

รายงานการวิจัย

การพัฒนาซอฟต์แวร์สำหรับการวินิจฉัยฟอลต์ในมอเตอร์เหนี่ยวนำ
DEVELOPMENT OF A SOFTWARE TOOL FOR FAULT
DIAGNOSIS IN INDUCTION MOTORS



ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณเงินรายได้ ประจำปีงบประมาณ 2552

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รายงานการวิจัย

การพัฒนาซอฟต์แวร์สำหรับการวินิจฉัยฟอลต์ในมอเตอร์เหนี่ยวนำ
**DEVELOPMENT OF A SOFTWARE TOOL FOR FAULT
DIAGNOSIS IN INDUCTION MOTORS**



RCM
QA
76-76
.D47
D 4177
D. 2

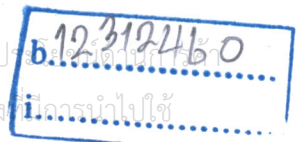
สงพ.....
เลขทะเบียน 115566
น.เดือน.ปี. 21 ส.ค. 2554

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณเงินรายได้ ประจำปีงบประมาณ 2552

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง



กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้สนับสนุนทุนวิจัยจากเงินงบประมาณเงินรายได้ ประจำปีงบประมาณ 2552 ในครั้งนี้จนกระทั่งงานวิจัยเล่มนี้สำเร็จลงด้วยดี ขอขอบคุณนายพงศกร จิรลาภผล และนายธีรวุฒิ คุหาเกษมสิน ที่มีส่วนช่วยในการเขียนและออกแบบโปรแกรมคอมพิวเตอร์ และขอขอบคุณสาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เป็นแหล่งสนับสนุนทางด้านเครื่องมือ อุปกรณ์ในการทำวิจัย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อภาษาไทยและบทคัดย่อภาษาอังกฤษ

ส่วนที่ 1 รายละเอียดเกี่ยวกับโครงการ

ชื่อโครงการการพัฒนาซอฟต์แวร์สำหรับการวินิจฉัยฟอลต์ในมอเตอร์เหนี่ยวนำ..

.....Development of a software tool for fault diagnosis in induction motors.....

ได้รับทุนอุดหนุนการวิจัยจาก.....คณะวิศวกรรมศาสตร์...ด้วยงบประมาณเงินรายได้.....

ประจำปี.....2552..... จำนวนเงิน.....100,000 บาท.....

ระยะเวลาทำการวิจัย.....1..... ปี ตั้งแต่.....19 มกราคม 52.....ถึง.....20 มกราคม 53.....

หน่วยงานและผู้ดำเนินการวิจัยพร้อมหน่วยงานที่สังกัดและเลขหมายโทรศัพท์

.....สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์.....

.....สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.....

ส่วนที่ 2 บทคัดย่อ

โครงการวิจัยนี้นำเสนอเครื่องมือซอฟต์แวร์และฮาร์ดแวร์ สำหรับการวินิจฉัยฟอลต์ในเครื่องจักรกลไฟฟ้า ซึ่งฟอลต์ที่เกิดขึ้นในเครื่องจักรกลไฟฟ้าที่จะทำการตรวจจับและวินิจฉัยคือ ฟอลต์สเตเตอร์ในสภาวะการลัดวงจรระหว่างรอบถึงรอบในขดลวดสเตเตอร์ ซึ่งเป็นฟอลต์ที่เกิดขึ้นเริ่มแรกก่อนที่จะเป็นฟอลต์ที่รุนแรงต่อไป เทคนิคการตรวจจับที่นำมาใช้นี้คือองค์ประกอบแรงดันลำดับศูนย์ โครงสร้างของชุดทดสอบที่ได้ออกแบบขึ้นประกอบด้วยส่วนของฮาร์ดแวร์และซอฟต์แวร์ โดยส่วนฮาร์ดแวร์ประกอบด้วยตัวกรองสัญญาณรบกวน และชุดรับสัญญาณแรงดัน ในส่วนซอฟต์แวร์เขียนด้วยโปรแกรม LabVIEW สำหรับแสดงข้อมูลของแรงดันที่รับเข้ามา ผลการคำนวณองค์ประกอบแรงดันลำดับศูนย์และการแสดงผลต่าง ๆ จากการทดลองกับมอเตอร์เหนี่ยวนำพิกัด 2 แรงม้าสำหรับการวินิจฉัยพบว่าสามารถตรวจจับฟอลต์สเตเตอร์ได้อย่างแม่นยำ

This research presents software and hardware tools for diagnosis of fault in electrical machines. The focus condition is turn-to-turn short-circuit in stator winding beginning fault before to be severe. The presented detection technique is zero sequence voltage components. Test set structure consists of two main parts, hardware and software. The hardware part is composed of noise filter and voltage sensor. The second one is implemented with LABVIEW for showing voltage data, calculating zero sequence voltage components etc. From experimental results, a 2-hp induction motor was used for diagnosis which is capable of accurately identifying percentage of stator fault.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญเรื่อง

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อภาษาไทยและบทคัดย่อภาษาอังกฤษ	II
สารบัญเรื่อง	III
สารบัญตาราง	V
สารบัญรูป	VI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตการวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 มอเตอร์เหนี่ยวนำ	3
2.1 ทฤษฎีมอเตอร์เหนี่ยวนำ	3
2.1.1 โรเตอร์แบบกรงกระรอก	3
2.1.2 โรเตอร์ชนิดลวดพัน	3
2.2 ประเภทความล้มเหลวของขดลวดสเตเตอร์และสาเหตุ	4
2.3 ผลกระทบต่อประสิทธิภาพของเครื่องจักรกลไฟฟ้า	7
2.4 การวิเคราะห์ขดลวดสเตเตอร์ถัดรอบ	11
บทที่ 3 องค์ประกอบแรงดันลำดับศูนย์	14
3.1 ทฤษฎีพื้นฐาน	14
3.2 ตำราวิชาการประยุกต์ใช้องค์ประกอบแรงดันลำดับศูนย์สำหรับการตรวจจับฟอลต์สเตเตอร์	16
บทที่ 4 การออกแบบและการสร้างเครื่องมือวินิจฉัยฟอลต์	20
4.1 อุปกรณ์ที่ใช้ในการทดลอง	20
4.1.1 มอเตอร์เหนี่ยวนำ 3 เฟสที่ใช้ในการทดลอง	20
4.1.2 ชุดการวัดรับส่งข้อมูล NI USB-6009	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3	โพรบแรงดัน	21
4.1.4	ชุด PLC Omron	21
4.2	ซอฟต์แวร์ LabVIEW	22
4.2.1	รายละเอียด	22
4.2.2	Menu และ Palette	30
4.2.3	HELP	40
4.3	ไดอะแกรมและวงจรของการทดลอง	43
บทที่ 5	ผลการวิจัย	45
5.1	การทดสอบที่สภาวะโหลดต่างกัน	45
5.2	การแสดงผลการเฝ้าตรวจสอบสถานะมอเตอร์	49
บทที่ 6	สรุป	52
บรรณานุกรม		53
ภาคผนวก เอกสารข้อมูลของ Data Acquisition Board, NI USB-6009		55
ประวัตินักวิจัย		78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้าที่
ตารางที่ 2.1 เปรียบเทียบเปอร์เซ็นต์การสูญเสียของมอเตอร์ 2 และ 4 ขั้วแม่เหล็ก	8
ตารางที่ 2.2 เปรียบเทียบค่าการสูญเสียในมอเตอร์เหนี่ยวนำกับการพันใหม่	9
ตารางที่ 4.1 ส่วนประกอบ LabVIEW	26
ตารางที่ 4.2 ลักษณะของเส้น wire	28
ตารางที่ 4.3 อุปกรณ์ต่าง ๆ บน Tools Palette	39



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้าที่
รูปที่ 2.1 โครงสร้างของมอเตอร์เหนี่ยวนำ 3 เฟส	4
รูปที่ 2.2 โหมคความล้ามหลวของขดลวดสเตเตอร์ในเครื่องจักรกลไฟฟ้าเมื่อขดลวดต่อแบบวาย	5
รูปที่ 2.3 รูปจริงที่เกิดจากความล้ามหลวของขดลวดสเตเตอร์ในลักษณะต่าง ๆ	6
รูปที่ 2.4 การเปรียบเทียบแกนสเตเตอร์ (ซ้าย) ดีและ (ขวา) ตัวนำสเตเตอร์ใหม่	7
รูปที่ 2.5 ลักษณะของการลงขดลวดสเตเตอร์ (ซ้าย) ไม่ดีและ (ขวา) ดี	7
รูปที่ 2.6 การลัดวงจรในขดลวดสเตเตอร์ระหว่างรอบถึงรอบที่จุด a กับ b	11
รูปที่ 2.7 การเปลี่ยนแปลงของฟลักซ์กรณีเกิดการลัดวงจรระหว่างรอบถึงรอบ	12
รูปที่ 3.1 วิธีการวัดสัญญาณองค์ประกอบลำดับศูนย์เมื่อ (บน) มอเตอร์ต่อแบบวายและ (ล่าง) มอเตอร์ต่อแบบเดลต้า	15
รูปที่ 3.2 วิธีการวัดสัญญาณองค์ประกอบลำดับศูนย์เมื่อ (บน) มอเตอร์ต่อแบบวายและ (ล่าง) มอเตอร์ต่อแบบเดลต้า	16
รูปที่ 3.3 สัญญาณองค์ประกอบลำดับศูนย์เมื่อมอเตอร์ทำงานที่สภาวะพิกัด โหลดและแรงดัน (บน) สัญญาณองค์ประกอบแรงดันลำดับศูนย์เมื่อมอเตอร์ต่อแบบวายและ (ล่าง) สัญญาณองค์ประกอบกระแสลำดับศูนย์เมื่อมอเตอร์ต่อแบบเดลต้า	17
รูปที่ 3.4 สเปกตร้าของสัญญาณองค์ประกอบแรงดันลำดับศูนย์ (ซ้าย) ในสภาวะมอเตอร์สุขภาพดี และ (ขวา) มอเตอร์มีความผิดปกติ เมื่อมอเตอร์ทำงานที่สภาวะพิกัด โหลดและแรงดัน	17
รูปที่ 3.5 สเปกตร้าของสัญญาณองค์ประกอบกระแสลำดับศูนย์ (ซ้าย) ในสภาวะมอเตอร์สุขภาพดี และ (ขวา) มอเตอร์มีความผิดปกติ เมื่อมอเตอร์ทำงานที่สภาวะพิกัด โหลดและแรงดัน	17
รูปที่ 3.6 สัญญาณองค์ประกอบแรงดันลำดับศูนย์หลังจากหยุดจ่ายไฟให้กับมอเตอร์ (บน) มอเตอร์สุขภาพดีและ (ล่าง) เมื่อมอเตอร์มีความผิดปกติ	18
รูปที่ 3.7 การประเมินคุณลักษณะด้านขนาดของสัญญาณองค์ประกอบแรงดันลำดับศูนย์ ซึ่งคำนวณจากสัญญาณแรงดันหลังจากหยุดจ่ายไฟภายใต้การทดสอบ (บน) ที่สภาวะไร้โหลดและ (ล่าง) ที่พิกัด โหลด	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

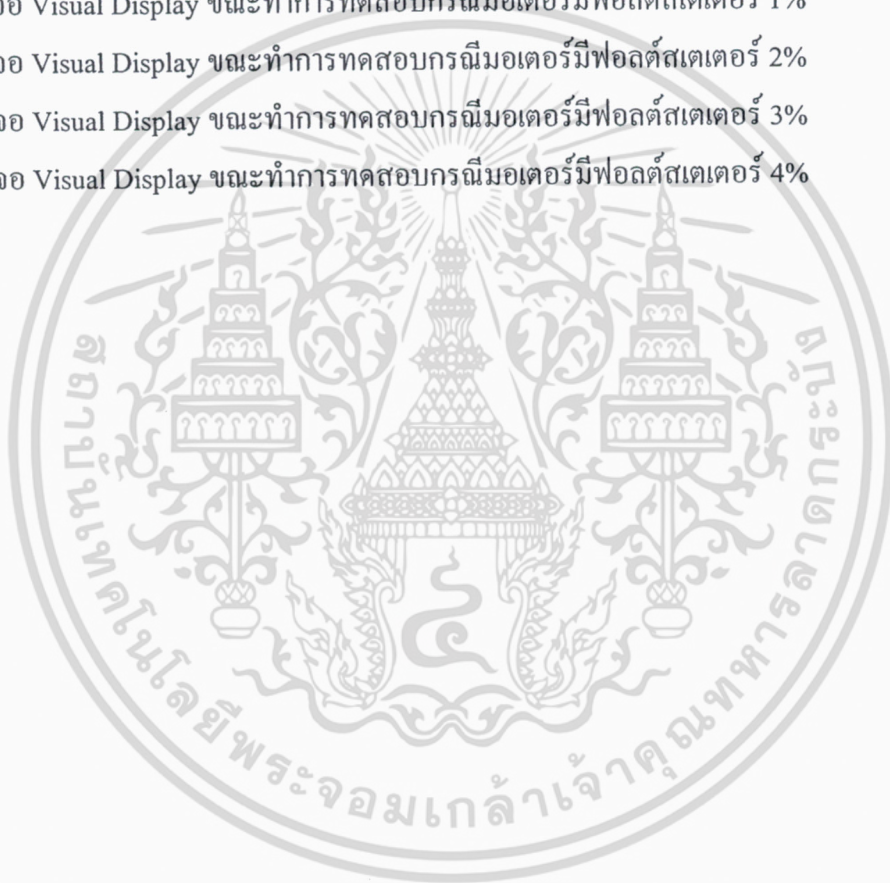
สารบัญรูป (ต่อ)

	หน้าที่
รูปที่ 4.1 ชุดทดลองมอเตอร์เหนี่ยวนำ 3 เฟสและมอเตอร์โหลด	20
รูปที่ 4.2 Data Acquisition Board, NI USB-6009	21
รูปที่ 4.3 Differential Probe	21
รูปที่ 4.4 ชุด PLC Omron CPM1A	22
รูปที่ 4.5 Front Panel	24
รูปที่ 4.6 Block Diagram	25
รูปที่ 4.7 Connector ของ subVI	25
รูปที่ 4.8 ลักษณะของ Node และ Terminal ที่บรรจุอยู่ใน Block Diagram	27
รูปที่ 4.9 ไอคอนและ Connector	30
รูปที่ 4.10 หน้าต่างของ LabVIEW	31
รูปที่ 4.11 ลักษณะของ Pop-up menu ของแต่ละ Object	35
รูปที่ 4.12 ลักษณะของ Controls และ Functions Palette	38
รูปที่ 4.13 ส่วนประกอบของ Sub Palette	38
รูปที่ 4.14 ตัวอย่าง Help	41
รูปที่ 4.15 ตัวอย่าง Simple Help	42
รูปที่ 4.16 ตัวอย่าง Details Help	42
รูปที่ 4.17 ไดอะแกรมการทดลอง	42
รูปที่ 4.18 วงจรการวัดและประมวลผล	43
รูปที่ 4.19 อุปกรณ์ที่ใช้ในการทำงานของระบบ	44
รูปที่ 4.20 ชุดทดสอบทั้งหมด	44
รูปที่ 5.1 ลักษณะของการลัดวงจรระหว่างรอบถึงรอบที่ใช้ในการทดลอง	45
รูปที่ 5.2 สัญญาณองค์ประกอบแรงดันลำดับศูนย์กรณีทดสอบที่สภาวะไร้อะไรโหลดด้วย ขดลวดสเตเตอร์ (บนซ้าย) ลัดรอบ 1% (บนขวา) ลัดรอบ 2% (ล่างซ้าย) ลัดรอบ 3% และ (ล่างขวา) ลัดรอบ 4%	46
รูปที่ 5.3 สัญญาณองค์ประกอบแรงดันลำดับศูนย์กรณีทดสอบที่สภาวะโหลด 50% ด้วย ขดลวดสเตเตอร์ (บนซ้าย) ลัดรอบ 1% (บนขวา) ลัดรอบ 2% (ล่างซ้าย) ลัดรอบ 3% และ (ล่างขวา) ลัดรอบ 4%	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้าที่
รูปที่ 5.4 สัญลักษณ์ประกอบแรงดันลำดับศูนย์กรณีทดสอบที่สภาวะโหลด 100% ด้วยขดลวดสเตเตอร์ (บนซ้าย) ลัดรอบ 1% (บนขวา) ลัดรอบ 2% (ล่างซ้าย) ลัดรอบ 3% และ (ล่างขวา) ลัดรอบ 4%	48
รูปที่ 5.5 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์สุขภาพดี	49
รูปที่ 5.6 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์มีฟอลต์สเตเตอร์ 1%	49
รูปที่ 5.7 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์มีฟอลต์สเตเตอร์ 2%	50
รูปที่ 5.8 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์มีฟอลต์สเตเตอร์ 3%	50
รูปที่ 5.9 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์มีฟอลต์สเตเตอร์ 4%	51



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

มอเตอร์ไฟฟ้าเหนี่ยวนำ (Induction Motor) โดยเฉพาะที่ใช้กับระบบไฟ 3 เฟส เป็นเครื่องจักรกลไฟฟ้าที่ถูกนำมาใช้งานเป็นจำนวนมาก เนื่องจากมีคุณลักษณะที่ดีหลายด้านด้วยกันที่สำคัญ ๆ ได้แก่ มีโครงสร้างที่แข็งแรงทนทาน สามารถทำงานในหลายสภาวะแวดล้อม การบำรุงรักษาค่า มีราคาถูกกว่าเครื่องจักรกลไฟฟ้าชนิดอื่น และสามารถควบคุมการทำงานให้มีประสิทธิภาพที่สูงได้ อย่างไรก็ตาม ปัญหาส่วนประกอบของมอเตอร์เองที่เป็นจุดอ่อนที่อาจทำให้เกิดความเสียหายได้ ในปัจจุบันวิธีการตรวจสอบในแต่ละปัญหาของมอเตอร์จะมีด้วยกันหลายวิธี ซึ่งมีข้อดีและข้อเสียที่แตกต่างกันไป เช่น สามารถตรวจสอบได้ในบางกรณี หรือสามารถตรวจจับสิ่งผิดปกติได้เมื่อความเสียหายเกิดขึ้นรุนแรงแล้ว เป็นต้น จากสถิติที่ได้มีการวิจัยไว้พบว่าลวดตัวนำสเตเตอร์มักจะพบกับความผิดปกติได้บ่อยครั้งถึงประมาณ 40% ของความผิดปกติทั้งหมดที่เกิดขึ้นกับเครื่องจักรกลไฟฟ้า [1-2] ความผิดปกติดังกล่าวมักจะเกิดขึ้นเนื่องจากขดลวดสเตเตอร์ลัดวงจรหรือเรียกเป็น ฟอลต์สเตเตอร์ (Stator fault) สภาวะการลัดวงจรของขดลวดสเตเตอร์ที่มักพบได้บ่อยคือการลัดวงจรระหว่างรอบถึงรอบ สาเหตุในเบื้องต้นที่ทำให้เกิดการลัดวงจรระหว่างรอบถึงรอบคือการที่ระบบฉนวนถูกทำลาย โดยมีสาเหตุมาจากฉนวนขดลวดสกปรก การระบายความร้อนของเครื่องจักรกลที่ไม่มีประสิทธิภาพ การสั้นของเครื่องจักรกลไฟฟ้า โดยเฉพาะในขณะที่ทำงาน ระบบทางไฟฟ้าหรือสภาพแวดล้อมที่แย่ เช่น สภาพแวดล้อมที่เต็มไปด้วยฝุ่นละออง สารเคมี การหมุนเวียนของอากาศภายในโรงงานที่ไม่ดี เป็นต้น [3] ซึ่งปัญหาเหล่านี้จะทำให้ตำแหน่งที่อ่อนแอในขดลวดเป็นตำแหน่งที่เกิดความร้อนจนกระทั่งที่ตำแหน่งนั้นฉนวนถูกทำลาย และกลายมาเป็นความเสียหายที่รุนแรงยิ่งขึ้นไปอีก ซึ่งนั่นก็คือปัญหาของขดลวดไหม้และเครื่องจักรกลไม่สามารถทำงานได้อีกต่อไป ดังนั้นการตรวจจับความผิดปกติในขั้นแรกสุดนี้จึงเป็นสิ่งสำคัญมาก ทั้งนี้วงจรป้องกันโดยทั่วไปแล้วยังไม่สามารถตรวจจับความผิดปกติในสภาวะเริ่มต้นนี้ได้

จากที่กล่าวมามอเตอร์ไฟฟ้านั้นจึงจำเป็นต้องมีระบบตรวจสอบความผิดปกติที่มีประสิทธิภาพสำหรับเป็นเครื่องเตือนให้การเฝ้าระวังมอเตอร์ ทั้งนี้การตรวจสอบปัญหาหรือการป้องกันอันตรายที่จะเกิดแก่ตัวมอเตอร์นั้นจะเห็นได้ว่า จำเป็นต้องใช้วิธีการและอุปกรณ์หลาย ๆ ชนิดซึ่งแยกจากกัน จึงก่อให้เกิดความยุ่งยากในการติดตั้ง อีกทั้งเป็นการสิ้นเปลืองค่าใช้จ่ายในการจัดซื้ออุปกรณ์หลาย ๆ ชนิด ประการหนึ่งซึ่งเป็นที่ทราบกันดีคือการพัฒนาของโปรแกรมต่าง ๆ ที่มีความสามารถหรือมีประสิทธิภาพสูงขึ้นในปัจจุบัน ทำให้เทคโนโลยีซอฟต์แวร์ (Software) มีความสามารถมากยิ่งขึ้นตามไปด้วย ทั้งนี้จะสังเกตได้จากการที่มีนักวิจัยหลายท่านที่ได้หันมาใช้ในการพัฒนาซอฟต์แวร์เป็นเครื่องมือวัดเทียบเคียง (Virtual instrumentation) สำหรับการวัดค่าสัญญาณต่าง ๆ ที่เกิดขึ้นได้อย่างถูกต้อง [4-5] ด้วยเหตุผลที่

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวมาข้างต้นจึงเป็นที่มาของโครงการวิจัยนี้ คือการใช้ประโยชน์จากซอฟต์แวร์ที่เขียนขึ้นสำหรับการวินิจฉัยฟอลต์ในมอเตอร์เหนี่ยวนำ

1.2 วัตถุประสงค์

- 1.2.1 เพื่อออกแบบสร้างซอฟต์แวร์สำหรับใช้เป็นเครื่องมือการวินิจฉัยฟอลต์สเตเตอร์ที่เกิดขึ้นในมอเตอร์เหนี่ยวนำ
- 1.2.2 เพื่อลดค่าใช้จ่ายในการตรวจสอบมอเตอร์แทนการจัดซื้อเครื่องมือวัดที่มีราคาแพง
- 1.2.3 เพื่อเป็นต้นแบบสำหรับการใช้ประโยชน์จากโปรแกรมคอมพิวเตอร์ในการนำมาประยุกต์ใช้งานด้วยต้นทุนที่ต่ำ

1.3 ขอบเขตการวิจัย

ซอฟต์แวร์ที่เขียนขึ้นสามารถตรวจจับฟอลต์ที่เกิดขึ้นในเครื่องจักรกลไฟฟ้าได้ โดยเน้นเฉพาะฟอลต์สเตเตอร์ที่มีเปอร์เซ็นต์ของการเกิดขึ้นอยู่ในระดับสูงจากฟอลต์ที่เกิดขึ้นทั้งหมด ซอฟต์แวร์ดังกล่าวจะถูกเขียนขึ้นโดยใช้โปรแกรม LABVIEW สัญญาณที่นำมาประมวลผลจะเป็นสัญญาณของเครื่องจักรกลไฟฟ้าเหนี่ยวนำ 3 เฟส ได้แก่ สัญญาณแรงดันทั้ง 3 เฟส โดยสัญญาณเหล่านี้จะถูกเก็บในลักษณะออนไลน์ (On-line technique) โดยผ่านชุดฮาร์ดแวร์ที่สนับสนุนโดยโปรแกรม LABVIEW ร่วมกับการประมวลผลสัญญาณดิจิทัลสำหรับลดค่าความคลาดเคลื่อนต่างๆ ที่เกิดขึ้น ทั้งนี้การพิจารณาฟอลต์จะเป็นการพิจารณาจากองค์ประกอบแรงดันลำดับศูนย์ หรือเรียกเป็นสัญญาณผลรวมแรงดัน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ทำให้เครื่องจักรกลไฟฟ้ามีอายุการใช้งานที่ยาวนานขึ้นได้
- 1.4.2 ทำให้สามารถวางแผนการซ่อมบำรุงเครื่องจักรกลไฟฟ้า โดยมีผลกระทบไม่มากนักต่อกระบวนการผลิตโดยรวมได้
- 1.4.3 เป็นจุดเริ่มต้นสำหรับการหันมาพึ่งเทคโนโลยีและบุคลากรภายในประเทศของเราเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

มอเตอร์เหนี่ยวนำ

มอเตอร์เหนี่ยวนำได้ถูกนำมาใช้ในงานอุตสาหกรรมเป็นจำนวนมาก เนื่องจากมีข้อได้เปรียบทางลักษณะของโครงสร้าง ประกอบกับเทคโนโลยีการควบคุมที่ดีเยี่ยม อย่างไรก็ตามปัญหาซึ่งควรได้รับการพัฒนาคือเทคนิคการเฝ้าตรวจสอบสถานะ (Condition monitoring) สำหรับการแจ้งเตือนและทำนายความเสียหายของมอเตอร์เหนี่ยวนำเพื่อเป็นแนวทางการบำรุงรักษาได้อย่างทันท่วงทีต่อไป

2.1 ทฤษฎีมอเตอร์เหนี่ยวนำ

มอเตอร์เหนี่ยวนำ 3 เฟส ส่วนใหญ่แล้วถูกนำมาเป็นต้นกำลังให้กับเครื่องจักรในกระบวนการผลิต โครงสร้างหลักของมอเตอร์เหนี่ยวนำประกอบด้วย ส่วนที่อยู่กับที่ (Stator) และส่วนที่เคลื่อนที่ (Rotor) ซึ่งสามารถแบ่งออกเป็น 2 ชนิดตามโครงสร้างที่ต่างกันของโรเตอร์ได้แก่ โรเตอร์กรงกระรอก (Squirrel cage rotor) และโรเตอร์ชนิดลวดพัน (Wound rotor) ดังรายละเอียดต่อไปนี้

2.1.1 โรเตอร์แบบกรงกระรอก

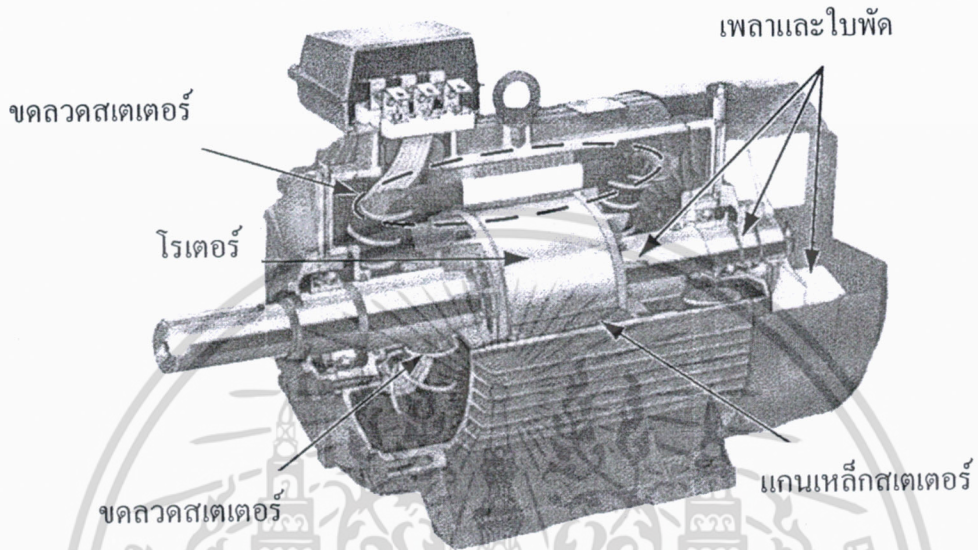
โรเตอร์ประเภทนี้ที่แกนของโรเตอร์ทำด้วยเหล็กแผ่นลามิเนต (Laminated sheet steel) เสาะผิวด้านนอกให้เป็นร่อง ตรงกลางแผ่นจะเจาะรูสำหรับสอดเพลลา เมื่อนำแผ่นลามิเนตเหล่านี้มาอัดเข้าด้วยกันจะได้แกนโรเตอร์ เมื่อนำแกนโรเตอร์เข้าไปวางในแบบพิมพ์ (Mold) แล้วฉีดอลูมิเนียมเหลวเข้าไป จะได้อลูมิเนียมอัดแน่นอยู่ในร่องสลอตและมีครีบบนออกไปทั้งสองข้าง เพื่อใช้เป็นใบพัดสำหรับระบายความร้อน และเมื่ออัดเพลลาเข้าไปที่รูกลางของแกนโรเตอร์จะได้โรเตอร์กรงกระรอก โรเตอร์ในลักษณะนี้ไม่ต้องการใช้วงแหวน (Slip ring) ติดอยู่ที่ตัวหมุน ในบางครั้งจึงเรียกมอเตอร์ชนิดนี้ว่า non-slip ring machines รูปที่ 2.1 แสดงส่วนประกอบของเครื่องจักรกลไฟฟ้าเหนี่ยวนำที่มีโรเตอร์เป็นแบบกรงกระรอก

2.1.2 โรเตอร์ชนิดลวดพัน

โรเตอร์แบบนี้ที่ตัวหมุนจะมีขลวดพันอยู่รอบ ๆ โดยที่ขั้วแม่เหล็กที่เกิดขึ้นบนตัวหมุนนี้ จะต้องมีจำนวนขั้วเท่ากับที่มีอยู่บนสเตเตอร์ ซึ่งโรเตอร์ในลักษณะนี้ต้องใช้วงแหวนติดไว้ที่บนตัวหมุน เพื่อนำปลายของขลวดที่พันอยู่บนตัวหมุนนั้นต่อออกมายังวงจรภายนอก แกนของโรเตอร์ทำด้วยเหล็กแผ่นลามิเนตป้อนเป็นร่องสลอตแล้วนำมาอัดติดกันเป็นรูปทรงกระบอก มีขลวดทองแดงอบน้ำยาฉนวนไฟฟ้าจำนวน 3 ชุด (3 เฟส) พันอยู่ในร่องสลอต โดยขลวดแต่ละเฟสจะพันเรียงห่างกัน 120 องศาไฟฟ้า และต้องพันให้เกิดจำนวนขั้วแม่เหล็กเท่ากับจำนวนขั้วแม่เหล็กบนสเตเตอร์ ปลายสายของขลวดทั้งสามเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะต่อรวมกันเป็นแบบสตาร์ ปลายสายที่เหลือจะต่อเข้ากับสลีปริง (Slip ring) 3 อัน ซึ่งมีแปรงถ่านสัมผัสอยู่ และมีสายต่อจากแปรงถ่านออกไปภายนอกเพื่อต่อเข้ากับความต้านทานที่ปรับค่าได้เพื่อใช้ในการเริ่มหมุน และปรับแต่งความเร็ว ซึ่งในบางครั้งจะเรียกมอเตอร์ชนิดนี้ว่า slip ring machines



รูปที่ 2.1 โครงสร้างของมอเตอร์เหนี่ยวนำ 3 เฟส

2.2 ประเภทความล้มเหลวของขดลวดสเตเตอร์และสาเหตุ

ปัจจัยที่มีผลต่อการชำรุดเสียหาย ของฉนวนขดลวดตัวนำสเตเตอร์จะมีสาเหตุมาจากความเค้น (Stresses) ที่กระทำกับขดลวดซึ่งทำให้เกิดการเสื่อมสภาพก่อนกำหนดที่สำคัญ ๆ มีดังต่อไปนี้ [3]

ก. ผลของความร้อน รูปแบบของความร้อนที่เกิดขึ้นแบ่งออกได้เป็น ความร้อนที่เกิดจากการเสื่อมอายุ (Aging) และความร้อนที่เกิดจากการมีโหลดเกิน (Thermal Overloading) ทั้งนี้ AIEE 510 และ IEEE 275 ได้ดำเนินการทดสอบ พบว่าอายุของฉนวนจะลดลงครึ่งหนึ่งของอุณหภูมิที่เพิ่มสูงขึ้นทุก ๆ 10 องศาเซลเซียสในแต่ละชั้น (Class) ของฉนวน ทั้งนี้อายุการใช้งานตามชั้นของฉนวนปกติจะอยู่ที่ 20,000 ชั่วโมง

ข. ผลเนื่องมาจากไฟฟ้า มักพบกับมอเตอร์ที่ใช้กับแรงดันที่เกิน 600 V ขึ้นไป โดยที่พื้นผิวของฉนวนตัวนำจะมีลักษณะของความชื้นและสิ่งสกปรกสะสม ทำให้เกิดกระแสดิซชาร์จลงกราวด์เป็นเหตุให้เกิดการเผาไหม้เป็นจุดเล็ก ๆ ทำให้ฉนวนเสื่อมสภาพลง นอกจากนี้แล้วปรากฏการณ์โคโรนา (Corona) ก็เป็นปัญหาที่สำคัญเช่นกันซึ่งมักพบกับมอเตอร์ที่ใช้กับแรงดันที่สูงกว่า 5 kV

ค. ผลเนื่องมาจากแรงทางกล แรงทางกลเกิดจากกระแสที่ไหลในขดลวดโดยจะมีขนาดเป็นสัดส่วนกับกระแสกำลังสอง (FCI^2) แรงที่เกิดขึ้นจะเป็นสาเหตุทำให้ขดลวดเกิดการสั่นสะเทือน และเป็นสาเหตุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

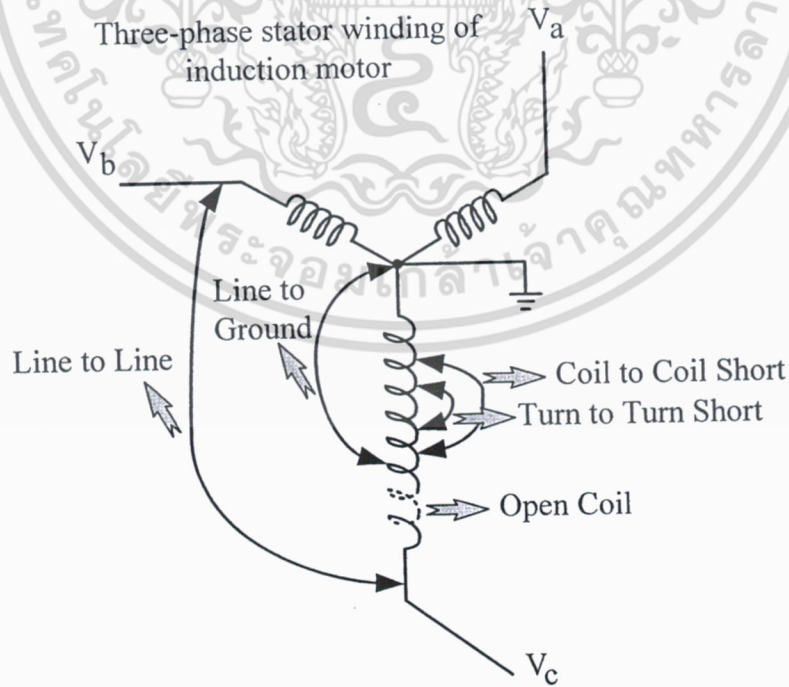
ทำให้เกิดความเสียหายที่ฉนวนตัวนำ การขัดถูกันระหว่างโรเตอร์กับสเตเตอร์ก็เป็นสาเหตุทำให้เกิดความร้อนขึ้นที่จุดสัมผัสนั้น และทำให้อายุการใช้งานของขดลวดเสื่อมเร็วกว่ากำหนดด้วย

ง. ผลเนื่องมาจากสิ่งแวดล้อม สิ่งสำคัญที่ผู้ใช้มอเตอร์ควรคำนึงถึงก็คือควรให้มอเตอร์ทำงานในอากาศที่แห้งและสะอาด และควรมีช่วงเวลาในการดูแลรักษามอเตอร์ด้วย ปัญหาหนึ่งที่มีการกล่าวถึงก็คือ การจับของไอน้ำบนขดลวดสเตเตอร์ ซึ่งเป็นสาเหตุทำให้เกิดการลัดวงจรลงกราวด์ในสล็อต

โหมคความล้มเหลวของตัวนำสเตเตอร์จะแบ่งออกตามชนิดของการนำไปสู่การลัดวงจร โดยสามารถแบ่งออกได้เป็น 5 ประเภทแสดงในรูปที่ 2.2 ได้แก่

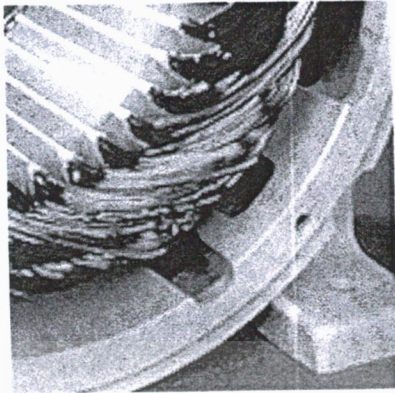
- ก. การลัดวงจรระหว่างรอบถึงรอบ (Turn to turn)
- ข. การลัดวงจรระหว่างขดลวดถึงขดลวด (Coil to coil)
- ค. การลัดวงจรระหว่างเฟสถึงเฟส (Line to Line)
- ง. การลัดวงจรระหว่างขดลวดถึงกราวด์ (Coil to ground)
- จ. และกรณีวงจรเปิด (Open coil)

ทั้งนี้ในรูปที่ 2.3 แสดงตัวอย่างรูปจริงที่เกิดจากความล้มเหลวของขดลวดสเตเตอร์ในลักษณะต่างกัน

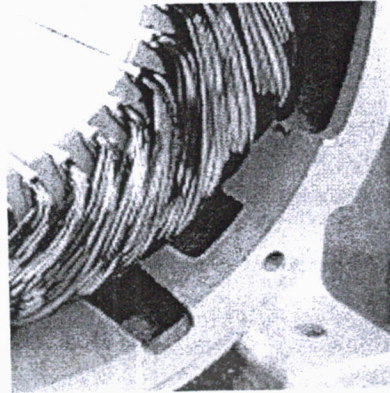


รูปที่ 2.2 โหมคความล้มเหลวของขดลวดสเตเตอร์ในเครื่องจักรกลไฟฟ้าเมื่อขดลวดต่อแบบวาย [6]

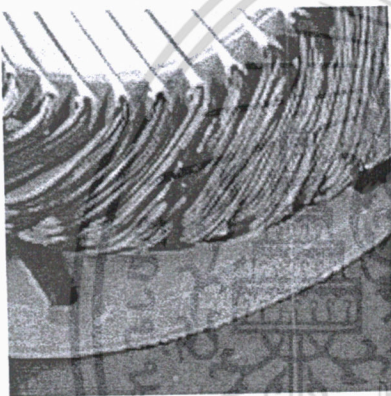
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



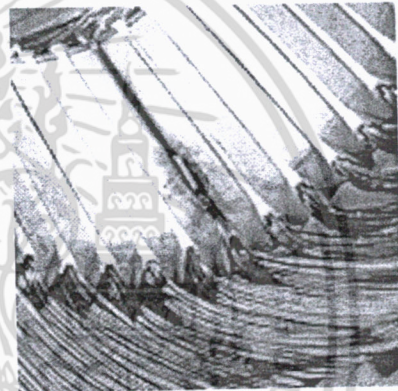
ก. รอบถึงรอบลัดวงจร



ข. เฟสถึงเฟสลัดวงจร



ค. ลัดวงจรลงกราวด์



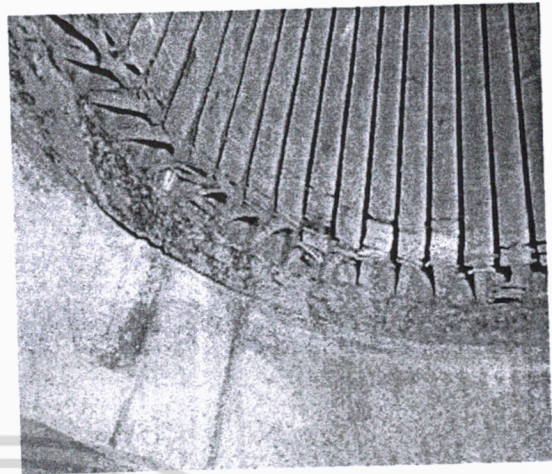
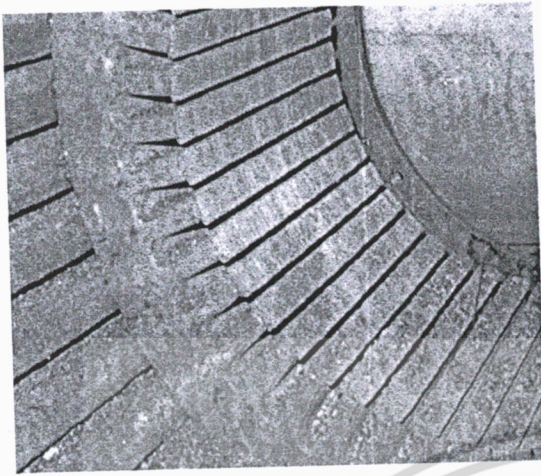
ง. ลัดวงจรลงกราวด์ในสล็อต

รูปที่ 2.3 รูปจริงที่เกิดจากความล้มเหลวของขดลวดสเตเตอร์ในลักษณะต่าง ๆ

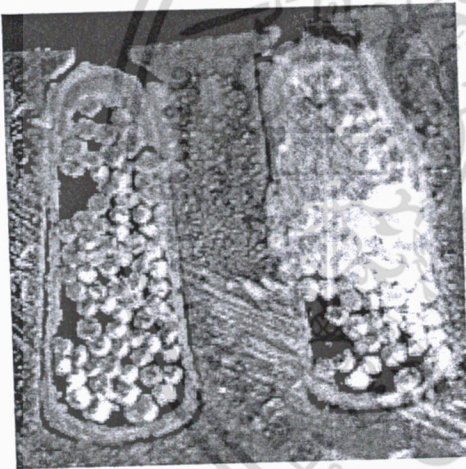
(www.maintenanceworld.com)

ถึงแม้ว่าแกนสเตเตอร์จะเป็นส่วนที่เกี่ยวข้องโดยตรงในการซ่อมแซม และยิ่งเฉพาะกระบวนการเผาที่แกนสเตเตอร์ ทั้งนี้เป็นผลดีจากการตั้งอุณหภูมิของเตาอบให้เพิ่มขึ้นทีละน้อย ๆ จากคำแนะนำของสมาคม EASA บอกว่าให้ตั้งอุณหภูมิของตู้อบที่ 343°C สำหรับแผ่นออแกนิกและ 400°C สำหรับที่เป็นออแกนิกแต่อุณหภูมิที่ตั้งไว้นี้ได้พิสูจน์ว่าสูงพอที่จะเผาขดลวดได้ ดังนั้นการดึงขดลวดออกจะต้องคอยให้คำแนะนำและชั้นของสเตเตอร์ดูเหมือนจะชำรุดในระหว่างการดึงขดลวดออก สิ่งที่ต้องระมัดระวังเป็นพิเศษเมื่อใช้อุณหภูมิที่ต่ำในมอเตอร์ 2-3 ตัวแรกเพื่อหลีกเลี่ยงการชำรุด ผลที่ตามคืออุณหภูมิการไหม้เพิ่มขึ้นเป็น 360°C-370°C ดังนั้นขดลวดจึงสามารถถอดออกได้ง่ายและแกนจะสะอาด ทำให้ลดโอกาสเกิดความเสียหายที่แกนเหล็กได้ รูปที่ 2.4 แสดงรูปจริงของแกนสเตเตอร์ที่ดีและไม่ดี และรูปที่ 2.5 แสดงภาพขยายของการลงขดลวดที่ดีและไม่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การเปรียบเทียบแกนสเตเตอร์ (ซ้าย) ดีและ (ขวา) ตัวนำสเตเตอร์ใหม่ [7]



รูปที่ 2.5 ลักษณะของการลงขดลวดสเตเตอร์ (ซ้าย) ไม่ดีและ (ขวา) ดี [3]

2.3 ผลกระทบต่อประสิทธิภาพของเครื่องจักรกลไฟฟ้า

ในโรงงานอุตสาหกรรมที่มีการใช้งานของเครื่องจักรกลไฟฟ้าเป็นจำนวนมากนั้น โดยประมาณ 70% ของเครื่องจักรกลไฟฟ้าเหล่านั้นจะถูกขับเคลื่อนด้วยอุปกรณ์โซลิสเตด และส่วนใหญ่ของเครื่องจักรกลไฟฟ้าเหล่านั้นจะเป็นมอเตอร์เหนี่ยวนำ ทั้งนี้การถูกนำมาใช้งานของมอเตอร์เหนี่ยวนำจะมีการจัดซื้อใหม่ และการพันขดลวดใหม่ (จัดซื้อมือสอง) เข้ามาติดตั้ง นอกจากนี้แล้วมอเตอร์เหนี่ยวนำที่มีการใช้งานอยู่หากเกิดความเสียหายอันเนื่องมาจากปัญหาของขดลวดสเตเตอร์ ก็จะถูกนำมาพันใหม่แทนการจัดซื้อมอเตอร์ตัวใหม่เพื่อลดค่าใช้จ่ายที่ถูกลงกว่าการซื้อใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเหตุการณ์ดังกล่าวผลที่ตามมาของการพันขดลวดใหม่ ซึ่งรวมถึงการพันใหม่ซ้ำ ๆ กันอีกด้วย จากรายงานการวิจัยในเรื่องดังกล่าวพบว่ามอเตอร์ที่มีการพันใหม่เมื่อใช้งานไปนาน ๆ ผลที่ตามมาหลัก ๆ จะมีอยู่ด้วยกัน 2 กรณี ได้แก่ การลดลงของประสิทธิภาพ และการเสื่อมลงของสมรรถนะ ซึ่งจากการศึกษาเชิงวิจัยในเรื่องการพันใหม่ของขดลวดจะสามารถทำให้ประสิทธิภาพลดลงได้ 1-2 เปอร์เซ็นต์ [7-8]

ตารางที่ 2.1 เปรียบเทียบเปอร์เซ็นต์การสูญเสียของมอเตอร์ 2 และ 4 ขั้วแม่เหล็ก [8]

เปอร์เซ็นต์การกระจายค่าการสูญเสียมอเตอร์ โดยเฉลี่ยสำหรับมอเตอร์ที่ใช้ความถี่ 60 Hz			
ค่าการสูญเสีย	2 ขั้ว	4 ขั้ว	แฟลคเตอร์ที่มีผลต่อค่าการสูญเสีย
ค่าการสูญเสียแกนเหล็ก (Core losses)	19%	21%	แผ่นเหล็ก ช่องอากาศ การอิมตัว
ค่าการสูญเสียจากแรงลมและแรงเสียดทาน	25%	10%	ประสิทธิภาพของใบพัด การหล่อลื่น ร่องลื่น
ค่าการสูญเสียสเตเตอร์	26%	34%	พื้นที่หน้าตัดตัวนำ ความยาวของรอบขดลวด การกระจายของความร้อน
ค่าการสูญเสียโรเตอร์	19%	21%	ประเภทของวัสดุและพื้นที่หน้าตัดของแท่งตัวนำและวงแหวนลัดวงจร
ค่าการสูญเสียโหลดสเตรย์	11%	14%	กระบวนการผลิต รูปแบบของสลอตช่องอากาศ

โดยปกติแล้วค่าการสูญเสียกำลังโดยรวมในเครื่องจักรกลไฟฟ้า ได้ถูกแบ่งออกเป็นตามส่วนองค์ประกอบของการสูญเสียตัวนำสเตเตอร์ การสูญเสียตัวนำโรเตอร์ การสูญเสียแกนเหล็ก การสูญเสียจากแรงลม การสูญเสียจากแรงเสียดทาน และการสูญเสียโหลดสเตรย์ ตามที่ได้กล่าวมาแล้ว ทั้งนี้สัดส่วนของค่าการสูญเสียเหล่านี้ จะมีการเปลี่ยนแปลงตามพิกัดและจำนวนขั้วแม่เหล็กของเครื่องจักรกลไฟฟ้า โดยที่เครื่องจักรกลขนาดเล็กจะมีการสูญเสียจากตัวนำสเตเตอร์มากกว่าเครื่องจักรกลขนาดใหญ่ และส่วนใหญ่แล้ว การศึกษาเกี่ยวกับการพันขดลวดใหม่ก็จะอยู่บนพื้นฐานของขนาดเครื่องจักรกลขนาดเล็กด้วย ตารางที่ 2.1 เป็นตัวอย่างของการแบ่งเปอร์เซ็นต์การสูญเสียที่เกิดขึ้นในเครื่องจักรกลไฟฟ้าที่มีพิกัดเดียวกัน แต่มีจำนวนขั้วแม่เหล็กเป็น 2 และ 4 ขั้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 เปรียบเทียบค่าการสูญเสียในมอเตอร์เหนี่ยวนำกับการพันใหม่

	ค่าการสูญเสีย					η	โครงสร้าง ขดลวด
	สเตเตอร์ I ² R (W)	โรเตอร์ I ² R (W)	แกน เหล็ก	แรงลม+ เสียดทาน	โหลด สเตรย์		
มอเตอร์ A ใหม่	411.2	212.9	131.5	22.5	72.8	86.7	—
พันใหม่	365.6	177.9	153.5	69.2	53.7	86.9	—
มอเตอร์ B ใหม่	578.1	229.1	196.6	40.6	56.3	83.2	—
พันใหม่	557.8	194.5	214	42.7	40.3	83.4	—
มอเตอร์ C ใหม่	852	752.4	705.4	1161.4	440.6	95	Lap
พันใหม่ครั้งที่ 1	930.7	774.7	752	1137.4	719	94.5	Lap
พันใหม่ครั้งที่ 2	895.1	745	686.2	1159.9	562.2	94.9	Lap
พันใหม่ครั้งที่ 3	894.9	744.9	693	1140.7	596.2	94.8	Lap
มอเตอร์ D ใหม่	1805.3	1204.2	1093.7	319.7	1280.7	93	Concentric
พันใหม่ครั้งที่ 1	1546	1102.9	1078.3	272.4	1117.3	93.6	Lap
พันใหม่ครั้งที่ 2	1523.1	1098	1078.7	309.3	1138.6	93.6	Lap
พันใหม่ครั้งที่ 3	1489.3	1059.7	1131.9	297.6	1094.6	93.7	Lap
มอเตอร์ E ใหม่	1326.8	759.7	1123	1394.8	123.4	95.9	Lap
พันใหม่ครั้งที่ 1	1280.2	852.8	1108.8	1296.7	282.1	95.9	Lap
พันใหม่ครั้งที่ 2	1243.1	830.9	1050	1307.2	380.1	95.9	Lap
พันใหม่ครั้งที่ 3	1295.6	817.2	1093.6	1427.8	216.4	95.8	Lap
มอเตอร์ F ใหม่	1922.6	1129.1	1459.6	448.1	1490	95.8	Concentric
พันใหม่ครั้งที่ 1	1817.9	1216.7	1617.2	633.3	2337.2	95.1	Concentric
พันใหม่ครั้งที่ 2	1772.1	1121	1618.8	671.4	1621.3	95.6	Lap

ตารางที่ 2.2 เป็นตัวอย่างของการเปรียบเทียบค่าการสูญเสียในมอเตอร์เหนี่ยวนำตัวใหม่กับการพันใหม่ โดยทำการศึกษาจากกลุ่มตัวอย่างคือมอเตอร์ 6 ตัว (A-F) ทั้งนี้สาเหตุของค่าการสูญเสียที่เปลี่ยนแปลงไปอาจจะเป็นผลมาจาก โครงสเตเตอร์ การพันขดลวด รวมถึงการเปลี่ยนแปลงรูปแบบของใบพัดด้วย

สำหรับจุดประสงค์ในการทดสอบของวิธีการพันขดลวด มอเตอร์จะถูกแบ่งออกเป็น 2 กลุ่มคือ กลุ่มที่ควบคุมไม่ได้กับกลุ่มที่ควบคุมได้ ผลการเปรียบเทียบประสิทธิภาพของมอเตอร์ทั้ง 6 ตัวอย่างแรก เปรียบเทียบกันระหว่างมอเตอร์ที่พันใหม่กับมอเตอร์ที่พันเดิม เหมือนญาติเห็นไปใจประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกลุ่มที่ 1 ซึ่งมีค่าเฉลี่ยของประสิทธิภาพที่ลดลง 0.6% และมากที่สุด 1% โดยสาเหตุส่วนใหญ่มาจากการเพิ่มขึ้นของการสูญเสียในการเหนี่ยวนำที่สเตเตอร์ ซึ่งเกิดจากการเปลี่ยนรูปแบบการพันขดลวดพื้นที่หน้าตัดของลวดและค่า MLT จากการซ่อมแซมในโรงซ่อม การพันขดลวดที่เหลือทั้งหมดถูกควบคุมโดยคำแนะนำพิเศษ และผลที่ตามมาคือประสิทธิภาพลดลงน้อยมากเฉลี่ยประมาณ 0.1% ส่วนในกลุ่มของมอเตอร์ที่ควบคุมได้พบว่ามอเตอร์ 3 ตัวไม่มีการเปลี่ยนแปลงและการลดลงของประสิทธิภาพของมอเตอร์ทั้ง 8 ตัวพบว่ามากที่สุด 0.7% ซึ่งจากการตรวจสอบพบว่าเกินจากความเสียหายที่แกนสเตเตอร์โดยสังเกตเห็นได้ง่ายจากการทดสอบก่อนพันขดลวด

อย่างไรก็ตามมอเตอร์ทั้ง 6 ตัวที่มีประสิทธิภาพเพิ่มขึ้นนี้โดยทั้งหมดเป็นมอเตอร์ที่ใช้กับความถี่ไฟที่ 50Hz จากการวิเคราะห์สาเหตุพบว่ามอเตอร์ทั้ง 5 ตัวที่มีประสิทธิภาพเพิ่มขึ้นมาจากการลดความสูญเสียที่การเหนี่ยวนำที่สเตเตอร์ (เกิดจากการเปลี่ยนรูปแบบการพันขดลวดและการลดอุณหภูมิที่ใช้งาน) ส่วนสาเหตุอื่นมาจากการลดลงของแรงเสียดทาน (การเปลี่ยนจาระบีที่รอกเลื่อน) นอกเหนือจากระดับประสิทธิภาพ ประสิทธิภาพบางส่วนสามารถเกิดจากสาเหตุของการพันขดลวดหรือไม่ก็ได้ ซึ่งจะกล่าวในหัวข้อต่อไป

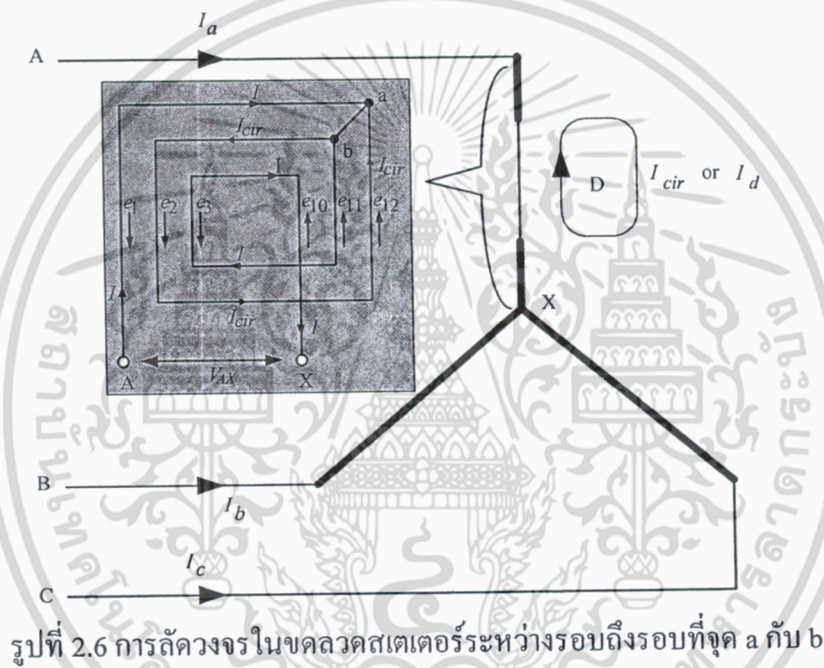
ข้อมูลในตารางที่ 2.2 นี้เป็นข้อมูลจากการเปรียบเทียบค่าการสูญเสียในมอเตอร์เหนี่ยวนำกับการพันใหม่ กล่าวคือมอเตอร์ทั้ง 6 ตัว (A-F) มีการไหม้ของขดลวดซ้ำหลายครั้งและนำมาพันขดลวดใหม่ มอเตอร์ A และ B ขดลวดใหม่ 5 ครั้งในเตาอบแต่พันใหม่แค่ครั้งเดียว สำหรับมอเตอร์ตัว A ผลจากการทดลองพบว่าการลดลงของการสูญเสียความเหนี่ยวนำที่สเตเตอร์ ความเหนี่ยวนำโรเตอร์และการสูญเสียเนื่องจากการใช้งานเพียงเล็กน้อยแต่การสูญเสียที่แกนเหล็กเพิ่มขึ้น ส่วนใหญ่แล้วการสูญเสียที่เพิ่มขึ้นมีสาเหตุมาจากการประกอบรอกเลื่อนที่ไม่เหมาะสม และการหล่อลื่นไม่เพียงพอ สำหรับมอเตอร์ B การเพิ่มขึ้นของการสูญเสียที่แกนเหล็กเพียงเล็กน้อย มอเตอร์ C D และ E เป็นมอเตอร์ที่ผ่านการไหม้และได้ถูกพันใหม่มาแล้ว 3 ครั้ง แต่ละตัวถูกประกอบและทดลองซ้ำหลังจากการพันใหม่ทุกครั้ง มอเตอร์ F ถูกพันขดลวดใหม่ 2 ครั้ง ได้ผลการทดลองดังตาราง

จากที่กล่าวมาสามารถบอกได้ว่าการพันขดลวดทำให้ประสิทธิภาพลดลงน้อยกว่า 0.2% กกับการเปลี่ยนแปลงรูปร่างของขดลวดพบว่าการเปลี่ยนแปลงของประสิทธิภาพอยู่ระหว่าง -0.5% ถึง +0.7% และในมอเตอร์ทั้ง 3 ตัว (A, B และ D) พบว่าประสิทธิภาพดีขึ้น ในทางปฏิบัติเป็นไปได้ว่าการที่จะเพิ่มประสิทธิภาพของมอเตอร์ทำได้โดยการเปลี่ยนแปลงรูปร่างของขดลวด และเพิ่มขนาดของลวดทองแดง ยกตัวอย่างเช่นการเปลี่ยนรูปแบบของการพันขดลวดจากแบบซ้อนกัน (Concentric) ไปเป็นแบบแถบหรือจากชั้นเดียวไปเป็นสองชั้น โดยทั่วไปแล้วการพันขดลวดซ้ำถูกเชื่อว่าจะทำให้เกิดการลดลงแบบสะสมของประสิทธิภาพมอเตอร์ แต่จากการศึกษานี้สามารถสรุปได้ว่าการไหม้และการพันขดลวดซ้ำไม่ทำให้ประสิทธิภาพของมอเตอร์ลดลงได้แบบชัดเจน (น้อยกว่า 0.7%)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การวิเคราะห์ขดลวดสเตเตอร์ลัดรอบ

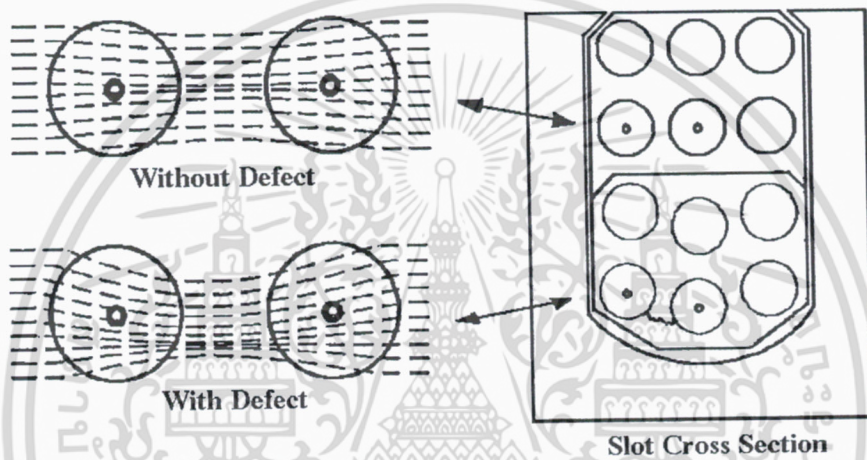
การเกิดสภาวะลัดวงจรระหว่างรอบถึงรอบของชุดขดลวดแสดงในรูปที่ 2.6 ซึ่งแสดงการลัดวงจรระหว่างจุด a กับ b ซึ่งจะทำให้เกิดกระแสไหลวน (Circulate current, I_{cir}) ถ้าพิจารณาที่จุด A และ X สามารถแยกพิจารณาได้เป็น 2 วงจรที่อิสระต่อกันนั่นคือวงจรที่เกิดจากกระแส I และวงจรที่เกิดจากกระแส I_{cir} เมื่อกระแส I คือกระแสเฟสทั้งนี้กระแส I_{cir} จะสร้างแรงเคลื่อนแม่เหล็ก (MMFs) ในทิศทางตรงข้ามกันกับแรงเคลื่อนแม่เหล็กหลักที่สร้างโดยขดลวดเฟส ส่งผลให้แรงเคลื่อนแม่เหล็กหลักลดลง สำหรับการลัดวงจรหลาย ๆ รอบสามารถวิเคราะห์โดยใช้วิธีการเดียวกันนี้ได้ [9]



รูปที่ 2.6 การลัดวงจรในขดลวดสเตเตอร์ระหว่างรอบถึงรอบที่จุด a กับ b

ผลที่เกิดขึ้นจากการลัดวงจรระหว่างรอบถึงรอบ จะส่งผลให้ MMFs มีค่าเปลี่ยนแปลงไปจากเดิม และทำให้ความเหนี่ยวนำร่วม (Mutual inductance) ระหว่างเฟสกับขดลวดสเตเตอร์วงจรอื่นในมอเตอร์มีค่าเปลี่ยนแปลงไปด้วย แสดงในรูปที่ 2.7 ซึ่งโดยทั่วไปแล้วความเหนี่ยวนำที่เกิดขึ้นภายในและผลของความเหนี่ยวนำร่วมระหว่างรอบขดลวดซึ่งจะมีค่าที่เท่ากันในแต่ละชุดขดลวดและเป็นสัดส่วนกันในแต่ละชั้นลวดตัวนำ ซึ่งก็จะจะเป็นผลให้ค่าความเหนี่ยวนำและค่าความต้านทานไฟฟ้ากระแสสลับ (Impedance) มีความสมมูลกัน แต่ถ้าเกิดสภาวะรอบขดลวดลัดวงจรเกิดขึ้นจะทำให้ค่าความเหนี่ยวนำร่วมและของมันเองมีการเปลี่ยนแปลงไปเนื่องจากการเปลี่ยนแปลงของค่าอิมพีแดนซ์ระหว่างเฟสที่ไม่สมมูล และด้วยเหตุนี้เองที่ทำให้เกิดการเปลี่ยนแปลงความเข้มของฟลักซ์แม่เหล็กที่ตำแหน่งของการลัดรอบ [10] นอกจากนี้แล้วสัญญาณฟอลต์ที่เกิดขึ้นจะมีเฟสใหม่โดยในที่นี้จะเรียกเป็นเฟส f_s และเมื่อขดลวดสเตเตอร์ต่อเป็นแบบวาย (Star) และไม่ได้ต่อสายนิวตรอลระหว่างแหล่งจ่ายกับจุดนิวตรอลของเอกสาร์นี้เป็นเอกสาร์ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจักรกลไฟฟ้า สมการแรงดันสำหรับขดลวดสเตเตอร์ใน 1 เฟสแสดงในสมการที่ (2.1)-(2.2) ทั้งนี้ หากเครื่องจักรกลไฟฟ้าไม่สมมาตร (Asymmetric) เมตริกอินดักแตนซ์ (Inductance) ซึ่งแสดงในสมการที่ (2.3) สามารถแยกพิจารณาได้เป็น 2 องค์ประกอบ [11] เทอมแรกแสดงในสมการที่ (2.4) คือส่วนสมมูลของเมตริกอินดักแตนซ์สเตเตอร์ ซึ่งเป็นส่วนที่พิจารณาจากเครื่องจักรกลไฟฟ้าที่มีความสมมูลและในเทอมที่ 2 แสดงในสมการที่ (2.5) ซึ่งเป็นส่วนความไม่สมมาตรของเมตริกอินดักแตนซ์สเตเตอร์ที่มีสาเหตุมาจากการลัดวงจรของขดลวดสเตเตอร์



รูปที่ 2.7 การเปลี่ยนแปลงของฟลักซ์กรณีเกิดการลัดวงจรระหว่างรอบถึงรอบ

$$|v_{xx}| = |R_x| |I_x| + \frac{d|\Psi_x|}{dt} + v_{on} \quad (2.1)$$

$$|\Psi_x| = |L_{ABC}| |I_x| \quad (2.2)$$

$$L_{ABC} = L_{constant} + L_{varying} \quad (2.3)$$

$$L_{constant} = \begin{vmatrix} L_{ls} + L_{ms} & M & M \\ M & L_{ls} + L_{ms} & M \\ M & M & L_{ls} + L_{ms} \end{vmatrix} \quad (2.4)$$

$$L_{varying} = \begin{vmatrix} \Delta L_1 \cos(\varphi_0) & M & M \\ M & L_{\sigma s} + L_{ms} & M \\ M & M & L_{\sigma s} + L_{ms} \end{vmatrix} \quad (2.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ x คือเฟส a, b และ c, X คือที่จุดต่อสายของมอเตอร์, L_{ls} คืออินดักแตนซ์รั่วไหล, L_{ms} คืออินดักแตนซ์เหนี่ยวนำ, และ M คือ $L_{ms}/2$ ทั้งนี้ฟลักซ์สืบเนื่อง Ψ_x ใน (2.2) จะสัมพันธ์กับกระแสในแต่ละเฟส รวมถึงเฟสใหม่ที่เกิดขึ้นด้วย สมการแรงดันเหนี่ยวนำสเตเตอร์ลัดรอบ (e_{st}) ที่เกิดขึ้นในขดลวดสเตเตอร์สามารถหาได้จาก

$$e_{st} = -\frac{d\Psi_{st}}{dt} \quad (2.6)$$

จาก (2.6) เป็นสมการแรงดันของฟอลต์สเตเตอร์ ซึ่งถ้านำมาหารด้วย R_{st} จะได้เป็น I_{st} หรือ I_{cir} และถ้าจำนวนของการลัดรอบมากขึ้นจะทำให้ e_{st} มีค่าเพิ่มขึ้น ดังนั้นสัญญาณฟอลต์สเตเตอร์ก็จะมีค่ามากขึ้นด้วย ทางด้านพฤติกรรมขององค์ประกอบความถี่ในรูปคลื่นฟลักซ์ช่องอากาศในฟังก์ชันของการลัดรอบจะขึ้นอยู่กับค่าความถี่มูลฐาน (f_1) จำนวนขั้วแม่เหล็ก (p) และสลิป (s) ดังสมการ

$$f_{st} = f_1 \left\{ \frac{m}{p}(1-s) \pm k \right\} \quad (2.7)$$

เมื่อ f_{st} คือความถี่ของฟอลต์สเตเตอร์, m คือ 1, 2, 3, ... และ k คือ 1, 3, 5, ... [1] สเปกตรัม ฟอลต์สเตเตอร์นี้จะเกิดขึ้นรวมอยู่ในหลายสัญญาณที่นำมาวิเคราะห์ แต่อาจจะมีขนาดที่แตกต่างกันได้ตามพารามิเตอร์ของเครื่องจักรกลไฟฟ้า

บทที่ 3

องค์ประกอบแรงดันลำดับศูนย์

องค์ประกอบแรงดันลำดับศูนย์ถูกนำมาใช้สำหรับการตรวจจับฟอลต์ที่เกิดขึ้นในมอเตอร์เหนี่ยวนำ ซึ่งที่ผ่านมาได้แก่ การตรวจจับฟอลต์สเตเตอร์และการตรวจจับฟอลต์โรเตอร์เป็นต้น ทั้งนี้ในบางครั้งชื่อ องค์ประกอบแรงดันลำดับศูนย์ได้ถูกเรียกเป็นสัญญาณผลรวมแรงดัน (3 เฟส) ซึ่งเรียกตามวิธีการ คำนวณหาองค์ประกอบแรงดันลำดับศูนย์นั่นเอง

3.1 ทฤษฎีพื้นฐาน

ความไม่สมดุลของค่าอิมพีแดนซ์ (Impedance, Z) ที่เกิดขึ้นในขดลวดสเตเตอร์ของมอเตอร์เหนี่ยวนำ 3 เฟส มีสาเหตุมาจากการลัดวงจรหรือเรียกเป็นการลัดรอบภายในเฟสใดเฟสหนึ่ง ซึ่งโดยความเป็นจริงแล้วค่าอิมพีแดนซ์ในแต่ละเฟสของเครื่องจักรกลไฟฟ้าหลายเฟสจะมีความสมดุลกันแต่ก็อาจจะเกิดกรณีที่ไม่สมดุลได้ ทั้งนี้มีสาเหตุมาจากอุปกรณ์ภายนอกที่นำมาต่อเพิ่มเข้ากับเครื่องจักรกลไฟฟ้า เช่น เครื่องตรวจสถานะสัญญาณ เครื่องวัดต่าง ๆ เป็นต้น แต่สาเหตุที่สำคัญมากจะมาจากอิมพีแดนซ์ของเฟส ซึ่งเมื่อพิจารณาเฉพาะองค์ประกอบมูลฐาน (Fundamental) แล้ว องค์ประกอบแรงดันลำดับศูนย์ (V_0) ในรูปของเฟสเซอร์ (Phasor) จะมีความสัมพันธ์เป็น [12]

$$\tilde{V}_0 = \tilde{V}_a + \tilde{V}_b + \tilde{V}_c \tag{3.1}$$

หรือ

$$\tilde{V}_0 = Z_a \tilde{I}_a + Z_b \tilde{I}_b + Z_c \tilde{I}_c \tag{3.2}$$

สำหรับมอเตอร์ที่มีความสมดุล ค่าอิมพีแดนซ์ในแต่ละเฟสจะเท่ากัน

$$Z_a = Z_b = Z_c \tag{3.3}$$

และจากสมการที่ (3.1) – (3.3) แสดงถึงสภาวะที่มีความสมดุล

$$\tilde{V}_a + \tilde{V}_b + \tilde{V}_c = 0 \tag{3.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจากสมการที่ (3.1) ถ้าอิมพีแดนซ์ของมอเตอร์มีความเป็นเชิงเส้นก็จะหมายความว่ารวมไปถึงแรงดันไฟฟ้าที่เวลาใด ๆ ด้วย

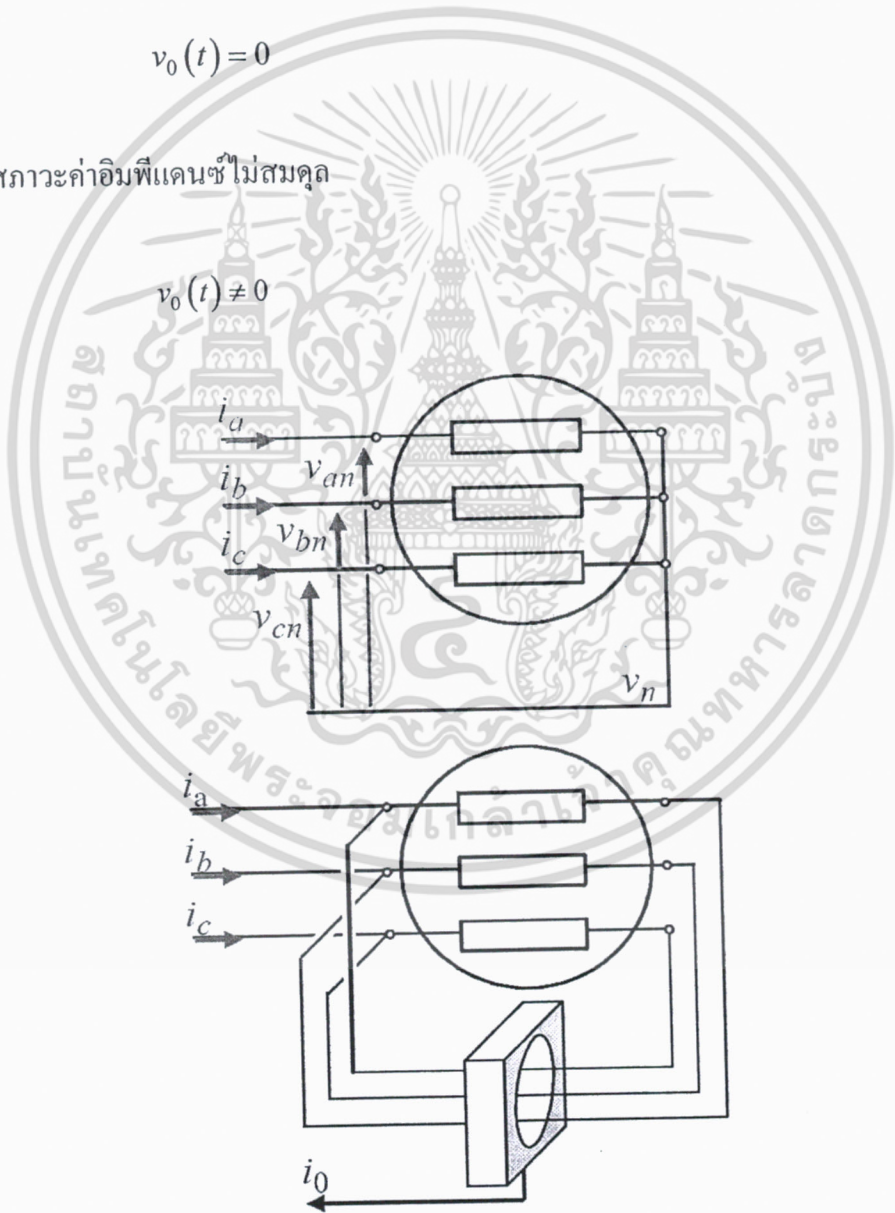
$$v_a(t) + v_b(t) + v_c(t) = v_0(t) \tag{3.5}$$

จากสมการที่ (3.5) สามารถพิจารณาได้เป็น 2 กรณี ได้แก่ ที่สภาวะค่าอิมพีแดนซ์สมดุล

$$v_0(t) = 0 \tag{3.6}$$

และที่สภาวะค่าอิมพีแดนซ์ไม่สมดุล

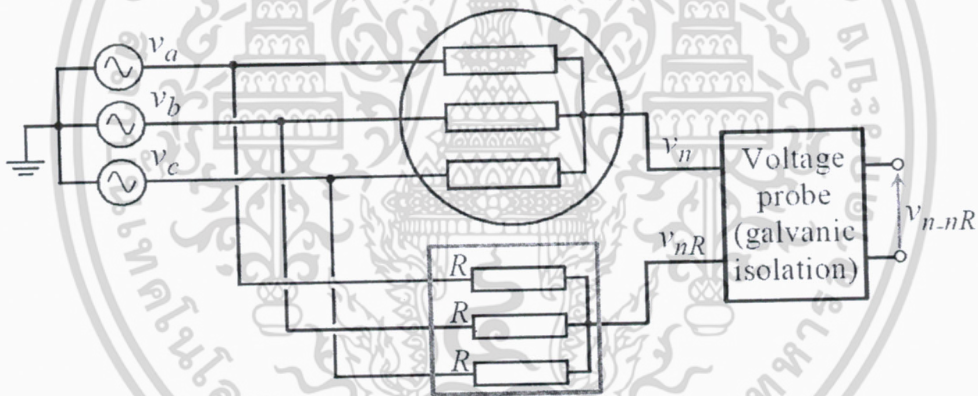
$$v_0(t) \neq 0 \tag{3.7}$$



รูปที่ 3.1 วิธีการวัดสัญญาณองค์ประกอบลำดับศูนย์เมื่อ (บน) มอเตอร์ต่อแบบวายและเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกา (ล่าง) มอเตอร์ต่อแบบเดลต้า [13] ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ตำรวการประยุกต์ใช้องค์ประกอบแรงดันลำดับศูนย์สำหรับการตรวจจับฟอลต์สเตเตอร์

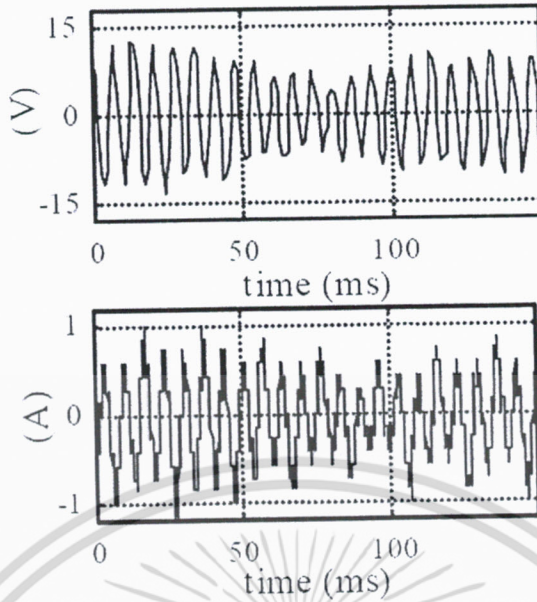
การต่อวงจรสำหรับตรวจจับสัญญาณองค์ประกอบแรงดันลำดับศูนย์ในการวินิจฉัยมอเตอร์นั้นสามารถตรวจสอบได้ทั้งองค์ประกอบแรงดันลำดับศูนย์ของสัญญาณกระแสและแรงดัน ซึ่งจะมีความแตกต่างเฉพาะในส่วนของวิธีการ กล่าวคือการวัดสัญญาณองค์ประกอบแรงดันลำดับศูนย์จะสามารถกระทำได้กับกรณิมอเตอร์ต่อแบบวาย (Wye-connected) และสำหรับกรณิการวัดสัญญาณองค์ประกอบกระแสลำดับศูนย์จะสามารถกระทำได้กับกรณิมอเตอร์ต่อแบบเดลต้า (Delta-connected) แสดงในรูปที่ 3.1 ซึ่งการตรวจวัดสัญญาณแรงดันนั้นอาจใช้โพรบวัดแรงดันซึ่งใช้หลักการฮอลล์ (Hall effect voltage sensor) และสำหรับการตรวจวัดสัญญาณกระแสสามารถใช้เซ็นเซอร์วัดกระแสซึ่งใช้หลักการฮอลล์ (Hall effect current sensor) ก็จะทำให้ลดความคลาดเคลื่อนของสัญญาณลงได้ นอกจากนี้แล้วในรูปที่ 3.2 ได้ใช้ความต้านทานที่มีค่าเท่ากันนำมาต่อเป็นเน็ตเวิร์ค ทั้งนี้เพื่อลดปัญหาขององค์ประกอบแรงดันลำดับศูนย์ที่เกิดจากแหล่งจ่าย



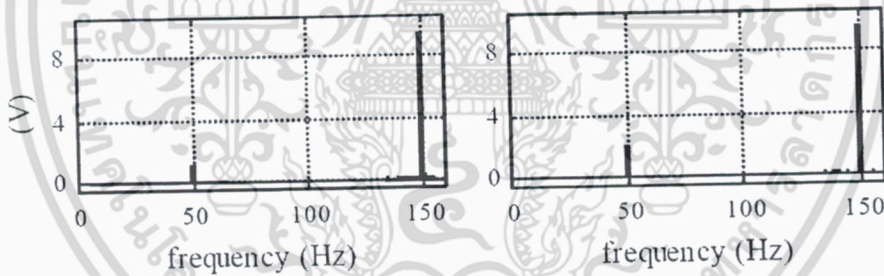
รูปที่ 3.2 วิธีการวัดสัญญาณองค์ประกอบลำดับศูนย์เมื่อ (บน) มอเตอร์ต่อแบบวายและ (ล่าง) มอเตอร์ต่อแบบเดลต้า

ในรูปที่ 3.3 แสดงตัวอย่างของทั้งสัญญาณองค์ประกอบแรงดันและกระแสลำดับศูนย์ในขณะที่มอเตอร์ทำงานที่สภาวะพิกัดโหลดและแรงดัน ในขณะที่รูปที่ 3.4 และรูปที่ 3.5 ได้แสดงสเปกตรัมของสัญญาณองค์ประกอบทั้งสัญญาณแรงดันและกระแสลำดับศูนย์ในสภาวะมอเตอร์สุขภาพดีและมอเตอร์มีความผิดปกติโดยได้ทำการทดลองเมื่อมอเตอร์ทำงานที่สภาวะพิกัด โหลดและแรงดันเช่นเดียวกัน [13]

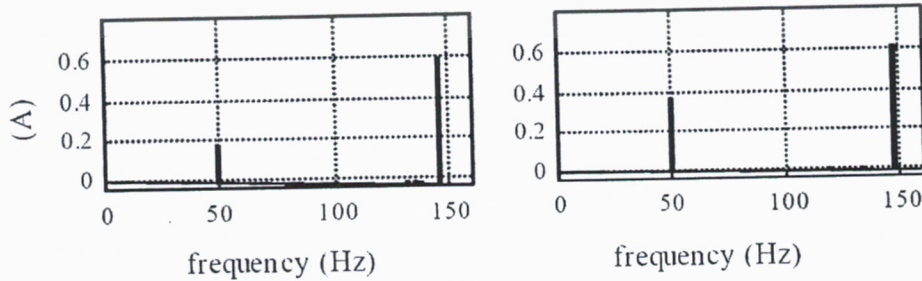
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 สัญญาณองค์ประกอบลำดับศูนย์เมื่อมอเตอร์ทำงานที่สภาวะพิกัดโหลดและแรงดัน
 (บน) สัญญาณองค์ประกอบแรงดันลำดับศูนย์เมื่อมอเตอร์ต่อแบบวายและ
 (ล่าง) สัญญาณองค์ประกอบกระแสลำดับศูนย์เมื่อมอเตอร์ต่อแบบเดลต้า



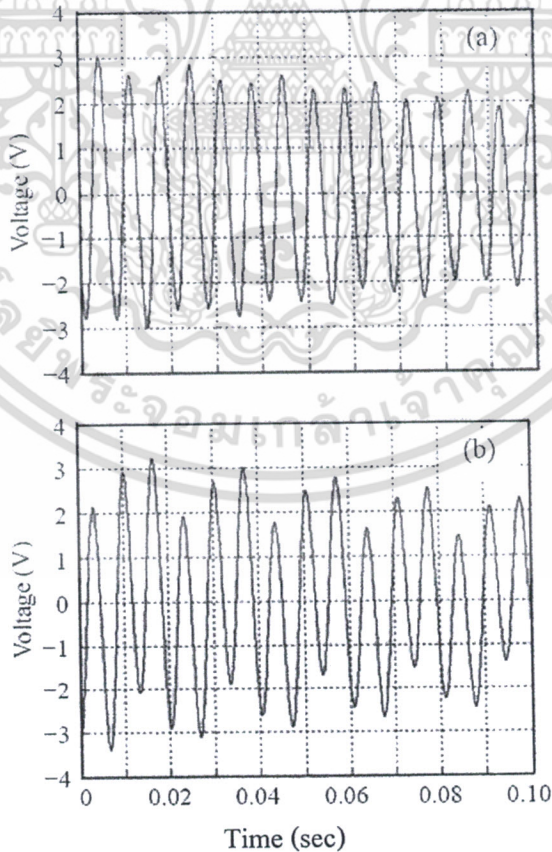
รูปที่ 3.4 สเปกตราของสัญญาณองค์ประกอบแรงดันลำดับศูนย์ (ซ้าย) ในสภาวะมอเตอร์สุขภาพดี และ
 (ขวา) มอเตอร์มีความผิดปกติ เมื่อมอเตอร์ทำงานที่สภาวะพิกัดโหลดและแรงดัน



รูปที่ 3.5 สเปกตราของสัญญาณองค์ประกอบกระแสลำดับศูนย์ (ซ้าย) ในสภาวะมอเตอร์สุขภาพดี และ
 (ขวา) มอเตอร์มีความผิดปกติ เมื่อมอเตอร์ทำงานที่สภาวะพิกัดโหลดและแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

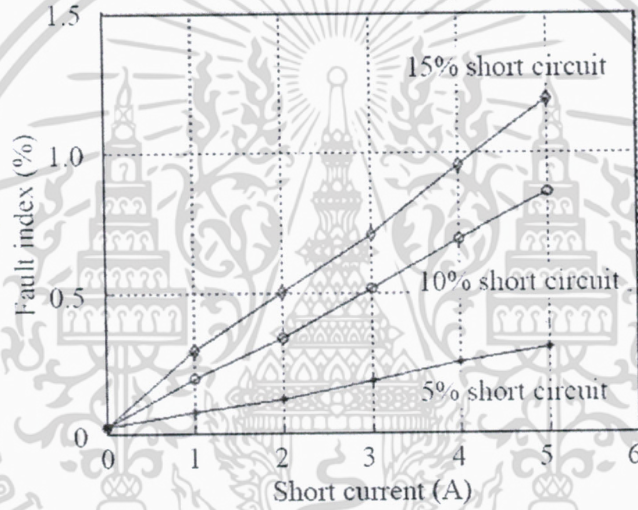
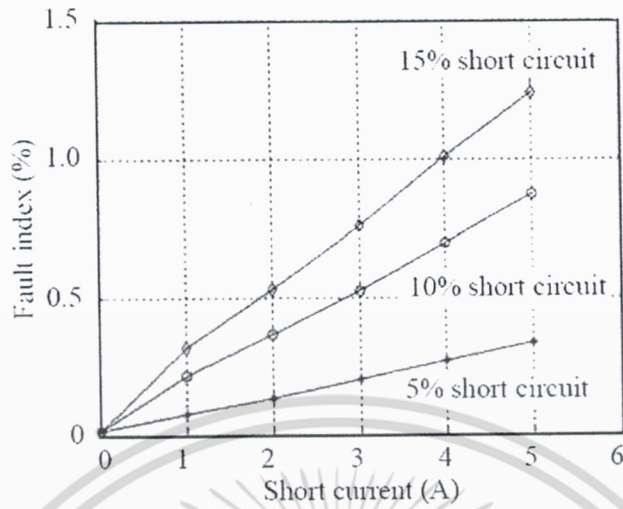
นอกจากนี้แล้วในงานวิจัยของ [6] ได้ทำการศึกษาและนำมาประยุกต์ใช้คือการตรวจจับสัญญาณองค์ประกอบแรงดันลำดับศูนย์ ซึ่งเป็นการคำนวณจากสัญญาณแรงดันหลังจากได้หยุดจ่ายไฟให้กับมอเตอร์แล้ว ซึ่งจะพบว่าสัญญาณแรงดันดังกล่าวจะมีลักษณะค่อย ๆ ลดลง ทั้งนี้ในรูปที่ 3.6 ได้แสดงรูปคลื่นของสัญญาณองค์ประกอบแรงดันลำดับศูนย์ของมอเตอร์ที่มีสุขภาพดี และในกรณีที่มอเตอร์มีความผิดปกติ (Faulty) ซึ่งเป็นการทดลองในขณะที่มอเตอร์ทำงานในสภาวะไร้อโหลด ด้วยสภาวะขดลวดสเตเตอร์ที่ 5% ของจำนวนรอบทั้งหมดในเฟส “a” โดยจากการทดลองนี้ได้แสดงให้เห็นอย่างชัดเจนถึงกรณีของมอเตอร์มีฟอลต์สเตเตอร์เกิดขึ้น พบว่าสัญญาณองค์ประกอบแรงดันลำดับศูนย์จะมีขนาดขององค์ประกอบมูลฐานที่มากกว่าในสภาวะมอเตอร์มีสุขภาพดี ในรูปที่ 3.7 ได้แสดงการประเมินคุณลักษณะด้านขนาดของสัญญาณองค์ประกอบแรงดันลำดับศูนย์ ซึ่งคำนวณจากสัญญาณแรงดันหลังจากหยุดจ่ายไฟภายใต้การทดสอบที่สภาวะไร้อโหลดและที่พิกัดโหลด ซึ่งพบว่าการประยุกต์ใช้สัญญาณแรงดันหลังจากหยุดจ่ายไฟสำหรับการตรวจจับสัญญาณฟอลต์สเตเตอร์ที่เกิดขึ้นในมอเตอร์นี้ภายใต้การทดลองในสภาวะมีและไม่มีโหลดนั้น พบว่าสภาวะของโหลดนั้นจะมีผลน้อยมากต่อระดับของสัญญาณองค์ประกอบแรงดันลำดับศูนย์ที่ได้



รูปที่ 3.6 สัญญาณองค์ประกอบแรงดันลำดับศูนย์หลังจากหยุดจ่ายไฟให้กับมอเตอร์

(บน) มอเตอร์สุขภาพดีและ (ล่าง) เมื่อมอเตอร์มีความผิดปกติ [6]

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในเพื่อใช้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 การประเมินคุณลักษณะด้านขนาดของสัญญาณองค์ประกอบแรงดันลำดับศูนย์ซึ่งคำนวณจากสัญญาณแรงดันหลังจากหยุดจ่ายไฟภายใต้การทดสอบ (บน) ที่สภาวะไว้โหลดและ (ล่าง) ที่พิกัดโหลด

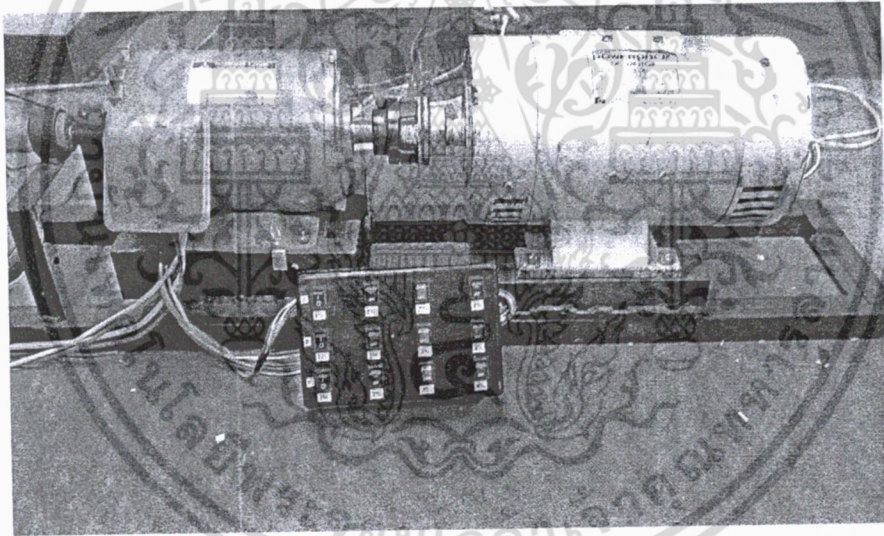
บทที่ 4

การออกแบบและการสร้างเครื่องมือวินิจฉัยฟอลต์

4.1 อุปกรณ์ที่ใช้ในการทดลอง

4.1.1 มอเตอร์เหนี่ยวนำ 3 เฟสที่ใช้ในการทดลอง

มอเตอร์ไฟฟ้าเหนี่ยวนำ Mitsubishi ขนาดพิกัด 1.5 kW Delta /Star 220/380V 4 P / 50 Hz / 1430 RPM / Winding Isolation Class B มอเตอร์เหนี่ยวนำตัวนี้จะถูกใช้เป็นตัวทดสอบการเกิดฟอลต์ที่สเตเตอร์ โดยจะติดตั้งอุปกรณ์ในการจำลองการเกิดฟอลต์แบบรอบต่อรอบ ซึ่งจะทำการลัดรอบของตัวมอเตอร์ในแต่ละเฟสออกมา เป็นจำนวน 1 ถึง 4 เปอร์เซ็นต์ของจำนวนรอบทั้งหมดของมอเตอร์ไฟฟ้าเหนี่ยวนำ เพื่อทำการวิเคราะห์ฟอลต์ที่เกิดขึ้นในระดับต่าง ๆ กันแสดงในรูปที่ 4.1

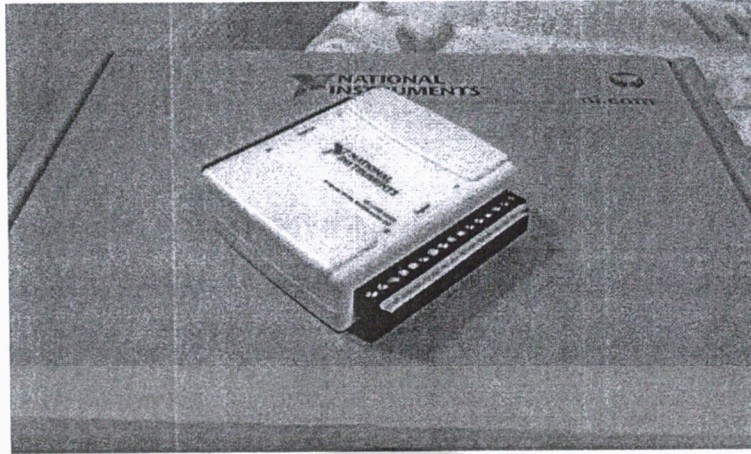


รูปที่ 4.1 ชุดทดลองมอเตอร์เหนี่ยวนำ 3 เฟสและมอเตอร์โหลด

4.1.2 ชุดการ์ดรับส่งข้อมูล NI USB-6009

ชุด DAQ Board ชุดนี้เป็นตัวประมวลผลสัญญาณจากอุปกรณ์ตรวจวัด และทำหน้าที่เป็นอุปกรณ์เชื่อมต่อข้อมูลระหว่างระบบตรวจวัด ชุดควบคุมมอเตอร์ไฟฟ้าและหน้าจอแสดงผล โดย DAQ Board ตัวนี้ประกอบด้วยพอร์ต I/O PORT 32-bit และพอร์ตรับสัญญาณอนาล็อก ขนาด 14 BIT จำนวน 8 ช่อง รับส่งข้อมูลระหว่างตัวบอร์ดกับเครื่องคอมพิวเตอร์ผ่าน Full Speed USB port type B (USB 1.1, 2.0)

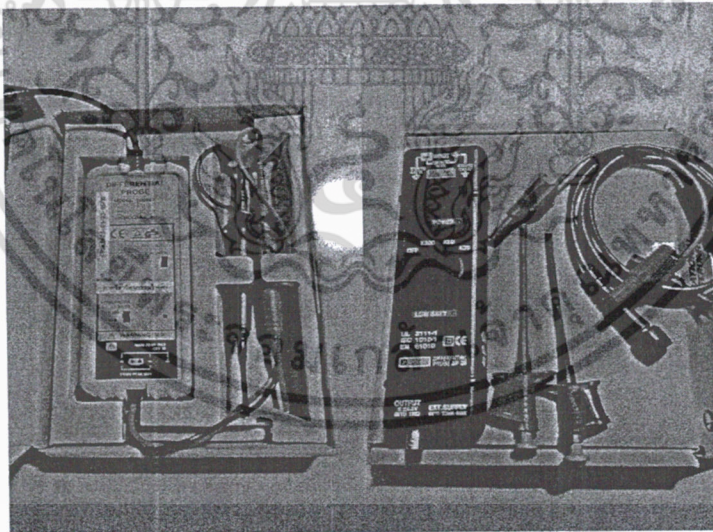
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 Data Acquisition Board, NI USB-6009

4.1.3 โพรบแรงดัน

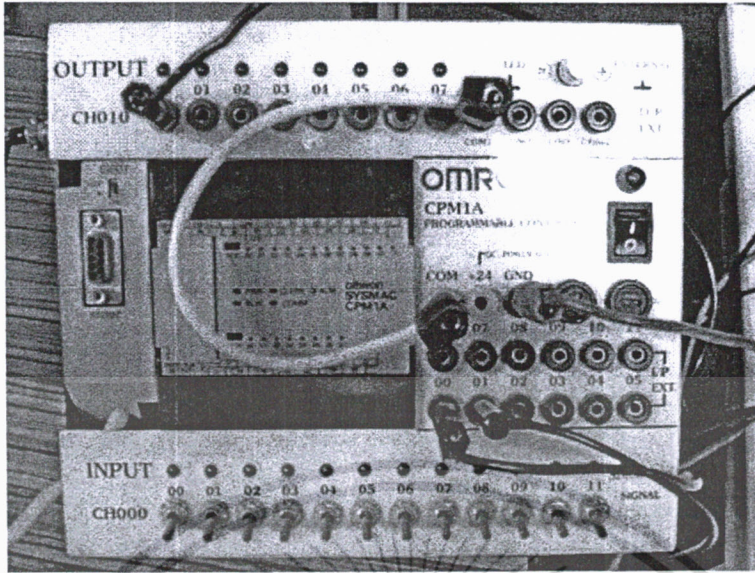
ทำหน้าที่รับสัญญาณแรงดันและทำการลดระดับแรงดันในแต่ละเฟสลงในอัตราส่วน 1/100 จาก 220 Volt ให้เหลือ 2.2 Volt เพื่อส่งสัญญาณแรงดันเข้าสู่บอร์ด DAQ แสดงในรูปที่ 4.3



รูปที่ 4.3 Differential Probe

4.1.4 ชุด PLC Omron

อุปกรณ์ชุด PLC (Programmable Logic Controller) นี้ทำหน้าที่เป็นชุดตัดต่อวงจรกำลังของตัวมอเตอร์ไฟฟ้าเหนี่ยวนำ เพื่อใช้เริ่มเดินหรือหยุดเดินมอเตอร์ไฟฟ้าเหนี่ยวนำ แสดงในรูปที่ 4.4 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ชุด PLC Omron CPM1A

4.2 ซอฟต์แวร์ LabVIEW [13]

4.2.1 รายละเอียด

LabVIEW เป็นโปรแกรมคอมพิวเตอร์ที่สร้างเพื่อนำมาใช้ในด้านการวัดและเป็นเครื่องมือวัดสำหรับงานทางด้านวิศวกรรม LabVIEW ข้อมาจาก Laboratory Virtual Instrument Engineering Workbench ซึ่งหมายความว่า เป็นโปรแกรมที่สร้างเครื่องมือวัดเสมือนจริงในห้องปฏิบัติการทางวิศวกรรม ดังนั้นจุดประสงค์หลักของการทำงานของโปรแกรมนี้ก็คือการจัดการในด้านการวัดและเครื่องมือวัดอย่างมีประสิทธิภาพ และในตัวของโปรแกรมจะประกอบไปด้วยฟังก์ชันที่ใช้ช่วยในการวัดมากมายและแน่นอนที่สุด โปรแกรมนี้จะมีประโยชน์อย่างสูงเมื่อใช้ร่วมกับเครื่องมือวัดทางวิศวกรรมต่าง ๆ

สิ่งที่ LabVIEW แตกต่างจากโปรแกรมอื่นอย่างเห็นได้ชัดที่สุดก็คือ LabVIEW นี้เป็นโปรแกรมประเภท GUI (Graphic User Interface) โดยสมบูรณ์ นั่นคือไม่จำเป็นต้องเขียน code หรือคำสั่งใด ๆ ทั้งสิ้น และที่สำคัญลักษณะภาษาที่ใช้ในโปรแกรมนี้จะเรียกว่าเป็น ภาษารูปภาพ หรือเรียกอีกอย่างว่า ภาษา G (Graphical Language) ซึ่งจะแทนการเขียนโปรแกรมเป็นบรรทัดอย่างที่คุ้นเคยกับภาษาพื้นฐาน เช่น C, BASIC หรือ FORTRAN ด้วยรูปภาพหรือสัญลักษณ์ทั้งหมด ซึ่งแม้ว่าในเบื้องต้นอาจจะสับสนกับการจัดเรียงหรือเขียนโปรแกรมบ้าง แต่เมื่อคุ้นเคยกับการใช้โปรแกรมนี้แล้วจะพบว่า LabVIEW นี้มีความสะดวกและสามารถลดเวลาในการเขียนโปรแกรมลงไปได้มาก โดยเฉพาะในงานเขียนโปรแกรมคอมพิวเตอร์เพื่อเชื่อมต่อกับอุปกรณ์อื่น ๆ เพื่อใช้ในการวัดและการควบคุม

สำหรับผู้ที่เคยใช้โปรแกรมประเภทที่ใช้ตัวหนังสือ หรือที่เรียกว่า Text Base ทั้งหมด คงจะทราบถึงความยุ่งยากในการจัดการกับตำแหน่งการส่งผ่านข้อมูลตามอุปกรณ์เชื่อมต่อเช่น Port หรือ Card ต่าง ๆ แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

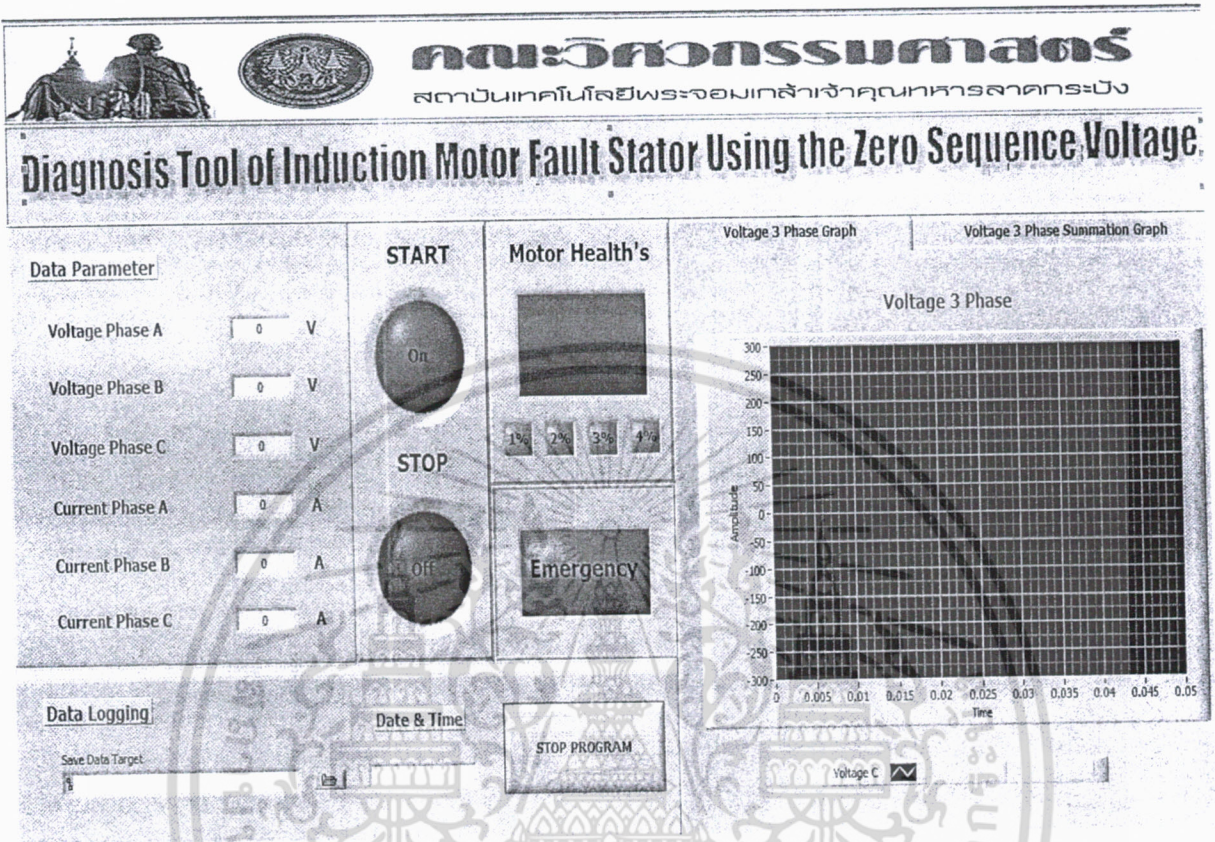
รวมถึงการจัดวางตำแหน่งในหน่วยความจำเพื่อที่จะสามารถรวบรวม ข้อมูลมาใช้ในการคำนวณและเก็บ ข้อมูลให้ได้ประโยชน์สูงสุด ปัญหาเหล่านี้ได้รับการแก้ไขใน LabVIEW โดยได้มีการบรรจุโปรแกรม จำนวนมาก หรือ Libraries ไว้สำหรับจัดการกับปัญหาเหล่านั้น ไม่ว่าจะอุปกรณ์การเชื่อมต่อจะเป็น DAQ (Data Acquisition), GPIB (General Purpose interface Bus หรือก่อนหน้านี้รู้จักกันในชื่อ Hewlett Packard Interface Bus, HP-IB), พอร์ตอนุกรม หรือ Serial Port เพื่อใช้ติดต่อกับอุปกรณ์ที่ส่งผ่านข้อมูล แบบอนุกรม (Serial Instrument) รวมถึงการวิเคราะห์ข้อมูลที่ได้ด้วยวิธีการต่าง ๆ นอกจากนี้ใน Libraries เหล่านี้ยังได้บรรจุฟังก์ชันการทำงานที่สำคัญอีกหลายประการเช่น signal generation, signal processing, filters, สถิติ, พีชคณิต และคณิตศาสตร์อื่น ๆ ดังนั้น LabVIEW จึงทำให้การวัดและการใช้เครื่องมือวัด กลายเป็นเรื่องง่ายลงไปมาก และทำให้คอมพิวเตอร์ส่วนบุคคลกลายเป็นเครื่องมือทางด้านการวัดหลายชนิด อยู่ในเครื่องเดียว

โปรแกรมที่เขียนขึ้นมาโดย LabVIEW จะเรียกว่า Virtual Instrument (VI) เพราะลักษณะที่ปรากฏ ทางจอภาพเมื่อผู้ใช้ใช้งานจะเหมือนกับเครื่องมือหรืออุปกรณ์ทางวิศวกรรม ในขณะเดียวกันหลังจากของ อุปกรณ์เสมือนจริงเหล่านั้นจะเป็นการทำงานของ ฟังก์ชัน, Subroutines และ โปรแกรมหลักเหมือนกับ ภาษาทั่วไป สำหรับ VI หนึ่ง ๆ จะประกอบด้วยส่วนประกอบที่สำคัญสามส่วนคือ

1. Front Panel
2. Block Diagram
3. Icon และ Connector

ทั้งสามส่วนนี้จะประกอบกันขึ้นมาเป็นอุปกรณ์เสมือนหน้าที่ของส่วนประกอบทั้งสามมี ดังต่อไปนี้

ก. Front Panel หรือหน้าปัทม์ จะเป็นที่ใช้สื่อความกันระหว่างผู้ใช้กับ โปรแกรม (หรือ ที่นิยมเรียก user interface) โดยทั่วไปจะมีลักษณะเหมือนกับหน้าปัทม์ของของเครื่องมือหรืออุปกรณ์ที่ใช้ งานด้านการวัดทั่ว ๆ ไป โดยทั่วไปจะประกอบด้วย สวิตช์ปิดเปิด, ปุ่มบิด, ปุ่มกด จอแสดงผลหรือแม้แต่ ค่าที่ผู้ใช้สามารถกำหนด ดังนั้นสำหรับผู้คุ้นเคยกับการเขียน โปรแกรมประเภท Visual ทั้งหมดคงจะ เข้าใจถ้าหากบอกว่า Front Panel นี้จะเปรียบเสมือนเป็น GUI ของ โปรแกรมหรือ VI นั่นเอง ลักษณะของ Front Panel แสดงในรูปที่ 4.5

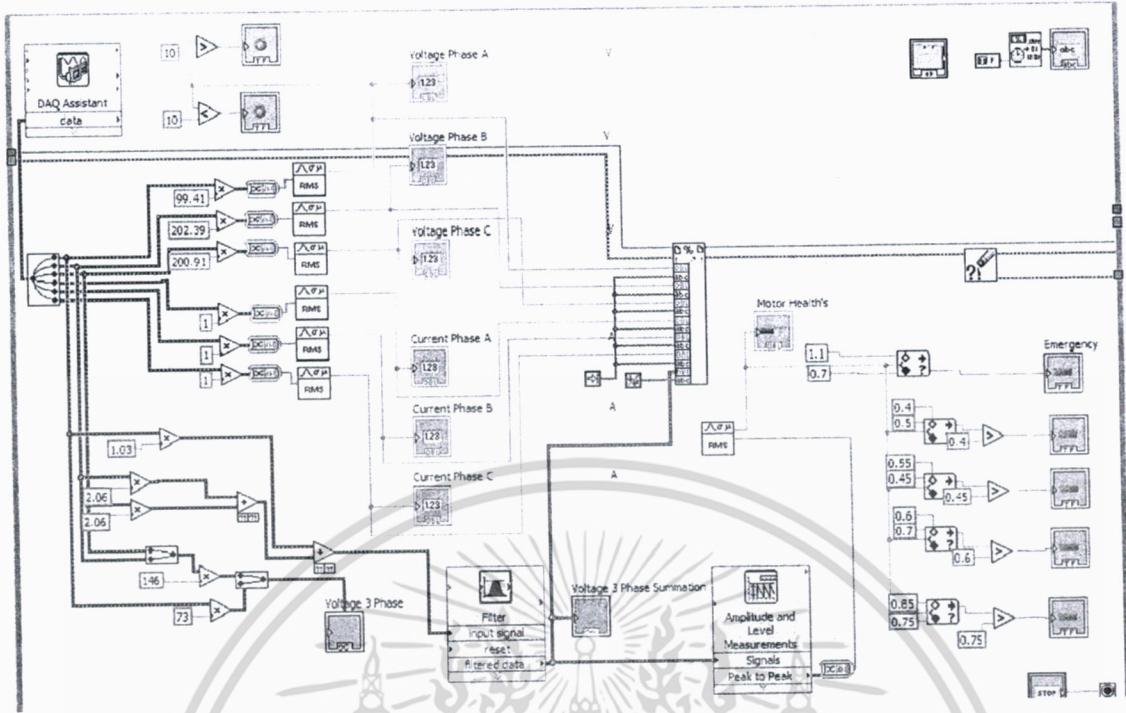


รูปที่ 4.5 Front Panel

ข. Block Diagram เพื่อให้เกิดความเข้าใจง่ายขึ้น อาจมอง Block Diagram นี้เป็นเสมือนกับ Source Code หรือ โปรแกรมของ LabVIEW ซึ่งปรากฏว่าอยู่ในรูปของภาษา G ซึ่ง Block Diagram นี้ ถือเป็น Executable Program คือสามารถที่จะทำงานได้ทันที และข้อดีอีกประการหนึ่งก็คือ LabVIEW จะมีการตรวจสอบความผิดพลาดของโปรแกรมตลอดเวลา ทำให้โปรแกรมจะทำงานได้ก็ต่อเมื่อไม่มีข้อผิดพลาดในโปรแกรมเท่านั้น โดยผู้ใช้งานสามารถที่จะดูรายละเอียดของความผิดพลาดแสดงให้เห็นได้ตลอดเวลาทำให้การเขียนโปรแกรมนั้นง่ายขึ้นมาก

ส่วนประกอบภายใน Block Diagram นี้จะประกอบด้วย ฟังก์ชัน ค่าคงที่ โปรแกรมควบคุมการทำงานหรือโครงสร้าง จากนั้นในแต่ละส่วนเหล่านี้ ซึ่งจะปรากฏในรูปของ Block จะได้รับการต่อสาย (wire) สำหรับ Block ที่เหมาะสมเข้าด้วยกัน เพื่อกำหนดลักษณะการไหลของข้อมูลระหว่าง block เหล่านั้น ทำให้ข้อมูลได้รับการประมวลผลตามที่ต้องการ และแสดงผลออกมาให้แก่ผู้ใช้ต่อไป

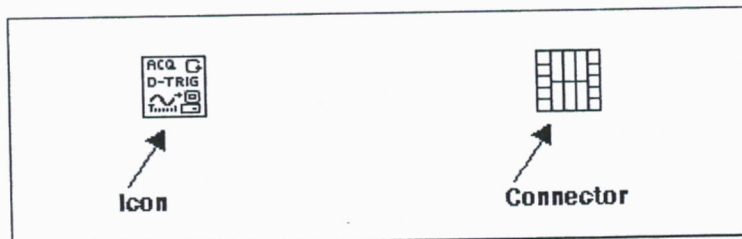
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 Block Diagram

ค. Icon และ Connector เปรียบเสมือนโปรแกรมย่อย Subroutine ในโปรแกรมปกติทั่วไป โดย icon จะหมายถึง block diagram ตัวหนึ่งที่มีการส่งข้อมูลเข้าและออกผ่านทาง Connector ซึ่งใน LabVIEW จะเรียก Subroutine นี้ว่า subVI ข้อดีของการเขียนโปรแกรมด้วยภาษา G นี้ก็คือสามารถสร้าง VI ที่ละส่วนขึ้นมาให้ทำงานด้วยตัวเองได้อย่างอิสระ จากนั้นในภายหลังหากต้องการก็สามารถเขียนโปรแกรมอื่นขึ้นมาเพื่อเรียกใช้งาน VI ที่เคยสร้างขึ้นก่อนหน้านั้นทีละตัว ซึ่งทำให้ VI ที่เขียนขึ้นก่อนกลายเป็น subVI ไป การเขียนในลักษณะนี้เรียกว่า เขียนเป็น module

สำหรับลักษณะทั่วไปของ Icon และ Connector จะแสดงในรูปที่ 4.7 ซึ่งจะเห็นว่าเมื่อแสดงในรูปของ Connector จะพบว่ามีช่องต่อข้อมูลหรือที่เรียกว่า Terminal ให้เห็น



รูปที่ 4.7 Connector ของ subVI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำศัพท์ต่าง ๆ ที่ใช้กันใน LabVIEW นี้มักจะแตกต่างจากที่ใช้กันในภาษาการเขียนโปรแกรมตัวหนังสือทั่ว ๆ ไปในหลาย ๆ ด้าน ดังนั้นเพื่อให้ผู้ที่เริ่มใช้ LabVIEW เข้าใจถึงศัพท์ต่าง ๆ ที่ใช้ในโปรแกรม จึงขอเปรียบเทียบศัพท์ใช้ใน LabVIEW กับโปรแกรมพื้นฐานทั่ว ๆ ไปตามตารางที่ได้แสดงต่อไปนี้

ตารางที่ 4.1 ส่วนประกอบ LabVIEW

LabVIEW	โปรแกรมพื้นฐาน	หน้าที่
VI	Program	ตัวโปรแกรมหลัก
Function	function	ฟังก์ชันสำเร็จรูปที่สร้างขึ้นมากับโปรแกรมนั้นเช่น sin, log เป็นต้น
SubVI	Subroutine	โปรแกรมย่อยที่ถูกเรียกใช้โดยโปรแกรมหลัก
Front Panel	user interface	ส่วนที่ติดต่อกับผู้ใช้
Block Diagram	Program code	การเขียนตามขั้นตอนของทีแต่ละโปรแกรมกำหนดขึ้น

การสร้างโปรแกรม SubVI

SubVI คือ ในการสร้างโปรแกรม VIs ที่มีขนาดใหญ่และมีความสลับซับซ้อนจะทำให้ยากในการตรวจสอบการทำงานของโปรแกรม ในโปรแกรม LabVIEW นั้นจะสามารถสร้างโปรแกรมที่มีขนาดใหญ่ให้เหลือเพียงไอคอนตัวเดียวซึ่งเรียกวิธีการนี้ว่าการสร้าง SubVI ในตัวอย่างการสร้าง SubVI เป็นโปรแกรม บวก คูณ และเปรียบเทียบ ซึ่งฟังก์ชันการทำงานของโปรแกรมจะมีอยู่ 3 ตัวซึ่งเริ่มที่จะมีฟังก์ชันการทำงานที่มากขึ้นสามารถ ขุดให้เป็นไอคอนเพียงตัวเดียวได้เพื่อให้สามารถตรวจสอบโปรแกรมได้ง่ายขึ้นและลดขนาดของโปรแกรมด้วยการสร้าง SubVI สามารถสร้างได้จาก VI ในหน้าต่างบล็อกไดอะแกรมเมื่อสร้าง SubVI เสร็จแล้วฟังก์ชันการทำงานจะเหลือเป็นไอคอนตัวเดียวและที่ไอคอนจะมีคอนเน็คเตอร์ของอินพุตและเอาต์พุตเพื่อต่อใช้งานกับ VI

Terminal

ทุกครั้งที่สร้าง Control หรือ Indicator บน Front Panel ใน Window ของ Block Diagram จะปรากฏ Terminal ขึ้น Terminal ก็คล้ายกับสถานีของข้อมูลก็จะเป็นทั้งสถานีต้นทางของข้อมูลถ้า Terminal นั้นเป็น Terminal ของ Controls และขณะเดียวกันจะเป็นทั้งสถานีปลายทางของข้อมูลถ้า Terminal นั้นเป็น Terminal ของ Indicator

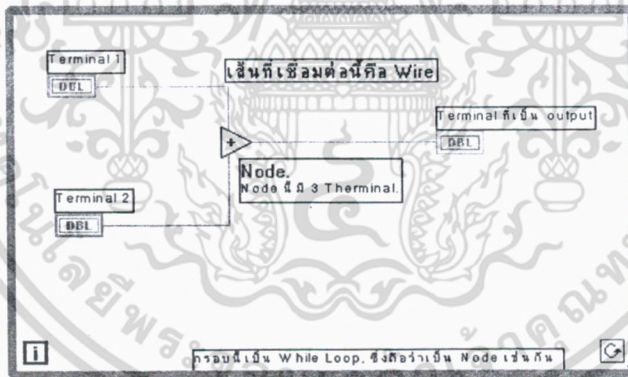
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อที่ควรเข้าใจอย่างหนึ่งก็คือ Object นี้เกิดขึ้นจากการเขียนขึ้นบน Front Panel ดังนั้นเมื่อจะไม่สามารถลบ Terminal ออกจาก Block Diagram ได้ และถ้าหากจะลบ Control หรือ Indicator นั้นออกไปจาก Front Panel แล้ว Terminal เหล่านี้ก็จะหายไปจาก Block Diagram เช่นกัน

Node

Node เป็นคำที่ใช้เรียก object ที่ทำการวิธีใด ๆ เพื่อประมวลข้อมูลใน Block Diagram เช่นเดียวกับที่เขียน Flow Chart แล้วใช้สัญลักษณ์ต่าง ๆ แทนวิธีการวิเคราะห์ข้อมูล เมื่อมีข้อมูลเข้าสู่ node สิ่งที่เกิดขึ้นภายใน node ก็จะขึ้นอยู่กับว่าจะกำหนดให้ข้อมูลที่ส่งเข้าไปนั้น จะมีการประมวลผลอย่างไร ซึ่งอาจจะเป็นการบวก ลบ คูณ หาร หาค่ายกกำลัง หรือเป็นประเภทการเปรียบเทียบข้อมูล ว่ามากหรือน้อยกว่า หรืออื่น ๆ ซึ่งจะเป็นการประมวลผลทางคณิตศาสตร์ทั่วไป นอกเหนือจากนั้น node นี้จะมีส่วนที่เรียกว่า Function แบบต่าง ๆ ซึ่งจะเหมือนกับ Function สำเร็จรูปเช่น sine, cosine, log เป็นต้น ซึ่งก็จะเหมือนกับในภาษาที่เป็นตัวอักษรทั่ว ๆ ไป

ในรูปที่ 4.8 จะแสดงถึงลักษณะของ Node และ Terminal ที่บรรจุอยู่ภายใน Block Diagram



รูปที่ 4.8 ลักษณะของ Node และ Terminal ที่บรรจุอยู่ภายใน Block Diagram

นอกเหนือจากการประมวลผลทางคณิตศาสตร์แล้ว ยังมี node ประเภท Structure หรือ Control Flow อีกด้วย (ในภาษาตัวอักษร Structure Command จะเป็น คำสั่งจำพวก IF...THEN, FOR..., WHILE...)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wires

ขณะที่มีที่มาของข้อมูล ส่วนประมวลหรือปรับแต่งข้อมูล และส่วนแสดงผลข้อมูลเรียบร้อยแล้ว ขั้นตอนต่อไปคือจะต้องสามารถควบคุมการส่งผ่านข้อมูลให้เป็นไปตามที่ต้องการ อุปกรณ์ที่ใช้ใน LabVIEW ก็คือ การต่อสาย หรือ Wires ซึ่งจะเป็นการเชื่อมการส่งข้อมูลระหว่าง terminal หรือ node ต่าง ๆ ที่มีใน Block Diagram นี้เข้าด้วยกัน โดย wires นี้จะเป็นการกำหนดเส้นทางของข้อมูลว่าเมื่อออกจาก terminal หนึ่งแล้ว จะกำหนดการไหลไปที่ node ใดบ้าง มีลำดับเป็นอย่างไร และสุดท้ายจะแสดงผลที่ terminal ใดนั่นเอง ซึ่งการเชื่อมต่อสายนี้จะทำให้เข้าใจถึงหลักการของ Data Flow Programming ได้ดีขึ้น เนื่องจากข้อมูลนั้นมีหลายแบบไม่ว่าจะเป็นเลขทศนิยม, เลขจำนวนจริง, ตัวอักษร หรือค่าจริง-เท็จ (Boolean) ดังนั้นเพื่อแสดงถึงความแตกต่างของข้อมูลแต่ละแบบ LabVIEW จึงได้กำหนดให้ลักษณะของ wires สำหรับข้อมูลแต่ละแบบมีลักษณะของเส้นและสีที่แตกต่างกัน นอกจากนี้ข้อมูลแต่ละแบบดังกล่าวยังอาจมีลักษณะเป็น scalars, 1-D array, 2-D array ได้ซึ่งลักษณะของเส้นของข้อมูลแต่ละแบบก็จะแตกต่างออกไปอีก

ตารางที่ 4.2 ลักษณะของเส้น wire

	Scalars	1-D Array	2-D Array	สี
เลขทศนิยม				ส้ม
เลขจำนวนเต็ม				น้ำเงิน
Boolean				เขียว
ตัวอักษร				ชมพู

โดยปกติการต่อข้อมูลระหว่าง Node หรือ Terminal จะกำหนดแน่นอนว่าต้องการข้อมูลลักษณะใด เช่นถ้า Node ต้องการตัวเลข ก็จำเป็นต้องต่อสายตัวเลขเข้ากับ Node นั้น ถ้าหากต่อสายจากตัวอักษรเข้าไปใน terminal ที่ต้องการตัวเลข สายต่อนั้นจะกลายเป็นสายต่อเสีย หรือ Bad Wire และโปรแกรมก็จะไม่สามารถทำงานต่อไปได้

ทั้งนี้ก็จะเข้าใจหลักการของ Data Flow Programming ได้ในขณะที่ได้ต่อสายระหว่าง terminal และ Node ต่าง ๆ เข้าด้วยกัน ซึ่งหลักการทำงานของ Data Flow Programming จะต่างจากการเขียน โปรแกรมโดยใช้ตัวอักษร เพราะในโปรแกรมตัวอักษรคำสั่งจะถูกส่งเข้าสู่ส่วนประมวลผลทีละบรรทัด เพื่อการคำนวณตามลำดับบรรทัด แต่ใน Data Flow นั้น โปรแกรมจะคำนวณเมื่อมีข้อมูลส่งเข้ามาถึง Input ของ Node นั้น ๆ ครบ และเมื่อข้อมูลส่งครบเข้าถึง Node นั้นแล้ว จะมีการประมวลผลและส่งค่าที่ได้ออกไปตามการต่อเชื่อมสาย การคำนวณจะเสร็จสิ้นในแต่ละรอบเมื่อข้อมูลส่งข้ามไปถึง Terminal สุดท้าย การเชื่อมสายที่ผิดพลาดหรือการคำนวณที่ผิดพลาดจะส่งผลให้โปรแกรมหยุดทำงานทันที และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลไม่ได้เป็นไปตามลำดับการจัดวางคือไม่ได้ทำจากซ้ายไปขวาหรือบนลงล่าง แต่เป็นไปตามขั้นตอนการเดินทางของข้อมูลซึ่งในการเขียน โปรแกรมในเบื้องต้นผู้ที่คุ้นเคยกับภาษาตัวหนังสือ โดยทั่วไปอาจจะต้องใช้เวลาสักครู่เพื่อที่จะทำความเข้าใจการทำงานของ LabVIEW

Icon และ Connector

ถ้าคุ้นเคยกับการเขียนโปรแกรมเป็นตัวอักษร คงทราบว่าสามารถเขียน โปรแกรมย่อยขึ้นมาเพื่อใช้ร่วมกับโปรแกรมหลัก โดยการเขียน โปรแกรมย่อยหรือ Subroutine นี้จะมีประโยชน์มากในกรณีที่จะต้องทำการประมวลผลย่อย ในภาษารูปภาพก็เช่นกันสามารถที่จะสร้าง Subroutine ขึ้นมาได้ ซึ่งจะเรียกว่า subVI โดย สำหรับข้อดีของการเขียนด้วยภาษารูปภาพนี้ก็คือ ทุก VI ที่เขียนขึ้นมาสามารถทำหน้าที่เป็น subVI ได้ แต่เนื่องจากจำเป็นจะต้องกำหนดลักษณะของ subVI ให้เป็นรูปภาพ และมีช่องที่จะต้องส่งเข้าสู่ subVI นั้น จึงได้กำหนด Icon และ Connector ขึ้น

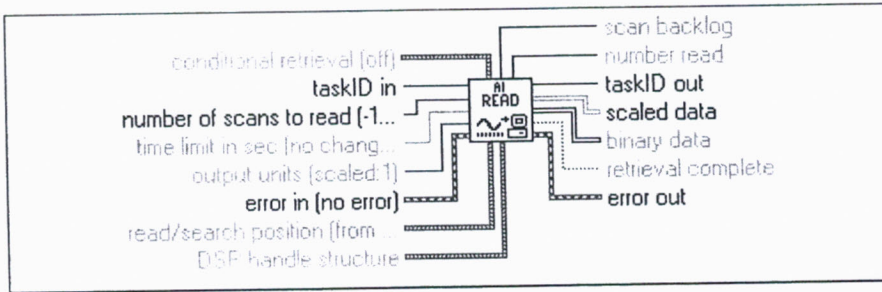
หากจะกล่าวคร่าว ๆ ไอคอนนี้ก็อาจจะมองว่าเป็น node ในอีกรูปแบบหนึ่งก็ได้ โดยไอคอนจะหมายถึง Node ของ subVI ในทุกครั้งที่เขียน VI จะพบว่า LabVIEW จะให้ VI นั้นสามารถทำงานเป็น subVI ได้ถ้าหากต้องการ โดยที่ LabVIEW จะกำหนด icon ให้กับทุก VI ที่เขียนขึ้น ซึ่งสามารถเปลี่ยนแปลงรูปแบบของไอคอนที่ LabVIEW เขียนขึ้นได้



โดยหากเปิด LabVIEW ใหม่ขึ้นมาจะสามารถสังเกตรูปของ Icon ที่ LabVIEW กำหนดขึ้นมาได้ โดยรูปของ ไอคอนจะปรากฏอยู่ที่มุมบนด้านซ้ายใต้ Title Bar ของหน้าต่าง Block Diagram และ Front Panel

เมื่อนำ VI นี้ ไปใช้เพื่อเป็น subVI สามารถที่จะกำหนดให้มี Input และ Output ที่จะเข้าและออกจาก VI ได้ ซึ่งการเข้าและออกของข้อมูลสู่ subVI นั้นจะผ่านทาง Connector ซึ่งหากจะเปรียบเทียบกับภาษาตัวอักษรที่คุ้นเคยแล้ว การใช้ Connector ก็เหมือนกับการใช้คำสั่ง parameter ในภาษา C หรือการใช้คำสั่ง Function ในภาษา FORTRAN นั่นคือการกำหนดว่าข้อมูลใดเป็นข้อมูลที่ส่งไปสู่ Subroutine และข้อมูลใดเป็นข้อมูลที่จะรับกลับออกมาจาก Subroutine ทำนองเดียวกันกับใน LabVIEW ก็คือส่งข้อมูลเข้าสู่ไอคอนหรือ subVI โดยผ่านทาง Input connector เมื่อข้อมูลได้รับการประมวลผลใน subVI แล้วก็ส่งกลับมาทาง Output Connector Terminal

โดยปกติแล้ว Connector จะถูกบังอยู่ด้านหลังของรูป Icon สามารถแสดงให้เห็น Connector ได้ โดยการใช้คำสั่ง Show Connector ซึ่งรายละเอียดเหล่านี้จะกล่าวในภายหลัง ในรูปที่ 4.9 จะเป็นการแสดง Icon และ Connector ของ VI หนึ่งซึ่งเป็นส่วนที่มาพร้อมกับ LabVIEW ส่วนที่เป็นรูปตรงกลางเรียก ไอคอนและส่วนที่เป็นสายต่อต่าง ๆ โดยเรียกเป็น Connector



รูปที่ 4.9 ไอคอนและ Connector

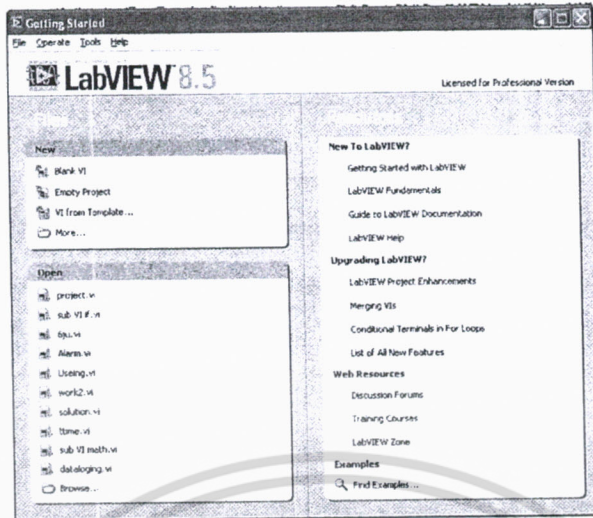
สำหรับรายละเอียดเกี่ยวกับการสร้าง Icon และการกำหนด Connector นี้จะกล่าวถึงในบทต่อไป หลังจากที่ได้ทำความคุ้นเคยกับการเขียนโปรแกรมด้วย LabVIEW แล้ว

4.2.2 Menu และ Palette

เมื่อเปิดโปรแกรม LabVIEW ขึ้นมาทำงานในครั้งแรก ใน LabVIEW 8.5 จะพบกับหน้าต่างของ แรกของ LabVIEW ในลักษณะดังรูปที่ 4.10 หน้าต่างนี้จะเป็นการสอบถามว่า ต้องการจะทำอะไรต่อไปนี้ เช่น ต้องการจะเขียน VI ใหม่หรือแก้ไข VI เดิม หรืออาจจะค้นหาตัวอย่าง VI รวมถึงการใช้ LabVIEW Tutorial ด้วย สำหรับในขั้นนี้ต้องการจะดูส่วนประกอบต่าง ๆ ของ LabVIEW ดังนั้นต้องเปิด VI ใหม่ขึ้น นั่นคือจากหน้าต่างแรกให้เลือก New VI ซึ่งเมื่อเลือกส่วนนี้ LabVIEW จะปรากฏหน้าต่าง ขึ้นอีกหลาย ชุด ซึ่งจะประกอบด้วย หน้าต่างต่อไปนี้

- Front Panel
- Block Diagram
- Controls Palette
- Functions Palette
- Tools Palette

อย่างไรก็ตามหน้าต่าง และ Palette เหล่านี้จะไม่ได้ปรากฏพร้อมกันทั้งหมดในตอนแรก จะมี เฉพาะบางส่วนเท่านั้นที่พบในตอนแรกที่เปิด LabVIEW เข้าไปทำงานใหม่ โดยหน้าต่างที่ active หรือ พร้อมใช้งานหน้าต่างแรกคือหน้าต่าง front panel



รูปที่ 4.10 หน้าต่างของ LabVIEW

รูปร่างของ Front Panel Windows ก็จะมีลักษณะคล้ายกับหน้าต่างของโปรแกรมที่ทำงานบน Windows ทั่ว ๆ ไป คือจะประกอบด้วย

Title Bar

Title Bar ซึ่งจะบอกชื่อของ Program ซึ่งในขณะนี้ควรได้ชื่อ Untitled 1 เพราะยังไม่ได้กำหนดชื่อ VI เพื่อ Save และจะมี ปุ่มหรือ Button เพื่อ Minimize, Maximize และ exit Button อยู่ด้านขวามือของ Title Bar เหมือนโปรแกรมอื่น ๆ ใน Windows



Menu Bar

Menu Bar ประกอบด้วยเมนูต่าง ๆ คล้ายกับโปรแกรมอื่นใน Windows คือ จะประกอบด้วย File, Edit, Operate, Project, Windows และ Help ซึ่งการเลือกใช้ได้ก็ทำได้โดย Click ที่เมอนูนั่น



หากลอง Click เมนูทีละส่วนตามไปจะได้ Pull Down Menu ออกมาจากแต่ละ menu หลัก ซึ่งในแต่ละ menu มีคำสั่งในแต่ละลักษณะต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- File Menu จะเป็นคำสั่งที่จะเปิด, เก็บ, พิมพ์ VI
- Edit Menu จะเป็นคำสั่งที่จะแก้ไข เช่น undo (เริ่มมีใน LabVIEW 5.0), cut, copy,

Preference

- Operate Menu จะเป็นชุดคำสั่งเพื่อให้ LabVIEW ทำงานเช่น run, stop และอื่น ๆ
- Project Menu จะเป็นชุดคำสั่งเพื่อบังคับการทำงานของ LabVIEW เมนูนี้จะมีประโยชน์

เมื่อทำงานกับ VI ขนาดใหญ่และมี subVI หลาย ๆ ชุดใน VI หลัก

- Windows Menu จะเป็นการบังคับให้เปิดหน้าต่างที่ต้องการ เช่น Front Panel หรือ Block Diagram รวมทั้งแสดง Palette ต่าง ๆ ด้วย
- Help menu จะเป็นการใช้เมื่อต้องการคำอธิบายหรือความช่วยเหลือต่าง ๆ ใน LabVIEW




เนื่องจาก LabVIEW มีคำสั่งเป็นจำนวนมากในเอกสารนี้ไม่ได้มีจุดมุ่งหมายที่จะให้ผู้อ่านเข้าใจคำสั่งทุกคำสั่งแต่เป็นการอธิบายเพื่อให้สามารถเริ่มใช้ LabVIEW ได้ง่ายขึ้น สำหรับรายละเอียดของทุกคำสั่ง ผู้อ่านคงจะต้องอ่านจากเอกสารที่มาพร้อมกับ LabVIEW


Toolbar

บน Toolbar ของ LabVIEW นี้ก็จะคล้ายกับ Toolbar ของโปรแกรมต่าง ๆ ที่ทำงานบน Windows นั่นคือเป็นการรวบรวมคำสั่งที่ใช้อยู่ประจำให้มาอยู่ในรูปของ button เพื่อสะดวกในการใช้ สำหรับ toolbar ของ block diagram จะมีมากกว่า ของ front panel อยู่เล็กน้อย






สำหรับแต่ละ button บน toolbar จะมีชื่อและหน้าที่ดังนี้

 **Run Button** มีลักษณะเป็นลูกศร ใช้สั่งการให้ VI ทำงานเมื่อคลิก ที่ปุ่มนี้ลักษณะของ Run button จะเปลี่ยนไปตามการทำงานของ VI ในขณะนั้นคือ จะเปลี่ยนลักษณะเป็นลูกศรวิ่ง  เมื่อโปรแกรมกำลังทำงาน และถ้าหากว่า Run button ปรากฏเป็นรูปลูกศรขาด  แสดงว่า VI นั้นยังไม่พร้อมทำงานหรือมีข้อผิดพลาดอยู่ใน VI และคำว่า BROKEN VI จะเป็นศัพท์ที่นิยมใช้เมื่อ โปรแกรมหรือ VI นั้นเกิดความผิดพลาดขึ้นแล้วทำให้ลูกศรขาด

 **Continuous Run Button** มีลักษณะเป็นลูกศรวน เนื่องจากการทำงานของ Data Flow จะเริ่มต้นเมื่อได้รับข้อมูล และสิ้นสุดลงเมื่อเสร็จสิ้นการประเมินผลข้อมูล อย่างไรก็ตามในกรณีที่ต้องการให้ VI ทำงานซ้ำต่อไป นั่นคือ เมื่อจบการทำงานในครั้งแรกก็ให้กลับไปเริ่มต้นทำใหม่ไปเรื่อย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถจะใช้ปุ่มนี้เป็นเครื่องมือทำคำสั่งนี้ได้ และเมื่อ VI ทำงานอย่างต่อเนื่องปุ่มนี้จะมีลักษณะเป็น ลูกศรวนสี่ดำ 

 หรือ  **Abort Button** มีลักษณะเป็นเครื่องหมายจราจรให้หยุด ก่อนที่จะสั่งให้ VI ทำงานปุ่มนี้จะใช้งานไม่ได้ (Inactive และมีสีเทา) แต่เมื่อให้โปรแกรมทำงานด้วย Run หรือ Continuous Run ก็ตาม ปุ่มนี้จะปรากฏให้ใช้งานได้ (active) หากกดปุ่มนี้จะเป็นการยกเลิกการทำงานของ VI ข้อควรระวังอย่างหนึ่งก็คือ LabVIEW ไม่แนะนำให้ใช้ Abort button เพื่อใช้เป็นการหยุดการทำงานในสภาพปกติเพราะจะทำให้ข้อมูลบางส่วนสูญหายหรือค้างอยู่ในหน่วยความจำได้ เพราะกระบวนการคำนวณยังไม่สิ้นสุด LabVIEW แนะนำว่าในโปรแกรมควรสร้าง คำสั่ง ขึ้นมาหยุดการทำงาน of โปรแกรมที่เหมาะสม ซึ่งจะแสดงให้เห็นต่อไป


 **Pause Button** ปุ่มนี้เหมือนกับปุ่มบนเครื่องเสียงหรือ VCR ทั่วไป คือสั่งให้โปรแกรมหยุดการทำงานชั่วคราว เพื่อจะสามารถแก้ไขโปรแกรมได้ เช่น สั่งให้ข้ามบางขั้นตอนหรือออกจากบางขั้นตอน (step out) เป็นต้น


   **Single Step Button** ประกอบด้วยปุ่ม Step Into, Step Over, Step Out เป็นการบังคับให้กับ VI ทำงานตามขั้นตอนที่ต้องการ

 **Execution Highlighting Button** จะสั่งให้ VI ทำการ Highlight การไหลผ่านของข้อมูลเมื่อผ่านไปบนส่วนต่าง ๆ ของ diagram และจะเห็นว่ามีการแสดงค่าของข้อมูลเมื่อผ่านส่วนต่าง ๆ ในขณะนั้นไปด้วย

 **Warning Button** เป็นการเตือนการทำงานของ LabVIEW ถ้าหากว่ากำหนดให้ LabVIEW มีการเตือนและเมื่อปุ่มนี้ปรากฏสามารถให้แสดงข้อมูลที่มีการเตือน โดยการ click ที่ปุ่มนี้ การปรากฏปุ่มเตือนนี้ไม่ให้เห็นว่าเกิดการผิดพลาดเพียงแต่เตือนว่ามีสิ่งผิดปกติเกิดขึ้นในโปรแกรม

 **Front Ring** สามารถเปลี่ยนรูปแบบตัวหนังสือที่ปรากฏบน VI ได้โดยใช้ Front Ring นีบน toolbar ซึ่งก็จะเหมือนกับ Front Ring บนโปรแกรม word Programming ทั่ว ๆ ไป ที่ทำงานบน Windows

 สำหรับผู้เคยใช้โปรแกรมประเภท GUI เช่น โปรแกรม Visual ต่าง ๆ คงจะเคยใช้คำสั่ง **Alignment** และ **Distribution** ซึ่งคำสั่งเหล่านี้จะเป็นการจัดวางตำแหน่งของส่วนประกอบต่าง ๆ ใน Front Panel และ Block Diagram ให้ตรงกันหรือมีระยะห่างที่ต้องการซึ่งเป็นคำสั่งเพื่อเพิ่มความสวยงามให้กับ VI

 **Alignment Ring** ใช้เมื่อต้องการวางแนวของ Object ต่าง ๆ ให้อยู่ในลักษณะแนวที่ต้องการ

 **Distribution Ring** ใช้เมื่อต้องการจัดระยะระหว่าง Object ตามแนว ให้เป็นไป

เอกตามต้องการ สารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Reorder Ring เป็นส่วนที่ใช้จัดอันดับบน Front Panel และ Block Diagram ว่า

วัตถุที่วางอันใดจะอยู่หน้าสุด อันใดจะอยู่หลังสุดใหม่ เพราะปกติ LabVIEW จะให้วัตถุที่วางลงอันแรก อยู่ล่างสุด และที่วางหลังจะทับอันแรกไปเรื่อย โดยสามารถจัดลำดับได้โดยเลือกตัวเลือกต่อไปนี้ ซึ่งคง คำนึงเกี่ยวกับโปรแกรมวาดรูปต่าง ๆ คืออยู่แล้ว

Move Forward	Ctrl+K
Move Backward	Ctrl+J
Move To Front	Ctrl+Shift+K
Move To Back	Ctrl+Shift+J

โหมดการทำงานของ VI

เมื่อเปิด VI ขึ้นมาครั้งแรก VI นั้นจะอยู่ในโหมดการแก้ไข (Edit Mode) นั่นคือสามารถแก้ไข เปลี่ยนแปลงค่าต่าง ๆ ที่ต้องการได้แต่เมื่อให้ VI ทำงาน VI จะเข้าสู่โหมดการทำงาน (Run Mode) ซึ่งหมายความว่าไม่สามารถแก้ไขโปรแกรมได้ในขณะนี้ แต่อาจเปลี่ยนแปลงค่าของ Control ต่าง ๆ บน Front Panel ได้ โดยทั่วไปแล้ว VI จะอยู่ในโหมดการทำงานจนกว่าจะทำงานเสร็จหรือสั่งให้หยุดการทำงาน และเมื่อ VI หยุดการทำงานใน โหมดการทำงานแล้ว VI จะกลับเข้ามาอยู่ในโหมดการแก้ไข

Pop-Up Menus

นอกจากเมนูที่มีอยู่บน Menu Bar ซึ่งเป็นประเภท Pull-Down Menu แล้วในการเขียน VI จะมีเมนูอีกประเภทหนึ่ง ซึ่งเรียกว่า Pop-up menu ซึ่งเมนูประเภทนี้ จะใช้มากกว่าเมนูแบบแรกในการเขียน VI

การที่จะเรียก Pop-up menu ขึ้นมาใช้วิธีการง่ายที่สุดใน Windows 95, Windows 98 หรือ Windows NT ก็คือ Click บน Object นั้น ๆ ด้วยเมาส์ปุ่มขวา ลักษณะของ Pop-up menu ของแต่ละ Object จะแตกต่างกันออกไปแล้วแต่นิคมของ Object นั้น ๆ

ส่วนประกอบของ Pop-Up Menu

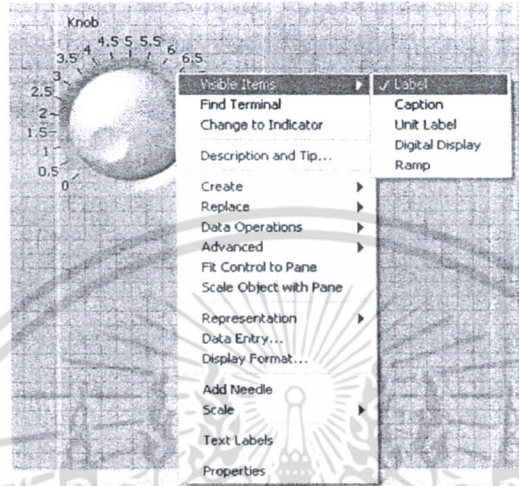
เมื่อเลือก object และสั่งให้แสดง Pop-Up Menu ขึ้นมาแล้ว จะพบว่าเมนูเหล่านั้นบางเมนูสามารถขยายต่อไปสู่ เมนูย่อย ซึ่งเรียก hierarchical menu ต่อไปได้โดยเมื่อดังกล่าวจะมีหัวลูกศรชี้ไปทางขวา ▶ ต่อจากตัวเลือกของ menu นั้น ดังที่แสดงไว้ในรูปที่ผ่านมา

Hierarchical menu นี้บางครั้งจะเป็นการเลือกตัวเลือก โดยจะเป็นการเลือกว่าจะใช้หรือไม่ใช้ตัวเลือกนั้น โดยจะมีเครื่องหมายลูก 4 อยู่หน้าตัวเลือกนั้นหากเลือกใช้

ส่วนในบางตัวเลือกของ Pop-up Menu จะเป็นการกำหนดค่าต่อไปนี้ โดยหากเลือกตัวเลือกนั้นจะเปิด Dialog Box ขึ้นมาใหม่ ซึ่งเมนูตัวเลือกเหล่านั้นจะคำมด้วย ellipses (...) เองนั้นจะเปิด Dialog Box ขึ้นมาใหม่ ซึ่งเมนูตัวเลือกเหล่านั้นจะคำมด้วย ellipses (...) ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนเมนูที่ไม่มีหัวลูกศร ▶ หรือจุด ... จะเป็นเมนูที่เป็นคำสั่งและจะทำงานทันทีที่เลือก
เมื่อนั้น ๆ เช่น Change to indicator, Find Terminal เป็นต้น

สำหรับส่วนประกอบหลัก ๆ ใน Pop-up menu ใน Object ต่าง ๆ จะมีดังนี้



รูปที่ 4.11 ลักษณะของ Pop-up menu ของแต่ละ Object

Change to Control หรือ Change to Indicator

เป็นการเปลี่ยนชนิดของ Control และ Indicator เช่น ถ้าหากว่า object เป็น Indicator อยู่ใน
Pop-up Menu จะแสดงคำสั่ง Change to Control หากเลือกคำสั่งนี้ Indicator ก็จะเปลี่ยนเป็น Control
ทันทีและในทางกลับกันกับคำสั่ง Change to Indicator ข้อควรระวังในการใช้คำสั่งนี้ก็คือ Control Object
และ Indicator Object มีหน้าที่ตรงกันข้ามกันและไม่สามารถที่จะทำงานแทนกันได้ หากเปลี่ยนโดยไม่
ระวังอาจเกิดความผิดพลาดขึ้นมาใน VI ได้

Find Terminal และ Find Control/Indicator

ถ้าหากเลือก Find Terminal จาก Pop-up menu ของ Front Panel จะพบว่า LabVIEW จะ
แสดงและเน้นสีกับ terminal ของวัตถุนั้นของ Object ใน Block Diagram และถ้าหากเลือก Find
Control/Indicator จะพบว่า LabVIEW จะแสดงและเน้นสี Object นั้นบน Front Panel คำสั่งทั้ง 2 นี้ จะ
ปรากฏบน Pop-up menu เฉพาะในกรณีที่มี Object อยู่บนทั้ง Front Panel และ Block Diagram เท่านั้น
เช่น จะไม่พบคำสั่งนี้ถ้าหาก Pop-up menu ของ Node

Show

หลาย ๆ object จะมี Show menu และมักจะเป็น Hierarchical menu ก็จะเป็นการ
กำหนดให้แสดงส่วนต่าง ๆ ของ Object นั้น เช่นแสดงชื่อ (Label) หน่วย เป็นต้น
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Operation

เมนูนี้เป็นเมนูที่จะปรากฏขึ้นเฉพาะขณะที่ VI อยู่ในโหมดการทำงานเท่านั้น ซึ่งจะใช้กับ Control และ Indicator โดยจะมีเมนูย่อยต่อไปนี้

Reinitialize to Default

จะเป็นการกำหนดค่าของ Object ให้กลับไปหาค่าเริ่มต้น (Default) ที่ LabVIEW กำหนดมาให้ครั้งแรก

Make Current Value Default

จะกำหนดค่าในขณะนั้นให้เป็นค่าเริ่มต้นเมื่อเริ่มให้ VI ทำงาน

Cut Data, Copy Data และ Paste Data

เป็นการตัด, คัดลอก หรือใส่ข้อมูล ออกจากหรือลงใน Control หรือ Indicator

Description

เมนูนี้จะเป็นการนำ Dialog Box หรือช่วงข้อความขึ้นมาเพื่อจะได้ใส่หรืออ่านข้อความที่ Object นั้น ๆ ใช้ ในโหมดการทำงานสามารถดูข้อความที่เกี่ยวข้องกับ Object นั้น ได้เท่านั้นไม่สามารถแก้ไขได้

Show หรือ Hide Control/Indicator

เป็นการสั่งให้แสดงหรือไม่แสดง Front Panel Terminals ของ Object นั้นเพื่อผู้ใช้จะไม่สามารถเห็น Object นั้นใน Front Panel แต่ยังคงปรากฏอยู่ใน Diagram ซึ่งอาจจำเป็นต้องใช้ในบางกรณี

Create

เป็นคำสั่งที่ใช้สร้าง Attribute Node หรือ Local Variable ของ Object นั้นซึ่งจะกล่าวต่อไปในภายหลัง

Key Navigation

จะใช้ Key Navigation... เพื่อสั่งการ โดยผ่านแป้นพิมพ์ (Keyboard) โดยการกดปุ่มหลาย ๆ ปุ่ม ขณะที่แสดง Front Panel อยู่ เมื่อผู้ใช้กดปุ่มตามที่ตั้งไว้ขณะที่ VI กำลังทำงาน LabVIEW จะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมือนกับ ผู้ใช้ใช้เมาส์กดบนวัตถุนั้น จะทำให้วัตถุนั้นเป็น Key Focus คือ Active หรือพร้อมทำงานใน Object นั้น

Online Help

ขอความช่วยเหลือในส่วนที่เกี่ยวข้องกับ Object หรือวัตถุนั้น

Replace

เป็นการแทนที่ Object นั้นด้วย Object อื่น เมื่อเลือกตัวเลือกนี้ จะพบว่าจะเข้าสู่ Controls หรือ Functions Palette ขึ้นอยู่กับว่าวัตถุนั้นเป็นวัตถุใดหรืออยู่บนหน้าต่างใด และจะทำให้สามารถแทน Object เพิ่มด้วย Object ใหม่ได้ขณะที่สายเชื่อมเดิมยังคงอยู่

เมื่อเปิด โปรแกรม LabVIEW ขึ้นมาจะพบ หน้าต่างที่บรรจุคำสั่ง หรือ Palette ที่อาจปรากฏให้เห็น 3 หน้าต่างคือ Tools, Controls และ Functions Palettes แม้ว่าโดยปกติหน้าต่างเหล่านี้จะไม่ปรากฏขึ้นพร้อมกัน ถ้าหากว่า Palettes ที่ต้องการ ไม่ปรากฏหรือปรากฏบางตัวอาจใช้ตัวเลือก Show Palettes ภายใต้คำสั่ง Windows บน menu bar เพื่อแสดง Palette นั้นได้

Controls และ Function Palettes

แม้ว่าเมื่อเปิด VI ขึ้นมาจะมี Controls และ Functions Palettes ปรากฏขึ้นแต่ทั้งคู่จะใช้งานไม่พร้อมกันคือ

Controls Palettes

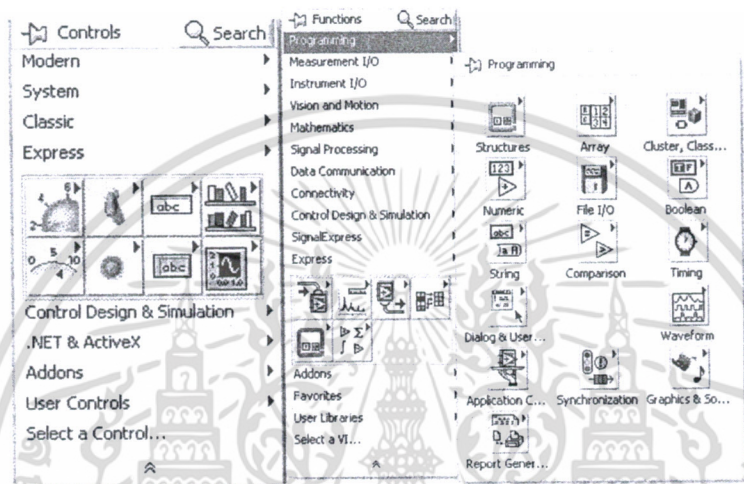
จะปรากฏขึ้นให้เห็นได้ก็ต่อเมื่อ Front Panel อยู่ในสภาพพร้อมทำงานอยู่เท่านั้น ถ้าหาก Front Panel ไม่อยู่ในสภาพพร้อมทำงานส่วนของ Controls Palettes จะหายไป หรืออาจปิดเฉพาะ Controls palettes โดยใช้เมาส์กดที่ Exit Button บน Title bar ก็ได้

Function Palettes

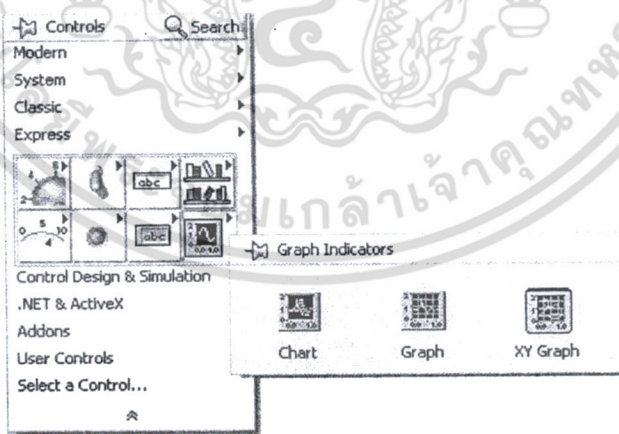
จะปรากฏขึ้นให้เห็นได้ก็ต่อเมื่อ Block Diagram อยู่ในสภาพพร้อมทำงานอยู่เท่านั้นถ้าหาก Block Diagram ไม่อยู่ในสภาพพร้อมทำงาน ส่วนของ Tools Palettes นี้จะหายไป หรือปิด Palette นี้เอง ลักษณะของ Controls และ Functions Palette เป็นไปตามรูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Palettes ทั้งสองจะมี Subpalette บรรจุอยู่ถ้าหากว่าเลื่อนลูกศรของเมาส์ไปที่ปุ่มของ Subpalette ในชื่อของ Subpalette นั้นจะปรากฏขึ้น และหากกดลงไปปุ่มของ Subpalette ใด และกดค้างไว้จะพบว่าจะมีการแสดงส่วนประกอบของ Subpalette นั้นต่อออกไปอีก และในหลายกรณีใน Subpalette ก็จะแยกย่อยเป็น Subpalette ต่อไปอีกดังที่แสดงต่อไปในรูปที่ 4.13



รูปที่ 4.12 ลักษณะของ Controls และ Functions Palette



รูปที่ 4.13 ส่วนประกอบของ Sub Palette







โดยปกติจะแสดง Subpalette ได้ปรากฏขึ้นได้ก็ต่อเมื่อกดเมาส์ปุ่มซ้ายค้างไว้เท่านั้นอย่างไรก็ตามบนทุก subpalette ที่มุมบนด้านขวาจะมีรูปที่ติดกระดาษตามด้วยชื่อของ Subpalette นั้น ที่ติดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระดาศหรือ Thumbtack นี้เป็นตัวช่วยให้สามารถแสดง subpalette นั้นตลอดไปโดยไม่จำเป็นต้องเลือกจาก palette หลัก ซึ่งจะมีประโยชน์ในกรณีที่ใช้ subpalette นั้นบ่อย ๆ วิธีการใช้ก็คือให้ click ที่ตำแหน่งที่ติดกระดาศนั้น Subpalette ก็จะติดอยู่บนจอต่อไปแม้ว่าจะเลื่อนเมาส์ออกไปแล้วก็ตาม และสามารถปิด subpalette นั้นได้เมื่อต้องการเหมือนกับการปิดหน้าต่างอื่น ๆ นอกเหนือจากนั้นสามารถที่จะตกแต่งและเปลี่ยนแปลงลักษณะและสิ่งที่บรรจุอยู่ใน Palette หรือ Subpalette ได้ตามที่ต้องการโดยใช้คำสั่ง Edit Control and Function Palettes จาก Edit menu ซึ่งจะกล่าวต่อไปในภายหลัง





Tools Palette

Tools คือหน้าที่พิเศษของตัวชี้ของเมาส์ จะใช้ Tools เพื่อจะให้ทำงานในการแก้ไขหรือปฏิบัติหน้าที่ซึ่งต้องการ คล้ายกันกับที่ใช้ในโปรแกรมการวาดรูปทั่ว ๆ ไป ใน Tools Palette จะประกอบด้วยปุ่มที่มีหน้าที่ต่าง ๆ ดังนี้ สำหรับหน้าที่ของอุปกรณ์ต่าง ๆ บน Tools Palette นี้สรุปได้ตามตารางที่ 4.3

ตารางที่ 4.3 อุปกรณ์ต่าง ๆ บน Tools Palette

ลักษณะปุ่ม	ชื่อ	หน้าที่
	Operating Tool	ช่วยให้เปลี่ยนค่าของ control บน front panel ขณะ VI ทำงานหรือเปลี่ยนค่าอื่น ๆ ในขณะที่ VI อยู่ในโหมดแก้ไข
	Positioning Tool	ช่วยปรับขนาด, เคลื่อนย้ายที่ หรือเลือก object
	Labeling Tool	สร้างหรือแก้ไข Text
	Wiring Tool	ใช้ในการต่อเชื่อมสายใน block diagram เข้าด้วยกัน
	Pop-up Menu Tool	ถ้าตัวชี้ชี้ไปที่ object ใดก็จะเกิด Pop-up Menu ของ object นั้นขึ้น ใช้แทนการกดเมาส์ปุ่มขวาที่ Object ได้
	Scroll Tool	เลื่อนภาพบนหน้าต่างที่กำลัง active อยู่ไปในทิศทางที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะปุ่ม	ชื่อ	หน้าที่
	Breakpoint Tool	เป็นการใส่ตำแหน่งหยุดลงใน block diagram เมื่อข้อมูลเดินทางมาถึงจุดนี้ การประมวลผลจะชะลอชั่วคราว เพื่อให้ตรวจสอบ และแก้ไขการทำงาน
	Probe Tool	สร้างเครื่องวัดลงบนเส้นเชื่อมเพื่อแสดงค่าข้อมูลในขณะผ่านเครื่องวัดนั้น ๆ
	Color Copy Tool	ใช้ในการคัดลอกสีจาก object ที่ต้องการเพื่อสามารถปรับแก้สีที่ object อื่นให้เหมือน object นั้น
	Color Tool	ใช้ในการปรับแต่งสีของ VI ให้เป็นไปตามต้องการ

4.2.3 HELP

ในการจัดทำเอกสารเพื่อแนะนำการใช้โปรแกรม LabVIEW ขั้นพื้นฐานนี้ คงเป็นไปได้ที่จะบรรยายรายละเอียดทั้งหมดของ LabVIEW ลงในเอกสารนี้ ดังนั้นวิธีการหนึ่งที่จะช่วยได้คือการใช้ help ที่บรรจุมากับ LabVIEW โดย LabVIEW ได้บรรจุคำอธิบายการใช้งานของโปรแกรมมาให้ในหลายลักษณะ

เมื่อต้องการความช่วยเหลือหรือคำอธิบายใด ๆ สามารถสั่งให้ LabVIEW แสดง Help ของ object นั้น ๆ ได้โดยจาก Help menu เลือก Show Help หน้าต่างของ Help ก็จะปรากฏขึ้น หลังจากนั้นถ้านำมาชี้ไปที่ Object ใด Help ของ Object นั้นก็จะปรากฏขึ้นบน Windows โดยปกติ Help ของ Object นั้นก็จะปรากฏขึ้นบน Windows โดยปกติ Help ของ Object นั้นจะบอกถึง

- หน้าที่ของ Object นั้น
- การต่อเชื่อมสายหรือแสดง Connection ของ Object นั้นและอาจมีข้อมูลหรือข้อควรระวังของ Object

ตัวอย่างของ Help แสดงในรูปที่ 4.14 ซึ่งจากในรูปจะเห็นว่าที่มุมล่างซ้ายของหน้าต่าง Help จะพบปุ่มเล็ก ๆ 3 ปุ่ม แต่ละปุ่มมีหน้าที่ดังนี้

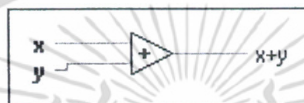
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

☒ **Simple Help** คือบอกความช่วยเหลืออย่างคร่าว ๆ และ Detailed Help คือบอกรายละเอียดต่าง ๆ มากขึ้น ใน Simple Help จะเป็นการบอกรายละเอียดเฉพาะส่วนที่จำเป็น เช่น สายต่อที่จำเป็นเข้าสู่ Icon เป็นต้น จะอธิบายในรายละเอียดต่อไป

☒ **Lock Help** คือได้แสดง Help ของ Object นั้นต่อไปไม่ว่าจะเลื่อนลูกศรของเมาส์ไปที่ใดก็ตามบน Front Panel หรือ Block Diagram

☒ **Online Help** เป็นการแสดงการทำงานของ Object นั้นอย่างละเอียดและอาจมีตัวอย่างการทำงานของ object บางแบบด้วย

Details Example



รูปที่ 4.14 ตัวอย่าง Help

Simple และ Detailed Help

สำหรับจุดเชื่อมหรือ Connection ของ Object นั้นบาง Connector อาจจะเป็น Connector ที่จำเป็นต้องมีค่าเข้าสู่ Object หรือ Node นั้น ในขณะที่บาง Connector อาจเป็นเพียงตัวเลือกว่าจะมีหรือไม่ก็ได้ หากไม่ต่อสายที่ถูกต้องเข้ากับ Connector ที่มีความจำเป็นต้องใช้เข้าสู่ VI จะไม่อยู่ในสภาพที่พร้อมทำงาน และเครื่องหมาย RUN จะเป็นลูกศรขาด รูปที่ 4.15 และ 4.16 จะแสดงความแตกต่างของการแสดง Help ทั้งสองแบบ

การใช้ Detailed Help จะบอกว่า Connector ตัวใดมีความต้องการข้อมูลในลักษณะใด และมีความจำเป็นหรือไม่ ส่วน Simple Help จะเป็นการแสดงเฉพาะ Connector ที่จำเป็นต้องใช้ในการนำข้อมูลไปใช้ในการประมวลผลข้อมูลเท่านั้น โดยทั่วไป

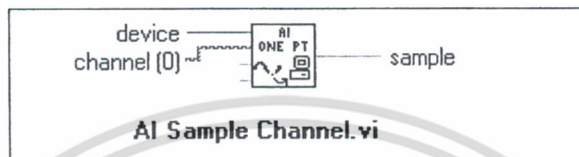
- Connection ที่จำเป็นจะแสดงด้วยตัวหนา
- Connection ที่แนะนำให้ต่อ (แต่ไม่จำเป็น) จะเป็นตัวหนังสือปกติ
- Connection ที่เป็นตัวเลือกจะเป็นตัวหนังสือสีจางและไม่แสดงใน Simple Help

ถ้า Input ของ Function ใดไม่จำเป็นต้องเชื่อมต่อ ซึ่งอาจเป็นในกรณีของแนะนำให้ต่อหรือเป็นตัวเลือกก็ตาม ค่าเริ่มต้นจะปรากฏอยู่ในวงเล็บต่อจากชื่อของ Connection นั้น

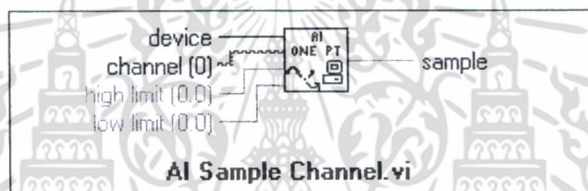
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Online Help

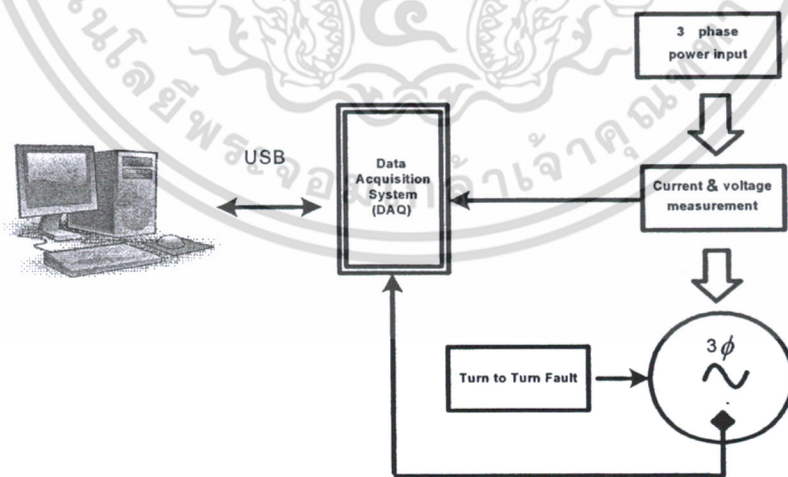
Online Help จะเป็นการอธิบายการทำงานของ LabVIEW อย่างละเอียดซึ่ง Help ในส่วนนี้จะอธิบายขั้นตอน และหน้าที่ของแต่ละ Object อย่างละเอียด ซึ่งจะไม่ขอกกล่าวในที่นี้เพราะการทำงานของ Online Help ของ LabVIEW ก็จะเหมือนกับ Help ของ Program อื่น ๆ บน Windows



รูปที่ 4.15 ตัวอย่าง Simple Help

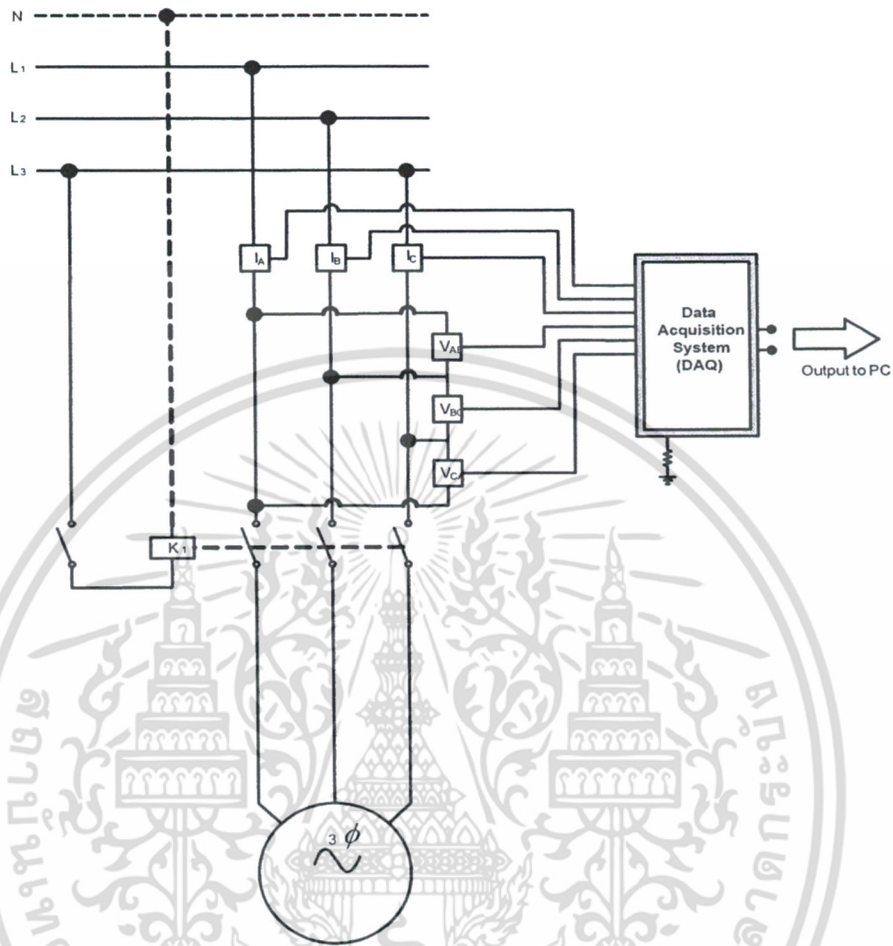


รูปที่ 4.16 ตัวอย่าง Details Help



รูปที่ 4.17 ไคอะแกรมการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

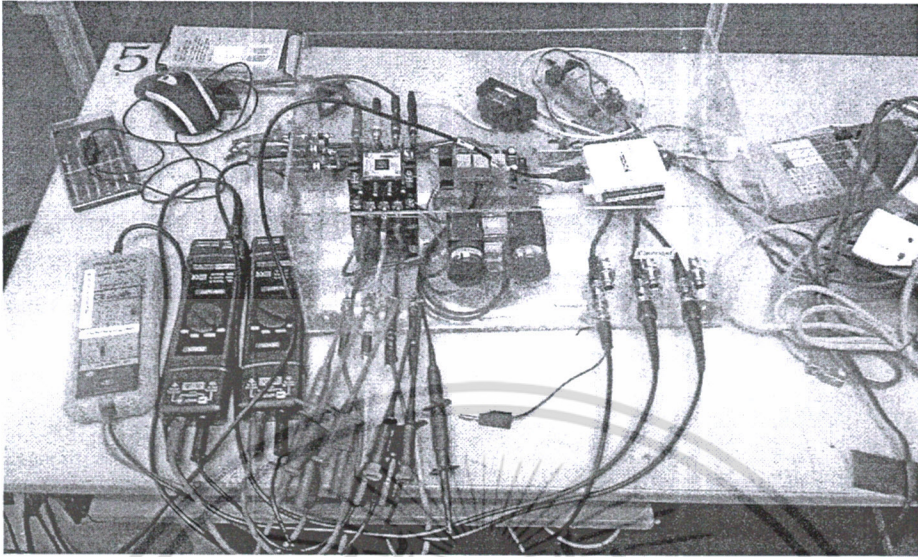


รูปที่ 4.18 วงจรการวัดและประมวลผล

4.3 ไตอะแกรมและวงจรของการทดลอง

เมื่อทำการจ่ายแรงดันไฟฟ้าให้กับตัวมอเตอร์ไฟฟ้า และทำการผ่านอุปกรณ์ตรวจวัดแรงดัน ซึ่งในการทดสอบนี้จะใช้ Differential Probe เป็นตัววัดสัญญาณแรงดัน สัญญาณที่ได้จากอุปกรณ์ตรวจวัดจะทำการส่งเข้าสู่บอร์ด DAQ โดยตรงเพื่อทำการประมวลผล และนำค่าแรงดันและกระแสที่รับมาได้แสดงบนหน้าจอ Visual Display บนโปรแกรม LabVIEW โดยค่าตัวแปรที่จะทำการวัดและประมวลผลคือ ค่าแรงดันรวมของทั้ง 3 เฟส ซึ่งเมื่อเกิดฟอลต์แบบรอบต่อรอบขึ้นที่เฟสใดเฟสหนึ่งที่จะเป็นอันตรายต่อตัวมอเตอร์ไฟฟ้า บอร์ด DAQ ก็จะทำการประมวลผลและแสดงค่าและการแจ้งเตือนต่าง ๆ ผ่านหน้าจอ Visual Display รูปที่ 4.17 และรูปที่ 4.18 แสดงไตอะแกรมการทดลองและวงจรการวัดและประมวลผล ทั้งนี้รูปจริงของอุปกรณ์ทรานสดิวเซอร์และชุดการทดสอบทั้งหมดแสดงในรูปที่ 4.19 และรูปที่ 4.20

ตามลำดับเอกสารที่ส่งจนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 อุปกรณ์ที่ใช้ในการทำงานของระบบ



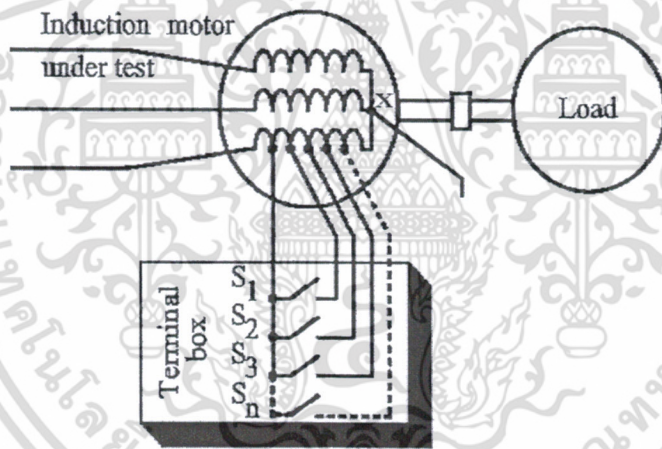
รูปที่ 4.20 ชุดทดสอบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการวิจัย

ในการจำลองฟอลต์สแตเตอร์ให้เกิดขึ้นนั้น ได้ทำการออกแบบการพันขดลวดมอเตอร์ใหม่โดยให้มีการลัดวงจรระหว่างรอบถึงรอบที่ 1% 2% 3% และ 4% ของจำนวนรอบมอเตอร์ทั้งหมด จากนั้นสังเกตการณ์เปลี่ยนแปลงของกราฟแรงดันรวม 3 เฟส ที่หน้าจอ Visual Display รวมถึงตัวมอเตอร์แล้วทำการบันทึกผลการทดลอง ที่หน้าจอ Visual Display จะแสดงค่าแรงดัน กระแส สถานะของการทำงาน สภาพของมอเตอร์ ระดับของการเกิดฟอลต์แบบรอบต่อรอบ ไฟแสดงการแจ้งเตือน และกราฟแรงดันรวมสามเฟส ซึ่งสามารถสังเกตสถานะทั้งหมดโดยผ่านหน้าจอนี้ได้ โดยรูปที่-5.1 ลักษณะของการลัดวงจรระหว่างรอบถึงรอบที่ใช้ในการทดลอง



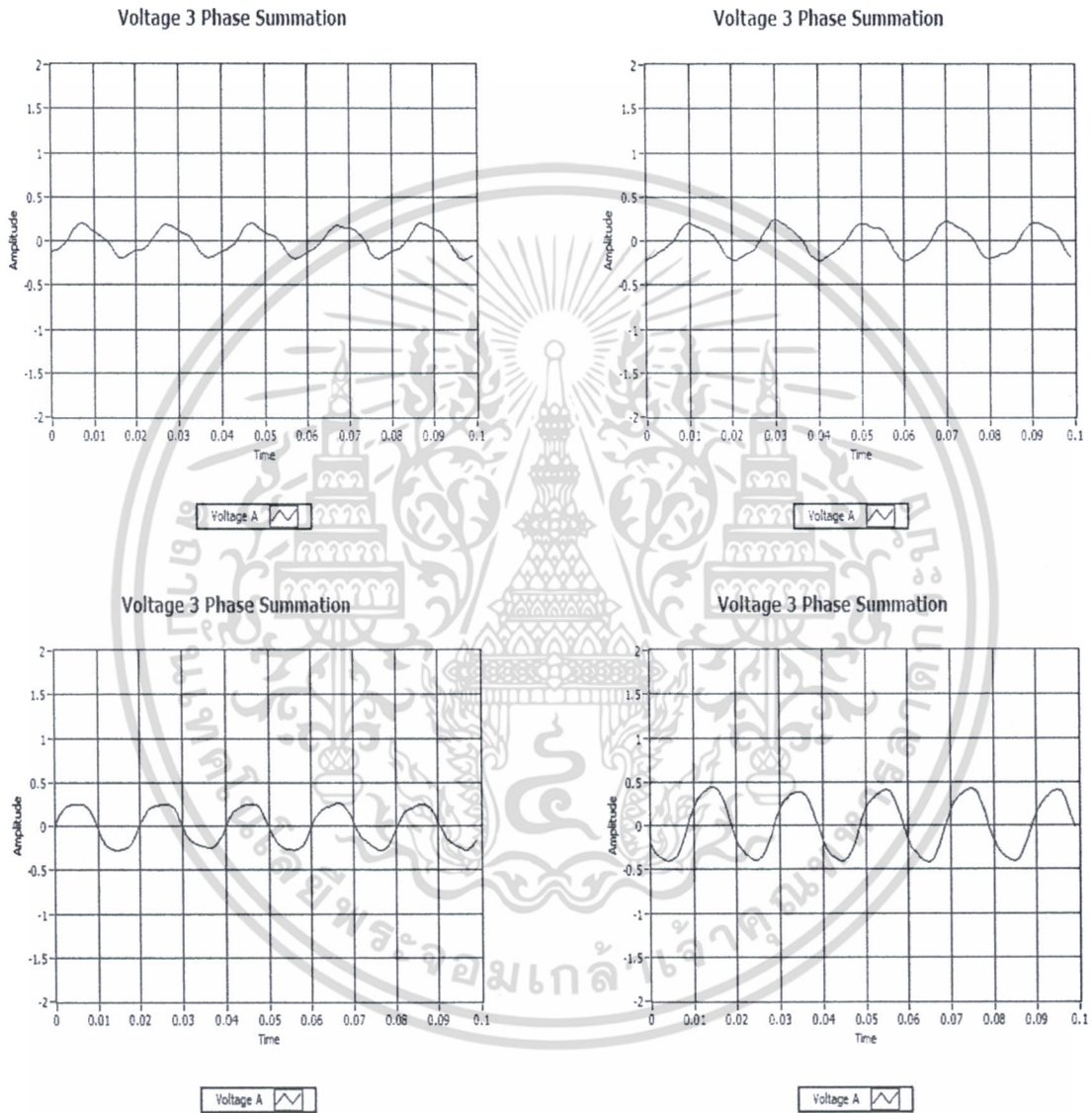
รูปที่ 5.1 ลักษณะของการลัดวงจรระหว่างรอบถึงรอบที่ใช้ในการทดลอง

5.1 การทดสอบที่สภาวะโหลดต่างกัน

ทำการทดลองที่สภาวะไร้อะโหลดแล้วสังเกตกราฟแรงดันรวม 3 เฟส จากนั้นทำการบันทึกผลการทดสอบ แสดงผลการทดลองดังกล่าวในรูปที่ 5.2 ถึงรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

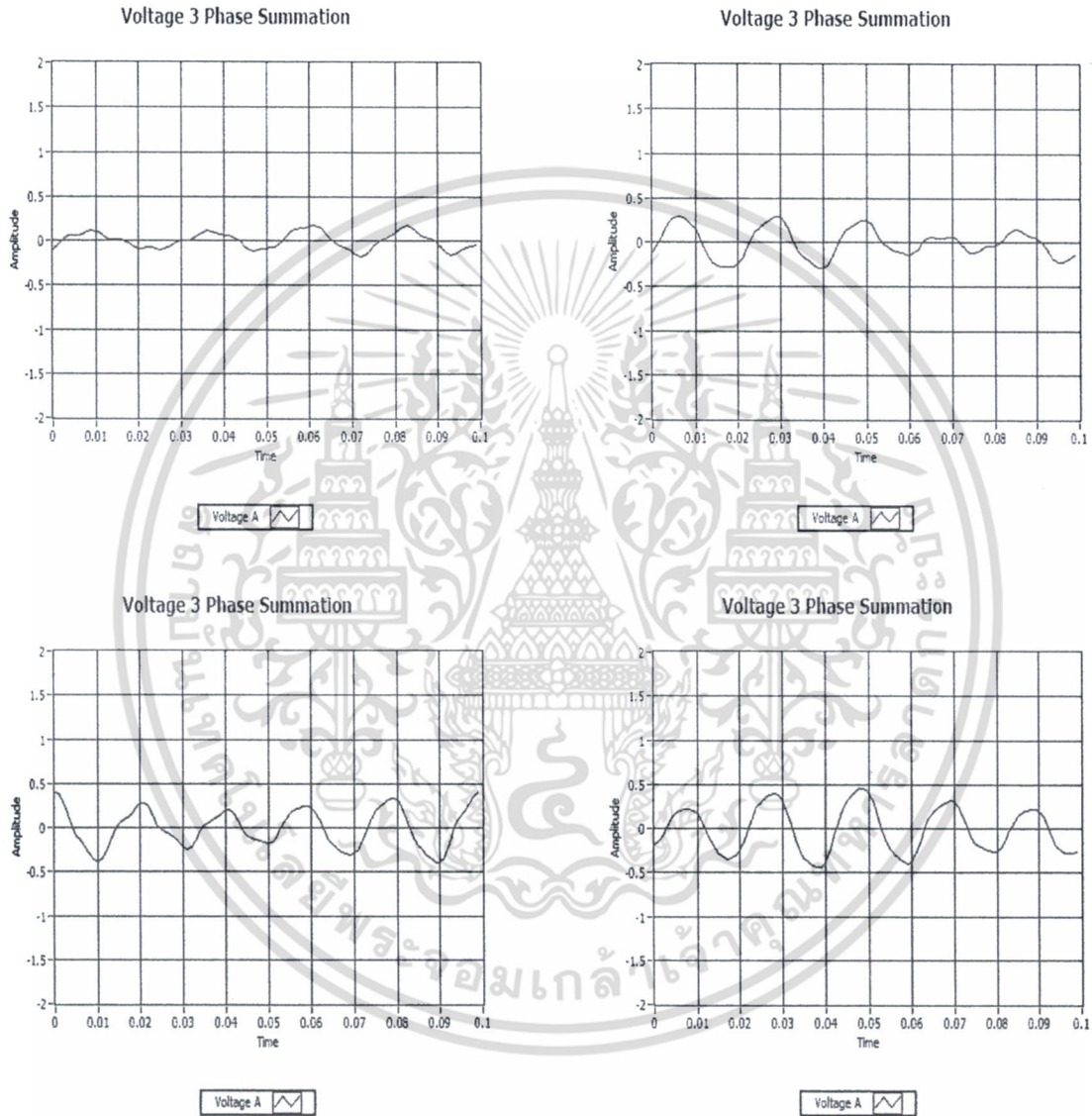
5.1.1 ผลการทดสอบสัญญาณองค์ประกอบแรงดันลำดับศูนย์ที่สถานะความรุนแรงของฟอลต์สเตเตอร์ต่างกันเมื่อทดสอบที่สถานะไร้โหลด



รูปที่ 5.2 สัญญาณองค์ประกอบแรงดันลำดับศูนย์กรณีทดสอบที่สถานะไร้โหลดด้วยขดลวดสเตเตอร์ (บนซ้าย) ถัดรอบ 1% (บนขวา) ถัดรอบ 2% (ล่างซ้าย) ถัดรอบ 3% และ (ล่างขวา) ถัดรอบ 4%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

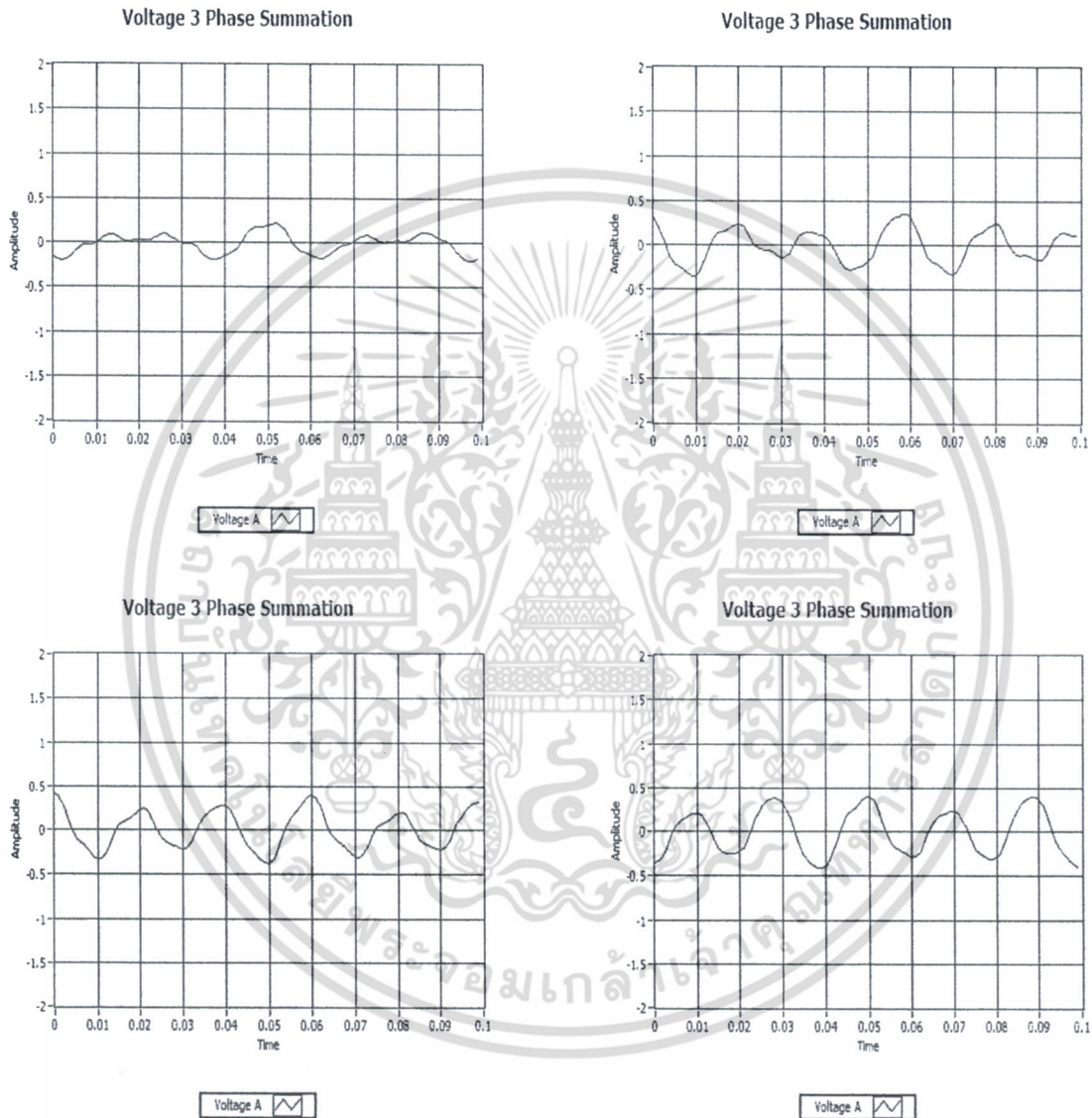
5.1.2 ผลการทดสอบสัญญาณองค์ประกอบแรงดันลำดับศูนย์ที่สภาวะความรุนแรงของฟอลต์ สเตเตอร์ต่างกันเมื่อทดสอบที่สภาวะโหลด 50%



รูปที่ 5.3 สัญญาณองค์ประกอบแรงดันลำดับศูนย์กรณีทดสอบที่สภาวะ โหลด 50% ด้วยขดลวดสเตเตอร์
(บนซ้าย) ลัดรอบ 1% (บนขวา) ลัดรอบ 2% (ล่างซ้าย) ลัดรอบ 3% และ (ล่างขวา) ลัดรอบ 4%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

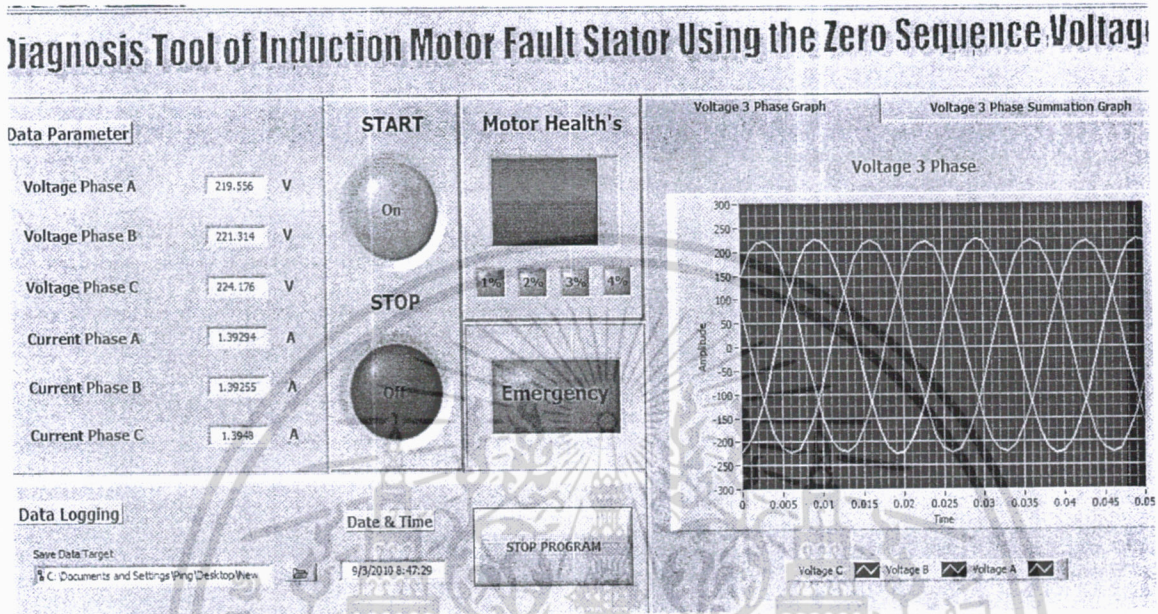
5.1.3 ผลการทดสอบสัญญาณองค์ประกอบแรงดันลำดับศูนย์ที่สภาวะความรุนแรงของฟอลต์ สแตเตอร์ต่างกันเมื่อทดสอบที่สภาวะพิกัดโหลด



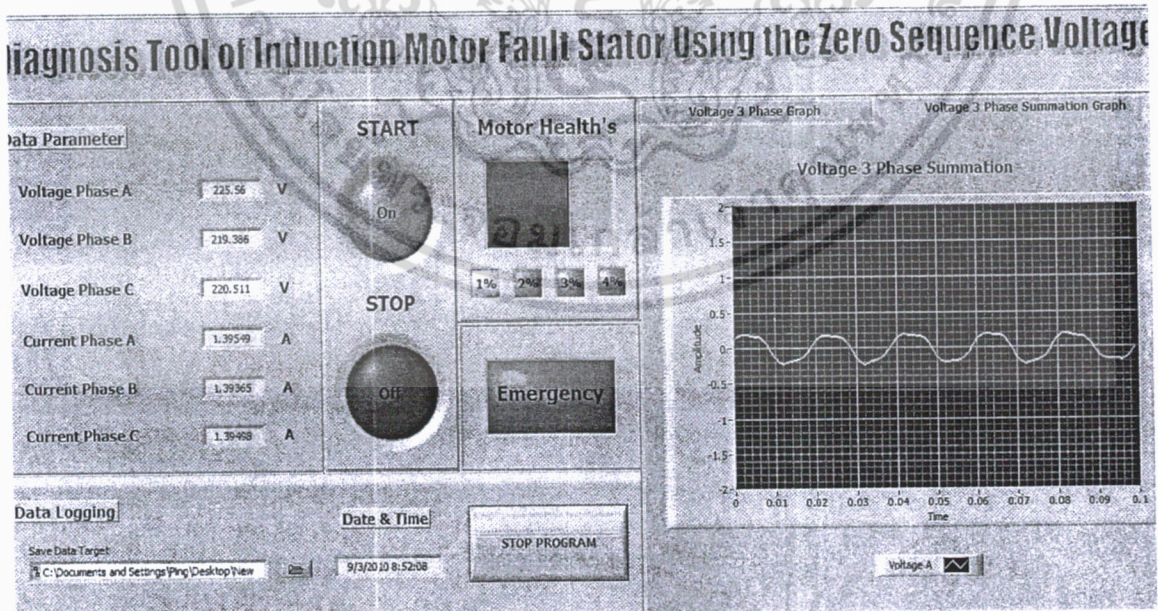
รูปที่ 5.4 สัญญาณองค์ประกอบแรงดันลำดับศูนย์กรณีทดสอบที่สภาวะโหลด 100% ด้วยขนาดลวดสแตเตอร์ (บนซ้าย) ลัดรอบ 1% (บนขวา) ลัดรอบ 2% (ล่างซ้าย) ลัดรอบ 3% และ (ล่างขวา) ลัดรอบ 4%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การแสดงผลการเฝ้าตรวจสภาวะมอเตอร์



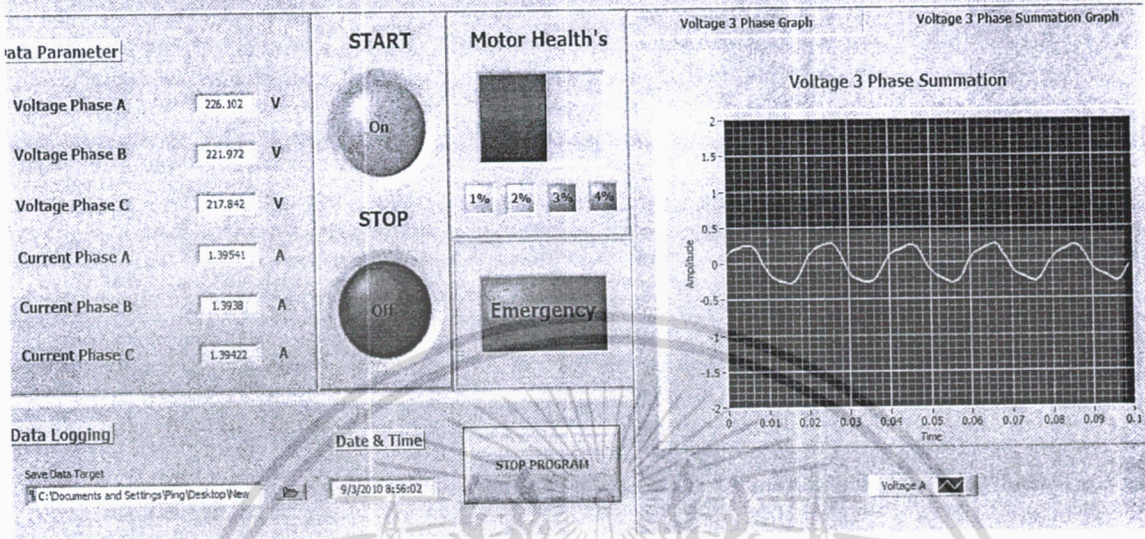
รูปที่ 5.5 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์สุขภาพดี



รูปที่ 5.6 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์มีฟอลต์สเตรเตอร์ 1%

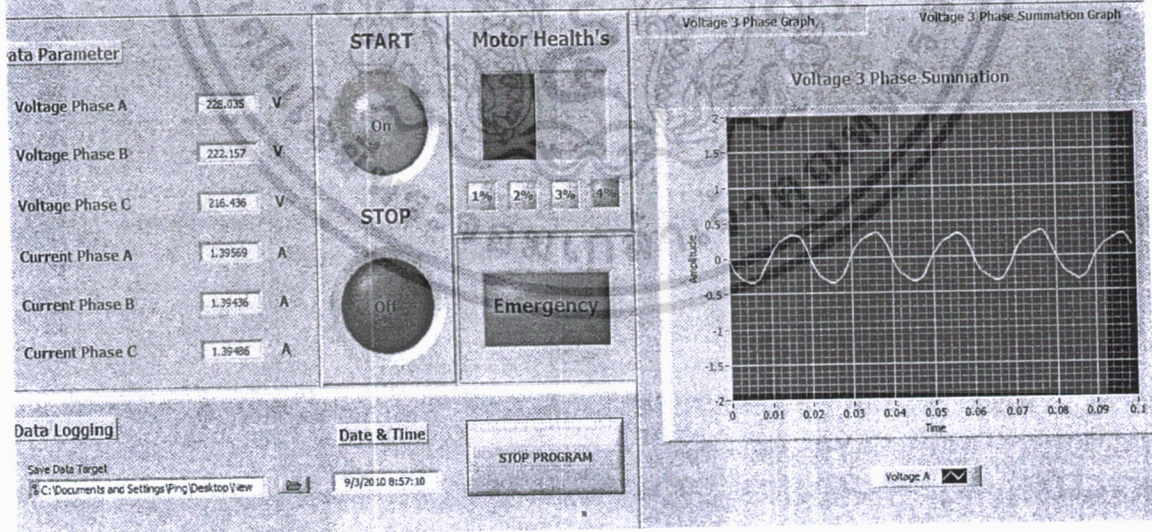
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Diagnosis Tool of Induction Motor Fault Stator Using the Zero Sequence Voltage



รูปที่ 5.7 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์มีฟอลต์สเตรเตอร์ 2%

Diagnosis Tool of Induction Motor Fault Stator Using the Zero Sequence Voltage



รูปที่ 5.8 หน้าจอ Visual Display ขณะทำการทดสอบกรณีมอเตอร์มีฟอลต์สเตรเตอร์ 3%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุป

การพัฒนาซอฟต์แวร์ที่ได้ออกแบบขึ้นนี้สามารถนำมาใช้ประโยชน์ในการเฝ้าตรวจสอบสถานะมอเตอร์ได้เป็นอย่างดี ซึ่งในงานวิจัยนี้ได้มุ่งเน้นไปที่ความล้มเหลวของตัวนำสเตเตอร์กรณีเกิดการลัดวงจรระหว่างรอบถึงรอบ ซึ่งเป็นสาเหตุเริ่มที่ทำให้เกิดฟอลต์ที่มีความรุนแรงตามมา ซึ่งจากการนำมาทดลองเฝ้าตรวจสอบสถานะมอเตอร์เหนี่ยวนำ 3 เฟส สามารถทำการแยกแยะและระดับความรุนแรงของฟอลต์สเตเตอร์ที่ระดับความรุนแรงต่าง ๆ ได้อย่างถูกต้อง ทั้งนี้หากทำการตรวจพบสถานะความรุนแรงของฟอลต์สเตเตอร์ถึงระดับความรุนแรงที่ได้กำหนดไว้ก็จะกระพริบเตือนอีกด้วย

จากที่กล่าวมาทั้งหมดข้างต้นสามารถสรุปได้ว่าซอฟต์แวร์สำหรับการวินิจฉัยมอเตอร์เหนี่ยวนำที่ได้พัฒนาขึ้นนี้มีประสิทธิภาพในการนำมาใช้งานได้อย่างน่าพอใจ ภายได้ต้นทุนที่มีราคาต่ำซึ่งสามารถทำให้ลดการซื้อมาใช้งานที่มีราคาแพงมากได้เป็นอย่างดี โดยเฉพาะสำหรับการต่อยอดเพื่อใช้ในงานอุตสาหกรรมเชิงพาณิชย์ ทั้งนี้แนวทางในการพัฒนาต่อไปสามารถใช้ประโยชน์จากเทคโนโลยีสารสนเทศเช่น เครื่องข่ายอินเทอร์เน็ต สำหรับการส่งผ่านข้อมูลต่าง ๆ สำหรับการแจ้งต่อผู้รับผิดชอบได้ทราบต่อไป

บรรณานุกรม

- [1] S. Nandi, H. A. Toliyat and X. Li, "Condition monitoring and fault diagnosis of electrical motors- a review," IEEE Trans. Energy Conversion, Vol. 20, No. 4, pp.719-729, Dec. 2005.
- [2] M. E. H. Benbouzid and G. B. Kliman, "What stator current processing based technique to use for induction motor rotor faults diagnosis," IEEE Trans. Energy Conversion, Vol. 18, No. 2, pp. 238-244, June 2003.
- [3] A. H. Bonnett and G. C. Soukup, "Cause and analysis of stator and rotor failures in three-phase squirrelcage induction motors," IEEE Trans. Industry Application, Vol. 28, pp. 921-937, July/Aug. 1992.
- [4] P. Pillay, Z. Xu, "Labview implementation of speed detection for mains-fed motors using motor current signature analysis," IEEE Power Engineering Review, pp. 47-48, June 1998.
- [5] P. Vas, "Parameter estimation, condition monitoring and diagnosis of electrical machines," Oxford, Clarendon Press, pp.145-156, 1993.
- [6] J. Q. Yang, J. Huang, and T. Liu, "Diagnosis of stator faults in induction motor based on zero sequence voltage after switch-off," Springer, Vol. 9, No. 2, pp. 165-172, Feb. 2008.
- [7] W. Cao and K. J. Bradley, "Assessing the impacts of rewind and repeated rewinds on induction motors: Is an opportunity for redesigning the machine being wasted?," IEEE Trans. Industry Application, Vol. 42, No. 4, pp. 958-964, July/Aug. 2006.
- [8] W. Cao, K. J. Bradley and J. Allen, "Evaluation of additional loss in induction motors consequent on repair and rewinding," IEE Proc. Electronics, Power Application, Vol. 153, No. 1, pp. 1-6, Jan. 2006.
- [9] G. M. Joksimovic and J. Penman, "The detection of inter-turn short circuits in the stator windings of operating motors," IEEE Trans. on Industrial Elect., Vol. 47, No. 5, Oct. 2000, pp. 1078-1084.
- [10] H. W. Penrose and J. Jette, "Static motor circuit analysis: an introduction to theory and application," IEEE Electrical Insulation Magazine, Vol. 16, No. 4, pp. 6-10, July/Aug. 2000.
- [11] P. Garcia, F. Briz, M. W. Degner and A. B. Diez, "Diagnostics of induction machines using the zero sequence voltage," IEEE-IAS Annu. Meeting, pp.735-742, Oct. 2004.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [12] M. A. Cash, T. G. Habetler and G. B. Kliman, "Insulation failure prediction in AC machines using line-neutral voltages," IEEE Trans. Industry Applications, Vol. 34, No. 6, pp.1234-1239, Nov./Dec. 1998.
- [13] F. Briz, M. W. Degner, P. Garcia, and A. B. Dicz, "Induction machine diagnostics using zero sequence component," IEEE-IAS Annu. Meeting, pp. 34-41, Oct. 2006.
- [14] เอกสารประกอบการสอน, "LabVIEW ขั้นต้น," มหาวิทยาลัยเทคโนโลยีสุรนารี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารข้อมูลของ Data Acquisition Board, NI USB-6009



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

USER GUIDE AND SPECIFICATIONS

NI USB-6008/6009

Français Deutsch 日本語 한국어 简体中文
ni.com/manuals

This user guide describes how to use the National Instruments USB-6008/6009 data acquisition (DAQ) devices and lists specifications.

Introduction

The NI USB-6008/6009 provides connection to eight analog input (AI) channels, two analog output (AO) channels, 12 digital input/output (DIO) channels, and a 32-bit counter with a Full-Speed USB interface.



Note This manual revision updates naming conventions to reflect the conventions used in NI-DAQmx. Table 1 notes the correlation between the old and updated names.

Table 1. Digital Output Driver Type Naming Conventions

Hardware Functionality	NI-DAQmx Terminology
Open-drain	Open collector
Push-pull	Active drive



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Differences Between the NI USB-6008 and NI USB-6009

Feature	NI USB-6008	NI USB-6009
AI Resolution	12 bits differential, 11 bits single-ended	14 bits differential, 13 bits single-ended
Maximum AI Sample Rate, Single Channel*	10 kS/s	48 kS/s
Maximum AI Sample Rate, Multiple Channels (Aggregate)*	10 kS/s	48 kS/s
DIO Configuration	Open collector	Open collector or active drive
* System dependent.		

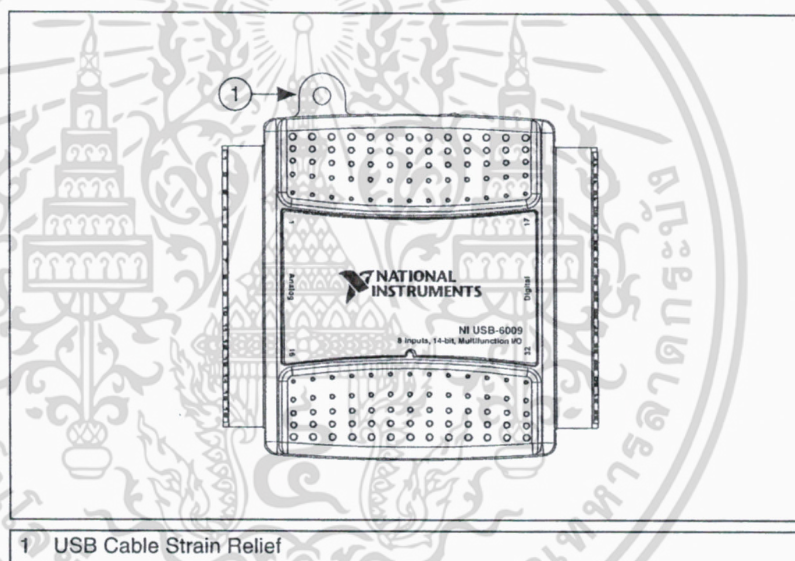


Figure 1. NI USB-6008/6009 Top View

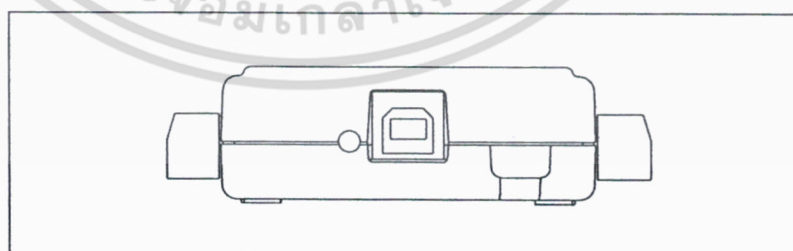


Figure 2. NI USB-6008/6009 Back View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions

Figure 3 illustrates the dimensions of the NI USB-6008/6009 device.

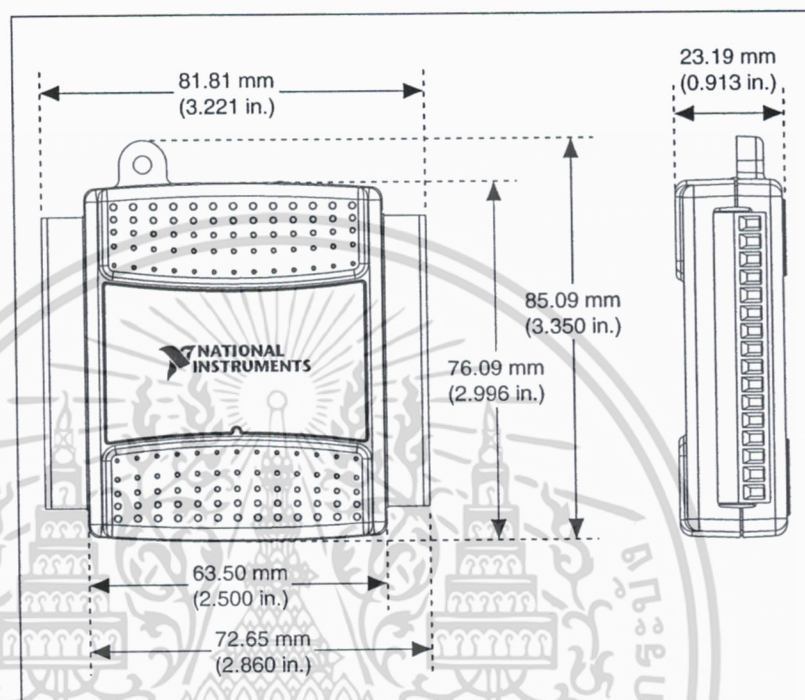


Figure 3. NI USB-6008/6009 in Millimeters (Inches)

Safety Guidelines



Caution Operate the NI USB-6008/6009 only as described in these operating instructions.

The following section contains important safety information that you must follow when installing and using the NI USB-6008/6009.



Caution Do not operate the NI USB-6008/6009 in a manner not specified in this document. Misuse of the device can result in a hazard. You can compromise the safety protection built into the device if the device is damaged in any way. If the device is damaged, contact National Instruments for repair.



Caution Do not substitute parts or modify the device except as described in this document. Use the device only with the chassis, modules, accessories, and cables specified in the installation instructions. You must have all covers and filler panels installed during operation of the device.



Caution Do not operate the device in an explosive atmosphere or where there may be flammable gases or fumes. If you must operate the device in such an environment, it must be in a suitably rated enclosure.

If you need to clean the device, use a dry cloth. Make sure that the device is completely dry and free from contaminants before returning it to service.

Operate the device only at or below Pollution Degree 2. Pollution is foreign matter in a solid, liquid, or gaseous state that can reduce dielectric strength or surface resistivity. The following is a description of pollution degrees:

- Pollution Degree 1 means no pollution or only dry, nonconductive pollution occurs. The pollution has no influence.
- Pollution Degree 2 means that only nonconductive pollution occurs in most cases. Occasionally, however, a temporary conductivity caused by condensation must be expected.
- Pollution Degree 3 means that conductive pollution occurs, or dry, nonconductive pollution occurs that becomes conductive due to condensation.

You must insulate signal connections for the maximum voltage for which the device is rated. Do not exceed the maximum ratings for the device. Do not install wiring while the device is live with electrical signals. Do not remove or add connector blocks when power is connected to the system. Avoid contact between your body and the connector block signal when hot swapping modules. Remove power from signal lines before connecting them to or disconnecting them from the device.

Operate the device at or below the Measurement Category I¹. Measurement circuits are subjected to working voltages² and transient stresses (overvoltage) from the circuit to which they are connected during measurement or test. Measurement categories establish standard impulse withstand voltage levels that commonly occur in electrical distribution systems. The following is a description of measurement categories:

- Measurement Category I is for measurements performed on circuits not directly connected to the electrical distribution system referred to as MAINS³ voltage. This category is for measurements of voltages from specially protected secondary circuits. Such voltage measurements include signal levels, special equipment, limited-energy parts of equipment, circuits powered by regulated low-voltage sources, and electronics.

¹ Measurement Category as defined in electrical safety standard IEC 61010-1. Measurement Category is also referred to as Installation Category.

² Working Voltage is the highest rms value of an AC or DC voltage that can occur across any particular insulation.

³ MAINS is defined as a hazardous live electrical supply system that powers equipment. Suitably rated measuring circuits may be connected to the MAINS for measuring purposes.

- Measurement Category II is for measurements performed on circuits directly connected to the electrical distribution system. This category refers to local-level electrical distribution, such as that provided by a standard wall outlet (for example, 115 V for U.S. or 230 V for Europe). Examples of Measurement Category II are measurements performed on household appliances, portable tools, and similar E Series devices.
- Measurement Category III is for measurements performed in the building installation at the distribution level. This category refers to measurements on hard-wired equipment such as equipment in fixed installations, distribution boards, and circuit breakers. Other examples are wiring, including cables, bus-bars, junction boxes, switches, socket-outlets in the fixed installation, and stationary motors with permanent connections to fixed installations.
- Measurement Category IV is for measurements performed at the primary electrical supply installation (<1,000 V). Examples include electricity meters and measurements on primary overcurrent protection devices and on ripple control units.

Related Documentation

Each application software package and driver includes information about writing applications for taking measurements and controlling measurement devices. The following references to documents assume you have NI-DAQmx 8.7 or later, and where applicable, version 7.1 or later of the NI application software.

NI-DAQmx for Windows

The *DAQ Getting Started Guide* describes how to install your NI-DAQmx for Windows software, how to install your NI-DAQmx-supported DAQ device, and how to confirm that your device is operating properly. Select **Start»All Programs»National Instruments»NI-DAQ»DAQ Getting Started Guide**.

The *NI-DAQ Readme* lists which devices are supported by this version of NI-DAQ. Select **Start»All Programs»National Instruments»NI-DAQ»NI-DAQ Readme**.

The *NI-DAQmx Help* contains general information about measurement concepts, key NI-DAQmx concepts, and common applications that are applicable to all programming environments. Select **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx Help**.



Note For information about non-Windows operating system support, refer to ni.com/info and enter BaseGSGML.

LabVIEW

If you are a new user, use the *Getting Started with LabVIEW* manual to familiarize yourself with the LabVIEW graphical programming environment and the basic LabVIEW features you use to build data acquisition and instrument control applications. Open the *Getting Started with LabVIEW* manual by selecting **Start»All Programs»National Instruments»LabVIEW»LabVIEW Manuals** or by navigating to the `labview\manuals` directory and opening `LV_Getting_Started.pdf`.

Use the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help** in LabVIEW, to access information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. Refer to the following locations on the **Contents** tab of the *LabVIEW Help* for information about NI-DAQmx:

- **Getting Started»Getting Started with DAQ**—Includes overview information and a tutorial to learn how to take an NI-DAQmx measurement in LabVIEW using the DAQ Assistant.
- **VI and Function Reference»Measurement I/O VIs and Functions**—Describes the LabVIEW NI-DAQmx VIs and properties.
- **Taking Measurements**—Contains the conceptual and how-to information you need to acquire and analyze measurement data in LabVIEW, including common measurements, measurement fundamentals, NI-DAQmx key concepts, and device considerations.

LabWindows/CVI

The **Data Acquisition** book of the *LabWindows/CVI Help* contains measurement concepts for NI-DAQmx. This book also contains *Taking an NI-DAQmx Measurement in LabWindows/CVI*, which includes step-by-step instructions about creating a measurement task using the DAQ Assistant. In LabWindows™/CVI™, select **Help»Contents**, then select **Using LabWindows/CVI»Data Acquisition**.

The **NI-DAQmx Library** book of the *LabWindows/CVI Help* contains API overviews and function reference for NI-DAQmx. Select **Library Reference»NI-DAQmx Library** in the *LabWindows/CVI Help*.

Measurement Studio

If you program your NI-DAQmx-supported device in Measurement Studio using Visual C++, Visual C#, or Visual Basic .NET, you can interactively create channels and tasks by launching the DAQ Assistant from MAX or from within Visual Studio .NET. You can generate the configuration code based on your task or channel in Measurement Studio. Refer to the *DAQ Assistant Help* for additional information about generating code. You also

can create channels and tasks, and write your own applications in your ADE using the NI-DAQmx API.

For help with NI-DAQmx methods and properties, refer to the NI-DAQmx .NET Class Library or the NI-DAQmx Visual C++ Class Library included in the *NI Measurement Studio Help*. For general help with programming in Measurement Studio, refer to the *NI Measurement Studio Help*, which is integrated with the Microsoft Visual Studio .NET help. To view this help file in Visual Studio .NET, select **Measurement Studio»NI Measurement Studio Help**.

To create an application in Visual C++, Visual C#, or Visual Basic .NET, follow these general steps:

1. In Visual Studio .NET, select **File»New»Project** to launch the New Project dialog box.
2. Find the Measurement Studio folder for the language you want to create a program in.
3. Choose a project type. You add DAQ tasks as a part of this step.

ANSI C without NI Application Software

The *NI-DAQmx Help* contains API overviews and general information about measurement concepts. Select **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx Help**.

The *NI-DAQmx C Reference Help* describes the NI-DAQmx Library functions, which you can use with National Instruments data acquisition devices to develop instrumentation, acquisition, and control applications. Select **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx C Reference Help**.

.NET Languages without NI Application Software

With the Microsoft .NET Framework version 1.1 or later, you can use NI-DAQmx to create applications using Visual C# and Visual Basic .NET without Measurement Studio. You need Microsoft Visual Studio .NET 2003 or Microsoft Visual Studio 2005 for the API documentation to be installed.

The installed documentation contains the NI-DAQmx API overview, measurement tasks and concepts, and function reference. This help is integrated into the Visual Studio .NET documentation. To view the NI-DAQmx .NET documentation, go to **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx .NET Reference Help**. Expand **NI Measurement Studio Help»NI Measurement Studio .NET Class Library»Reference** to view the function reference. Expand **NI Measurement Studio Help»NI Measurement Studio .NET Class**

Library»Using the Measurement Studio .NET Class Libraries to view conceptual topics for using NI-DAQmx with Visual C# and Visual Basic .NET.

To get to the same help topics from within Visual Studio, go to **Help»Contents**. Select **Measurement Studio** from the **Filtered By** drop-down list and follow the previous instructions.

Device Documentation and Specifications

Documentation for supported devices and accessories, including PDF and help files describing device terminals, specifications, features, and operation are on the NI-DAQmx CD that includes Device Documentation. Insert the CD, open the Device Documentation directory, and double-click the Device Documents shortcut for your language to find, view, and print device documents.



Note You can also download these documents at ni.com/manuals.

NI-DAQmx Base (Linux/Mac OS X/LabVIEW PDA 8.x)

The *NI-DAQmx Base Getting Started Guide* describes how to install your NI-DAQmx Base software, your NI-DAQmx Base-supported DAQ device, and how to confirm that your device is operating properly. In Windows, select **Start»All Programs»National Instruments»NI-DAQmx Base»Documentation»NI-DAQmx Base Getting Started Guide**.

Getting Started with NI-DAQmx Base for Linux and Mac Users describes how to install your NI-DAQmx Base software, your NI-DAQmx Base-supported DAQ device, and how to confirm that your device is operating properly on your Mac/Linux machine.

The *NI-DAQmx Base Readme* lists which devices are supported by this version of NI-DAQmx Base. In Windows, select **Start»All Programs»National Instruments»NI-DAQmx Base»DAQmx Base Readme**.

The *NI-DAQmx Base VI Reference Help* contains VI reference and general information about measurement concepts. In LabVIEW, select **Help»NI-DAQmx Base VI Reference Help**.

The *NI-DAQmx Base C Reference Help* contains C reference and general information about measurement concepts. In Windows, select **Start»All Programs»National Instruments»NI-DAQmx Base»Documentation»C Function Reference Help**.



Note All NI-DAQmx Base documentation for Linux is installed at `/usr/local/natinst/nidaqmxbase/documentation`.



Note All NI-DAQmx Base documentation for Mac OS X is installed at
`/Applications/National Instruments/NI-DAQmx Base/documentation`.

Training Courses

If you need more help getting started developing an application with NI products, NI offers training courses. To enroll in a course or obtain a detailed course outline, refer to ni.com/training.

Technical Support on the Web

For additional support, refer to ni.com/support or zone.ni.com.

Installing the Software

Software support for the NI USB-6008/6009 for Windows Vista/XP/2000 is provided by NI-DAQmx. The *DAQ Getting Started Guide*, which you can download at ni.com/manuals, offers NI-DAQmx users step-by-step instructions for installing software and hardware, configuring channels and tasks, and getting started developing an application.



Note For information about non-Windows operating system support, refer to ni.com/info and enter BaseGSGML.

Installing Other Software

If you are using other software, refer to the installation instructions that accompany your software.

Example Programs

The NI-DAQmx CD contains example programs that you can use to get started programming with the NI USB-6008/6009. Refer to the *NI-DAQmx for USB Devices Getting Started Guide* that shipped with your device, and is also accessible from **Start»All Programs»National Instruments»NI-DAQ**, for more information.

The NI-DAQmx Base software ships with example programs you can use to get started programming with the NI USB-6008/6009. Refer to the *NI-DAQmx Base Getting Started Guide* that shipped with your device, and is also accessible from **Start»All Programs»National Instruments»NI-DAQmx Base»Examples**, for more information.



Note For information about non-Windows operating system support, refer to ni.com/info and enter BaseGSGML.

Installing the NI USB-6008/6009 Device

Before installing the device, you must install the software you plan to use with the device. Refer to the *Installing the Software* section of this guide and the documentation included with the software for more information.

Figure 4 shows key functional components of the NI USB-6008/6009.

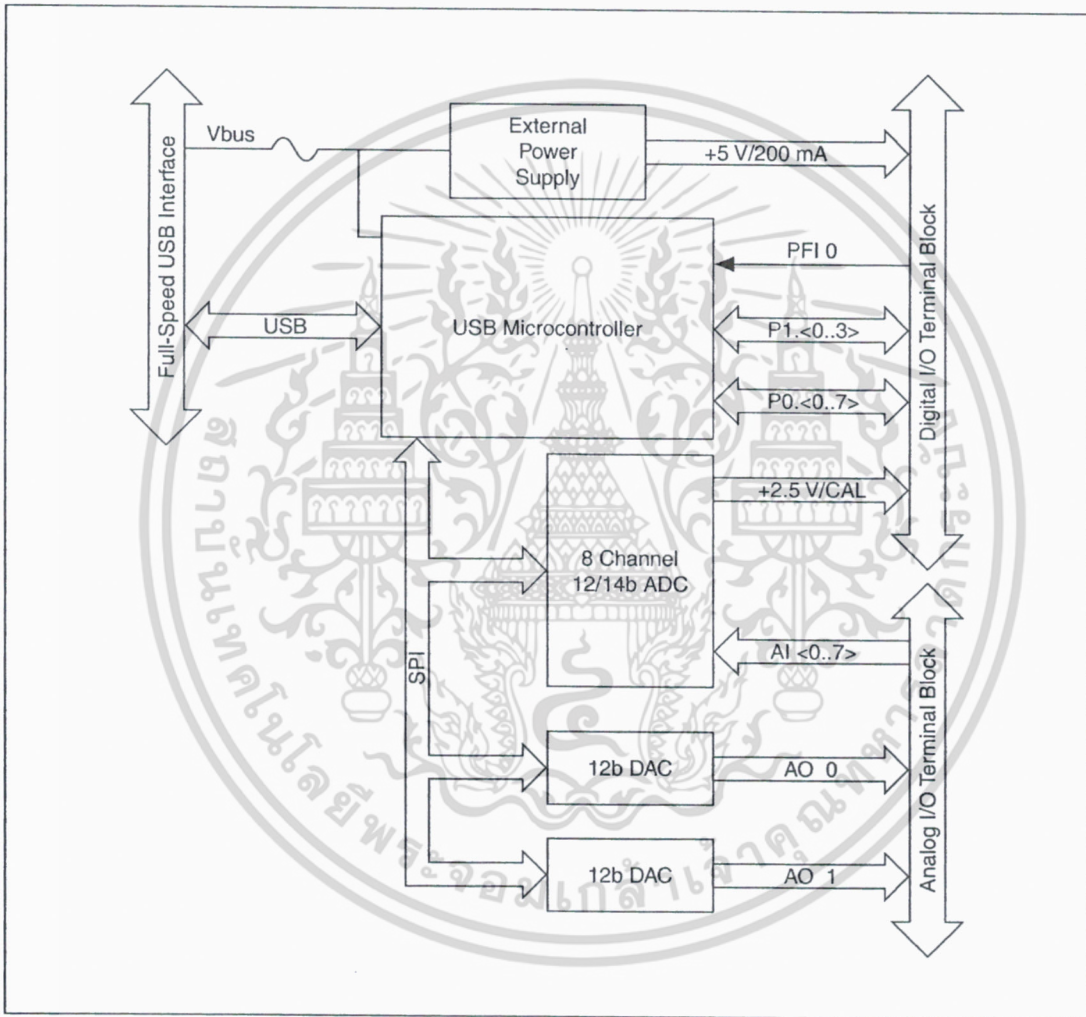


Figure 4. Device Block Diagram

Setting Up the NI USB-6008/6009 Device

Complete the following steps to set up the NI USB-6008/6009:

1. Install combicon screw terminal blocks by inserting them into the combicon jacks.

- Figure 5 illustrates the signal labels that ship in the NI USB-6008/6009 kit. You can apply the signal labels to the screw terminal blocks for easy signal identification.

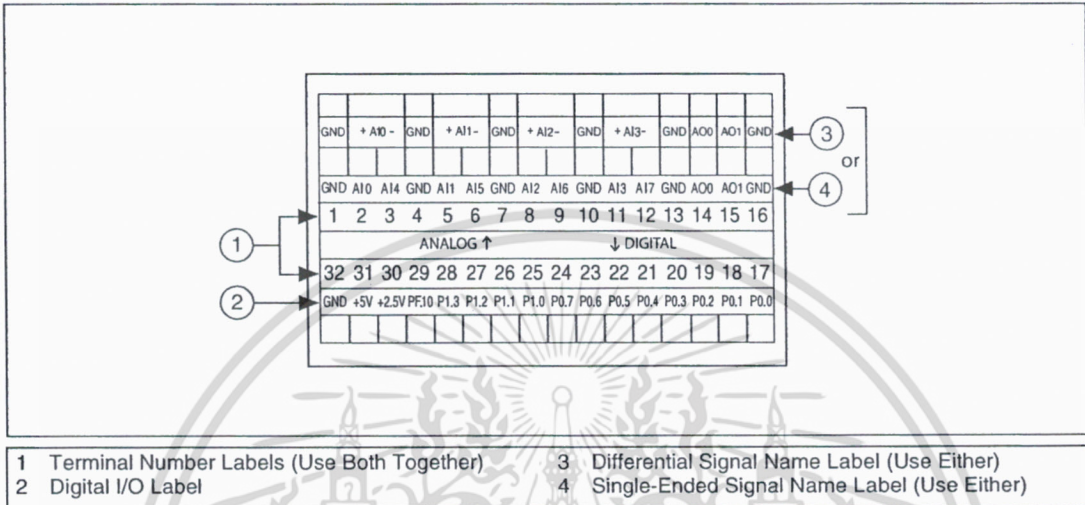


Figure 5. NI USB-6008/6009 Signal Labels

- Refer to Table 4 and Figures 5 and 6 for signal label orientation and affix the provided signal labels to the screw terminal blocks. Until the signal labels are applied, you can insert the screw terminal blocks into either of the combicon jacks.

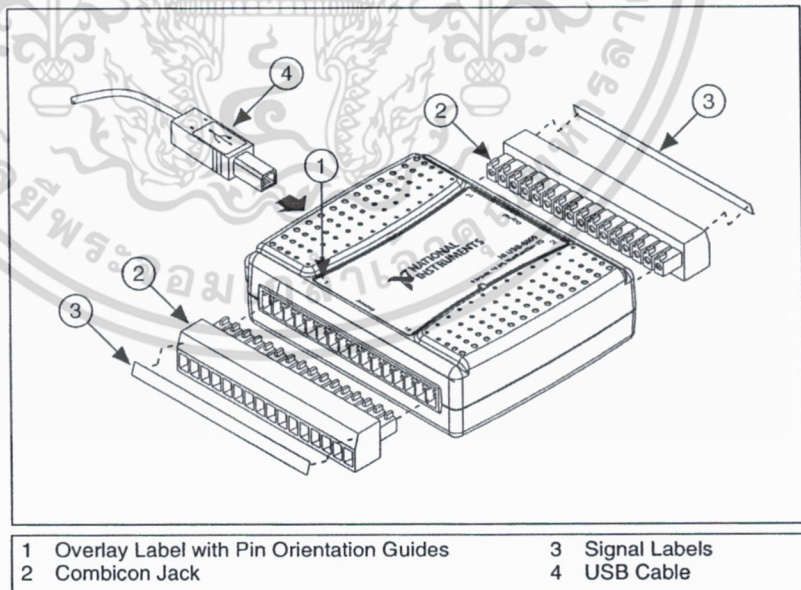


Figure 6. Signal Label Application Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Note Once you label the screw terminal blocks, you must only insert them into the matching combicon jack, as indicated by the overlay label on the NI USB-6008/6009 device.

4. Connect the wiring to the appropriate screw terminals.

Connecting the NI USB-6008/6009 to a Computer

Plug one end of the USB cable into the NI USB-6008/6009 and the other end into an available USB port on the computer.

LED Indicator

The NI USB-6008/6009 device has a green LED next to the USB connector. The LED indicator indicates device status, as listed in Table 3. When the device is connected to a USB port, the LED blinks steadily to indicate that the device is initialized and is receiving power from the connection.

If the LED is not blinking, it may mean that the device is not initialized or the computer is in standby mode. In order for the device to be recognized, the device must be connected to a computer that has NI-DAQmx installed on it. If your device is not blinking, make sure your computer has the latest version of NI-DAQmx installed on it, and the computer is not in standby mode.

Table 3. LED State/Device Status

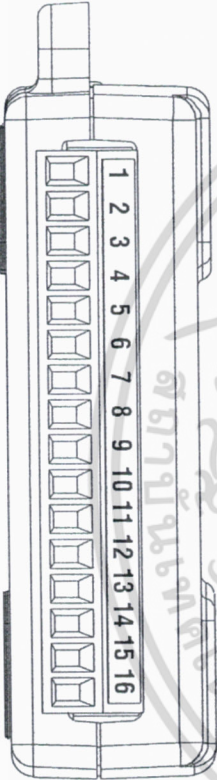
LED State	Device Status
Not lit	Device not connected or in suspend.
On, not blinking	Device connected.
Single-blink	Operating normally.

I/O Connector

The NI USB-6008/6009 ships with one detachable screw terminal block for analog signals and one detachable screw terminal block for digital signals. These terminal blocks provide 16 connections that use 16 AWG to 28 AWG wire.

Table 4 lists the analog terminal assignments, and Table 5 lists the digital terminal assignments.

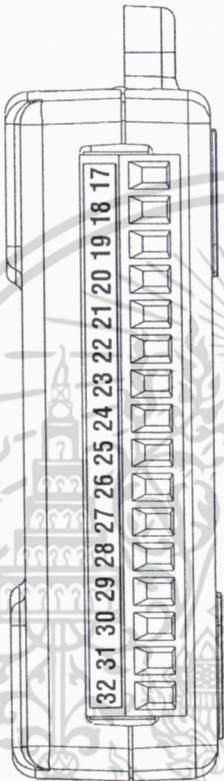
Table 4. Analog Terminal Assignments



Module	Terminal	Signal, Single-Ended Mode	Signal, Differential Mode
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0-
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1-
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2-
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3-
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 5. Digital Terminal Assignments

Module	Terminal	Signal
	17	P0.0
	18	P0.1
	19	P0.2
	20	P0.3
	21	P0.4
	22	P0.5
	23	P0.6
	24	P0.7
	25	P1.0
	26	P1.1
	27	P1.2
	28	P1.3
	29	PFI 0
	30	+2.5 V
	31	+5 V
	32	GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal Descriptions

Table 6 describes the signals available on the I/O connectors.

Table 6. Signal Descriptions

Signal Name	Reference	Direction	Description
GND	—	—	Ground —The reference point for the single-ended AI measurements, bias current return point for differential mode measurements, AO voltages, digital signals at the I/O connector, +5 VDC supply, and the +2.5 VDC reference.
AI <0..7>	Varies	Input	Analog Input Channels 0 to 7 —For single-ended measurements, each signal is an analog input voltage channel. For differential measurements, AI 0 and AI 4 are the positive and negative inputs of differential analog input channel 0. The following signal pairs also form differential input channels: <AI 1, AI 5>, <AI 2, AI 6>, and <AI 3, AI 7>.
AO 0	GND	Output	Analog Channel 0 Output —Supplies the voltage output of AO channel 0.
AO 1	GND	Output	Analog Channel 1 Output —Supplies the voltage output of AO channel 1.
P1.<0..3> P0.<0..7>	GND	Input or Output	Digital I/O Signals —You can individually configure each signal as an input or output.
+2.5 V	GND	Output	+2.5 V External Reference —Provides a reference for wrap-back testing.
+5 V	GND	Output	+5 V Power Source —Provides +5 V power up to 200 mA.
PFI 0	GND	Input	PFI 0 —This pin is configurable as either a digital trigger or an event counter input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Analog Input

You can connect analog input signals to the NI USB-6008/6009 through the I/O connector. Refer to Table 6 for more information about connecting analog input signals.

Analog Input Circuitry

Figure 7 illustrates the analog input circuitry of the NI USB-6008/6009.

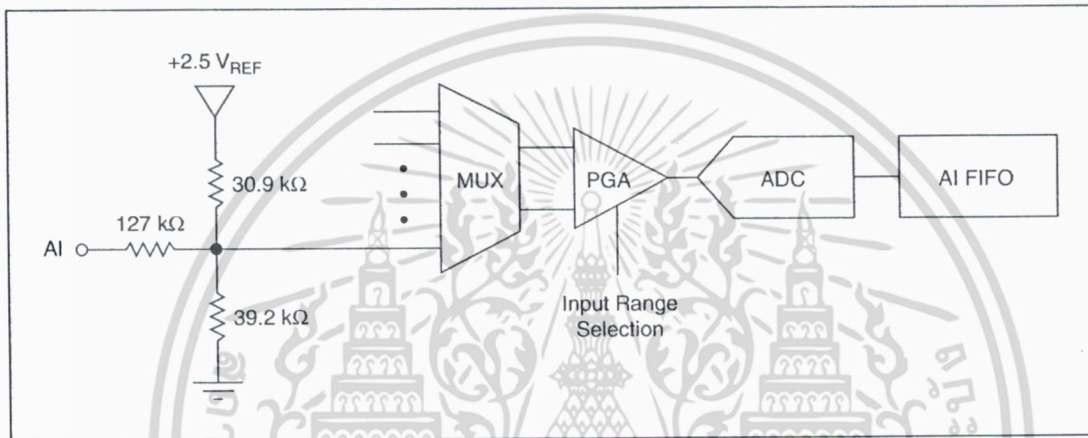


Figure 7. Analog Input Circuitry

MUX

The NI USB-6008/6009 has one analog-to-digital converter (ADC). The multiplexer (MUX) routes one AI channel at a time to the PGA.

PGA

The programmable-gain amplifier provides input gains of 1, 2, 4, 5, 8, 10, 16, or 20 when configured for differential measurements and gain of 1 when configured for single-ended measurements. The PGA gain is automatically calculated based on the voltage range selected in the measurement application.

A/D Converter

The analog-to-digital converter (ADC) digitizes the AI signal by converting the analog voltage into a digital code.

AI FIFO

The NI USB-6008/6009 can perform both single and multiple A/D conversions of a fixed or infinite number of samples. A first-in-first-out (FIFO) buffer holds data during AI acquisitions to ensure that no data is lost.

Analog Input Modes

You can configure the AI channels on the NI USB-6008/6009 to take single-ended or differential measurements. Refer to Table 6 for more information about I/O connections for single-ended or differential measurements.

Connecting Differential Voltage Signals

For differential signals, connect the positive lead of the signal to the AI+ terminal, and the negative lead to the AI- terminal.

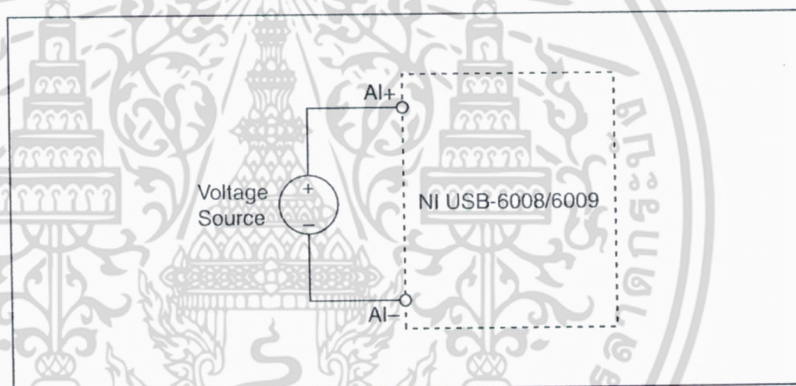


Figure 8. Connecting a Differential Voltage Signal

The differential input mode can measure ± 20 V signals in the ± 20 V range. However, the maximum voltage on any one pin is ± 10 V with respect to GND. For example, if AI 1 is +10 V and AI 5 is -10 V, then the measurement returned from the device is +20 V.

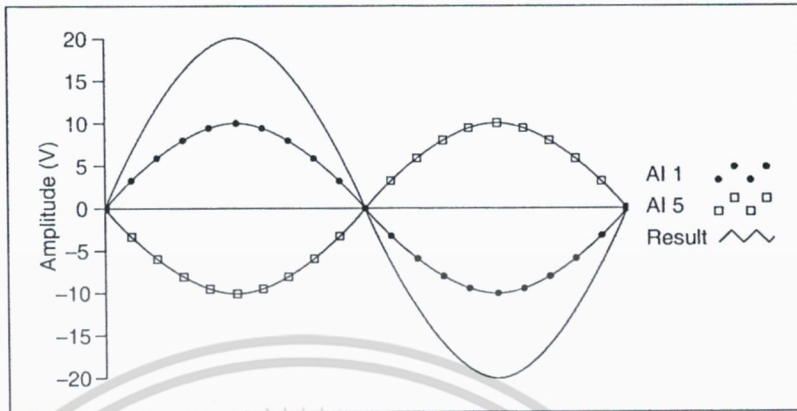


Figure 9. Example of a Differential 20 V Measurement

Connecting a signal greater than ± 10 V on either pin results in a clipped output.

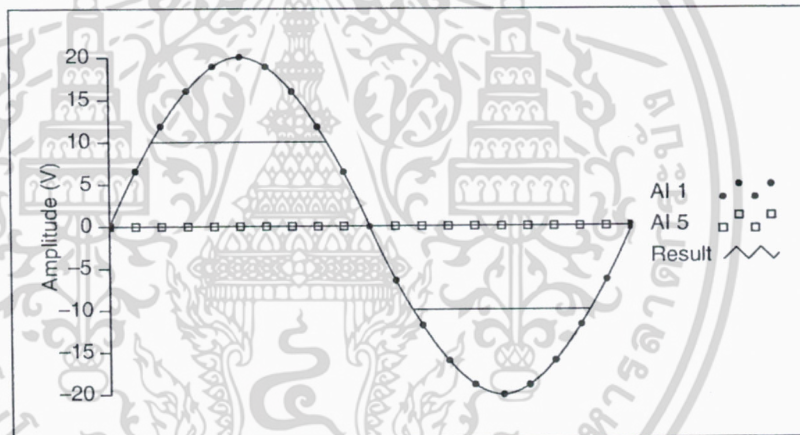


Figure 10. Exceeding ± 10 V on AI Returns Clipped Output

Connecting Referenced Single-Ended Voltage Signals

To connect referenced single-ended voltage signals (RSE) to the NI USB-6008/6009, connect the positive voltage signal to the desired AI terminal, and the ground signal to a GND terminal, as illustrated in Figure 11.

When no signals are connected to the analog input terminal, the internal resistor divider may cause the terminal to float to approximately 1.4 V when the analog input terminal is configured as RSE. This behavior is normal and does not affect the measurement when a signal is connected.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

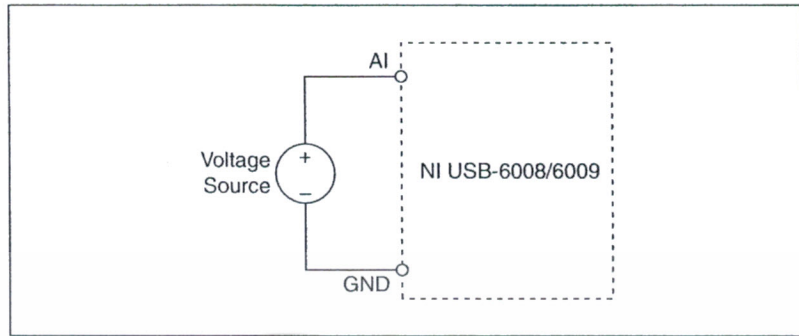


Figure 11. Connecting a Referenced Single-Ended Voltage Signal

Digital Trigger

When an AI task is defined, you can configure PFI 0 as a digital trigger input. When the digital trigger is enabled, the AI task waits for a rising or falling edge on PFI 0 before starting the acquisition. To use ai/Start Trigger with a digital source, specify PFI 0 as the source and select rising or falling edge.

Analog Output

The NI USB-6008/6009 has two independent AO channels that can generate outputs from 0–5 V. All updates of AO lines are software-timed.

Analog Output Circuitry

Figure 12 illustrates the analog output circuitry for the NI USB-6008/6009.

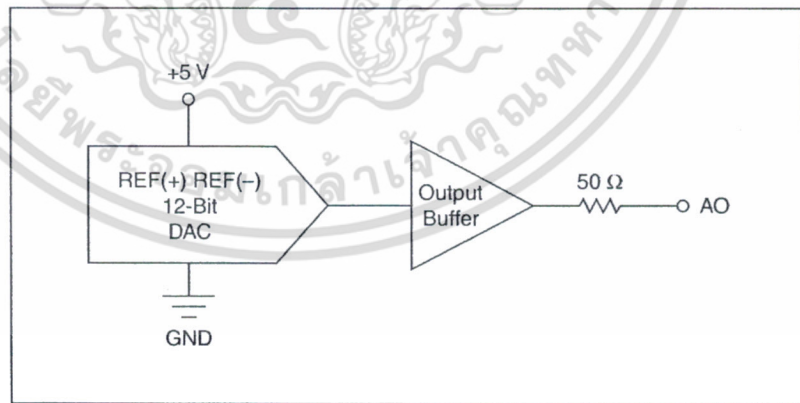


Figure 12. Analog Output Circuitry

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DACs

Digital-to-analog converters (DACs) convert digital codes to analog voltages.

Connecting Analog Output Loads

To connect loads to the NI USB-6008/6009, connect the positive lead of the load to the AO terminal, and connect the ground of the load to a GND terminal.

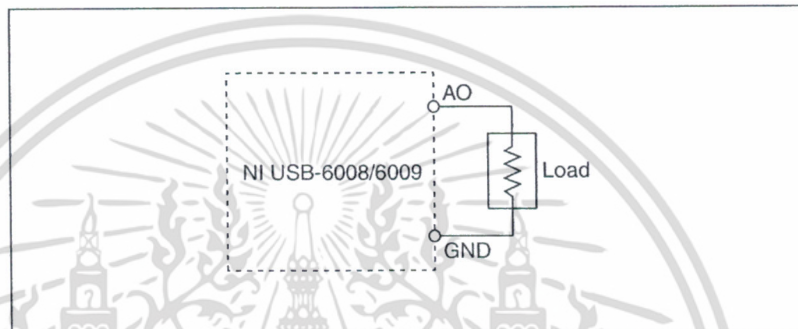


Figure 13. Connecting a Load

Minimizing Glitches on the Output Signal

When you use a DAC to generate a waveform, you may observe glitches in the output signal. These glitches are normal; when a DAQ switches from one voltage to another, it produces glitches due to released charges. The largest glitches occur when the most significant bit of the DAC code changes. You can build a lowpass deglitching filter to remove some of these glitches, depending on the frequency and nature of the output signal. Refer to ni.com/support for more information about minimizing glitches.

Digital I/O

The NI USB-6008/6009 has 12 digital lines, P0.<0..7> and P1.<0..3>, which comprise the DIO port. GND is the ground-reference signal for the DIO port. You can individually program all lines as inputs or outputs.

Digital I/O Circuitry

Figure 14 shows P0.<0..7> connected to example signals configured as digital inputs and digital outputs. You can configure P1.<0..3> similarly.

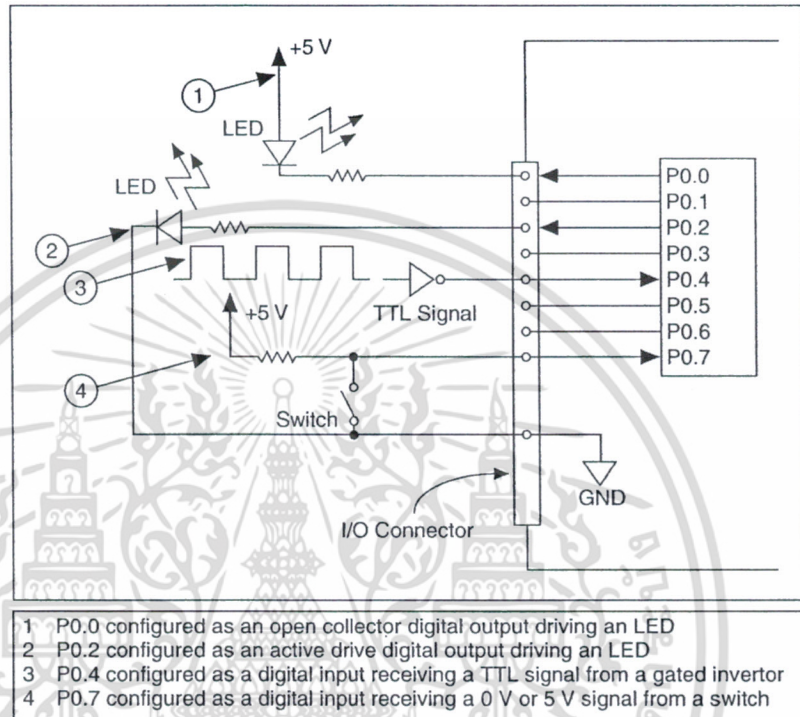


Figure 14. Example of Connecting a Load



Caution Exceeding the maximum input voltage ratings or maximum output ratings, which are listed in the *Specifications* section, can damage the device and the computer. National Instruments is not liable for any damage resulting from such signal connections.

Source/Sink Information

The default configuration of the NI USB-6008/6009 DIO ports is open collector, allowing 5 V operation, with an onboard 4.7 k Ω pull-up resistor. An external user-provided pull-up resistor can be added to increase the source current drive up to a 8.5 mA limit per line as shown in Figure 15.

The NI USB-6009 ports can also be configured as active drive using the NI-DAQmx API, allowing 3.3 V operation with a source/sink current limit of ± 8.5 mA. Refer to the *NI-DAQmx Help* for more information about how to set the DIO configuration.

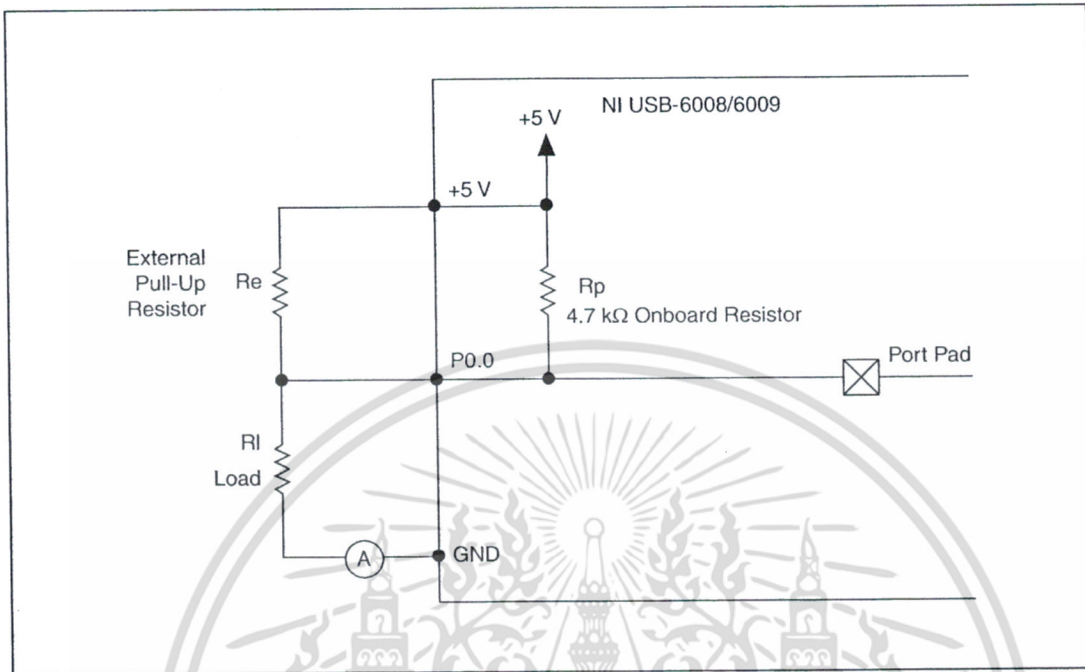


Figure 15. Example of Connecting an External User-Provided Resistor

Complete the following steps to determine the value of the user-provided pull-up resistor:

1. Place an ammeter in series with the load.
2. Place a variable resistor between the digital output line and the +5 V supply.
3. Adjust the variable resistor until the ammeter current reads as the intended current. The intended current must be less than 8.5 mA.
4. Remove the ammeter and variable resistor from your circuit.
5. Measure the resistance of the variable resistor. The measured resistance is the ideal value of the pull-up resistor.
6. Select a static resistor value for your pull-up resistor that is greater than or equal to the ideal resistance.
7. Re-connect the load circuit and the pull-up resistor.

ประวัตินักวิจัย

ชื่อ-นามสกุล ผู้ช่วยศาสตราจารย์ ดร. เฉลิมชาติ มานพ

ประวัติการศึกษา

- พ.ศ. 2541 ครุศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้ากำลัง
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
- พ.ศ. 2544 วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- พ.ศ. 2553 วิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการทำงาน

- พ.ศ. 2544-2549 อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- พ.ศ. 2549-ปัจจุบัน ผู้ช่วยศาสตราจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

งานวิจัยที่สนใจ

การเฝ้าตรวจสอบสถานะ การวินิจฉัย การป้องกัน และการจัดการพลังงานใน
เครื่องจักรกลไฟฟ้า