bar{\mathbf{u}}^{\psi_s} = u_{sd} + ju_{sq} = u_{s\alpha}^{s} + ju_{s\beta}^{s}$$

ดัชนี ψs หมายถึง สเปซเวกเตอร์เทียบกับแกนอ้างอิงสเตเตอร์ฟลักซ์(Stator Flux Reference Frame)

ดัชนี s หมายถึง ปริมาณที่อยู่บนแกนอ้างอิงสเตเตอร์ฟลักซ์(Stator Flux Reference Frame)

เนื่องจากพิจารณาที่แกนอ้างอิงสเตเตอร์ฟลักซ์ดังนั้น $\psi_{sd} = |\Psi_s|, \psi_{sq} = 0$

ข้อดีของการพิจารณาตัวแปรควบคุมของมอเตอร์เหนี่ยวนำจากแกนอ้างอิงของฟลักซ์สเตเตอร์ คือมีความซับซ้อนน้อยกว่าทางโรเตอร์อีกทั้งยังมีเพียงความต้านทานทางสเตเตอร์เพียงตัวเดียวที่เป็น พารามิเตอร์ จากสมการข้างต้นเราจะแยกเป็น 2 องค์ประกอบในแนว คือในแนวตั้งฉากและขนานกับแกนอ้างอิงสเตเตอร์ฟลักซ์

$$u_{sd} = R_s i_{sd} + \frac{d}{dt}\psi_{sd} \tag{3-6}$$
$$u_{sq} = R_s i_{sq} + \omega_{\psi_s}\psi_{sd} \tag{3-7}$$

เนื่องจาก $\psi_{sd} = |\Psi_s|, \psi_{sq} = 0$ ดังนั้นสมการแรงบิดของมอเตอร์เหนี่ยวนำบนแกนอ้างอิงสเตเตอร์ฟลักซ์ คือ

$$Te = \frac{3p}{2}(\psi_{sd} i_{sq}) \tag{3-8}$$

ถ้านำสมการ (3-5) และ (3-6) มาแปลงในรูปของ i_{sq} และ $\psi_{sd} = |\Psi_s|$ จะได้ดังนี้

$$|\Psi_s| = \int (u_{sd} - R_s i_{sd})\, dt \tag{3-9}$$

$$i_{sq} = (u_{sq} - \omega_{\psi_s} |\Psi_s|) / R_s \tag{3-10}$$

จากสมการ (3-7) พบว่าว่าแรงบิดสามารถควบคุมได้ด้วย กระแสสเตเตอร์ในแกนคิวของแกนอ้างอิงของฟลักซ์สเตเตอร์ ซึ่งนั่นคือการควบคุมด้วย แรงดันสเตเตอร์บนแกนอ้างอิงของฟลักซ์สเตเตอร์ ส่วนขนาดของฟลักซ์ในแกนดีจะถูกควบคุมผ่านแรงดันบนแกนดีของแกนอ้างอิงของฟลักซ์สเตเตอร์ เช่นเดียวกัน

3.2 ระบบควบคุมโครงสร้างผันแปร (Variable-Structure Control , VSC)

ระบบควบคุมแบบโครงสร้างผันแปรจะแตกต่างกับระบบควบคุมชนิดอื่นโดยที่โครงสร้างของระบบจะมีการผันแปรไปในระหว่างการควบคุม ซึ่งการผันแปรของโครงสร้างนี้จะกระทำเพื่อให้ตัวแปรสเตทของระบบควบคุมวิ่งไปตามเส้นสวิทชิงฟังก์ชันที่กำหนดบนระนาบเฟส (Phase Plane) การกระทำเช่นนี้จะเรียกว่าการทำงานแบบสไลด์ดิงโหมด เมื่อเกิดสไลด์ดิงโหมดระบบควบคุมจะมีความคงทนต่อการเบี่ยงเบนค่าพารามิเตอร์ของกระบวนการและสิ่งรบกวนภายนอก โดยที่สมการของการสวิทช์จะต้องเลือกให้สอดคล้องกับกระบวนการที่ต้องการควบคุมเพื่อให้การควบคุมกระบวนการนั้นมีเสถียรภาพ ซึ่งในวิทยานิพนธ์นี้ ได้ทำการเลือกสวิทชิงฟังก์ชันของฟลักซ์ และแรงบิดเป็นดังนี้

$$S_{\Psi s} = e_{\Psi s} + C_{\Psi s} \frac{d}{dt} e_{\Psi s} \tag{3-11}$$

$$S_{Te} = e_{Te} + C_{Te} \frac{d}{dt} e_{Te} \tag{3-12}$$

โดยที่ $e_{\Psi s} = \Psi_{sref} - |\Psi_s|, e_{Te} = Te_{ref} - Te$

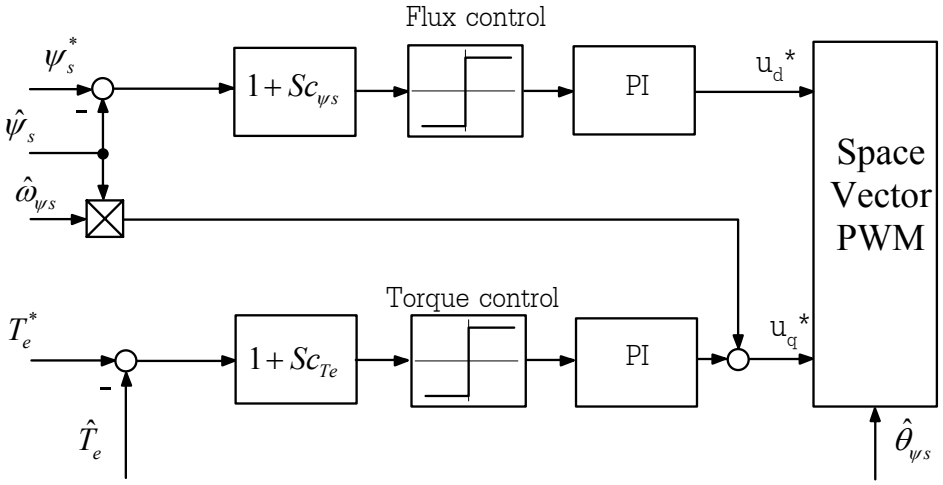
จากการกำหนดสมการของเส้นสไลดิงดังกล่าวจะทำให้ผลเฉลยของค่าความผิดพลาดเมื่อระบบเข้าสู่สไลดิงโหมดเป็นเอ็กสโปเนนเชียล $C_{\Psi s}$ และ C_{Te} คือค่าคงที่ถ้ามีค่าน้อยจะทำให้ผลตอบสนองของ แรงบิดและฟลักซ์ เร็ว แต่จะลดเสถียรภาพและความกว้างของย่านการทำงาน

เงื่อนไขสำหรับสเถียรภาพของระบบคือ $K_{P\Psi} > |Rsi_{sd}|, K_{Te} \frac{3p}{2Rs} |\Psi_s| > 1$

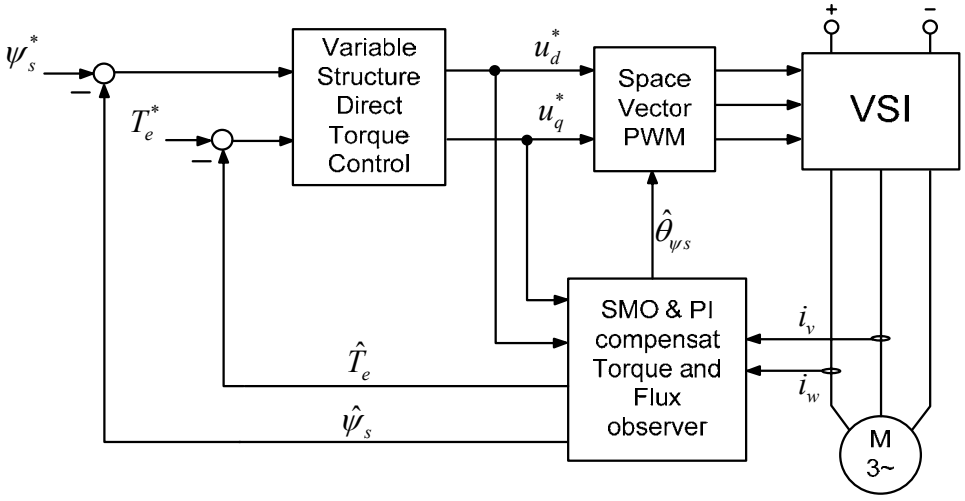
$$u_d^* = (K_{P\Psi} + \frac{K_{I\Psi}}{s}) \operatorname{sgn}(S_{\Psi s}) \tag{3-13}$$

$$u_q^* = (K_{PTe} + \frac{K_{ITe}}{s}) \operatorname{sgn}(S_{Te}) + \hat{\omega}_{\Psi s} |\Psi_s| \tag{3-14}$$

สมการ (3-12) และ (3-13) เป็นสมการการควบคุมแบบสไลดิงแรงบิดตรงที่วิทยานิพนธ์นี้เลือกใช้ คือ ชนิดอัตราขยายสวิทซ์ป้อนกลับแบบปรับตัว เนื่องจากต้องการลดผลของการสวิทชิ่งที่เกิดขึ้นเมื่อระบบเข้าสู่สภาวะคงตัว โดยที่เทอมสุดท้ายของสมการแรงดันที่ตั้งฉากกับแกนของ ฟลักซ์สเตเตอร์ดังสมการ (3-13) คือ การชดเชย แรงดันไฟฟ้าป้อนกลับที่เกิดจากความเร็วเชิงมุมของ ฟลักซ์คล้องสเตเตอร์ ทำให้สามารถลดขนาดของอัตราขยายสไลดิงของเทอมตัวควบคุม พีไอที่อยู่ด้านหน้าของเทอมแรงดันไฟฟ้าป้อนกลับซึ่งจะส่งผลให้การกระเพื่อมของผลตอบสนองของแรงบิดลดลงได้ส่วนปริมาณที่ใช้เป็นปริมาณที่ต้องถูกคำนวณอยู่แล้วจากการหาขนาดของเวกเตอร์ของฟลักซ์คล้องสเตเตอร์และมุม $\theta_{\psi s}$



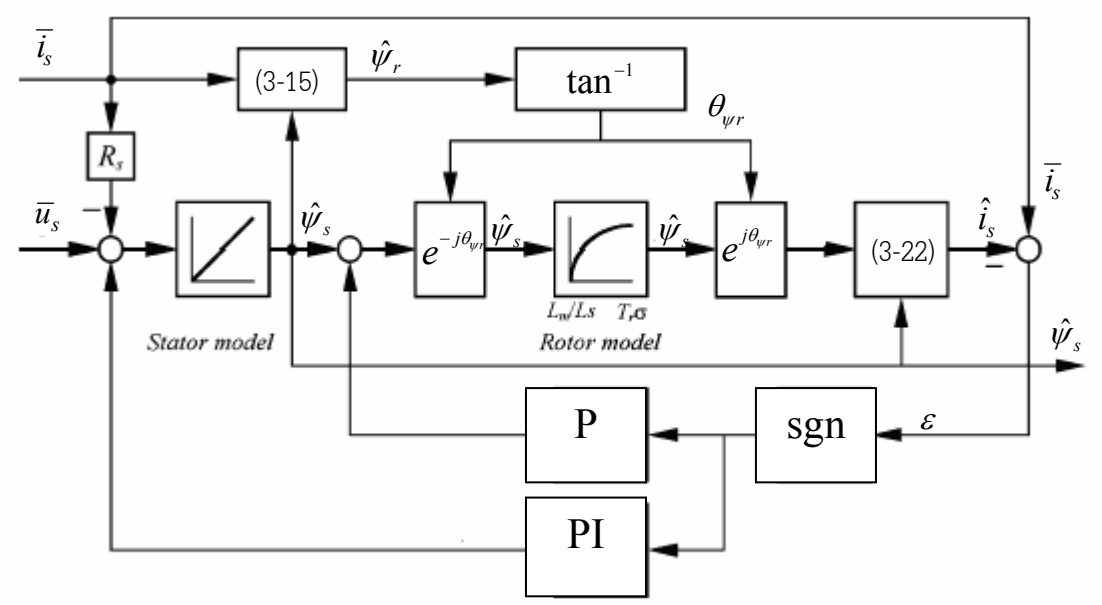
ภาพที่ 3-1 โครงสร้างภายในของการควบคุมฟลักซ์และแรงบิดแบบแรงบิดโดยตรง



ภาพที่ 3-2 การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบ โดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI

3.3 ตัวสังเกตการณ์แบบสไลดิงและตัวชดเชยแบบ PI (7)

เนื่องจากเป็นการทำ Sensorless ดังนั้นต้องหลีกเลี่ยงการนำตัวแปร ความเร็วของมอเตอร์มาใช้ในสมการพลวัตรของตัวสังเกตการณ์ เนื่องจาก ค่าความผิดพลาด ของความเร็วรอบจริงกับความเร็วรอบที่ได้จากการประมาณ จะส่งผลไปถึงปริมาณที่ได้ จากตัวสังเกตการณ์ ผิดเพี้ยนไปด้วยในส่วนแรกของตัวสังเกตการณ์จะเป็นการสร้างตัวแปรสถานะสเตเตอร์ฟลักซ์จากแบบจำลองของมอเตอร์เหนี่ยวนำซึ่งใช้ เทอมปรับค่าความผิดพลาดเป็นชนิดสไลดิงเนื่องจากเป็นตัวสังเกตการณ์ที่ออกแบบง่ายไม่ต้องใช้เวลาในการปรับแต่งมากอีกทั้งยังมีความน่าเชื่อถือสูงอีกด้วย ส่วนของแบบจำลองมอเตอร์เหนี่ยวนำที่ใช้ในการทำวิจัยผู้วิจัยได้เลือกแบบจำลองพลวัตรของฟลักซ์คล้องโรเตอร์และฟลักซ์คล้องสเตเตอร์มาใช้เนื่องจากเป็นปริมาณที่สามารถนำไปประมาณความเร็วของมอเตอร์เหนี่ยวนำในแบบลูปเปิดได้ทันที



ภาพที่ 3-3 โครงสร้างของตัวสังเกตการณ์แบบสไลดิ้งและตัวชดเชยแบบ PI (7)

พิจารณาสมการของสเปซเวกเตอร์ของสเตเตอร์ฟลักซ์ในแกนอ้างอิงคงที่ (Stationary Reference Frame)

$$\frac{d\bar{\boldsymbol{\psi}}_s}{dt} = \bar{\mathbf{u}}_s - R_S \bar{\mathbf{i}}_s \qquad (3\text{-}15)$$

และสมการ (2-52) พลวัตรของโรเตอร์ฟลักซ์บน แกนอ้างอิงโรเตอร์ฟลักซ์ (Rotor Flux Reference Frame)

$$\frac{d\bar{\boldsymbol{\psi}}_r^e}{dt} = \frac{L_m}{T_r}\bar{\mathbf{i}}_s^e + \left(-\frac{1}{T_r} + j\left(\omega_e - \omega_r\right)\right)\bar{\boldsymbol{\psi}}_r^e \tag{3-16}$$

โดยที่ ω_r เป็นความเร็วทางไฟฟ้าของโรเตอร์ แทน (2-53) $\bar{\mathbf{i}}_s^e = \frac{L_r}{L_\sigma}\bar{\boldsymbol{\psi}}_s^e - \frac{L_m}{L_\sigma}\bar{\boldsymbol{\psi}}_r^e$ ลงใน (2-52) ได้ดังนี้

$$\frac{d}{dt}\bar{\boldsymbol{\psi}}_r^e = \frac{L_m R_r}{L_\sigma}\bar{\boldsymbol{\psi}}_s^e - \left(\frac{L_s R_r}{L_\sigma} - j(\omega_e - \omega_r)\right)\bar{\boldsymbol{\psi}}_r^e \tag{3-17}$$

ให้ $\bar{\boldsymbol{\psi}}_r^e = \psi_{rd} + j\psi_{rq} = \psi_{r\alpha}^e + j\psi_{r\beta}^e$

$\bar{\boldsymbol{\psi}}_s^e = \psi_{sd} + j\psi_{sq} = \psi_{s\alpha}^e + j\psi_{s\beta}^e$

จากการเลือกแกนอ้างอิงโรเตอร์ฟลักซ์ (Rotor Flux Reference Frame) ทำให้ได้ว่าฟลักซ์คล้องโรเตอร์จะวางอยู่ตามแกนดีเท่านั้น $\psi_{r\alpha}^e = \psi_{rd}$, $\psi_{r\beta}^e = \psi_{rq} = 0$ นั่นคือ

$$\frac{d}{dt}\psi_{r\alpha}^e = \frac{L_m R_r}{L_\sigma}\psi_{s\alpha}^e - \frac{L_s R_r}{L_\sigma}\psi_{r\alpha}^e \tag{3-18}$$

สมการ (3-14)และ (3-16) เป็นความสัมพันธ์ที่ใช้ในการใช้คำณวนค่าฟลักซ์โรเตอร์และฟลักซ์สเตเตอร์ โดยการเพิ่มเทอมปรับของสไลดิงเพื่อช่วยในการแก้ไขความผิดพลาดอันเนื่องมาจากพารามิเตอร์ไม่ถูกต้อง กำหนดให้ $\varepsilon = \bar{\mathbf{i}}_s - \hat{\mathbf{i}}_s = \varepsilon i_{s\alpha} + j\varepsilon i_{s\beta} = (i_{s\alpha} - \hat{i}_{s\alpha}) + j(i_{s\beta} - \hat{i}_{s\beta})$

$$\frac{d\bar{\boldsymbol{\psi}}_s}{dt} = \bar{\mathbf{u}}_s - R_S\bar{\mathbf{i}}_s + \left(K_{1p} + \frac{K_{1i}}{s}\right)\mathrm{sgn}(\bar{\mathbf{i}}_s - \hat{\mathbf{i}}_s) \tag{3-19}$$

สมการ (3-17) เป็นสเปซเวกเตอร์ของฟลักซ์สเตเตอร์บนแกนอ้างอิงคงที่ (Stationary Reference Frame) ที่ถูกเติมเทอมปรับของสไลดิง ในทำนองดียวกันฟลักซ์โรเตอร์บนแกนอ้างอิงโรเตอร์ฟลักซ์ (Rotor Flux Reference Frame) ที่ถูกเติมเทอมปรับของสไลดิง

$$\frac{d}{dt}\bar{\boldsymbol{\psi}}_r^e = \frac{L_m R_r}{L_\sigma}\bar{\boldsymbol{\psi}}_s^e - \left(\frac{L_s R_r}{L_\sigma} - j(\omega_e - \omega_r)\right)\bar{\boldsymbol{\psi}}_r^e + K_2\,\mathrm{sgn}(\varepsilon\bar{\mathbf{i}}_s^e) \tag{3-20}$$

โดยที่ $\varepsilon \bar{\mathbf{i}}_s^e = (\bar{\mathbf{i}}_s - \hat{\mathbf{i}}_s)e^{-j\theta_{\psi r}}$ และ $\theta_{\psi r} = \tan^{-1}\left(\frac{\psi_{r\beta}}{\psi_{r\alpha}}\right)$ ฟลักซ์คล้องโรเตอร์ $\left(\psi_{r\alpha}, \psi_{r\beta}\right)$ ที่ถูกใช้ในการหามุมฟลักซ์โรเตอร์คือความสัมพันธ์ในสมการ (2-46) และ (2-47) $\psi_{r\alpha} = \frac{L_r}{L_m}\psi_{s\alpha} - \frac{L_\sigma}{L_m} i_{s\alpha}$ และ $\psi_{r\beta} = \frac{L_r}{L_m}\psi_{s\beta} - \frac{L_\sigma}{L_m} i_{s\beta}$

จากสมการของตัวสังเกตการณ์ (3-17) และ (3-18) เขียนในรูปที่จะนำมาใช้ในการคำณวน

$$\psi_{s\alpha} = \int\left(u_{s\alpha} - R_s i_{s\alpha} + (K_{1p} + \frac{K_{1i}}{S})\operatorname{sgn}(ei_{s\alpha})\right)dt \tag{3-21}$$

$$\psi_{s\beta} = \int\left(u_{s\beta} - R_s i_{s\beta} + (K_{1p} + \frac{K_{1i}}{S})\operatorname{sgn}(ei_{s\beta})\right)dt \tag{3-22}$$

$$\psi_{r\alpha}^e = \int(\frac{L_m R_r}{L_\sigma}\psi_{s\alpha}^e - \frac{L_s R_r}{L_\sigma}\psi_{r\alpha}^e + K_2 \operatorname{sgn}(e\bar{\mathbf{i}}_s^e))dt \tag{3-23}$$

และทำการปรับเวกเตอร์ของฟลักซ์โรเตอร์บนแกนอ้างอิงอ้างอิงคงที่ (Stationary Reference Frame)

$$\psi_{r\alpha} + j\psi_{r\beta} = e^{j\theta_{\psi r}}\psi_{r\alpha}^e = e^{j\theta_{\psi r}}\left|\bar{\boldsymbol{\psi}}_r^e\right| \tag{3-24}$$

ส่วนการหาค่ากระแสสเตเตอร์ประมาณหลังจากทำการปรับเวกเตอร์ของฟลักซ์โรเตอร์บนแกนอ้างอิงอ้างอิงคงที่แล้ว ทำได้โดยใช้ความสัมพันธ์ของสมการ (2-46) และ(2-47) นำมาแปลงได้ดังนี้

$$\hat{i}_{s\alpha} = \frac{L_r}{L_\sigma}\psi_{s\alpha} - \frac{L_m}{L_\sigma}\psi_{r\alpha} \tag{3-25}$$

$$\hat{i}_{s\beta} = \frac{L_r}{L_\sigma}\psi_{s\beta} - \frac{L_m}{L_\sigma}\psi_{r\beta} \tag{3-26}$$

เมื่อได้ปริมาณของฟลักซ์สเตเตอร์และฟลักซ์โรเตอร์แล้วปริมาณเหล่านี้จะถูกนำไปใช้ในการประมาณค่าแรงบิด และ ขนาดของฟลักซ์สเตเตอร์ เพื่อนำไปใช้ในการควบคุมในวงรอบปิดของ แรงบิด และ ขนาดของฟลักซ์สเตเตอร์ตามลำดับ

การประมาณค่าแรงบิดชั่วขณะของมอเตอร์เหนี่ยวนำทำได้โดยใช้ความสัมพันธ์ของการครอสเวกเตอร์ (Cross-vectorial Product) สเปซเวกเตอร์ของสเตเตอร์ฟลักซ์และสเปซเวกเตอร์ของกระแสสเตเตอร์ โดยที่ทั้งคู่ถูกพิจารณาในแกนอ้างอิงคงที่ (Stationary Reference Frame)

$$T_e = \frac{3}{2} P \bar{\boldsymbol{\psi}}_s \times \bar{\mathbf{i}}_s \tag{3-27}$$

โดยที่ P คือ จำนวนคู่ของขั้วแม่เหล็ก (Pole Pairs)

$\bar{\boldsymbol{\psi}}_s$ คือ สเปซเวกเตอร์ของสเตเตอร์ฟลักซ์

$\bar{\mathbf{i}}_s$ คือ สเปซเวกเตอร์ของกระแสสเตเตอร์

$$T_e = \frac{3}{2} P \left(\psi_{s\alpha} i_{s\beta} - \psi_{s\beta} i_{s\alpha} \right) \tag{3-28}$$

สมการ(3-26)คือสมการที่ใช้ในการคำนวณหาแรงบิดชั่วขณะ ส่วนการหาขนาดของเวกเตอร์ของสเตเตอร์ฟลักซ์โดยทั่วไปคำนวณจากการหาขนาดของเวกเตอร์ในแกนตั้งฉาก $\left|\bar{\boldsymbol{\psi}}_s\right| = \sqrt{\psi_{s\alpha}^2 + \psi_{s\beta}^2}$ แต่เนื่องจากวิทยานิพนธ์นี้จำเป็นต้องหามุมของฟลักซ์สเตเตอร์เพื่อใช้ในวงรอบฟลักซ์และแรงบิดอยู่แล้ว สมการดังกล่าวจะไม่ถูกนำมาใช้ในการคำนวณจริงเนื่องจาก การประมวลผลด้วยตัวประมวลผลสัญญาณดิจิตอล (DSP) การหารากที่ 2 จำเป็นต้องใช้ อัลกอริทึมทางการคำนวณเชิงตัวเลขที่ซับซ้อนและอาศัย การทำซ้ำหลายรอบ ดังนั้นในวิทยานิพนธ์นี้จึงใช้ความสัมพันธ์ของการแปลงรูปจากอัลฟ่าเบต้าไปที่ดีคิว (Park Transformation : $\alpha\beta$ to dq, $e^{-j\theta_{\psi s}}$)

$$\theta_{\psi s} = \tan^{-1}\left(\frac{\psi_{s\beta}}{\psi_{s\alpha}} \right) \tag{3-29}$$

เมื่อได้มุมฟลักซ์แล้วทำการการแปลงรูปของ ฟลักซ์สเตเตอร์จาก อัลฟ่าเบต้าไปที่ดีคิวของ ฟลักซ์สเตเตอร์ ดังนั้นจึงมีเฉพาะขนาด ฟลักซ์สเตเตอร์บนแกนดีเท่านั้น

$$\bar{\boldsymbol{\psi}}_s e^{-j\theta_{\psi s}} = \bar{\boldsymbol{\psi}}_s^{\psi_s} = \left|\Psi_s\right| \tag{3-30}$$

การทำงานของตัวสังเกตการณ์สไลดิงเพื่อทำการหาตัวแปรควบคุม $T_e, \theta_{\psi s}, \left|\Psi_s\right|, \bar{\boldsymbol{\psi}}_r$ เมื่อได้ปริมาณเหล่านี้จะถูกนำไปใช้ในวงรอบความเร็ว แรงบิด และขนาดฟลักซ์สเตเตอร์ ตามลำดับ

3.4 การประมาณความเร็วของมอเตอร์เหนี่ยวนำแบบวงรอบเปิด (Open-loop Speed Estimator)

ในส่วนนี้การประมาณหาความเร็วรอบของโรเตอร์และความเร็วของสลิป จะได้รับจากการพิจารณาของสมการแรงดันมอเตอร์เหนี่ยวนำซึ่งได้ใช้แรงดันสเตเตอร์และกระแสสเตเตอร์ ในส่วนของแบบจำลองทาง

คณิตศาสตร์ของมอเตอร์เหนี่ยวนำที่ใช้อยู่ในแกนอ้างอิงคงที่ (Stationary Reference Frame) ดังนั้นวิธีการของการประมาณความเร็วแบบวงรอบเปิดจะสิ่งที่มีผลกระทบสูงได้แก่ความเที่ยงตรงของพารามิเตอร์ของมอเตอร์เหนี่ยวนำ ที่ความเร็วรอบของโรเตอร์ต่ำความถูกต้องของตัวประมาณก็จะลดลงด้วย โดยเฉพาะอย่างยิ่งเมื่อพารามิเตอร์มีการเบี่ยงเบนไม่ตรงตามจริงก็จะมีผลกระทบอย่างมากด้วยในสภาวะคงที่และสภาวะชั่วขณะของระบบขับเคลื่อนที่ใช้วิธีการประมาณความเร็วแบบลูปเปิด นอกจากนี้แล้วความถูกต้องที่มากขึ้นของตัวประมาณความเร็วจะขึ้นกับความถูกต้องของตัวประมาณ สเตเตอร์ฟลักซ์และ โรเตอร์ฟลักซ์ซึ่งเป็นปริมาณที่มีความสำคัญมากในการควบคุมความเร็วแบบเวกเตอร์ โดยส่วนมาก สเตเตอร์ฟลักซ์ที่ถูกคำนวณเป็นปริมาณแรกสมการที่ (3-19) และ (3-20) จะไม่ถูกนำมาคำนวณจริงเนื่องจากการใช้ ตัวอินติเกรต (Pure Integrator) จะทำให้ฟลักซ์ที่ได้รับไม่สมมาตร เมื่อมีการเปลี่ยนความถี่ของแรงดันอ้างอิง สมการที่ (3-19) และ (3-20) จะถูกนำมาดัดแปลงโดยใช้ตัวกรองความถี่ต่ำ (Low-pass Filter) แทนตัวอินทีเกรตซึ่งให้ผลได้ดี

ความต้านทานของสเตเตอร์จะมีผลกระทบต่อฟลักซ์คล้องสเตเตอร์ (Stator Flux Linkage) โดยเฉพาะที่ความเร็วรอบต่ำและฟลักซ์คล้องโรเตอร์ (Rotor Flux Linkage) ที่ได้จากฟลักซ์คล้องสเตเตอร์ ดังนั้นฟลักซ์คล้องโรเตอร์ก็จะได้รับผลกระทบของความถูกต้องด้วย อย่างไรก็ตามโครงสร้างของตัวประมาณจะอยู่ในรูปแบบที่ง่ายๆ เมื่อเปรียบเทียบกับวิธีการอื่นๆ สมการของแบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำที่ใช้อยู่ในแกนอ้างอิงคงที่ (Stationary Reference Frame) ซึ่งจะแสดงได้ดังข้างล่าง

$$\left|\bar{\boldsymbol{\psi}}_r\right| = \sqrt{\psi_{r\alpha}^2 + \psi_{r\beta}^2} \tag{3-31}$$

$$\theta_{\psi r} = \tan^{-1}\left(\frac{\psi_{r\beta}}{\psi_{r\alpha}}\right) \tag{3-32}$$

ความเร็วของฟลักซ์โรเตอร์ $\left(\omega_e\right)$ สามารถคำนวณได้จากอนุพันธ์ของมุมโรเตอร์ฟลักซ์จากสมการ (3-29)

$$\omega_e = \frac{d\theta_{\psi r}}{dt} = \frac{d\left[\tan^{-1}\dfrac{\psi_{r\beta}}{\psi_{r\alpha}}\right]}{dt} \tag{3-33}$$

จาก $\dfrac{d\left(\tan^{-1}U\right)}{dt} = \dfrac{1}{1+U^2}\dfrac{dU}{dt}$ ที่ให้ $U = \dfrac{\psi_{r\beta}}{\psi_{r\alpha}}$ จะได้ดังนี้

$$\omega_e = \frac{d\theta_{\psi r}}{dt} = \frac{\psi_{r\alpha}^2}{\left|\bar{\boldsymbol{\psi}}_r\right|^2}\left(\frac{\psi_{r\alpha}\dfrac{d\psi_{r\beta}}{dt} - \psi_{r\beta}\dfrac{d\psi_{r\alpha}}{dt}}{\psi_{r\alpha}^2}\right) \tag{3-34}$$

แทน(2-33) ถึง (2-34) ลงในสมการ (3-31) โดยที่ $u_{r\alpha} = u_{r\beta} = 0$ จัดรูปใหม่จะได้ดังนี้

$$\omega_e = \frac{d\theta_{\psi r}}{dt} = \omega_r + \frac{1}{\left|\bar{\psi}_r\right|^2}\frac{L_m}{T_r}\left(\psi_{r\alpha} i_{s\beta} - \psi_{r\beta} i_{s\alpha}\right) \tag{3-35}$$

โดยที่ ω_r คือ ความเร็วเชิงมุมของโรเตอร์ (rad/sec)

R_r คือ ความต้านทานของขดลวดโรเตอร์ (Ohm)

$T_r = \frac{L_r}{R_r}$ คือ ค่าเวลาคงที่ของโรเตอร์

จากสมการ (3-13) เทอมที่สองทางซ้ายมือก็คือสลิปที่ซึ่งแปรผันโดยตรงกับแรงบิดเมื่อขนาดของโรเตอร์ฟลักซ์ถูกรักษาให้คงที่ สมการของแรงบิดแสดงได้ดังนี้

$$T_e = \frac{3}{2} P \frac{L_m}{L_r}\left(\psi_{r\alpha} i_{s\beta} - \psi_{r\beta} i_{s\alpha}\right) \tag{3-36}$$

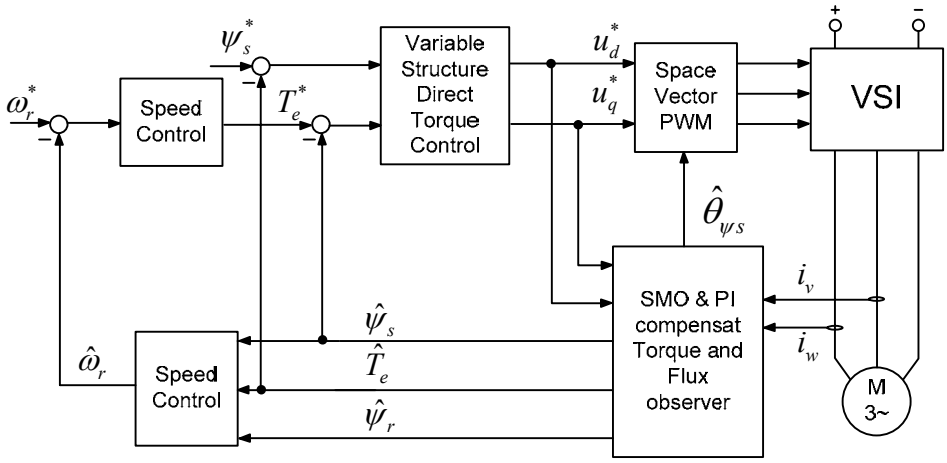
โดยที่ P คือ จำนวนคู่ขั้วแม่เหล็ก

ดังนั้นความเร็วของโรเตอร์สามารถหาได้ดังนี้

$$\omega_r = \omega_e - \frac{1}{\left|\bar{\psi}_r\right|^2}\frac{L_m}{T_r}\left(\psi_{r\alpha} i_{s\beta} - \psi_{r\beta} i_{s\alpha}\right) \tag{3-37}$$

$$\omega_r = \left(\frac{\psi_{r\alpha}\frac{d\psi_{r\beta}}{dt} - \psi_{r\beta}\frac{d\psi_{r\alpha}}{dt}}{\left|\bar{\psi}_r\right|^2}\right) - \frac{2T_e R_r}{3P\left|\bar{\psi}_r\right|^2} \tag{3-38}$$

เมื่อได้ความเร็วประมาณแล้วความเร็วนี้จะถูกนำไปใช้ในวงรอบปิดของความเร็วดังภาพที่ 3-3



ภาพที่ 3-4 โครงสร้างของการควบคุมแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็ว

3.5 สรุป

ในบทนี้ จะกล่าวถึงโครงสร้างของการควบคุมแบบแรงบิดโดยตรง วิธีการควบคุมรวมไปถึงตัวสังเกตการณ์และวิธีการประมาณค่าฟลักซ์ กระแส และความเร็วรอบของมอเตอร์เหนี่ยวนำที่ใช้ในงานวิจัยนี้

บทต่อไปเป็นการทดสอบการทำงานของการควบคุมแบบสไลดิงแรงบิดตรงชนิดไร้ตัวตรวจจับความเร็วบนแบบจำลองคอมพิวเตอร์ด้วยโปรแกรม MATLAB Simulink

บทที่ 4
ผลการจำลองระบบด้วยคอมพิวเตอร์

ในบทนี้กล่าวถึงผลของการทดสอบแบบจำลองระบบด้วยคอมพิวเตอร์ ที่ได้ทำการทดสอบระบบควบคุมในสภาวะต่างๆ การทดสอบทั้งหมดได้ทำการจำลองระบบควบคุมบน โปรแกรม MATLAB simulink โดยการปรับเปลี่ยนสภาวะการทำงานของ ระบบในช่วงความเร็ว ทิศทาง และการใส่ภาระ

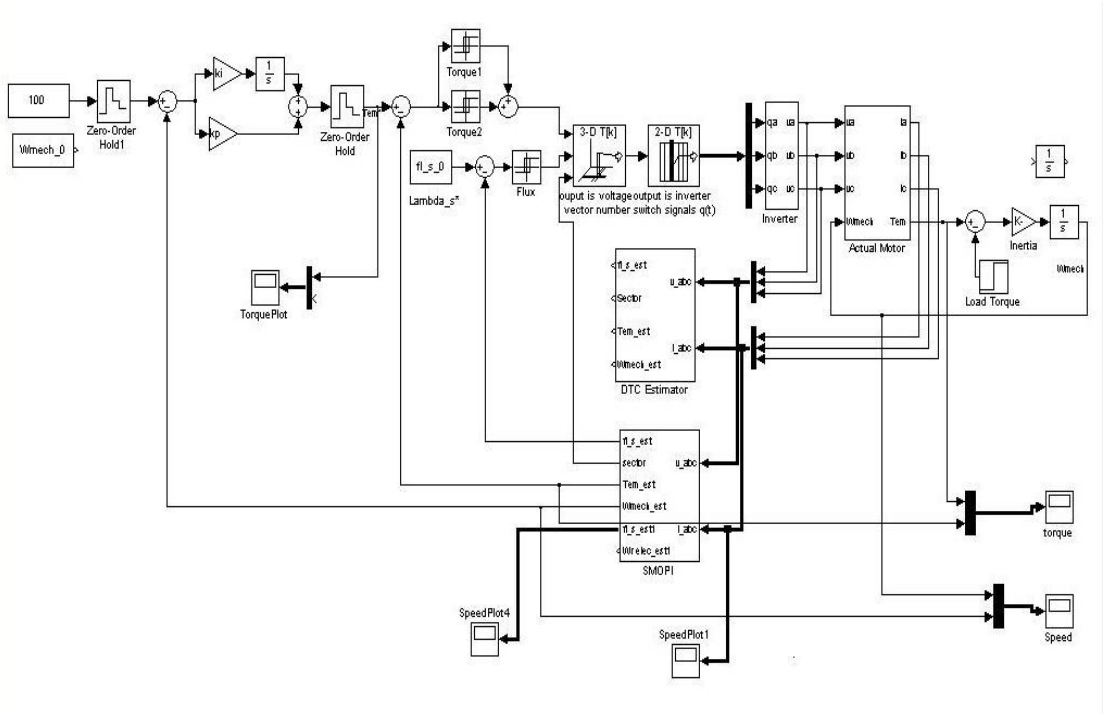
4.1 ผลการทดสอบแบบจำลองระบบในแบบต่างๆ

จากโครงสร้างระบบควบคุมแบบแรงบิดโดยตรงและตัวสังเกตุการณ์แบบสไลดิ้ง นั้นได้มีการทดสอบโดยใช้แบบจำลองด้วย โปรแกรม MATLAB Simulink โดยมีข้อมูลของมอเตอร์ดังตารางที่ 1 ซึ่งได้ทำการทดสอบโดยการใส่ค่า แรงบิดเริ่มต้นที่ 8 นิวตันเมตร และทำการเพิ่มเป็น 14 นิวตันเมตร ที่เวลา 0.6 วินาที ซึ่งให้ผลตอบสนองดังนี้

ตารางที่ 4-1 ข้อมูลของมอเตอร์เหนี่ยวนำ

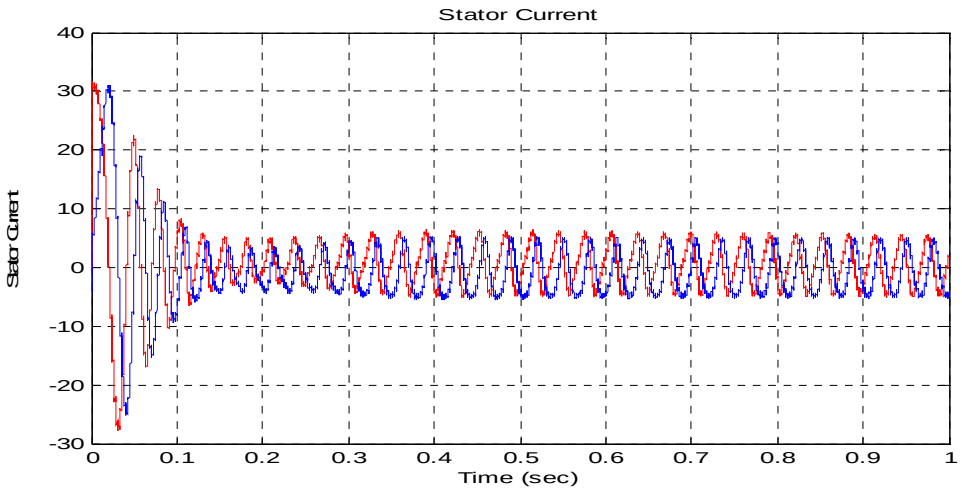
พิกัดกำลัง	1.1	kW
แรงดัน (u_s)	400/690	V
กระแส (i_s)	2.6/1.49	A
ตัวประกอบกำลัง	0.83	
พิกัดความเร็ว (ω_r)	1400	rpm
จำนวนคู่ของขั้วแม่เหล็ก (P)	2	คู่
ความต้านทานสเตเตอร์ (R_s)	26.3975	Ω /phase
ความต้านทานโรเตอร์ (R_r)	11.2708	Ω /phase
ความเหนี่ยวนำตัวเองด้านสเตเตอร์ (L_s)	1.3910	H/phase
ความเหนี่ยวนำตัวเองด้านโรเตอร์ (L_r)	1.3496	H/phase
ความเหนี่ยวนำของสนามแม่เหล็ก (L_m)	1.2866	H
ความเฉื่อยของโรเตอร์ (J_m)	0.0016	$kg \cdot m^2$

การทดสอบในส่วนนี้ ได้พิจารณาในส่วนของตัวสังเกตุการณ์แบบสไลดิ้งเป็นหลัก เพื่อพิสูจน์ให้ได้ว่าตัวสังเกตุการณ์นี้สามารถใช้งานได้จริง โดยมีแบบจังลองดังรูป

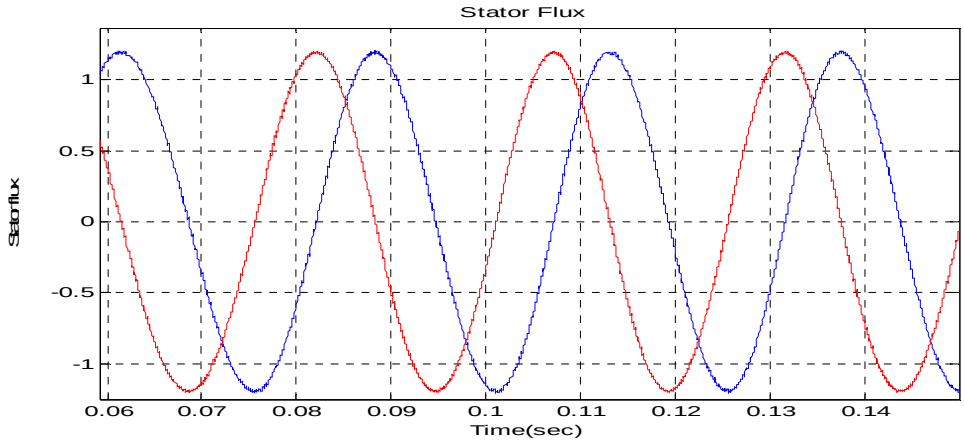


ภาพที่ 4-1 แบบจำลองของระบบ

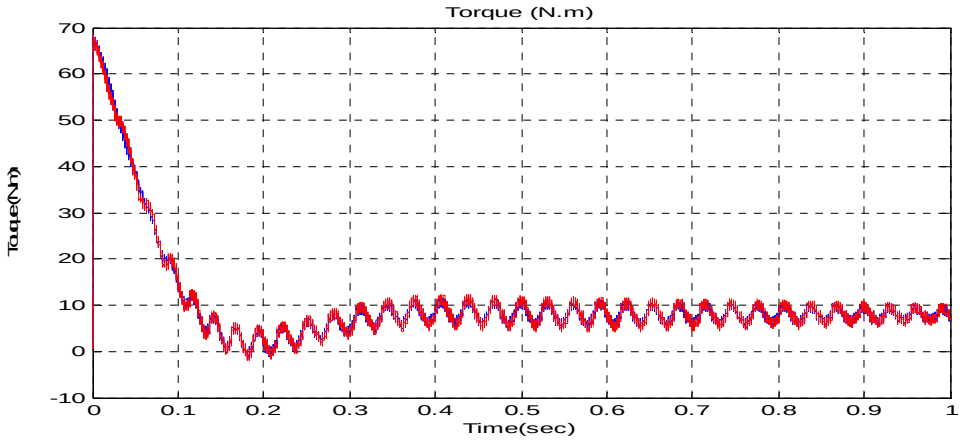
การทดสอบแบบจำลองในสภาวะ ความเร็วรอบ 1000 รอบต่อนาทีและใส่ภาระขนาด 8 นิวตันเมตร คงที่ (ใส่ภาระก่อนมอเตอร์หมุน)



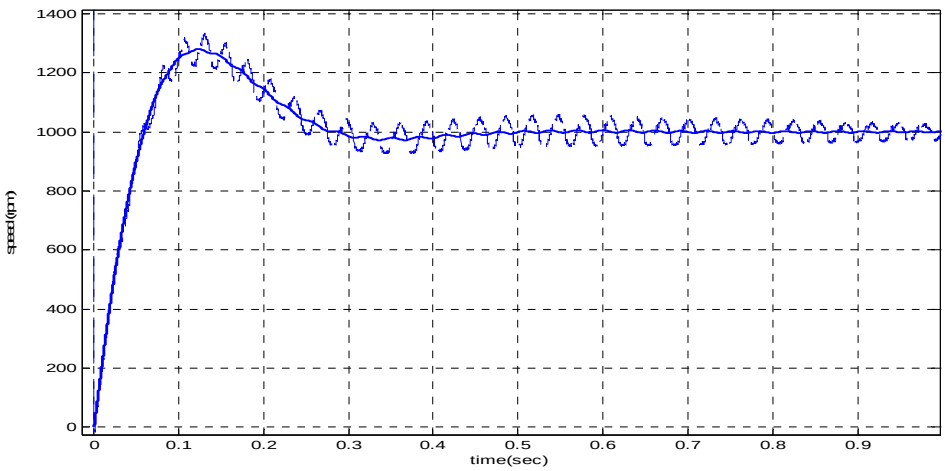
ภาพที่ 4-2 สัญญาณกระแสสเตเตอร์



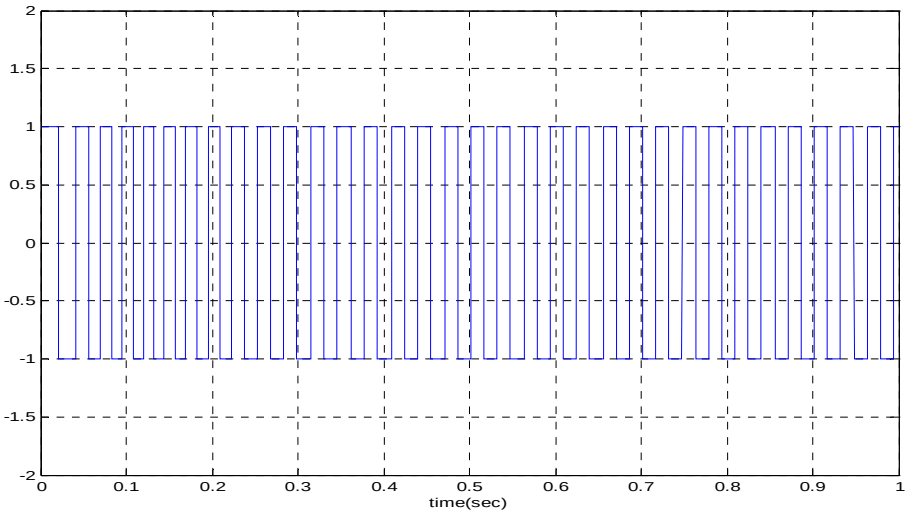
ภาพที่ 4-3 ฟลักซ์ที่สเตเตอร์



ภาพที่ 4-4 สัญญาณแรงบิด



ภาพที่ 4-5 สัญญาณความเร็วรอบ



ภาพที่ 4-6 สัญญาณระนาบการสไลดิง

4.2 สรุป

ในบทนี้กล่าวถึงผลจากการจำลองระบบด้วยคอมพิวเตอร์ การทดสอบระบบควบคุมและตัวสังเกตการแบบสไลดิงร่วมกับตัวชดเชย แบบ PI พบว่าในสภาวะต่างพบว่าสามารถควบคุมมอเตอร์เหนี่ยวนำให้เป็นไปในแบบที่ต้องการได้และเมื่อทำการทดสอบโดยให้ระบบมีตัวควบคุมเดียวกัน และมีการปรับตัวสังเกตุการณ์ในส่วนของ ตัวชดเชย แบบ PI นั้นส่งผลให้ระบบมีเสถียรภาพดีขึ้นอีกทั้งสามารถลดในเรื่องของการ Chattering ของตัวสไลดิงได้จริง

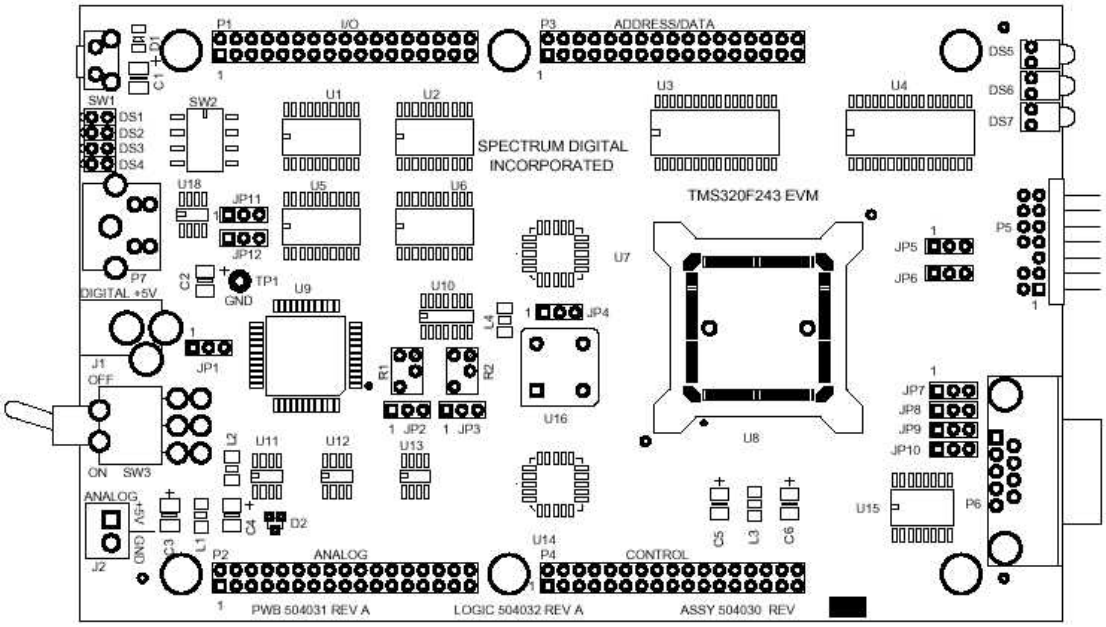
บทที่ 5

การทดลอง

ในบทนี้กล่าวถึงอุปกรณ์ต่างๆ ที่ใช้ในการทำวิจัยทั้งฮาร์ดแวร์และซอฟต์แวร์ ลำดับขั้นตอนการทำงานของโปรแกรมต่างๆ รวมถึงขั้นตอนและวิธีในการทดลองการต่างๆ

5.1 อุปกรณ์ที่ใช้ในการทดลอง

5.1.1 ตัวประมวลผลดิจิตอล รุ่น TMS320F243 Evaluation Module (Spectrum Digital, 2000) แสดงดังภาพที่ 5-1 คำนวณแบบจำนวนเต็ม (fixed-point) ขนาด 16 บิต ทำงานที่ 20 ล้านคำสั่งต่อวินาที (MIPS) หน่วยความจำ 128K ตัวแปลงจากอนาลอกเป็นดิจิตอลขนาด 10 บิต 8 ช่อง มีวงจรควบคุมความกว้างพัลส์ (Multiple PWM & capture channels onchip) ตัวแปลงดิจิตอลเป็นอนาลอก 4 ช่อง (MP7680) ในการเขียนโปรแกรมติดต่อกับตัวประมวลผลนั้นใช้โปรแกรม Code Composer เวอร์ชัน 4.10.36



ภาพที่ 5-1 ชุดประมวลผลสัญญาณดิจิตอล TMS320F243 Evaluation Module

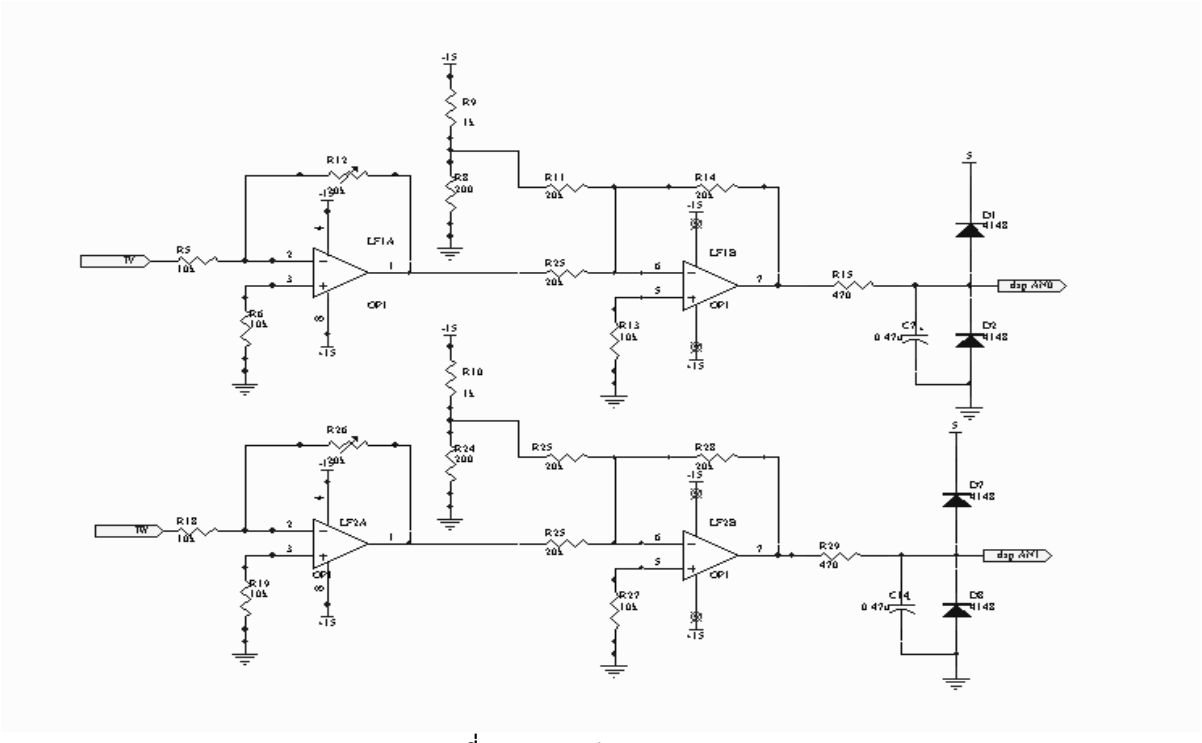
5.1.2 ชุดขับเคลื่อนมอเตอร์เหนี่ยวนำ อินพุท 3 เฟส 380 V 50-60 H_Z เอาท์พุท 18 A 0.5-300 H_Z

5.1.3 มอเตอร์เหนี่ยวนำแบบกรงกระรอกแบบสามเฟสของบริษัท Siemens ซึ่งมีค่าพิกัด 1.1 kW 2.6/1.49 A แรงดัน 400/690 V ความเร็วรอบ 1400 rad/s ค่าตัวประกอบกำลัง 0.83 ความละเอียดของเอ็นโคดเดอร์ 2048 พัลส์ต่อรอบ และชุดภาระของ SIEMENS

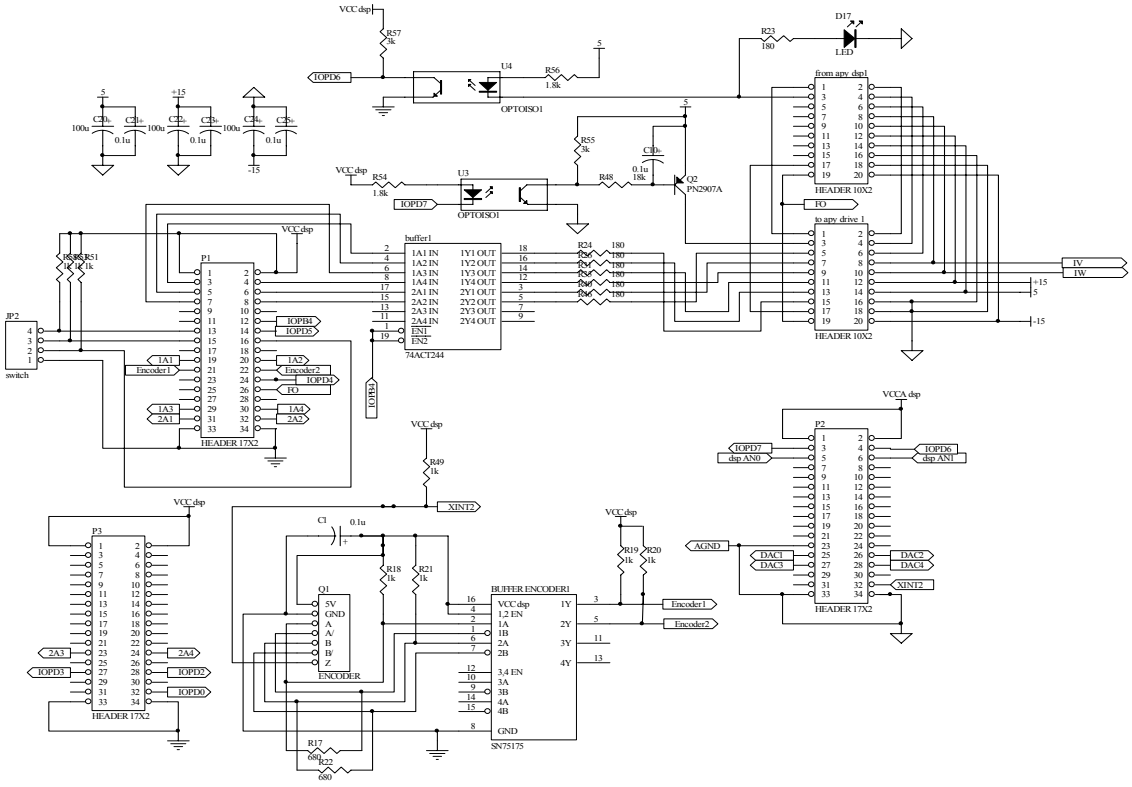
ตารางที่ 5-1 ค่าพารามิเตอร์ของมอเตอร์เหนี่ยวนำที่ทดลอง

พารามิเตอร์	ขนาด	หน่วย
ความต้านทานฝั่งสเตเตอร์ (R_s)	26.3975	Ω
ความเหนี่ยวนำตัวเองด้านสเตเตอร์ (L_s)	1.3910	H
ความเหนี่ยวนำของสนามแม่เหล็ก (L_m)	1.2866	H
ความต้านทานฝั่งโรเตอร์(R_r)	11.2708	Ω
ความเหนี่ยวนำตัวเองด้านโรเตอร์ (L_r)	1.3496	H
จำนวนคู่ของขั้วแม่เหล็ก (P)	2	คู่
โมเมนต์ความเฉื่อย (J)	0.0016	$kg\text{-}m^2$

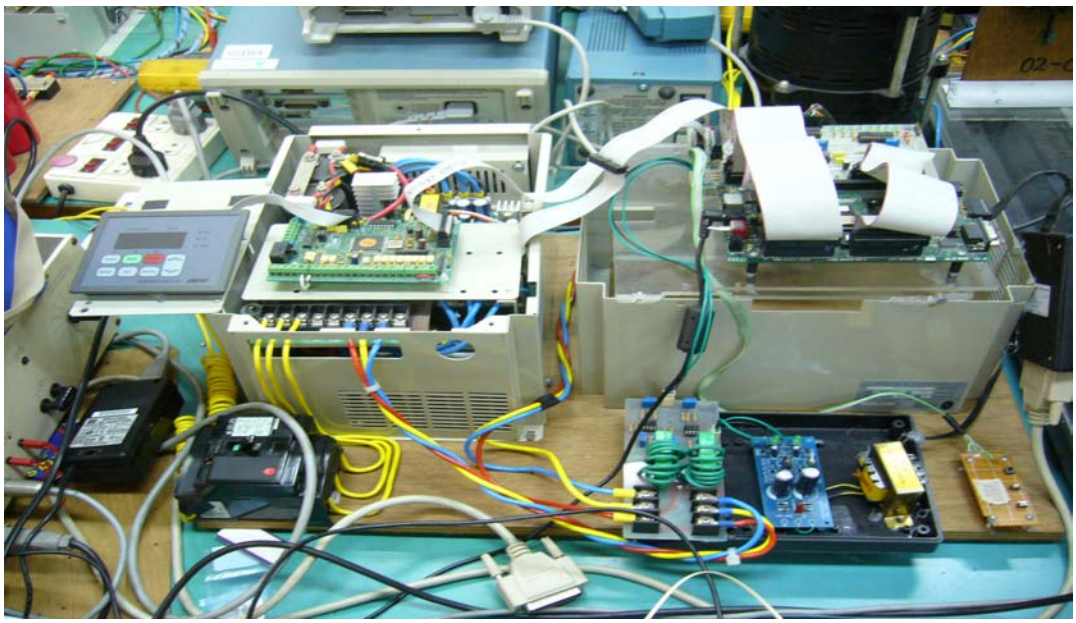
5.1.4 ชุดเชื่อมต่อระหว่างตัวประมวลผลกับชุดขับเคลื่อนมอเตอร์ เนื่องจากเป็นชุดขับเคลื่อนมอเตอร์เหนี่ยวนำ ซึ่งในงานวิจัยนี้ต้องการเฉพาะชุดที่จ่ายไฟให้มอเตอร์กับชุดป้องกันเท่านั้น ดังนั้นจึงต้องออกแบบชุดเชื่อมต่อ เพื่อตัดสัญญาณควบคุมเดิมออกแล้วนำสัญญาณควบคุมจากการวิจัยใส่เข้าไปแทน ภาพที่ 5-4 แสดงให้เห็นถึงวงจรต่างๆ ที่อยู่ในชุดเชื่อมต่อนี้



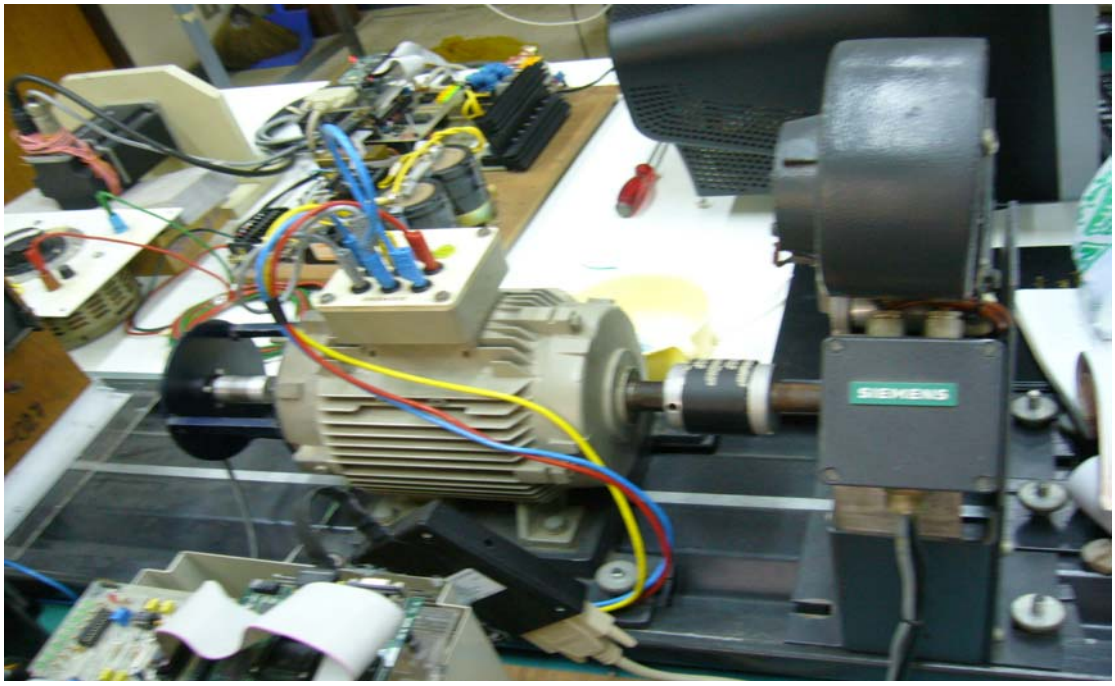
ภาพที่ 5-2 วงจรวัดกระแส



ภาพที่ 5-3 วงจรภายในชุดเชื่อมต่อระหว่างตัวประมวลผลกับชุดขับเคลื่อนมอเตอร์



ภาพที่ 5-4 การเชื่อมต่อระหว่างตัวประมวลผลกับชุดขับเคลื่อนมอเตอร์

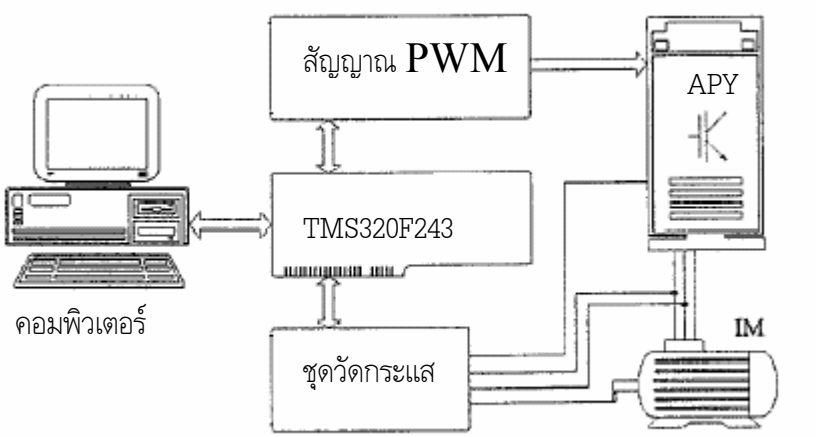


ภาพที่ 5-5 มอเตอร์เหนี่ยวนำติดตั้งชุดภาระและตัวตรรวจจับความเร็วเอนโคดเดอร์

5.2 การทดลอง

ในส่วนนี้กล่าวถึงโครงสร้างของการทดลองทั้งฮาร์ดแวร์และซอฟต์แวร์

5.2.1 โครงสร้างฮาร์ดแวร์ของการทดลอง แสดงดังภาพที่ 5-5 มีการทำงานดังนี้ คอมพิวเตอร์ทำหน้าที่ส่งสัญญาณ PWM ไปยังชุดขับเคลื่อนเพื่อทำการหมุนมอเตอร์โดยผ่านโปรแกรม Code Composer โดยมอเตอร์ถูกต่ออยู่กับชุดวัดกระแสเพื่อป้อนให้ตัวประมวลผลเพื่อทำการคำนวณในลูปต่อไป



ภาพที่ 5-6 โครงสร้างฮาร์ดแวร์ของการทดลอง

5.2.2 การทำงานของซอฟต์แวร์ สามารถแบ่งออกเป็นสองส่วนได้แก่ โปรแกรมหลักซึ่งทำงานตลอดเวลาและโปรแกรมย่อยซึ่งทำงานในช่วงการเกิดการขัดจังหวะ (Interrupt) ส่วนลำดับการทำงานของโปรแกรมที่ใช้ในงานวิจัยนี้ดูได้ในภาคผนวก ข

5.2.2.1 โปรแกรมหลัก เป็นโปรแกรมที่ทำงานตลอดเวลา โดยเริ่มแรกของโปรแกรม ทำการกำหนดค่าต่างๆ ให้กับตัวประมวลผล จากนั้นทำการวนลูปเพื่อรอรับคำสั่ง โดยใช้เป็นปุ่มกดติดปล่อยดับจำนวนสามปุ่ม รับคำสั่งสองคำสั่งคือ คำสั่งหมุนมอเตอร์ และคำสั่งหยุดหมุนมอเตอร์ ได้มีการใส่การหน่วงเวลาไปด้วย เนื่องจากตัวประมวลผลมีความเร็วมาก ถ้าไม่ทำการหน่วงเวลาจะทำให้เวลากดปุ่มเกิดความผิดพลาดเหมือนกับการกดปุ่มซ้ำ ลำดับการทำงานของโปรแกรมดูได้ในภาคผนวก ข

5.2.2.2 โปรแกรมย่อย เป็นโปรแกรมที่ทำงานในช่วงการเกิดการขัดจังหวะของวงจรควบคุมความกว้างพัลส์ จะทำงานทุกๆ 100 μS เมื่อทำงานเสร็จแล้วก็จะกลับไปทำงานต่อในโปรแกรมหลักตามเดิม การทำงานของโปรแกรมย่อยมีดังนี้ เมื่อเข้าโปรแกรมจะทำการเก็บสถานะเดิมของโปรแกรมหลักเอาไว้ จากนั้นทำการอ่านกระแสเข้ามา และทำการตรวจสอบว่าเกินค่าที่ตั้งเอาไว้หรือเปล่า ถ้าเกินก็เรียกสถานะเดิมของ

โปรแกรมหลักมาแล้วจบการทำงาน ถ้าไม่เกินก็ทำการแปลง $i_{abc} \rightarrow i_{\alpha\beta}$ จากนั้นทำการตรวจสอบสถานะว่าถ้าทำการหมุนมอเตอร์ จะทำการเก็บข้อมูลของตัวแปรที่ต้องการเข้าในหน่วยความจำ จากกนั้นทำการอ่านเอ็ดโคดเดอร์เพื่อหาตำแหน่งของโรเตอร์และหาความเร็วของมอเตอร์ จากนั้นเข้าวงรอบควบคุมความเร็ว ซึ่งจะให้เอาท์พุตเป็นแรงบิดอ้างอิงซึ่งจะมีการจำกัดด้วยค่าสูงสุดและต่ำสุดเอาไว้ และเข้าวงรอบควบคุมแรงบิดและวงรอบฟลักซ์ ซึ่งจะสร้างแรงดันอ้างอิงบนแกนสเตเตอร์ฟลักซ์จากการควบคุมแบบสไลดิง จากนั้นทำการแปลง $V_{dq} \rightarrow V_{\alpha\beta}$ และเข้าในส่วนของวงจรควบคุมความกว้างพัลส์ และตัวสังเกตการณ์แบบสไลดิงและตัวประมาณความเร็วรอบ เมื่อทำงานเสร็จก็เรียกสถานะเดิมของโปรแกรมหลักก่อนจบการทำงาน โปรแกรมต่างๆ ในงานวิจัยนี้ดูได้ในภาคผนวก ค

5.3 วิธีการทดลอง

5.3.1 การทดสอบโปรแกรมการทำงานของตัวสังเกตการณ์แบบเปิดวงรอบ(Open-loop V/f test) เพื่อทดสอบการทำงานเบื้องต้น ในส่วนการทำงานของของตัวสังเกตการณ์สไลดิง โดยใช้สัญญาณกระแสจากการขับมอเตอร์ด้วยอินเวอร์เตอร์ในโหมดสเกลาร์ (V/f) และนำความเร็วที่ได้จากการประมาณมาเปรียบเทียบกับความเร็วรอบจากเอ็นโคดเดอร์ เพื่อทดสอบความแม่นยำและถูกต้องของการประมาณค่าต่าง ๆ

5.3.2 การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI แบบไม่ใส่ภาระ (มีภาระแรงเสียดทานเล็กน้อยจากชุดภาระ)

5.3.3 การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI แบบใส่ภาระก่อนเริ่มสตาร์ท เป็นการทดสอบวงปิดของความเร็ว วงปิดแรงบิด และวงปิดฟลักซ์ และความเร็วที่ได้จากการประมาณ ในช่วงความเร็วรอบต่างๆกันโดยเฉพาะที่สภาวะคงตัว

5.4 สรุป

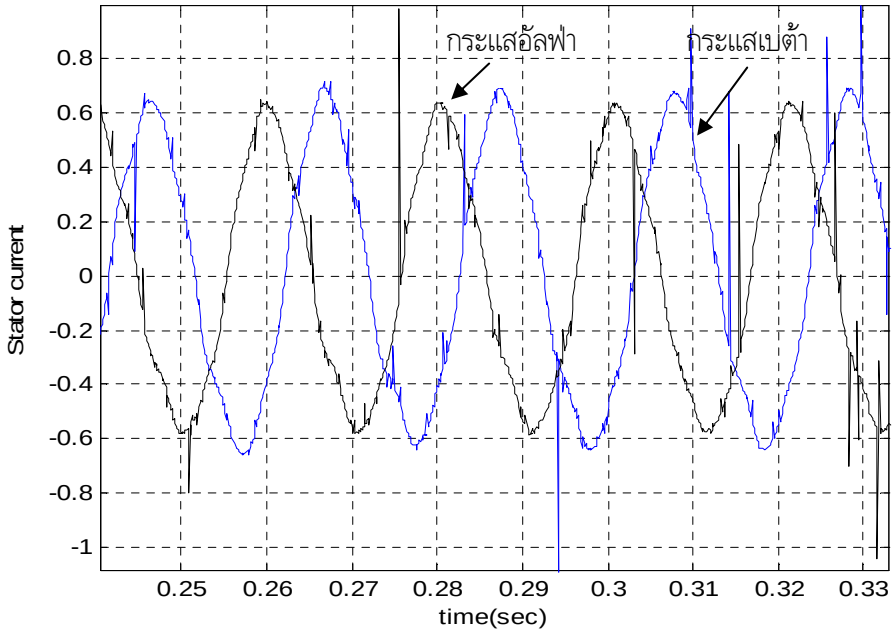
ในบทนี้กล่าวถึงอุปกรณ์ที่ใช้ในการทดลองทั้งฮาร์ดแวร์และซอฟต์แวร์ เช่น ตัวประมวลผลดิจิตอล มอเตอร์เหนี่ยวนำ ชุดขับเคลื่อน ชุดเชื่อมต่อ โปรแกรมที่ใช้ติดต่อกับตัวประมวลผล โปรแกรมที่ใช้ในการทดลอง รวมถึงขั้นตอนการทดลองต่างๆ ที่สภาวะไม่ใส่ภาระ และใส่ภาระ ผลการทดลองทั้งหมดนี้ จะกล่าวในบทต่อไป

บทที่ 6
การทดสอบระบบควบคุมมอเตอร์เหนี่ยวนำ

บทนี้กล่าวถึงขั้นตอนและผลการทดลองต่างๆ เพื่อศึกษาพฤติกรรมของ การควบคุมมอเตอร์เหนี่ยวนำแบบสไลดิงแรงบิดตรง โดยจะมีการแบ่งการทดสอบออกเป็น 3 การทดสอบได้แก่ การทดสอบโปรแกรมการทำงานของตัวสังเกตการสไลดิงแบบเปิดวงรอบ (Open-loop V/f Test) การทดสอบการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบ โดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ไม่ใส่ภาระ และการทดสอบการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบ โดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ใส่ภาระ

6.1 การทดสอบโปรแกรมการทำงานของตัวสังเกตการณ์แบบสไลดิง แบบเปิดวงรอบ (Open Loop Test)

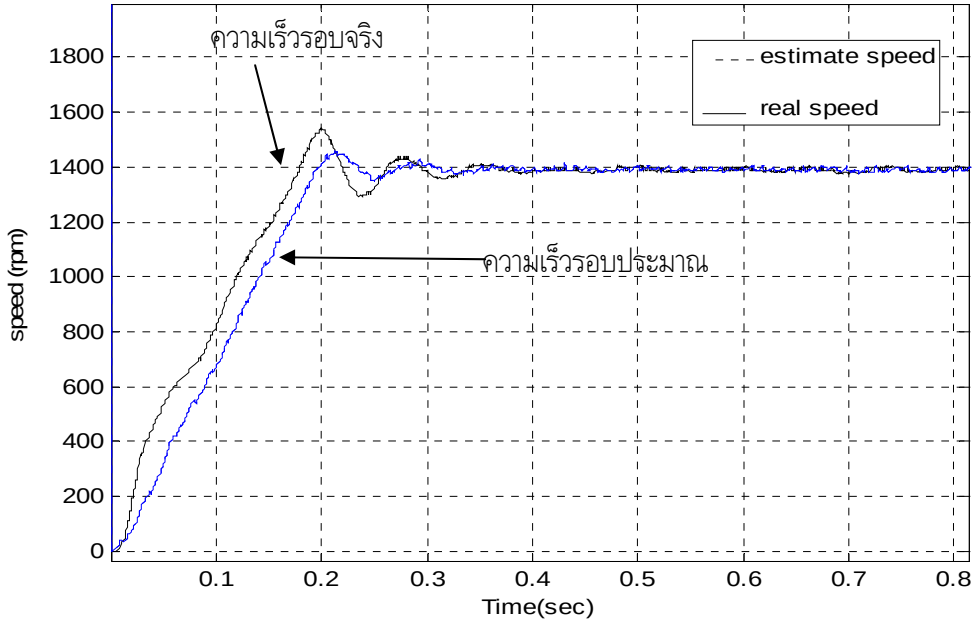
การทดสอบนี้เป็นการทดสอบในส่วนของตัวสังเกตการณ์ โดยใช้สัญญาณกระแสจากการขับมอเตอร์ด้วยอินเวอร์เตอร์ในโหมดสเกลาร์ (V/f)



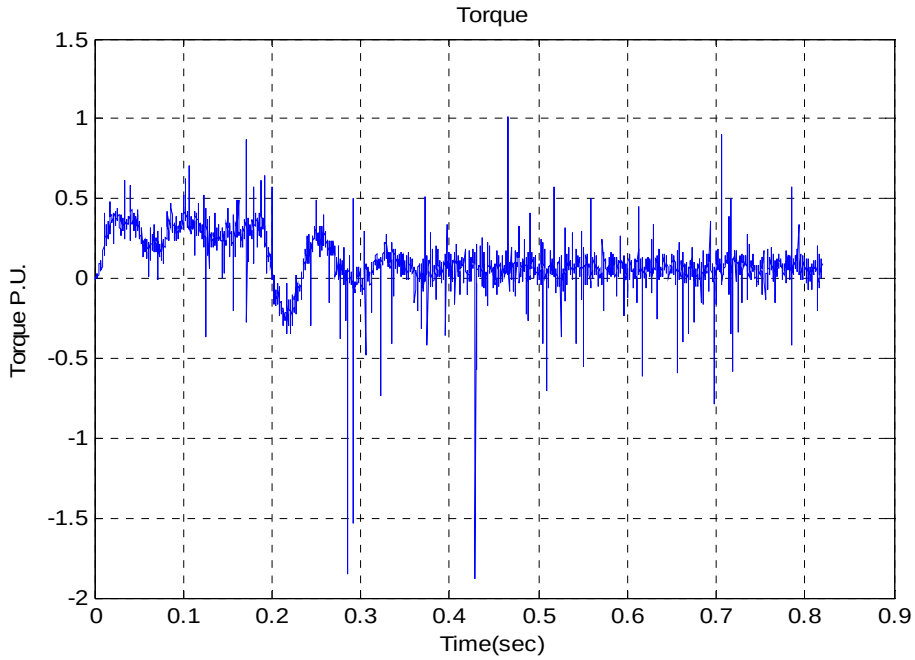
ภาพที่ 6-1 กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ

6.2 ผลการทดสอบการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ไม่ใส่ภาระ

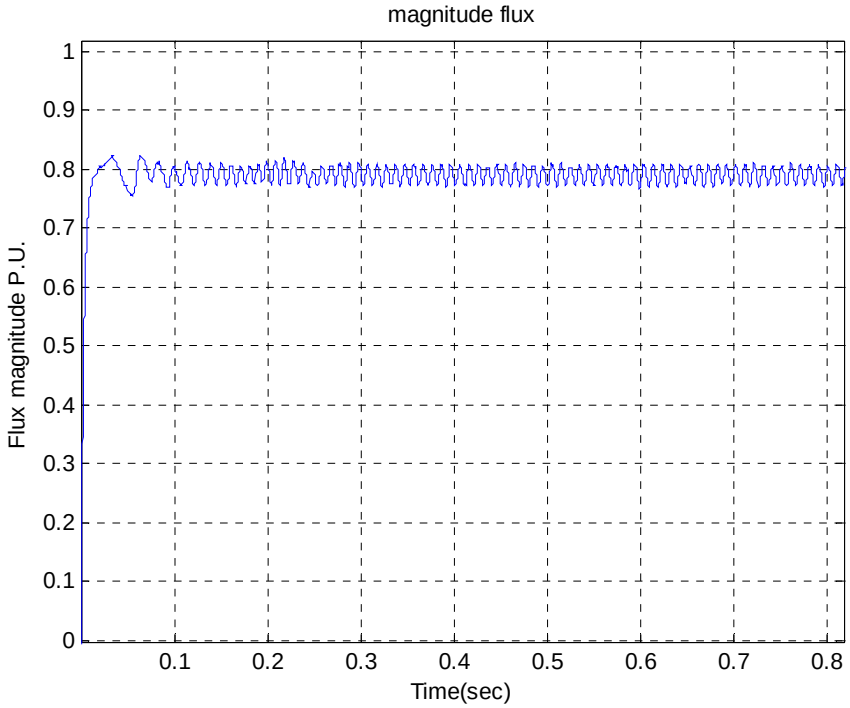
6.2.1 การทดสอบแบบไม่ใส่ภาระแรงเสียดทานก่อนเริ่มหมุนมอเตอร์ ความเร็วอ้างอิง 1400 รอบต่อนาที ระดับของฟลักซ์อ้างอิงอยู่ที่ 0.8 p.u



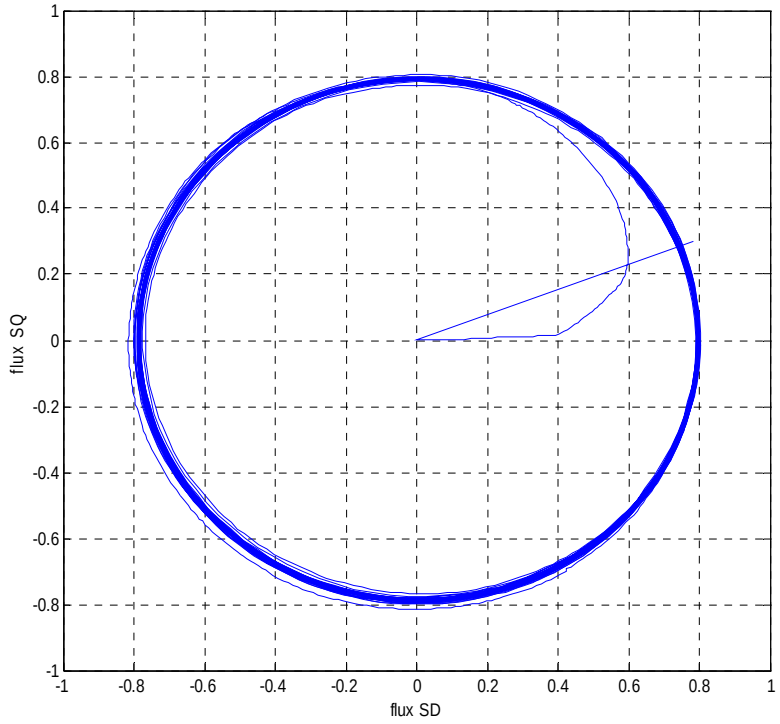
ภาพที่ 6-2 ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ



ภาพที่ 6-3 แรงบิดประมาณของมอเตอร์เหนี่ยวนำ

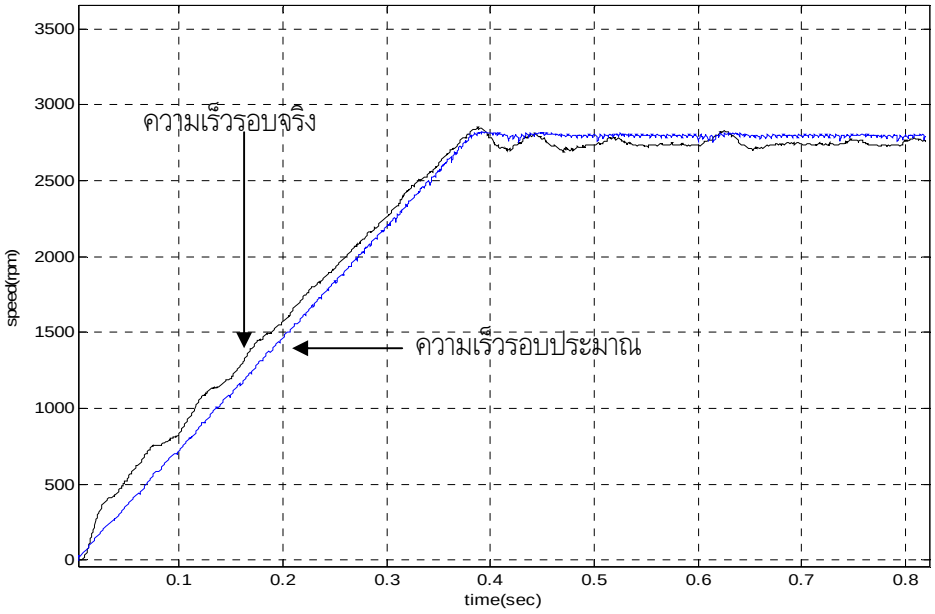


ภาพที่ 6-4 ขนาดของสเตเตอร์ฟลักซ์ของมอเตอร์เหนี่ยวนำ

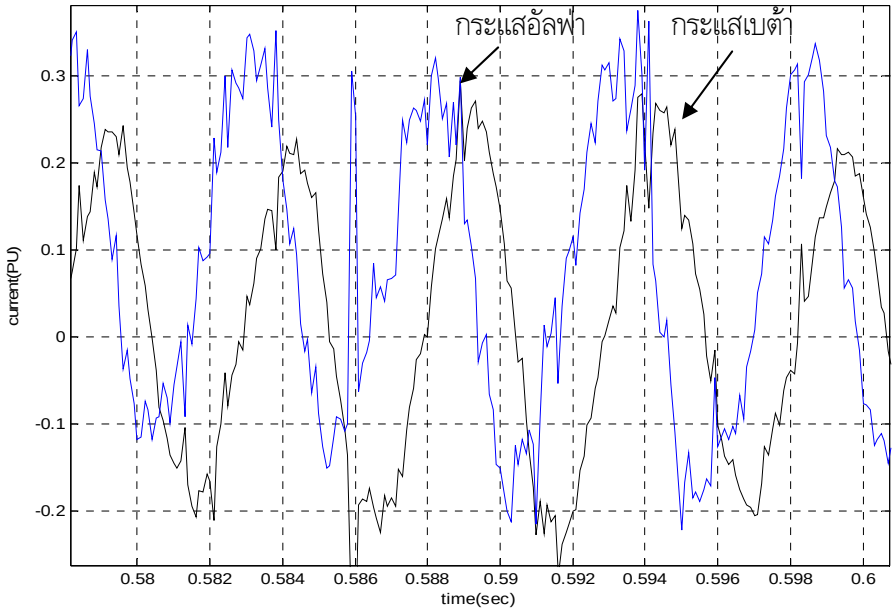


ภาพที่ 6-5 ทางเดินของสเตเตอร์ฟลักซ์ของแกนอัลฟ่าและเบต้า

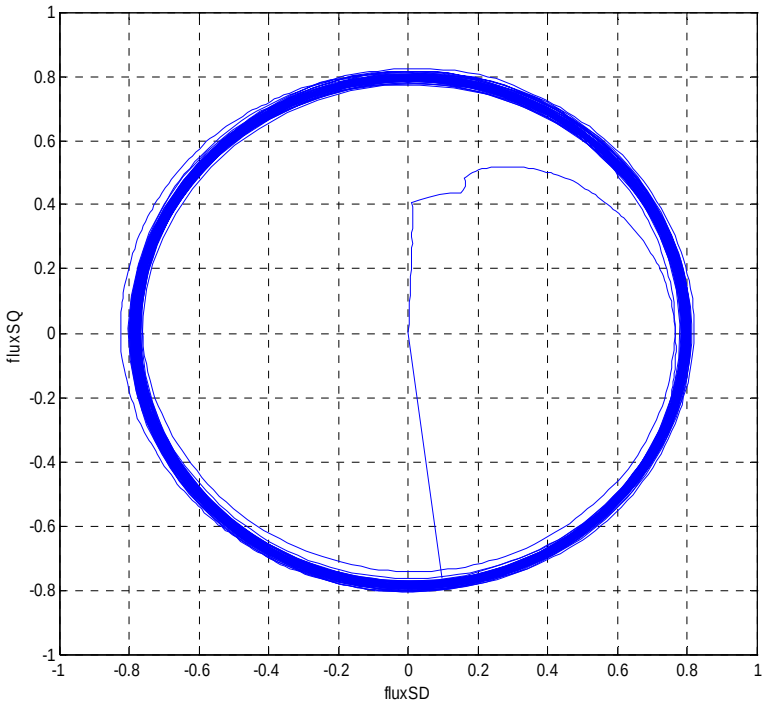
6.2.2 การทดสอบแบบไม่ใส่ภาระแรงเสียดทานก่อนเริ่มหมุนมอเตอร์ ความเร็วอ้างอิง 2800 รอบต่อนาที ระดับของฟลักซ์อ้างอิงอยู่ที่ 0.8 p.u



ภาพที่ 6-6 ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ



ภาพที่ 6-7 กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ

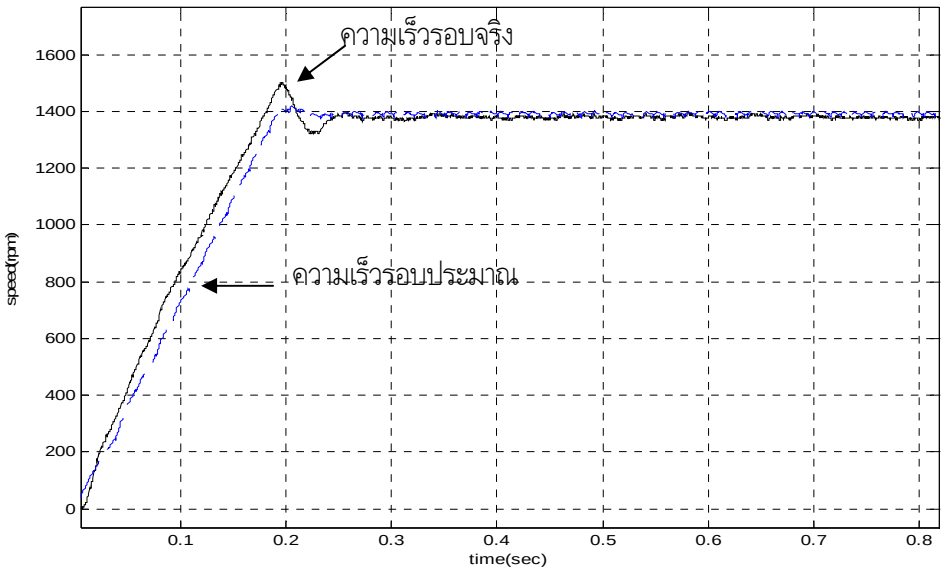


ภาพที่ 6-8 ทางเดินของสเตเตอร์ฟลักซ์ของแกนอัลฟ่าและเบต้า

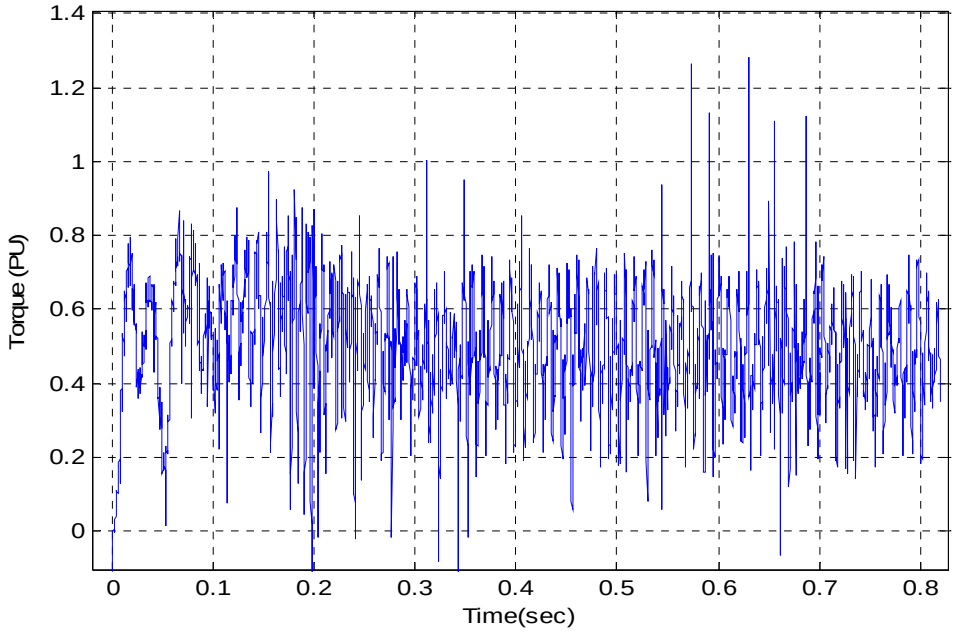
จากการทดสอบที่ 6.2.1 แบบไม่ใส่ภาระแรงเสียดทานก่อนเริ่มหมุนมอเตอร์พบว่าระบบควบคุมสามารถขับมอเตอร์ไปที่ความเร็วอ้างอิงได้ในสภาวะคงตัว ขนาดของสเตเตอร์ฟลักซ์ถูกควบคุมให้มีขนาดตามฟลักซ์อ้างอิง

6.3 ผลการทดสอบการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ใส่ภาระ

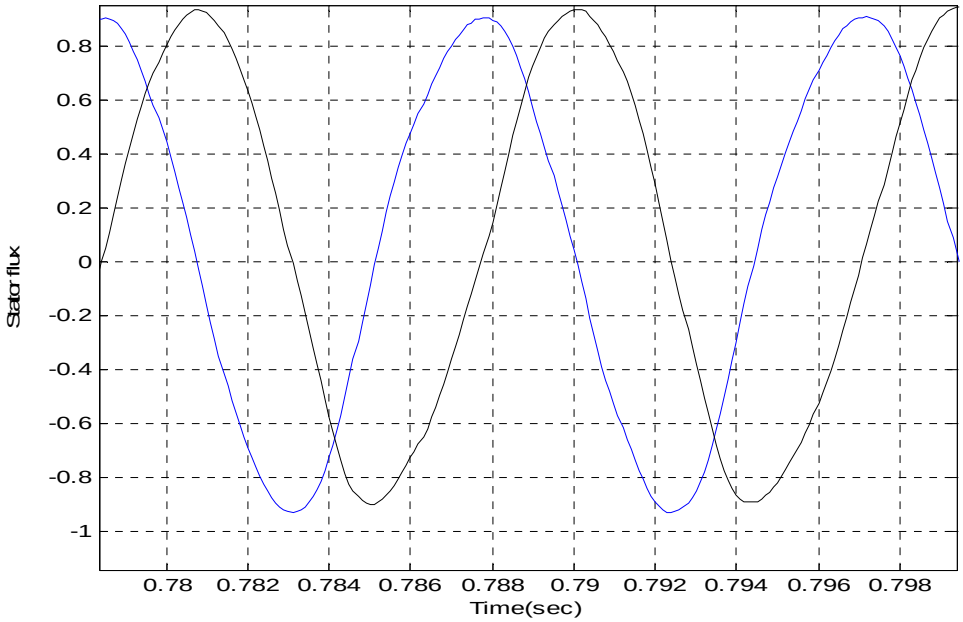
6.3.1 การทดสอบแบบใส่ภาระแรงเสียดทานก่อนเริ่มหมุนมอเตอร์ ความเร็วอ้างอิง 1400 รอบต่อนาที ระดับของฟลักซ์อ้างอิงอยู่ที่ 0.8 ต่อหน่วยใส่ภาระขนาด 0.5 เท่าของกระแสพิกัดมอเตอร์



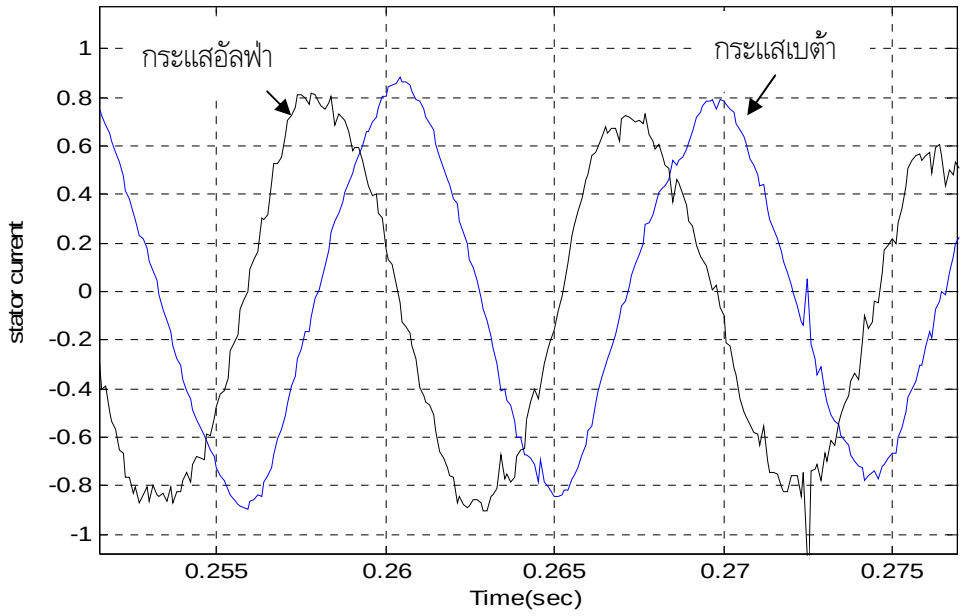
ภาพที่ 6-9 ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ



ภาพที่ 6-10 แรงบิดประมาณของมอเตอร์เหนี่ยวนำ

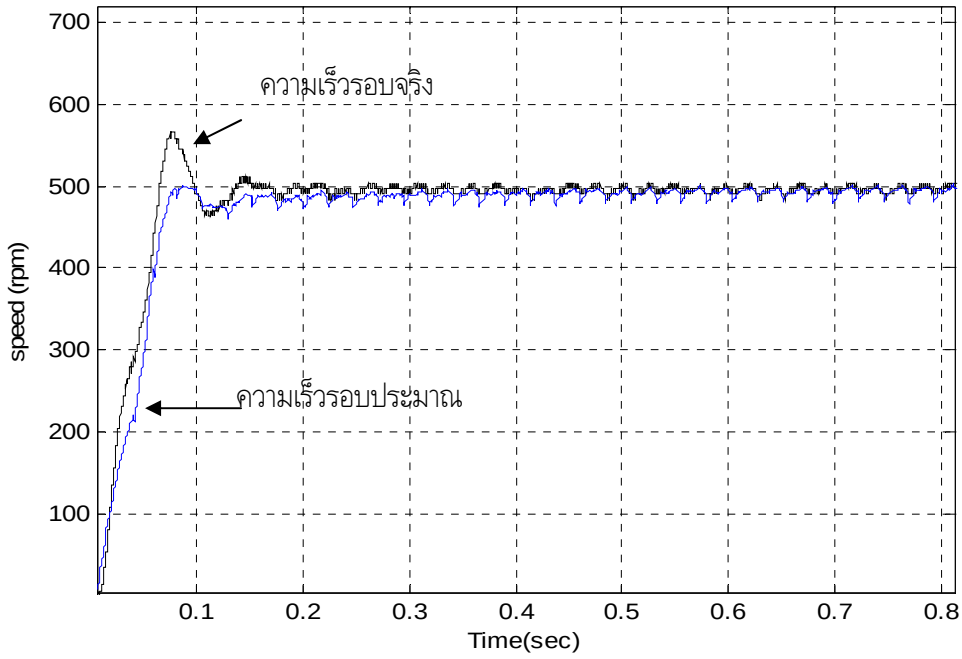


ภาพที่ 6-11 สเตเตอร์ฟลักซ์ของมอเตอร์เหนี่ยวนำ

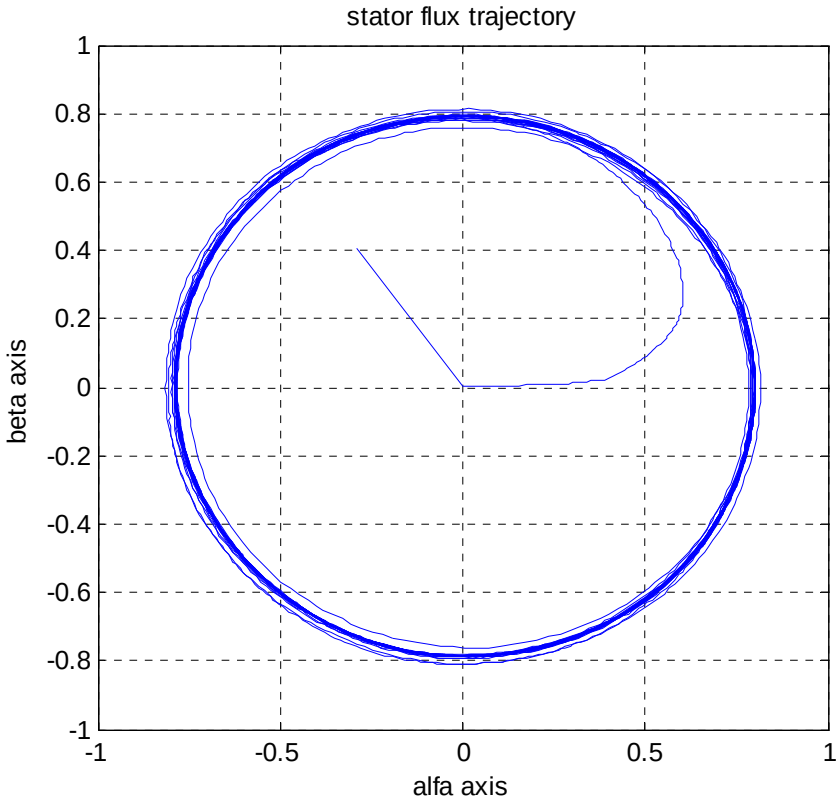


ภาพที่ 6-12 กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ

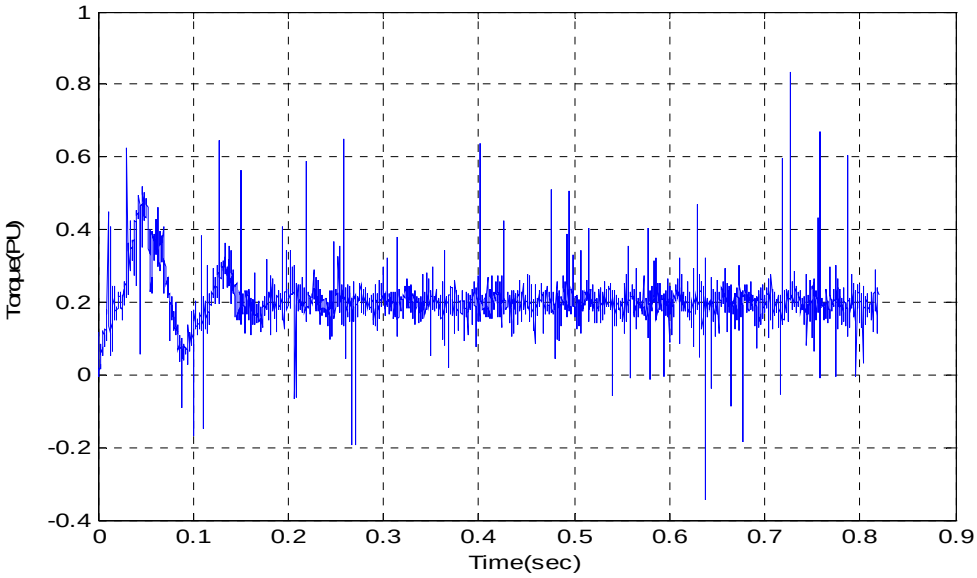
6.3.2 ผลทดสอบระบบควบคุมแบบวงรอบปิด ที่ความเร็วรอบอ้างอิง 500 รอบต่อนาทีระดับฟลักซ์อ้างอิงอยู่ที่ 0.8 ต่อหน่วย ใส่ภาระขนาด 0.5 เท่าของกระแสพิกัดมอเตอร์



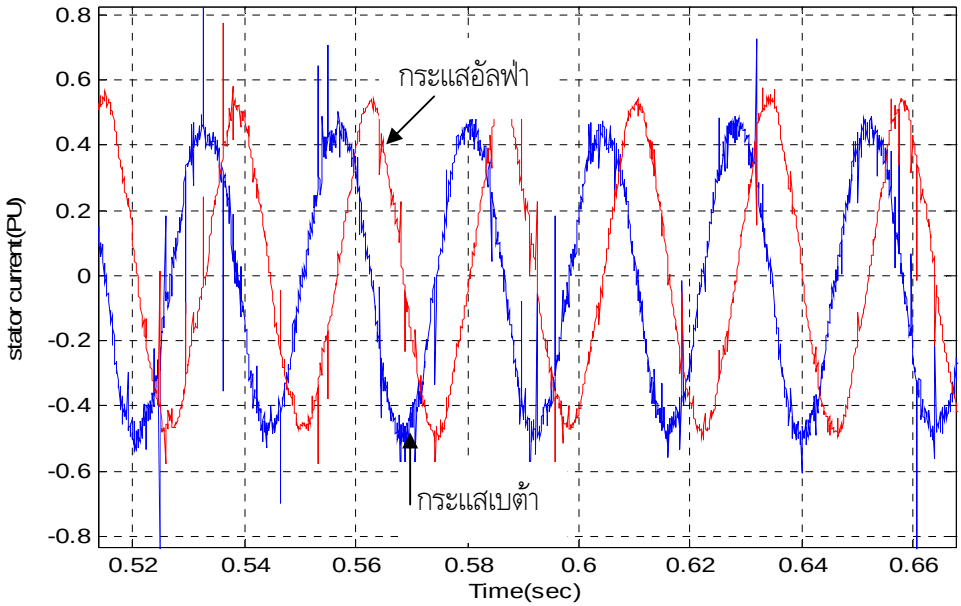
ภาพที่ 6-13 ความเร็วรอบจริงและความเร็วรอบประมาณของมอเตอร์เหนี่ยวนำ



ภาพที่ 6-14 ทางเดินของสเตเตอร์ฟลักซ์ของแกนอัลฟ่าและเบต้า



ภาพที่ 6-15 แรงบิดประมาณของมอเตอร์เหนี่ยวนำ



ภาพที่ 6-16 กระแสที่สเตเตอร์ของมอเตอร์เหนี่ยวนำ

จากภาพที่ 6-9 ถึง 6-16 พบว่าระบบสามารถชดเชยแรงบิดได้และความเร็วประมาณเทียบกับความเร็วจริงที่สภาวะคงตัวมีค่าใกล้เคียงกันมาก ส่วนโอเวอร์ชู้ตที่เกิดขึ้นเนื่องจากผลของการประมาณความเร็วที่ผิดพลาดในช่วงลาดเนื่องการชดเชยของตัวควบคุมพีไอของวงรอบปิดความเร็ว

6.4 วิเคราะห์และสรุปผลการทดลอง

การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ให้ผลทดสอบได้ดี ตัวชดเชยแบบ PI มีประโยชน์ในการลดการสั่นของตัวสไลดิงได้มาก อีกทั้งสามารถช่วยขจัดปัญหาสัญญาณรบกวนในบางส่วนที่เข้ามาปนกับสัญญาณกระแส

จากผลการทดสอบแบบปิดวงรอบพบว่าระบบมีพลวัตรค่อนข้างดี และในส่วนของตัวประมาณนั้นมีความแม่นยำและความเร็วในการประมาณค่อนข้างดี มีผลในการประมาณค่อนข้างแม่นยำและสามารถชดเชยปริมาณของฟลักซ์และกระแส ทำให้การส่งต่อไปประมาณค่าความเร็วรอบของมอเตอร์มีความแม่นยำขึ้น ถึงแม้การประมาณความเร็วจะเป็นแบบลูปเปิด และตัวสังเกตการณ์แบบสไลดิงและตัวชดเชยแบบ PI ทำงานได้ค่อนข้างดีเมื่อนำมาประกอบกับระบบควบคุมดังกล่าว ก็ให้ผลในเรื่องของความเร็วและความแม่นยำได้ค่อนข้างดี

บทที่ 7
สรุปผลและข้อเสนอแนะ

7.1 สรุปผลการวิจัย

งานวิทยานิพนธ์นี้ได้ทำการศึกษาและนำเสนอการการควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบโดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI ในส่วนของตัวควบคุมนั้นเป็นการควบคุมแบบโครงสร้างผันแปรร่วมกับสเปซเวกเตอร์พีดับลิวเอ็มซึ่งยังคงผลในเรื่องความราบรื่นที่สภาวะผลตอบสนองคงดีกว่าการใช้สวิทช์ของการควบคุมแรงบิดตรงดั้งเดิม เนื่องจากต้องการทำงานที่ราบรื่นที่สภาวะผลตอบสนองคงตัว มาประยุกต์ใช้งานรวมกับการควบคุมแบบโครงสร้างผันแปร ที่ให้ผลดีในเรื่องความเร็ว ความน่าเชื่อถือและความคงทน ต่อการเบี่ยงเบนค่าพารามิเตอร์ของมอเตอร์ และสิ่งรบกวนจากภายนอก และในส่วนของการประมาณแรงบิดและฟลักซ์นั้นได้ใช้ตัวสังเกตการณ์แบบสไลดิงโดยพิจารณาแบบจำลองของมอเตอร์ทั้งทางด้านสเตเตอร์และโรเตอร์ เพื่อผลในเรื่องของความแม่นยำและความถูกต้องของแบบจำลองของมอเตอร์และให้ได้ค่าการประมาณที่ใกล้เคียงความเป็นจริง อีกทั้งยังนำตัวชดเชยแบบ PI มาเพื่อชดเชยปริมาณฟลักซ์ และกระแสซึ่งถูกอินทิเกรตโดยแบบจำลองของมอเตอร์ โดยตัวชดเชยแบบ PI ยังสามารถช่วยลดการสั่นของตัวสไลดิงลงทำให้ปริมาณของฟลักซ์และแรงบิดที่ประมาณมีความราบเรียบขึ้น แล้วนำมาคำนวณหาความเร็วรอบโดยใช้การประมาณแบบลูปเปิด เพื่อนำมาประมาณค่าความเร็วรอบ ที่ความเร็วต่าง ๆ ในงานวิจัยนี้ใช้แบบจำลองมอเตอร์เหนี่ยวนำรูปแบบสเปซเวกเตอร์ จากการจำลองระบบด้วยคอมพิวเตอร์ แสดงให้เห็นว่าวิธีการนี้สามารถนำมาประมาณค่าต่างๆได้ และยังลดการสั่นของตัวสไลดิงและให้ผลการควบคุมมอเตอร์นั้นดีขึ้น

วิทยานิพนธ์นี้มีการทดลองการประยุกต์ใช้ตัวควบคุมประมวลผลสัญญาณแบบดิจิตอล ในการควบคุมมอเตอร์เหนี่ยวนำแบบควบคุมแรงบิดโดยตรง การออกแบบใช้วิธีการแบบดิจิตอลทั้งหมดโดยใช้ตัวประมวลผลสัญญาณแบบดิจิตอล TMS320F243 ชุดขับเคลื่อนมอเตอร์เหนี่ยวนำ 3 เฟสแรงดัน 600V และกระแส 18A ใช้ขับเคลื่อนมอเตอร์ขนาด 1.1 kW 2.6A 400V วัดความเร็วรอบด้วยเอ็นโคดเดอร์ขนาด 2048 พัลส์ต่อรอบ

ในส่วนของการทดลองนั้น ตัวประมาณค่าที่ใช้ในงานวิจัยนี้สามารถชดเชยปริมาณของฟลักซ์และกระแสที่ถูกอินทิเกรตโดยแบบจำลองของมอเตอร์ซึ่งสังเกตได้จากผลการทดลองจะเห็นว่าปริมาณฟลักซ์จะอยู่ที่ระดับอ้างอิงตลอด แม้ว่าในกรณีมีภาระซึ่งเมื่อปริมาณที่ได้มีค่าความถูกต้องเมื่อนำมาคำนวณในส่วนต่อไปทั้งกระแส หรือความเร็วรอบ ค่าที่ได้ก็จะมีความถูกต้องมากขึ้นอีกทั้งทำให้ปริมาณต่าง ๆ มีความเรียบมากขึ้นเพราะตัวชดเชยแบบ PI ที่ใส่เข้าไปนั้นสามารถลดการสั่นของตัวสไลดิ้งลดลงสังเกตได้จากการ

Simulation โดยระนาบของการสไลดิงมีความถี่น้อยลง ส่วนในการทดลองนั้นปริมาณของแรงบิดความเร็วรอบมีความเรียบมากขึ้น ในขณะทดลองมอเตอร์มีการหมุนที่ราบเรียบขึ้นไม่มีอาการสั่นมากนัก และในเรื่องเสถียรภาพของระบบที่ใช้ในงานวิจัยแสดงให้เห็นว่าระบบไวค่อนข้างมาก

7.2 ข้อเสนอแนะ

7.2.1 จากผลการทดลอง จะเห็นว่าที่ความเร็วต่ำ จะมีค่าผิดพลาดในการประมาณมากขึ้น ดังนั้นควรทำการวิจัยหาวิธีแก้ไขปรับปรุงให้ดีขึ้น เช่นการเพิ่มตัวประมาณความต้านทานด้านโรเตอร์

7.2.1 เนื่องจากโปรแกรมภาษาแอสเซมบลีที่ใช้ในงานวิจัยนี้ปริมาณตัวแปรต่างๆ ยังมีรูปแบบการคำนวณที่ยังไม่ค่อยละเอียดนัก ดังนั้นการเปลี่ยนรูปแบบอาจให้ผลที่ละเอียดกว่าในการ คำนวณ โดยเฉพาะช่วงที่ค่าเปลี่ยนแปลงน้อยๆ เช่นความเร็วต่ำ การคูณ หาร จะให้ผลเดิมเป็นเวลานาน

7.2.3 ทดลองเปลี่ยนวิธีการใช้แบบจำลองทีอ้างอิงทางด้านโรเตอร์จะเป็นการควบคุมที่ถูกต้องมากขึ้น เพราะจะทำให้ได้การควบคุมที่เพลาของมอเตอร์จริง

7.2.4 พัฒนาให้เป็นระบบแบบปรับตัวเองได้ หรือใช้การควบคุมแบบอัจฉริยะเข้ามาใช้ในการปรับค่าของตัวโปรแกรมเนื่องจากยังมีค่าพารามิเตอร์หลายตัวที่ต้องปรับในขณะทดลอง

เอกสารอ้างอิง

1. Vas, P. Sensorless Vector and Direct Torque Control. Oxford : Oxford University Press, 1998.
2. Utkin, V. Yan, Z. and Shi, J. "Sensorless sliding-mode control of induction motors." IEEE Trans. Ind. Electron. (December 2000) : 1286-1297.
3. Casadei, D. G. Serra, and Tani, A. "Implementation of direct torque control algorithm for induction motors based on discrete space vector modulation." IEEE Trans. Ind. Electron. (July 2000) : 769-777.
4. Lascu, C. Boldea, I. and Blaabjerg, F. "Variable - structure direct torque control –A class of fast and robust controllers for induction machine drive." IEEE Trans. Ind. Electron. (August 2004) : 785-792.
5. Krause, P. C. Wasynczuk, O. and Sudhoff, S. D. Analysis of Electrical Machinery. New York : IEEE Press, 1995.
6. Cristian Lascu and Ghe rghe-Daniel Andreescu. “Sliding-Mode Observer and Improved Integrator With DC-Offset Compensation for Flux Extimation in Sensorless-Controlled Induction Motor.” IEEE Trans. Ind.Applicat. 53 (June 2006).
7. Benchaib, A. Rachid, A. and Tadjine, M. "Real–time sliding-mode observer and control of an induction motor." IEEE Trans. Ind. Electron. 46 (February 1999) : 128-137.
8. Bolognani, S. Zigliotto, M. and Zordan, M. "Extended-range PMSM sensorless speed drive based on stochastic filtering." IEEE Trans. Pow. Elect. 16 (January 2001) : 110 -117.
9. Bolognani, S. Tubiana, L. Zigliotto, M. "Extended Kalman filter tuning in sensorless PMSM drives." 2002. PCC-Osaka 2002. Power Conversion Conference. 1 (2002) : 276-281.
10. Shi, K. L. Chan, T. F. Wong, Y. K. and Ho, S. L. "Speed Estimation of an Induction Motor Drive Using an Optimized Extended Kalman Filter." IEEE Trans. Ind. Elect. 49 (February 2002) : 124-133.

ภาคผนวก ก

ข้อมูลมอเตอร์เหนี่ยวนำ และค่าฐาน

ตารางที่ ก-1 ข้อมูลของมอเตอร์เหนี่ยวนำ

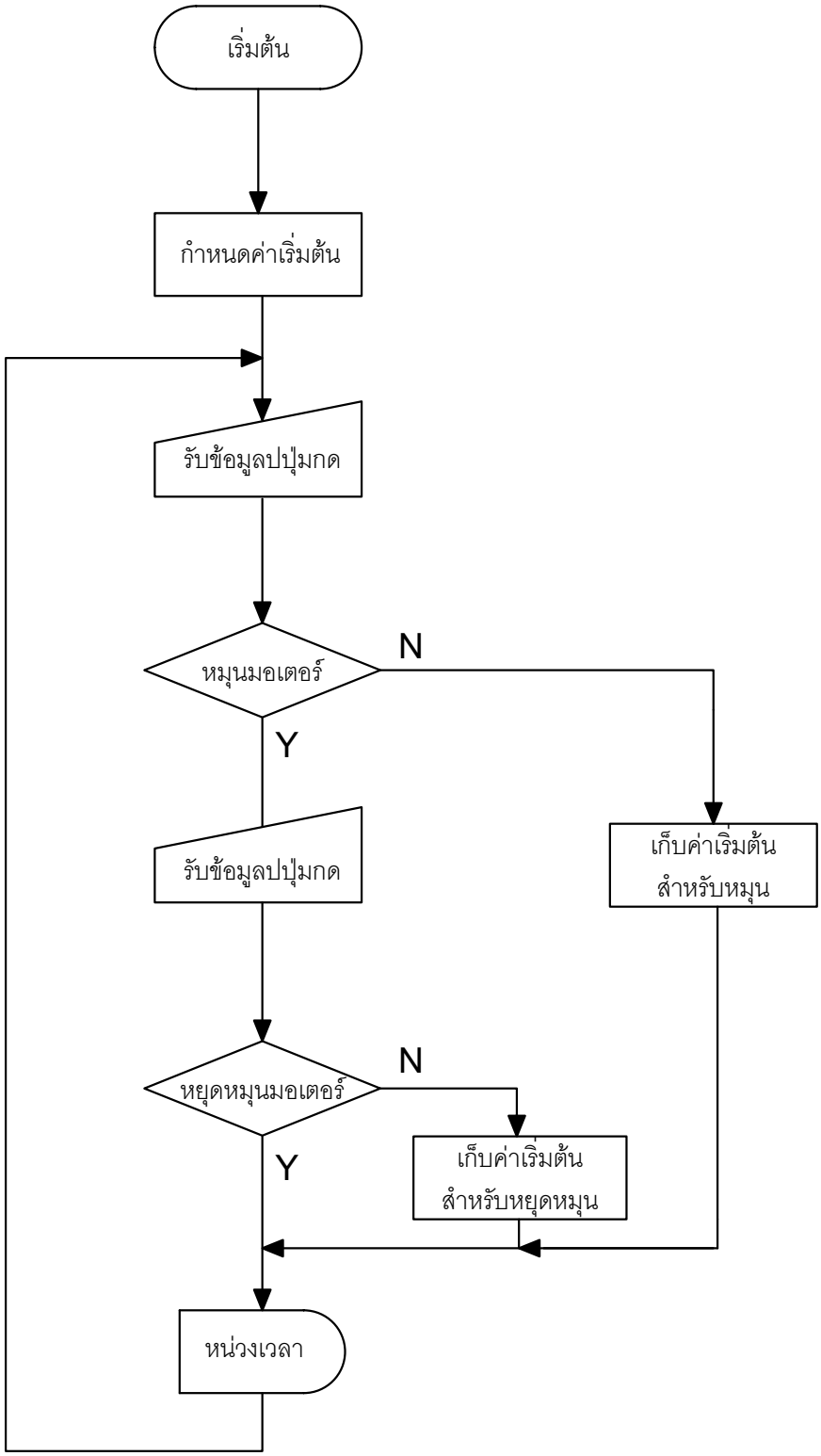
พิกัดกำลัง	1.1	kW
แรงดัน (u_s)	400/690	V
กระแส (i_s)	2.6/1.49	A
ตัวประกอบกำลัง	0.83	
พิกัดความเร็ว (ω_r)	1400	rpm
จำนวนคู่ของขั้วแม่เหล็ก (P)	2	คู่
ความต้านทานสเตเตอร์ (R_s)	26.3975	Ω /phase
ความต้านทานโรเตอร์ (R_r)	11.2708	Ω /phase
ความเหนี่ยวนำตัวเองด้านสเตเตอร์ (L_s)	1.3910	H/phase
ความเหนี่ยวนำตัวเองด้านโรเตอร์ (L_r)	1.3496	H/phase
ความเหนี่ยวนำของสนามแม่เหล็ก (L_m)	1.2866	H
ความเฉื่อยของโรเตอร์ (J_m)	0.0016	$kg \cdot m^2$

ตารางที่ ก-2 ค่าฐาน

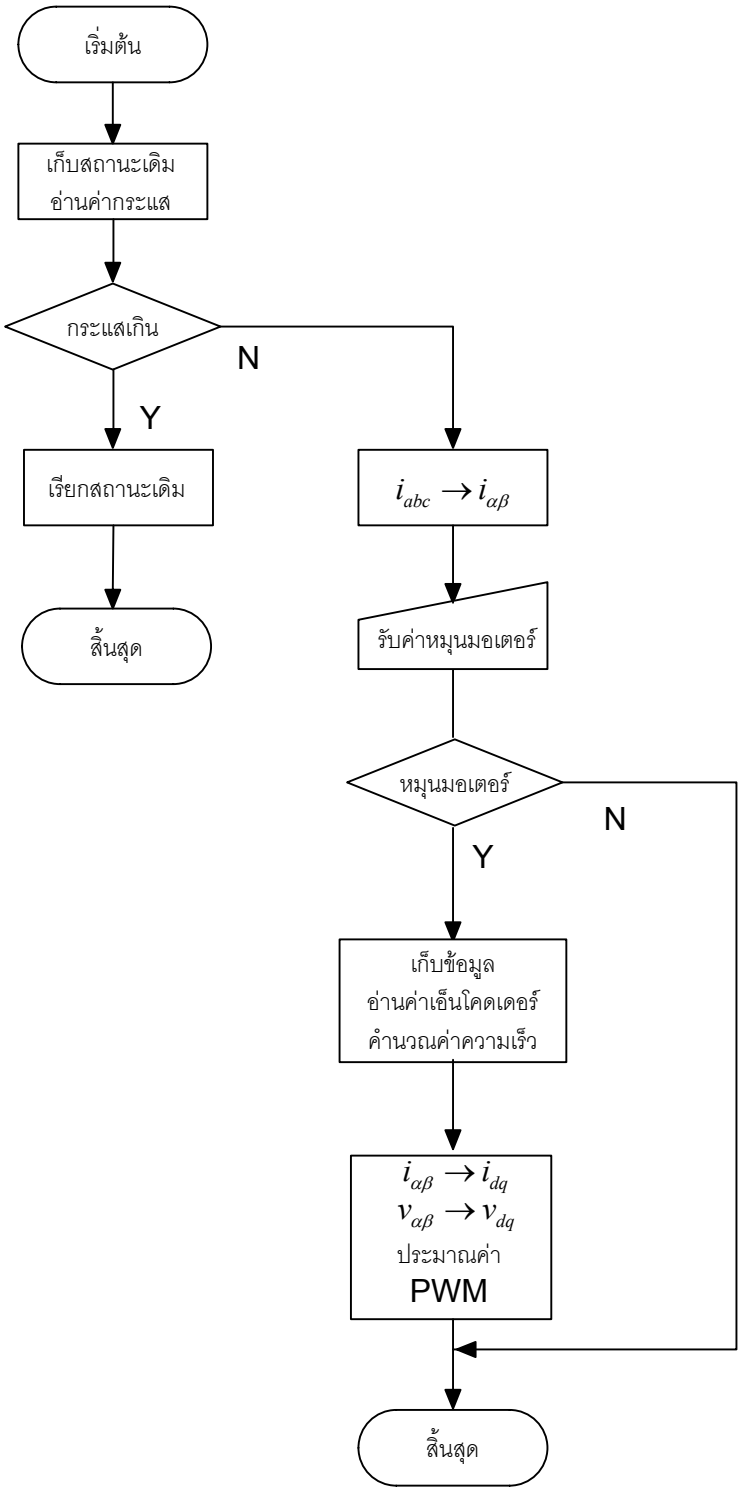
กระแส (I_b)	$\sqrt{2} * i_s$	2.1229 A
แรงดัน (U_b)	$\sqrt{2} * u_s$	326.60 Volt
ความต้านทาน (Z_b)	U_b / I_b	153.846 Ω
ความเร็วเชิงมุม (ω_b)	$2\pi f_b$	314.1592 rad/s
ความเร็วรอบ (N_b)	N_b	1400 rpm
ฟลักซ์แม่เหล็ก (ψ_b)	U_b / ω_b	1.0395 Wb
ความเหนี่ยวนำ (L_b)	ψ_b / I_b	489.66 mH
ความถี่ไฟฟ้า (f_b)	f_b	50 Hz
แรงบิด T_b	$\left(\frac{3}{2} * U_b * I_b\right) * \left(\frac{P}{2 * \omega_b}\right)$	7.0 N.m

ภาคผนวก ข

ลำดับการทำงานของโปรแกรมที่ใช้ในงานวิจัย



ภาพที่ ข-1 ลำดับการทำงานของโปรแกรมหลัก



ภาพที่ ข-2 ลำดับการทำงานของโปรแกรมย่อย

ภาคผนวก ค

โปรแกรมที่ใช้ในงานวิจัย

```
*****closedloop DTC and observer**********
***********i compensate*****************
*********** on 4/4/50 ******************
*****************************************
* Auxiliary Register used *
* ar4 pointer for context save stack *
* ar5 used as general purpose table pointer *
*********************************************
****************************************************************
* Variables and constants initializations
****************************************************************
********************  Base value  *************************
**  Rs=26.3975 ohm,Ls= 1.3910 H ,Rr'=11.2708 ohm,
**  Lr'=1.3496 H, Lm=1,2866 H
**  pole pairs =2 , J=0.0016 Kg-m^2,speed=1400 rpm
**  Name plate motor :
**  400 V (Delta) , 2.6 A ,f=50 Hz , Nbase motor =1400 rpm
**  PF=0.83 ,P=1.1 KW
**  Encoder 2048 P/R :supply 5 Vdc
**  Vb  = sqrt(2) *400 / sqrt(3) = 326.5986 V (Vphase max)
**  Ib   = sqrt(2) * 1.49 =2.107 A
**  Zb  =  Vb / Ib = 155 Ohm
**  fb   = 50 Hz
**  Lb=flux base(Wb)= Lm*Ib        = 1.2866*3.6769=4.7307 (volt.sec/rad)
**  wb  = 2 * pi * fb = 314.1592 rad/sec
**  Fb = Vb / wb = 1.0395   volt.sec/rad :flux base
**  It is replaced by Lm*Ib=Fb (flux base)
**  Lb = Fb/Ib  :Inductance base
**  tb  = 2*pi / wb = 0.02 Sec.        ; time base
**  pb=P/2 ; number of pole pairs
**  Tb = (3*Vb*Ib/2)*(pb/(wb)) = 11.44 ; Base torque N.m
**  Sb =(3/2)*Ub*Ib : Power base
**  SPb=120*fb/P = 120*50/4=1500 ; Base synchronous speed (rpm)
*********************************************
.include "24x.h"
.global _c_int0
.sect "vectors"
b        _c_int0 ;reset vector
_c_int1 b       _c_int1
b        _c_int2 ;PWM interrupt handler
stack            .usect "blockb2",15 ;space for ISR indirect context save
dac_val          .usect "blockb2",5 ;space for dac values in Page 0
        .ref ATANTAB_45
****************************************************************
* Look-up tables .includes
* N.B. all tables include 256 elements
****************************************************************
                .sect "table"
sintab          .include sine.tab
;sine wave look-up table for sine and cosine waves generation
;4.12 format
****************************************************************
* Variables and constants initializations
****************************************************************
                .data
GPR0 .word 0          ; Gen purp reg
_T_sampling .set     0405h ;         1.15              0.1msec / tbase (1/50)
*** PWM modulation constants
.bss    _v_meas,1
PWMPRD     .set 1000       ;PWM Period=2*600 -> Tc=2*1000*50ns=60us
(50ns
                                        ; resolution)
Tonmax          .set 0            ;minimum PWM duty cycle
MAXDUTY .set PWMPRD-2*Tonmax ;maximum utilization of the inverter
K1_se_          .set     077E5h                    ; for K1_se (Q15)
K2_se_          .set     03200h                ; for K2_se (Q6)
K3_se_          .set     07Faeh; 07F99h              ;7fbe  for K3_se (Q15)
K4_se_          .set     051h   ; 066h     ; for K4_se (Q15)
Klow_ .set     07F20h      ;7d7d  low pass stator flux
DIFF_MAX_LIMIT   .set   0FDDh   ; The maximum limit to differentiate
theta_r_se (Q15)
DIFF_MIN_LIMIT   .set   023h   ; The minimum limit to differentiate
theta_r_se (Q15)
_Kcurrent       .set     05541h          ;    current scaling Q10
_Rs             .set     015CCh                  ; .17 pu Q15
_LrLm .set     010C8h                ;  4.12    1.09 P.U
_L2qLm          .set     0592h           ;  Q12       0.349 pu
_LrL2q          .set     03000h                  ; Lr*Zb/L2q*Wb  Q12    3 . P.U
_LmL2q          .set     02DC2h                  ;  Lm*Zb/L2q*Wb  Q12   2.8 pu
_KR11           .set    01A9Eh      ; 0.2 pu Q15
_KR12           .set    01CC7h      ; 0.22.. pu Q15
_K2d  .set     0a0h             ; -0.023 Q12  kslide
_K2q  .set     0A4h             ;  0.1        kslide
```

```
_Kslip .set    0941h;0941h     ;011EAh      ;Q15 Rr/(Zb)
_K1flux        .set    07FCEh    ;for low-pass flux
_K2flux        .set    041h    ;for low pass flux
OVCTMP     .set    02000h     ; 1.8times,Ibase 4.12
_Ks11 .set    030h            ;D  Q12 format xxx for sliding mode stator
fluxSD
_Ks11i         .set    06h             ;Q  Q12 format xxx for sliding mode
stator fluxSD
_Ks12 .set    030h
_Ks12i         .set    06h
FMIN  .word  05h             ;0.01
FMAX .word  0555h ; eefh=1400rpm,999h        ;0.5 p.u
ktheta .word  051EBh;       ;50*360/10k=1.08    ,6.10format   ;
kvpf    .word  05333h     ; 0.65 Q15

************************************************************
VKencoder    .set 06000h ;this constant is needed only with
                        ;encoder !
                        ;it is used to convert encoder pulses [0;8192] to an
                        ;electric angle [0;360]=[0000h;1000h]
                        ;2.14 format unsigned = 1.5 (see "Theta calculation"
block
                        ;for details)
Encpulses     .set 8192 ;this constant is needed only with encoder !
                                    ;number of encoder pulses per round
(mechanical)
_Kspeed .set 06D1h
SPEEDSTEP          .set 10
_SPEEDscale        .set    05DCh          ; invers lowpass Q14
**       speed rad/sec multiply by  2BC0h Q15 to n rpm(1400 base)
*********************************************
******** control law generation part ********
*********************************************

_FluxREF     .set    0ccch; Q12 0.8 perunit
*_TorqeREF  .set    04000h ;Q15 0.5 P.U t
*bandTorque .set    0.2;
*bandFluxs   .set    0.04;
_c_slidF       .set    02a7h;1.15     0.01
_c_slidTe      .set    02a7h;  1.15    0.01
*_Kpslid_F    .set    0580h; 8.8    5.5
*_Kpslid_T    .set    0300h;  8.8   3
```

```
*_KIslid_F     .set    06400h;   8.8
*_KIslid_T     .set    06400h;
_Te_max              .set    01800h; .12            1.5 pu torqmax limit
_Te_min          .set    0E800h ; Q12 -1.5
VKiTe          .set 015h ;  14 0.0125 7ah ;4.12 format = 0.03
VKpiTe         .set 0cFEh ;15fe 14 ;6.10 format = 6.5
VKcorTe        .set 26;
VKiflux        .set 035h ;11 26 0.04 7ah ;4.12 format = 0.03
VKpiflux       .set 0FA0h ;ea0 ;.12 format = 6.5
VKcorflux      .set 54          ;
VKispeed       .set 070h       ;60 85 0.04 7ah ;4.12 format = 0.03
VKpispeed    .set 02500h ;7800 ;6.10 format = 6.5
VKcorspeed  .set 13          ; 8 0.00285 4.12 format = 0.033
_USqs_min   .set 0E800h   ;0.8
_USds_min   .set 0F1C3h    ; -0.5 PU     Q12
_USqs_max  .set 01800h    ;  0.3 Q12
_USds_max   .set 0E3Dh      ;0.8
*** axis transformation constants
_VDCinvTc    .set 0233h  ;  1000/(580/Vbase)
SQRT3inv .word 049e6h ;1/SQRT(3) 1.15 format
SQRT32       .word 0ddbh ;SQRT(3)/2 4.12 format
.bss    t1,1 ;SVPWM T1 (see SV PWM references for details)
.bss    t2,1 ;SVPWM T2 (see SV PWM references for details)
.bss    taon,1 ;PWM commutation instant phase 1
.bss    tbon,1 ;PWM commutation instant phase 2
.bss    tcon,1 ;PWM commutation instant phase 3
.bss    X,1 ;SVPWM variable
.bss    Y,1 ;SVPWM variable
.bss    Z,1 ;SVPWM variable
.bss    sectordisp,1 ;SVPWM sector for display
.bss    sector,1 ;SVPWM sector
.bss    VDCinvT,1 ;used in SVPWM
.bss    K2_se,1
.bss    K3_se,1
.bss    K4_se,1
.bss    Klow,1
.bss    Ks11,1
.bss    Kslip,1
.bss    ovcurrent,1 ;over current
.bss    fovcurrent,1 ;flag over current
.bss    tmpovcurrent,1 ; temp for check over current
.bss    max_theta,1
```

```
.bss    min_theta,1
.bss    stop,1
.bss    initphase,1
*           .bss        n,1 ;speed
*                .bss   n_ref,1 ;speed reference
.bss    n_es,1   ; estimate speed rpm
.bss    epispeed,1 ;speed error (used in speed regulator)
.bss    tmp,1
.bss    tmp1,1
.bss    tmp2,1
.bss    temp,1
.bss    tmp_sv,1
.bss    fre,1 ;frequency
.bss    Voltmodu,1
.bss    theta,1
.bss      seno1,1 ;generated sine wave value
.bss      coseno,1 ;generated cosine wave value
.bss      indice1,1 ;pointer used to access sine look-up table
.bss      Va,1 ;Phase 1 voltage for sector calculation
.bss      Vb,1 ;Phase 2 voltage for sector calculation
.bss      Vc,1 ;Phase 3 voltage for sector calculation
.bss      Valfar,1 ;alfa-axis reference voltage
.bss      Vbetar,1 ;beta-axis reference voltage
.bss      tangent,1
.bss      accb,2 ;2 words buffer
.bss      ialfa,1
.bss      ibeta,1
.bss      ialfae,1   ; estimate current
.bss      ibetae,1
.bss      start_stop,1 ;bit check for start or stop
.bss      tmp_start,1 ;use for double push button
.bss      index,1
.bss      T_sampling,1
.bss      FluxS_mag,1
.bss      FluxS_mag_old,1
.bss      FluxS_speed,1
.bss      FluxS_speed_old,1
.bss      FluxSD,1
.bss      FluxSQ,1
.bss      FluxSdr,1
.bss      theta_fluxS,1
.bss      theta_fluxS_old,1
.bss      sin_fluxS,1
.bss      cos_fluxS,1
.bss      temp1,1
.bss      temp2,1
.bss      temp3,1
.bss      speed_es,1
.bss      speed_es_old,1
.bss      Te,1
.bss      W_slip,1
.bss      Rs,1
.bss    LrLm,1
.bss    L2qLm,1
.bss    LrL2q,1
.bss    LmL2q,1
.bss    KR11,1
.bss    KR12,1
.bss    Kr,1
.bss    FluxRD,1
.bss    FluxRQ,1
.bss    FluxRdr,1
.bss    theta_fluxR,1
.bss    theta_fluxR_old,1
.bss    sin_fluxR,1
.bss    cos_fluxR,1
.bss    FluxR_speed,1
*               .bss    speed_es,1 ;estimate speed
*               .bss    test,1
.bss    TorqeREF_min,1
.bss    TorqeREF_max,1
.bss    USqs_min,1
.bss    USqs_max,1
.bss    USds_min,1
.bss    USds_max,1
.bss    TorqeREF,1
.bss    FluxREF,1
.bss    Sflux,1
.bss    STe,1
.bss    Cflux,1
.bss    CTe,1
.bss    er_flux,1
.bss    er_Te,1
.bss    KiTe,1 ;Cf explanation given above
```

```
.bss    KpiTe,1 ;Cf explanation given above
.bss    KcorTe,1 ;Cf explanation given above
.bss    Kiflux,1 ;Cf explanation given above
.bss    Kpiflux,1 ;Cf explanation given above
.bss    Kcorflux,1
.bss    Kispeed,1 ;Cf explanation given above
.bss    Kpispeed,1 ;Cf explanation given above
.bss    Kcorspeed,1 ;Cf explanation given above
.bss    USds,1
.bss    USqs,1
.bss    upi,1 ;PI regulators (current and speed) output
.bss    elpi,1 ;PI regulators (current and speed) limitation
.bss    xispeed,1 ;speed regulator integral component
.bss    epiTe,1
.bss    epiflux,1
.bss    xiTe,1
.bss    xiflux,1
.bss    Vsqs,1
.bss    count,1
.bss    i1,1
.bss    i2,1
.bss    i3,1
.bss    i2_old,1
.bss    i3_old,1
.bss    erisd,1
.bss    erisq,1
.bss    dsign,1
.bss    qsign,1
.bss    eiSdr,1
.bss    eiSqr,1
.bss    K2d,1
.bss    K2q,1
.bss    Kcurrent,1
.bss    encoderold,1 ;encoder pulses value stored in the previous
sampling;period
.bss    encincr,1 ;encoder pulses increment between two consecutive
;sampling periods
.bss    speedtmp,1 ;used to accumulate encoder pulses increments (to
                   ;calculate the speed each speed sampling period)
.bss    speedstep,1 ;sampling periods down counter used to define speed
.bss    encoder,1
.bss    theta_m,1
.bss    encpulses,1
.bss    Kencoder,1
.bss    Kspeed,1
.bss    theta_e,1
.bss    n,1
.bss    speed_scale,1
.bss    FluxRD_old,1
.bss    FluxRQ_old,1
.bss    K1flux,1        ;       0745Dh
.bss    K2flux,1        ;   02836h
.bss    Ks11i,1
.bss    X11,1
.bss    X12,1
.bss    Ks12i,1
.bss    Ks12,1
.text
*****************************************************************
* Initialisation Module *
*****************************************************************
_c_int0:
************************
* Board general settings
************************
SETC INTM               ;Disable interrupts
SPLK #0h, IMR           ;Mask all Ints
SPLK #0FFh, IFR         ;Clear all Int Flags
setc    SXM                     ;set Sign Extension Mode
setc    OVM                     ;set Overflow Mode
CLRC CNF                        ;Config Block B0 to Data mem.
spm             0
*****************************************************************
* Function to disable the watchdog timer *
*****************************************************************
LDP             #00E0h          ;(E0=224)(E0*80=7000) dp-->7000h-
707fh
SPLK #006Fh, WDCR       ;Disable WD if VCCP=5V (706F)
SPLK #05555h, WDKEY
SPLK #0AAAAh, WDKEY
SPLK #006Fh, WDCR
lacc    #00h
ldp     #tmp
sacl    tmp
```

```
OUT         tmp,WSGR   ;Program Space, 0–7 wait states
; Clear All EV Registers
zac
ldp    #DP_EV
sacl   GPTCON
sacl   T1CNT
sacl   T1CMPR
sacl   T1PR
sacl   T1CON
sacl   T2CNT
sacl   T2CMPR
sacl   T2PR
sacl   T2CON
sacl   COMCON
sacl   ACTR
sacl   DBTCON
sacl   CMPR1
sacl   CMPR2
sacl   CMPR3
sacl   CAPCON
sacl   CAPFIFO
sacl   CAP1FIFO
sacl   CAP2FIFO
sacl   CAP3FIFO
; Initialise PWM ;1us software dead-band
;Nicol activation
splk   #0999h,ACTR ;Bits 15-12 not used, no space vector
                   ;PWM compare actions
                   ;PWM5/PWM6 - Active Low/Active High
                   ;PWM3/PWM4 - Active Low/Active High
                   ;PWM1/PWM2 - Active Low/Active High
splk   #2000,CMPR1
splk   #2500,CMPR2
splk   #3000,CMPR3
splk   #0cech,DBTCON ;4.8us dead band
splk   #0207h,COMCON ;FIRST enable PWM operation
       ;Reload Full Compare when T1CNT=0
       ;Disable Space Vector
       ;Reload Full Compare Action when T1CNT=0
       ;Enable Full Compare Outputs
       ;Disable Simple Compare Outputs
       ;Full Compare Units in PWM Mode
splk   #8207h,COMCON ;THEN enable Compare operation
splk   #PWMPRD,T1PR ;Set T1 period
splk   #0,T1CNT
splk   #0a800h,T1CON ;Ignore Emulation suspend
                     ;Cont Up/Down Mode
                     ;x/1 prescalar
                     ;Use own TENABLE
                     ;Disable Timer,enable later
                     ;Internal Clock Source
                     ;Reload Compare Register when T1CNT=0
                     ;Disable Timer Compare operation
                     ; Enable Timer 1
 lacc  T1CON
or          #40h
sacl   T1CON
****** configure I/O ***************
ldp         #0e1h          ;e1=225(e1*80=7080)
splk   #0000111111011011b,OCRA          ;IOPA,B0-7 as I/O
splk   #0000001100000011b,OCRB          ;IOPC0-7 as I/O
splk   #0ff00h,PADATDIR ;1 is output/0 is input send IOPA5 hi turn off
transistor
splk   #01010h,PBDATDIR        ;and forc high
splk   #0ff00h,PCDATDIR
splk   #09999h,PDDATDIR   ;send IOPD4 low for enable 74244
************************************
*** END: PWM enable
*********************************************************************
* Incremental encoder initialization
* Capture for Incremental encoder correction with Xint2
*********************************************************************
ldp    #DP_EV
splk   #0000h,T2CNT ;configure counter register
splk   #0FFFFh,T2PR ;configure period register
splk   #0F870h,T2CON ;configure for QEP and enable Timer T2
splk   #0E0F0h,CAPCON ;T2 is selected as Time base for QEP
*** END encoder/capture initialization
*********************************************************************
* A/D initialization
*********************************************************************
ldp    #DP_PF1
*           splk   #000bh,ADCTRL2;prescaler set for a 10MHz SYSCLK
lacc   ADCFIFO1 ;empty FIFO
```

```
lacc    ADCFIFO1
lacc    ADCFIFO2
lacc    ADCFIFO2
SPLK #0000000000000000b,ADCTRL2
                ; ||||||||||||||||
                ; FEDCBA9876543210
              ; bit 0-2 000 Prescaler value
              ; bit 3-4 FIFO2 status
              ; bit 5 Reserved
              ; bit 6-7 FIFO1 status
              ; bit 8 Reserved
              ; bit 9 0 Mask external SOC input
              ; bit A 0 Mask EV SOC input
              ; bit B 0
              ; bit C 1
              ; bit D 1
              ; bit E 0
              ; F Reserved

SPLK #1001100000010000b,ADCTRL1 ;||||||||||||||||
              ;FEDCBA9876543210
              ; bit 0 1 Start of conversion
              ; bit 1-3 000 Channel 0 addres adc1
              ; bit 4-6 001 Channel 1 address adc2
              ; bit 7 0 End of convert
              ; bit 8 1 Interrupt flag  write 1 to clear
              ; bit 9 1 Interrupt mask  enable with 1, mask 0
              ; bit A 0 Continuous run mode disabled
              ; bit B 1 Enable ADC2
              ; bit C 1 Enable ADC1
              ; bit D 1 Immediate start  0 = no action
              ; bit E 1 Free run  ignore suspend
              ; bit F 1 Soft  Not applicable with bit E = 1
*** END A/D initialization
*****************************************************************
* Initialize ar4 as the stack for context save
* space reserved: DARAM B2 60h-80h (page 0)
*****************************************************************
lar     ar4,#79h
lar     ar5,#60h
lar     ar7,#0e000h
```

```
*************************************************************
******* initialization ************************************
ldp             #FMIN
lacc    FMIN
sacl    fre
*               splk    #0800h,FluxSD
*               splk    #0800h,FluxSD_old
zac
sacl    FluxSD
sacl    FluxSQ
sacl    FluxS_speed
sacl    FluxS_mag
sacl    FluxS_speed_old
sacl    FluxS_mag_old
sacl    FluxSdr
sacl    theta_fluxS
sacl    theta_fluxS_old
sacl    sin_fluxS
sacl    cos_fluxS
sacl    FluxRD
sacl    FluxRQ
sacl    FluxRdr
sacl    theta_fluxR
sacl    theta_fluxR_old
sacl    sin_fluxR
sacl    cos_fluxR
sacl    FluxR_speed
sacl    ialfa
sacl    ibeta
sacl    ialfae
sacl    ibetae
sacl    upi ;PI regulators (current and speed) output
sacl    elpi ;PI regulators (current and speed) limitation
        ;error      .bss epispeed,1 ;speed error (used in speed regulator)
sacl    xispeed ;speed regulator integral component
sacl    epiTe
sacl    epiflux
sacl    xiTe
sacl    xiflux
sacl    Te
sacl    TorqeREF
sacl    Valfar
```

```
sacl    Vbetar
sacl    theta
sacl    index
sacl    speed_es
sacl    speed_es_old
sacl    W_slip
sacl    n_es
sacl    temp1
sacl    temp2
sacl    temp3
;               splk    #01000h,cos_fluxS
ldp     #LrLm
splk    #_LrLm,LrLm
splk    #_L2qLm,L2qLm
splk    #_LrL2q,LrL2q
splk    #_LmL2q,LmL2q
splk    #_KR11,KR11
splk    #_KR12,KR12
*               LDP             #K1_se
*       SPLK #K1_se_,K1_se      ; K1 = 1/(Wb*Tr)            (Q15)
SPLK #K2_se_,K2_se      ; K2 = 1/(fb*T)             (Q6)
SPLK #K3_se_,K3_se   ; K3 = Tau/(Tau+T)          (Q15)
SPLK #K4_se_,K4_se   ; K4 = T/(Tau+T)
splk    #Klow_,Klow     ;       (Q15)
splk    #_Ks11,Ks11  ;
splk    #_Kslip,Kslip
splk    #VKiflux, Kiflux
splk    #VKpiflux, Kpiflux
splk    #VKcorflux, Kcorflux
splk    #VKispeed, Kispeed
splk    #VKpispeed,Kpispeed
splk    #VKcorspeed,Kcorspeed
splk    #VKiTe, KiTe
splk    #VKpiTe,KpiTe
splk    #VKcorTe,KcorTe
splk    #_c_slidF,Cflux
splk    #_c_slidTe,CTe
splk    #_FluxREF,FluxREF
splk    #_Te_min,TorqeREF_min
splk    #_Te_max,TorqeREF_max
splk    #_USqs_min,USqs_min
splk    #_USqs_max,USqs_max
splk    #_USds_min,USds_min
splk    #_USds_max,USds_max
splk    #OVCTMP,ovcurrent
SPLK #DIFF_MAX_LIMIT,max_theta   ; Initialize the max limit of rotor flux
angle (Q15)
SPLK #DIFF_MIN_LIMIT,min_theta ; Initialize the min limit of rotor flux
angle (Q15)
splk    #_Rs,Rs
splk    #_VDCinvTc,VDCinvT
splk    #_T_sampling,T_sampling
*               splk    #100h,vSal_ref ;alfa-axis reference voltage
*               splk    #100h,vSbe_ref ;beta-axis reference voltage
*               ldp             #start_stop
*               splk    #0,start_stop
*               splk    #0,tmp_start
*               ldp             #kvpf
lacl    #0Eh
sacl    Voltmodu
zac
ldp             #i1
sacl    erisd
sacl    erisq
sacl    i1
sacl    i2
sacl    i3
sacl    i2_old
sacl    i3_old
sacl    n
sacl    theta_m
sacl    theta_e
sacl    encincr
sacl    FluxRD_old
sacl    FluxRQ_old
sacl    speedtmp
sacl    X11
sacl    X12
splk    #01h,count
splk    #Encpulses,encpulses
splk    #VKencoder,Kencoder
splk    #_Kspeed,Kspeed
splk    #SPEEDSTEP,speedstep
splk    #_K2d,K2d
```

```
splk    #_K2q,K2q
splk    #_Kcurrent,Kcurrent
splk    #_SPEEDscale,speed_scale
splk    #_K1flux,K1flux
splk    #_K2flux,K2flux
splk    #_Ks11i,Ks11i  ;
splk    #_Ks12,Ks12
splk    #_Ks12i,Ks12i
*****************************************************************
* Enable PWM control Interrupt
*****************************************************************
; Clear EV IFR and IMR regs
ldp             #EVIFRA>>7
splk    #07FFh,EVIFRA
splk    #0FFh,EVIFRB
splk    #000Fh,EVIFRC
; Enable T1 Underflow Int
splk    #0201h,EVIMRA
splk    #0000h,EVIMRB
splk    #0000h,EVIMRC
;Set IMR for INT2
ldp     #0h
lacc    IFR
sacl    IFR ;clear interrupt flags
lacc    #0000010b
sacl    IMR
clrc    INTM ;enable all interrupts
splk    #10,tmp_sv
*****************************************************************
* Table initialization
*****************************************************************
*               mar     *,AR5
*               lar     AR5,#sintab
*               rpt     #255
*               blpd    #sintab,*+
***************
* Virtual Menu
***************
Menu
*************************************************
* run ?
*************************************************
```

```
LDP             #0E1h ;read data from IOPB7
bit             PBDATDIR,BIT7
bbnz    IOPB7eq1    ;branch if IOPB7 as hi
ldp             #initphase
splk    #13,initphase ; 13 for run
*               splk    #03A2h,n_reff ; 3A2h for 1000 rpm 745h for 2000 rpm
splk    #0,stop       ; 1D1h for 500 rpm
b               notall ;return to the main polling cycle
*****************************
* stop
*****************************
LDP             #0E1h ;read data from IOPB5
bit             PBDATDIR,BIT5
bbnz    notstop         ;branch if IOPB5 as hi
ldp             #stop
splk    #1,stop
b               notall
*****************************
notstop
****************** **********
LDP             #0E1h ;read data from IOPB6
bit             PBDATDIR,BIT6
bbnz    notall  ;branch if IOPB6 as hi
ldp             #initphase
splk    #9,initphase ;
lacl    #0
sacl    stop
splk    #04,fovcurrent
;               sacl    n
*               sacl    iq
*               sacl    id
*               ldp             #0e1h
*               lacl    PBDATDIR
*               and             #0ff0fh
*               sacl    PBDATDIR   ;send option to IOPB4-7
*               b               notall ;return to the main polling cycle
*** END Option 4): stop n_ref = 0
notall
lacc    #018fh,10 ;delay
loop
sub             #1
bcnd    loop,NEQ
```

```
b               menu
*********************************************************
* _c_int2 Interrupt Service Routine
* synchronization of the control algorithm with the PWM
* underflow interrupt
*********************************************************
_c_int2:
************************
* Context Saving
************************
mar     *,ar4 ;AR4 active auxiliary reg (stack pointer)
mar     *-
sst     #1,*- ;save status register 1
sst     #0,*- ;save status register 0
sach    *- ;save MS word of accu
sacl    *- ;save LS word of accu
* END Context Saving *
mar     *,ar5 ;AR5 active auxiliary reg
ldp     #0e0h ;DP points to Event Manager control reg page
lacc    PIVR ;read the interrupt vector
ldp     #tmp
sacl    tmp
sub     #20H
bcnd    PDPRoutine,EQ
sub     #9
bcnd    ControlRoutine,EQ
b               ContextRestoreReturn
* END Int2 Interrupt Service Routine *
ContextRestoreReturn
*************************************
* Context restore and Return
*************************************
ldp             #0e8h
splk    #0200h,EVIFRA
larp    ar4
mar     *+
lacl    *+ ;Accu. restored for context restore
add     *+,16
lst     #0,*+
lst     #1,*+
clrc    INTM
ret
* END Context Restore and Return *
PDPRoutine
b               ContextRestoreReturn
ldp     #EVIFRA>>7
splk    #001h,EVIFRA ;Clear all flags, may be change
                        ;with only T1 underflow int.
ldp     #DP_EV
splk    #0fffh,ACTR
splk    #0207h,COMCON ;FIRST enable PWM operation
splk    #8207h,COMCON ;THEN enable Compare operation
ldp             #0e1h
lacl    PBDATDIR
or              #0080h
sacl    PBDATDIR ;send hi IOPB7
b               ContextRestoreReturn
* END PDPRoutine
ControlRoutine
*************************************
* check stop command from switch
ldp             #index
lacc    index
add             #1
sacl    index
ldp             #stop
lacc    stop
bcnd    OFF,NEQ
* check ON signal from apy
LDP             #0E1h ;read data from IOPD6
bit             PDDATDIR,BIT6
bcnd    IOPD6eq1,ntc        ;branch if IOPD6 as hi
*       b IOPD6eq1
OFF
LDP             #0E1h
lacl    PDDATDIR
or              #080h
sacl    PDDATDIR   ;send IOPD7 as hi to off Transistor
lacl    PBDATDIR
or              #010h
sacl    PBDATDIR   ;IOPB4 as hi disable buffer
b               ContextRestoreReturn
IOPD6eq1
LDP             #0E1h
```

```
lacl    PDDATDIR
and             #0ff7fh
sacl    PDDATDIR   ;send IOPD7 as Low to on Transistor
lacl    PBDATDIR
and             #0FFEFh
sacl    PBDATDIR   ;IOPB4 as low enable buffer
*       b Initial?
ADconversions
*****************************************************************
* Current sampling - AD conversions
* N.B. we will have to take only 10 bit (LSB)
*****************************************************************
ldp     #ADCTRL1>>7
splk    #9811h,ADCTRL1
        ;       lacl    ADCTRL1
        ;       or              #01h
        ;       sacl    ADCTRL1;ia and ib conversion start
                                        ;ADCIN0 selected for ia A/D1
                                        ;ADCIN8 selected for ib A/D2
conversion
bit     ADCTRL1,8
bcnd    conversion,tc ;wait approximatly 6.6us
lacc    ADCFIFO1,10
ldp     #i2
sach    i2
ldp     #DP_PF1
lacc    ADCFIFO2,10
ldp     #i3
sach    i3
Current_scaling
*********************************************
ldp     #i2
lacc    i2
;       add     #5 ;then we compensate the DC offset (that should be zero,
but
                                        ;it isn't)
and     #3ffh
sub     #482 ;then we have to subtract the offset (2.5V) to have
                                ;positive and negative values of the sampled
current
ldp             #tmp
sacl    tmp
lt              tmp
ldp             #Kcurrent
mpy     Kcurrent
pac ;
        ;       sfr
        ;       sfr
*               neg ;needed for ACPM
ldp             #i2
sach    i2,6 ;sampled current i2, f 4.12
lacc    i2
;               sub             #11
sacl    i2
********* nois digital filter **************
        ;       lacc    i2
        ;       sub             i2_old
        ;       ldp             #temp
        ;       sacl    temp
        ;       lacc    temp
        ;       abs
        ;       sub             #800
        ;       bcnd    i2_ok,LT
        ;       ldp             #i2
        ;       lacc    i2_old
        ;       sacl    i2
i2_ok
        ;       ldp             #i2
        ;       lacc    i2
        ;       sacl    i2_old
*********************************************


lacc    i3
;       sub     #09h ;then we compensate the DC offset (that should be
and     #3ffh
sub     #516
ldp             #tmp
sacl    tmp
lt              tmp
ldp             #Kcurrent
mpy     Kcurrent
pac
        ;       sfr
        ;       sfr
```

```
*               neg             ;needed for ACPM
ldp             #i3
sach    i3,6
        ;       lacc    i3
        ;       sub             #20
        ;       neg
        ;       sacl    i3
********* nois digital filter **************
        ;       lacc    i3
        ;       sub             i3_old
        ;       ldp             #temp
        ;       sacl    temp
        ;       lacc    temp
        ;       abs
        ;       sub             #800
        ;       bcnd    i3_ok,LT
        ;       ldp             #i3
        ;       lacc    i3_old
        ;       sacl    i3
i3_ok
        ;       ldp             #i3
        ;       lacc    i3
        ;       sacl    i3_old
*******************************************
lacc    i3
add     i2
neg
sacl    i1 ;i1 = -(i3+i2)
spm     0
*** END current sampling - AD conversions
*Sampled current scaling and
*******************************
* over current *
ldp             #i2
lacc    i2
abs
ldp             #ovcurrent
sub             ovcurrent
bcnd    notov,lt
lacl    fovcurrent
sub             #1
sacl    fovcurrent
bcnd    notov2,gt
splk    #1,stop
*******************************
ldp             #0e1h
lacl    PCDATDIR
or              #00aah
sacl    PCDATDIR   ;send to IOPC
lacl    PDDATDIR
and             #0fffeh
sacl    PDDATDIR
b               OFF
notov
splk    #04,fovcurrent ; 4 times over current
notov2
*******************************
* (a,b,c) -> (alfa,beta) axis transformation
* ialfa = i1
* ibeta = (2 * i2 + i1) / sqrt(3)
*********************************************
ldp             #i1
lacc    i1
ldp             #ialfa
;       sacl    ialfa
sacl    tmp
lacc    #0
lt              tmp
mpy             #0EAAh
lta     ialfa
mpy             #0155h
apac
sach    ialfa,4
ldp             #i2
lacc    i2,1    ;ibeta = (2 * i2 + i1) / sqrt(3)
add     i1 ;
ldp             #tmp
sacl    tmp ;
lt              tmp ;
mpy     SQRT3inv ;SQRT3inv = (1 / sqrt(3)) = 093dh;4.12 format =
0.577350269
pac ;
;       sach    ibeta,1
sach    tmp,1
```

```
lacc   #0
lt             tmp
mpy            #0EAAh
lta    ibeta
mpy            #0155h
apac
sach   ibeta,4
*** END Sampled current scaling and (a,b,c) -> (alfa,beta) axis
transformation
****************************
*** Encoder pulses reading
clrc   xf
ldp            #0e8h
lacc   T2CNT          ;we read the encoder pulses and ...
*              neg                    ;encoder plug in the opposite
direction ?
ldp            #tmp
sacl   tmp
ldp            #encoderold
sub            encoderold    ;subtract the previous sampling period value
                             ;to have the increment that we'll
                             ;accumulate in encoder
sacl   encincr ;
add            theta_m ;
bcnd   encmagzero,GT,EQ;here we start to normalize teta_m
                             ; value to the range [0;Encpulses-1]
add            encpulses ;the value of teta_m could be negative
                             ;it depends on the rotating direction
                             ;(depends on motor windings
                             ;to PWM Channels connections)
encmagzero
sacl   theta_m               ;now teta_m value is positive but could be
                      ;greater than Encpulses-1
sub            #Encpulses    ;we subtract Encpulses and we check whether
                             ;the difference is negative. If it is we
                             ;already have the right value in teta_m
bcnd   encminmax,LT ;
sacl   theta_m               ;otherwise the value of teta_m is greater
                      ;than Encpulses and so we have to store the
*              lar    ar0,theta_m
encminmax                              ;right value ok, now teta_m
contains the
ldp            #tmp                    ; right value in the range
lacc   tmp                    ;[0,Encpulses-1]
                              ;the actual value will be the old one during
ldp            #encoderold             ;the next sampling period
sacl   encoderold
*** END Encoder pulses reading
********************
Theta_calculation
********************
ldp    #theta_m
lt     theta_m ;multiply teta_m pulses by Kencoder (4.12
                      ;format constant) to have the rotor
                      ;electrical position
mpyu   Kencoder ;encoder pulses = 0 -> teta = 0fffh = 0 degrees
pac            ;encoder pulses = 1600 -> teta = 1fffh = 1*360
               ;encoder pulses = 3200 -> teta = 2fffh = 2*360
sach   theta_e,2
lacl   theta_e
and    #0fffh
sacl   theta_e
*** END theta calculation
*********************************************************
* Calculate speed and update reference speed variables
*********************************************************
lacc   speedstep     ;are we in speed control loop ? (SPEEDSTEP
                     ;times current control loop)
sub            #1 ;
sacl   speedstep     ;
bcnd   nocalc,GT     ;if we aren't, skip speed calculation
*** Speed calculation from encoder pulses
lt             speedtmp      ;multiply encoder pulses by Kspeed (8.8
                      ; format constant)
                      ;to have the value of speed
mpy            Kspeed                ;
pac                                  ;
*              rpt            #7                    ;
sfl                                  ;
sach   n,7          ; n=speed
*              lar            ar1,n
lacc   #0                     ;zero speedtmp for next calculation
sacl   speedtmp      ;
lacc   #SPEEDSTEP     ;restore speedstep to the value SPEEDSTEP
```

```
sacl    speedstep     ;for next speed control loop
*****************************************
*************************************************************
*************************************************************
ldp             #fre
lt              fre
mpy             kvpf    ;Q15
pac
sach    Voltmodu,1    ;Voltmodu = frequency*kvpf
lacc    fre
add             #10     ;
sacl    tmp
sub             FMAX
bcnd    fre_ok,LT
lacl    FMAX
sacl    fre
 b              sin_cos
fre_ok
lacl    tmp
sacl    fre
sin_cos
*************************************************************
*               lacc    n_reff
*               sub     n
*               lacc    n_reff    ; Wreff - Wr_estimate
ldp             #FluxS_mag
lacc    FluxS_mag
sub             #0600h
bcnd    no_gen,LT
ldp             #fre
lacc    fre
ldp             #speed_es
sub             speed_es
;       sub     speed_es    ;
*switch
ldp             #epispeed
sacl    epispeed   ; speed error
lacc    xispeed,10 ; Kpispeed(6.10)
lt              epispeed
mpy     Kpispeed   ;
apac
sach    upi,6
```

```
;here start to saturate
bit     upi,0
bcnd    upimagzeros,NTC ;If value +ve branch
lacc    TorqeREF_min
sub     upi
bcnd    neg_sat,GT ;if upi<Tor_refmin then branch to saturate
lacc    upi ;value of upi is valid
b               limiters
neg_sat
lacc    TorqeREF_min ;set acc to -ve saturated value
b               limiters
upimagzeros ;Value is positive
lacc    TorqeREF_max
sub     upi ;
bcnd    pos_sat,LT ;if upi>iqrmax then branch to saturate
lacc    upi ;value of upi valid
b               limiters
pos_sat
lacc    TorqeREF_max ;set acc to +ve saturated value
limiters
sacl    TorqeREF ;Store the acc as reference value
sub     upi
sacl    elpi
lt              elpi
mpy     Kcorspeed
pac
lt              epispeed
mpy     Kispeed
apac
add     xispeed,12
sach    xispeed,4
b               nocalc
no_gen          ldp             #TorqeREF
lacc    #0
sacl    TorqeREF
*** END Speed regulator with integral component correction
* ************************************
nocalc                                          ;branch here if we don't have to
calculate
                ldp             #speedtmp                       ; the speed
lacc    speedtmp ;use the actual encoder increment to update
                        ;the increments accumulator used to
```

```
                        ; calculate the speed
add     encincr ;
sacl    speedtmp ;

*****************************************************************
****    SLIDNG MODE GENERATION USd (flux component )  *******
*****************************************************************
*************************************************************************
** Control Law using LFSG (Linear feedbaack with switch gains Control)
**************
*************************************************************************

ldp             #FluxS_mag
lacc    FluxS_mag
sub             #0600h
bcnd    fluxmaglow,LT
ldp             #FluxREF
lacc    FluxREF  ;4.12 limit 0.5 Per unit
sub             FluxS_mag       ;4.12 limit
sacl    temp     ;temp store error Torqe new value
 lacc   temp
sub             er_flux    ;errorTe new- error Te old
sacl    er_flux     ;now ACC = errTe ,LT =CTe
lacc    #0
lt              er_flux
mpy             Cflux      ;CTe*(erTe)/dt ,CTe = Cte/Ts
lacc    temp    ; load new value of er_Te
sacl    er_flux
pac                     ;now, ACC = erT_e + CTe*(erTe)/dt
sach    temp,1   ; temp=CTe*(erTe)/dt
lacl    temp
add             er_flux
sacl    Sflux    ; sliding surface
***************         calculate er_flux*Sflux  *************************
   ;    lt              er_flux
   ;   mpy     Sflux
    ;  pac
    ;  sach    temp,4    ;temp = eTe*STe
    ;  lacc    temp
    ;  bcnd    neg_signflux,LT  ; if sign = -
    ;  lacc    er_flux
    ;  b               keepd

;neg_signflux
;       lacc    er_flux
;       neg
lacc    er_flux
abs
 sacl   temp
lacc    Sflux
bcnd    neg_signflux,LT  ; if sign = -
lacc    temp
b               keepd
neg_signflux
lacc    temp
neg
keepd sacl      epiflux   ; input PI= eTe*sign(eTe*STe)
lacc    xiflux,12 ; Kpispeed(6.10)
lt              epiflux
mpy     Kpiflux
apac
sach    upi,4
;here start to saturate
bit     upi,0
bcnd    upimagzeroflux,NTC ;If value +ve branch
lacc    USds_min
sub     upi
bcnd    neg_satflux,GT ;if upi<iqrmin then branch to saturate
lacc    upi ;value of upi is valid
b               limitersflux
lacc    USds_max ;set acc to +ve saturated value
neg_satflux
lacc    USds_min ;set acc to -ve saturated value
b               limitersflux
upimagzeroflux ;Value is positive
lacc    USds_max
sub     upi ;
bcnd    pos_satflux,LT ;if upi>iqrmax then branch to saturate
lacc    upi ;value of upi valid
b               limitersflux
pos_satflux
lacc    USds_max ;set acc to +ve saturated value
limitersflux
sacl    USds ;Store the acc as reference value
sub     upi
```

```
sacl    elpi
lt              elpi
mpy     Kcorflux
pac
lt              epiflux
mpy     Kiflux
apac
add     xiflux,12
sach    xiflux,4
b               fluxmag_ok
fluxmaglow
lacc    #0650h
sacl    USds
fluxmag_ok
*****************************************************************************
****    SLIDNG MODE GENERATION USq (Torqe component )
***************
*****************************************************************************
** Control Law using LFSG (Linear feedbaack with adaptive gains Control)
**************
*****************************************************************************
ldp             #FluxS_mag
lacc    FluxS_mag
sub             #0600h      ; check for low fluxd
bcnd    flux_verylow,LT
ldp             #TorqeREF
lacc    TorqeREF  ;4.12 limit 1.5 Per unit
sub             Te        ;4.12 limit
sacl    temp     ;temp store error Torqe new value
sub             er_Te    ;errorTe new- error Te old
 sacl   er_Te     ;now ACC = errTe ,LT =CTe
*       lacc    #0
lt              er_Te
mpy             CTe      ;CTe*(erTe)/dt ,CTe = Cte/Ts
  lacc  temp    ; load new value of er_Te
sacl    er_Te    ;
pac                     ;now, ACC = erT_e + CTe*(erTe)/dt
sach    temp,1   ; temp=CTe*(erTe)/dt
lacl    temp
  add           er_Te
  sacl  STe
***************         calculate er_Te*STe  *************************
    ;     lt              er_Te
    ;     mpy     STe
    ;     pac
    ;     sach    temp,4    ;temp = eTe*STe
    ;     lacc    temp
    ;     bcnd    neg_signTe,LT  ; if sign = -
     ;    lacc    er_Te
     ;    b               keepq
;neg_signTe
          ;         lacc    er_Te
          ;         neg
lacc    er_Te
abs
sacl    temp
lacc    STe
bcnd    neg_signTe,LT  ; if sign = -
lacc    temp
b               keepq
neg_signTe
lacc    temp
neg
keepq sacl    epiTe   ; input PI= eTe*sign(eTe*STe)
lacc    xiTe,12 ;
lt              epiTe
mpy     KpiTe    ; KpiTe(4.12)
apac
sach    upi,4
;here start to saturate
bit     upi,0
bcnd    upimagzeroTe,NTC ;If value +ve branch
lacc    USqs_min
sub     upi
bcnd    neg_satTe,GT ;if upi<iqrmin then branch to saturate
lacc    upi ;value of upi is valid
b               limitersTe
neg_satTe
lacc    USqs_min ;set acc to -ve saturated value
b               limitersTe
upimagzeroTe ;Value is positive
lacc    USqs_max
sub     upi ;
bcnd    pos_satTe,LT ;if upi>iqrmax then branch to saturate
```

```
lacc    upi ;value of upi valid
b               limitersTe
pos_satTe
lacc    USqs_max ;set acc to +ve saturated value
limitersTe
sacl    USqs ;Store the acc as reference value
sub     upi
sacl    elpi
lt              elpi
mpy     KcorTe
pac
lt              epiTe
mpy     KiTe
apac
add     xiTe,12
sach    xiTe,4
ldp             #FluxS_speed
*       lacc    USqs
lt              FluxS_speed
mpy             FluxS_mag
apac
ldp             #Vsqs
  sach Vsqs,4       ; control law stator flux q
lacc    Vsqs
          ;   lacc        #0
add             USqs
 sacl   Vsqs
lacc    Vsqs
sub             #01B33h ;limiter
bcnd    pos_,GT
lacc    Vsqs
sub             #0E4CDh
bcnd    neg_,LT
b               Vsqs_ok
pos_    lacc    #01B33h
sacl    Vsqs
b               Vsqs_ok
neg_    lacc    #0E4CDh
sacl    Vsqs
b               Vsqs_ok
flux_verylow
lacc    #0   ;100h
sacl    Vsqs
Vsqs_ok
***********************************************
***********************************************
* alfa-axis and beta-axis voltages calculation
* (d,q) -> (alfa,beta) axis transformation
* Vbetar = Vqr * cos(theta) + Vdr * sin(theta)
* Valfar =-Vqr * sin(theta) + Vdr * cos(theta)
***********************************************
lacc    #0
lt              USds            ;TREG0=Vdr
mpy     sin_fluxS       ;PREG=Vdr*sin(theta)
lta     Vsqs            ;ACC+=PREG ; TREG0=Vqr
mpy     cos_fluxS       ;PREG=Vqr*cos(theta)
mpya    sin_fluxS       ;ACC+=PREG ; PREG=Vqr*sin(theta)
sach    Vbetar,4
lacc    #0              ;ACC=0
lt              USds            ;TREG0=Vdr
mpys    cos_fluxS       ;ACC-=(PREG=Vqr*sin(theta))
apac                    ;ACC+=PREG
sach    Valfar,4
*** END alfa-axis and beta-axis voltages calculation
*               splk    #0,Vdr
*               splk    #0200h,Vqr
ldp             #fre
lt              fre
mpy             ktheta    ; 6.10
pac
rpt             #5
sfr
sach    tmp
*               lacc    theta,12
*               apac                    ;acc+=PREG
lacc    tmp
add             theta
*               sach    theta,4         ;theta = theta_old+fre*ktheta
*               lacl    theta
and             #0fffh
sacl    theta
***********************************************
* Sine and cosine wave calculation from
* theta values using sine look-up table
```

```
*************************************************
lacc    theta ;theta range is [0;1000h] 4.12 format = [0;360] degrees
                        ;so we have a pointer (in the range [0;0ffh]) to the
                        ;sine look-up table in the second and third nibble
rpt     #3
sfr
and     #0ffh ;now ACC contains the pointer to access the table
sacl    indice1 ;
add     #sintab ;
sacl    tmp ;
lar     ar5,tmp ;
nop
nop                     ;
mar     *,ar5
lacl    * ;
nop
sacl    seno1 ;now we have sine value
lacl    indice1 ;the same thing for cosine ... cos(theta) = sin(theta+90°)
add     #040h ;90 degrees = 40h elements of the table
and     #0ffh ;
sacl    indice1 ;we use the same pointer (we don't care)
add     #sintab ;
sacl    tmp ;
lar     ar5,tmp ;
lacc    * ;
sacl    coseno ;now we have cosine value
*****************************************************
*********SLIDINGMODE STATOR flux observer **********************
*****************************************************
*       ldp             #FluxSD
*      lacc     FluxSD
*               sacl    FluxSD_old
ldp             #ialfa
*               lacc    #0
lt              ialfa
mpy             Rs    ;Q15
pac
sach    temp,1
lacc    Valfar     ;Q12
sub             temp
*               sacl    test
sacl    temp     ;temp = VsD-Rs*isD
ldp             #ialfa ;
lacc    ialfa
sub             ialfae  ; isD-isDes
ldp             #erisd
sacl    erisd
bcnd    POS_erisd,GT    ; sign definition
ldp             #Ks11
lacc    Ks11
neg
ldp             #X11
sacl    X11
ldp             #Ks11i
lacc    Ks11i
neg
sacl    Ks11i
lt              Ks11i
mpy     #040Ch  ;Q15
pac
ldp             #tmp
sach    tmp,1
lacc    tmp
ldp             #X11
add             X11
sacl    X11
b               add_signD
POS_erisd
ldp             #Ks11
lacc    Ks11
ldp             #X11
sacl    X11
ldp     #Ks11i
lt              Ks11i
mpy     #040Ch  ;Q15
pac
ldp             #tmp
sach    tmp,1
lacc    tmp
ldp             #X11
add             X11
sacl    X11
add_signD
ldp             #Ks11
```

```
lacc    Ks11
ldp             #X11
add             X11
ldp     #temp
add             temp
sacl    temp    ;   temp = VsD-Rs*isD+Ks1(sign erisd)
lacc    #0
lt              temp
mpy             #040Ch;T_sampling   ;Q15
lta             FluxSD
mpy             Klow
apac
sach    FluxSD,1
        ;       ldp             #temp
        ;       lt              temp
        ;       ldp             #K2flux     ;
        ;       mpy             K2flux ;T_sampling   ;Q15
        ;       pac
        ;       ldp             #temp
        ;       sach    temp,1
        ;       lacc    temp
        ;       add             FluxSD
;               lta             FluxSD
        ;       sacl    temp
        ;       lt              temp
        ;       ldp             #K1flux
        ;       mpy             K1flux
        ;       mpy             Klow    ;using lowpass to eliminate DC
component
        ;       apac
         ;  pac
         ;  ldp          #FluxSD
        ;       sach    FluxSD,1

************************************
*       ldp             #FluxSQ
*       lacc    FluxSQ
*               sacl    FluxSQ_old
ldp             #ibeta
lacc    #0
lt              ibeta
mpy             Rs
pac
sach    temp,1
lacc    Vbetar      ;Q12
sub             temp
sacl    temp    ;   temp = VsD-Rs*isD
ldp             #ibeta ;
lacc    ibeta
sub             ibetae ; isQ-isQes
ldp             #erisq
sacl    erisq
ldp             #Ks12
bcnd    POS_erisq,GT    ; sign definition
lacc    Ks12
neg
sacl    X12
ldp             #Ks12i
lacc    Ks12i
neg
sacl    Ks12i
lt              Ks12i
mpy     #040Ch  ;Q15
pac
ldp             #tmp
sach    tmp,1
lacc    tmp
ldp             #X12
add             X12
sacl    X12
b               add_signQ
POS_erisq
ldp             #Ks12
lacc    Ks12
ldp             #X12
sacl    X12
ldp     #Ks12i
lt              Ks12i
mpy     #040Ch  ;Q15
pac
ldp             #tmp
sach    tmp,1
lacc    tmp
ldp             #X12
```

```
add             X12
sacl    X12
add_signQ
ldp             #Ks12
lacc    Ks12
ldp             #X12
add             X12
ldp     #temp
add             temp
sacl    temp    ;   temp = VsD-Rs*isD+Ks2(sign erisd)
lacc    #0
lt              temp
mpy             #040Ch  ;T_sampling   ;Q15
lta             FluxSQ
mpy             Klow
apac
sach    FluxSQ,1
        ;       ldp             #K2flux     ;
        ;       mpy             K2flux ;T_sampling   ;Q15
        ;       pac
        ;       ldp             #temp
        ;       sach    temp,1
        ;       lacc    temp
        ;       add             FluxSQ
;               lta             FluxSD
        ;       sacl    temp
        ;       lt              temp
        ;       ldp             #K1flux
        ;       mpy             K1flux
        ;       mpy             Klow    ;using lowpass to eliminate DC
component
        ;       apac
         ;  pac
        ;       ldp         #FluxSQ
        ;       sach    FluxSQ,1
*********************************************************************
********** end SMO stator flux improve magnitude ********************
*********************************************************************
******** Calculate flux rotor in DQ component using current MODEL
***********
******  FluxRD =  (Lr*FluxSD-Lq2*iSD)/Lm ************
ldp             #ialfa
lacc    #0
lt              ialfa   ; load iSD to TREG
mpy             L2qLm                   ; now PREG = iSD*Lq2/Lm    2.30
lts             FluxSD                  ; Acc = -iSD*Lq2/Lm ,TREG = fluxSD
mpy             LrLm            ; PREG = fluxSD*Lr/Lm
apac
        ;       sach    FluxRD,4  ;
sach    tmp,4
lacc    #0
lt              tmp
mpy             #0F3Ch
lta             FluxRD
mpy             #0C3h
apac
sach    FluxRD,4
    ;   mpy             #07B7Eh
    ;   pac
    ;   sach    tmp,1
    ;   lacc    tmp
    ;   add             FluxRD
    ;   sacl    tmp
    ;   lt              tmp
    ;   mpy             #0189h
    ;   pac
    ;   sach    FluxRD,1
        ;       lt              K3_se
        ;       mpy             tmp
        ;       pac
        ;       lt              K4_se       ;low pass for noisy current
        ;       mpy             FluxRD
        ;       apac
        ;       sach    FluxRD,1

;*******  FluxRQ =  (Lr*FluxSQ-Lq2*iSQ)/Lm ************************
ldp             #ibeta
lacc    #0
lt              ibeta   ; load iSD to TREG
mpy             L2qLm                   ; now PREG = iSD*Lq2/Lm
lts             FluxSQ                  ; Acc = -iSD*Lq2/Lm ,TREG = fluxSD
mpy             LrLm            ; PREG = fluxSD*Lr/Lm
apac                            ; add preg and store in acc
        ;       sach    FluxRQ,4
```

```
sach   tmp,4
lacc   #0
lt             tmp
mpy            #0F3Ch
lta            FluxRQ
 mpy           #0C3h
apac
sach   FluxRQ,4
   ;   mpy             #07B7Eh
   ;   pac
   ;   sach    tmp,1
   ;   lacc    tmp
    ;  add             FluxRQ
    ;  sacl    tmp
    ;  lt              tmp
    ;  mpy             #0189h
    ;  pac
    ;  sach    FluxRQ,1
         ;          lt              K3_se
         ;          mpy             tmp
         ;          pac
         ;          lt              K4_se
         ;          mpy             FluxRQ
         ;          apac
         ;          sach   FluxRQ,1

************************************************
************* module to find rotor flux angle ***************
; Compute the theta_flux
; Check |FluxRQ|/|FluxRD|>1 ?
ldp            #FluxRD
LACC FluxRD                     ; ACC = fluxrd
ABS             ; ACC = |flux|
SACL tmp1                       ; tmp1 = |wRD|
LACC FluxRQ                     ; ACC = sine
ABS             ; ACC = |sine|
SACL tmp                        ; tmp = |fluxrQ|
SUB            tmp1                     ; ACC = tmp - tmp1
BCND LARGE_Tanr,GT     ; Branch to LARGE_Tan if tmp > tmp1
BCND EQ_Tan,EQ         ; Branch to EQ_Tan if tmp = tmp1
BCND EQ_Tanr,EQ
        ; Here, |sine|/|cosine| is less than 1.
SMALL_Tanr
LACC tmp,12             ; ACC = tmp left shifted by 12 (tangent=Q12)
RPT            #15              ; Repeat SUBC 16 times
SUBC tmp1       ; Dividing tmp/tmp1
SACL tangent    ; tangent = tmp/tmp1   (Q12)
; Looking at the table for arctan(|sine|/|cosine|) directly
LACC tangent,12  ; ACC = tangent/128  (2^12/2^4 = 2^8 = 256)
SACH tmp2       ; ptr_fte = tangent/128
LACC #ATANTAB_45
ADD            tmp2
TBLR theta_fluxR   ; theta_re = arctan(tmp/tmp1) in Q12
B              DIV_Tanr_END
; Here, |sine|/|cosine| is greater than 1,
; so |cosine|/|sine| < 1 is computed instead.
LARGE_Tanr
LACC tmp1,12            ; ACC = tmp1 left shifted by 12(tangent=Q12)
RPT            #15              ; Repeat SUBC 16 times
SUBC tmp            ; Dividing tmp1/tmp
SACL tangent        ; tangent = tmp1/tmp   (Q12)
; Looking at the table for arctan(|cosine|/|sine|)
LACC tangent,12    ; ACC = tangent/128  (2^15/2^4 = 2^8 = 256)
SACH tmp2           ; atan_ptr = tangent/128
LACC #ATANTAB_45
ADD            tmp2
TBLR theta_fluxR   ; theta_re = arctan(tmp1/tmp) in Q12
LACC #1024          ; ACC = 8192 (Q15),1024 (Q12) = 90 deg
SUB            theta_fluxR   ; ACC = 90 - arctan(tmp1/tmp)   (Q12)
SACL theta_fluxR   ; theta_re = arctan(tmp/tmp1) (Q12)
B              DIV_Tanr_END
; Here, |sine|/|cosine| is equal to 1,
; i.e., |sine| = |cosine|
EQ_Tanr
SPLK #512,theta_fluxR     ; theta_re = 45  (Q12)
DIV_Tanr_END
; Determine the quadrant by looking at signs of cosine and sine
LACC FluxRD                     ; ACC = cosine
;       NEG
BCND           NEG_Dr, LT           ; Branch to NEG_D
POS_Dr                                  ; if cosine < 0 POS_D
LACC FluxRQ                     ; ACC = sine
        ;       NEG
BCND POS_D_NEG_Qr, LT       ; Branch to POS_D_NEG_Q
```

```
POS_D_POS_Qr            ;if sine < 0 POS_D_POS_Q
; Do nothing here
B               QUADRANT_Tanr_END
POS_D_NEG_Qr
LACC #4095              ; ACC = 4095 = 360 deg
SUB             theta_fluxR             ; ACC = 360 - theta_re       (Q12)
SACL theta_fluxR            ; theta_re = 360 - theta_re   (Q12)
B               QUADRANT_Tanr_END
NEG_Dr
LACC FluxRQ                             ; ACC = sine
        ;       NEG
BCND NEG_D_NEG_Qr, LT       ; Branch to NEG_D_NEG_Q if sine < 0
NEG_D_POS_Qr
LACC #2048                              ; ACC = 2048 = 180 deg
SUB             theta_fluxR             ; ACC = 180 - theta_re       (Q12)
SACL theta_fluxR            ; theta_re = 180 - theta_re    (Q12)
B               QUADRANT_Tanr_END
NEG_D_NEG_Qr
LACC #2048                  ; ACC = 2048 = 180 deg
ADD             theta_fluxR             ; ACC = 180 + theta_re       (Q12)
SACL theta_fluxR            ; theta_re = 180 + theta_re
QUADRANT_Tanr_END
        ;       lacc    theta_fluxR
        ;       sub             theta_fluxR_old
        ;       sacl    temp
        ;       lacl    temp
        ;       abs
        ;       sub             #250h
        ;       bcnd    theta_fluxROK,LT
        ;       lacc    theta_fluxR_old
        ;       sacl    theta_fluxR
theta_fluxROK
******************END angle rotorflux calculation **************
************ now We use angle to tranform any variable to fluxR axis **
      ldp               #theta_fluxR
      lacc      theta_fluxR
rpt     #3
sfr
and     #0ffh   ;now ACC contains the pointer to access the table
sacl    indice1         ;
add     #sintab         ;
sacl    tmp             ;
lar     ar5,tmp         ;
nop
nop                     ;
mar     *,ar5
lacl    *               ;
nop
sacl    sin_fluxR       ;now we have sine value
lacl    indice1         ;the same thing for cosine ... cos(theta) =
;sin(theta+90°)
add     #040h           ;90 degrees = 40h elements of the table
and     #0ffh   ;
sacl    indice1         ;we use the same pointer (we don't care)
add     #sintab         ;
sacl    tmp             ;
lar     ar5,tmp         ;
lacc    *               ;
sacl    cos_fluxR       ;now we have cosine value

********  end fine sine cos angle flux rotor**************************************
;*****************************************
* d-axis and q-axis stator flux calculation in rotor flux flame
* (alfa,beta) -> (d,q) axis transformation
* FluxSdr = FluxSD * cos(thetafluxR) + FluxSQ * sin(thetaflux_R)
* FluxSqr =-FluxSD * sin(thetafluxR) + FluxSQ * cos(thetaflux_R)
*****************************************
ldp             #FluxSQ
lacc    #0
lt              FluxSQ          ;TREG0=ibeta
mpy     sin_fluxR       ;PREG=ibeta*sin(theta)
lta     FluxSD          ;ACC+=PREG ; TREG0=ialfa
mpy     cos_fluxR       ;PREG=ialfa*cos(theta)
apac                    ;ACC+=PREG
sach    FluxSdr,4   ; value of stator current in rotor flux q component
*** END d-axis and q-axis stator flux in rotor flux frame calculation
********** calculate sign(eror iSdr) *****************
 ldp    #erisd
  lacc  erisd
bcnd    signd_plus,GT
lacl    #07FFEh         ;  1.15    value of sign
neg
b               value_signd
signd_plus      lacl    #07FFEh    ; nearest 1 of value sign of 1.15 format
```

```
value_signd  sacl    dsign     ; sign of error stator current d component
*******  calculate ksign iqr *****************
ldp     #erisq
lacc    erisq
bcnd    signq_plus,GT
lacc    #07FFEh
neg
b               value_signq
signq_plus    lacc    #07FFEh
value_signq  sacl    qsign    ; sign of error stator current d component
*********************************************
* d-axis and q-axis error of stator current calculation in  rotor flux flame
* (alfa,beta) -> (d,q) axis transformation
* Serid = Serialfa * cos(theta) + Seribeta * sin(theta)
* Seriq =-Serialfa * sin(theta) + Seribeta * cos(theta) * S notation is sign
*********************************************
ldp             #qsign      ; out put  iSqr  iSdr are 1.15 format
lacc    #0
lt              qsign  ;TREG0=ibeta
ldp             #sin_fluxR
mpy     sin_fluxR       ;PREG=ibeta*sin(theta)
ldp             #dsign
lta     dsign  ;ACC+=PREG ; TREG0=ialfa
ldp             #cos_fluxR
mpy     cos_fluxR       ;PREG=ialfa*cos(theta)
mpya  sin_fluxR       ;ACC+=PREG ; PREG=ialfa*sin(theta)
ldp             #eiSdr
sach    eiSdr,1      ; value ofstator current in rotor flux d component
lacc    #0              ;ACC=0
lt              dsign  ;TREG0=ibeta
ldp             #cos_fluxR
mpys  cos_fluxR       ;ACC-=(PREG=ialfa*sin(theta))
apac                    ;ACC+=PREG
ldp             #eiSqr
sach    eiSqr,1  ; value of stator current error in rotor flux q component
*************************************************************************
* corection term Real{(K2d+j*K2q)(iSdr+j*iSqr)} = -K2d*iSdr-K2q*iSqr *****
*************************************************************************
lacc    #0
lt              eiSqr;   Q12
mpy             K2q  ; Q15   ACC = 0 ,PREG = K2q*iSqr 5.27
lts             eiSdr  ; Q12   ACC = -K2q*iSqr ,TREG = iSdr 1.15
mpy             K2d    ;        Q15  PREG =  K2d*iSdr  5.27
spac        ;   5.27
ldp             #Kr   ;
sach    Kr,4 ; 4.12       corrective term of an rotorflux magnitude
*****************************************************
*******   now calculate flux rotor magnitude  ***********
******* wRdr(k+1) = wRdr(k)+Ts*(-KR11*wSdr+KR12*wRdr(k)+Kr)
*********************************************************
 lacc   #0
 ldp    #FluxRdr
lt              FluxRdr
mpy    KR12
lts       FluxSdr         ;lts
mpy    KR11     ; Kr11 constant
  apac
ldp     #temp
  sach temp,1
lacc    Kr
add     temp
  sacl  temp              ; temp= -KR11*wSdr+KR12*wRdr(k)+Kr
lt              temp        ;4.12
mpy    T_sampling   ;1.15
pac
sach    temp,1          ;4.12
lacc    temp
add     FluxRdr      ; add wRdr(k)
sacl    FluxRdr             ; store for next itteration
sacl    temp
    ;   lacc     temp
    ;   abs
    ;   sacl     temp
*       lacc     FluxSdr
*       add      temp
*       sacl     temp
******** update rotor flux DQ frame ********
 lt             temp
 mpy    cos_fluxR
 pac                ;       4.12
 sach   FluxRD,4     ; new value
 lt             temp
 mpy    sin_fluxR
 pac
```

```
 sach  FluxRQ,4      ; new value
************************************************************
********** update  calculate iSDQ estimate ***************
*******  iSDes =  (Lr*FluxSD-Lm*FluxRD)/Lq2 ************
lacc    #0
lt              FluxRD          ; load iSD to TREG
mpy             LmL2q                   ; now PREG = FluxRD*Lm/Lq2     5.27
lts             FluxSD                  ; Acc = -iSD*Lq2/Lm ,TREG = fluxSD
mpy             LrL2q           ; PREG = fluxSD*Lr/Lm          5.27
apac                            ; add preg and store in acc
sach   ialfae,4     ;** new value of estimate current       1.15

;*******  iSQes =  (Lr*FluxSQ-Lm*FluxRQ)/Lq2 ************
 lacc   #0
 lt             FluxRQ                  ; load iSD to TREG
 mpy            LmL2q                   ; now PREG = iSD*Lm/Lq2
 lts            FluxSQ                  ; Acc = -iSD*Lq2/Lm ,TREG = fluxSQ
 mpy            LrL2q           ; PREG = fluxSD*Lr/Lm                  5.27
 apac                           ; add preg and store in acc
 sach  ibetae,4     ;** new value of estimate current     1.15
************************************************************
************** calculate electrical torque PER UNIT **********
*                               Te per unit = fluxSD*iSQ - fluxSQ*iSD
*************
*************************************************************************
Te_cal
ldp             #FluxSQ                         ; 1.15
lacc    #0
lt              FluxSQ
mpy             ialfa   ; PREG = wSQ*iSD,ACC = 0            2.30
lts             FluxSD          ; ACC -= wSQ*iSD ,TREG = wSD
mpy             ibeta   ;
apac
   ;    sach    Te,4   ;
sach   tmp,4
lacc    #0
lt              tmp
  mpy           #0F3Ch
lta             Te
mpy             #0C3h
apac
sach   Te,4
```

```
************************************************************
********* OPEN-LOOP speed estimator ***********
***** Wr = FluxR_speed - Wslip *********
*************************************************
   ;    lacc      #0
   ;    ldp               #FluxRD_old
   ;    lt                FluxRD_old
   ;    ldp               #FluxRQ
   ;    mpy               FluxRQ
   ;    ldp               #FluxRQ_old
   ;    lta               FluxRQ_old
   ;    ldp               #FluxRD
    ;   mpy     FluxRD
    ;   spac
    ;   lt                tmp2
    ;   mpy               #2037
    ;   pac
    ;   rpt               #5
    ;   sfr
    ;   sacl      FluxR_speed
ldp             #FluxRdr
lacc    FluxRdr
abs
sacl    temp
sub             #0400h
bcnd    low,LT
b               good
low
lacc    #0400h
sacl    temp
good
sqra    temp
pac
sach   temp,4     ; |fluxR|^2   Q12
ldp             #Te
lt              Te       ;Q12
mpy             Kslip     ;Q15
pac
sach   tmp,1     ;Q12
     ; lacc      FluxR_speed
     ; sub               tmp
     ; sacl      tmp
```

```
 lacc  tmp,12
bcnd   neg_slip,LT
rpt             #15
subc   temp     ;Te*Kslip/|fluxR|^2
b               result
neg_slip
                neg
                rpt             #15
                subc    temp
                sacl    tmp
                lacc    tmp
                neg
result      sacl        W_slip ;FluxR_speed
        ;       lt              FluxR_speed;#07FBEh ;K3_se
        ;       mpy             #07FBEh
        ;       pac
        ;       lt              temp  ;K4_se
        ;       mpy             #041h
        ;       apac
        ;       sach    FluxR_speed,1

                ldp             #FluxRD
                lacc    FluxRD
                ldp             #FluxRD_old
      sacl      FluxRD_old
      ldp               #FluxRQ
lacc    FluxRQ
ldp             #FluxRQ_old
sacl    FluxRQ_old
ldp             #speed_es
lacc    speed_es
sacl    speed_es_old
************ module find rotor flux speed ***************
      ; Differentiator
; Check theta_r_se is within the differentiable range (min_theta,max_theta)
LACC max_theta
SUB             theta_fluxR
BCND LESS_MAXr,GT               ; Branch to LESS_MAX if max_theta >
theta_r_se
B               SKIP_DIFFr
LESS_MAXr
LACC min_theta
SUB             theta_fluxR
BCND MORE_MINr,LT    ; Branch to MORE_MIN if min_theta < theta_r_se
B               SKIP_DIFFr
MORE_MINr
        ; Differentiating the rotor flux angle
        ; if min_theta < theta_r_se < max_theta
LT              K2_se                   ; TREG = K2             (Q6)
MPY             theta_fluxR     ; PREG = K2*theta_r(k)    (Q21)
PAC                                             ; ACC = K2*theta_r(k)     (Q21)
MPY             theta_fluxR_old     ; PREG = K2*theta_r(k-1)  (Q21)
SPAC                            ; ACC = K2*theta_r(k)-K2*theta_r(k-1)  (Q21)
SFL                             ; ACC = K2*theta_r(k)-K2*theta_r(k-1)  (Q22)
SFL                             ; ACC = K2*theta_r(k)-K2*theta_r(k-1)  (Q23)
SFL                             ; ACC = K2*theta_r(k)-K2*theta_r(k-1)  (Q24)
SACH tmp,7      ; tmp1_se = K2*theta_r(k)-K2*theta_r(k-1)  (Q15)
                ; Low-pass filter
LT              K3_se                   ; TREG = K3             (Q15)
MPY             FluxR_speed         ; PREG = K3*w_psi_r(k-1)  (Q30)
PAC                                     ; ACC =  K3*w_psi_r(k-1)  (Q30)
LT              K4_se                   ; TREG = K4             (Q15)
MPY             tmp         ; PREG = K4*tmp1_se       (Q30)
APAC                            ; ACC = K3*w_psi_r(k-1)+K4*tmp1_se  (Q30)
SACH FluxR_speed,1      ; w_psi_r(k) = K3*w_psi_r(k-1)+K4*tmp1_se
(Q15)
SKIP_DIFFr
; Update the rotor flux angle
LACC theta_fluxR            ; ACC = theta_r_se          (Q15)
SACL theta_fluxR_old    ; theta_r_old = theta_r_se    (Q15)
    ;   lacc    FluxR_speed
    ;   sacl    speed_es
********** module to eliminate over acceration or noise in speed
    ;   lacc    speed_es
    ;   sub             speed_es_old
    ;   sacl    temp
    ;   lacc    temp
    ;   abs
    ;   sub             #800h       ; condition to prevent over acceration of
fluxS
    ;   bcnd    sat_speed,GT
    ;   b               speed_ok
sat_speed
        ;       lacc    speed_es_old
```

```
        ;       sacl    speed_es
speed_ok
ldp             #speed_es
lt              speed_es
ldp             #speed_scale
mpy             speed_scale   ;Q15 constant for change to rpm
pac
ldp             #n_es
sach   n_es,4 ; speed estimate in rpm
*********************************************
***************** stator flux magnitude **********
   ;    ldp             #FluxS_mag
;               lacc    FluxS_mag
;               sacl    FluxS_mag_old
FluxS_magnitude
;           mar         *,ar0
;               lar             ar0,#temp1
;           mpy             #0
 ;      zac
 ;      spm             2
  ;     sqra    FluxSD
  ;     sqra    FluxSQ
  ;     apac
  ;     sach    *     ;  temp store FluxSD^2+FluxSQ^2
  ;     sach    tmp
*       sach    FluxS_mag
  ;             clrc    sxm
;               lacc    *+              ;lacc Vr, X(0)
;               sfr                     ;Initial approximate root = N/2
;               sacl    *+              ;(AR0+1) = X(0)/2
;               splk    #10,*  ;(AR0+2) = 10 (iterations for square root)
fluxSsqrt:                      ;X(n) = 0.5 * ( X(n-1) + N/X(n-1) ):
;               sbrk    #2
        ;       lacc    *+
;               rpt     #15
        ;       subc    *
        ;       and     #0FFFFh
        ;       add     *
        ;       sfr
        ;       sacl    *+
        ;       lacc    *               ;Repeat until iterations completed:
        ;       sub     #1
        ;       sacl    *
        ;       bcnd    fluxSsqrt,NEQ
        ;       mar     *-              ;Restore AR5 & put result in acclow:
        ;       lacc    *
        ;       mar     *-
        ;       setc    sxm
        ;       spm     0
        ;       ldp             #FluxS_mag
        ;       sacl    FluxS_mag,6     ;result
*************************************************************
   ;    lacc    FluxS_mag
   ;    sub             FluxS_mag_old
   ;    sacl    temp
   ;    lacc    temp
   ;    abs
   ;    sub             #80h      ; condition to prevent over acceration of fluxS
   ;    bcnd    sat_fluxS,GT
   ;    b               fluxS_ok
sat_fluxS
;               lacc    FluxS_mag_old
;               sacl    FluxS_mag
fluxS_ok
*************************************************************
***************************************
* END Sine and cosine wave calculation from theta values using sine look-
up table
*               ldp             #count
*               lacl    count

*               sub             #20h
*               bcnd    update,EQ
*               b               no_update
***********************************************
* alfa-axis and beta-axis voltages calculation
* Vbetar =  Voltmod * sin(theta)
* Valfar =  Voltmod * cos(theta)
***********************************************
no_update
        ;       lacc    #0
        ;       lt              Voltmodu ;TREG0=Voltmod
        ;       mpy     seno1 ;PREG=Voltmod*sin(theta)
        ;       mpya    coseno          ;acc+=PREG PREG=Voltmod*cos(theta)
```

```
        ;       sach   Vbetar,4
        ;       pac             ;ACC=PREG
        ;       sach   Valfar,4
*** END alfa-axis and beta-axis voltages calculation
      ldp               #FluxS_mag
      lacc      FluxS_mag
      sub               #0600h
      bcnd      lowFluxS,LT
************ module to find stator flux angle ***************
; Compute the theta_flux
; Check |FluxSQ|/|FluxSD|>1 ?
ldp             #FluxSD
LACC FluxSD                     ; ACC = fluxrd
ABS             ; ACC = |flux|
SACL tmp1                       ; tmp1 = |wRD|
LACC FluxSQ                     ; ACC = sine
ABS             ; ACC = |sine|
SACL tmp                        ; tmp = |fluxrQ|
SUB             tmp1                    ; ACC = tmp - tmp1
BCND LARGE_Tans,GT    ; Branch to LARGE_Tan if tmp > tmp1
BCND EQ_Tan,EQ          ; Branch to EQ_Tan if tmp = tmp1
BCND EQ_Tans,EQ
; Here, |sine|/|cosine| is less than 1.
SMALL_Tans
LACC tmp,12             ; ACC = tmp left shifted by 12 (tangent=Q12)
RPT             #15             ; Repeat SUBC 16 times
SUBC tmp1       ; Dividing tmp/tmp1
SACL tangent    ; tangent = tmp/tmp1   (Q12)
; Looking at the table for arctan(|sine|/|cosine|) directly
LACC tangent,12  ; ACC = tangent/128  (2^12/2^4 = 2^8 = 256)
SACH tmp2       ; ptr_fte = tangent/128
LACC #ATANTAB_45
ADD             tmp2
TBLR theta_fluxS    ; theta_re = arctan(tmp/tmp1) in Q12
B               DIV_Tan_ENDs
; Here, |sine|/|cosine| is greater than 1,
; so |cosine|/|sine| < 1 is computed instead.
LARGE_Tans
LACC tmp1,12            ; ACC = tmp1 left shifted by 12(tangent=Q12)
RPT             #15             ; Repeat SUBC 16 times
SUBC tmp            ; Dividing tmp1/tmp
SACL tangent        ; tangent = tmp1/tmp   (Q12)
; Looking at the table for arctan(|cosine|/|sine|)
LACC tangent,12    ; ACC = tangent/128  (2^15/2^4 = 2^8 = 256)
SACH tmp2           ; atan_ptr = tangent/128
LACC #ATANTAB_45
ADD             tmp2
TBLR theta_fluxS    ; theta_re = arctan(tmp1/tmp) in Q12
LACC #1024          ; ACC = 8192 (Q15),1024 (Q12) = 90 deg
SUB             theta_fluxS    ; ACC = 90 - arctan(tmp1/tmp)   (Q12)
SACL theta_fluxS    ; theta_re = arctan(tmp/tmp1) (Q12)
B               DIV_Tan_ENDs
; Here, |sine|/|cosine| is equal to 1,
; i.e., |sine| = |cosine|
EQ_Tans
SPLK #512,theta_fluxS     ; theta_re = 45  (Q12)
DIV_Tan_ENDs
; Determine the quadrant by looking at signs of cosine and sine
LACC FluxSD                     ; ACC = cosine
        ;       NEG
BCND            NEG_Ds, LT          ; Branch to NEG_D
POS_Ds; if cosine < 0 POS_D
LACC FluxSQ                     ; ACC = sine
        ;       NEG
BCND POS_D_NEG_Qs, LT       ; Branch to POS_D_NEG_Q
POS_D_POS_Qs;if sine < 0 POS_D_POS_Q
; Do nothing here
        B               QUADRANT_Tans_END
POS_D_NEG_Qs
LACC #4095              ; ACC = 4095 = 360 deg
SUB             theta_fluxS             ; ACC = 360 - theta_re       (Q12)
SACL theta_fluxS            ; theta_re = 360 - theta_re   (Q12)
B               QUADRANT_Tans_END
NEG_Ds
LACC FluxSQ                     ; ACC = sine
        ;       NEG
BCND NEG_D_NEG_Qs, LT       ; Branch to NEG_D_NEG_Q if sine < 0
NEG_D_POS_Qs
LACC #2048                  ; ACC = 2048 = 180 deg
SUB             theta_fluxS             ; ACC = 180 - theta_re       (Q12)
SACL theta_fluxS            ; theta_re = 180 - theta_re    (Q12)
B               QUADRANT_Tans_END
NEG_D_NEG_Qs
LACC #2048                  ; ACC = 2048 = 180 deg
```

```
ADD         theta_fluxS          ; ACC = 180 + theta_re       (Q12)
SACL theta_fluxS          ; theta_re = 180 + theta_re
QUADRANT_Tans_END
ldp         #FluxS_speed
lacc   FluxS_speed
sacl   FluxS_speed_old
      ; Differentiator
; Check theta_r_se is within the differentiable range (min_theta,max_theta)
LACC max_theta
SUB         theta_fluxS
BCND LESS_MAXs,GT            ; Branch to LESS_MAX if max_theta >
theta_r_se
B           SKIP_DIFFs
LESS_MAXs
LACC min_theta
SUB         theta_fluxS
BCND MORE_MINs,LT    ; Branch to MORE_MIN if min_theta <
theta_r_se
B           SKIP_DIFFs
MORE_MINs
        ; Differentiating the rotor flux angle
        ; if min_theta < theta_r_se < max_theta
LT          K2_se                ; TREG = K2            (Q6)
MPY         theta_fluxS     ; PREG = K2*theta_r(k)    (Q21)
PAC                                  ; ACC = K2*theta_r(k)     (Q21)
MPY         theta_fluxS_old     ; PREG = K2*theta_r(k-1)   (Q21)
SPAC                                 ; ACC = K2*theta_r(k)-K2*theta_r(k-1)
(Q21)
SFL                  ; ACC = K2*theta_r(k)-K2*theta_r(k-1)   (Q22)
SFL                  ; ACC = K2*theta_r(k)-K2*theta_r(k-1)   (Q23)
SFL                  ; ACC = K2*theta_r(k)-K2*theta_r(k-1)   (Q24)
SACH tmp,7      ; tmp1_se = K2*theta_r(k)-K2*theta_r(k-1)   (Q15)
            ; Low-pass filter
LT          K3_se                ; TREG = K3            (Q15)
MPY         FluxS_speed       ; PREG = K3*w_psi_r(k-1)   (Q30)
 PAC                                 ; ACC =  K3*w_psi_r(k-1)   (Q30)
LT          K4_se                ; TREG = K4            (Q15)
MPY         tmp       ; PREG = K4*tmp1_se       (Q30)
APAC                         ; ACC = K3*w_psi_r(k-1)+K4*tmp1_se  (Q30)
SACH FluxS_speed,1      ; w_psi_r(k) = K3*w_psi_r(k-1)+K4*tmp1_se
(Q15)
SKIP_DIFFs
lacc   tmp
sub         W_slip
sacl   tmp
LT          K3_se                        ; TREG = K3            (Q15)
MPY         speed_es       ; PREG = K3*w_psi_r(k-1)   (Q30)
PAC                                  ; ACC =  K3*w_psi_r(k-1)   (Q30)
LT          K4_se                ; TREG = K4            (Q15)
MPY         tmp       ; PREG = K4*tmp1_se       (Q30)
APAC                ; ACC = K3*w_psi_r(k-1)+K4*tmp1_se  (Q30)
 SACH       speed_es,1
; Update the rotor flux angle
ldp         #theta_fluxS
LACC theta_fluxS          ; ACC = theta_r_se          (Q15)
SACL theta_fluxS_old     ; theta_r_old = theta_r_se     (Q15)
lacc   FluxS_speed
sub         FluxS_speed_old
sacl   temp
lacc   temp
abs
sub         #150h      ; condition to prevent over acceration of fluxS
bcnd   sat_fluxS_speed,GT
b           fluxS_speed_ok
sat_fluxS_speed
lacc   FluxS_speed_old
sacl   FluxS_speed
fluxS_speed_ok
b          theta_ok
lowFluxS
ldp         #theta_fluxS
lacc   theta_fluxS
sacl   theta_fluxS_old
add         #01h
and         #0FFFh
*           sacl          #04h
sacl   theta_fluxS
theta_ok
************ now We use angle to tranform any variable to fluxR axis **
ldp         #theta_fluxS
lacc   theta_fluxS
rpt    #3
sfr
and    #0ffh  ;now ACC contains the pointer to access the table
```

```
sacl    indice1         ;
add     #sintab         ;
sacl    tmp             ;
lar     ar5,tmp         ;
nop
nop                     ;
mar     *,ar5
lacl    *               ;
nop
sacl    sin_fluxS       ;now we have sine value
lacl    indice1         ;the same thing for cosine ... cos(theta) =
;sin(theta+90°)
add     #040h           ;90 degrees = 40h elements of the table
and     #0ffh  ;
sacl    indice1         ;we use the same pointer (we don't care)
add     #sintab         ;
sacl    tmp             ;
lar     ar5,tmp         ;
lacc    *               ;
sacl    cos_fluxS       ;now we have cosine value
;***********************************************
;
* d-axis and q-axis stator flux calculation in rotor flux flame
* (alfa,beta) -> (d,q) axis transformation
* FluxSdr = FluxSD * cos(thetafluxR) + FluxSQ * sin(thetaflux_R)
* FluxSqr =-FluxSD * sin(thetafluxR) + FluxSQ * cos(thetaflux_R)
*************************************
ldp             #FluxSQ
lacc    #0
lt              FluxSQ          ;TREG0=ibeta
mpy     sin_fluxS       ;PREG=ibeta*sin(theta)
lta     FluxSD          ;ACC+=PREG ; TREG0=ialfa
mpy     cos_fluxS       ;PREG=ialfa*cos(theta)
apac                    ;ACC+=PREG
sach    FluxS_mag,4   ; value of stator current in rotor flux q component
*****************************************************
****************************************
********* save to data external ************
ldp             #count
lacc    count    ;loop control save
sub             #1
sacl    count
 bcnd   not,GT
```

```
ldp             #tmp
*               splk    #0400h,Vqr
mar             *,ar7   ;address e000h-ffffh
sar             ar7,tmp
lacc    tmp
bcnd    nosave,gt
ldp     #n
lacl    n
ldp             #tmp
sacl    *    ; 0e000h-0ffffh
lacl    tmp             ;;subtract ar with 2000h
sub             #2000h
sacl    tmp
lar             ar7,tmp
ldp     #n_es
lacl    n_es
ldp             #tmp
sacl    *    ; 0D000h-0Dfffh
lacl    tmp             ;;subtract ar with 2000h
sub             #2000h
sacl    tmp
lar             ar7,tmp
*               mar             *,ar7   ;address c000h-cfffh
ldp             #ialfa
lacl    ialfa
ldp             #tmp
sacl    *
lacl    tmp             ;;subtract ar with 2000h
sub             #2000h
sacl    tmp
lar             ar7,tmp
*               mar             *,ar3   ;address a000h-bfffh
*               mar             *,ar2   ;address 8000h-9fffh
ldp             #ibeta
lacl    ibeta
ldp             #tmp
sacl    *+
sar             ar7,tmp
lacl    tmp             ;;add ar with 6000h
add             #6000h
sacl    tmp
lar             ar7,tmp
```

```
lacl    #2
ldp             #count
sacl    count
nosave
not
****************************************
***********************************************
* Phase 1(=a) 2(=b) 3(=c) Voltage calculation
* (alfa,beta) -> (a,b,c) axis transformation
* modified exchanging alfa axis with beta axis
* for a correct sector calculation in SVPWM
* Va = vSbe_ref
* Vb = (-vSbe_ref + sqrt(3) * vSal_ref) / 2
* Vc = (-vSbe_ref - sqrt(3) * vSal_ref) / 2
***********************************************
ldp             #Valfar
lt              Valfar ;TREG0=vSal_ref
mpy     SQRT32 ;PREG=vSal_ref*(SQRT(3)/2)
pac                     ;ACC=PREG
sub     Vbetar,11 ;ACC-=vSbe_ref*2^11
sach    Vb,4
pac             ;ACC=PREG
neg             ;ACC=-ACC
sub     Vbetar,11 ;ACC-=vSbe_ref*2^11
sach    Vc,4
lacl    Vbetar ;ACC=vSbe_ref
sacl    Va ;Va=ACCL
****************************************
* SPACE VECTOR Pulse Width Modulation
* (see SVPWM references)
****************************************
lt              VDCinvT
mpy     SQRT32
pac
sach    tmp,4
lt              tmp
mpy     Vbetar
pac
sach    X,4
lacc    X ;ACC = vSbe_ref*K1
sach    accb
sacl    accb+1 ;ACCB = vSbe_ref*K1
sacl    X,1 ;X=2*vSbe_ref*K1
lt              VDCinvT
splk    #1800h,tmp
mpy     tmp ;implement mpy #01800h
pac
sach    tmp,4
lt              tmp
mpy     Valfar
pac
sach    tmp,4
lacc    tmp ;reload ACC with vSal_ref*K2
add     accb+1
add     accb,16
sacl    Y ;Y = K1 * vSbe_ref + K2 * vSal_ref
sub     tmp,1
sacl    Z ;Z = K1 * vSbe_ref - K2 * vSal_ref
*** 60 degrees sector determination
lacl    #0
sacl    sector
lacc    Va
bcnd    Va_neg,LEQ ;If Va<0 do not set bit 1 of sector
lacc    sector
or              #1
sacl    sector ;implement opl #1,sector
Va_neg          lacc    Vb
bcnd    Vb_neg,LEQ ;If Vb<0 do not set bit 2 of sector
lacc    sector
or              #2
sacl    sector ;implement opl #2,sector
Vb_neg          lacc    Vc
bcnd    Vc_neg,LEQ ;If Vc<0 do not set bit 3 of sector
lacc    sector
or              #4
sacl    sector ;implement opl #4,sector
Vc_neg
*** END 60 degrees sector determination
*** T1 and T2 (= t1 and t2) calculation depending on the sector number
lacl    sector ;(see SPACE VECTOR Modulation references for ;details)
sub     #1
bcnd    no1,NEQ
lacc    Z
```

```
sacl   t1
lacc   Y
sacl   t2
b              t1t2out
no1
lacl   sector
sub    #2
bcnd   no2,NEQ
lacc   Y
sacl   t1
lacc   X
neg
sacl   t2
b              t1t2out
no2
lacl   sector
sub    #3
bcnd   no3,NEQ
lacc   Z
neg
sacl   t1
lacc   X
sacl   t2
b              t1t2out
no3
lacl   sector
sub    #4
bcnd   no4,NEQ
lacc   X
neg
sacl   t1
lacc   Z
sacl   t2
b              t1t2out
no4
lacl   sector
sub    #5
bcnd   no5,NEQ
lacc   X
sacl   t1
lacc   Y
neg
sacl   t2
b              t1t2out
no5
lacc   Y
neg
sacl   t1
lacc   Z
neg
sacl   t2
t1t2out
lacc   t1 ;t1 and t2 minumum values must be Tonmax
sub    #Tonmax
bcnd   t1_ok,GEQ ;if t1>Tonmax then t1_ok
lacl   #Tonmax
sacl   t1
t1_ok
lacc   t2
sub    #Tonmax
bcnd   t2_ok,GEQ ;if t2>Tonmax then t2_ok
lacl   #Tonmax
sacl   t2
t2_ok
** END t1 and t2 calculation
lacc   t1 ;if t1+t2>2*Tonmax we have to saturate t1 and t2
add    t2 ;
sacl   tmp ;
sub    #MAXDUTY ;
bcnd   nosaturation,LT,EQ
** t1 and t2 saturation
lacc   #MAXDUTY,15 ;divide MAXDUTY by (t1+t2)
rpt    #15 ;
subc   tmp ;
sacl   tmp ;
lt             tmp ;calculate saturate values of t1 and t2
mpy    t1 ;t1 (saturated)=t1*(MAXDUTY/(t1+t2))
pac            ;
sach   t1,1 ;
mpy    t2 ;t2 (saturated)=t2*(MAXDUTY/(t1+t2))
pac                    ;
sach   t2,1 ;
*** END t1 and t2 saturation
nosaturation
```

```
*LOOP
** defind t1 & t2 ********
** t1 + t2 must < PWMPRD(258h)
*               ldp     #4
*               lacc    tmp_sv
*               add     #1
*               sacl    tmp_sv
*               sub     #PWMPRD
*               bcnd    tmp_sv_ok,LT
*               splk    #0,tmp_sv
*tmp_sv_ok
*               lacc    tmp_sv
*               sacl    t1
*               lacc    #PWMPRD
*               sub     t1
*               sub     #20
*               sacl    t2
*               splk    #50h,t1
*               splk    #80h,t2
*** taon,tbon and tcon calculation
lacc    #PWMPRD ;calculate the commutation instants taon,  ;tbon and
tcon
sub     t1 ;of the 3 PWM channels
sub     t2 ;taon=(PWMPRD-t1-t2)/2
sfr             ;
sacl    taon ;
add     t1 ;tbon=taon+t1
sacl    tbon ;
add     t2 ;tcon=tbon+t2
sacl    tcon ;
*** END taon,tbon and tcon calculation
*** sector switching
lacl    sector ;depending on the sector number we have
sub     #1 ;to switch the calculated taon, tbon and tcon
bcnd    nosect1,NEQ ;to the correct PWM channel
        ;(see SPACE VECTOR Modulation references for      ; details)
bldd    tbon,#CMPR1 ;sector 1
bldd    taon,#CMPR2
bldd    tcon,#CMPR3
b               dacout
nosect1
lacl    sector
sub     #2
bcnd    nosect2,NEQ
bldd    taon,#CMPR1 ;sector 2
bldd    tcon,#CMPR2 ;
bldd    tbon,#CMPR3 ;
b               dacout
nosect2
lacl    sector
sub     #3
bcnd    nosect3,NEQ
bldd    taon,#CMPR1 ;sector 3
bldd    tbon,#CMPR2 ;
bldd    tcon,#CMPR3 ;
b               dacout
nosect3
lacl    sector
sub     #4
bcnd    nosect4,NEQ
bldd    tcon,#CMPR1 ;sector 4
bldd    tbon,#CMPR2 ;
bldd    taon,#CMPR3 ;
b               dacout
nosect4
lacl    sector
sub     #5
bcnd    nosect5,NEQ
bldd    tcon,#CMPR1 ;sector 5
bldd    taon,#CMPR2 ;
bldd    tbon,#CMPR3 ;
b               dacout
nosect5
bldd    tbon,#CMPR1 ;sector 6
bldd    tcon,#CMPR2 ;
bldd    taon,#CMPR3 ;
*** END sector switching
*** END * SPACE VECTOR Pulse Width Modulation
dacout
*** END DAC out channel 'da1'
b       ContextRestoreReturn
```

ประวัติผู้วิจัย

ชื่อ : นายเอกชัย พรรณวัลย์

ชื่อวิทยานิพนธ์ : การควบคุมมอเตอร์เหนี่ยวนำแบบแรงบิดโดยตรงชนิดไร้ตัวตรวจจับความเร็วรอบ โดยใช้ตัวสังเกตการณ์แบบสไลดิงร่วมกับตัวชดเชยแบบ PI

สาขาวิชา : วิศวกรรมไฟฟ้า

ประวัติ

ประวัติส่วนตัว เกิดเมื่อวันที่ 18 กันยายน พ.ศ. 2521 ปัจจุบันอยู่บ้านเลขที่ 161 ซ.วงศ์สว่าง 11 แขวงบางซื่อ เขตบางซื่อ จังหวัดกรุงเทพฯ บิดาชื่อ นายศิริพงศ์ มารดาชื่อ นางกนกรัตน์ มีพี่น้องทั้งหมด 4 คน เป็นบุตรคนโต จบการศึกษาระดับประกาศนียบัตรวิชาชีพและประกาศนียบัตรวิชาชีพชั้นสูงจากวิทยาลัยเทคนิคลพบุรี จบการศึกษาระดับปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือปีการศึกษา 2544 ปัจจุบันกำลังศึกษาอยู่คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ