

ภาคผนวก ก

ค่าพิกัดและค่าพารามิเตอร์ของมอเตอร์

ก.1 มอเตอร์เหนี่ยวนำที่ใช้ในงานวิจัยนี้มีค่าพิกัดและค่าพารามิเตอร์ดังนี้

พิกัดมอเตอร์ $2HP, 220/380V, 50Hz, 6.3/3.7A, 1430rpm, 4poles$

แรงบิดพิกัด: $T_{rated} = 10Nm$

กระแสสร้างแรงบิดพิกัด: $i_{sq(rated)} = 9.25A$

กระแสสร้างฟลักซ์แม่เหล็กพิกัด: $i_{sd(rated)} = 5.788A$

ความเร็วพิกัด: $1,430rpm$

จำนวนคู่ของขั้วแม่เหล็ก: $p = 2$

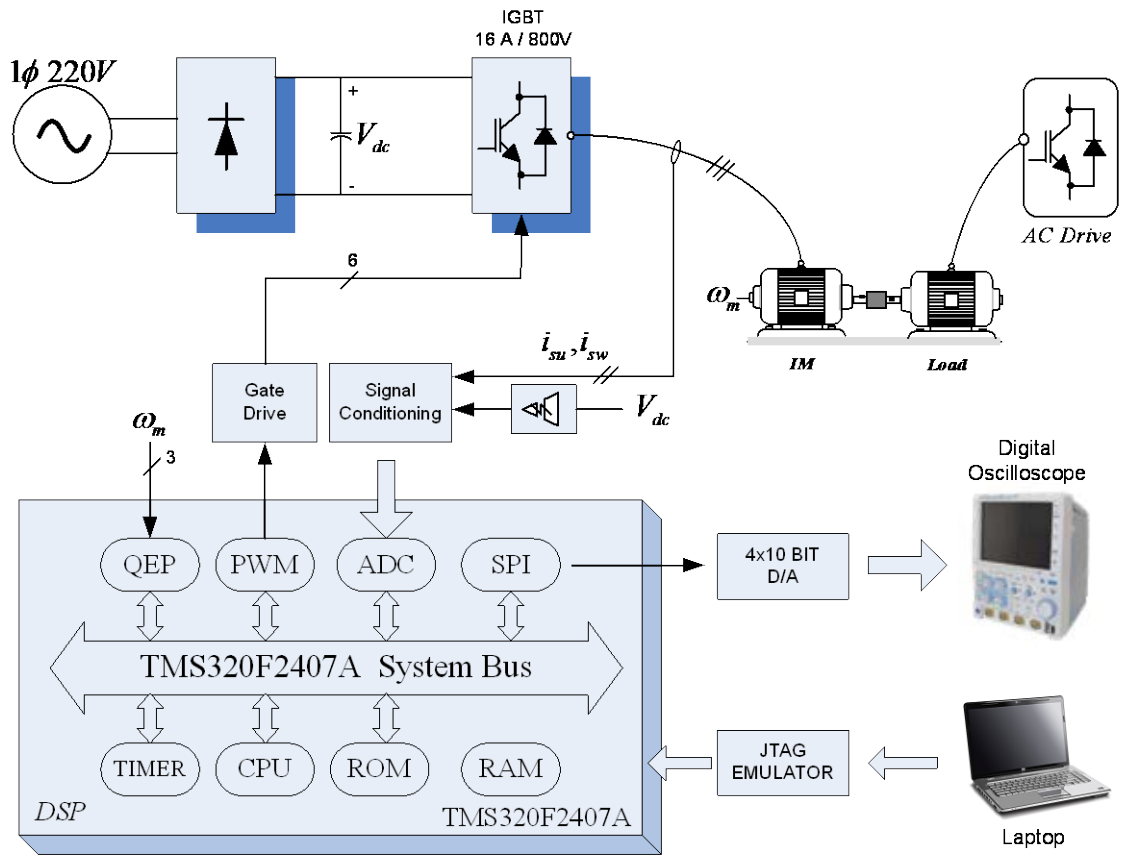
ค่าความต้านทานสเตเตอร์: $R_s = 2.15\Omega$

ค่าความต้านทานโรเตอร์: $R_r = 0.963\Omega$

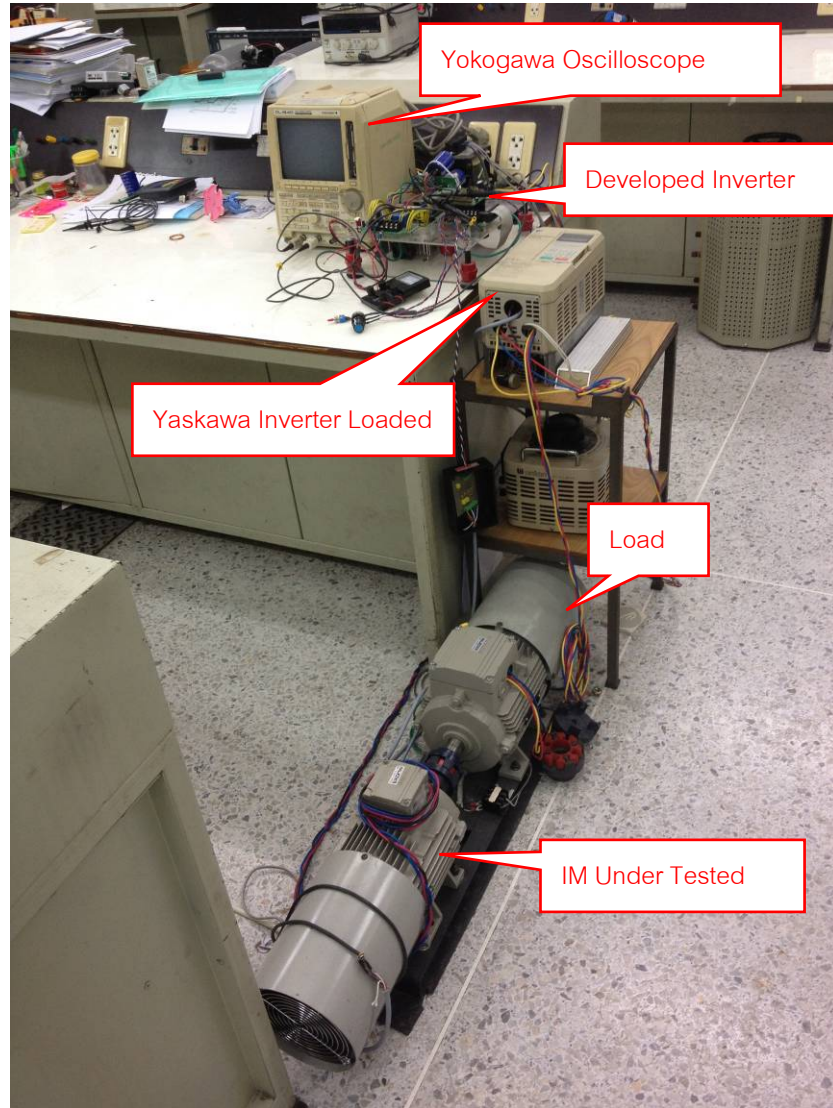
ค่าความเหนี่ยวนำสเตเตอร์: $L_s = 104.9mH$

ค่าความเหนี่ยวนำร่วม: $M = L_r = 21.5mH$

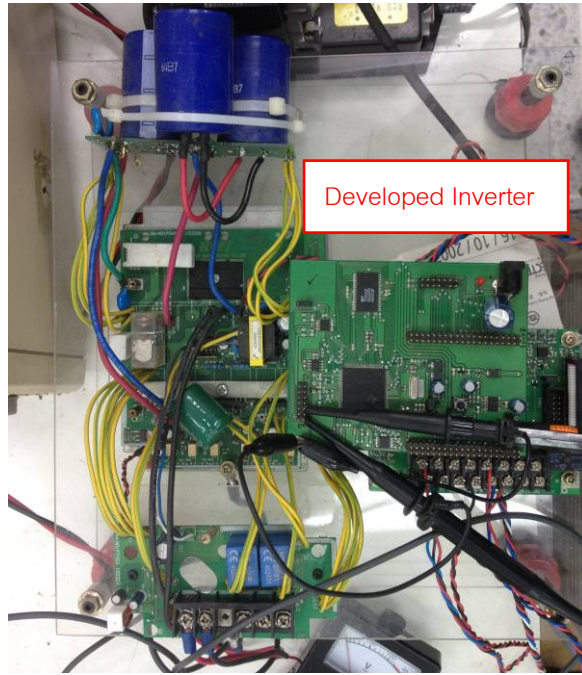
ค่าโมเมนต์ของความเฉื่อย: $J = 0.021kg.m^2$



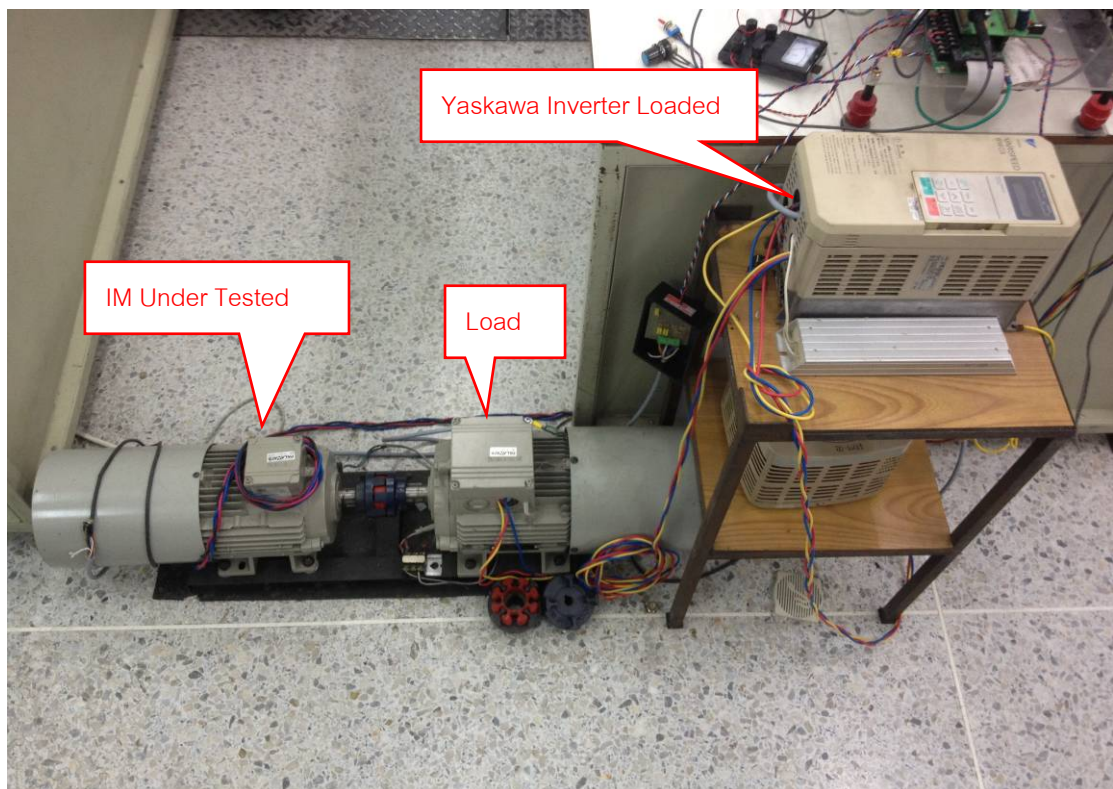
รูป ก.1 โครงสร้างทางฮาร์ดแวร์ของระบบในวิทยานิพนธ์นี้



รูป ก.2 ฮาร์ดแวร์จริงที่ใช้ในวิทยานิพนธ์



รูป ก.3 ฮาร์ดแวร์จริงที่ใช้ในวิทยานิพนธ์นี้ (ต่อ 1)



รูป ก.4 ฮาร์ดแวร์จริงที่ใช้ในวิทยานิพนธ์นี้ (ต่อ 2)

ก.2 ซอฟต์แวร์แวร์ของระบบ

จากโครงสร้างส่วนการควบคุมในรูปที่ 2.4 ตัวประมวลผลสัญญาณดิจิทัลจะทำการคำนวณกระแสสร้างแรงบิดคำสั่ง (i_{sq}^*) จากผลต่างระหว่างความเร็วคำสั่งกับความเร็วประมาณ ผ่านตัวควบคุม PI ที่มีการจำกัดค่ากระแสคำสั่ง ไม่ให้เกินค่าพิกัด กระแสคำสั่ง i_{sq}^* ที่คำนวณได้ จะถูกส่งไปยังตัวควบคุมเวกเตอร์ไร้เซนเซอร์วัดความเร็ว ซึ่งจะประมาณค่าความเร็วจากค่าผิดพลาดระหว่างกระแสประมาณกับกระแสเดเตอร์ที่ตรวจจับ เพื่อใช้ในวงรอบควบคุมความเร็วด้านนอก และคำนวณค่าแรงดันสำหรับการควบคุมแยกการเชื่อมร่วมในส่วนของระบบควบคุมเวกเตอร์ ค่าแรงดันคำสั่งที่ได้จะถูกนำไปสร้างสัญญาณปรับความกว้างพัลส์ (PWM) สำหรับขับนำเกตของอินเวอร์เตอร์โดยอาศัยหลักการทางสเปซเวกเตอร์ของแรงดัน (Voltage space vector) ซึ่งจะใช้แรงดันบัสไฟตรงที่ตรวจจับได้เป็นแรงดันฐาน ซอฟต์แวร์ทั้งหมดสามารถเขียนได้ดังแสดงใน PDL (Program Development Language) ต่อไปนี้ และสามารถแสดงไคอะแกรมเวลาได้ดังรูปที่ ก.5 ซอฟต์แวร์โมดูลนี้จะใช้การอินเทอร์รัปต์ทุกๆ $200 \mu s$ ซึ่งจะเห็นว่าจะทำการอ่านกระแสก่อนเป็นอันดับแรก ทั้งนี้เพื่อให้กระแสที่อ่านได้ใกล้เคียงกับกระแสที่ความถี่หลักมูลมากที่สุด

SPEED-SENSORLESS VECTOR CONTROL PROGRAM
OF A INDUCTION MOTOR (MAIN PROGRAM)

MODULE : MAIN PROGRAM

Initialize

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

Loop here and wait for interrupt only

Switching frequency Interrupt Service Routine

Analog to Digital conversion

Input i_{su}, i_{sv}, E_{dc} from A/D

Convert to rotating $\hat{d} - \hat{q}$ axis ($i_{s\hat{d}}, i_{s\hat{q}}$)

Get speed command

Get estimated speed from previous interrupt service routine

Speed regulator

Calculate speed error

Calculate Speed Controller output ($i_{s\hat{q}}^*$)

Stator dynamics

Calculate estimated currents ($\hat{i}_{s\hat{d}}, \hat{i}_{s\hat{q}}$)

Adaptive Controller

Calculate current error ($\hat{i}_{s\hat{q}} - i_{s\hat{q}}$)

Calculate estimated speed $\hat{\omega}_m$

Decoupling control

Calculate u_{sd}^*, u_{sq}^*

Generate PWM signal

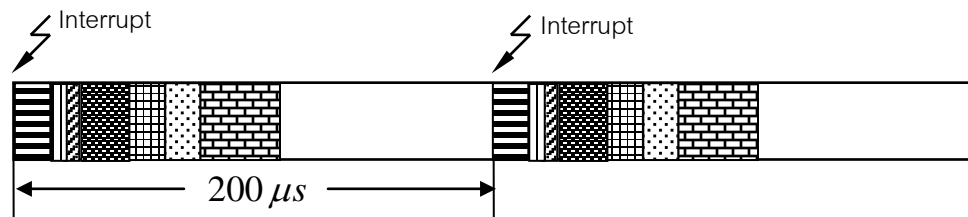
Calculate zero sequence voltage and add it to phase voltage command


Calculate compare register


Digital to Analog conversion


Return


END MAIN PROGRAM




 Analog to Digital conversion (17 μs)


 Speed regulator (3.4 μs)

 Stator dynamic (2 μs)

 Adaptive controller (16 μs)

 Decoupling control (8 μs)

 Generate PWM signal (10 μs)

 Digital to Analog conversion (SPI) (40 μs)

รูปที่ ก.5 ไคอะแกรมเวลาของซอฟต์แวร์โมดูล