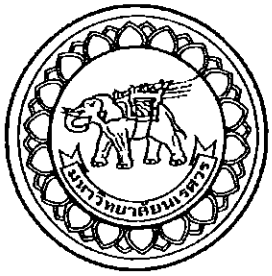
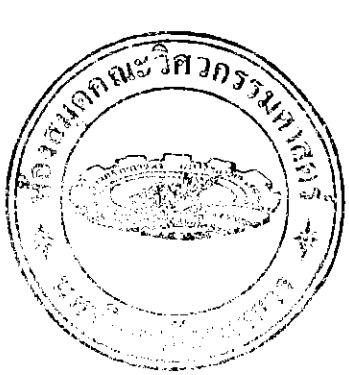




การศึกษาความผิดปกติของแบริ่งในอินดักชั่นมอเตอร์

A STUDY OF BEARING FAULT IN INDUCTION MOTOR

นายอนุชิต วังใจ รหัส 50383158

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 มิ.ย. 2555
เลขทะเบียน 1574042 3
เลขเรียกหนังสือ ฝ.
มหาวิทยาลัยนเรศวร D168 ก
2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การศึกษาความผิดปกติของแบริ่งในอินดักชั่นมอเตอร์

ผู้ดำเนินโครงงาน นายอนุชิต วังใจ รหัส 50383158

ที่ปรึกษาโครงงาน ดร. สมพร เรืองสินชัยวานิช

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

..... ที่ปรึกษาโครงงาน

(ดร. สมพร เรืองสินชัยวานิช)

..... กรรมการ

(ผศ.ดร. สุชาติ แย้มเม่น)

..... กรรมการ

(ดร. แสงชัย มังกรทอง)

ชื่อหัวข้อโครงงาน	การศึกษาความผิดปกติของแบริ่งในอินดักชั่นมอเตอร์
ผู้ดำเนินโครงงาน	นายอนุชิต วังใจ รหัส 50383158
ที่ปรึกษาโครงงาน	ดร.สมพร เรืองสินชัยวานิช
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

.....

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอการศึกษาการตรวจสอบความผิดพร่องแบริ่งในอินดักชั่นมอเตอร์ 3 เฟส ด้วยการวิเคราะห์สัญญาณกระแส และการวิเคราะห์สัญญาณเสียง ซึ่งแบ่งการวิเคราะห์ได้ 3 วิธีคือการวิเคราะห์ด้วยสัญญาณกระแสสเตเตอร์โดยใช้วิธีการแปลงฟลูเรียร์ การวิเคราะห์สัญญาณกระแสสเตเตอร์โดยวิธีปาร์คเวกเตอร์ และการวิเคราะห์สัญญาณเสียงโดยใช้วิธีการแปลงฟลูเรียร์ การศึกษาพบว่าการตรวจสอบความผิดพลาดของแบริ่งด้วยวิธีการวิเคราะห์สัญญาณเสียง มีความแม่นยำและชัดเจนกว่าการวิเคราะห์ด้วยสัญญาณกระแสเนื่องจากการวิเคราะห์ด้วยสัญญาณกระแสมีสัญญาณอื่นมารบกวนมากทำให้ขาดความแม่นยำในการวิเคราะห์ความผิดพร่องของแบริ่ง

Project title	A Study of Bearing Fault in Induction Motor
Name	Mr. Anuchit..Wungjai ID. 50383158
Project advisor	Mr. Somporn Ruangsinchaiwanich, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2010

.....

Abstract

This thesis presents a study examining the fault of the bearing in induction motor 3 phases from the current signal analysis and audio signal analysis. The analysis was divided three ways. The first way is stator current signal by using Fourier transform. The second way is the stator current signal by using Park Vector and the third way is audio signal analysis by using Fourier transform The experiment result show that examining the bearing fault from audio signal analysis has a precision and clarify more than the current signal analysis because the analysis from the current signal has a lot of other signal to noise which makes the analysis without precision in the analysis of the bearing.

กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์ในครั้งนี้ คณะผู้จัดทำขอขอบพระคุณ ดร.สมพร เรืองสินชัยวานิช อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ได้ให้ความกรุณา แนะแนวความคิด ความรู้ แนวทางแก้ไขปัญหาตลอดจนให้โอกาสในการจัดทำปริญญานิพนธ์ รวมไปถึงการเอื้อเฟื้อเอกสารแหล่งข้อมูลในการค้นคว้าเพิ่มเติมและอุปกรณ์ต่างๆที่จำเป็นในการทดลองช่วยให้ปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี รวมทั้งคณะกรรมการการสอบปริญญานิพนธ์ทุกท่านที่ได้ให้คำแนะนำ ตรวจทานเพื่อให้ปริญญานิพนธ์นี้มีความสมบูรณ์มากยิ่งขึ้น และผู้มีพระคุณทุกท่านที่มิได้กล่าวถึงในที่นี้ที่ได้ให้คำแนะนำที่เป็นประโยชน์ต่อการทำปริญญานิพนธ์นี้

สุดท้ายนี้สิ่งที่คณะผู้จัดทำรำลึกและซาบซึ้งอยู่เสมอนั่นคือพระคุณของบิดา มารดา ซึ่งเข้าใจและให้ความสนับสนุนช่วยเหลือในทุกๆด้าน ในการศึกษาและทำปริญญานิพนธ์จนกระทั่งประสบผลสำเร็จ คณะผู้จัดทำจึงขอกราบขอบพระคุณมา ณ โอกาสนี้

นายอนุชิต วังใจ

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์.....ก
บทคัดย่อภาษาไทย.....ข
บทคัดย่อภาษาอังกฤษ.....ค
กิตติกรรมประกาศ.....ง
สารบัญ.....จ
สารบัญตาราง.....ช
สารบัญรูป.....ซ

บทที่ 1 บทนำ
1.1 ที่มาและความสำคัญของโครงงาน.....1
1.2 วัตถุประสงค์ของโครงงาน.....1
1.3 ขอบเขตโครงงาน.....2
1.4 แผนการดำเนินงาน.....3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....4
1.6 งบประมาณ.....4

บทที่ 2 หลักการและทฤษฎีเบื้องต้น
2.1 อินดักชั่นมอเตอร์สามเฟส (Three phase motors).....5
2.1.1 การจัดแบ่งประเภทของมอเตอร์ไฟฟ้าสามเฟส.....5
2.1.2 หลักการทำงานของอินดัคชั่นมอเตอร์สามเฟส.....8
2.2 ความผิดพร่องของเครื่องจักรกลไฟฟ้า.....9
2.3 แบริ่ง (Bearing).....10
2.4 การวิเคราะห์ความผิดปกติในมอเตอร์ (Motor Failure Analysis).....11
2.4.1. ความเสียของมอเตอร์.....12
2.5. การวินิจฉัยความผิดปกติของมอเตอร์.....13

สารบัญ (ต่อ)

หน้า

บทที่ 3 ออกแบบการทดลอง

3.1. อุปกรณ์ที่ใช้ในการทดลอง..... 16

3.2. การวิเคราะห์สัญญาณกระแสสเตเตอร์.....17

3.2.1. วิเคราะห์โดยการแปลงฟูเรียร์.....18

3.2.2. วิเคราะห์โดยเวคเตอร์ปาร์ค.....18

3.3. การวิเคราะห์สัญญาณเสียง.....19

บทที่ 4 ผลการทดลอง

4.1 การวิเคราะห์ด้วยการแปลงสัญญาณกระแสฟูเรียร์..... 20

4.2 การวิเคราะห์ด้วยการแปลงสัญญาณกระแสเวคเตอร์ปาร์ค.....25

4.3 การวิเคราะห์สัญญาณเสียง.....29

4.4. ผลที่เกิดจากโหลด.....30

4.4.1. ผลที่เกิดจากโหลดของการแปลงสัญญาณกระแสฟูเรียร์.....34

4.4.2. ผลที่เกิดจากโหลดของการแปลงสัญญาณกระแสเวคเตอร์ปาร์ค.....35

4.4.2. ผลที่เกิดจากโหลดของการแปลงสัญญาณเสียง.....36

บทที่ 5 สรุปผลการทดลอง และข้อเสนอแนะ

5.1. สรุปผลการทดลอง..... 39

5.2. ปัญหาที่พบขณะดำเนินโครงงาน..... 40

5.3. ข้อเสนอแนะ.....40

เอกสารอ้างอิง.....41

ภาคผนวก.....42

ประวัติผู้เขียนโครงงาน.....85

สารบัญตาราง

ตารางที่ หน้า

1.1 แสดงแผนการดำเนินงาน..... 3

2.1 การเปรียบเทียบเทคนิคการตรวจสอบความเสียหายของมอเตอร์..... 13

3.1 แสดงตำแหน่งสัญญาณที่เกิดขึ้นเมื่อแบริ่งเกิดความเสียหายที่ร่องลื่นด้านนอก..... 18

สารบัญรูป

รูปที่ หน้า

2.1 ส่วนประกอบของมอเตอร์ไฟฟ้าสามเฟสแบบโรเตอร์กรงกระรอก..... 5

2.2 ส่วนประกอบของมอเตอร์ไฟฟ้าสามเฟสแบบวาวด์โรเตอร์.....7

2.3 ลักษณะการเกิดสนามแม่เหล็กหมุน ในส่วนอยู่กับที่ ของอินดัคชั่นมอเตอร์สามเฟส..... 9

2.4 เปอร์เซ็นต์ความผิดพร่องของส่วนประกอบต่างๆ ในมอเตอร์เหนี่ยวนำ..... 10

2.5 แบริ่ง(Bearing)..... 10

2.6 ส่วนประกอบของแบริ่ง (Bearing)..... 11

2.7 แสดงค่าพารามิเตอร์ในการคำนวณหาคุณลักษณะของความถี่.....14

2.8 แสดงการจำลองความเสียหายของแบริ่ง (Bearing).....15

3.1 แสดงอุปกรณ์ในการทดลอง.....16

3.2 การเก็บค่าสัญญาณกระแสสเตเตอร์.....17

3.3 การเก็บค่าสัญญาณเสียง..... 19

4.1 แสดงสัญญาณกระแสเมื่อร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลังปกติ..... 20

4.2 แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 1 ที่ ทั้งด้านหน้าและด้านหลัง.....21

4.3. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 2 ที่ ด้านหลังปกติ.....21

4.4. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลัง.....26

4.5. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 3 ที่ ด้านหลังปกติ.....22

4.6. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ทั้งด้านหน้าและด้านหลัง.....23

4.7. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ.....23

4.8. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 4 ที่ ทั้งด้านหน้าและด้านหลัง.....24

4.9 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลังปกติ.... 25

4.10 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกของแบริ่งเสียหาย 1 ที่ ด้านหน้าและด้านหลัง... 25

4.11 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกแบริ่งด้านหน้าเสียหาย 2 ที่ ด้านหลังปกติ..... 26

4.12 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ด้านหน้าและด้านหลัง... 26

4.13 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกแบริ่งด้านหน้าเสียหาย 3 ที่ ด้านหลังปกติ..... 27

4.14 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ด้านหน้าและด้านหลัง... 27

4.15 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ..... 28

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.16 เปรียบเทียบแบริ่งปกติ กับร่องลื่นด้านนอกแบริ่งเสียหาย 4 ที่ ทั้งด้านหน้าและด้านหลัง..... 28
4.17 แสดงสัญญาณเสียงเมื่อแบริ่งปกติ.....29
4.18 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลัง..... 29
4.19 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งเสียหาย 1 ที่ ทั้งด้านหน้าและด้านหลัง....30
4.20 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 2 ที่ ด้านหลังปกติ.....30
4.21 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลัง....31
4.22 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 3 ที่ ด้านหลังปกติ.....31
4.23 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ด้านหน้าและด้านหลัง..... 32
4.24 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ..... 32
4.25 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งเสียหาย 4 ที่ ด้านหน้าและด้านหลัง..... 33
4.26 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ..... 34
4.27 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ..... 35
4.28 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 0%..... 36
4.29 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 25%..... 36
4.30 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 50%..... 37
4.31 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 75%..... 37
4.32 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 100%..... 38

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เครื่องจักรกลไฟฟ้าในโรงงานอุตสาหกรรมทั่วๆ ไปจะมีมอเตอร์เหนี่ยวนำเป็นส่วนประกอบหลักของเครื่องจักรกล ส่วนใหญ่จะเป็นมอเตอร์เหนี่ยวนำชนิดที่มีโรเตอร์เป็นแบบกรงกระรอก (squirrel cage rotor) เพราะมีความแข็งแรง ทนทาน ราคาถูก ประสิทธิภาพสูง แต่เมื่อใช้งานไประยะหนึ่งมอเตอร์ย่อมเกิดความเสียหายได้เนื่องจากสาเหตุหลายประการ หากเราทราบถึงความผิดปกติของมอเตอร์ตั้งแต่ระยะแรก จะสามารถลดความเสียหายที่อาจจะเกิดขึ้นได้มาก

การเกิดความเสียหายของมอเตอร์เหนี่ยวนำโดยส่วนใหญ่นั้นจะเริ่มจากความเสียหายที่แบริ่ง (Bearing) ทำให้เกิดการสั่นสะเทือน และเกิดความเสียหายที่ส่วนอื่นตามมา โครงงานนี้จึงสนใจตรวจสอบความเสียหายของแบริ่ง เพื่อมิให้ความเสียหายลุกลามไปในส่วนอื่นๆของมอเตอร์

ความเสียหายที่เกิดขึ้นกับแบริ่ง สามารถตรวจสอบได้หลายวิธี โดยปกติแล้วนิยมใช้การตรวจสอบโดยการวัดการสั่นสะเทือนของมอเตอร์ แต่เนื่องจากเครื่องมือวัดการสั่นสะเทือนมีราคาแพง จึงไม่เหมาะกับการนำมาใช้กับมอเตอร์ขนาดกลางและขนาดเล็ก โครงงานนี้จึงนำเสนอวิธีการตรวจสอบความผิดพร่องของแบริ่ง ด้วยวิธีการวิเคราะห์สัญญาณกระแสมอเตอร์ และการวิเคราะห์สัญญาณเสียง

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาคุณลักษณะสัญญาณกระแสของมอเตอร์เมื่อแบริ่ง เกิดความเสียหายในรูปแบบต่างๆ โดยวิธีการแปลงฟูเรียร์
2. เพื่อศึกษาคุณลักษณะสัญญาณกระแสของมอเตอร์เมื่อแบริ่ง เกิดความเสียหายในรูปแบบต่างๆ โดยวิธีเวคเตอร์ปาร์ค
3. เพื่อศึกษาคุณลักษณะสัญญาณเสียงของมอเตอร์เมื่อแบริ่ง เกิดความเสียหายในรูปแบบต่างๆ
4. เพื่อเปรียบเทียบการตรวจสอบความผิดปกติของแบริ่ง ที่สภาวะโหลด 0% 25% 50% 75% และ100%

1.3 ขอบเขตโครงงาน

1. วิเคราะห์การตรวจสอบความผิดพร่องของแบริ่ง โดยวิธีการวิเคราะห์การแปลงสัญญาณกระแสฟูเรียร์ การแปลงสัญญาณกระแสเวคเตอร์ปาร์ค และวิธีการวิเคราะห์สัญญาณเสียง

1.4 แผนการดำเนินงาน

ตารางที่ 1.1 แสดงแผนการดำเนินงาน

กิจกรรม	ปี 2553							ปี 2554		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. หาหัวข้อโครงงานที่สนใจ	←	→								
2. ศึกษาการวิเคราะห์คุณลักษณะของสัญญาณฮาร์มอนิกส์ในโคเมนความถี่			←	→						
3. ออกแบบการทดลอง					←	→				
4. เก็บค่าสัญญาณกระแสและสัญญาณเสียงของมอเตอร์ที่แบริ่ง เกิดความเสียหายในรูปแบบต่างๆ						←	→			
5. วิเคราะห์ผลการทดลองและสรุปผลการทดลอง								←	→	
6. จัดทำรายงาน										← →

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.1 สามารถตรวจสอบความผิดปกติที่เกิดขึ้นกับแบริ่ง ของมอเตอร์ได้

1.2 สามารถป้องกันความเสียหายที่อาจจะเกิดกับมอเตอร์ในอนาคตได้

1.3 สามารถนำความรู้ไปประยุกต์ใช้ในโรงงานอุตสาหกรรมได้

1.6 งบประมาณ

1. ค่าอุปกรณ์ที่ใช้ในการทดลอง	500 บาท
2. ค่าจัดทำรายงาน	500 บาท
รวม	1,000 บาท

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

มหาวิทยาลัยนเรศวร

บทที่ 2
หลักการและทฤษฎีเบื้องต้น

2.1 อินดักชั่นมอเตอร์สามเฟส (Three phase motors)

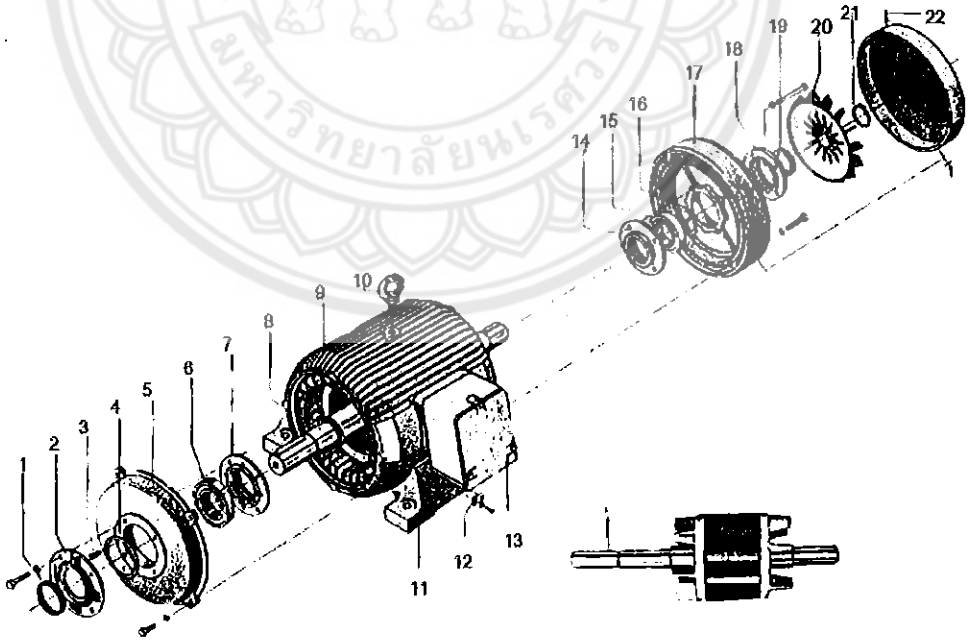
มอเตอร์ไฟฟ้าสามเฟสเป็นมอเตอร์ไฟฟ้ากระแสสลับที่นิยมใช้มากโดยเฉพาะกำลังแรงม้าสูงๆ มีคุณสมบัติพิเศษที่ให้ความเร็วคงที่ และมีการออกแบบให้เหมาะสมกับการนำไปใช้งานที่แตกต่างกัน เช่นมีแรงบิดเริ่มต้น(Starting torque) สูงมีกระแสเริ่มต้นหมุน (Strating current) ต่ำ ใช้ได้กับแรงดันไฟฟ้า และความถี่ทุกๆ มาตรฐานหรือใช้กับแรงดันไฟฟ้าได้สองระดับเป็นต้น

2.1.1 การจัดแบ่งประเภทของมอเตอร์ไฟฟ้าสามเฟส

การจัดแบ่งประเภทของมอเตอร์ไฟฟ้าสามเฟส ตามความแตกต่างของ ส่วนหมุน (Rotor) แบ่งเป็นหลายแบบ แต่ในที่นี้จะศึกษา 2 แบบดังนี้

2.1.1.1 โรเตอร์แบบกรงกระรอก (Squirrel cage rotor)

โครงสร้างและส่วนประกอบของมอเตอร์ไฟฟ้าสามเฟสแบบโรเตอร์กรง กระรอก มีส่วนประกอบตามรูปที่ 2.1



รูปที่ 2.1 ส่วนประกอบของมอเตอร์ไฟฟ้าสามเฟสแบบโรเตอร์กรงกระรอก [1]

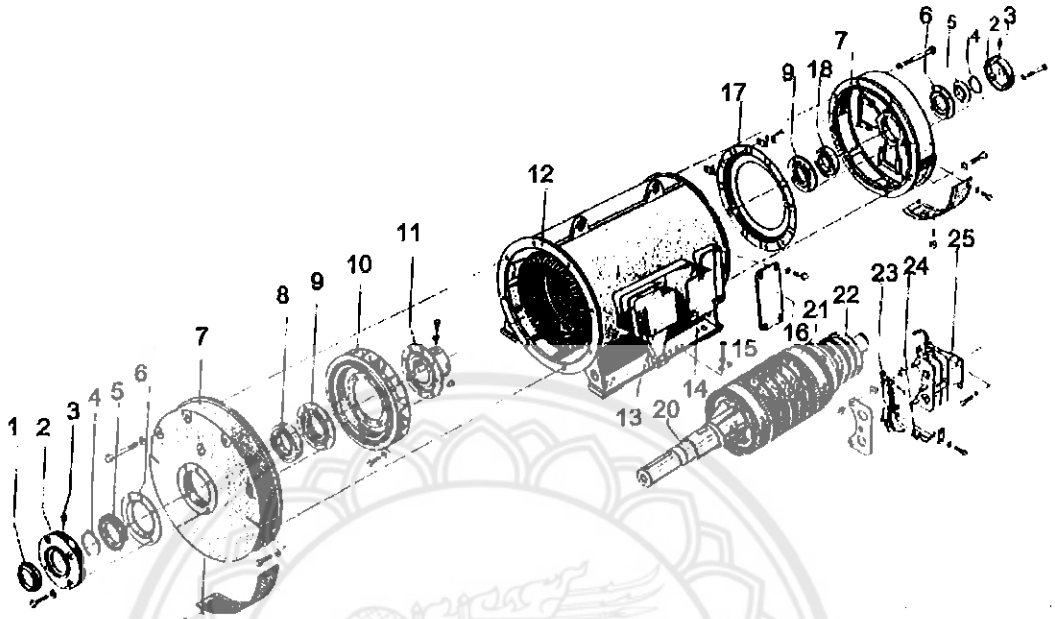
1. แหวนด้านเพลาขับ
2. ฝาครอบแบริ่งด้านนอก
3. สปริงกด
4. แหวนกดแบริ่ง
5. ฝาครอบ(ด้านเพลางาน)
6. แบริ่งเพลางาน
7. ฝาครอบแบริ่งด้านในด้านเพลางาน
8. เพลาที่ยึดติดกับโรเตอร์
9. โครงพร้องตัวสเตเตอร์และขดลวด
10. หูหิ้ว
11. ฐานรับ
12. จุดต่อลงดิน
13. กล่องต่อสาย
14. ฝาครองแบริ่งด้านในด้านผัดลม
15. แบริ่งด้านผัดลม
16. แหวนสปริงสำหรับแบริ่ง
17. ฝาครอบด้านผัดลม
18. ฝาครอบแบริ่งด้านนอก
19. แหวนด้านผัดลม
20. ใบผัดลม
21. แหวนสปริงสำหรับผัดลม
22. ฝาครอบผัดลม

จากรูปที่ 2.1 ส่วนอยู่กับที่ (Stator) ทำจากเหล็กแผ่นบางที่อัดกัน (Laminated sheet steel) โดย ปกติแผ่นเหล็กมีความหนา 0.35 – 0.5 มม. และแผ่นเหล็กแต่ละแผ่นจะฉาบด้วยน้ำยาวานิช หรือออกไกด์ บนแกนนี้จะมีขดลวด 3 ชุด พันวางอยู่ในร่องรอบแกนจัดวางตำแหน่งต่างกัน 120 องศา ทางไฟฟ้าแล้ว ต่อปลายสายออกมา 6 ปลายด้วยกัน เพื่อเลือกต่อใช้งานเป็นแบบสตาร์ (Star) หรือเดต้า (Delta)

ส่วนหมุน(Rotor) เป็นแบบขดลวดที่อยู่บนส่วนหมุนแต่เดิมเป็นแท่งทองแดง เส้นโต และปลายทั้งสองเชื่อมติดกันด้วยวงแหวนทองแดง ในปัจจุบันนิยมใช้อลูมิเนียมแทนทองแดง มอเตอร์แบบนี้เรียกอีกชื่อหนึ่งว่า สตวีรเคจอินดักชั่นมอเตอร์ (Squirrel cage motor)

2.1.1.2 โรเตอร์แบบวาวด์ (Wound rotor)

โครงสร้างและส่วนประกอบของมอเตอร์ไฟฟ้าสามเฟสแบบวาวด์โรเตอร์ มีส่วนประกอบตามรูปที่ 2.2



รูปที่ 2.2 ส่วนประกอบของมอเตอร์ไฟฟ้าสามเฟสแบบวาวด์โรเตอร์ [1]

1. แหวนก้านเพลาขับ
2. ฝาครอบแบริ่งด้านนอก
3. ที่ยึดจารบี
4. แหวนสปริง
5. แป้นปรับศูนย์
6. แหวนกัน
7. ฝาครอบด้านหลัง
8. แบริ่งก้นเพลา
9. ฝาครอบแบริ่งก้นเพลา
10. ใบผัดลม
11. ที่ยึดใบผัดลม
14. กล่องต่อสายสำหรับโรเตอร์
15. จุดต่อสายดิน
16. ฝาปิด(สำหรับซ้อมบำรุง)
17. แผ่นกัน
18. แบริ่ง
19. ตาข่ายป้องกัน
20. เพลา
21. ตัวโรเตอร์พร้อมขดลวด
22. สปริง
23. ซ่องแปรงถ่าน
24. แปรงถ่าน

12. โครงพร้อมตัวสเตเตอร์และขดลวด 25. ที่ยึดแปรง

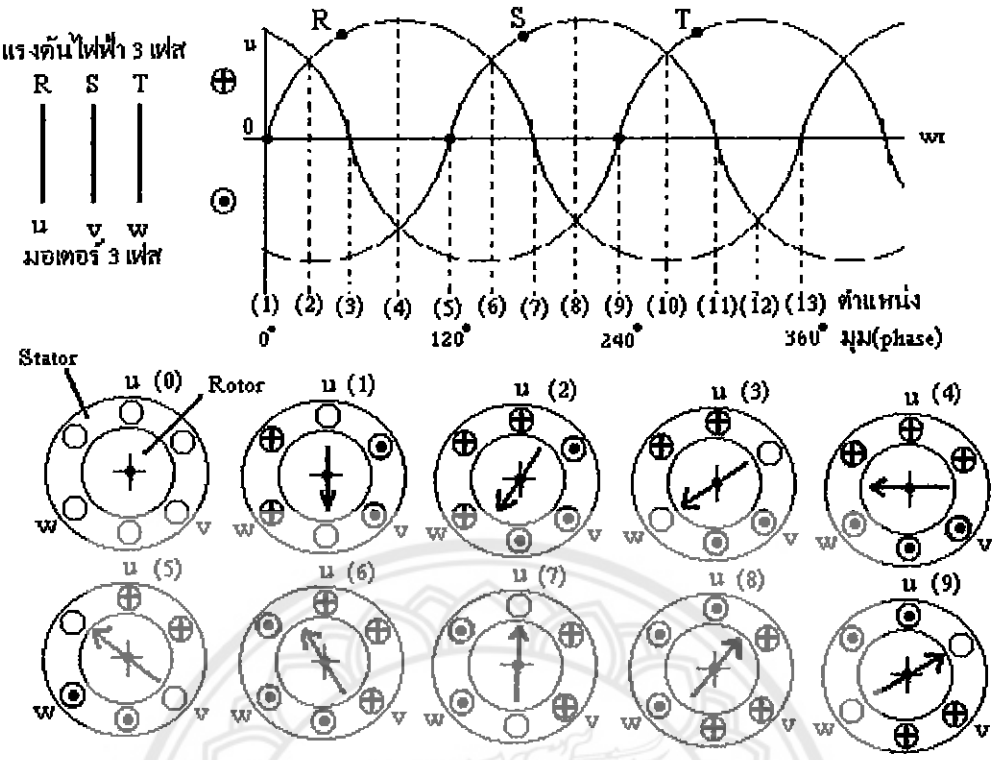
13. กล่องต่อสายสำหรับสเตเตอร์

จากรูปที่ 2.2 ส่วนอยู่กับที่ (Stator) จะเหมือนกับแบบโรเตอร์กรงกระรอก

ส่วนหมุน(Rotor) เป็นแบบขดลวดที่อยู่บนส่วนหมุนจะคล้ายกับขดลวดที่พันอยู่บนส่วนอยู่กับที่ คือเป็นขดลวด 3 ชุด วางตำแหน่ง ต่างกัน 120 องศาทางไฟฟ้า และต่อวงจรเป็นแบบสตาร์ และปลายสายนำไปต่อกับวงแหวนลื่น 3 วง วงแหวนนี้จะต่อเข้ากับความต้านทานภายนอกเพื่อต้องการปรับค่าความต้านทานของส่วนหมุนในขณะเริ่มต้นหมุน จากลักษณะของส่วนหมุนนี้ จึงเรียกมอเตอร์แบบนี้อีกชื่อหนึ่งว่า วาวด์โรเตอร์อินดักชั่นมอเตอร์ (Wound rotor induction motor) หรือ สลิปริงมอเตอร์ (Slip ring motor)

2.1.2 หลักการทำงานของอินดักชั่นมอเตอร์สามเฟส

มอเตอร์แบบนี้ทำงานโดยอาศัยหลักการเหนี่ยวไฟฟ้าและแม่เหล็ก โดยเมื่อจ่ายแรงดันไฟฟ้าเข้าส่วนอยู่กับที่ (Stator) ทำให้เกิดสนามแม่เหล็กหมุนที่ส่วนหมุนด้วยความเร็วซิงโครนัส (Sychronus speed) สนามแม่เหล็กนี้จะไปตัดกับตัวนำบนตัวส่วนหมุน(Rotor) เกิดแรงเคลื่อนไฟฟ้าเหนี่ยวนำขึ้นที่ตัวนำ โดยทิศทางของกระแสไฟฟ้าจะเป็นไปตามกฎของเลนซ์ คือจะพยามสร้างสนามแม่เหล็กขึ้นมาต้านกับสนามแม่เหล็ก เป็นผลให้เกิดแรงบิด(Torque) ทำให้ส่วนหมุน เกิดการหมุนไปได้ โดยทิศทางการหมุนไปในทิศทางเดียวกับสนามแม่เหล็กหมุนที่เกิดบนส่วนอยู่กับที่ และจะพยามหมุนให้ได้ความเร็วทันกับสนามแม่เหล็กหมุนนี้

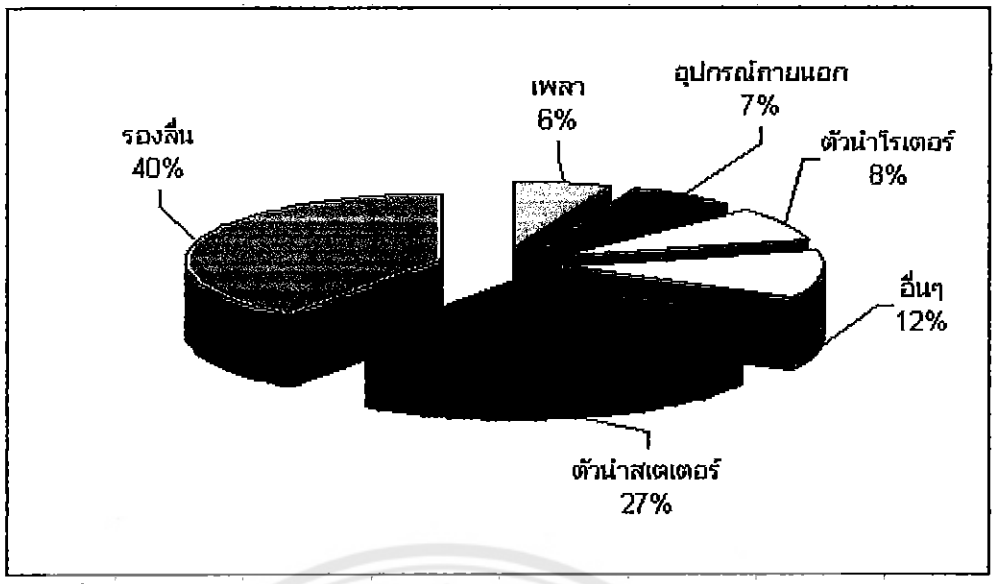


รูปที่ 2.3 ลักษณะการเกิดสนามแม่เหล็กหมุน ในส่วนอยู่กับที่ ของอินดัคชั่นมอเตอร์สามเฟส [1]

จากรูปที่ 2.3 เมื่อจ่ายแรงดันไฟฟ้า 3 เฟส ให้กับมอเตอร์ไฟฟ้า 3 เฟสโดยต่อเฟส R ให้กับขั้ว u เฟส S ให้กับขั้ว v เฟส T ให้กับขั้ว w เมื่อพิจารณาทิศทางการไหลของกระแสไฟฟ้าในขดลวดต่าง ๆ และสนาม แม่เหล็กรวมที่เกิดจากแรงดันไฟฟ้า-กระแสไฟฟ้าในขดลวด ณ.ตำแหน่งต่าง ๆ (1 , 2 , 3 ...) จะเห็นว่า สนามเหล็กรวมจะค่อย ๆ หมุนไปในทิศทางตามเข็มนาฬิกา

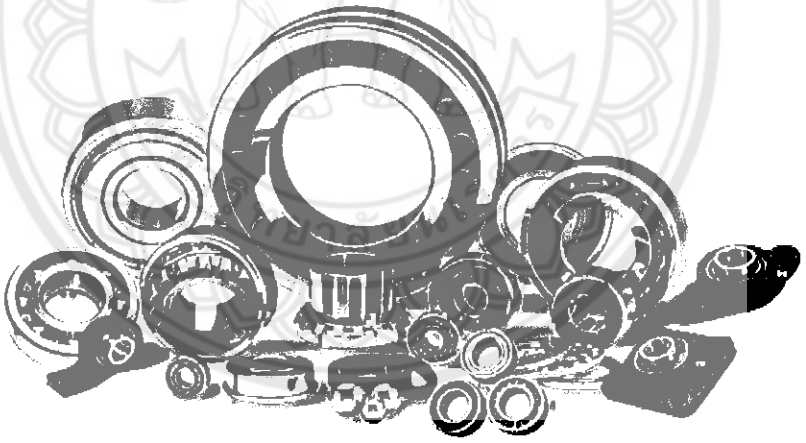
2.2 ความผิดพร่องของเครื่องจักรกลไฟฟ้า

ความผิดพร่องของเครื่องจักรกลไฟฟ้าแบ่งออกได้เป็น 2 ส่วน คือ ความผิดพร่องภายในและความความผิดพร่องภายนอก สำหรับมอเตอร์เหนี่ยวนำนั้นความผิดพร่องภายในตัวมอเตอร์ได้แก่short-turn หรือ short-coil ในขดลวดสเตเตอร์ ขดลวดรั่วลงดิน (earth fault) ขดลวดหายไปบางเฟส(open circuit) และความไม่สมมาตรของแกนโรเตอร์ (unbalance rotor core) ส่วนความผิดพร่องภายนอกตัวมอเตอร์ได้แก่ โหลดทางกลเกิน (mechanical overload) แรงเคลื่อนเปลี่ยนแปลง (supplyvoltage changes) หรือเกิดการเปลี่ยนแปลงทางความถี่ (frequency changes) ความผิดพร่องของ เครื่องจักรกลไฟฟ้าแบ่งเป็นเปอร์เซ็นต์ความเสียหายได้ดังรูปที่ 2.4



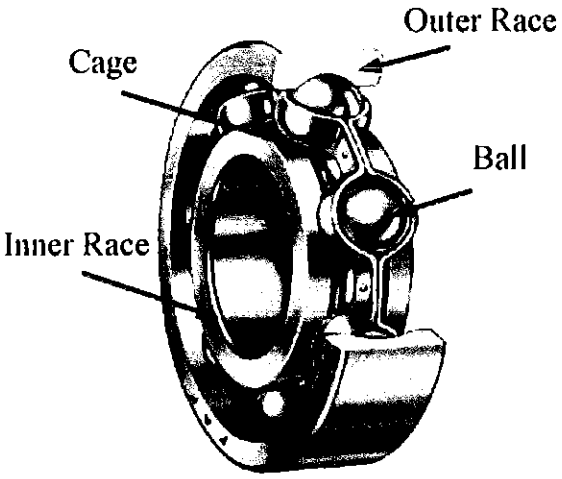
รูปที่ 2.4 เปอร์เซ็นต์ความผิดพร่องของส่วนประกอบต่างๆ ในมอเตอร์เหนี่ยวนำ [2]

2.3 แบริ่ง (Bearing)



รูปที่ 2.5 แบริ่ง(Bearing) [3]

แบริ่ง เป็นองค์ประกอบสำคัญของเครื่องจักรที่ต้องการ การหล่อลื่น และแทบจะกล่าวได้ว่าเครื่องจักรเกือบทุกเครื่องจะต้องมีแบริ่งคือสิ่งที่ช่วยรองรับหรือช่วยยึดชิ้นส่วนต่าง ๆของเครื่องจักรที่มีการหมุนให้อยู่ในตำแหน่งที่ถูกต้อง แบริ่ง มีหลายชนิด ขึ้นอยู่กับการออกแบบ และใช้งาน เช่น บอลแบริ่ง โรลเลอร์แบริ่ง บอลทรัสต์แบริ่ง โรลเลอร์ทรัสต์แบริ่ง และเทปเปอร์โรลเลอร์ทรัสต์แบริ่ง



รูปที่ 2.6 ส่วนประกอบของแบริ่ง [4]

แบริ่งลูกปืน (Rolling Bearing) การเคลื่อนไหวของ แบริ่งกาบ จะเกิดในลักษณะเลื่อนสัมผัส (Sliding) ของผิวสัมผัสทั้งสอง ซึ่งจะทำให้เกิดแรงเสียดทานขึ้น แรงเสียดทาน นี้สามารถลดลงได้โดยการเปลี่ยนการเคลื่อนไหวแบบเลื่อนสัมผัส (Rolling) โดยการติดตั้งวงแหวนซึ่งประกอบด้วยลูกปืน ที่ทำด้วยโลหะแข็ง อาจจะมีลักษณะกลมเหมือนลูกบอลล์ หรือเป็นแบบลูกกลิ้งเคลื่อนที่อยู่ระหว่างวงแหวนชั้นในและชั้นนอก

แบริ่งลูกปืนส่วนใหญ่จะใช้ จาระบี เป็นตัวหล่อลื่น จารบียังทำหน้าที่เป็นซีลป้องกันไม่ให้ความชื้นหรือสิ่งสกปรกต่าง ๆ เข้าไปทำความเสียหายแก่ลูกปืน การเลือกชนิดของ จาระบี ขึ้นอยู่กับความเร็วรอบแรงกด และอุณหภูมิของแบริ่ง ในขณะใช้งาน โดยทั่วไปมักใช้ จาระบี เอนกประสงค์ที่ทำด้วยสบู่ลิเทียมในงานบางประเภทอาจมีความต้องการ จาระบี ที่สามารถทนต่ออุณหภูมิสูงและอุณหภูมิต่ำคือ ไม่เหลวและไม่ทำปฏิกิริยาออกซิเดชั่นในขณะที่ใช้งานภายใต้อุณหภูมิต่ำ

2.4 การวิเคราะห์ความผิดปกติในมอเตอร์ (Motor Failure Analysis)

2.4.1. ความเสียของมอเตอร์ ความเสียหายของอินดักชั่นมอเตอร์ประกอบด้วย ความเสียหายทางกล และความเสียหายของฉนวน

2.4.1.1. ความเสียหายของฉนวน (Insulation system failure) คิดเป็น 5 เปอร์เซ็นต์ของความเสียหายทั้งหมด เนื่องจากสาเหตุดังนี้

1. ความชื้น (Moisture)
2. อุณหภูมิสูง (Over-temperature)
3. ระบบไฟฟ้าไม่เสถียร (System surge)
4. การลัดวงจร (Faulty earth)

2.4.1.2. ความเสียหายทางกล (Mechanical failures) เป็นความผิดปกติที่เกิดขึ้นบ่อย สามารถแบ่งประเภทได้ดังนี้

1. ความผิดปกติของสเตเตอร์ผลเกิดจากการเปิดวงจรของขดลวดสเตเตอร์
2. ตัวนำโรเตอร์แตกหรือวงแหวนโรเตอร์แตก
3. บริเวณช่องว่างอากาศไม่สม่ำเสมอ
4. เพลาโค้งงอ (ความผิดปกติจากการเคลื่อนที่)
5. การเยื้องศูนย์
6. ความเสียหายของแบริ่ง

2.4.1.3. ความเสียหายที่เกิดขึ้นก่อให้เกิดความผิดปกติของมอเตอร์ดังนี้

1. ความไม่สมดุลของแรงดันบริเวณช่องว่างอากาศ และกระแส
2. เพิ่มการสั่นสะเทือนของยอร์ก และผลของค่าเฉลี่ยทอร์กลดลง
3. มอเตอร์มีความสูญเสียเพิ่มขึ้น, ความร้อนสูงขึ้น และประสิทธิภาพลดลง
4. ฮาร์โมนิกของกระแสเพิ่มขึ้น
5. เกิดการรั่วไหลของฟลักแม่เหล็ก

2.5. การวินิจฉัยความผิดปกติของมอเตอร์

1. การเฝ้าสังเกตการณ์สั่นสะเทือน (Vibration monitoring)
2. การวิเคราะห์สัญญาณกระแสสเตเตอร์ (MCSA)
3. การเฝ้าสังเกตสนามแม่เหล็ก (Electromagnetic field monitoring)
4. การวิเคราะห์ทางเคมี (Chemical analysis)
5. การวัดอุณหภูมิ (Temperature measurement)
6. การวัดอินฟราเรด (Infrared measurement)
7. การวัดริชาร์ดบางส่วน (Partial discharge measurement)

จากตารางที่ 2.1 แสดงการเปรียบเทียบเทคนิคการตรวจสอบความเสียหายของมอเตอร์ พบว่า แต่ละเทคนิคการตรวจสอบจะมีความสามารถในการตรวจสอบความผิดปกติที่แตกต่างกัน ขึ้นอยู่กับต้องการตรวจสอบความผิดปกติ ที่เกิดขึ้นประเภทใด ซึ่งโครงงานนี้พิจารณา เทคนิคการวิเคราะห์ความเสียหายที่เกิดขึ้นกับแบริ่ง เป็นหลัก โดยการวิเคราะห์จากสัญญาณกระแสสเตเตอร์ และเพิ่มการศึกษาสัญญาณเสียงเมื่อแบริ่ง เกิดการเสียหายอีกด้วย

ตารางที่ 2.1 การเปรียบเทียบเทคนิคการตรวจสอบความเสียหายของมอเตอร์

วิธีการ	ความผิดปกติ			
	ฉนวน	ขดลวดสเตเตอร์	ตัวนำโรเตอร์	แบริ่ง
การสั่นสะเทือน			√	√
กระแสสเตเตอร์		√	√	√
ฟลักแม่เหล็ก		√	√	
ก๊าซ	√	√	√	
ริชาร์ดบางส่วน	√			

2.5.2สเปคตรัมที่เกิดขึ้นเนื่องจากแบริ่งเสียหาย (Bearing Faults)

คุณลักษณะของความถี่ที่เกิดขึ้นตามชนิดความเสียหายของแบริ่ง(Bearing) แสดงดังสมการ (1)-(4)

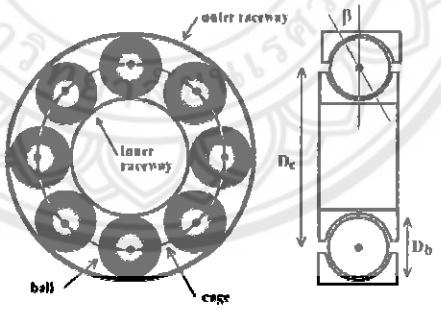
ความเสียหายที่เกิดจากทางด้านนอกของรองลื่น $$f_o = \frac{N_b}{2} f_r\left(1 - \frac{D_b}{D_c}\cos\beta\right) \quad (1)$$

ความเสียหายที่เกิดจากด้านในของรองลื่น $$f_i = \frac{N_b}{2} f_r\left(1 + \frac{D_b}{D_c}\cos\beta\right) \quad (2)$$

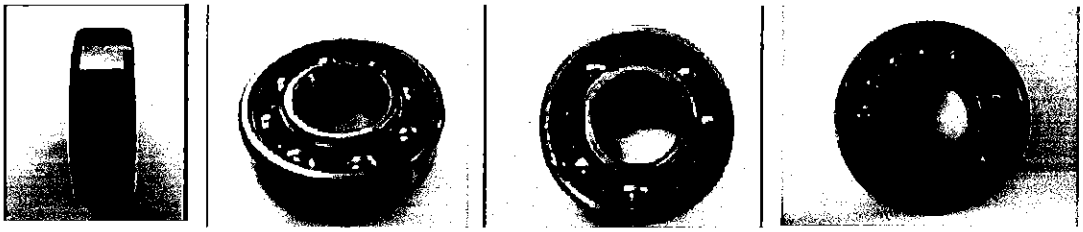
ความเสียหายเนื่องจากลูกปืน $$f_b = \frac{D_c}{D_b} f_r\left(1 - \frac{D_b^2}{D_c^2}\cos^2\beta\right) \quad (3)$$

ความเสียหายเนื่องจากตัวล็อกลูกปืนหลุด $$f_c = \frac{1}{2} f_r\left(1 - \frac{D_b}{D_c}\cos\beta\right) \quad (4)$$

จากรูปที่ 2.7 ค่าพารามิเตอร์ที่ใช้ในการคำนวณหาค่าคุณลักษณะของความถี่ที่เกิดขึ้นเนื่องจากความเสียหาย โดยแบ่งออกเป็น 3 ชนิด ได้แก่ ความเสียหายที่เกิดจากทางด้านนอกของรองลื่น (Outer raceway), ความเสียหายที่เกิดจากด้านในของรองลื่น (Inner raceway) และความเสียเนื่องจากลูกปืน (Ball) ดังแสดงในรูปที่ 2.8



รูป 2.7 แสดงค่าพารามิเตอร์ในการคำนวณหาคุณลักษณะของความถี่ [5]



(a) (b) (c) (d)

รูปที่ 2.8 แสดงการจำลองความเสียหายของแบริ่ง (a) ความเสียหายที่เกิดจากทางด้านนอกของรองลื่น (Outer raceway) (b) ความเสียหายที่เกิดจากด้านในของรองลื่น (Inner raceway) (c) ความเสียเนื่องจากลูกปืน (Ball) (d) ความเสียหายเนื่องจากตัวล็อกลูกปืนหลุด [6]

เมื่อแบริ่ง เสียหาย (Bearing Faults) จะส่งผลให้เกิดความถี่ฮาร์โมนิคที่กระแสสเตเตอร์ ดังสมการ (5)

$$f_{br} = f_s \pm kf_c \tag{5}$$

ซึ่ง f_{br} คือ ความถี่ฮาร์โมนิคที่กระแสสเตเตอร์ เนื่องจากแบริ่งเสียหาย, f_s คือ ความถี่ของแหล่งจ่าย, f_c คือ คุณลักษณะของความถี่ที่เกิดขึ้นตามชนิดความเสียหายของแบริ่ง และ $k = 1, 2, 3, \ldots$ จากความถี่ที่ปรากฏในกระแสสเตเตอร์เมื่อแบริ่งเสียหาย สามารถที่จะตรวจสอบความเสียหายที่เกิดขึ้นก่อนจะเกิดความเสียหายที่รุนแรง

บทที่ 3

ออกแบบการทดลอง

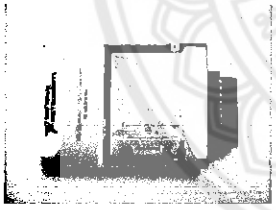
ในบทนี้จะกล่าวถึงวิธีการดำเนินงานศึกษาความผิดปกติของแบริ่ง ในอินดักชั่นมอเตอร์ ด้วยการวิเคราะห์สัญญาณกระแสสเตเตอร์ และสัญญาณเสียง

3.1. อุปกรณ์ที่ใช้ในการทดลอง

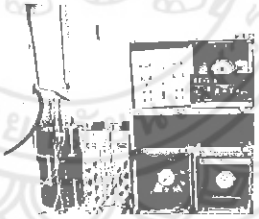
1 อินดักชั่นมอเตอร์ 3 เฟส 220 V 0.4 kW 4 pole 1410r/min

2 CT และตัวต้านทาน 0.1 โอมห์ 3 ชุด

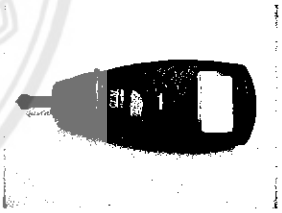
3 มัลติมิเตอร์ (Multimeter)



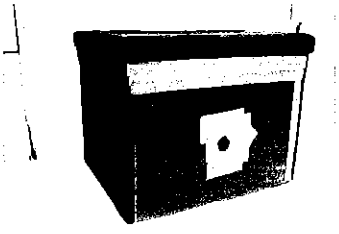
4 เครื่องเก็บค่าสัญญาณกระแส



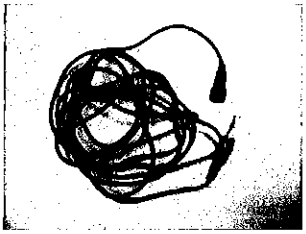
5 แหล่งจ่ายไฟฟ้าและชุดปรับโหลดของมอเตอร์



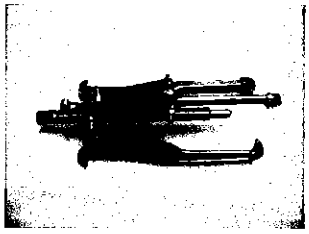
6 เครื่องวัดความเร็วรอบ (Hand Tachometer)



7 กล่องเก็บเสียง



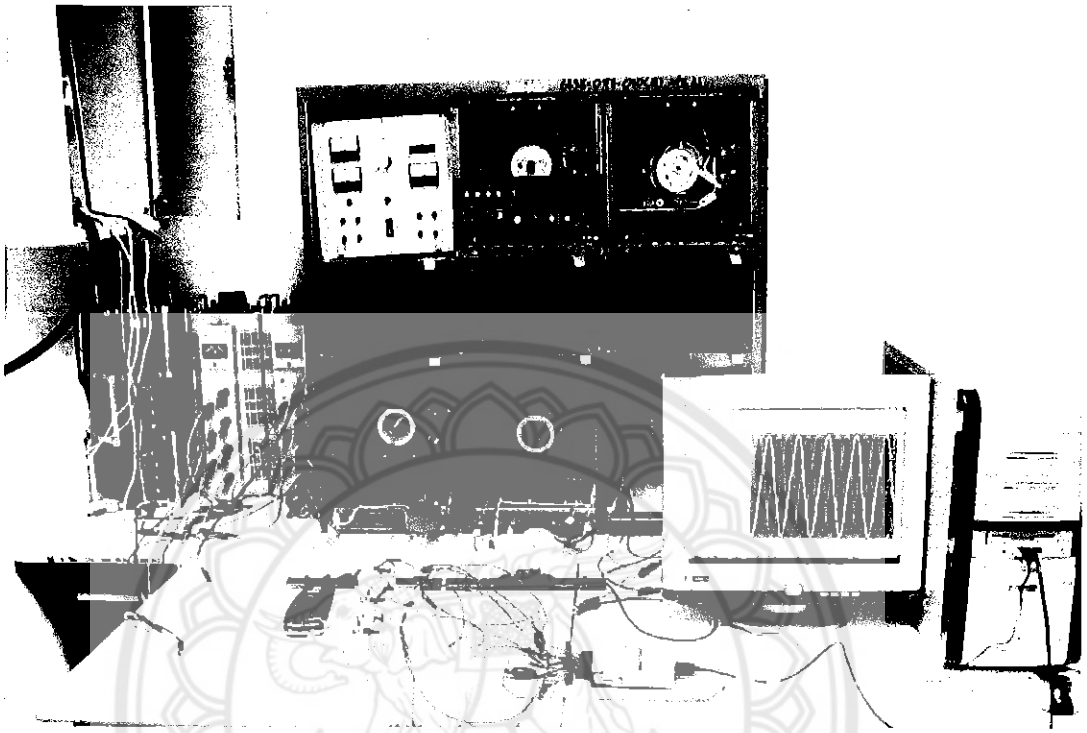
9 ไมโครโฟน Microphone



10.เครื่องมือถอดแบริ่ง

รูปที่ 3.1 แสดงอุปกรณ์ในการทดลอง

3.2. การวิเคราะห์สัญญาณกระแสสเตเตอร์



รูปที่ 3.2 การเก็บค่าสัญญาณกระแสสเตเตอร์

การเก็บค่าสัญญาณกระแส เก็บค่าโดยใช้ CT ลดขนาดของกระแสลงมาเพื่อให้เครื่องเก็บสัญญาณสามารถเก็บสัญญาณกระแสได้ ดังรูปที่ 3.2

3.2.1. วิเคราะห์โดยการแปลงฟูเรียร์ เมื่อได้ข้อมูลจากการทดลองมาแล้ว จะทำการแปลงข้อมูลจากโดเมนเวลามาเป็นโดเมนความถี่ โดยใช้โปรแกรมแมทแลป

เมื่อได้ข้อมูลสัญญาณกระแสเป็นโดเมนความถี่แล้วเราสามารถวิเคราะห์ความผิดปกติของแบริ่งของมอเตอร์ โดยใช้สมการที่ (1) หาตำแหน่งสัญญาณที่เกิดจากความเสียหายของแบริ่งได้ดังตารางที่ 3.1 -- ตารางที่ 3.3

โดยที่ แบริ่งด้านหน้ามีขนาด $D_b = 0.60cm$, $D_c = 2.53cm$, $N_b = 8$

โดยที่ แบริ่งด้านหลังมีขนาด $D_b = 0.56cm$, $D_c = 2.23cm$, $N_b = 7$

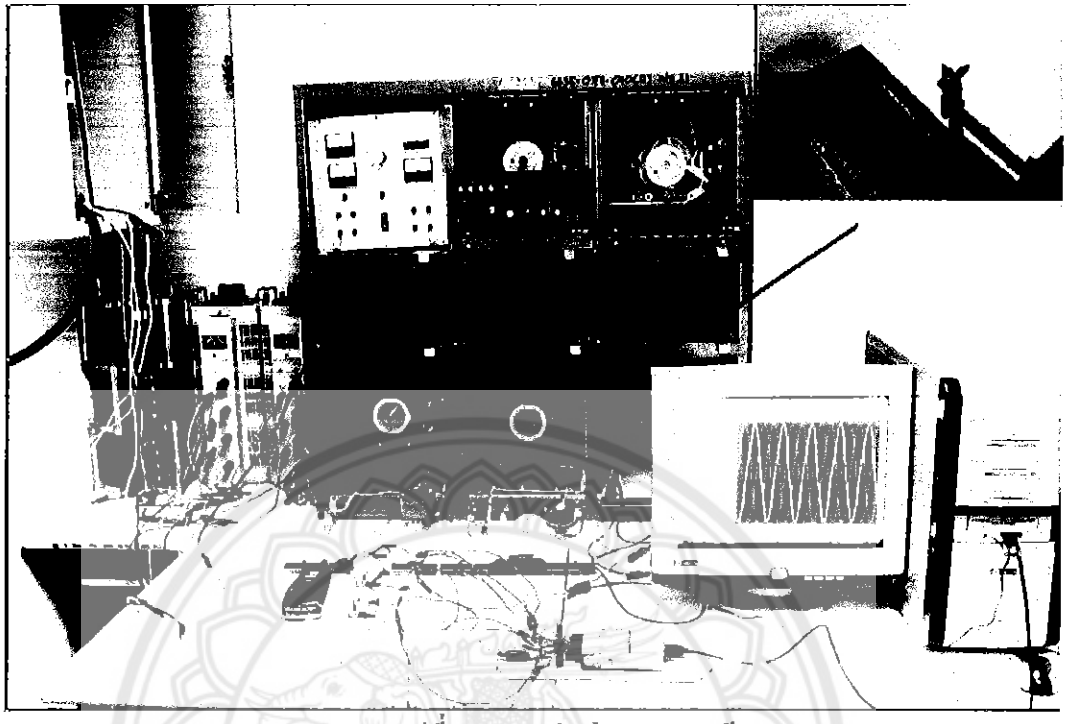
ตารางที่ 3.1 แสดงตำแหน่งสัญญาณที่เกิดขึ้นเมื่อแบริ่งเกิดความเสียหายที่ร่องลื่นด้านนอกขณะมอเตอร์ขับโหลด 100%

	แบริ่งด้านหน้า		แบริ่งด้านหลัง	
k=1	12.74407	75.48814	11.59529	73.19058
k=2	112.7441	138.2322	111.5953	134.7859
k=3	175.4881	200.9763	173.1906	196.3812
k=4	238.2322	263.7204	234.7859	257.9765
k=5	300.9763	326.4644	296.3812	319.5717
k=6	363.7204	389.2085	357.9765	381.167
k=7	426.4644	489.2085	419.5717	481.167

3.2.2. วิเคราะห์โดยเวคเตอร์ปาร์ค เมื่อได้ข้อมูลจากการทดลองมาแล้ว จะทำการแปลงข้อมูลจากโดเมนเวลามาเป็นเวคเตอร์ปาร์คโดยใช้โปรแกรมแมทแลป

เมื่อได้ข้อมูลสัญญาณกระแสเป็นปาร์คเวกเตอร์แล้วก็นำผลที่ได้นำมาวิเคราะห์เปรียบเทียบกันเพื่อวิเคราะห์การเปลี่ยนแปลงของข้อมูลในแต่ละกรณี

3.3. การวิเคราะห์สัญญาณเสียง



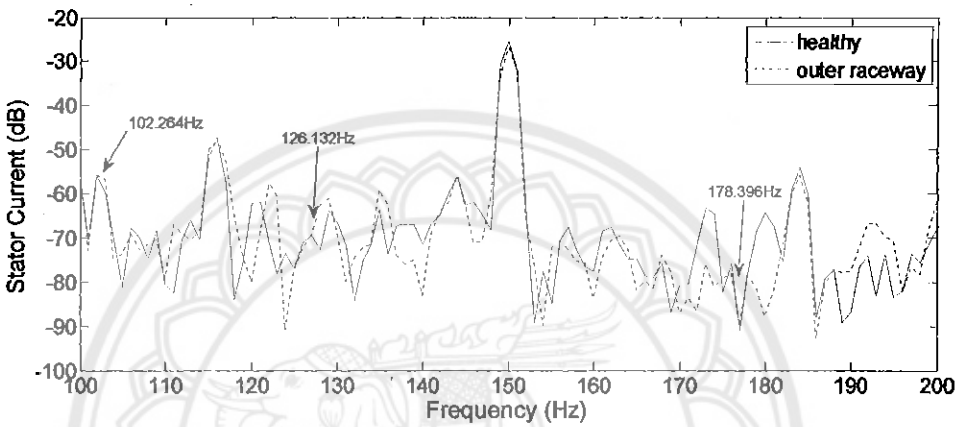
รูปที่ 3.3 การเก็บค่าสัญญาณเสียง

การเก็บสัญญาณเสียงจากรูปที่ 3.3 แสดงการเก็บเสียงโดยใช้ไมโครโฟนต่อเข้ากับคอมพิวเตอร์ และแปลงค่าเป็นสัญญาณความถี่โดยใช้โปรแกรมแลปวิว เนื่องจากโครงงานนี้ต้องการวิเคราะห์สัญญาณของเสียงในโดเมนความถี่เมื่อแบริ่ง เกิดการเสียหายในกรณีต่างๆ จึงจำเป็นต้องใช้กล่องเก็บเสียงเพื่อป้องกันสัญญาณเสียงจากภายนอกมารบกวน โดยกล่องเก็บเสียงนี้ใช้แคมป์เป็นวัสดุในการเก็บเสียง

บทที่ 4

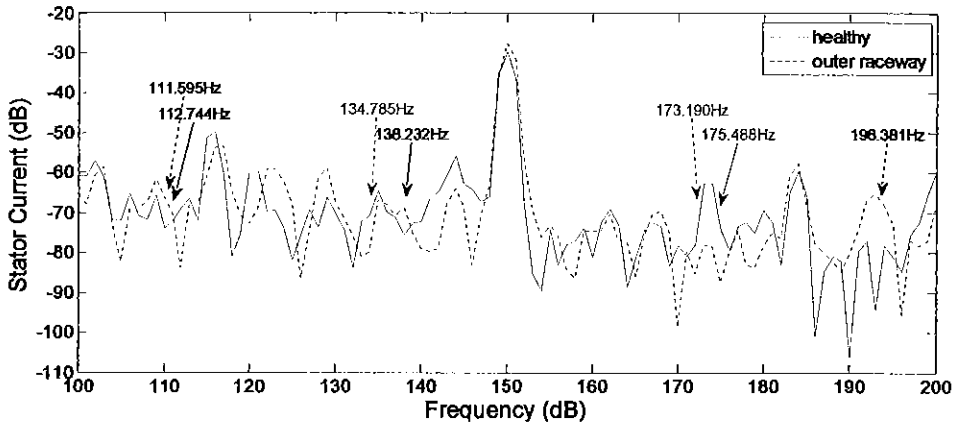
ผลการทดลอง

4.1 การตรวจสอบความผิดพร่องของแบริ่ง โดยวิธีการวิเคราะห์ด้วยการแปลงสัญญาณกระแสฟูเรียร์



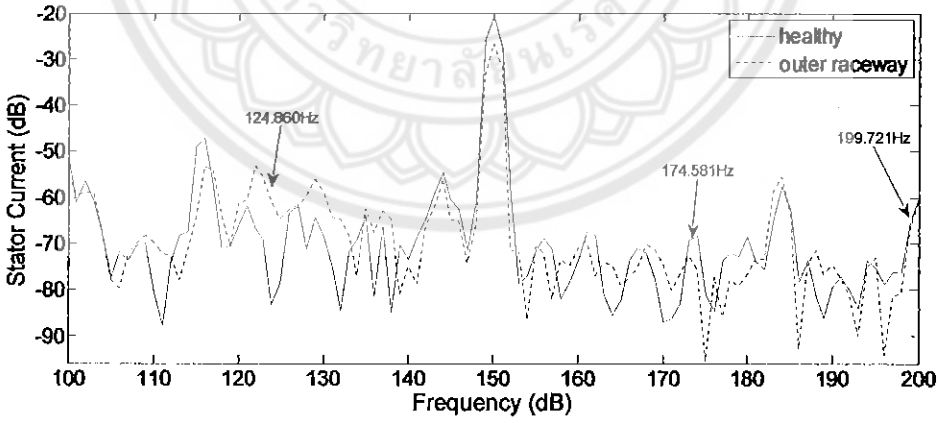
รูปที่ 4.1 แสดงสัญญาณกระแสเมื่อร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลังปกติ

จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลังปกติ ควรเกิดขึ้นที่ความถี่ 102.26Hz 126.132Hz และ178.396Hz จากรูปที่ 4.1 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลังปกติ เกิดขึ้นตำแหน่งที่ความถี่ 102.264Hz และ126.132Hz



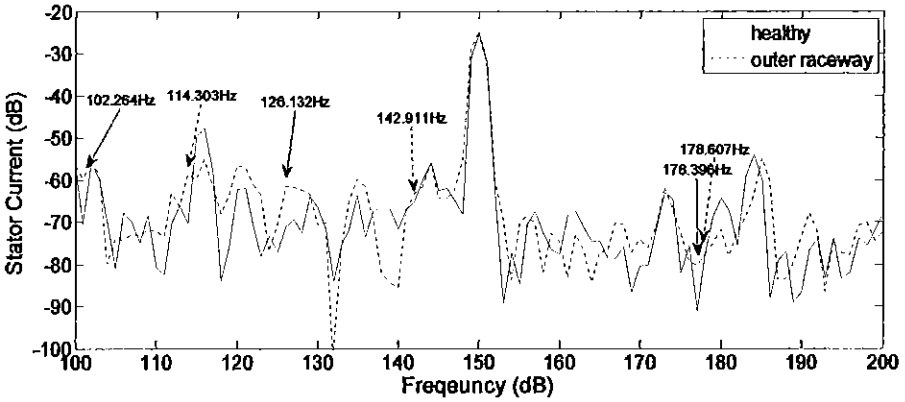
รูปที่ 4.2 แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 1 ที่ ทั้งด้านหน้าและด้านหลัง

จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งเสียหาย 1 ที่ ทั้งด้านหน้าและด้านหลัง โดยแบ่งเป็น ความถี่ที่เกิดขึ้นจากแบริ่งด้านหน้าผิดปกติที่ความถี่ 112.744Hz 138.232Hz และ175.488Hz ความถี่ที่เกิดขึ้นจากแบริ่งด้านหลังผิดปกติที่ความถี่ 111.595Hz 134.785Hz 173.190Hz และ196.381Hz จากรูปที่ 4.2 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ทั้งด้านหน้าและด้านหลัง เกิดขึ้นตำแหน่งที่ความถี่ 111.595Hz 112.744Hz 138.232Hz 196.381Hz



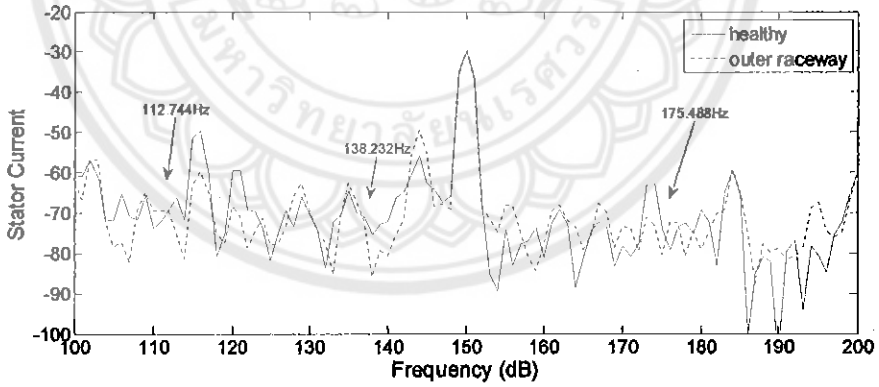
รูปที่ 4.3. *แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 2 ที่ ด้านหลังปกติ*

จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 2 ที่ ด้านหลังปกติ ควรเกิดขึ้นที่ความถี่ 124.860Hz 174.581Hz และ199.721Hz จากรูปที่ 4.3 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 2 ที่ ด้านหลังปกติ เกิดขึ้นเพียงตำแหน่งที่ความถี่ 124.860Hz



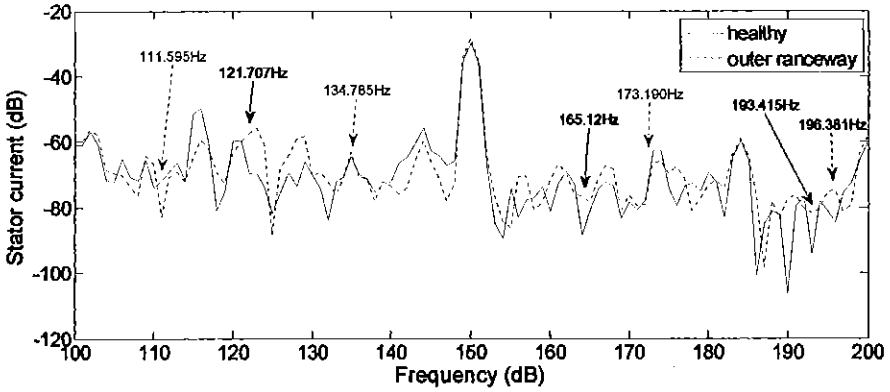
รูปที่ 4.4. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลัง

จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลัง โดยแบ่งเป็น ความถี่ที่เกิดขึ้นจากแบริ่งด้านหน้าผิดปกติที่ความถี่ 102.264Hz 126.132Hz และ178.396Hz ความถี่ที่เกิดขึ้นจากแบริ่งด้านหลังผิดปกติที่ความถี่ 114.303Hz 142.911Hz และ178.607Hz จากรูปที่ 4.4 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลังเกิดขึ้นทุกตำแหน่งจากการคำนวน



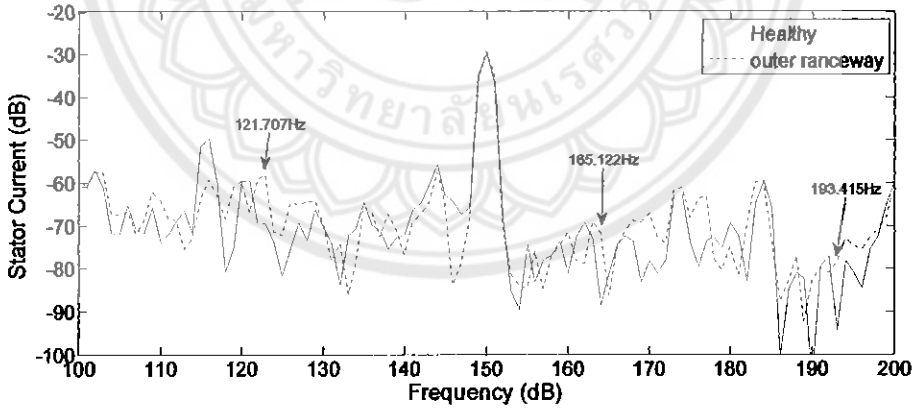
รูปที่ 4.5. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 3 ที่ ด้านหลังปกติ

จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 3 ที่ ด้านหลังปกติ ควรเกิดขึ้นที่ความถี่ 112.744Hz 138.232Hz และ175.488Hz จากรูปที่ 4.5 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 3 ที่ ด้านหลังปกติ เกิดขึ้นเพียงตำแหน่งที่ความถี่ 112.744Hz และ 175.488Hz



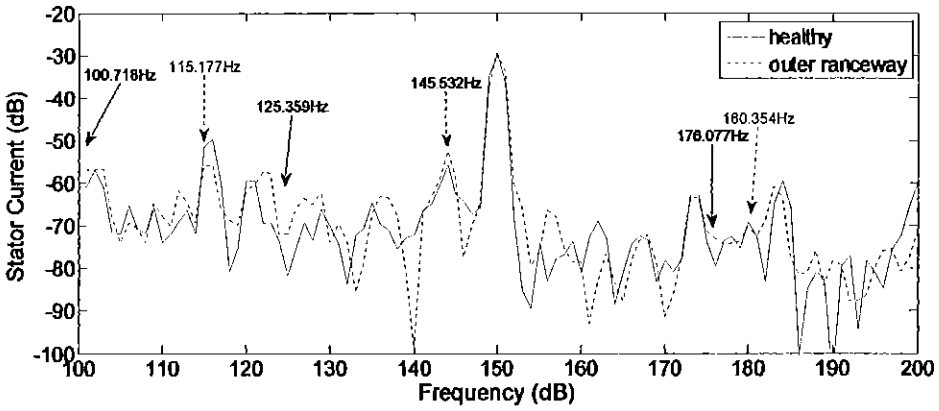
รูปที่ 4.6. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ทั้งด้านหน้าและด้านหลัง

จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ทั้งด้านหน้าและด้านหลัง โดยแบ่งเป็น ความถี่ที่เกิดขึ้นจากแบริ่งด้านหน้าผิดปกติที่ความถี่ 121.707Hz 165.12Hz และ193.415Hz ความถี่ที่เกิดขึ้นจากแบริ่งด้านหลังผิดปกติที่ความถี่ 111.595Hz 134.785Hz 173.190Hz และ196.381 จากรูปที่ 4.6 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ทั้งด้านหน้าและด้านหลัง เกิดขึ้นที่ความถี่ *121.707Hz 165.12Hz 193.415Hz และ196.38Hz*



รูปที่ 4.7. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ

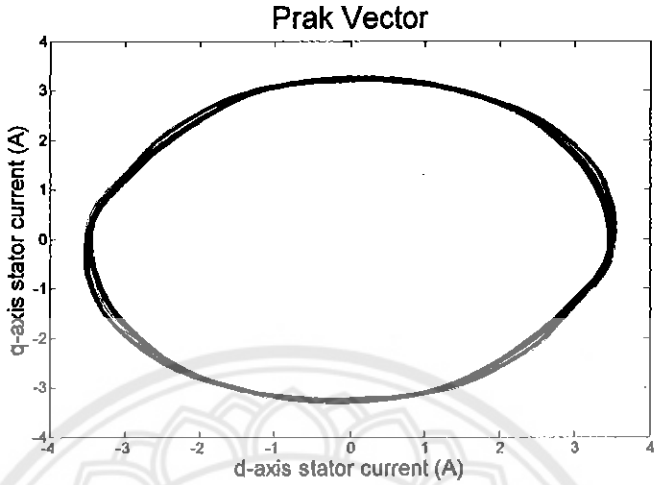
จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ ควรเกิดขึ้นที่ความถี่ 121.707Hz 165.122Hz และ193.415Hz จากรูปที่ 4.7 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ เกิดขึ้นทุกตำแหน่งจากที่คำนวนได้



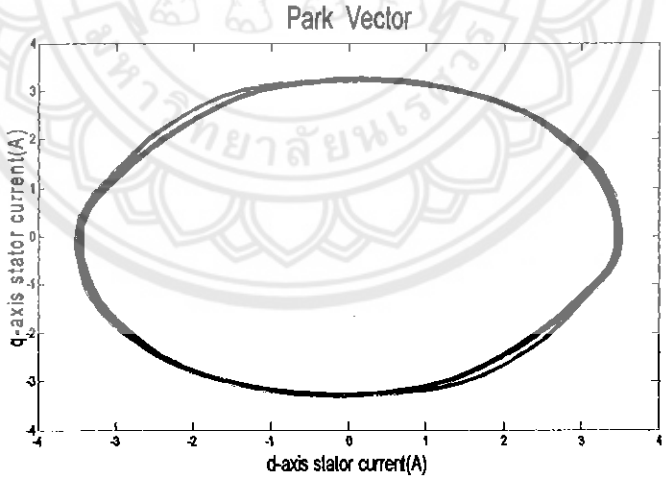
รูปที่ 4.8. แสดงสัญญาณกระแสร่องลื่นด้านนอกของแบริ่งเสียหาย 4 ที่ ทั้งด้านหน้าและด้านหลัง

จากตารางที่ 3.1 สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลัง โดยแบ่งเป็น ความถี่ที่เกิดขึ้นจากแบริ่งด้านหน้าผิดปกติที่ความถี่ 100.718Hz 125.359Hz และ176.077Hz ความถี่ที่เกิดขึ้นจากแบริ่งด้านหลังผิดปกติที่ความถี่ 115.177Hz 145.532Hz และ180.354Hz จากรูปที่ 4.8 แสดงให้เห็นว่าในการทดลองจริง สัญญาณกระแสสเตเตอร์ของมอเตอร์ที่ร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลังเกิดขึ้นที่ตำแหน่ง 100.781Hz 125.359Hz และ176.077Hz

4.2 การตรวจสอบความผิดพร่องของแบริ่ง โดยวิธีการวิเคราะห์การแปลงสัญญาณกระแสเวคเตอร์ปาร์ค

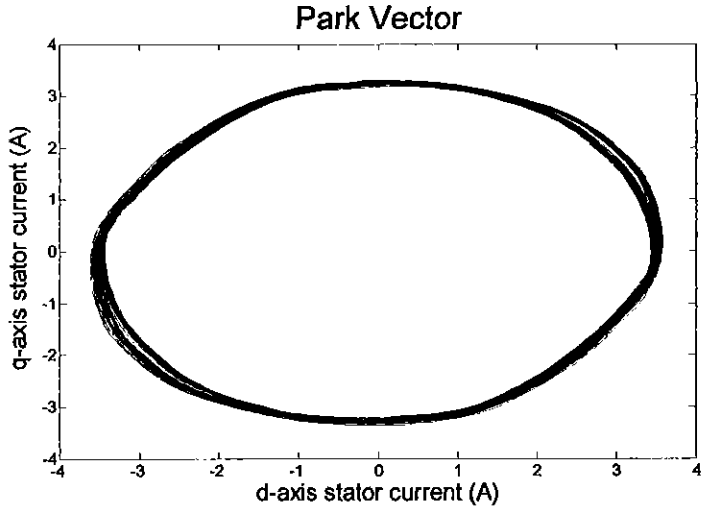


รูปที่ 4.9 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลังปกติ (เส้นสีแดง)

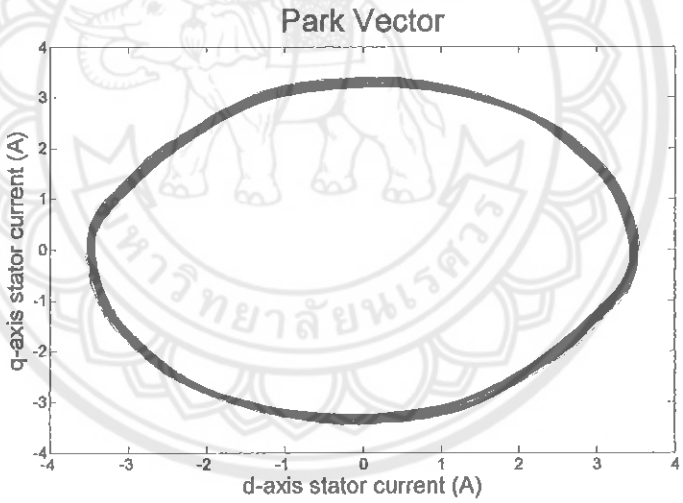


1574042 3
2/5.
D/887
2853

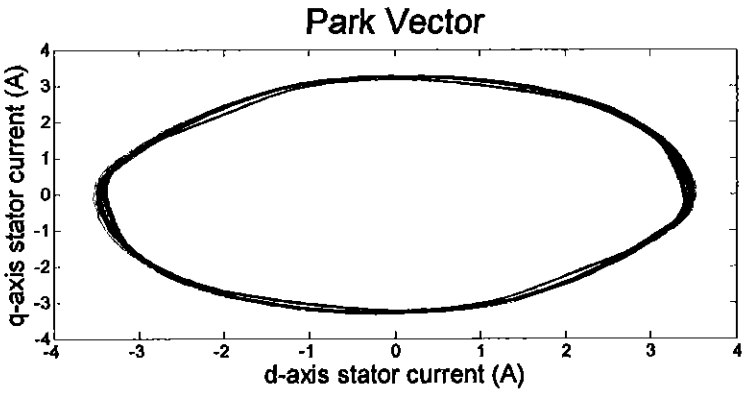
รูปที่ 4.10 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลื่นด้านนอกของแบริ่งเสียหาย 1 ที่ ทั้งด้านหน้าและด้านหลัง (เส้นสีแดง)



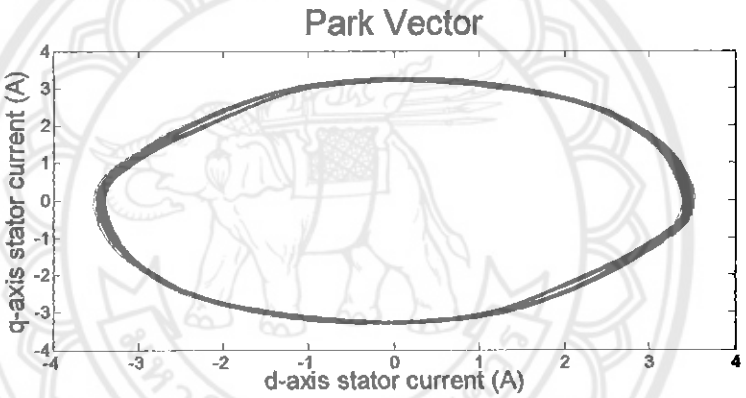
รูปที่ 4.11 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลื่นด้านนอกของแบริ่ง ด้านหน้าเสียหาย 2 ที่ ด้านหลังปกติ (เส้นสีแดง)



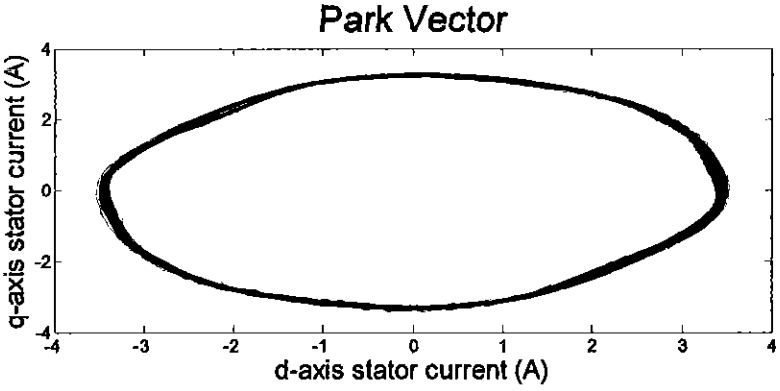
รูปที่ 4.12 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลื่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลัง (เส้นสีแดง)



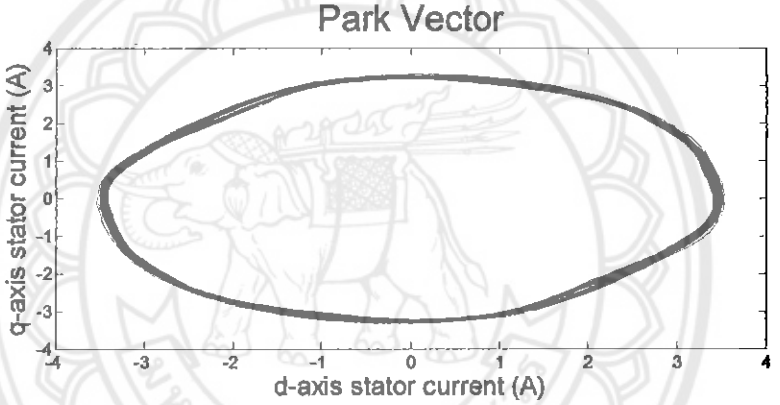
รูปที่ 4.13 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 3 ที่ ด้านหลังปกติ (เส้นสีแดง)



รูปที่ 4.14 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ทั้งด้านหน้าและด้านหลัง (เส้นสีแดง)



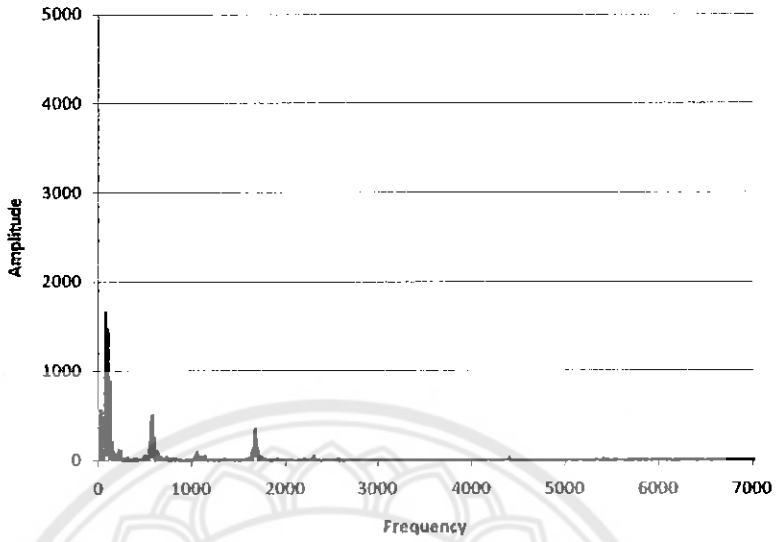
รูปที่ 4.15 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลึ่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ (เส้นสีแดง)



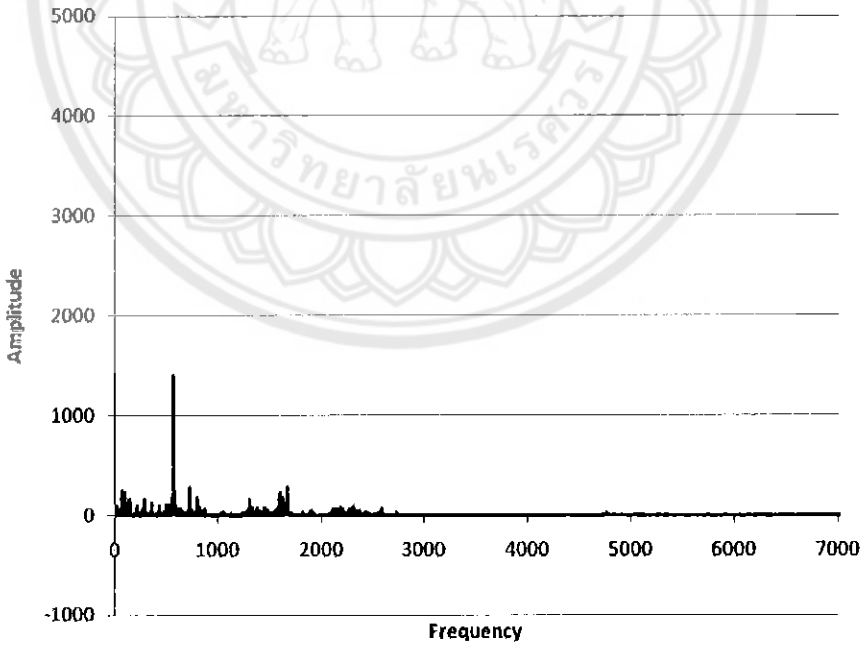
รูปที่ 4.16 แสดงการเปรียบเทียบแบริ่งปกติ (เส้นสีน้ำเงิน) กับร่องลึ่นด้านนอกของแบริ่งเสียหาย 4 ที่ ทั้งด้านหน้าและด้านหลัง (เส้นสีแดง)

จากรูปที่ 4.9 – 4.16 เมื่อแบริ่งมีความเสียหายที่ร่องลึ่นด้านนอก เมื่อนำมาวิเคราะห์ด้วยการใช้วิธีเวคเตอร์ปาร์ค จะเห็นได้ว่ารูปกราฟของแบริ่งที่มีความเสียหายที่ร่องลึ่นด้านนอกจะมีการเปลี่ยนรูปทรงไปจากเดิมอย่างเห็นได้ชัด โดยตำแหน่งของความเสียหายไม่มีผลต่อการเปลี่ยนแปลงของรูปกราฟมากนัก

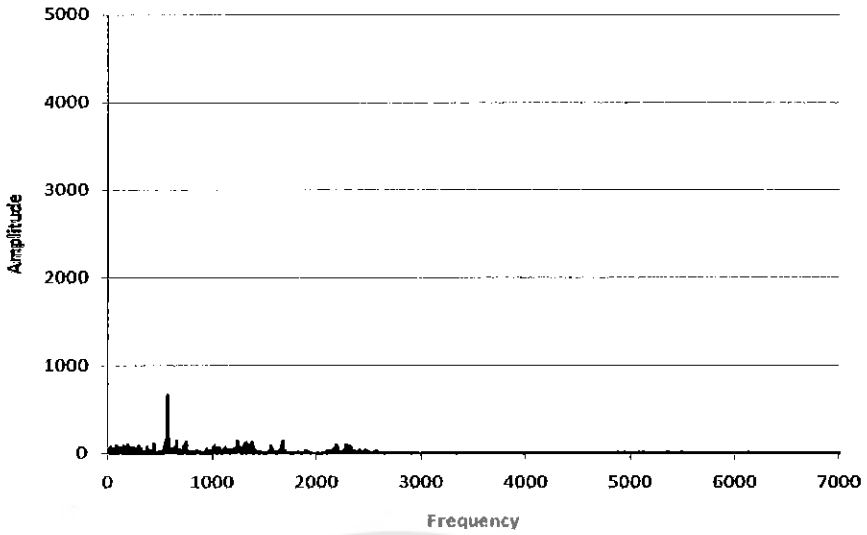
4.3 ตรวจสอบความผิดพร่องของแบริ่ง โดยวิธีการวิเคราะห์สัญญาณเสียง



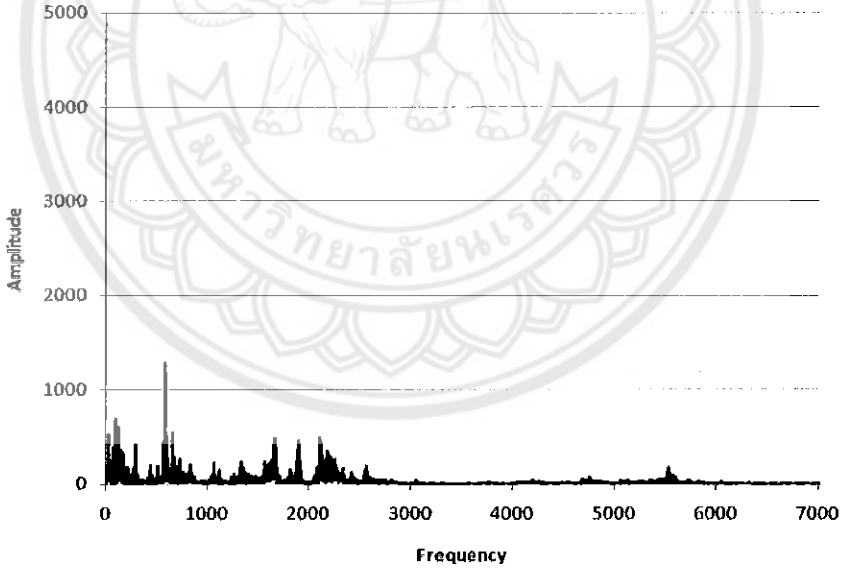
รูปที่ 4.17 แสดงสัญญาณเสียงเมื่อแบริ่งปกติ



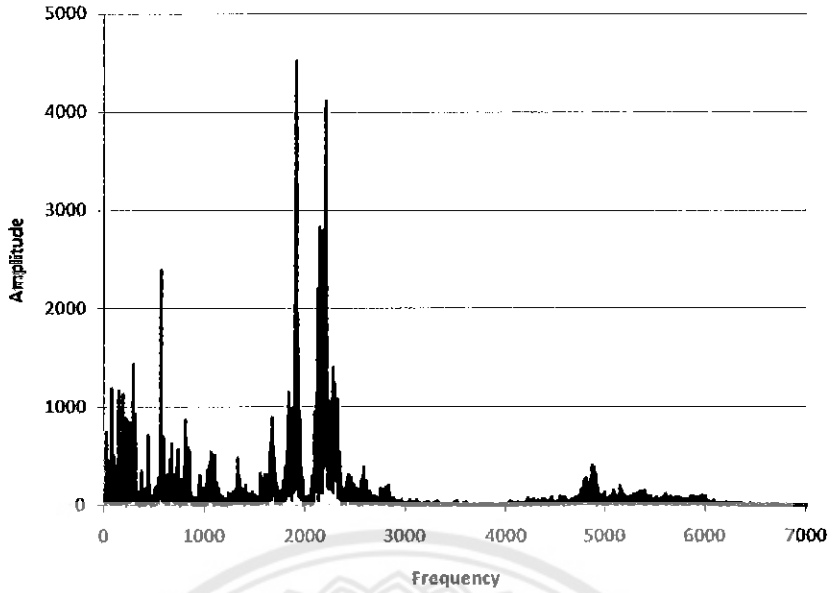
รูปที่ 4.18 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 1 ที่ ด้านหลัง ปกติ



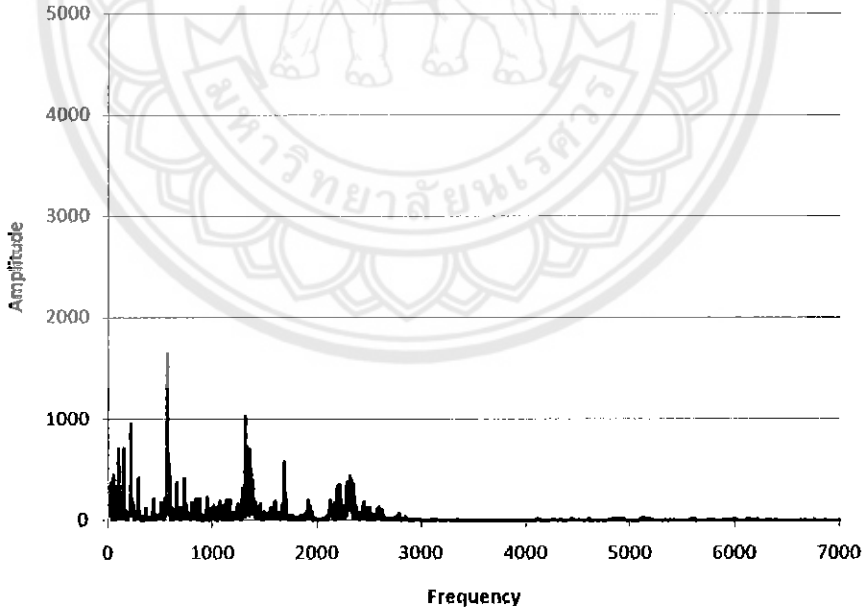
รูปที่ 4.19 แสดงสัญญาณเสียงเมื่อร่องลึ่นด้านนอกของแบริ่งเสียหาย 1 ที่ ทั้งด้านหน้าและด้านหลัง



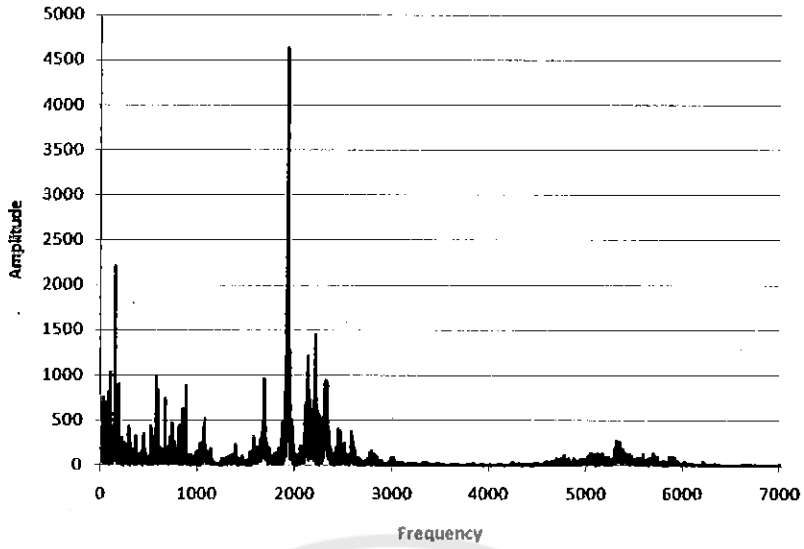
รูปที่ 4.20 แสดงสัญญาณเสียงเมื่อร่องลึ่นด้านนอกของแบริ่งด้านหน้าเสียหาย 2 ที่ด้านหลังปกติ



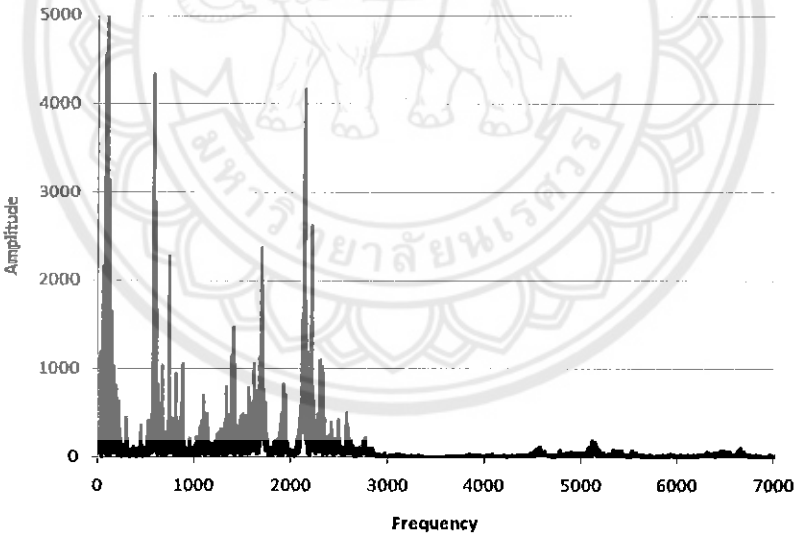
รูปที่ 4.21 แสดงสัญญาณเสียงเมื่อร่องลึ่นด้านนอกของแบริ่งเสียหาย 2 ที่ ทั้งด้านหน้าและด้านหลัง



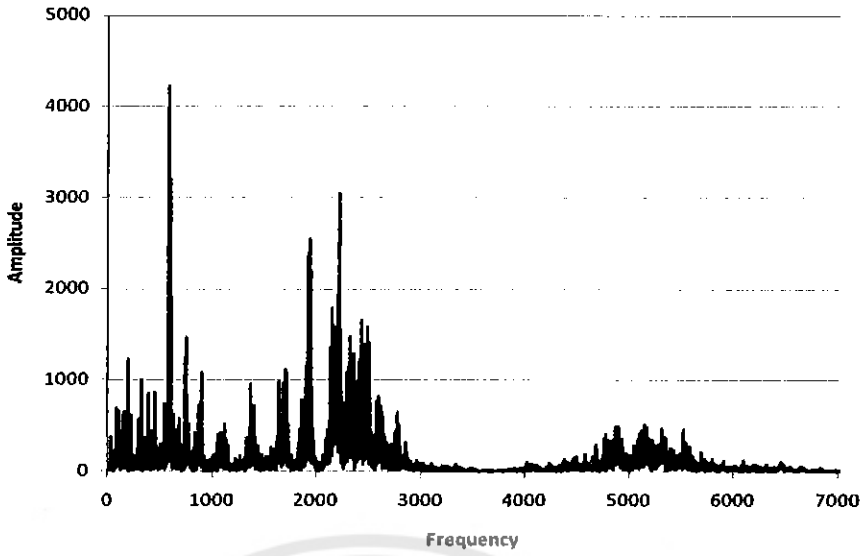
รูปที่ 4.22 แสดงสัญญาณเสียงเมื่อร่องลึ่นด้านนอกของแบริ่งด้านหน้าเสียหาย 3 ที่ด้านหลังปกติ



รูปที่ 4.23 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งเสียหาย 3 ที่ ทั้งด้านหน้าและด้านหลัง



รูปที่ 4.24 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ด้านหลังปกติ

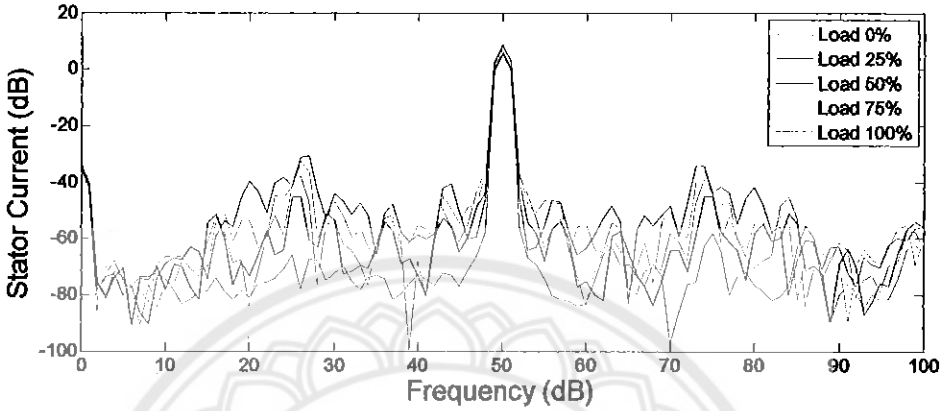


รูปที่ 4.25 แสดงสัญญาณเสียงเมื่อร่องลื่นด้านนอกของแบริ่งเสียหาย 4 ที่ ทั้งด้านหน้าและด้านหลัง

จากรูปที่ 4.17 – รูปที่ 4.25 ร่องลื่นด้านนอกของแบริ่งเสียหาย เมื่อนำมาเปรียบเทียบกับสัญญาณเสียงของมอเตอร์ที่มีแบริ่งปกติรูปที่ 4.22 จะพบว่าเมื่อร่องลื่นด้านนอกของแบริ่งเสียหายจะพบสัญญาณเสียงที่ความถี่ที่ตำแหน่งประมาณ 2000Hz – 3000Hz และความถี่ที่ตำแหน่งประมาณ 4000Hz – 7000Hz โดยยิ่งมีความเสียหายมากขึ้น Amplitude บริเวณตำแหน่งดังกล่าว จะสูงขึ้น และสัญญาณช่วงความถี่ที่ 4000Hz – 7000Hz จะเกิดกับกรณีที่แบริ่งด้านหลังมีความเสียหายด้วย

4.4. ผลที่เกิดจากโหลด

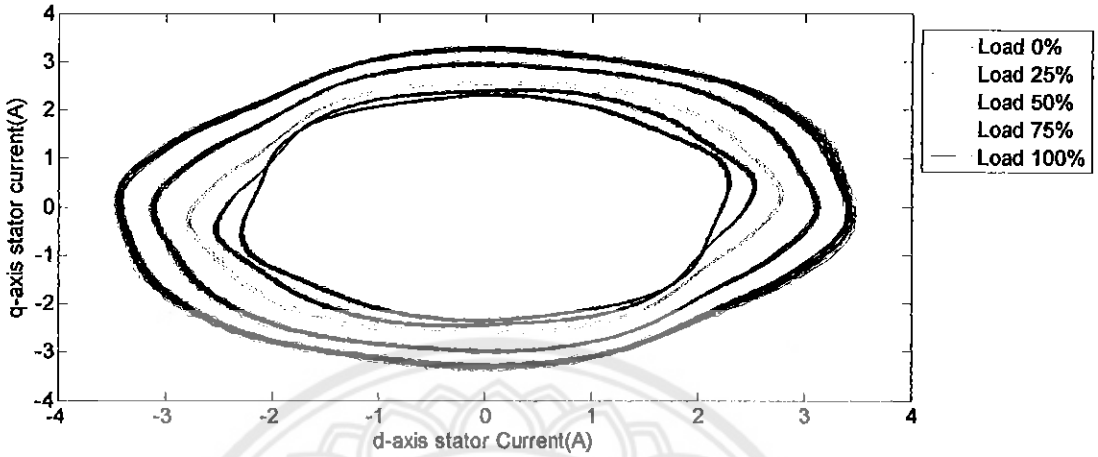
4.4.1. ผลที่เกิดจากโหลดของการแปลงสัญญาณกระแสฟูเรียร์



รูปที่ 4.26 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลัง**ปกติ**

จากรูปที่ 4.26 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลัง**ปกติ** จะพบว่า**ภาระโหลดมีผลต่อสัญญาณกระแสฟูเรียร์** กล่าวคือยิ่งมอเตอร์มีภาระโหลดมาก**ขึ้นสัญญาณกระแส**ในแต่ละความถี่จะมีสูงขึ้น ทำให้การตรวจจับกระแสที่เกิดจากความผิดพร่องของ**แบริ่งทำได้ง่ายขึ้น** เหตุที่เป็นแบบนี้เพราะเมื่อมอเตอร์รับโหลดมากขึ้น ทำให้แบริ่งรับน้ำหนักมาก**ขึ้นตามไปด้วย** สัญญาณตำแหน่งที่เกิดจากความผิดพร่องของแบริ่งจึงเห็นได้ชัดเจนขึ้น

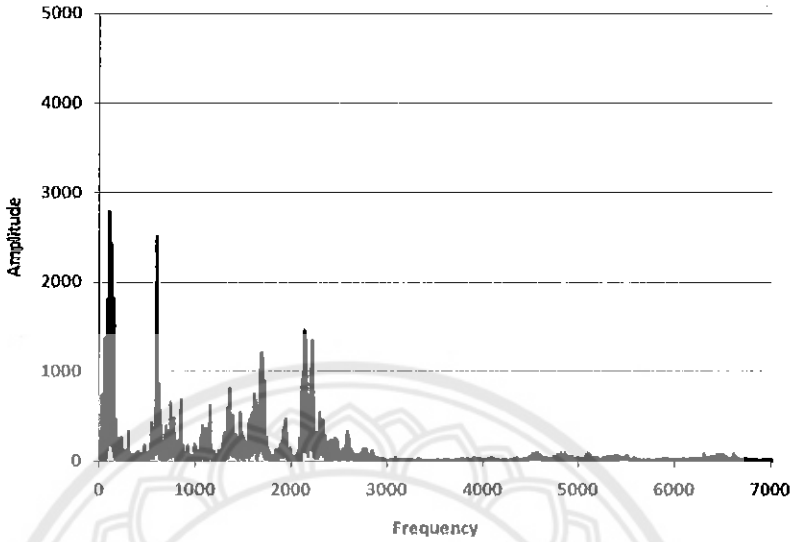
4.4.2. ผลที่เกิดจากโหลดของการแปลงสัญญาณกระแสเวคเตอร์ปาร์ค



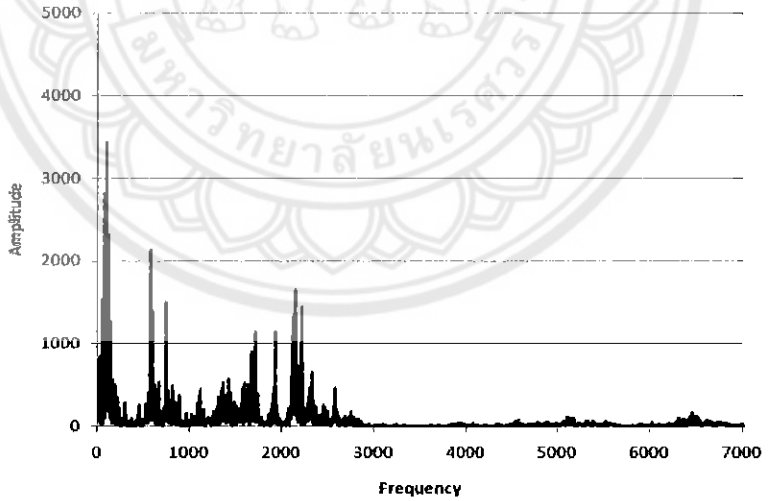
รูปที่ 4.27 ร่องลึ่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ

จากรูปที่ 4.27 ร่องลึ่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ จะพบว่า ภาระโหลดมีผลต่อรูปกราฟสัญญาณกระแสเวคเตอร์ปาร์ค กล่าวคือยิ่งมอเตอร์มีภาระโหลดมากขึ้น รูปกราฟจะวงใหญ่ และหนาขึ้น ทำให้สามารถวิเคราะห์ความผิดปกติของแบริ่งได้ง่ายขึ้น

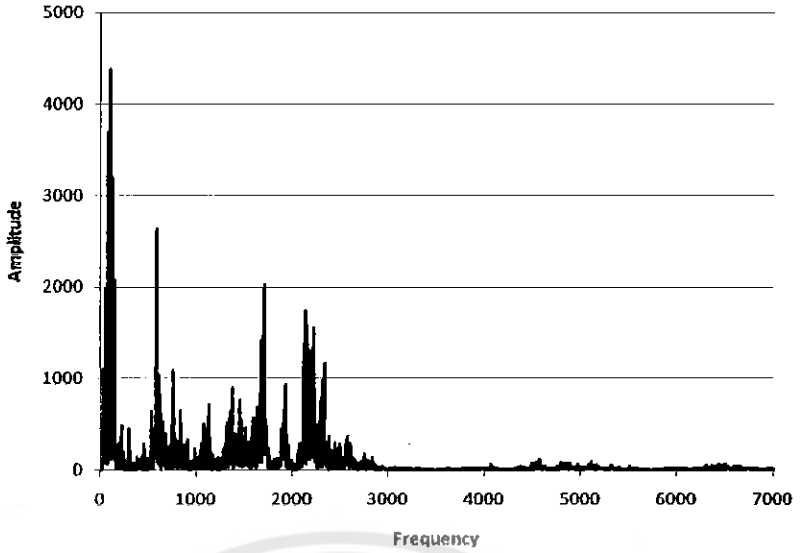
4.4.2. ผลที่เกิดจากโหลดของการแปลงสัญญาณเสียง



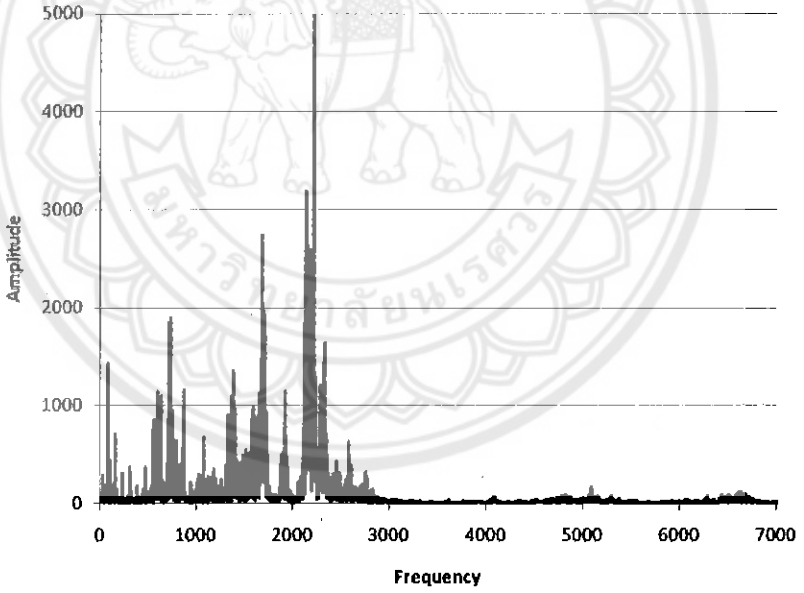
รูปที่ 4.28 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 0%



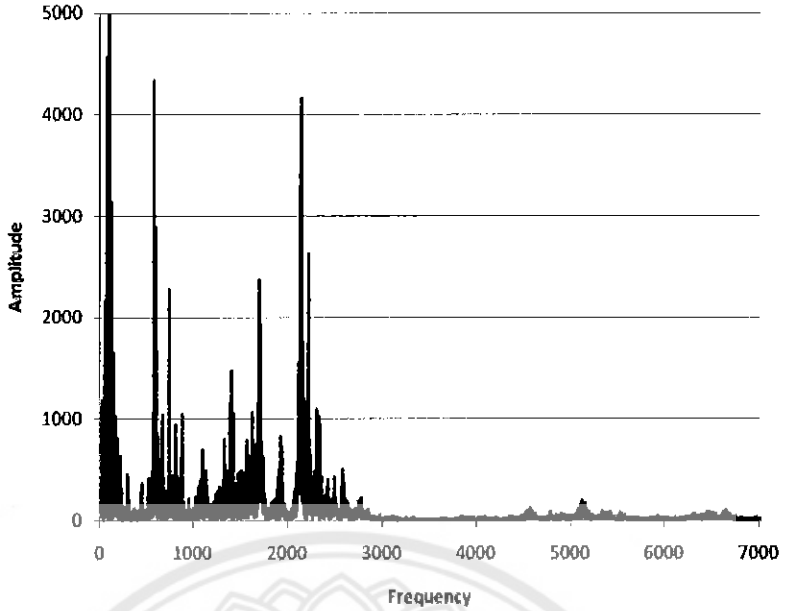
รูปที่ 4.29 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 25%



รูปที่ 4.30 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ **โหลด 50%**



รูปที่ 4.31 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 75%



รูปที่ 4.32 ร่องลื่นด้านนอกของแบริ่งด้านหน้าเสียหาย 4 ที่ ด้านหลังปกติ โหลด 100%

จากรูปที่ 4.28 – รูปที่ 4.32 จะเห็นได้ว่า เมื่อมอเตอร์มีภาระโหลดเพิ่มขึ้นสัญญาณเสียงบริเวณที่เกิดเนื่องจากความผิดปกติของแบริ่งจะมีแนวโน้มเพิ่มสูงขึ้นเนื่องจากเมื่อมอเตอร์รับภาระโหลดมากยิ่งขึ้น แรงกระทำที่มีในแบริ่งก็จะเพิ่มสูงขึ้นด้วย ทำให้เกิดเสียงดังขึ้น

บทที่ 5
สรุปผลการทดลอง และข้อเสนอแนะ

5.1. สรุปผลการทดลอง

การวิเคราะห์ความผิดพร่องของแบริ่งโดยใช้การแปลงสัญญาณกระแสแบบฟูเรียร์ พบสัญญาณกระแสสเตเตอร์ที่เกิดจากความเสียหายของแบริ่งในกรณีต่างๆ ตามที่คำนวณได้ ไม่ครบทุกตำแหน่ง เนื่องมาจากสัญญาณรบกวนที่มาจากแหล่งจ่ายไฟฟ้า หรืออุปกรณ์ การวิเคราะห์ความผิดพร่องของแบริ่งโดยใช้การแปลงสัญญาณกระแสแบบฟูเรียร์ จึงวิเคราะห์ความผิดพร่องของแบริ่งได้ในระดับหนึ่งเท่านั้น

การวิเคราะห์ความผิดพร่องของแบริ่งโดยใช้การแปลงสัญญาณกระแสเวคเตอร์ปาร์ค เห็นผลค่อนข้างชัดเจนเมื่อแบริ่งเกิดความเสียหายที่ร่องลื่นด้านนอกรูปกราฟจะเปลี่ยนไปอย่างเห็นได้ชัดเมื่อเปรียบเทียบกับผลของสัญญาณจากแบริ่งปกติ เป็นการวิเคราะห์ที่น่าสนใจมากเนื่องจากนำกระแสสเตเตอร์มาวิเคราะห์พร้อมกันทั้ง 3 เฟส

การวิเคราะห์ความผิดพร่องของแบริ่งโดยการวิเคราะห์สัญญาณเสียงเป็นการวิเคราะห์ที่แม่นยำ และชัดเจนที่สุด เมื่อแบริ่งมีความเสียหายที่ร่องลื่นด้านนอกจะมีสัญญาณเสียงเกิดขึ้นในย่านความถี่ ประมาณ 2000Hz – 3000Hz และย่านความถี่ประมาณ 4000Hz – 7000Hz เนื่องจากการที่แบริ่งเกิดความเสียหายย่อมเกิดเสียงขึ้น เพราะแบริ่งขณะใช้งานต้องมีการเสียดสีระหว่างร่องลื่นกับลูกปืนตลอดเวลา

ผลจากโหลด จากการทดลองจะเห็นได้ว่า การวิเคราะห์ความผิดพร่องของแบริ่งโดยใช้การแปลงสัญญาณกระแสแบบฟูเรียร์ การวิเคราะห์ความผิดพร่องของแบริ่งโดยใช้การแปลงสัญญาณกระแสเวคเตอร์ปาร์ค และการวิเคราะห์ความผิดพร่องของแบริ่งโดยการวิเคราะห์สัญญาณเสียง จะเห็นผลชัดเจนที่สุดเมื่อมอเตอร์ทำงานที่โหลด 100%

การวิเคราะห์ความผิดพร่องของแบริ่งให้แม่นยำที่สุดควรใช้การวิเคราะห์ทั้ง 3 วิธีควบคู่กันไป และควรเก็บข้อมูลขณะที่มอเตอร์ขับโหลดสูงๆ

5.2. ปัญหาที่พบขณะดำเนินโครงงาน

5.2.1. ในการทดสอบแต่ละกรณีการเสียหายของแบริ่ง ต้องมีการถอดเปลี่ยนแบริ่งบ่อยมาก จึงทำให้บางครั้งแบริ่งเกิดการแตกหักเสียหาย หรือถอดไม่ออก ทำให้การทดลองล่าช้าออกไป

5.2.2. ในการทดสอบที่ต่อเนื่องทำให้มอเตอร์ร้อนมาก ทำให้ไม่สามารถทดลองต่อเนื่องได้ จึงทำให้การทดลองล่าช้าออกไป

5.3. ข้อเสนอแนะ

5.3.1. ควรระมัดระวังในการทดลองอาจเกิดอัตรายได้

5.3.2. เนื่องจากการทดลองนี้มีข้อมูลการทดลองเป็นจำนวนมาก จึงควรเก็บผลการทดลองให้เป็นระเบียบ เพื่อง่ายต่อการทำมาใช้ และไม่ให้ข้อมูลสับเปลี่ยนกันได้

5.3.3. ควรมีการวางแผนในการทำงานที่ดี เพื่อความรวดเร็วและแม่นยำในการทดลอง

เอกสารอ้างอิง

[1] "**อินดักชั่นมอเตอร์สามเฟส (Three phase motors)**" [ออนไลน์] เข้าถึงได้จาก http://eestud.kku.ac.th/~u3868103/high%30motor.htm (วันที่สืบค้นข้อมูล 15 สิงหาคม พ.ศ. 2553)

[2] ศุภชัย อรุณพันธ์1) สาวิตร์ ตัณฑนุช2)และสุระพล เธียรมนตรี "**การวิเคราะห์สัญญาณฮาร์มอนิกส์กระแสเพื่อศึกษาความผิดพร่องของมอเตอร์เหนี่ยวนำ 3-เฟส โดยการใช้เครื่องมือวัดเสมือนด้วยโปรแกรมLabVIEW**" วิศวกรรมสาร มข. ปีที่ 34 ฉบับที่ 5 หน้า 3

[3] "**แบริ่ง(Bearing)**" [ออนไลน์] เข้าถึงได้จาก http://www.tsquare-lube.com/index.php?lay=show&ac=article&Ntype=5 (วันที่สืบค้นข้อมูล 15 สิงหาคม พ.ศ. 2553)

[4] Frosini, L. and others. (2008). **Use of the stator current for condition monitoring of bearings in induction motors.** Retrieved December 20, 2010, from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4799991

[5] Blodt, M. and others. (2008). Models for Bearing Damage Detection in Induction Motors Using Stator Current Monitoring. **IEEE Transactions on Industrial Electronics,** 55(4), 1813 - 1822.

[6] Onel, I.Y. and El Hachemi Benbouzid, M. (2008). Induction Motor Bearing Failure Detection and Diagnosis: Park and Concordia Transform Approaches Comparative Study. **IEEE/ASME Transactions on Mechatronics,** 13(2), 257 - 262.

ภาคผนวก

การเขียนโปรแกรม LabVIEWเบื้องต้น

ในบทนี้เราจะเรียนรู้ถึงวิธีการเขียนโปรแกรมและวิธีการแก้ไขโปรแกรมใน LabVIEW เบื้องต้น รวมถึงการเรียนรู้ถึง Controls และ Indicators แบบต่างๆ รวมถึงวิธีการเลือกและความหมายของตัวเลือกแบบต่างๆ สำหรับ Controls และ Indicators แต่ละแบบ นอกเหนือจากนั้นในบทนี้เราจะได้เรียนรู้ถึงวิธีการต่อสายในรายละเอียด รวมถึงการแก้ไขการทำงานหรือตรวจสอบการทำงานของ VI เมื่อเสร็จสิ้นบทนี้เราควรจะสามารถเขียน VI แบบง่ายๆ ได้อย่างมีประสิทธิภาพ กล่าวโดยสรุป ในบทนี้เรามีจุดประสงค์ที่จะทำความเข้าใจในเรื่องต่อไปนี้

สำหรับในส่วนแรกนี้เราจะทำความรู้จักกับวิธีการเขียนโปรแกรมเริ่มต้น โดยใช้กิจกรรมที่ 1 เป็นการอธิบายไปพร้อมๆกัน เพราะเราเชื่อว่าคุณจะสามารถเข้าใจการใช้เครื่องมือต่างๆ บน Tools Palette ได้ดียิ่งขึ้นหากมีการฝึกทำไปพร้อมๆ กันกับการอธิบาย

ในส่วนหลังจะเป็นการอธิบายวิธีการตรวจสอบ การแก้ไข VI และการสร้าง SubVI เพื่อให้เราสามารถนำ VI ที่เราเขียนนี้ไปใช้ใน VI อื่นได้เมื่อเราต้องการ

1. Creating VI

ในหัวข้อแรกนี้ จะเป็นการเสนอรายละเอียดเบื้องต้นเกี่ยวกับการเขียน VI การวางรูปแบบและการปรับเปลี่ยนลักษณะและขนาดของ VI เหล่านั้น การใช้เครื่องมือต่างๆบน Tools Palette การใช้ปุ่มบางแบบบน Toolbar และวิธีการต่อสายใน Block Diagram

กิจกรรมนี้จะเป็นการอธิบายการทำงานของเครื่องมือต่างๆ เราจะอธิบายด้วยการสร้าง VI แบบง่ายๆขึ้น คล้ายกับกิจกรรมสุดท้ายของบทที่ผ่านมา แต่จะเพิ่มรายละเอียดของการใช้อุปกรณ์ต่างๆบน Tools Palette ให้มากยิ่งขึ้น กิจกรรมมีลำดับขั้นตอนต่อไปนี้

1. เปิด VI ใหม่ เลือก New VI
2. ทำให้ Front Panel อยู่ในสภาพพร้อมใช้งานในขณะนี้ Controls Palette ควรจะปรากฏให้เห็นด้วย ถ้าไม่ปรากฏให้เห็นใช้คำสั่ง Show Controls Palette ภายใต้ Windows Menu
3. เลื่อนลูกศรไปบนปุ่มต่างๆ บน Controls Palette สังเกตดูการเปลี่ยนชื่อของ subpalette ต่างๆ ด้านบน

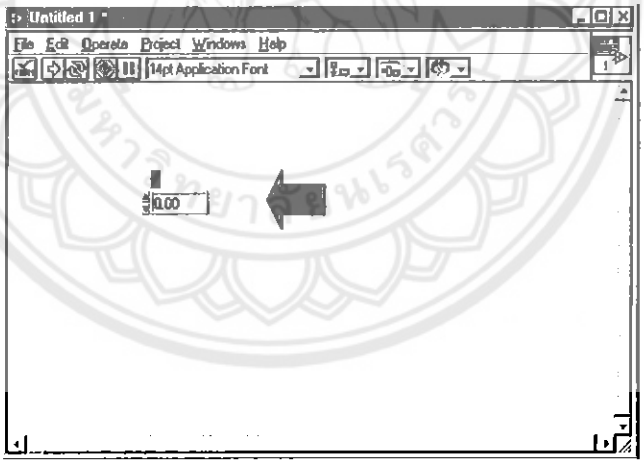
4. การเลือก Control และ Indicator เราสามารถเลือกจาก Numeric Subpalette ภายใต้ Control Palette ในทางปฏิบัติแล้ว Numeric ทุกตัวเป็นได้ทั้ง Control และ Indicator แต่ LabVIEW อาจจะตั้งค่าเบื้องต้นให้เป็นไปตามความเป็นจริงในการใช้งานมากที่สุด ยกตัวอย่างเช่น ปุ่มหมุนจะมีค่าเริ่มต้นเป็น Control, เข็มมาตรวัดจะมีค่าเริ่มต้นเป็น Indicator, เทอร์โมมิเตอร์ จะมีค่าเริ่มต้นเป็น Indicator, ปุ่มปรับเลื่อนจะมีค่าเริ่มต้นเป็น Control เป็นต้น เพราะธรรมชาติของอุปกรณ์เหล่านั้นที่ปรากฏบนหน้าปัทม์ของเครื่องมือจริง เราคงไม่เคยเห็นเครื่องมือใดที่ใช้ปุ่มหมุนเป็นตัวแสดงค่าหรือใช้เข็มชี้เป็นตัวกำหนดค่าควบคุม

5. อย่างไรก็ตามเนื่องจาก Front Panel ของ LabVIEW เป็นเครื่องมือเสมือนจริง ดังนั้นทุกอย่างเป็นไปได้ เราจะสามารถเปลี่ยนอุปกรณ์ทุกแบบเป็น Control หรือ Indicator ได้ โดยใช้คำสั่ง Change to Control/Indicator ภายใต้ Pop-up menu ของ object นั้น

6. เลือก Numeric Subpalette บน Controls Palette โดยการกดเมาส์ปุ่มซ้าย Numeric Subpalette จะปรากฏขึ้น ลองเลื่อนเมาส์เพื่อดูรายชื่อต่างๆ ของ Subpalette นั้น

7. เลือก Digital Control โดยกดเมาส์ปุ่มซ้าย

8. เลื่อนเมาส์กลับมาที่ Front Panel เมื่อลูกศรอยู่ในบริเวณ Front Panel เราจะเห็นรอยเส้นประของ Digital Control ปรากฏขึ้น เลือกวาง Digital Control ไว้ในตำแหน่งที่ต้องการ



9. หากเรากดเมาส์เพื่อวางตำแหน่งบน Front Panel แล้วเรายังไม่พอใจในตำแหน่งที่วาง เราสามารถเปลี่ยนตำแหน่งได้โดยไปที่ Tools Palette แล้วเลือก Positioning Tool ตัวชี้ของเมาส์จะเป็นลูกศรสีดำ หากเรานำเมาส์ไปกดบริเวณ Numerical Control ที่สร้างขึ้นจะปรากฏเส้นประรอบๆ Control นั้น เราสามารถที่จะขยายหรือเปลี่ยนวางตำแหน่งได้ลองทำตามขั้นตอนนี้ดู

10. ลองวาง Numerical Control อีกอันหนึ่งลงบน Front Panel คราวนี้ลองสังเกตดูอีกครั้งว่าหลังจากที่เราวางเสร็จจะมีสี่เหลี่ยมสีดำปรากฏอยู่เหนือ Control นั้นซึ่งเหมือนกับคราวก่อนแต่เราไม่ได้ให้ความสนใจเท่านั้น เพราะทุกครั้งที่เราวาง Indicator หรือ Control ลงไป LabVIEW จะเตรียมพร้อมที่จะรับชื่อหรือ Labels ของ Control หรือ Indicator นั้น ใน Numerical Control อันที่ 2 นี้ให้เราใส่ชื่อ **B** ลงไป

11. นำเมาส์ไปชี้บริเวณ Numerical control อันแรก แล้วกดเมาส์ปุ่มขวา เพื่อเรียก Pop-up menu ขึ้นมาจากนั้นเลือก **Show ▶ Label** ช่องสีดำเหนือ Control นี้จะเกิดขึ้นและเราสามารถที่จะกำหนดชื่อของ Control นี้ได้ เราจะให้ชื่อ Control นี้ว่า A

12. เลือก Position Tool ตัวชี้ของเมาส์จะเป็นลูกศร นำไปกดที่บริเวณ Numerical Control ที่สร้างขึ้นจะปรากฏเส้นประรอบๆ Control นั้น หากเราเคลื่อนย้ายตำแหน่งของ Numeric Control ส่วนต่างๆทั้งหมดจะติดตามกันไปด้วย แต่ถ้าเรานำเมาส์ไปกดเฉพาะที่ label เราสามารถจะเคลื่อนย้ายเฉพาะส่วน label ของ control นั้นเพียงอย่างเดียวได้

13. สร้าง Numerical Control อีก 1 อัน โดยตั้งชื่อเป็น A*B

14. ถ้าเราต้องการนำค่าจาก Control A และ B มารวมกันแล้วแสดงผลบน Control A*B จะทำไม่ได้ เพราะ Control A+B จะรับค่าไม่ได้ หากเราจะแสดงค่าของข้อมูลเราต้องใช้ Indicator ตามที่เราทราบมาแล้วจากบทที่ผ่านมา

15. การแก้ไขทำได้โดยจาก Pop-up menu ของ A*B เลือก Change to Indicator เราก็สามารถเปลี่ยนจาก Numerical Control เป็น Numerical Indicator A*B

16. ขั้นต่อไปจะลองวิธีการคัดลอก object โดยมีวิธีการดังนี้ เลือก Position Tool นำไปกดที่บริเวณ Numerical Indicator A*B จะปรากฏเส้นประรอบๆ Indicator นั้น หากเราต้องการจะ Copy ส่วนนี้ ภายใต้เมนู **Edit** เลือกคำสั่ง **Copy** จากนั้นหากต้องการคัดลอก object นั้นลงบน Front Panel ทำได้โดยภายใต้เมนู Edit เลือก Paste จะปรากฏ Indicator ชื่อเดียวกันนี้ขึ้นอีกตัวหนึ่งและจะมีคำว่า copy 2 ต่อท้ายชื่อเดิม

17. เราสามารถเปลี่ยนชื่อของ object ที่สร้างขึ้นแล้วนำมากดบริเวณชื่อของ Indicator ที่สร้างขึ้นใหม่ เราจะพบว่าเราสามารถแก้ไขชื่อนั้นได้โดยการเลือก Label Tool จาก Tools Palette แล้วนำนำเมาส์มากดบริเวณชื่อ Label ที่เราต้องการแก้ไข เราจะพบว่าเมื่อกดเมาส์ไปแล้วเราสามารถที่จะแก้ไขตัวหนังสือเหล่านั้นได้ ให้แก้ไขชื่อเป็น A/B และให้สังเกตอีกอย่างหนึ่งคือเมื่อพิมพ์เสร็จแล้วหากเรากด Enter บนแป้นพิมพ์ เราจะพบว่าเราจะได้บรรทัดของ Label นั้นเพิ่มขึ้นอีกบรรทัดหนึ่ง ซึ่งไม่ใช่สิ่งที่เราต้องการ วิธีการที่ถูกต้องคือใช้เมาส์กด Button ที่เขียนว่า Enter บน Toolbar หรือใช้เมาส์กดนอก Text Box นั้นๆ

18. ขั้นต่อไปเราจะสร้างตัวหนังสือขึ้นบน Front Panel โดยอันดับแรกเลือก Label Tool จากนั้นให้เรากดเมาส์ในบริเวณที่เราต้องการเขียนข้อความ จะปรากฏมี Text Box เล็กๆขึ้นเราสามารถใส่ข้อความได้ ถ้าเราไม่ใส่ข้อความใดๆ แล้วนำเมาส์ไปกดที่ใหม่ Text Box เดิมจะหายไป ให้เราวางกล่องข้อความบริเวณด้านบน VI แล้วพิมพ์คำว่า Simple Calculator

19. หากเราต้องการแก้ไขรูปแบบตัวหนังสือ อันดับแรกเลือก Label Tool แล้วนำไปเน้นข้อความบริเวณที่เราต้องการแก้ไข จากนั้นใช้ Font Ring ที่อยู่บน Toolbar ในการแก้ไขเปลี่ยนแปลงรูปแบบตัวอักษร รูปแบบตัวอักษรใน LabVIEW เป็นดังนี้

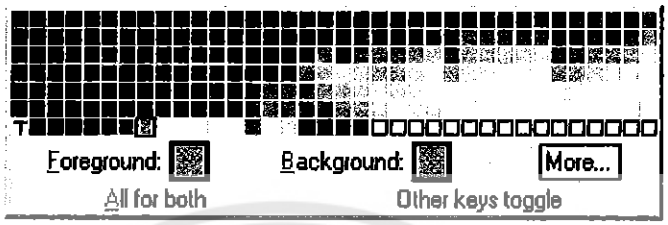
1. Application Font เป็น font หรือแบบตัวหนังสือที่ใช้กับตัวหนังสือบน Controls Palette, Function Palette และตัวหนังสือสำหรับ Controls ใหม่
2. System Font จะใช้กับตัวหนังสือในเมนู
3. Dialog Font จะใช้สำหรับตัวหนังสือใน Dialog Box ต่างๆ

20. หากเราต้องการจะเปลี่ยนแปลงตัวอักษรทั้งกลุ่ม เราไม่จำเป็นต้องใช้ Label Tool เน้นที่ตัวอักษรนั้นก็ได้ แต่ใช้ Positioning Tool เลือก Text Box หรือ object นั้นทั้งหมด ส่วนที่ถูกเลือกจะปรากฏเส้นประขึ้นรอบๆ จากนั้นเลือกแบบตัวหนังสือจาก Front Ring

21. ให้เปลี่ยนตัวหนังสือ Simple Calculator เป็นขนาด 24 pt ตัวหนา และมีสำน้ำเงิน

22. เปลี่ยน Label A, B, A*B, A/B เป็นตัวขนาด 18 pt การเลือกหลายๆ object พร้อมๆกันอาจใช้ Positioning Tool เลือกโดยเมื่อเลือกตัวแรกแล้วให้กดแป้น Shift ค้างไว้แล้วเลือกตัวอื่นๆต่อไป เส้นประรูปกรอบสี่เหลี่ยมประจะขึ้นกับทุก object ที่เลือก

23. ขั้นต่อไปจะเป็นวิธีการเปลี่ยนสี Control หรือ Indicator โดยสีจะแบ่งเป็นสองส่วนคือ ส่วนหน้า Foreground และสีพื้นหลัง Background เราสามารถจะเปลี่ยนสีได้โดยใช้ Coloring Tool เลือกเครื่องมือนี้จาก Tools Palette แล้วนำเมาส์ไปกดด้วยปุ่มขวาที่ object ใดเราจะได้หน้าต่างตามรูป

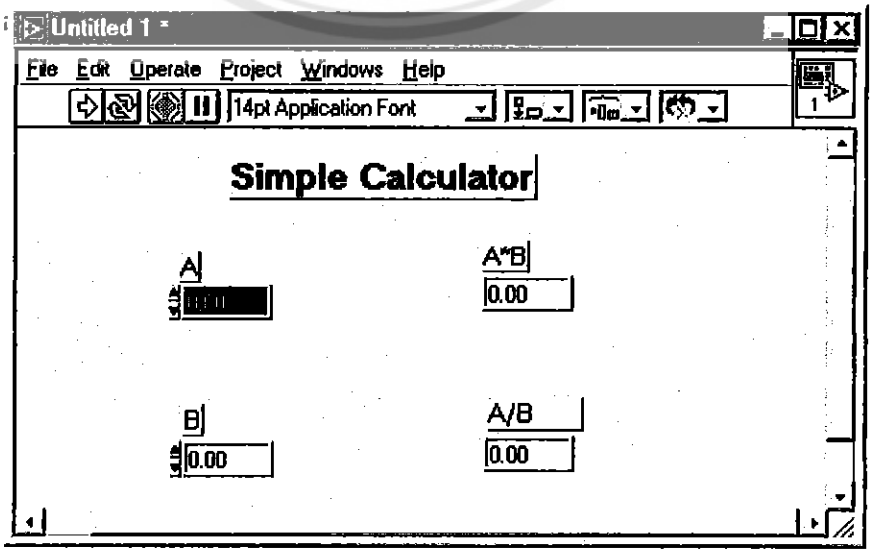


24. เมื่อเราเลื่อนเมาส์ไปบนสีต่างๆ object จะเปลี่ยนไปตามสีที่เราเลือก ถ้าเราเลือกให้เปลี่ยนเฉพาะสีด้านหน้าให้กดแป้น F (มาจาก Foreground) ถ้าต้องการเลือกเปลี่ยนเฉพาะด้านหลังให้กดแป้น B (มาจาก Background) ถ้าต้องการเปลี่ยนทั้งสองพร้อมกันให้ใช้เมาส์กดที่ A ถ้าต้องการสีเพิ่มให้ใช้เมาส์กดที่ More... สำหรับสีที่เขียนเป็นตัว T คือสีโปร่งใส (Transparent)

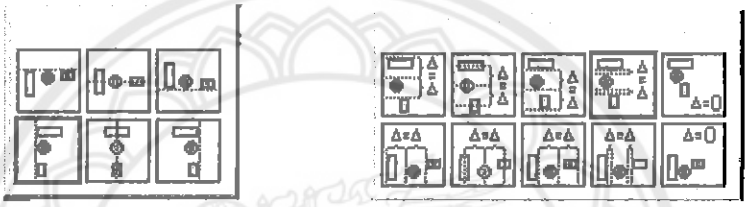
25. ให้เปลี่ยนสีของ Control A ให้มีสีพื้นเป็นสีดำ และให้ตัวเลขที่ปรากฏให้เป็นสีเขียว

26. หากเราต้องการจะคัดลอกสีที่มีอยู่เดิมเราสามารถใช้ Color Copy Tool โดย เมื่อเลือกเครื่องมือนี้แล้ว นำไปกดในที่มีสีที่ใด สีใน Coloring Tool จะเปลี่ยนตามไปกับสีนั้น ซึ่งเราสามารถสังเกตการเปลี่ยนแปลงได้จาก Tool Palette ขอให้ลองใช้เครื่องมือนี้ดู

27. เมื่อถึงขั้นตอนนี้ เราควรจะได้ Front Panel ตามรูป

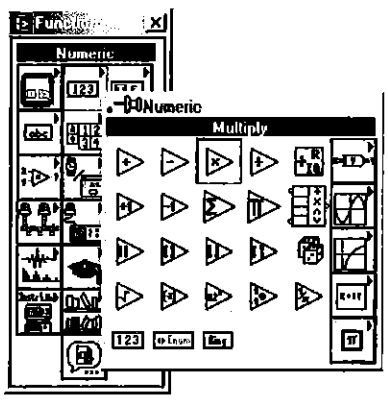


28. ขั้นต่อไปเราจะพิจารณาในส่วนของ Block Diagram ซึ่งเราได้พบจากกิจกรรมในบทที่ผ่านมาแล้วว่าทุกครั้งที่สร้าง object บน Front Panel จะปรากฏ terminal ขึ้นบน Block Diagram ขั้นแรกเราจะลองตกแต่งวางตำแหน่งต่างๆบน Block Diagram ให้เป็นระเบียบก่อน อันดับแรกเราจะใช้เครื่องมือช่วยในการจัดวาง object ซึ่งจะ มี 2 แบบอยู่บน Toolbar โดย แบบที่ 1 จะเป็นการจัดวางแนว **(Alignment Ring)** ใช้เมื่อต้องการวางแนวของ objectให้อยู่ในลักษณะที่ต้องการ และแบบที่สองคือการจัดระยะห่าง **(Distribution Ring)** ใช้เมื่อต้องการจัดระยะห่างให้เป็นไปตามที่เราต้องการ วิธีการใช้อันดับแรกให้เราเลือก object ที่ต้องการจะจัดแนวตั้งแต่ 2 object ขึ้นไปก่อน แล้วจึงเลือกว่าจะจัดแนวใด โดยใน Ring ทั้งสองจะมี subpalette ย่อยลักษณะดังที่แสดงในรูป รูปบน Palette เหล่านี้คงจะสามารถอธิบายตัวเลือกของการจัดวางได้เป็นอย่างดีอยู่แล้ว

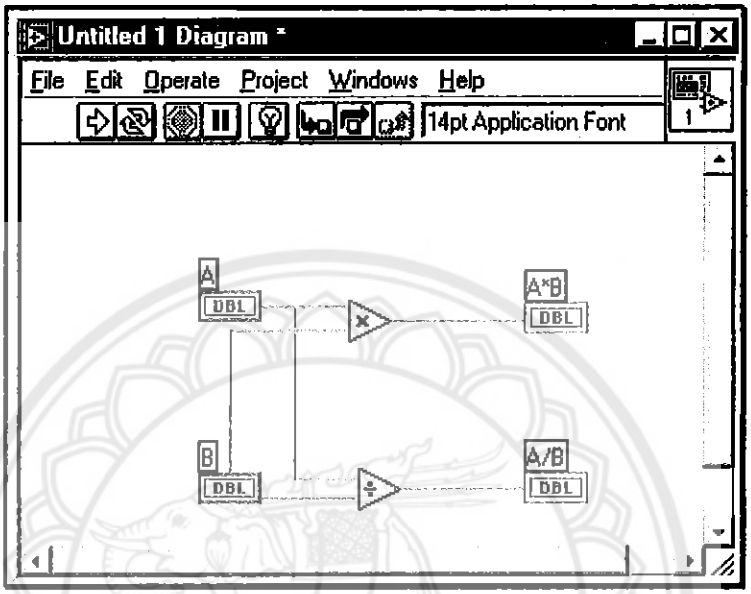


29. จัดวาง terminal ให้อยู่ในแนวนอนและแนวตั้งเดียวกันตามรูป วิธีการเลือก object หลายๆอันพร้อมกันอีกวิธีหนึ่งนอกจากใช้ปุ่ม Shift พร้อมกับ Position Tool เลือกทีละ object แล้ว เรายังสามารถทำได้โดย Position Tool กดที่บริเวณข้างๆ object ที่ต้องการจะเลือก แล้วกดค้างไว้จากนั้นดึงเมาส์ขยายออกเราจะเห็นสี่เหลี่ยมเป็นเส้นประ ตามที่แสดงในรูปเมื่อเราปล่อยเมาส์ object ที่อยู่ในกรอบสี่เหลี่ยมจะถูกเลือก

30. ที่ Functions Palette เลือก Numeric Subpalette และเลือก Multiply Function จากนั้นวางลงไปบน Block Diagram จากนั้นเลือก Division Function จาก Numeric subpalette บน Functions Palette แล้ววางลงบน Block Diagram



31. อันดับต่อไปเราจะเริ่มการต่อเชื่อมสายของ terminal ต่างๆบน Block Diagram เข้าด้วยกัน ขั้นแรกไปที่ Tools Palette เลือก Wiring Tool เมื่อกลับเข้าหาใน Block Diagram ตัวชี้ของเมาส์จะเป็นรูปสายไฟ สำหรับรายละเอียดและขั้นตอนการวางสายเราจะกล่าวถึงต่อไปในส่วนหลังของบทนี้ ในขั้นนี้ขอให้ต่อสายตามที่แสดงในรูปต่อไปนี้



32. ที่ Toolbar รูปลูกศร Run จะอยู่ในสภาพพร้อม คือ เป็นลูกศรต่อสีขาว เรากลับไปที่ Front Panel และขอให้ทดลองใช้ Continuous Run

33. หยุดการทำงานโดยกดปุ่ม Abort ทำให้ VI หยุดทำงานกลับมาอยู่ใน โหมดแก้ไขจาก File Menu เลือก Save และบันทึก VI ด้วยชื่อ Activity 3.1

2. PropertiesofControlsandIndicator

ในกิจกรรมที่ผ่านมาเราได้ทราบถึงหน้าที่ของ Control และ Indicator มาบ้างแล้ว ในส่วนนี้จะเป็นการกล่าวถึงคุณสมบัติต่างๆของ Control และ Indicator รวมถึงวิธีการแก้ไขเปลี่ยนแปลงค่าคุณสมบัติที่ LabVIEW ได้ตั้งเป็นค่าเริ่มต้นให้ด้วย

Representation ลักษณะของข้อมูลที่ปรากฏใน terminals ต่างๆภายใน Black Diagram จะขึ้นอยู่กับการกำหนดรูปแบบการแทนค่า (Representation) ของข้อมูล การเลือกรูปแบบการแทนค่าจะเป็นวิธีการเก็บข้อมูลในหน่วยความจำให้ในคอมพิวเตอร์ เราควรจะเลือกการกำหนดรูปแบบการแทนค่าข้อมูลให้เหมาะสมกับชนิดของข้อมูลที่ใช้ เพราะจำทำให้การใช้หน่วยความจำมีประสิทธิภาพมากขึ้น ยกตัวอย่างเช่น เลขจำนวนเต็ม (integer) จะใช้หน่วยความจำน้อยกว่าเลขทศนิยม (Floating point) ดังนั้นเมื่อเราทราบแน่นอนว่าตัวเลขควรจะมีค่าเท่ากับจำนวนเต็ม 2 พอดี เราก็ไม่ควรจะเก็บในหน่วยความจำด้วย 2.00 เป็นต้น นอกจากนี้ตัวเลขอาจจะเก็บไว้แบบมีเครื่องหมาย (sign) คือเป็นได้ทั้ง บวก ลบ หรือ ศูนย์ หรืออาจจะเก็บแบบ unsign หรือแบบไม่มีเครื่องหมาย คือ เป็นบวกหรือศูนย์เท่านั้น

ภายใน Block Diagram เราจะพบว่า terminal ต่างๆจะแสดงรูปแบบการแทนค่า หรือ Representation ของตัวเลขต่างกัน เราสามารถบอกรูปแบบการแทนค่าของข้อมูลเหล่านั้นด้วยตัวหนังสือย่อ และลักษณะของสี เช่นเลขจำนวนเต็มจะเป็นสีน้ำเงิน ส่วนเลขทศนิยมจะเป็นสีส้ม เป็นต้นและภายใน terminal จะมีตัวย่อ เช่น DBL จะแทนเลขแบบ Double-Precision Floating-Point เป็นต้น ลักษณะของข้อมูลที่เป็นตัวเลข LabVIEW ได้ใช้เป็นสัญลักษณ์กำหนด Representation ใน Block Diagram ได้แสดงในตารางต่อไปนี้

การแทน	ตัวย่อใน terminal	ขนาด (bytes)
Byte	I8	1
Unsigned byte	U8	1
Word	I16	2
Unsigned word	U16	2
Long	I32	4
Unsigned long	U32	4
Single precision	SGL	4
Double precision	DBL	8
Extended precision	EXT	10
Complex single	CSG	8
Complex double	CDB	16
Complex extended	CXT	20

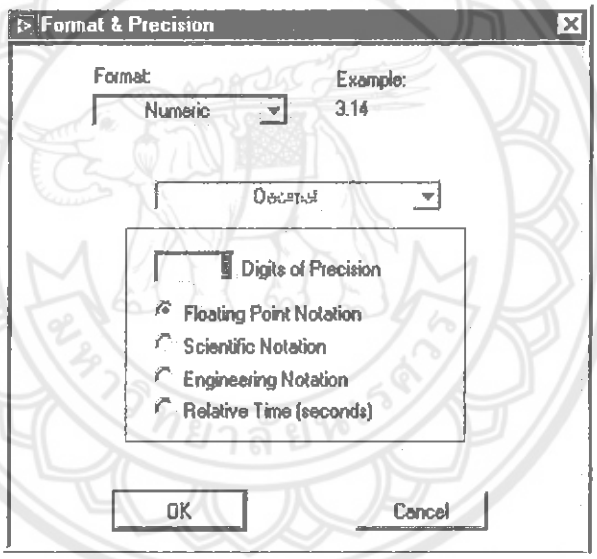
เราสามารถที่จะเปลี่ยนแปลง Representation ของค่าคงที่, Control หรือ Indicator ได้ โดยการให้ Pop-up menu ของ object นั้น แล้วเลือก Representation ตามความต้องการ การเลือกรูปแบบตัวเลขให้เหมาะสมนี้จะช่วยให้เราประหยัดหน่วยความจำได้มาก โดยเฉพาะในกรณีที่เราจะต้องใช้ Arrays ในหลายมิติจำนวนมาก

นอกจากนี้เรายังมีวิธีการเลือกการกำหนดรูปแบบตัวเลขให้เหมาะสมกับแหล่งข้อมูล โดยใช้ตัวเลือก Adapt to Source จาก Pop-up menu ซึ่งวิธีนี้ LabVIEW จะจัดการเลือกรูปแบบตัวเลขให้เหมาะสมกับชนิดของข้อมูลที่จะเข้าสู่ terminal นั้นโดยอัตโนมัติ

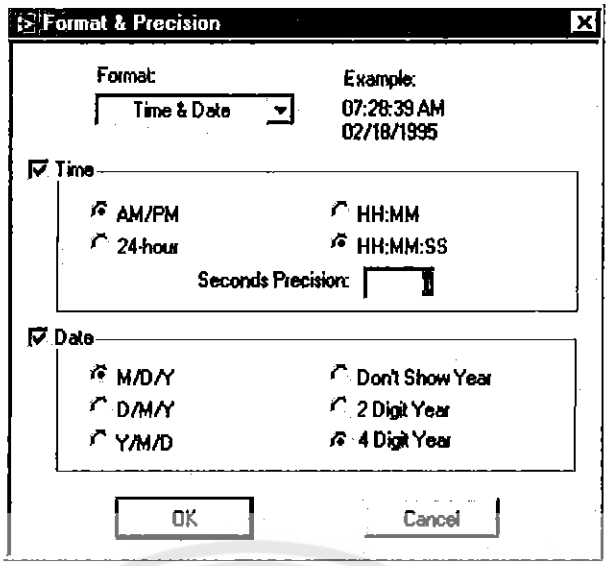
Format and Precision การแสดงค่าของข้อมูลที่ปรากฏใน Control และ Indicator บน Front Panel ของ VI เราสามารถจะกำหนดรูปแบบ (Format) และตำแหน่งทศนิยม (Precision) ของตัวเลขนั้นได้

การที่จะเลือกรูปแบบสามารถทำได้โดยใช้ Pop-up menu ของ object ที่ต้องการ จากนั้นเลือก Format & Precision สำหรับการเลือกรูปแบบนั้น อันดับแรกเราต้องเลือกว่าตัวเลขที่จะแสดงนั้นเป็นค่าที่จะใช้แสดงตัวเลข (Numeric) หรือใช้แสดง วัน-เวลา (Time & Date) หลังจากเลือกรูปแบบหลักแล้ว เราสามารถเลือกรูปแบบย่อยต่อไปได้อีก

ถ้าเราเลือกรูปแบบหลักว่าเป็นแบบ Numeric จาก Format Ring ซึ่งเป็นค่าเริ่มต้นที่ LabVIEW ตั้งขึ้นไว้ เราจะได้ลักษณะหน้าต่างเป็นไปตามรูป



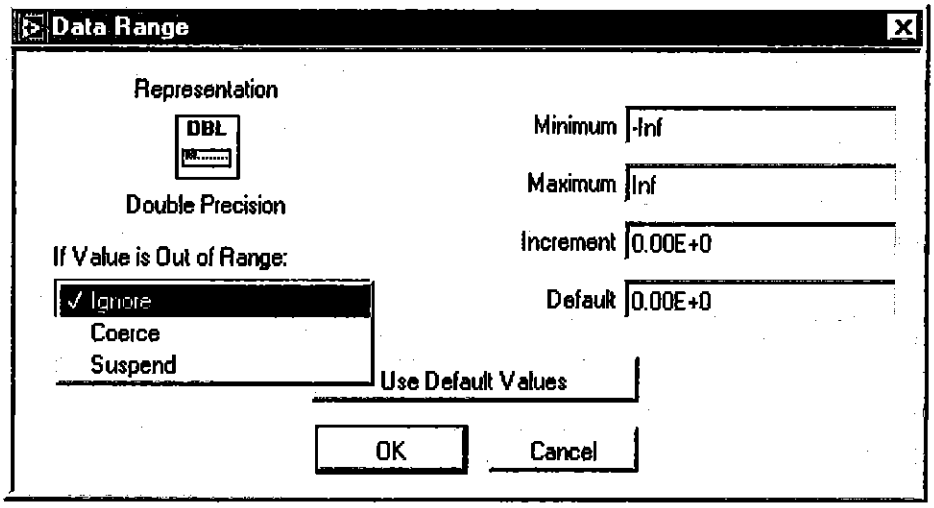
การเลือกตำแหน่งทศนิยมที่จะแสดงใน Control หรือ Indicator จะเลือกด้วยการใส่จำนวนตัวเลข ใน Digits of Precision นอกเหนือจากนั้นเรายังสามารถเลือกให้แสดงตัวเลขเป็นแบบทศนิยมปกติ หรือ แบบอื่น เช่น Scientific Notation เป็นต้น ส่วนในอีกกรณีคือถ้าหากใน Format ring เราเลือกที่จะให้แสดง Time & Date เราจะได้หน้าต่างลักษณะตามรูปต่อไปนี้



ซึ่งเราสามารถที่จะเลือกตัวเลือกต่างๆ ได้ตามที่เราต้องการ ในกรณีที่เราแสดงค่าด้วยเข็มหรือปุ่มหมุน อาจจะเป็นการยากที่จะอ่านค่าได้อย่างละเอียดหรือในบางกรณีเราอาจต้องการแสดงค่านี้ไปพร้อมๆกัน เราสามารถที่จะให้ LabVIEW แสดงตัวเลขไปพร้อมๆ กันได้ โดยใช้ Pop-up menu แล้วเลือก Show ▶ Digital Display และส่วนนี้จะเป็นส่วนหนึ่งของ object ดังนั้นจะไม่ปรากฏใน Block Diagram อีก กล่าวคือ LabVIEW จะมอง Digital Display ส่วนนี้เป็นส่วนหนึ่งของ object นั้น อย่างไรก็ตาม Digital Display ของ object ที่ปรากฏบน Front Panel สามารถเลื่อนไปมาบน Front Panel ได้อย่างอิสระไม่จำเป็นต้องติดอยู่กับ object หลักตลอดเวลาก็ได้

Data Range การออกแบบ VI ให้ผู้ใช้นำไปใช้งานนั้น ในบางครั้งเราต้องการที่จะบังคับช่วงของตัวเลขที่ผู้ใช้จะนำไปใช้ควบคุมให้อยู่ในช่วงที่แน่นอน เช่นเราอาจต้องการให้ค่าเปลี่ยนแปลงจาก -1 ถึง 1 เท่านั้น หรือต้องการกำหนดว่าหากผู้ใช้กดปุ่มลูกศรขึ้นหรือลงเพื่อเปลี่ยนค่าเราจะให้ตัวเลขเปลี่ยนไปที่จะ 0.1 การกำหนดที่จะตั้งตัวเลขเหล่านี้สามารถทำได้โดยการกำหนด Data Range ของ Control หรือ Indicator เหล่านั้น

จาก Pop-up menu ของ object นั้นเลือก Data Range... เราจะได้หน้าต่างของ Data Rang ซึ่งมีลักษณะดังต่อไปนี้ปรากฏขึ้น



จากส่วนนี้ เราสามารถกำหนดค่าสูงสุด (Maximum) ต่ำสุด (Minimum) การเพิ่มหรือลดเมื่อกดปุ่มลูกศรขึ้นลง (Increment) สำหรับค่าเบื้องต้นจะแสดงในรูปข้างบนนี้ โดย inf. จะแทน Infinity ส่วนการกำหนดในช่อง Default จะเป็นการกำหนดค่าเริ่มต้นหรือค่าที่เรากำหนดเมื่อ VI เริ่มเข้าสู่โหมดการทำงาน

ส่วนสำคัญอีกส่วนหนึ่งก็คือจะให้ LabVIEW ทำอย่างหากผู้ใช้ใส่ตัวเลขเกินกว่าช่วงที่เราได้กำหนด ในรูปเราจะเห็นว่ามี If value is Out of Range ซึ่งเป็น ring อยู่ เราสามารถเลือกวิธีการใน ring นี้ได้ 3 แบบ คือ

1. **Ignore** คือไม่ต้องสนใจ ให้ใช้ค่าที่ผู้ใช้ใส่ลงไปเลย ในกรณีนี้ถ้าผู้ใช้ใช้วิธีกดลูกศรขึ้นลง ค่าข้อมูลจะหยุดอยู่ที่ต่ำสุดหรือสูงสุด แต่ถ้าผู้ใช้เลือกใช้วิธีใส่ตัวเลขโดยตรงทางแป้นพิมพ์แล้วกด Enter ช่องข้อมูลก็จะแสดงค่าที่ผู้ใช้ใส่ลงไป และ LabVIEW จะทำงานต่อไปโดยใช้ข้อมูลเป็นตัวเลขที่ผู้ใช้ใส่ และไม่สนใจค่าตัวเลขสูงสุดหรือต่ำสุด

2. **Coerce** คือ ปรับให้ตัวเลขอยู่ในช่วงค่าที่กำหนดแล้วทำงานต่อไป โดยถ้าค่าสูงกว่าค่าสูงสุด LabVIEW จะปรับให้เท่ากับค่าสูงสุดแล้วทำงานต่อไป ถ้าค่าต่ำกว่าค่าต่ำสุด LabVIEW ก็จะปรับให้เท่ากับค่าต่ำสุด แล้วทำงานต่อไป

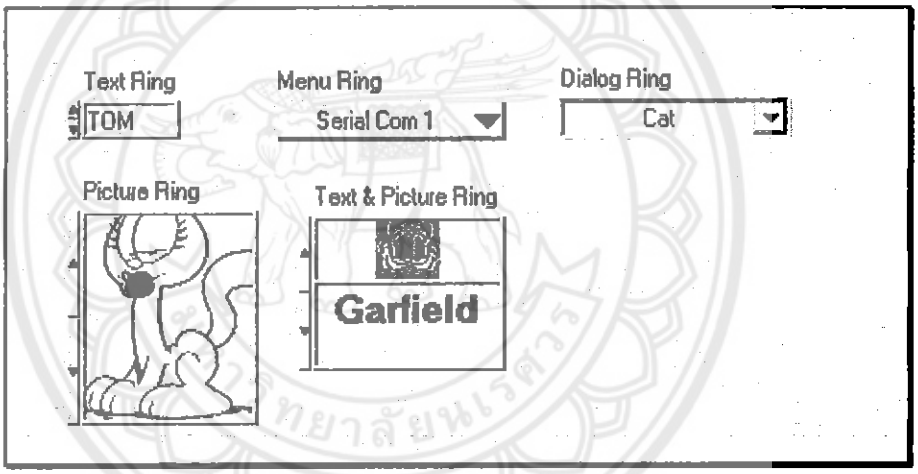
3. **Suspend** ถ้าเราเลือกตัวเลือกนี้แล้วข้อมูลออกนอกช่วงที่กำหนด LabVIEW จะระงับการประมวลผล control ที่กำหนดค่านั้นจะปรากฏกรอบสีแดงขึ้นรอบ และปุ่ม RUN จะมีวงกลมสีแดงพร้อมเส้นทะแยงคาดทับวางอยู่ ผู้ใช้จะต้องปรับค่าให้อยู่ในช่วงก่อน LabVIEW จึงจะทำงานต่อไป

ถ้าหากว่าต้องการกลับไปใช้ค่าที่ LabVIEW เตรียมไว้ให้ ให้กดปุ่ม Use Default Value เพื่อให้ค่ากลับมาเป็นค่าเริ่มต้นที่ LabVIEW เห็นว่ามีความเหมาะสมที่สุด

3. Others Controls and Indicators

ที่ผ่านมาเรามักจะเลือก Control และ Indicator ที่ปรากฏอยู่ภายใน Numeric Subpalette ในกิจกรรมต่างๆ อย่างไรก็ตาม LabVIEW มี Control และ Indicator ต่างๆอีกหลายแบบ ในหัวข้อนี้เราจะแนะนำ Control และ Indicator ในรูปแบบอื่นๆที่อาจต้องใช้บน Front Panel

Rings เป็น object ที่ใช้เป็นตัวเลือกว่าผู้ใช้ต้องการตัวเลือกแบบใด และเมื่อผู้ใช้เลือกค่าแล้ว ค่าที่ได้จาก Ring จะเป็นจำนวนเลขจำนวนเต็มแบบไม่มีเครื่องหมายชนิด 16 bit (U16) ในการให้ผู้ใช้เลือกค่าใน Rings นี้ เราอาจจะใช้เป็นตัวอักษรเป็นคำอธิบายหรืออาจใช้รูปภาพก็ได้ การใช้ Ring นี้จะมีประโยชน์มากในกรณีที่จำเป็นที่ต้องมีตัวเลือกหลายๆ แบบให้ผู้ใช้เลือกก่อนที่จะทำการประเมินข้อมูลต่อไป ลักษณะของ Ring แบบต่างๆ แสดงในรูปต่อไปนี้



ขั้นแรกในการใช้ Ring นั้น ให้เลือก ชนิดจาก List & Ring Subpalette จาก Controls Palette ซึ่งจะมีหลายแบบด้วยกัน อาจจะเป็นแบบที่มีได้เฉพาะตัวอักษร หรือแบบที่มีเฉพาะรูปภาพ หรืออาจมีทั้งสองแบบรวมกัน เมื่อเราเลือกแบบ Ring ที่ต้องการแล้ววาง Ring นั้นลงบน Front Panel ในครั้งแรก จะมีตัวเลือกเพียงตัวเดียว จากนั้นเราสามารถจะเพิ่มเติมตัวเลือกใน Ring เพิ่มในภายหลังได้ โดยใช้ Pop-up menu แล้วเลือก Add Item After หรือ Add Item Before ก็ได้

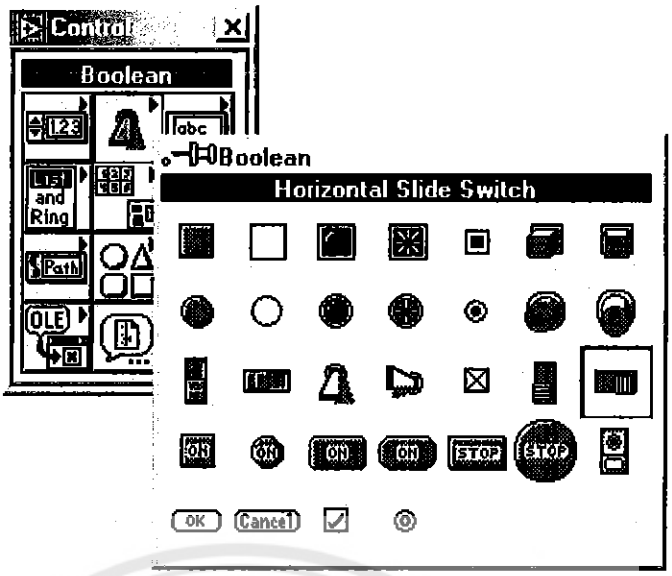
ในการใส่ตัวหนังสือลงใน Ring ให้เลือก Edit Text Tools ใน Tools Palette แล้วใช้เมาส์กดลงในบริเวณ Ring นั้น เมื่อใส่ข้อความเสร็จสิ้นแล้วให้เราใช้เมาส์กดปุ่ม Enter ที่อยู่บน Tool bar แต่ถ้าหากเรากดแป้น Enter แล้วเราจะได้ Text อีกบรรทัดหนึ่งซึ่งจะเป็นส่วนหนึ่งที่อยู่ในตัวเลือกเดียวกัน เมื่อต้องการใส่ตัวหนังสือเพิ่มในช่องเลือกต่อไป ก็ให้กดที่ปุ่มเลื่อนตำแหน่งด้านซ้ายมือ เพื่อให้ช่องข้อความนั้นปรากฏขึ้นแล้วเพิ่มเติมข้อความลงไป เราสามารถดูได้ว่าขณะนี้เราอยู่ที่ชิองตัวเลือกใดได้จากการใช้ Pop-up menu แล้วเลือก Show ▶ Digital Display

ในการใส่รูปภาพลงใน Picture Ring ขั้นแรกเราจำเป็นต้องมีรูปภาพอยู่ใน Clipboard ก่อนที่จะใส่รูปภาพลง วิธีการนำรูปภาพเข้าสู่ Clipboard ก็คือต้องใช้โปรแกรมวาดภาพดูรูปที่ต้องการ จากนั้นเลือกรูปที่ต้องการ โดยใช้คำสั่ง Copy ของโปรแกรมวาดรูปนั้นๆ เพื่อเก็บรูปภาพลงไปใน Clipboard ก่อน ต่อจากนั้นเมื่อจะนำรูปภาพมาวางลงใน Ring ลำดับใด ก็ให้เลื่อน Ring ไปที่ลำดับนั้นแล้วใช้คำสั่ง Import Picture ภายใต้ Pop-up menu ของ Ring นั้น ส่วนการใส่รูปที่ตัวเลือกอื่นๆ ก็ทำได้โดยการใช้ขั้นตอนเดียวกัน

เมื่อเราใส่ตัวเลือกลงใน Ring แล้ว ค่าที่ส่งออกจาก terminal เมื่อเราสั่ง VI ทำงานจะเป็นค่าตามลำดับการใส่ คือ ตัวเลือกแรกจะให้ค่าตัวเลขเป็นจำนวนเต็ม 0 ตัวเลือกต่อไปจะให้ค่าตัวเลขเป็นจำนวนเต็ม 1 และเพิ่มขึ้นเรื่อยๆ ตามลำดับ เราสามารถดูค่าเหล่านั้นบน Front Panel ได้โดยเลือก Show ▶ Digital Display จาก Pop-up menu ของ Ring นั้นตามที่ได้กล่าวมาแล้ว

โดยทั่วไปการใช้ Ring จะมีประโยชน์ก็ต่อเมื่อเราใช้ร่วมกับ Case Structure ซึ่งเรายังไม่ได้กล่าวถึง Structure ในขั้นนี้ ดังนั้นเราจึงจะไม่ยกตัวอย่างหรืออธิบายส่วนประกอบเพิ่มเติมในวันนี้ อย่างไรก็ตามเราแนะนำให้คุณลองเลือกและวาง Ring บน Front Panel แล้วลองใส่ตัวหนังสือ หรือรูปภาพลงใน Ring ที่สร้างขึ้น เพื่อความคุ้นเคยในการสร้าง Ring

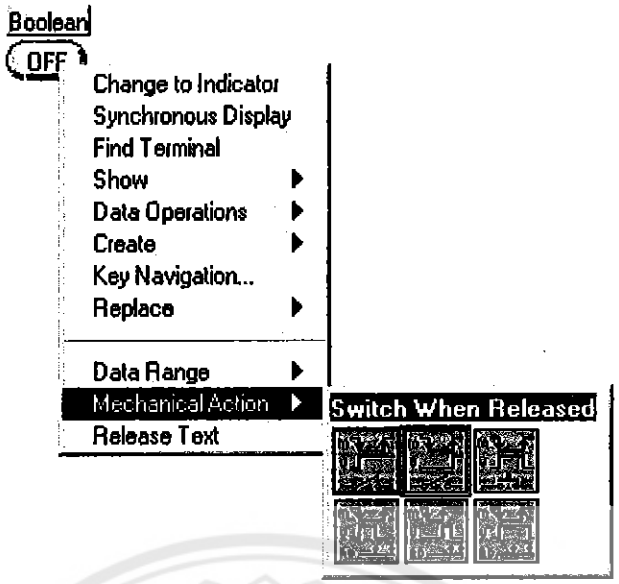
Boolean เป็นชื่อของนักคณิตศาสตร์ชาวอังกฤษชื่อ George Boolean ซึ่งเป็นผู้วางรากฐานของ Boolean Algebra สำหรับในงานวิศวกรรมนั้นเราสามารถที่จะมองว่า Boolean เป็นการกำหนดข้อมูลให้มีค่าเป็น จริง หรือ เท็จ หรืออาจจะมองเป็นการ เปิด - ปิด เพื่อใช้ในการควบคุมการทำงานของส่วนต่างๆ เราสามารถเลือกใช้ Boolean ได้จาก Boolean Subpalette ภายใต้ Controls Palette ที่แสดงในรูปต่อไปนี้



ใน Boolean Subpalette นี้จะประกอบด้วยสวิชท์ และไฟสัญญาณหลายแบบ ซึ่งการกำหนดเบื้องต้นว่าจะเป็น Control หรือ Indicator จะขึ้นอยู่กับลักษณะที่เราใช้ในเครื่องมือวัดจริง เช่น สวิชท์ จะเป็น Control และไฟสัญญาณ จะเป็น Indicator เป็นต้น อย่างไรก็ตาม เราสามารถเปลี่ยนให้อุปกรณ์เหล่านั้นเป็น Control หรือ Indicator ก็ได้ ตามความต้องการของเรา โดยใช้คำสั่ง Change to Control/ Indicator ภายใน Pop-up menu ของ Boolean นั้น

ใน Block Diagram ลักษณะของ Boolean terminal จะเป็นสีเขียว และมีตัวอักษร TF อยู่ตรงกลาง TF และเหมือนกับตัวเลขคือ ถ้าเป็น Control จะมีของที่หนากว่าขอบของ Indicator

การเลือกใช้ Boolean ใน LabVIEW นั้น นอกจากสัญลักษณ์มาตรฐานที่กำหนดแล้วยังมีตัวเลือกให้เราเลือกใช้ได้อีกหลายอย่างที่เลือกจาก Pop-up menu สำหรับตัวเลือกที่สำคัญมีดังต่อไปนี้



1 Labeled Buttons

ในสวิชท์และไฟสัญญาณต่างๆ ของ Boolean บางแบบอาจจะมีตัวหนังสือประกอบไว้บน object นั้นด้วย เช่น ON และ OFF โดยปกติตัวหนังสือประกอบนี้จะไม่ได้แสดงกับสวิชท์หรือไฟสัญญาณทุกชนิด แต่เราสามารถที่จะเลือกให้แสดงหรือไม่แสดงได้โดยใช้ตัวเลือกจาก Show ▶ Boolean Text ภายใต้ Pop-up menu

Boolean Text นี้จะแสดง ON หรือ OFF อย่างใดอย่างหนึ่งขึ้นกับสภาวะที่ object นั้นเป็นอยู่ขณะนั้น เช่นถ้าเปิดอยู่จะแสดงตัวหนังสือ ON เป็นต้น เราสามารถที่จะจัดวางตำแหน่งและแก้ไขตัวหนังสือเหล่านั้นได้โดยการใช้ Positioning Tool ในการจัดวางตำแหน่ง และ Edit Text Tool ในการเปลี่ยนหรือแก้ไขตัวหนังสือ ON-OFF นั้น ทั้งสองเลือกได้จาก Tools Palette

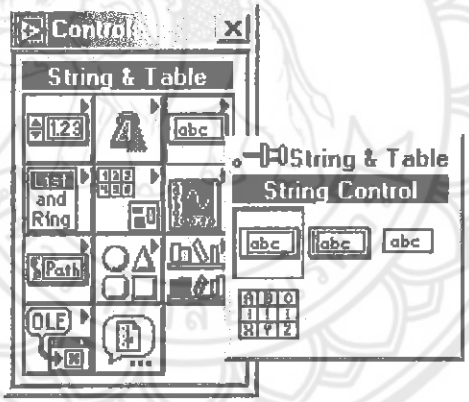
ค่าของ Boolean ใน Block Diagram จะเป็นจริง (TRUE) ถ้า object นั้นอยู่ที่ตำแหน่ง ON และเป็นเท็จ (FALSE) ถ้า object นั้นอยู่ในตำแหน่ง OFF

2 Mechanical Action Boolean Control จะมี Mechanical Action เป็นตัวเลือกใน Pop-up menu ตัวเลือกนี้จะเป็นตัวกำหนดลักษณะของ Control ว่ามีการทำงานอย่างไร เหมือนกับสวิชท์ในความเป็นจริงว่ามีหลายแบบ เช่น กดแล้วเปิด กดอีกครั้งจะปิด หรือต้องกดอยู่ตลอดเวลา จึงจะเปิด ถ้าปล่อยเมื่อใดจะปิด หรืออื่นๆ ตัวเลือกนี้จะเป็นตัวเลือกลักษณะการทำงานต่างๆ ของ Control เท่านั้น รายละเอียดต่างๆ ดูจาก Help ของ LabVIEW

3 **Data Range** แม้ว่า Boolean Control จะมีค่าเพียง 2 ค่า คือ จริง กับ เท็จ เท่านั้น แต่เราสามารถกำหนดได้ว่าจะให้ LabVIEW หยุดทำงานหรือไม่ ถ้าหากว่าเป็นค่าจริงหรือเท็จค่าใดค่าหนึ่ง โดยเลือกจาก Suspend if True หรือ Suspend if False จากคำสั่ง Data Range ใน Pop-up menu ของ object นั้น

4 **การเปลี่ยนลักษณะของ Boolean** แม้ว่า LabVIEW จะมีลักษณะของสวิชท์และไฟสัญญาณหลายแบบให้เราได้เลือกใช้ เราอาจมีความจำเป็นต้องใช้ Boolean ในรูปแบบอื่นก็ได้ เราสามารถที่จะนำรูปภาพอื่นมาใช้ ซึ่งเราสามารถทำได้โดยใช้คำสั่ง Edit Control ... จาก Edit menu จากนั้นเราจะสามารถนำรูปภาพใดก็ได้มาวางลงในตำแหน่งของ Boolean นั้นทั้งในสภาวะ ON และ OFF สำหรับรายละเอียดเหล่านี้จะกล่าวถึงในภายหลัง

Strings ก็คือตัวแปรหรือค่าคงที่ซึ่งอยู่ในรูปของตัวอักษร เราสามารถกำหนดให้ String นี้เป็นได้ทั้ง Control และ Indicator สำหรับ String นี้จะอยู่ใน String & Table Subpatette ภายใต้ Control Palette ซึ่งจะมีลักษณะดังรูป สำหรับรายละเอียดของ String นี้ เราจะกล่าวถึงภายหลัง



Path เป็น Control ที่ให้ LabVIEW แสดง Directory ที่ VI กำลังทำงานอยู่ ตัวอย่างเช่นใน Windows จะมีลักษณะของ Path เป็นเหมือนคำสั่ง DOS ยกตัวอย่างเช่น C:\Program Files\National Instruments\LabVIEW\MYWORK

แสดงว่าขณะนี้ VI ที่ทำงานอยู่นี้ คือ MYWORK อยู่ใน directory ชื่อ LabVIEW และอยู่ใน drive C หากว่าเรากำหนด Path ขึ้นแต่ว่า LabVIEW ไม่สามารถที่จะกำหนดที่อยู่ของ file นั้นได้ เราจะได้คำตอบเป็น <Not A Path>

คำสั่งนี้จะมีประโยชน์ในการเปิด-ปิด หรือเขียนไฟล์เพื่อแก้ข้อมูล เราจะกล่าวถึงวิธีการดังกล่าวในเรื่องของ File I/O ในภายหลัง

4. Wiring

ในกิจกรรมที่ผ่านมาเราได้ทำการต่อสายภายใน Block Diagram มาบ้างแล้ว ในหัวข้อนี้เราจะกล่าวถึงเทคนิคการต่อสายในรายละเอียด รวมถึงการแก้ไขปัญหาต่างๆเนื่องจากการต่อสายด้วย

การต่อสาย ที่ผ่านมาเราได้นำเสนอวิธีการจัดวาง Control และ Indicator แบบต่างๆ ลงใน Front Panel ซึ่งเป็นเพียงการเริ่มต้นการสร้าง IV เท่านั้น ความสำคัญที่แท้จริงของการเขียน VI คือการเลือกใช้ Function การทำงานให้เหมาะสมและต่อสายให้ถูกต้อง เพื่อจะได้ผลออกมาตามที่ต้องการ ในส่วนต่อไปของหัวข้อนี้ เราจะกล่าวถึงวิธีการต่อสายของส่วนต่างๆ เข้าด้วยกัน

เราจะใช้ Wiring Tool ในการต่อสายระหว่าง terminal mujxikdDvp^j4kp.o สนแเา ฎรฟ เพฟท เมื่อเราเลือก Wiring Tool แล้วลูกศรชี้จะเปลี่ยนเป็นรูปขดสายไฟ โดยตำแหน่งสำคัญของตัวชี้ หรือที่เรียกว่า "Hot Spot" ของตัวชี้จะอยู่ที่ปลายสายไฟที่ไม่ได้ม้วน ตามรูป

ในแบบฝึกหัดที่ผ่านมาเราคงได้ฝึกการต่อสายมาบ้างแล้ว แต่เราไม่ได้กล่าวถึงวิธีการต่อสายและขั้นตอนการต่อสายอย่างละเอียดซึ่งเราจะกล่าวถึงในที่นี้ ในการที่จะต่อสายระหว่าง terminal เข้าด้วยกัน อันดับแรกเลือก Wring Tool จาก Tools palette จากนั้นนำปลายของตัวชี้ไว้ที่ terminal หนึ่ง แล้วกดเมาส์ (Click) จากนั้นลากเม้าส์ไปที่ terminal ปลายทาง แล้วกดเมาส์อีกครั้งหนึ่ง ก็จะเสร็จสิ้น สำหรับรายละเอียดเพิ่มเติมและเทคนิกของการต่อสายให้ออกมาสวยงามและเป็นไปตามต้องการมีดังต่อไปนี้

การต่อสาย terminal ทั้งสองเข้าด้วยกัน ในการเชื่อมตัว 2 terminal เข้าด้วยกัน เราจะเริ่มจาก terminal ใดก็ได้ เมื่อเรานำปลายตัวชี้เข้าใกล้ terminal ใด เราจะสังเกตเห็นว่า terminal นั้นจะกระพริบและแสดงปลายสายให้เห็น นอกเหนือจากนั้นเราจะสังเกตว่ามีตัวอักษรที่เป็นชื่อของ terminal ที่เรากำลังชี้อยู่ปรากฏขึ้นอยู่ด้วย

การต่อสายเข้ากับ node ที่มีหลาย terminal อยู่ติดๆกัน สายต่อจะเข้าสู่ terminal ที่กำลังกระพริบอยู่เท่านั้น ไม่ว่าเราจะลากสายมาจากทิศทางใดของ node หรือเข้าสู่ node ที่ตำแหน่งใดก็ตาม ข้อเสนอแนะก็คือควรต่อสายให้เข้าสู่ terminal ตรงตามตำแหน่งและทิศทางของ terminal ใน node นั้นเพื่อความสะดวกในการแก้ไข

การกดเมาส์ปุ่มขวาจะเป็นการยกเลิกการต่อสาย

ในการลากสายในระหว่างการเชื่อมต่อนั้นเราไม่จำเป็นต้องกดเมาส์เอาไว้ เราสามารถปล่อยได้เลย แล้วกดอีกครั้งหนึ่งเมื่อถึงปลายทาง ยกเว้นเราต้องการจะเปลี่ยนทิศทางจากการเดินสายในแนวดิ่งให้เป็นแนวระดับหรือกลับกัน โดยปกติ LabVIEW จะยอมให้มีการเลี้ยวมุมฉาก (หรือเปลี่ยนแนวการเดินสาย) ได้ 1 ครั้ง โดยไม่ต้องกดเมาส์ แต่ถ้าเราต้องการจะเลี้ยวมากกว่า 1 ครั้ง เราจะต้องกดเมาส์เพื่อจะเลี้ยวในครั้งต่อไป

เมื่อเกิดการเลี้ยวสายโดย LabVIEW เราสามารถกำหนดให้ LabVIEW เปลี่ยนเส้นทางไปอยู่อีกแนวหนึ่งได้ เช่น LabVIEW เดินสายในแนวระดับก่อนแล้วจึงเดินสายในแนวดิ่ง แต่เราต้องการให้เดินสายในแนวดิ่งก่อนแล้วจึงเดินในแนวระดับ เราสามารถเปลี่ยนแปลงได้โดยให้เรากดแป้น Space Bar บนแป้นพิมพ์ไว้ การเดินสายทั้งสองแนวจะเกิดการกระพริบสลับกัน เราสามารถเลือกการเดินสายแบบใดแบบหนึ่งได้ โดยการปล่อยเมาส์ในจังหวะกระพริบที่เราต้องการ

การเชื่อมสายแยกสามารถทำได้โดยให้ปลายของตัวชี้ ชี้อยู่บนเส้นที่ต้องการจะแยกสาย เส้นที่ได้รับการชี้อยู่จะกระพริบ เราสามารถกดเมาส์หนึ่งครั้งเพื่อเชื่อมสายเข้าด้วยกัน จากนั้นเราสามารถลากสายไปสู่ terminal ที่ต้องการต่อไปได้

การลบสายให้ใช้ Positioning Tool เลือกสายแล้วกดแป้น Delete จะเป็นการลบสายในส่วนนั้นออกไป

การเคลื่อนย้ายสายสามารถทำได้โดยใช้ Positioning Tool เลือกสายที่ต้องการแล้วจัดวางในตำแหน่งใหม่หากเราต้องการ หากเราต้อการย้ายสายให้ดึงสายไปจากตำแหน่งเดิมแล้ววางลงไปในตำแหน่งใหม่ เมื่อได้ตำแหน่งที่ต้องการให้ปล่อยปุ่มเมาส์ สายที่ต่ออยู่ในแนวเดิมจะหายไปแล้วปรากฏขึ้นในตำแหน่งใหม่ ส่วนสายต่อส่วนอื่นๆจะยืดตามส่วนนั้นโดยอัตโนมัติ

การต่อสายจะมีความยุ่งยากมากขึ้นเมื่อ node นั้นมี terminal จำนวนมาก การต่อสายเข้ากับ object ที่มี terminal จำนวนมากนี้จะเป็นการดีถ้าหากเราไม่รีบร้อนและดูตัวหนังสือ (Tip) และสายของ terminal นั้นว่าต้องการข้อมูลลักษณะใด เพราะเมื่อเราใช้ปลายตัวชี้อยู่ในตำแหน่งของ terminal ใด terminal นั้นจะแสดง Tip และลักษณะของปลายลวดที่จะเชื่อมต่อ การสังเกตที่ปลายสาย โดยดูจากสี ขนาด และลักษณะของสายจะช่วยบอกเราได้ว่า terminal นั้นต้องการข้อมูลประเภทใด เพื่อช่วยในการต่อสายให้ถูกต้อง นอกเหนือจากนั้นถ้าที่ปลายสายของ terminal มีจุดอยู่แสดงว่า terminal นั้นเป็น input ของ node นั้น และถ้าหากที่ปลายสายของ terminal ไม่มีจุด แสดงว่าเป็น output ของ node นั้น

การใช้ Help Windows ในการเชื่อมสาย จะเป็นอีกวิธีหนึ่งที่จะสามารถผู้เริ่มต้นใช้งานในการต่อสายเข้ากับ object ที่เรายังไม่คุ้นเคย เพราะใน Help Windows จะบอกลักษณะของข้อมูลที่แต่ละ terminal ต้องการ นอกจากนั้นยังจะบอกถึงว่า Input ใด เป็นค่าที่จำเป็น (Required) หรือแนะนำ (Recommended) หรือเป็นตัวเลือก (Optional)

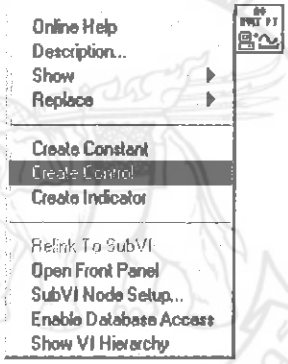
สายเสีย หากว่าเราต่อสายผิดพลาด เราจะพบว่าสายต่อจะปรากฏเป็นเส้นประสีดำ แทนที่จะเป็นเส้นสีเหมือนสายปกติอื่นๆ และเราจะพบว่าในขณะนั้น VI จะไม่สามารถทำงานได้ จนกว่าการต่อสายจะเป็นไปอย่างถูกต้อง เราสามารถลบสายเสีย (Bad Wire) ได้โดยการใช้คำสั่ง Remove Bad Wires จาก Edit menu หรือ กด <Ctrl> - B แล้วพยายามต่อสายใหม่ให้ถูกต้อง

ถ้าหากไม่ทราบสาเหตุว่าทำไมสายต่อจึงกลายเป็นสายเสีย ให้กดเมาส์ที่ปุ่ม Run (ซึ่งขณะนี้เป็นรูปลูกศรหักอยู่) หรือ Pop-up ที่สายเสียนั้นแล้วเลือก List Errors เราจะเห็น Dialog Box ที่แสดงความผิดพลาดที่ทำให้สายเกินขึ้น

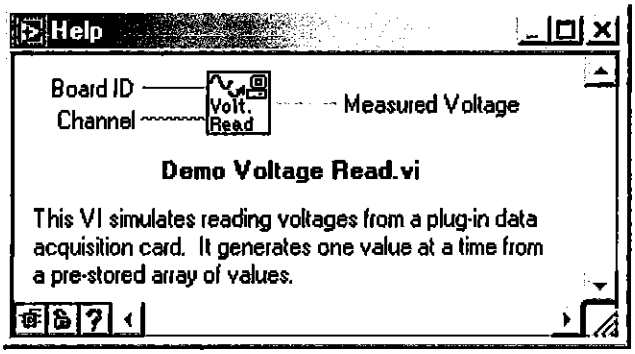
การเคลื่อนย้าย Object ปกติเมื่อเราต่อสายระหว่าง object คู่ใดเข้าด้วยกันแล้ว เมื่อเราเลื่อน object หนึ่งที่อยู่บน Block Diagram สายที่ต่อไว้ก่อนหน้านี้จะยืดตามหรือหดเข้าไปด้วยโดยอัตโนมัติ เราจึงไม่จำเป็นต้องต่อสายใหม่ ยกเว้นในกรณีที่เราย้าย object เข้าไปใน Structure สายต่อจะไม่ตามเข้าไปใน Structure นั้นด้วย และสายที่เราต่อไว้เดิมก็จะกลายเป็นสายเสีย เราจะกล่าวถึง Structure ในภายหลัง

การสร้าง Object โดยอัตโนมัติ นอกเหนือจากการสร้าง Object ไม่ว่าจะเป็นค่าคงที่ ค่า Control หรือ Indicator บน Front Panel แล้วต่อสายด้วยตนเองภายใน Block Diagram เราอาจจะสร้าง Object บน Front Panel จากภายใน Block Diagram ได้ โดยที่ terminal ของ node ที่ต้องการ Input/Output ให้ใช้ Pop-up menu ของ terminal นั้นแล้วเลือก Create Constant หรือ Create Control หรือ Create Indicator เพื่อสร้าง Input/Output ให้กับ terminal นั้นโดยอัตโนมัติ และ object ที่สร้างขึ้นจะเชื่อมสายเข้ากับ terminal นั้นโดยอัตโนมัติด้วย โดย LabVIEW จะเลือกลักษณะข้อมูลที่เหมาะสมที่สุดเชื่อมต่อเข้ากับ terminal นั้น

การใช้วิธีการนี้จะมีประโยชน์มากโดยเฉพาะกรณีที่เราจะต้องต่อสายหลายๆสายเข้าสู่ node หรือ subVI ซึ่งอาจมี input เป็นค่าข้อมูลลักษณะต่างๆกันหลายแบบ

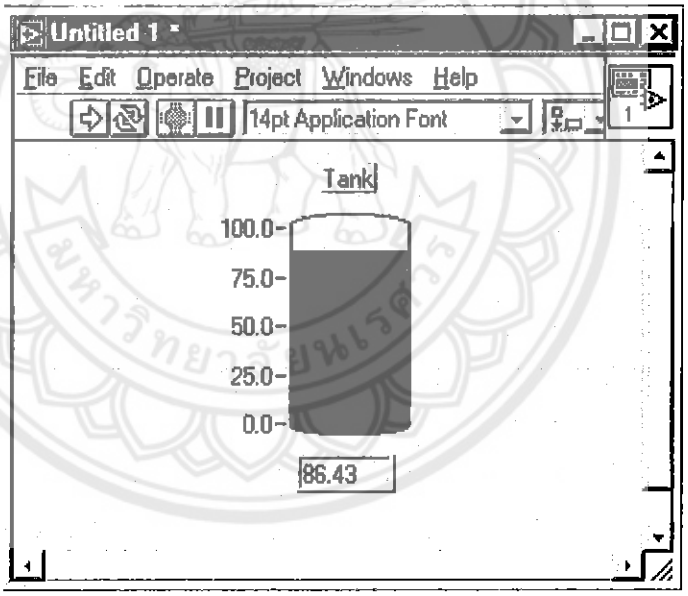


กิจกรรมนี้จะเป็นการฝึกหัดให้เราได้ทดลองการเขียน VI เสมือนกับว่าเราได้มีการอ่านข้อมูลจากภายนอกเข้าสู่คอมพิวเตอร์โดยเราจะใช้ VI สำเร็จรูปตัวหนึ่งแทนการวัดข้อมูลจริง VI นี้ชื่อ Demo Voltage Read.VI ซึ่งเป็น VI ที่สมมุติการอ่านค่า Voltage จาก DAQ Board โดย Output ของฟังก์ชันนี้จะเป็นค่า Voltage ทีละ 1 ค่าที่ได้จากข้อมูลที่มีการบันทึกไว้ล่วงหน้า สำหรับ Input ของ VI นี้ไม่มีผลต่อการประเมินค่าเป็นเพียงสร้างสภาวะให้ผู้ใช้ได้มีความรู้สึกเหมือนกับใช้ DAQ Board จริงๆ อยู่เท่านั้น เราจึงไม่จำเป็นต้องต่อ input ในที่นี้



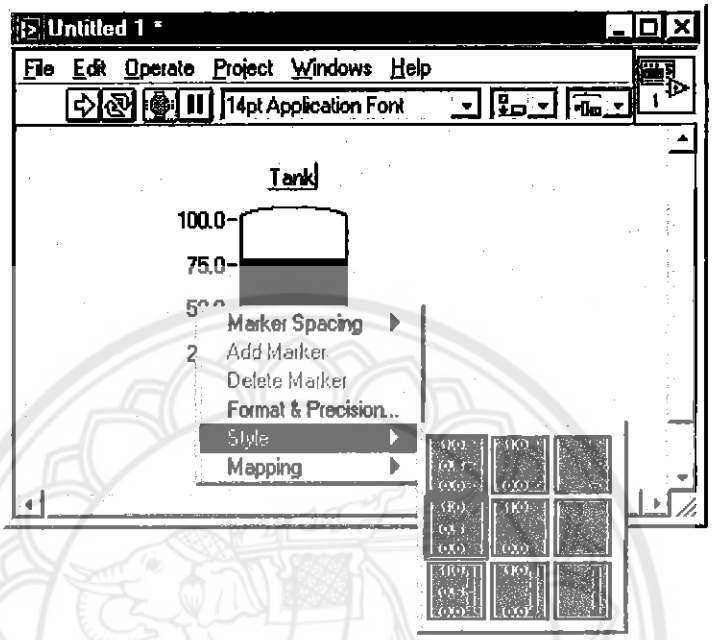
เราสามารถหา VI นี้จะอยู่ภายใต้ Tutorial subpalette ของ Function Palette ขั้นตอนของกิจกรรมนี้มีดังนี้

1. สร้าง front panel ดังแสดงในรูป



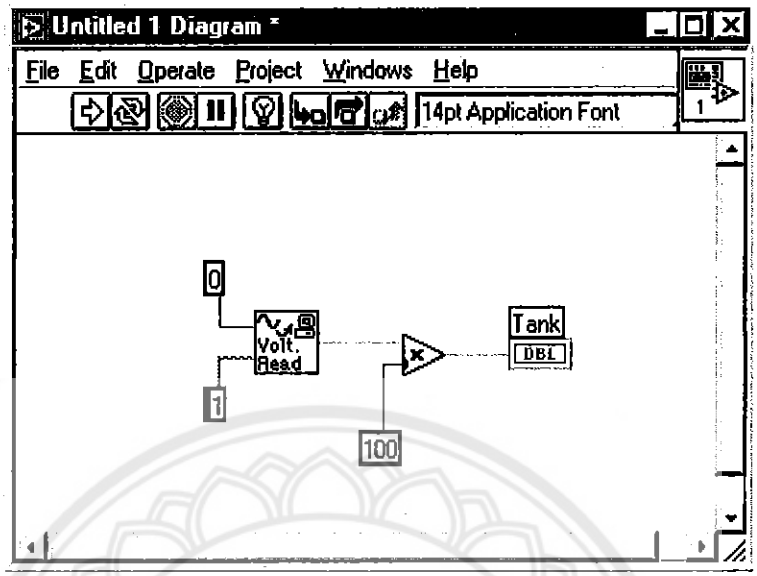
2. สำหรับขั้นตอนการเปลี่ยนแปลงตัวเลขที่ระดับน้ำในถัง ซึ่งเดิมค่าเริ่มต้นเป็น 0 ถึง 10 เราสามารถเปลี่ยนแปลงได้ โดยใช้ Operating Tool เลื่อนไปบริเวณตัวเลขที่ต้องการเปลี่ยน เช่นในกรณีนี้เลื่อนไปที่เลข 10 แล้วกดเมาส์ปุ่มซ้าย การแสดงตัวเลขจะเปลี่ยนไปอยู่ในลักษณะที่พร้อมจะสามารถแก้ไขได้ ใส่ตัวเลขที่ต้องการ เช่นในที่นี้เราใส่ 100 และเมื่อแก้ไขแล้วกด Enter ตัวเลขทั้งหมดจะเปลี่ยนและปรับค่ามาอยู่ในช่วง 0-100 โดยอัตโนมัติ

3. ลองใช้ Pop-Up menu ของส่วนเฉพาะที่เป็นตัวเลข คือเลื่อนเมาส์ไปบริเวณตัวเลขแล้วกดเมาส์ปุ่มขวาเพื่อดู Pop-Up menu และเลือกรูปแบบการจัดวางมาตรแบบต่างๆ โดยใช้คำสั่ง **Style** หรืออาจลองใช้คำสั่งอื่นๆ เช่น Marker Spacing ▶ หรืออื่นๆก็ได้



4. ไปที่ Block Diagram วาง function ชื่อ Demo Voltage Read.VI และฟังก์ชันการคูณลงไปใน Block Diagram จากนั้นเลือก Wiring Tool แล้วกลับมาที่ input terminal ของ Demo Voltage Read.VI แล้วลองใช้คำสั่ง Create Constant จาก Pop-up menu ของแต่ละ terminal เพื่อสร้าง input ให้กับ terminal ทั้งสอง

5. สร้าง Block Diagram ตามรูป



6. โดย node ชื่อ Volt Read นั้นหาได้จาก Function-Tutorial ซึ่ง Node นี้ ในขณะนี้เราสมมุติว่าเป็นค่าที่ได้ออกมาจากเครื่องมือวัด โดยจะมีค่าอยู่ระหว่าง 0 ถึง 1 สำหรับค่าตัวเลข 1 และ String 0 นั้น เป็นการสมมุติว่าให้อ่านข้อมูลจาก Analog input channel 0 โดยใช้อุปกรณ์ 1 เป็นตัวอ่าน แม้ว่าในความเป็นจริง ในตัวอย่างนี้เราไม่จำเป็นต้องต่อเชื่อม input ที่เป็นค่าคงที่และ String ก็ได้ เพราะ node นี้เป็นเพียงการสมมุติการอ่านค่าจาก DAQ Card แต่เพื่อให้เราคุ้นเคยกับการกำหนดการอ่านค่าจาก DAQ Card จึงได้กำหนด input ไว้อย่างนี้

7. ลอง Run VI นี้อย่างต่อเนื่อง สังเกตดูการเปลี่ยนแปลงของ Indicator ลองปรับแต่ง Front Panel ให้เป็นไปตามที่คุณชอบ โดยใช้คำสั่ง หรือ Control ต่างๆ ที่ได้กล่าวมาแล้ว

8. เราอยากให้คุณลองใช้ Coloring Tool เพื่อปรับแต่งสีต่างๆ บน Front Panel

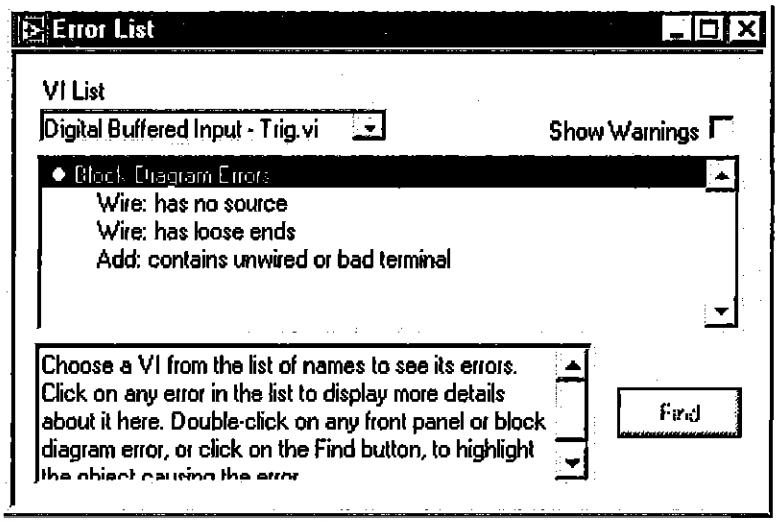
9. เลิกการทำงาน คุณไม่จำเป็นต้อง Save VI นี้ก็ได้

5.. Debugging

Bug หรือแมลง ในภาษาคอมพิวเตอร์หมายถึงการเกิดข้อผิดพลาดขึ้นในโปรแกรม คำนี้มาจากคอมพิวเตอร์ในยุคแรกๆ ที่ยังมีขนาดใหญ่และใช้วงจรไฟฟ้าจำนวนมากอยู่ ในการเขียนโปรแกรมครั้งหนึ่งปรากฏว่าเกิดความผิดพลาดในการทำงานของโปรแกรมขึ้น หลังจากตรวจสอบรายละเอียดทั้งในส่วนของโปรแกรมและเครื่องคอมพิวเตอร์แล้วพบว่ามีแมลงเข้าไปอยู่ในเครื่อง จึงทำให้เกิดไฟฟ้าลัดวงจร อันเป็นเหตุให้การทำงานของโปรแกรมผิดพลาดไป เราจึงได้ใช้คำว่า Bug แทนการเกิดความผิดพลาดขึ้นในโปรแกรมคอมพิวเตอร์ต่อมาจนถึงปัจจุบัน แม้ว่าคำนี้ในปัจจุบันจะหมายถึงความผิดพลาดที่เกิดขึ้นภายในโปรแกรมที่เขียนขึ้นมาเท่านั้น และคำว่า Debug จะหมายถึง การแก้ไขความผิดพลาดของโปรแกรมนั้น

Broken VI เป็นคำที่ใช้เรียก VI ที่ไม่สามารถทำงานได้ ซึ่งขณะนั้นปุ่ม Run จะเป็นรูปลูกศรหัก ซึ่งหมายความว่าในขณะนี้ VI ของเรามีปัญหาหรือข้อผิดพลาดอย่างใดอย่างหนึ่ง ซึ่งเหตุการณ์นี้ถือเป็นเรื่องปกติเพราะ VI จะอยู่ในสภาพที่เป็น Broken VI ถ้าหากว่าเราอยู่ในระหว่างขั้นตอนการต่อสายอยู่ และเมื่อเราต่อสายเสร็จและไม่มีข้อผิดพลาดเกิดขึ้นแล้ว VI ก็จะกลับเข้าสู่สภาพพร้อมทำงานปกติได้ สำหรับในกรณีที่เราได้ต่อสายผิดไว้และมีสายเสียเกิดขึ้น เราจะต้องแก้ไขหรือกำจัดสายเสียออกไปจาก Block Diagrams เสียก่อน เพื่อกำจัดสายที่ไม่ต้องการออกไป เพราะ LabVIEW จะไม่ยอมให้มีสายที่ต่อผิดหรือต่อสายไม่ครบ (คือไม่มีจุดเริ่มต้นหรือจุดสิ้นสุดที่แน่นอน) อยู่ภายใน Block Diagram แม้ว่าเราจะไม่ต้องการให้ข้อมูลไหลไปตามสายเหล่านั้นก็ตาม ขั้นตอนการลบสายเสียออกที่ง่ายที่สุดคือใช้คำสั่ง Remove Bad Wires จากภายใต้ Edit Menu หรือใช้ <Ctrl>-B

นอกเหนือจากการลบสายต่อที่เสียออกจาก Block Diagram แล้ว ยังอาจเกิดความผิดพลาดได้จากกรณีอื่นๆ ได้อีก ซึ่งเราสามารถจะมองหาจุดผิดพลาดเหล่านั้นได้ด้วยการใช้คำสั่งให้แสดงข้อผิดพลาดทั้งหมดที่เกิดขึ้น หน้าต่างที่แสดงความผิดพลาดนี้เรียก Error List การที่จะให้หน้าต่าง Error List แสดงขึ้นมานั้นทำได้โดยเลือกคำสั่ง Show Error List จาก Windows Menu หรือใช้เมาส์กดบริเวณปุ่มรูปลูกศรหักที่อยู่บน Tool Bar ก็ได้ จากนั้นหน้าต่าง Error List จะปรากฏขึ้นและแสดงความผิดพลาดทั้งหมดที่มีอยู่ภายใน Block Diagram ตัวอย่างของหน้าต่างนี้แสดงในรูป



ถ้าหากเราต้องการจะทราบว่า การเกิดความผิดพลาดแต่ละหัวข้อมีสาเหตุมาจากเรื่องใด เราสามารถเลือกหัวข้อนั้นภายใน Error List เพื่อให้หน้าต่างด้านล่างอธิบายความเป็นไปได้ที่ทำให้เกิดความผิดพลาดนั้นขึ้น สำหรับการที่จะหาตำแหน่งของความผิดพลาดบน Block Diagram สามารถทำได้โดยใช้เมาส์กดที่รายการผิดพลาด 2 ครั้ง (double click) หรือใช้เมาส์กดปุ่ม Find บนหน้าต่าง Error List จะทำให้ LabVIEW กลับไปแสดง Block Diagram และแสดงตำแหน่งของความผิดพลาดด้วยการเน้นตำแหน่งนั้นด้วยเส้นเป็นกรอบรอบส่วนนั้นไว้

การแจ้งเตือน การแจ้งเตือนหรือ Warning เป็นเครื่องมืออีกชนิดหนึ่งที่ช่วยการแก้ไขข้อผิดพลาดที่เกิดขึ้นใน VI เราสามารถให้ LabVIEW แสดงคำเตือนได้ โดยการเลือก Show Warnings บนหน้าต่าง Error List การเตือนของ LabVIEW ไม่ได้หมายความว่าเกิดความผิดพลาดขึ้นใน VI เพียงแต่ได้เกิดมีสิ่งที่ไม่มีเหตุผลที่ LabVIEW เข้าใจเกิดขึ้นใน VI เท่านั้น ตัวอย่างเช่น อาจมี Control Terminal ที่ไม่ได้ต่อเข้ากับส่วนใดเลย แม้ว่าจะไม่เป็นความผิดพลาด แต่ LabVIEW ก็สงสัยว่าจะมี object นั้นอยู่เพื่อเหตุผลอะไร

ถ้าหากเราเลือก Show Warnings แล้วเกิดสิ่งที่ทำให้ LabVIEW เตือน เราจะเห็นว่ามี ปุ่มเตือน (Warning Button) เกิดขึ้นบน Toolbar หากเราต้องการทราบว่า LabVIEW เตือนอยู่ เราก็สามารถกดที่ปุ่มนั้น เพื่อดูรายการเตือนได้

นอกจากนั้นเรายังสามารถเลือกให้แสดงการเตือนได้ โดยบน Menu Bar เลือก Edit ▶ Preferences... และบน dialog box การเลือก Show Warning In Error Box By Default เพื่อให้มีการแสดงการเตือนทุกครั้งที่มีความไม่สมบูรณ์เกิดขึ้นกับ VI ของเรา

การเข้าสู่ VI ทีละขั้น เนื่องจาก LabVIEW เป็นโปรแกรมที่ใช้การเดินทางของข้อมูลในการควบคุมการประมวลผลของโปรแกรม ซึ่งต่างจากการทำงานทีละบรรทัดของโปรแกรมตัวหนังสือทั่วๆ ไป ดังนั้นการดูการทำงานทีละลำดับอาจจะไม่เหมือนกับโปรแกรมที่ทำงานทีละบรรทัด

การที่เรากำหนดให้ LabVIEW ทำงานที่ละขั้นเป็นการกำหนดให้ LabVIEW ประมวลผลบน Block Diagram ทีละ node ได้ คำว่า node ในที่นี้รวมถึง subVI, Structure, Code Interface Node (CIN), Formula Nodes และ Attribute nodes ซึ่ง object เหล่านี้เราจะกล่าวถึงต่อไป สำหรับการกำหนดให้ VI ทำงานทีละ node สามารถทำได้หลายวิธีดังต่อไปนี้

ใช้คำสั่งใดคำสั่งหนึ่งของการทำงานทีละขั้นแทนที่จะใช้คำสั่ง Run การทำงานทีละขั้นประกอบด้วย

Step Into จะทำงานลำดับแรกของ node, subVI หรือของ Structure แล้วหยุดรอ (pause) เพื่อจะทำขั้นต่อไปของ subVI หรือ Structure นั้น

Step Over จะทำงานของ subVI หรือ Structure จากขั้นแรกจนกระทั่งเสร็จการทำงานที่ node, subVI หรือ Structure นั้นแล้วหยุดรอก่อนที่จะเข้าสู่ node ต่อไป

Step Out ให้โปรแกรมเสร็จสิ้นการประมวล Block Diagram, Structure หรือ VI ในขณะนั้นแล้วจากนั้นจึงหยุดรอ

การใส่ Breakpoints ในระหว่างการไหลของข้อมูล การใส่ Breakpoints ไม่ได้หมายความว่าให้ยกเลิกการทำงานของ VI แต่หมายถึงได้ VI หยุดรอการประมวลผลที่ตำแหน่งนั้นเพื่อเราจะสามารถแก้ไขการทำงานโปรแกรมได้ Breakpoint จะมีประโยชน์เมื่อเราต้องการจะตรวจดู input ที่เข้าสู่ VI, node หรือ สายต่อในระหว่างการประมวลผล เมื่อข้อมูลเดินทางมาถึง Breakpoint ปุ่ม Pause จะทำงาน เราอาจจะตรวจสอบข้อมูลด้วย Probe, อาจจะใช้ Single-Step ในการประมวลผลต่อไป VI เพื่อสังเกตการเปลี่ยนแปลงค่าบน Front Panel หรือถ้าทุกอย่างเป็นปกติเราอาจเปลี่ยนให้ LabVIEW ทำงานต่อไปโดยกดที่ปุ่ม Pause หรือปุ่ม Run

วิธีการตั้งจุด brake ทำได้โดยเลือก Breakpoint tool จาก Tools palette จากนั้นเลื่อนไปบน block diagram เมื่อได้ตำแหน่งที่ต้องการจะวางแล้วให้กดเมาส์ breakpoint จะปรากฏขึ้นบน Block Diagram แต่ถ้าหากเรากดซ้ำในจุดที่มี breakpoint อยู่แล้ว breakpoint นั้นจะหายไป

การทำงานของ Breakpoint จะแตกต่างไปตามชนิดของวัตถุตำแหน่งที่วาง Breakpoint ลงไปกล่าวคือ

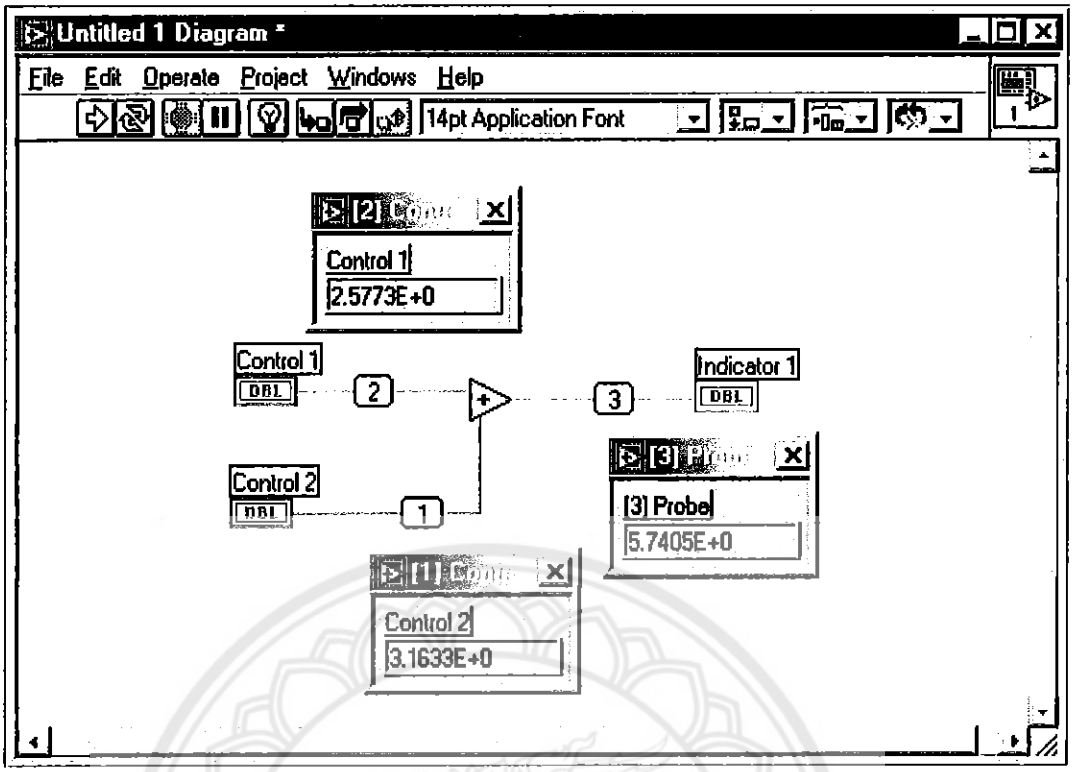
ถ้าวาง Breakpoint ลงบน Block Diagram จะมีกรอบสีแดงขึ้นรอบ Diagram และการหยุดรอจะเกิดขึ้นเมื่อการทำงานของ Diagram นั้นสิ้นสุดลง

ถ้าวาง Breakpoint ลงบน Node จะเกิดกรอบสีแดงรอบ Node นั้น และการทำงานจะหยุดรอก่อนที่ข้อมูลจะเข้าไปประมวลผลใน Node นั้น

ถ้าวาง Breakpoint ลงบนสายต่อ จะเกิดวงกลมสีแดงขึ้นบนสายต่อที่ตำแหน่งนั้น และถ้าหากมี Probe ต่ออยู่กับสายนั้น จะเกิดการรอบแดงขึ้นรอบกรอบของ Probe นั้น การประมวลข้อมูลจะหยุดรอเมื่อมีข้อมูลวิ่งผ่านเข้ามาที่สายต่อนั้น

เมื่อ VI ถูกระงับการทำงานเนื่องจาก breakpoint หน้าต่างของ Block Diagram จะ Active ขึ้นมาอยู่บนสุด ขณะที่ตำแหน่งที่ข้อมูลกำลังหยุดรออยู่นั้นจะมีการเน้นสีและเลื่อนกระพริบ เพื่อให้เราเห็นได้อย่างชัดเจนว่าขณะนี้ VI หยุดรอที่ตำแหน่งใด

การใช้ Probe เปรียบเหมือนกับเครื่องมือวัดข้อมูล ที่ใช้จุ่มลงไปบนสายต่อเพื่อแสดงค่าข้อมูลที่กำลังไหลผ่าน โดยปกติเราจะใช้เพื่อตรวจสอบความถูกต้องในการประมวลผลใน VI ว่ามีค่าเป็นอย่างไร เมื่อเราวาง Probe ลงไปบนสายต่อจะทำให้เกิดหน้าต่างของ Probe นั้นขึ้น และเมื่อ VI อยู่ในโหมดการทำงาน เมื่อข้อมูลเดินทางผ่านสายต่อที่วาง Probe ไว้ ค่าของข้อมูลจะแสดงบนหน้าต่างแสดงผลของ Probe นั้น จะเปลี่ยนไปตามค่าของข้อมูลที่เดินทางผ่านสายต่อนั้น ตามตัวอย่างที่แสดงในรูป



วิธีการวาง Probe ลงบนสายต่อ ทำได้โดย เลือก Probe Tool จาก Tools Palette จากนั้นเลือกกดเมาส์ลงบนสายต่อเส้นที่ต้องการภายใน Block Diagram จากนั้น LabVIEW จะแสดงเครื่องหมาย Probe เป็นรูปสี่เหลี่ยม และหมายเลขของ Probe อยู่ตรงกลาง พร้อมปรากฏหน้าต่างแสดงค่าของ Probe นั้น เราสามารถวาง Probe ไว้บน Block Diagram ได้เป็นจำนวนมากเท่าใดก็ได้เท่าที่เราต้องการ

ข้อมูลที่แสดงบน Probe Windows ไม่สามารถจะเปลี่ยนแปลงได้ แต่เราสามารถเปลี่ยนลักษณะของการแสดงได้ เช่นจะให้แสดงเป็นตัวเลข กราฟ หรืออื่นวิธีการคือการใช้คำสั่ง Custom Probe จาก Pop-up menu ของสายต่อที่ probe วางอยู่

การลบ Probe ทำได้โดยปิดหน้าต่างแสดงผลของ Probe นั้น และเครื่องหมายของ Probe บน Block Diagram ก็จะหายไปอย่างอัตโนมัติ

การใช้ Probe ในการตรวจสอบการทำงานของ VI เป็นสิ่งที่มีประโยชน์มาก โดยเฉพาะการใช้ Probe ที่ออกมาจาก subVI เพราะจะทำให้เราทราบได้ว่าค่าที่ออกมาจาก subVI นั้นเหมาะสมหรือสมควรจะเป็นไปตามนั้นหรือไม่ เพราะการผิดพลาดของการเขียนโปรแกรมส่วนหนึ่งจะมาจากการเขียน subVI ผิดพลาดด้วย

การเน้นเส้นทางข้อมูล เราสามารถดูเส้นทางการเดินทางของข้อมูลบน Block Diagram เพื่อสังเกตดูการเปลี่ยนแปลงค่าของข้อมูลในระหว่างที่ผ่านไปตาม node ต่างๆ และเส้นทางการเดินทางของข้อมูลตามสายต่อต่างๆว่ามีลำดับเป็นอย่างไร การใช้คำสั่งให้การประมวลข้อมูลเป็นไปในลักษณะนี้ใน LabVIEW เราเรียกว่า Execution Highlighting และสามารถสั่งให้ LabVIEW ทำการประเมินผลเช่นนี้ได้โดยใช้เมาส์กดปุ่มดังกล่าวบน Toolbar

เมื่อเราให้ LabVIEW ประมวลผลด้วยวิธีการนี้ การเดินทางของข้อมูล โดยจะแทนด้วย รูปฟองอากาศเล็กๆ เคลื่อนที่ไปตามสายต่อต่างๆ เราจะพบว่าการประมวลผลด้วยวิธีนี้ จะลดความเร็วในการประมวลผลไปอย่างมาก

ค่าต่างๆ บน node จะมีการแสดงผลโดยอัตโนมัติได้ หากเราเลือกตัวเลือก Auto Probe During Execution Highlighting ภายใต้ Preference... จาก Debugging Menu เมื่อเราเลือกตัวเลือกนี้ จะเกิด Probe อัตโนมัติขึ้นที่ Node และจะแสดงค่าเมื่อข้อมูลใหม่เดินทางมาถึง

เราสามารถยกเลิกการทำงานในลักษณะเช่นนี้ได้โดยใช้เมาส์กดปุ่ม Execution Highlighting บน Toolbar อีกครั้งหนึ่ง

6. Creating SubVI

ส่วนที่มีประโยชน์อย่างมากส่วนหนึ่งของ LabVIEW ก็คือโปรแกรมหรือ VI ใดๆ สามารถสร้างขึ้นแล้วเรียกใช้จาก VI อื่นๆ ได้ VI ที่ถูกเรียกใช้จะกลายเป็น subVI ไป และใน Block Diagram ของ VI หนึ่ง สามารถเรียกใช้ subVI ได้หลายชุด และสามารถจะเรียก subVI หนึ่งๆ มาใช้กี่ครั้งก็ได้บน VI หลักเดียวกันนั้น

เราสามารถทำให้ VI ใดๆ ทำหน้าที่เป็น subVI ใน Block Diagram ของ VI อื่นได้ เพียงแต่เราต้องจัดการสร้าง Icon และ Connector ให้กับ subVI นั้นให้เหมาะสมเสียก่อน อย่างไรก็ตาม VI ไม่สามารถเรียกตัวมันเองเป็น subVI ได้

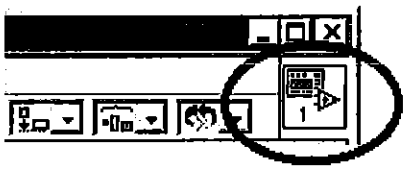
เมื่อเราจะเลือกใช้ subVI ของเรา สามารถทำโดยเลือก Select a VI... จาก Functions palette จากการเลือกตัวเลือกนี้ เราจะได้ File dialog box ซึ่งเราสามารถเลือก VI ใดๆ ที่ได้รับการเก็บอยู่ภายในเครื่องมาใช้ เมื่อเราเลือกแล้ว Icon ของ subVI จะปรากฏใน Block Diagram

การสร้าง subVI จาก VI ที่มีอยู่จะประกอบด้วย 2 ส่วนคือการสร้าง Icon และการสร้าง Connector สำหรับการที่เราจะสร้าง subVI จาก VI นั้นมีข้อควรคำนึงอยู่อย่างหนึ่งคือเราควรจะสร้างและทดสอบการทำงานของ VI ให้ถูกต้องเรียบร้อยเสียก่อน แล้วจึงสร้างเป็น subVI ขึ้นมา เพราะจะลดปัญหาในการ Debug ในส่วนของ VI หลักที่เรียก subVI นี้ไปใช้งานได้เป็นอย่างมาก

การสร้าง icon ทุก subVI จะต้องมี Icon เพื่อนำไปใช้ใน block diagram ของ VI ที่จะเรียก subVI นี้ไปใช้งาน โดย Icon นี้หมายถึงรูปภาพเล็กๆที่มีสัญลักษณ์ที่ผู้สร้างออกแบบขึ้น

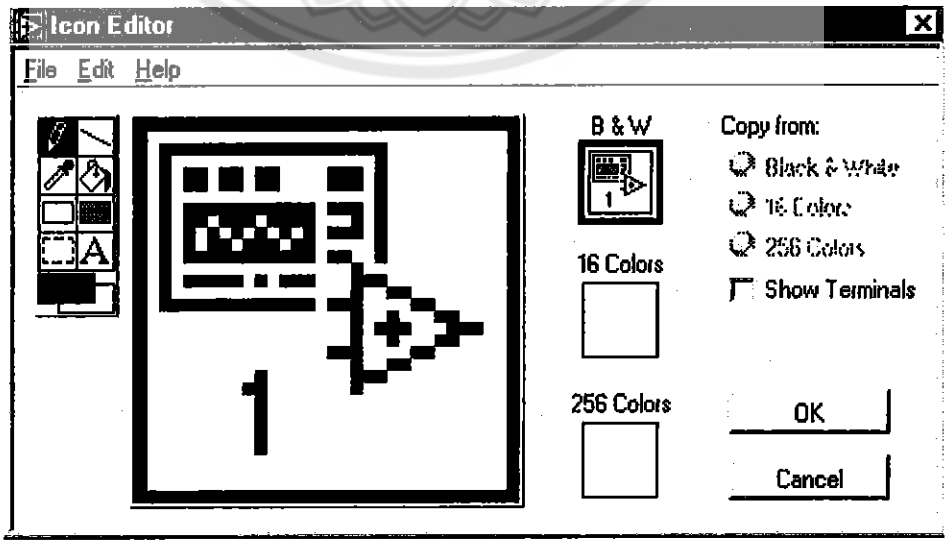
โดยปกติในการสร้าง VI ใหม่ทุกครั้ง LabVIEW ได้สร้าง icon ให้กับ VI นั้นๆ อยู่แล้ว ผู้ใช้อาจนำไปใช้ได้เลย อย่างไรก็ตามรูปของ icon ที่ LabVIEW สร้างให้จะมีลักษณะเป็นมาตรฐาน ถ้าหากเราไม่แก้ไขส่วนใดเลย อาจจะทำให้เกิดรูปซ้ำกับ icon อื่นๆ ได้ และเมื่อเราใช้ subVI จำนวนมากใน VI หลักของเรา

รูปของ Icon จะปรากฏที่มุมขวาบนของหน้าต่าง Front Panel ซึ่งลักษณะรูปของ Icon เริ่มต้นจะเป็นสัญลักษณ์ของ LabVIEW และมีตัวเลขประกอบ โดยตัวเลขจะเป็นลำดับการสร้าง VI ตามชื่อ Untitled 1, Untitled 2 ตามลำดับ ดังมีตัวอย่างในรูป



การแก้ไข Icon ที่เป็นรูปเริ่มต้น ขั้นแรกเราต้องอยู่ในโหมดการแก้ไข (Edit Mode) จากนั้นใช้เมาส์ขวากดที่ Icon เพื่อเรียก Pop-up menu ขึ้นมาก แล้วเลือก Edit Icon... หรือเราอาจกดเมาส์ 2 ตัวบน Icon นั้นก็ได้ ซึ่งหลังจากคำสั่งนี้ LabVIEW จะนำเราเข้าไปสู่ Icon Editor

หน้าต่าง Icon Editor จะมีลักษณะคล้ายกับโปรแกรมวาดรูปทั่วๆ ไปคือจะมีเครื่องมือ วาดรูปต่างๆ เขียน ลากเส้น ดูดสี แสงสีพื้น เขียนสี่เหลี่ยม ตัวอักษร และเลือกสี เป็นต้น สำหรับเครื่องมือลบเส้นไม่มีในที่นี้ เพราะการลบก็คือการเขียนทับด้วยสีพื้นนั่นเอง



นอกจากนั้นเรายังสามารถสร้าง Icon ต่างกันในโหมดสีต่างกันคือ สีขาวดำ (B&W), 16 สี หรือ 256 เพื่อใช้แสดงผลตามโหมดของจอ ในการแก้ไขครั้งแรก Icon จะสร้างขึ้นในโหมดสีขาวดำเท่านั้น จะไม่มีการสร้างในโหมดสีอื่น อย่างไรก็ตามเราสามารถคัดลอก Icon แล้วนำมาเปลี่ยนสีให้อยู่ในโหมดที่ต้องการได้ โดยใช้คำสั่ง Copy From... บนหน้าต่างนี้ เช่นถ้าหากว่าเรามีรูปในโหมดสีขาวดำอยู่ก่อนแล้วต้องการจะสร้าง Icon สำหรับโหมด 16 สี อันดับแรกให้เลือกเข้ามาอยู่ในโหมด 16 สีก่อนจากนั้นจึงใช้คำสั่ง Copy From... Black & White เราจะได้รูปขาวดำในโหมด 16 สี จากนั้นเราจึงแก้ไขเพิ่มเติมให้มีสีต่อไป

สำหรับ Icon ที่ออกแบบในแต่ละโหมดสีจะได้รับการเลือกใช้ในแต่ละโหมดของการแสดงผลของจอในขณะนั้น ไม่ว่าจะเป็น monochrome, 16-สี, หรือ 256-สี อย่างไรก็ตามเราจะต้องมี icon ในโหมดขาว-ดำ เพราะโหมดสีอื่นจะไม่แสดง Icon ในโหมดสีอื่นบน Palette Menu และถ้าหากไม่มี Icon ในโหมดสีขาวดำ LabVIEW จะแสดงเป็น Icon ว่างใน Palette Menu ต่างๆ

หลังจากสร้างและแก้ไข Icon ได้ตามต้องการ แล้วให้กดปุ่ม OK เพื่อเสร็จสิ้นการสร้าง Icon

การกำหนด Connector ก่อนที่เราจะใช้ VI ให้เป็น subVI เราจะต้องมีการกำหนด Connector Terminal ของ subVI นั้นเสียก่อน หน้าที่ของ Connector นี้จะเป็นการกำหนดให้ข้อมูลเข้าและออกจาก subVI โดยเริ่มต้น LabVIEW จะกำหนดให้ส่วนที่เป็น Control ใน VI จะได้รับการกำหนดให้เป็น Input Terminal ของ subVI และส่วนที่เป็น Indicator จะได้ถูกกำหนด ให้เป็น Output Terminal ของ subVI

ในการกำหนด Connector ใช้ Pop-up menu บน icon แล้วเลือก **Show Connector** จากนั้นรูปของ Icon จะเปลี่ยนเป็น Connector

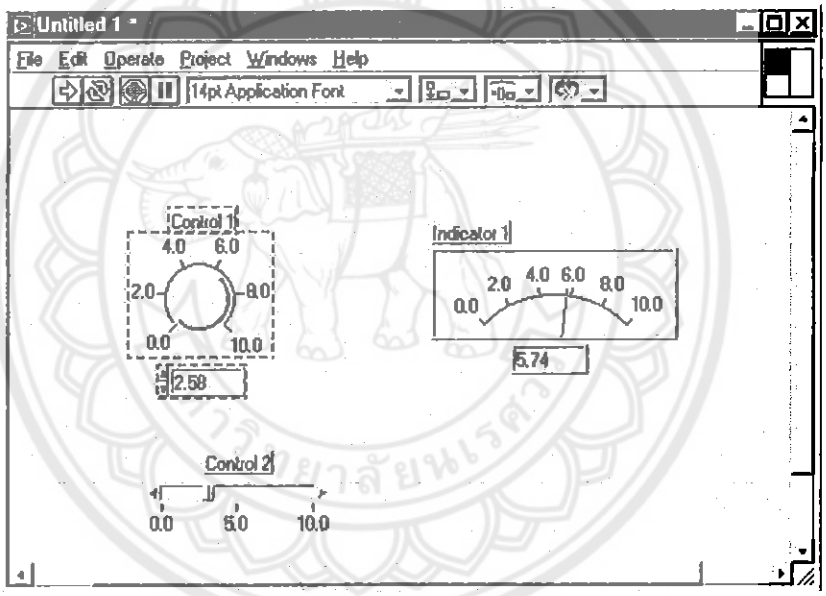
สำหรับจำนวนของ terminal เริ่มต้นที่ปรากฏขึ้นนั้น LabVIEW จะคำนวณจากจำนวนของ Control และ Indicator ทั้งหมดบน Front Panel โดย Input Terminal จะวางอยู่ทางซ้ายมือและ Output Terminal จะวางอยู่ทางขวามือ อย่างไรก็ตามถ้าหากเราต้องการรูปแบบที่แตกต่างออกไป

เราจะสามารถเลือกได้โดยจาก Pop-up menu บน Connector แล้วเลือกใช้คำสั่ง Patterns นอกเหนือจากนั้นเราสามารถหมุนกลับตำแหน่งของ Connector ได้ถ้าหากว่าค่าเริ่มต้นที่ LabVIEW เลือกให้ไม่เป็นไปตามที่เราต้องการ

หลังจากสร้าง Connector เสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการกำหนด terminal ต่างๆ ว่าจะหมายถึง Control หรือ Indicator ตัวใดบน Front Panel ขั้นตอนการกำหนด Connector มีดังต่อไปนี้

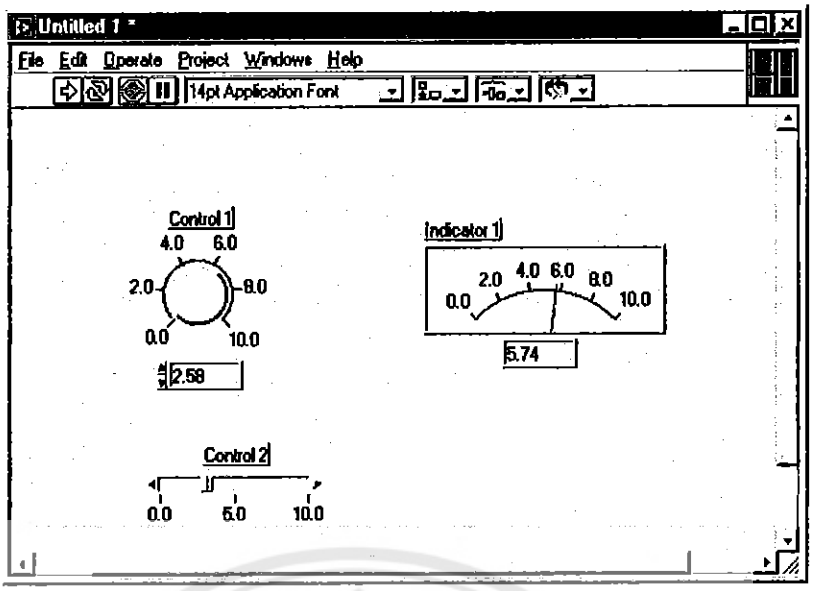
1. ทำให้ Front Panel อยู่ในสภาพพร้อมทำงาน

2. กดเมาส์บน terminal ใน Connector เราจะพบว่าตัวชี้จะเปลี่ยนเป็นรูป Wiring Tool และ terminal จะเปลี่ยนเป็นสีเทาแดง (สีอ่านเปลี่ยนไปตามรุ่นของเครื่องคอมพิวเตอร์ที่ใช้อยู่)



3. กดเมาส์บน Control หรือ Indicator ที่เราต้องการให้ terminal นั้นจะเชื่อมต่อเข้าเมื่อเรากดเมาส์แล้ว จะเกิดเส้นประเคลื่อนที่รอบกรอบของ Control หรือ Indicator นั้น

4. กดเมาส์ลงบนที่ว่างบน front panel เส้นประจะหายไป ส่วนสีเทาแดงบน connector จะกลายเป็นสีส้มซึ่งหมายความว่าเรากำหนด control ให้ terminal เรียบร้อยแล้ว



5. ถ้าหาก terminal ยังเป็นสีขาว แสดงว่าเรายังไม่ได้กำหนด Connection ได้ถูกต้อง ส่วนการกำหนด terminal อื่นก็ให้ทำซ้ำขั้นตอนที่ผ่านมา แต่ละ subVI จะมี terminal ได้ถึง 28 ช่อง

6. ถ้าหากเราต่อช่องผิดพลาด เราสามารถแก้ไขได้โดยการใช้คำสั่งจาก Pop-up menu แล้วเลือก Disconnect สำหรับ terminal นั้นๆ หรือ Disconnect All เพื่อยกเลือกทั้งหมด

เมื่อกำหนด object ให้กับ Connector เสร็จเรียบร้อยแล้วเราสามารถให้ Connector กลับไปแสดงรูป Icon ได้อีกโดยการใช้คำสั่ง Show Icon จาก Pop-up menu ของ Connector

การสร้าง subVI จากการเลือกจากบางส่วนของ VIบางครั้งเมื่อเราสร้าง VI ไปแล้วปรากฏว่า VI ของเรามีขนาดใหญ่ขึ้นเรื่อยๆ หรือมีส่วนที่มีลักษณะซ้ำๆ กันอยู่หลายจุด ทำให้เราคิดค่าเราน่าจะใช้ subVI แทน ซึ่ง LabVIEW ยอมให้เราสามารถที่จะทำเช่นนี้ได้ แม้ว่าค่อนข้างจะยุ่งยากบ้างก็ตาม

ขั้นตอนการเปลี่ยนบางส่วนใน Block Diagram เป็น subVI ก็คือ ด้วย Positioning Tool เลือกส่วนของ VI ที่ต้องการจะสร้างเป็น subVI วิธีเลือกใช้กดเมาส์ค้างไว้แล้วลากเมาส์ไปเรื่อยๆ ขนาดของกรอบสี่เหลี่ยมจะขยายไปเรื่อยๆ เมื่อเลือกส่วนของ VI ที่จะทำเป็น subVI ได้ตามที่ต้องการแล้ว จาก Edit menu เลือก SubVI From Selection เราจะเห็นว่า LabVIEW เปลี่ยนส่วนนั้นให้เป็น Icon และต่อสายให้เรียบร้อยเหมือนเดิม หากเรากดเมาส์ 2 ครั้ง บน icon เป็น LabVIEW จะแสดง Front Panel ของ subVI ทำให้เราสามารถแก้ไขเปลี่ยนแปลง Icon และ Connector ต่างๆ ได้ นอกเหนือจากนั้นเราสามารถ Save subVI นี้ภายใต้ชื่ออื่นได้อีกด้วย

อย่างไรก็ตามการใช้ subVI From Selection จะต้องใช้ด้วยความระมัดระวัง เพราะอาจจะเกิดปัญหาที่ไม่ได้คาดคิดหลายประการ เราแนะนำว่าหากจะสร้าง subVI ขอให้สร้างจาก VI ปกติแล้วทำการทดสอบการทำงานให้เรียบร้อยก่อนที่จะนำมาใช้งานSubVI Help : Recommended, Required และ Optional Inputs เมื่อเราเรียก Help Windows ขึ้นมาใช้กับ subVI หรือ node ใน block diagram คำอธิบายและการต่อสายจะปรากฏขึ้นโดยชื่อของ input จะอยู่ด้านซ้าย และ output จะอยู่ด้านขวา ในการต่อเชื่อมสายนั้นเราจะต้องนำข้อมูลที่มีลักษณะเดียวกันต่อเข้าด้วยกัน และเราควรจะกำหนดว่า terminal แต่ละช่องมีความจำเป็นในการต่อเชื่อมมากน้อยเพียงใด โดยแบ่งเป็น 3 ระดับ คือ

1. **Required** ผู้ใช้จำเป็นต้องต่อ terminal นี้ ไม่เช่นนั้น VI จะเป็น broken VI
2. **Recommended** แนะนำให้ต่อถ้าไม่ต่อ terminal นี้ VI จะสามารถ Run ได้ แต่จะปรากฏใน Warning ของ LabVIEW เพื่อเตือนว่าเราไม่ได้ต่อสายเข้ากับช่องนี้
3. **Optional** ช่องนี้จะต่อก็ได้หรือไม่ต่อก็ได้

การเลือกลำดับความสำคัญในการเชื่อมสายนี้ สามารถทำได้โดย Pop-up ที่ terminal นั้นๆ แล้วใช้คำสั่ง This Connection is ▶ แล้วเลือก Required, เมื่อเราเลือกตัวเลือกใดจะปรากฏเครื่องหมายถูกอยู่หน้าตัวเลือกนั้น สำหรับ Connection ที่ต่อออกจาก Indicator นั้นเราไม่สามารถกำหนดให้เป็น Required ได้ แต่อาจเลือกเป็น Recommended หรือ Optional ได้

7. Description

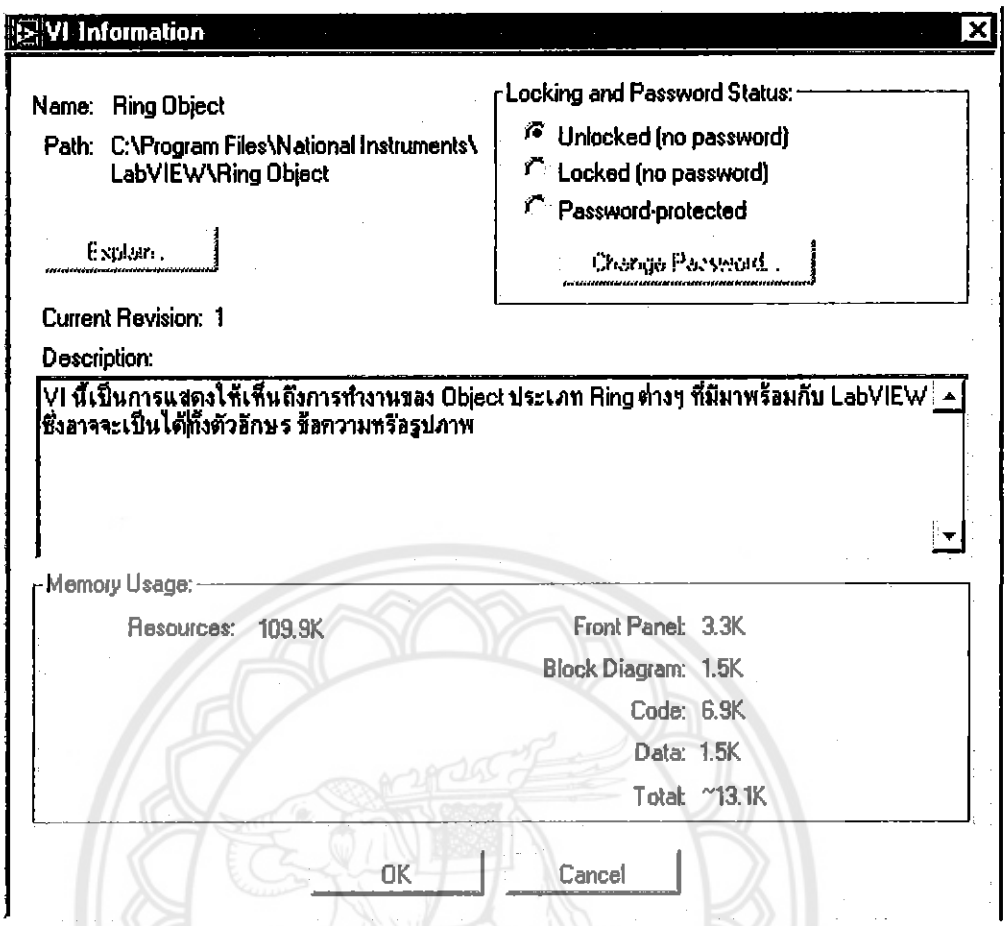
เมื่อเราสร้าง VI ขึ้นมา เราอาจต้องส่งให้ผู้อื่นนำไปใช้งานต่อ เรามีความจำเป็นที่จะต้องบอกให้ผู้ใช้ทราบว่าค่าต่างที่เราสร้างขึ้นมานั้นมีจุดประสงค์ใดหรือมีข้อจำกัดอะไรอยู่บ้าง นอกเหนือจากนั้นเมื่อผู้อื่นนำ VI ของเราไปใช้เป็น subVI เขาคงมีความต้องการที่จะทราบว่า VI ของเรามีจุดต่อต่างๆ เป็นอย่างไร และในหลายๆกรณีหากเราต้องนำ VI ที่เราสร้างขึ้นเองมาใช้งานก็มีโอกาสเป็นไปได้ว่าเราลืมไปแล้วว่า VI ของเรานั้นสร้างมาด้วยจุดมุ่งหมายหลักใดและมีข้อจำกัดใด ดังนั้นจะเป็นการดีถ้าหากว่าเราจะแสดงคำอธิบายไว้กับ VI ที่เราสร้างเสร็จเรียบร้อยแล้วบ้าง เพราะผู้ใช้จะได้มีความเข้าใจการทำงานของ VI ได้ง่ายขึ้น ในหัวข้อนี้จะกล่าวถึงวิธีการสร้างคำอธิบายต่างๆ ให้กับ VI ที่เราสร้างเสร็จเรียบร้อยแล้ว

การสร้างคำอธิบายสำหรับ VI ในการสร้าง VI ขึ้นมานั้น เราจะสามารถสร้างคำอธิบายได้ 2 แบบคือสร้างคำอธิบายให้กับแต่ละ object ที่อยู่บน Front Panel และอีกกรณีหนึ่งคือสร้างคำอธิบายให้กับ VI ทั้งหมด ขั้นตอนการสร้างคำอธิบายทั้งสองแบบสามารถทำได้ดังนี้

การสร้างคำอธิบายสำหรับแต่ละ Object ถ้าหากเราต้องการจะสร้าง คำอธิบายให้กับแต่ละ object ของ LabVIEW เช่น Control, Indicator, หรือ Function ให้เลือก Pop-up menu ของ object จากนั้นเลือก Data Operation ▶ Description... จากนั้น LabVIEW จะแสดง Dialog Box ขึ้น เพื่อให้เราพิมพ์คำอธิบายที่ต้องการลงไป เมื่อเราใส่ข้อความโดยสมบูรณ์ตามต้องการแล้ว กด OK เราสามารถแสดงและแก้ไขข้อความเหล่านั้นได้เมื่อเรียกหน้าต่าง Description ขึ้นมา ข้อความอธิบายนี้จะแสดงบน Help Windows ทุกครั้งที่เมาส์ลากผ่าน Control หรือ Indicator ที่เราใส่ข้อความไว้

อย่างไรก็ตามเมื่อเมาส์ลากผ่าน subVI หรือ Function จะไม่มีคำอธิบายที่เราใส่เข้าไปใน Description ปรากฏอยู่บน Help Windows เพราะว่าเราทำการบันทึกข้อมูลไว้คนละระดับ ถ้าหากเราต้องการดูคำอธิบายใน Help ของ subVI และ Function เราจำเป็นต้องเปิดหน้าต่างของ Description ขึ้นมาเพื่อดูเอง

การสร้างค่าอธิบายให้ VI เราสามารถทำคำอธิบายให้กับ VI ทั้งหมดได้ด้วยวิธีง่ายๆ เช่นกัน คือ เลือก Show VI Info... จาก Windows Menu ซึ่ง LabVIEW จะแสดงข้อมูลของ VI นั้น ดังที่แสดงในรูป



เราสามารถใช้หน้าต่าง VI Information นี้ทำหน้าที่ได้หลายๆ แบบตามนี้

- ใส่คำอธิบายการทำงานของ VI นี้เพื่อให้ผู้อื่นเข้าใจและถ้าหาก VI นี้ นำไปใช้เป็น subVI เมื่อเมาส์ผ่าน Icon ของ subVI ข้อความที่เราใส่เข้าไปนี้จะปรากฏใน Help window
- ดูรายการการเปลี่ยนแปลง VI ที่เกิดขึ้นนับจากการบันทึกครั้งสุดท้าย
- ดู Path หรือตำแหน่งที่ VI นี้เก็บอยู่บน disk
- ดูการใช้หน่วยความจำของ VI นี้ ทั้งที่ใช้ใน disk และ หน่วยความจำของเครื่อง

8. Save and Load VI

เมื่อเราสร้าง VI ขึ้นมาใช้แล้ว เป็นที่แน่นอนว่าเราจำเป็นต้องบันทึกข้อมูล (save) และนำข้อมูลหรือ VI นี้กลับมาใช้ใหม่ (Load) สำหรับการ Save และ Load ของ VI ใน LabVIEW ก็จะเหมือนกับโปรแกรมอื่นๆใน Windows ซึ่งเราคงทราบถึงวิธีการกันดีอยู่แล้ว อย่างไรก็ตามมีกรณีการ Save และ Load ที่เป็นกรณีการเลือกเฉพาะของ LabVIEW อยู่หลายกรณีซึ่งอาจจะแตกต่างจากโปรแกรมอื่น ในหัวข้อนี้เราจะอธิบายให้เข้าใจถึงตัวเลือกนั้นๆ

ครั้งแรกที่เราเรียก VI ใหม่ ขึ้นมาสร้าง LabVIEW จะสร้างชื่อให้เป็น Untitled 1 ซึ่งชื่อนี้ก็จะปรากฏใน Title bar เราจะสังเกตว่ามีเครื่องหมายดอกจันทร์ (*) ขึ้นต่อท้ายชื่อในกรณีนี้หมายความว่า VI นี้ยังไม่ Save หลังจากเกิดการแก้ไขใน VI ครั้งล่าสุด เพื่อเตือนให้เรา Save VI ของเรา อย่าลืมว่าในการทำงานกับคอมพิวเตอร์นั้นควรจะ Save งานคุณอยู่บ่อยๆ เพราะเราไม่ทราบว่าจะเกิดเหตุขัดข้องของโปรแกรมหรือกระแสไฟฟ้าขึ้นมาเมื่อใด และขอเตือนว่า LabVIEW ไม่มีฟังก์ชันที่ Save ข้อมูลให้โดยอัตโนมัติ

Save Optionsในการบันทึกข้อมูลลงสู่แผ่น disk เรามีตัวเลือกในการเก็บข้อมูลหลายๆแบบดังนี้

1. เลือก Save เพื่อจะบันทึก VI ใหม่ เพื่อจะกำหนดตำแหน่งบน disk และตั้งชื่อ VI หรือต้องการบันทึก VI เก่าซ้ำในชื่อเดิมและที่ตำแหน่งเดิม

2. เลือก Save As... เพื่อใช้บันทึก VI เดิมด้วยชื่อใหม่หรือในตำแหน่งใหม่ โดยไม่เปลี่ยนแปลง VI เก่า ในขณะเดียวกัน ทุก VI ที่อยู่ในหน่วยความจำของเครื่องในขณะนั้นที่เรียก VI ตัวเก่าจะเปลี่ยนมาเรียก VI ตัวใหม่นี้

3. เลือก Save A Copy As... ใช้ในการบันทึก VI เดิมด้วยชื่อใหม่หรือตำแหน่งใหม่ โดยไม่เปลี่ยนแปลง VI เก่า แต่ที่แตกต่างจาก Save As คือ VI ตัวอื่นที่อยู่ในหน่วยความจำจะชี้ไปที่ VI ตัวเดิม

4. เลือก Save With Option... เมื่อเราต้องการจะบันทึก VI ในกรณีพิเศษ เช่น บันทึก VI โดยไม่บันทึก block diagram เพื่อไม่ให้ผู้อื่นแก้ไขได้ ดูรายละเอียดของการบันทึกในลักษณะนี้ใน Online Help เพื่อเข้าใจตัวเลือกต่างๆ

5. ในกลุ่มนี้จะมีคำสั่งอีกคำสั่งหนึ่งคือ Revert... โดยเราจะเลือกใช้ตัวเลือกนี้ เมื่อต้องการนำ VI ที่เป็น Version ที่ได้รับการ Save ครั้งสุดท้ายกลับมาใช้

VI Libraries ก็คือการบริหารการจัดเก็บ VI ในกรณีที่เราสร้าง VI หลายๆ VI แล้วจัดเก็บรวมไว้ใน Library เดียวกันเพื่อความสะดวกในการเลือกใช้ นอกเหนือจากนั้นระบบปฏิบัติการของ Windows จะมองเห็น VI ทั้งหมดภายใน VI Libraries เป็น file เพียง file เดียวไม่ได้เป็นกลุ่มของ file ที่บรรจุ VI หลายๆ ชุด แต่เฉพาะ LabVIEW เท่านั้นจะมองออกว่าเป็นกลุ่มของไฟล์ ประโยชน์คือจะสะดวกและง่ายในการเคลื่อนย้ายทั้งหมด file เพราะ file ที่จัดเก็บอยู่ใน Library จะได้รับการอัดเก็บ (compress) เมื่อได้รับการบันทึก

อย่างไรก็ตาม ถ้าหากเราไม่มีความจำเป็นในกรณีดังกล่าวการเก็บ VI เป็น file ย่อยก็จะมีความสะดวกในการเคลื่อนย้าย VI ทีละส่วน และการ Load และ Save file ที่จัดเก็บแต่ละ VI จะรวดเร็วกว่า file ที่อยู่รวมกัน

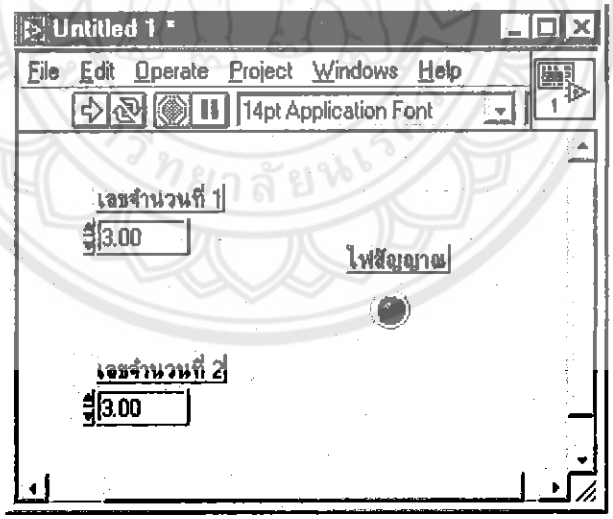
การสร้าง Library ทำได้โดยใน dialog box ของ Save..., Save As... , หรือ Save A Copy... จะมีปุ่ม New VI Library ให้เลือก เมื่อกดปุ่มนี้แล้ว เราจะได้ dialog ของ New Vi Library ขึ้นมาพิมพ์ชื่อที่ต้องการ แล้วกด VI Library สำหรับรายละเอียดในการจัดเก็บ file ในลักษณะนี้ศึกษาได้จากคู่มือของ LabVIEW

ในกิจกรรมนี้เราจะทดสอบความเข้าใจของคุณในการเขียน VI เบื้องต้น โดยขั้นตอนมีดังนี้

1. จงเขียน VI ให้มีลักษณะของ Front Panel ตามรูป โดยถ้าหากว่าเลขสองจำนวนที่กำหนดขึ้นด้วย Control ทั้งสองมีค่าเท่ากัน ไฟที่แสดงผลจะติดแสดงว่าค่าทั้งสองเท่ากัน

2. Function ที่จะใช้ใน Block Diagram คือฟังก์ชัน Equal To? ซึ่งเป็นการเปรียบเทียบเลขสองจำนวนว่าเท่ากันหรือไม่ หาได้จาก Comparison Subpalette ภายใต้ Functions Palette

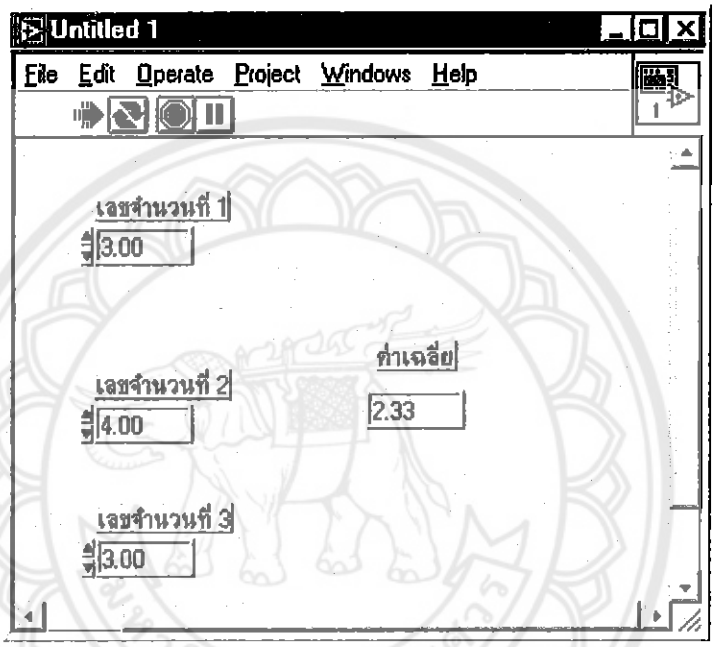
3. ลองใช้ Continuous Run เพื่อทดสอบการทำงานของVI โดยลองเปลี่ยนเลขทั้งสองจำนวนไปเรื่อยๆ



กิจกรรมนี้จะเป็นการฝึกหัดการใช้ฟังก์ชันพื้นฐานใน LabVIEW และเป็นการทดสอบความเข้าใจในการสร้าง Icon และ Connector ของคุณด้วย

ขั้นตอนของกิจกรรมนี้มีดังนี้

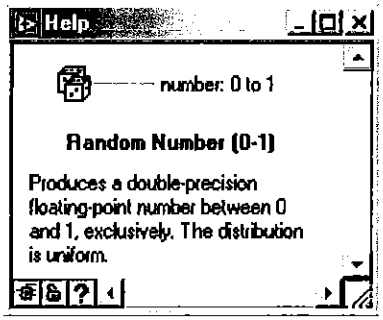
1. จงสร้าง VI ที่มีลักษณะของ Front Panel ดังรูป เพื่อใช้หาค่าเฉลี่ยของเลขสามจำนวน โดยเลขทั้งสามกำหนดด้วย Control ทั้งสามที่ปรากฏบน Front Panel



2. เขียน Block Diagram เพื่อให้ VI นี้ทำงานได้อย่างถูกต้อง
3. สร้าง VI นี้ให้เป็น subVI ด้วยการกำหนด Icon และ Connector

สำหรับ Block Diagram ของกิจกรรมนี้ไม่ควรจะมีฟังก์ชันที่เราไม่เคยพบมาก่อน และสำหรับในข้อ 3 นั้นเป็นการทดสอบความเข้าในในการเขียน subVI ของคุณ เราอยากให้คุณให้เวลากับการออกแบบ Icon ของคุณสักเล็กน้อย คุณคงจะยังไม่เห็นความสำคัญของรูป Icon ของคุณจนกระทั่งคุณต้องนำมันไปใช้เป็น subVI

จงเขียน VI เพื่อหาผลหารของเลขสองจำนวน ซึ่งทั้งสองจำนวนนี้ มีจำนวนหนึ่งได้มาจากการสุ่ม ซึ่งจะสามารถใช้ฟังก์ชัน Random ภายใต้ Numeric Subpalette ของ Functions Palette



ฟังก์ชัน Random Number (0-1) เราจะนำไปใช้ในกิจกรรมในบทต่อไป เพื่อยกตัวอย่างและแทนการวัดของเครื่องมือวัดจริง

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายอนุชิต วังใจ

ภูมิลำเนา 196 หมู่ 12 ต.รอบเวียง อ.เมือง จ.เชียงราย

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนเมืองเชียงราย
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์มหาวิทยาลัยนเรศวร

E-mail: w.anuchit@hotmail.com