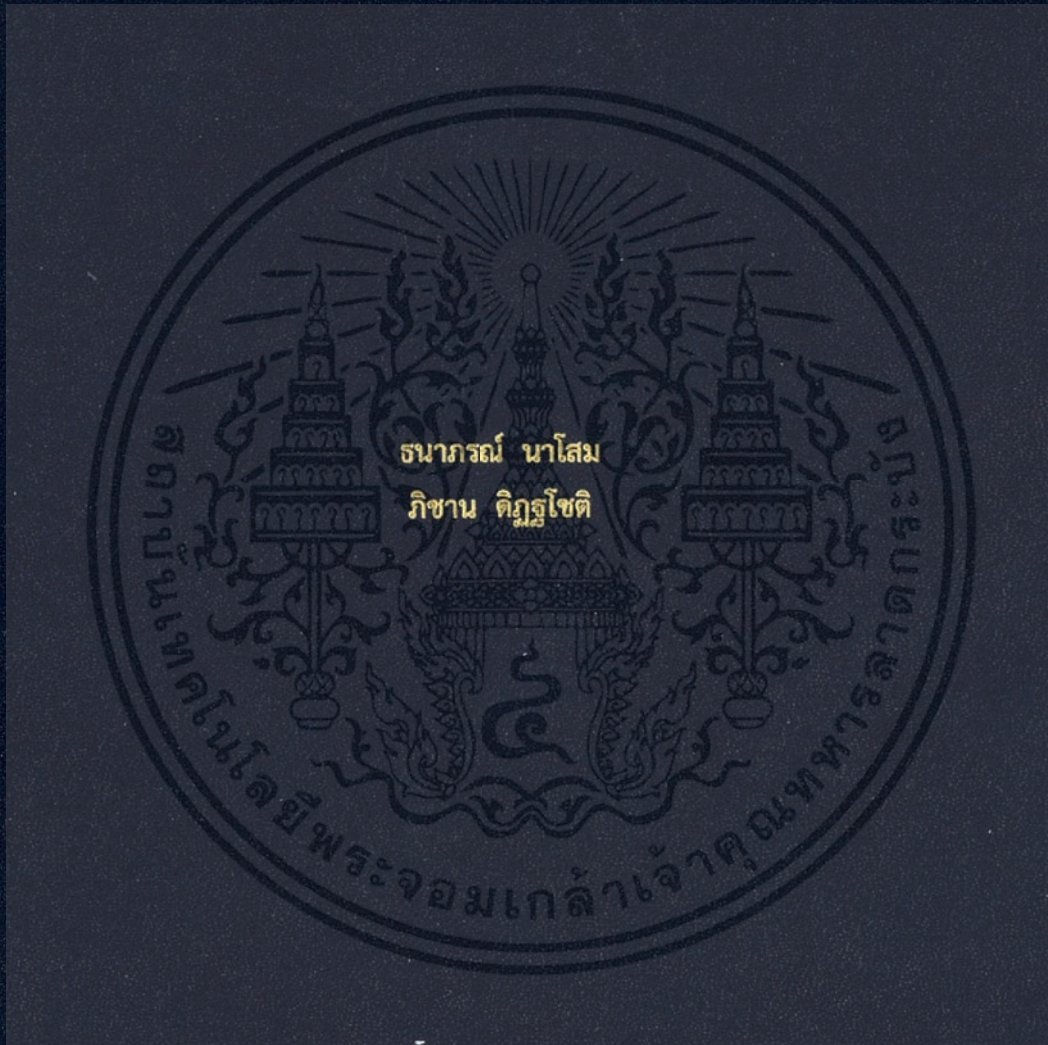


การพัฒนาโปรแกรมการวิเคราะห์การปรับสมดุลพัดลมระบายอากาศของเก้าอี้
รถยนต์

SOFTWARE DEVELOPMENT OF BALANCING ANALYSIS PROGRAM FOR CAR
SEAT VENTILATION FAN



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมขนส่งทางราง ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

การพัฒนาโปรแกรมการวิเคราะห์การปรับสมดุลพัดลมระบายอากาศของเก้าอี้
รถยนต์

SOFTWARE DEVELOPMENT OF BALANCING ANALYSIS PROGRAM FOR CAR
SEAT VENTILATION FAN



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมขนส่งทางราง ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOFTWARE DEVELOPMENT OF BALANCING ANALYSIS PROGRAM FOR CAR
SEAT VENTILATION FAN



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN RAIL TRANSPORTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2561

สาขาวิชาวิศวกรรมขนส่งทางราง ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาโปรแกรมการวิเคราะห์การปรับสมดุลพัดลมระบายอากาศของเก้าอี้รถยนต์
SOFTWARE DEVELOPMENT OF BALANCING ANALYSIS PROGRAM FOR CAR SEAT
VENTILATION FAN

ผู้จัดทำ

1. นางสาวธนาภรณ์ นาโสม รหัสประจำตัว 58010549
2. นายภิขาน ดิษฐโชติ รหัสประจำตัว 58010974



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมการวิเคราะห์การปรับสมดุลพัฒนาผลระบายอากาศของเก้าอี้รถยนต์

| | |
|-------------------------|------------------|
| ธนาภรณ์ นาโสม | 58010549 |
| ภิกษา ติฎฐโชติ | 58010974 |
| ผศ.ดร.มนต์ศักดิ์ พิมสาร | อาจารย์ที่ปรึกษา |
| ผศ.ดร.อุนนต์ พิณโสภณ | อาจารย์ที่ปรึกษา |
| ปีการศึกษา 2561 | |

บทคัดย่อ

เนื่องจากการผลิตพัฒนาผลระบายอากาศของเก้าอี้รถยนต์ของบริษัท เดลต้า อิเลคโทรนิคส์ (ประเทศไทย) จำกัด (มหาชน) ผู้ผลิตต้องทำการปรับสมดุลทางกลก่อนการจัดจำหน่าย เพื่อให้ได้พัฒนาผลตามมาตรฐาน และไม่มีปัญหาการสั่นสะเทือนและปัญหาเสียงรบกวน ดังนั้นปริญญาานิพนธ์ฉบับนี้มีจุดประสงค์เพื่อศึกษาและพัฒนาโปรแกรมคอมพิวเตอร์การวิเคราะห์การปรับสมดุลพัฒนาผลระบายอากาศของเก้าอี้รถยนต์ โปรแกรมที่พัฒนาขึ้นมาใช้พื้นฐานความรู้ของทฤษฎีการสั่นสะเทือน และทฤษฎีการปรับสมดุลวัตถุแบบสองระนาบ โดยโปรแกรมจะเก็บค่าและวิเคราะห์ข้อมูลและคำนวณความไม่สมดุลจากโปรแกรม LabVIEW จากนั้นโปรแกรมแสดงผลในรูปแบบแผนภาพต่อผู้ใช้งานเพื่อบอกมวลและตำแหน่งที่ต้องติดมวลเพิ่มเพื่อให้พัฒนาผลมีความไม่สมดุลอยู่ในเกณฑ์มาตรฐาน ISO1940 ผลลัพธ์ที่ได้จากการทดลองใช้โปรแกรมวิเคราะห์การปรับสมดุลพัฒนาผลระบายอากาศของเก้าอี้รถยนต์พบว่า โปรแกรมมีรูปแบบที่ง่ายต่อการใช้งาน ผู้ใช้งานสามารถทำการปรับค่าความไม่สมดุลของพัฒนาผลในระนาบแรกได้อย่างถูกต้อง โดยมีค่าความผิดพลาดที่ 5.97% และระนาบที่สองที่ 40.03% ซึ่งยังไม่สามารถนำไปใช้ในภาคอุตสาหกรรมได้ตามมาตรฐาน จึงจำเป็นต้องมีการพัฒนาต่อไปเพื่อที่จะใช้งานได้ในภาคอุตสาหกรรมได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOFTWARE DEVELOPMENT OF BALANCING ANALYSIS PROGRAM FOR CAR SEAT VENTILATION FAN

Thanaporn Nasom 58010549

Pichan Dittachod 58010974

Asst.Prof.Dr.Monsak Pimsarn Adviser

Asst.Prof.Dr.Unnat Pinsopon Adviser

Year 2018

ABSTRACT

For the car seat ventilation fan manufacturing of the Delta (Thailand) Public Company Limited, the manufacturer has to perform the balancing process before distribution to customers to make sure that it is complied with the standard and the vibration and noise problems can be eliminated. Therefore, this thesis aims to study and develop the balancing computer program used for a car seat ventilation fan. The developed program is based on the theories of vibration and a rigid rotor balancing, two plane balancing. This program collects and analyses the measured data via the LabVIEW program. Then, the program graphically displays the required balancing masses and phases to a user. According to the program test run, it was found that the program was easy to use and the user was able to balance in the first plane and the percent error was 5.97%. However, in the second plane, the percent error was 40.03% which is not applicable for the industry. Therefore, in order to be of practical use for industry, further development of this program is necessary.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีจากคำปรึกษา การชี้แนะ และความอนุเคราะห์ช่วยเหลือในด้านต่างๆ ตลอดระยะเวลาของการวิจัยจาก ผศ.ดร. มนต์ศักดิ์ พิมสาร และ ผศ.ดร. อุณนัตติ พิณโสภณ อาจารย์ที่ปรึกษา โดยได้ให้ความรู้เชิงวิชาการ คำปรึกษา คอยชี้จุดบกพร่องและชี้แนะแนวทางในการแก้ไขปัญหารวมทั้งการติดตามการดำเนินงานอย่างสม่ำเสมอ จึงขอขอบพระคุณอาจารย์ทั้งสองท่านเป็นอย่างสูง

ขอขอบพระคุณบริษัท เดลต้า อีเลคโทรนิคส์ (ประเทศไทย) จำกัด(มหาชน) ที่ให้ความอนุเคราะห์ในการให้คำปรึกษาแนะนำและวิธีการทำงานตลอดจนสนับสนุนอุปกรณ์และโปรแกรมขอขอบพระคุณ ดร. ณัฐวุฒิ เรืองตระกูล ผู้ให้คำแนะนำเกี่ยวกับงานด้านการวัดและคำปรึกษา ขอขอบพระคุณ ดร. จิตรกรภรณ์ วงศาางาม ผู้ให้คำแนะนำ ด้านการทดสอบ ขอขอบคุณสำนักหอสมุดกลางที่เป็นค้ำค้ำหนังสือและเอกสารอ้างอิง

ขอบคุณเพื่อนๆทุกคนที่ให้คำปรึกษา กำลังใจ และความช่วยเหลือที่ให้แก่มาโดยตลอด และขอขอบคุณบุคคลที่ไม่ได้กล่าวนาม ณ ที่นี้ได้เป็นส่วนหนึ่งสำหรับการทำงานและการใช้ชีวิต

ขอขอบคุณบิดา มารดาและครอบครัวผู้เป็นที่รักและเคารพเป็นอย่างยิ่ง ที่ได้เลี้ยงดูด้วยความรักและความเอาใจใส่ ให้โอกาสการศึกษา เป็นกำลังใจและสนับสนุนในทุกๆด้าน พวกขอระลึกในพระคุณและขอกราบขอบพระคุณ

และสุดท้ายนี้ ขอขอบคุณตัวเองที่พยายามในการทำงาน สามารถผ่านอุปสรรคหรือสิ่งบั่นทอนที่ผ่านมาได้ และสามารถนำพาปริญญาานิพนธ์ให้เสร็จสมบูรณ์ได้ด้วยดี

นางสาวธนาภรณ์ นาโสม

นายภิธาน ดิษฐโชติ

สารบัญ

| | |
|---|----|
| บทคัดย่อ..... | 5 |
| ABSTRACT..... | 6 |
| กิตติกรรมประกาศ | 7 |
| สารบัญ | 8 |
| สารบัญรูปภาพ..... | 12 |
| บทที่ 1 | 18 |
| 1.1 ความเป็นมาและความสำคัญของปัญหา..... | 18 |
| 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา..... | 18 |
| 1.3 สมมุติฐานของการศึกษา..... | 18 |
| 1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย..... | 19 |
| 1.5 ขอบเขตการวิจัย..... | 19 |
| 1.6 ขั้นตอนของการศึกษา..... | 19 |
| บทที่ 2 | 21 |
| 2.1 ทฤษฎีพื้นฐานการสันสะเทือน | 21 |
| 2.1.1 การสันสะเทือนของระบบ..... | 21 |
| 2.1.2 การสันสะเทือนทางกล..... | 24 |
| 2.1.3 การสันสะเทือนเนื่องจากการหมุนรอบแกนของมวลไม่สมดุล | 25 |
| 2.1.4 การเคลื่อนที่เป็นคาบ | 27 |
| 2.2 ทฤษฎีพื้นฐานการปรับความสมดุลวัตถุ | 30 |
| 2.2.1 สมดุลแบบสถิตยศาสตร์ (static balancing)..... | 29 |
| 2.2.2 สมดุลแบบพลศาสตร์ (dynamics balancing)..... | 31 |
| 2.3 LabVIEW..... | 35 |
| 2.3.1 หลักการและส่วนประกอบในการเขียน LabVIEW..... | 36 |
| 2.3.2 การไหลของข้อมูลและการเขียนโปรแกรม (Data flow and programing)..... | 41 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|----|
| 2.3.3 รูปแบบข้อมูล (Data type) | 43 |
| 2.3.4 ลูป (Loop) | 53 |
| 2.3.5 การหน่วงเวลา..... | 56 |
| 2.3.6 การจับเวลา..... | 58 |
| 2.3.7 ชุดตัวแปรแบบห้องแถว (Array)..... | 59 |
| 2.3.8 การสร้าง ชุดตัวแปรแบบห้องแถว (Array) ด้วยลูป | 60 |
| 2.3.9 กลุ่ม (Cluster)..... | 61 |
| 2.3.10 Error Cluster และการจัดการกับความผิดพลาด (Error)..... | 62 |
| 2.3.11 การพล็อตข้อมูล..... | 62 |
| 2.3.12 โครงสร้างกรณี (Case structure) | 63 |
| 2.3.13 โครงสร้างลำดับ (Sequence structure)..... | 64 |
| 2.3.14 โฟมูล่าโหนด (Formula node)..... | 64 |
| 2.3.15 การจัดการกับโปรเจก (Project explorer)..... | 65 |
| 2.4 ซับวีไอ (SubVI)..... | 66 |
| 2.5 เทคนิคการออกแบบ VI..... | 70 |
| 2.6 การส่งข้อมูลระหว่างลูป..... | 72 |
| 2.7 การควบคุมหน้าต่างโปรแกรม (User Interface) | 73 |
| 2.8 การบันทึกและการอ่านไฟล์..... | 74 |
| 2.9 การเขียนโปรแกรมแบบเหตุการณ์ (Event)..... | 76 |
| 2.10 การเก็บข้อมูล Data Acquisition | 78 |
| 2.10.1 องค์ประกอบระบบการเก็บข้อมูล..... | 79 |
| 2.10.2 ลักษณะสัญญาณทางไฟฟ้าที่ใช้กับอุปกรณ์ Data Acquisition..... | 80 |
| 2.10.3 ส่วนปรับสภาพสัญญาณ (Signal Conditioning) | 81 |
| 2.10.4 ขั้นตอนการแปลงสัญญาณจากอะนาล็อกเป็นดิจิตอล..... | 81 |
| 2.10.5 แซมปลิงอาร์คิเทคเจอร์ (Sampling Architecture)..... | 84 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|-----|
| 2.10.6 ประเภทของระบบการวัดสัญญาณ..... | 85 |
| 2.11 การสร้างแอปพลิเคชัน | 86 |
| 2.11.1 การเตรียมโค้ดก่อนการสร้างแอปพลิเคชัน | 86 |
| 2.11.2 การสร้าง EXE..... | 87 |
| 2.11.3 การสร้าง Installer | 91 |
| บทที่ 3 | 92 |
| 3.1 บทนำ..... | 92 |
| 3.2 ขั้นตอนการดำเนินงาน | 92 |
| 3.3 Fourier Transform..... | 93 |
| 3.4 วิธีการคำนวณค่าความไม่สมดุล | 98 |
| 3.4.1 วิธีการปรับสมดุลระนาบเดียวแบบเวกเตอร์ | 98 |
| 3.4.2 วิธีการปรับสมดุลสองระนาบแบบเวกเตอร์..... | 99 |
| 3.5 สร้างแบบจำลองทางคณิตศาสตร์และสมการการเคลื่อนที่..... | 102 |
| 3.6 การออกแบบโปรแกรม..... | 117 |
| 3.6.1 ส่วนควบคุม (Control)..... | 111 |
| 3.6.2 หน้าจอแสดงผล (Monitoring)..... | 114 |
| 3.6.3 การบันทึกข้อมูล (Data storage)..... | 116 |
| 3.6.4 การอ่านข้อมูลตั้งค่า (Read Setting Data) | 117 |
| 3.6.5 ส่วนการเตรียมตัวแปรต้นก่อนการคำนวณ (Variable Preparing)..... | 117 |
| 3.6.6 ส่วน Formular Node และการแสดงผลแบบทันที (Formular Node)..... | 118 |
| 3.6.7 ส่วนการแสดงผลแบบเฉลี่ย (Monitor Meaning Result)..... | 119 |
| บทที่ 4 | 122 |
| ผลการทดลอง | 122 |
| 4.1 ผลการทดลองเพิ่มมวล 10 mg ถึง 50 mg..... | 122 |
| 4.2 ผลการทดลองเพิ่มมวล 10 mg ในมุมที่แตกต่างกัน | 128 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|-----|
| 4.3 ผลการทดสอบวัดค่าความไม่สมดุลจากพัฒนาตัวที่ 1 กับเครื่องมือวัดเดิมของทางบริษัท | 132 |
| 4.4 ผลการทดสอบวัดค่าความไม่สมดุลจากพัฒนาตัวที่ 2 กับเครื่องมือวัดเดิมของทางบริษัท | 133 |
| 4.5 ผลการทดสอบวัดค่าความไม่สมดุลจากพัฒนาตัวที่ 3 กับเครื่องมือวัดเดิมของทางบริษัท | 134 |
| 4.6 ผลการทดสอบวัดค่าความไม่สมดุลจากพัฒนาตัวที่ 1 กับเครื่องมือวัดเดิมของทางบริษัท | 135 |
| 4.7 ผลการทดสอบวัดค่าความไม่สมดุลจากพัฒนาตัวทั้ง 4 ตัวกับเครื่องมือวัดเดิมของบริษัท | 136 |
| บทที่ 5 | 137 |
| สรุปและวิจารณ์ผลการทดลอง | 137 |
| 5.1 บทนำ | 137 |
| 5.2 สรุปผลการทดลอง | 137 |
| 5.3 ปัญหาจากการศึกษา | 137 |
| 5.4 ข้อเสนอแนะ | 138 |
| เอกสารอ้างอิง | 139 |
| ภาคผนวก | 140 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปร่างภาพ

| | |
|--|----|
| รูปที่ 2. 1 แบบจำลองการสั่นสะเทือนจากการสั่นสะเทือนของพื้น และ FBD..... | 20 |
| รูปที่ 2. 2 การเปรียบเทียบการขจัด กับความเร็ว และความเร่ง ของการสั่นสะเทือน..... | 21 |
| รูปที่ 2. 3 ลักษณะการเคลื่อนที่ของการสั่นสะเทือนที่เป็นรอบ..... | 21 |
| รูปที่ 2. 4 การสั่นสะเทือนแบบสุ่ม (Random)..... | 22 |
| รูปที่ 2.5 การสั่นสะเทือนที่เกิดจากแรงกระแทก (Impact)..... | 22 |
| รูปที่ 2. 6 การสั่นสะเทือนของระบบที่มีตัวหน่วง..... | 22 |
| รูปที่ 2.7 แบบจำลองการสั่นสะเทือนจากความไม่สมดุล และ FBD..... | 25 |
| รูปที่ 2.8 ผลตอบสนองเชิงความถี่ของปัญหาการสั่นสะเทือนจากความไม่สมดุล..... | 26 |
| รูปที่ 2.9 การเคลื่อนที่เป็นคาบด้วยคาบคงที่ T..... | 27 |
| รูปที่ 2.10 การปรับสมดุลแบบสถิตยศาสตร์..... | 29 |
| รูปที่ 2.11 แสดงผลจากการปรับสมดุลสถิตยศาสตร์..... | 31 |
| รูปที่ 2.12 การปรับสมดุลพลศาสตร์..... | 32 |
| รูปที่ 2.13 การสมดุลพลศาสตร์..... | 32 |
| รูปที่ 2.14 โปรแกรม LabVIEW..... | 35 |
| รูปที่ 2.15 Front Panel ของ LabVIEW..... | 36 |
| รูปที่ 2.16 Object ที่อยู่บน Front Panel ของ LabVIEW..... | 36 |
| รูปที่ 2.17 Controls Palette ที่ใช้ในการออกแบบ Front Panel..... | 37 |
| รูปที่ 2.18 Tools Palette ที่ใช้ในการออกแบบ Front Panel..... | 37 |
| รูปที่ 2.19 ตัวอย่าง Block Diagram Node..... | 38 |
| รูปที่ 2.20 เครื่องมือสำหรับ Dam – Data Acquisition..... | 39 |
| รูปที่ 2.21 แสดงลักษณะทั่วไปของ Icon และ Connector..... | 40 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|----|
| รูปที่ 2.22 ตัวอย่าง Block diagram ในโปรแกรม LABVIEW..... | 41 |
| รูปที่ 2.23 หลักการภายในโปรแกรม LABVIEW..... | 41 |
| รูปที่ 2.24 แสดงข้อมูลประเภท Numeric..... | 43 |
| รูปที่ 2.25 แสดงข้อมูลประเภท Boolean..... | 44 |
| รูปที่ 2.26 การเปลี่ยนรูปแบบการแสดงผลของ String Control/Indicator..... | 45 |
| รูปที่ 2.27 String Constant เพื่อระบุวิธีการ แสดงผลและป้องกันการสับสนเวลาอ่านโค้ด..... | 46 |
| รูปที่ 2.28 หน้าต่างการตั้งค่า Enum..... | 47 |
| รูปที่ 2.29 การใช้งานของ Enum..... | 47 |
| รูปที่ 2.30 การส่งข้อมูลแบบ DDT..... | 48 |
| รูปที่ 2.31 เครื่องมือใน Signal Manipulation..... | 49 |
| รูปที่ 2.32 ตัวอย่างการใช้เครื่องมือใน Signal Manipulation..... | 50 |
| รูปที่ 2.33 ตัวอย่างการใช้เครื่องมือใน Signal Manipulation..... | 50 |
| รูปที่ 2.34 ตัวอย่างการใช้ Time Stamp..... | 51 |
| รูปที่ 2.35 ตัวอย่างการสร้าง Waveform graph..... | 51 |
| รูปที่ 2.36 ตัวอย่างการสร้าง Waveform ได้ด้วยบล็อก Build Waveform..... | 52 |
| รูปที่ 2.37 การสร้าง While loop..... | 53 |
| รูปที่ 2.38 การสร้าง For loop..... | 54 |
| รูปที่ 2.39 สรุปการทำงานของ Conditional For Loop..... | 54 |
| รูปที่ 2.40 input หรือ output ของลูป..... | 55 |
| รูปที่ 2.41 แสดงการหน่วงเวลา..... | 55 |
| รูปที่ 2.42 ฟังก์ชันหน่วงเวลา Wait Until Next ms Multiple..... | 56 |
| รูปที่ 2.43 ฟังก์ชันหน่วงเวลา Wait (ms)..... | 56 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|----|
| รูปที่ 2.44 การใช้ฟังก์ชัน การจับเวลา..... | 57 |
| รูปที่ 2.45 ผลของโค้ดจากรูปที่ 2.43..... | 57 |
| รูปที่ 2.46 แผงควบคุม Front Panel..... | 58 |
| รูปที่ 2.47 การเพิ่มมิติของอาร์เรย์ (Add Dimension)..... | 58 |
| รูปที่ 2.48 For Loop ในการสร้างอาร์เรย์..... | 59 |
| รูปที่ 2.49 การสร้างอาร์เรย์ของ Random Number | 59 |
| รูปที่ 2.50 การเลือกให้ข้อมูลออกจากกลุ่มผ่าน Tunnel..... | 60 |
| รูปที่ 2.51 การให้ลูปนำค่าของอาร์เรย์เข้าไปใน ลูปที่ละ Element..... | 60 |
| รูปที่ 2.52 กราฟแบบ Waveform Graph และ Waveform Chart..... | 62 |
| รูปที่ 2.53 Case Structure..... | 62 |
| รูปที่ 2.54 การจัดทวนกรอบ Formula Node..... | 64 |
| รูปที่ 2.55 โค้ดที่ประกาศ..... | 64 |
| รูปที่ 2.56 การสร้าง SubVI..... | 65 |
| รูปที่ 2.57 หน้าต่าง Front Panel..... | 65 |
| รูปที่ 2.58 กำหนด Control / Indicator..... | 66 |
| รูปที่ 2.59 หน้าต่าง Icon Editor..... | 67 |
| รูปที่ 2.60 ขั้นตอนการใช้งาน SubVI..... | 67 |
| รูปที่ 2.61 โครงสร้างและ SucVI (Structure and SubVI)..... | 68 |
| รูปที่ 2.62 การเขียนโค้ดแบบง่าย..... | 69 |
| รูปที่ 2.63 การเขียนโค้ดแบบทั่วไป..... | 70 |
| รูปที่ 2.64 แสดงการไหลของโปรแกรมให้ออกมาในรูปแบบ State diagram หลังจากนั้นจึงแปลงไปเป็น โค้ด State Machine VI..... | 71 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|----|
| รูปที่ 2.65 การบันทึกไฟล์แบบ Low Level..... | 73 |
| รูปที่ 2.66 การบันทึกตัวเลขแบบสุ่ม..... | 74 |
| รูปที่ 2.67 แสดงการเขียนโค้ดการใช้ฟังก์ชัน Get File Size..... | 75 |
| รูปที่ 2.68 การสร้างโปรแกรมด้วยวิธีการหยังสัญญาณ (Polling)..... | 75 |
| รูปที่ 2.69 การเขียนโปรแกรมแบบหยังสัญญาณ (Event) มาทำการลบโหนด (Node)..... | 76 |
| รูปที่ 2.70 แสดงองค์ประกอบการเก็บข้อมูล (Data Acquisition หรือ DAQ)..... | 77 |
| รูปที่ 2.71 แสดงขั้นตอนการวิธีการสุ่มสัญญาณ..... | 81 |
| รูปที่ 2.72 แสดงขั้นตอนการ Resolution..... | 82 |
| รูปที่ 2.73 แสดงขั้นตอนการกำหนดความกว้าง..... | 82 |
| รูปที่ 2.74 แสดงหลักการของแอมป์ลิงอาร์คิเทคเจอร์แบบหลายทาง..... | 83 |
| รูปที่ 2.75 แสดงหลักการของแอมป์ลิงอาร์คิเทคเจอร์แบบพร้อมกัน..... | 84 |
| รูปที่ 2.76 แสดงประเภทของระบบการวัดสัญญาณ..... | 85 |
| รูปที่ 2.77 การสร้าง-Build Specification..... | 87 |
| รูปที่ 2.78 หน้าต่างสำหรับการตั้งค่าคุณสมบัติต่าง ๆ..... | 87 |
| รูปที่ 2.79 Source Files เป็นส่วนไว้สำหรับบอกว่า VI..... | 88 |
| รูปที่ 2.80 Destinations ซึ่งไว้สำหรับระบุเส้นทาง (path) ของไฟล์ต่างๆ..... | 88 |
| รูปที่ 2.81 Source File Settings..... | 89 |
| รูปที่ 2.82 Shared Variable Deployment..... | 89 |
| รูปที่ 3.1 ไดอะแกรมการไหล (dataflow diagram) ของการคำนวณ DFT..... | 95 |
| รูปที่ 3.2 Butterfly operations แบบtwiddle factorด้านเดียว และ สองด้าน..... | 96 |
| รูปที่ 3.3 ไดอะแกรมการไหล (dataflow diagram) ของการคำนวณ 8-point DFT..... | 97 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|-------------|--|-----|
| รูปที่ 3.4 | วิธีการปรับสมดุลระนาบเดี่ยวแบบเวกเตอร์..... | 98 |
| รูปที่ 3.5 | แสดงการติดตามลุ่มที่ระนาบที่ 1 และระนาบที่ 2..... | 99 |
| รูปที่ 3.6 | แผนภาพวัตถุอิสระของเครื่องมือวัดความไม่สมดุลของใบพัดลมแบบสองระนาบ ^[5] | 103 |
| รูปที่ 3.7 | แผนภาพวัตถุอิสระแสดงตัวอย่างแรงภายในจากการสั่นทั้งสองระนาบ..... | 103 |
| รูปที่ 3.8 | แผนผังขั้นตอนการทำงานของโปรแกรมที่พัฒนาขึ้นด้วยโปรแกรม LabVIEW..... | 108 |
| รูปที่ 3.9 | แผนผังขั้นตอนการคำนวณความไม่สมดุลของโปรแกรมที่พัฒนาขึ้นด้วยโปรแกรม LabVIEW..... | 109 |
| รูปที่ 3.10 | ตัวอย่าง Front Panel ของโปรแกรมที่ออกแบบ..... | 110 |
| รูปที่ 3.11 | ระนาบ 1 และระนาบ 2 บนพัดลม..... | 110 |
| รูปที่ 3.12 | การตั้งค่า (Set up) บน Front Panel..... | 111 |
| รูปที่ 3.13 | การเลือกระดับของมาตรฐาน ISO-1940..... | 112 |
| รูปที่ 3.14 | ของมาตรฐาน ISO-1940 ^[12] | 112 |
| รูปที่ 3.15 | หน้าต่าง System บน Front Panel..... | 113 |
| รูปที่ 3.16 | แสดงค่าที่เลือกใช้ในโปรแกรม..... | 113 |
| รูปที่ 3.17 | หน้าจอ Main Monitor ขณะกำลังประมวลผล..... | 114 |
| รูปที่ 3.18 | หน้าจอ Main Monitor แสดงในกรณีที่ไม่ผ่าน..... | 115 |
| รูปที่ 3.19 | หน้าจอ Main Monitor ขณะกำลังประมวลผล..... | 115 |
| รูปที่ 3.20 | หน้าจอ Main Monitor แสดงในกรณีที่ผ่าน..... | 116 |
| รูปที่ 3.21 | Block Diagram ของการเขียนไฟล์..... | 116 |
| รูปที่ 3.22 | Block Diagram โปรแกรมอ่านบันทึกการตั้งค่าจากไฟล์ binary..... | 117 |
| รูปที่ 3.23 | Block Diagram โปรแกรมส่วนการเตรียมตัวแปรต้น..... | 118 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|-----|
| รูปที่ 3.24 Block Diagram Formula Node และ Polar Plot | 119 |
| รูปที่ 3.25 Block Diagram Monitor Meaning Result..... | 120 |
| รูปที่ 3.26 Block Diagram โปรแกรมคิดคำนวณความไม่สมดุล..... | 121 |
| รูปที่ 4.1 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 0..... | 123 |
| รูปที่ 4.2 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 1..... | 124 |
| รูปที่ 4.3 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 2..... | 125 |
| รูปที่ 4.4 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 3..... | 126 |
| รูปที่ 4.5 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 4..... | 127 |
| รูปที่ 4.6 ขนาดของความเร่งของพัดลมหมายเลข 3 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ..... | 128 |
| รูปที่ 4.7 ขนาดของความเร่งของพัดลมหมายเลข 4 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ..... | 129 |
| รูปที่ 4.8 มุมเฟสของความเร่งของพัดลมหมายเลข 3 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ..... | 130 |
| รูปที่ 4.9 มุมเฟสของความเร่งของพัดลมหมายเลข 4 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ..... | 131 |
| รูปที่ 4.10 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 1 กับเครื่องมือวัดเดิมของทางบริษัท ระนาบที่ 2..... | 132 |
| รูปที่ 4.11 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 2 กับเครื่องมือวัดเดิมของทางบริษัท ระนาบที่ 2 | 133 |
| รูปที่ 4.12 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 3 กับเครื่องมือวัดเดิมของทางบริษัท ระนาบที่ 2..... | 134 |
| รูปที่ 4.13 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 4 กับเครื่องมือวัดเดิมของทางบริษัท ระนาบที่ 2..... | 135 |
| รูปที่ 4.14 แสดงค่าความไม่สมดุลของพัดลมทั้ง 4 ตัว ในระนาบที่ 1..... | 136 |
| รูปที่ 4.15 แสดงค่าความไม่สมดุลของพัดลมทั้ง 4 ตัว ในระนาบที่ 2 | 136 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการผลิตชิ้นส่วนอุตสาหกรรมผู้ผลิตจะต้องตรวจสอบชิ้นส่วนให้ได้มาตรฐานตามที่ได้กำหนดไว้ โดยทางบริษัท เอลต้า อิเล็กทรอนิกส์ (ประเทศไทย) จำกัด (มหาชน) ได้เห็นถึงปัญหาดังกล่าว โดยในการผลิตพัดลมแก้อั้วรถยนต์ผู้ผลิตต้องทำการปรับสมดุลทางกลก่อนการจัดจำหน่าย เนื่องจากความเร่งของวัตถุจะแปรผันตรงและมีทิศเดียวกับแรงลัพธ์ที่กระทำต่อวัตถุ แต่แปรผกผันกับมวลของวัตถุ เมื่อพัดลมหมุนจะมีความเร่งหนีศูนย์กลาง ผลจากวัตถุซึ่งมีการแผ่กระจายของมวลที่ไม่สม่ำเสมอจะเกิดแรงหนีศูนย์กลางเนื่องจากความไม่สมดุล ซึ่งทำให้เกิดการสั่นสะเทือนและเสียงให้สินค้าไม่เป็นไปตามมาตรฐาน แต่ในการสอบเทียบ การทดสอบเพื่อหามวลและตำแหน่งที่ไม่สมดุล ด้วยเครื่องและโปรแกรมที่ไม่ได้มาตรฐานเพียงพอและยากต่อการติดตั้ง ส่งผลให้เกิดความล่าช้าทางการผลิตและพัดลมที่ไม่ได้มาตรฐาน ISO1940 ตาม G6.3^[1] สำหรับโครงการฉบับนี้ได้ทำการออกแบบโปรแกรมเพื่อติดตั้งเข้ากับเครื่องปรับสมดุลพัดลมแก้อั้วรถยนต์ให้ง่ายต่อการสอบเทียบ เพิ่มความแม่นยำและเที่ยงตรง เหมาะกับการใช้งานในภาคอุตสาหกรรม

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- 1.2.1 เพื่อศึกษาการสั่นสะเทือนของระบบสมดุลลิจิดโรเตอร์ (rigid rotor)
- 1.2.2 เพื่อศึกษาและพัฒนาโปรแกรมคอมพิวเตอร์การวิเคราะห์การปรับสมดุลสองระนาบของพัดลมระบายอากาศของแก้อั้วรถยนต์
- 1.2.3 เพื่อออกแบบให้สามารถนำมาประยุกต์ใช้กับพัดลม สามารถทดสอบและพัฒนาโปรแกรมได้ในอนาคต

1.3 สมมุติฐานของการศึกษา

- 1.3.1 ทราบถึงสมการคำนวณความไม่สมดุลและการสั่นสะเทือน

1.3.2 สามารถออกแบบ และพัฒนาโปรแกรมในการเก็บข้อมูล ความหนาชิ้นงาน ไม่ยาวมาก สามารถปรับสมดุลด้วยวิธีสองระนาบได้ เพื่อใช้ในการหาค่ามวลที่ไม่สมดุลและตำแหน่งของความไม่สมดุลที่แม่นยำและเที่ยงตรง

1.3.3 สามารถออกแบบ และพัฒนาโปรแกรมในการวิเคราะห์ข้อมูล ให้ง่ายต่อการหาค่ามวลและตำแหน่งที่ไม่สมดุล

1.3.4 สามารถออกแบบโปรแกรมการสอบเทียบที่รวดเร็วและถูกต้อง

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

1.4.1 ทฤษฎีพื้นฐานการสั่นสะเทือนของเครื่องจักร

1.4.2 ทฤษฎีความไม่สมดุลแบบพลศาสตร์และวิธีการปรับความไม่สมดุล

1.5 ขอบเขตการวิจัย

1.5.1 เขียนสมการแสดงแบบจำลอง

1.5.2 เขียนโปรแกรมเพื่อใช้ในการเก็บค่าและคำนวณหาค่ามวลและตำแหน่งที่จะทำให้วัตถุสมดุลซึ่งอ่านจากเครื่องมือวัดสัญญาณ

1.5.3 เขียนโปรแกรมเพื่อประมวลผลและแสดงค่าการสั่นสะเทือนและมุมเฟสการสั่นสะเทือน

1.5.4 สามารถแก้ไขโปรแกรมให้สามารถใช้งานได้อย่างวนซ้ำ

1.5.5 สร้างโปรแกรม โดยสามารถแสดงผลผ่าน การใช้ภาพเป็นตัวประสานกับผู้ใช้ (Graphic user interface) เพื่อการใช้งานที่สะดวก

1.6 ขั้นตอนของการศึกษา

1.6.1 ศึกษาทฤษฎีพื้นฐานการสั่นสะเทือนทางกล ทฤษฎีความไม่สมดุล และวิธีการปรับความไม่สมดุล

1.6.2 ศึกษามาตรฐานระดับความสมดุลพัฒนา

1.6.3 ศึกษาโปรแกรม ในเก็บค่าข้อมูลของชุดอุปกรณ์

1.6.4 ออกแบบโปรแกรมในการสอบเทียบ (calibration) และการเก็บค่าข้อมูล

1.6.5 ติดตั้งโปรแกรมเข้ากับเครื่องปรับสมดุลพัฒนาความร้อนแก้อัตโนมัติ

1.6.6 ทดสอบการทำงานของโปรแกรม

1.6.7 วิเคราะห์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6.8 สรุปผลที่ได้รับ และจัดทำรายงานวิจัย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

งานวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีพื้นฐานการสั่นสะเทือน

การสั่นสะเทือนทั่วไป จะมีแรงภายนอกกระทำอยู่ด้วย เป็นการสั่นสะเทือนซึ่งเกิดจากพลังงานภายนอกกระทำกับระบบระหว่างการสั่นสะเทือน โดยพลังงานภายนอกกระทำกับระบบระหว่างการสั่นสะเทือน โดยพลังงานภายนอกที่กระทำอาจอยู่ในรูปของแรง แรงบิด หรืออาจเป็นการขจัดเพื่อบังคับให้เกิดการสั่น โดยการกระทำของแรงชนิดต่าง ๆ ทำให้เกิดการตอบสนอง (response) ของระบบขึ้น

2.1.1 การสั่นสะเทือนของระบบ

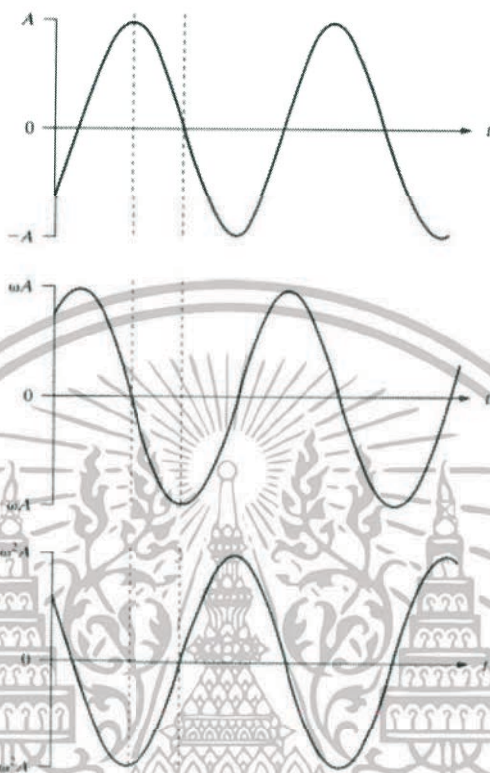
สมมติให้ระบบเครื่องจักรกลประกอบด้วยส่วนที่เป็นมวล(m) และส่วนที่เป็นอิลาสติกหรือสปริง(k) เช่นในรูปที่ 2.1



รูปที่ 2.1 แบบจำลองการสั่นสะเทือนจากกัวการสั่นสะเทือนของพื้น และ FBD^[2]

การสั่นสะเทือนจะเกิดขึ้นถ้ามีการสะสมพลังงานในระบบ โดยปกติมวล m จะสะสมพลังงานจลน์ สปริงหรือส่วนที่เป็นอิลาสติกจะสะสมพลังงานศักย์สมมติให้มวล m เคลื่อนที่ขึ้น-ลงได้เฉพาะในแนวตั้ง จากตำแหน่งสมดุลถ้าดึงมวล m ลงในแนวตั้งแล้วปล่อยมวล m จะเคลื่อนที่ขึ้น-ลงถ้าการเคลื่อนที่ดังกล่าวเคลื่อนที่ขึ้น-ลงซ้ำ ๆ กันในเวลาที่เหมาะสมจะเรียกการเคลื่อนที่ในลักษณะนี้ว่า “การเคลื่อนที่เป็นรอบ (Cycle หรือ Periodic)” ถ้าสปริงมีค่าคงที่ (แรง/ หน่วยยืดตัว) ความสัมพันธ์ระหว่างระยะเคลื่อนที่ (Displacement) และเวลาจะเป็นลักษณะ sine หรือ cosine และเรียกว่าเป็นการเคลื่อนแบบ “ฮาร์โมนิก” ดังแสดงในรูปที่ 2.2 ซึ่ง x คือระยะเคลื่อนที่ และเวลาความแตกต่างระหว่างภาพ, x และ c จะ

ขึ้นอยู่กับเงื่อนไขเริ่มต้นการเคลื่อนที่ (initial conditions of displacement and velocity) ขนาดสูงสุดของระยะเคลื่อนที่ (x) เรียกว่าขนาด (Amplitude) ส่วนเฟสหรือหรือมุมเฟสจะแทนด้วย ϕ และเรียก ω ว่า “ ความถี่รอบ ” (circular frequency) หรือ “ ความถี่ ”



รูปที่ 2.2 การเปรียบเทียบการจัด กับความเร็ว และความเร่ง ของการสั่นสะเทือน^[2]

ในบางครั้งเงื่อนไขการเคลื่อนที่อาจทำให้ระบบเคลื่อนที่ต่างจากแบบฮาร์มอนิกแต่ยังมีลักษณะเป็นรอบตัวอย่างเช่นการเคลื่อนที่ในรูปที่ 2.3 การเคลื่อนที่จากจุดหนึ่งถึงอีกจุดหนึ่งที่มีระยะการเคลื่อนที่ (s) เท่ากันในหนึ่งรอบเรียกว่า “รอบการเคลื่อนที่” หรือ “รอบ” (cycle) และเวลาที่ใช้ในการเคลื่อนที่หนึ่งรอบเรียกว่า “คาบ” ซึ่งแทนด้วย “มีหน่วยเป็นวินาทีจำนวนรอบการเคลื่อนที่ต่อหนึ่งหน่วยเวลาเรียกว่า “ความถี่” ซึ่งแทนด้วยโดยทั่วไปจะมีหน่วยเป็นรอบต่อวินาทีหรือใช้ตัวย่อ cps. ดังนั้น

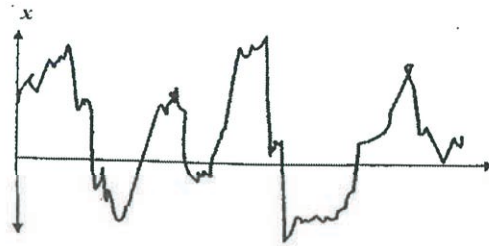
$$f = \frac{1}{T} \text{ หรือ } T = \frac{1}{f}. \quad (2.1)$$



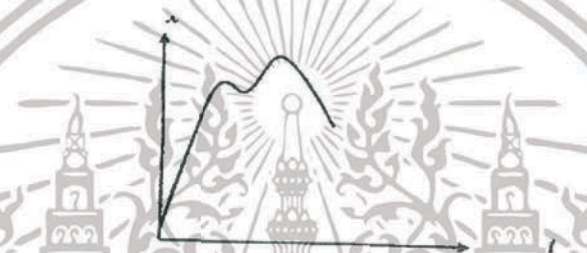
รูปที่ 2.3 ลักษณะการเคลื่อนที่ของการสั่นสะเทือนที่เป็นรอบ^[3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสั่นสะเทือนอาจมีรูปร่างลักษณะไม่แน่นอนเช่นในรูปที่ 2.4 การเคลื่อนที่จะไม่มีส่วนที่ซ้ำกัน จะเรียกว่าการสั่นสะเทือนแบบสุ่ม ซึ่งเกิดจากแรงที่มีขนาดไม่แน่นอนกระทำต่อระบบส่วนการสั่นสะเทือนที่เกิดจากแรงกระแทก (impact) การตอบสนองของระบบจะใช้เวลาสั้นเช่นในรูป 2.5

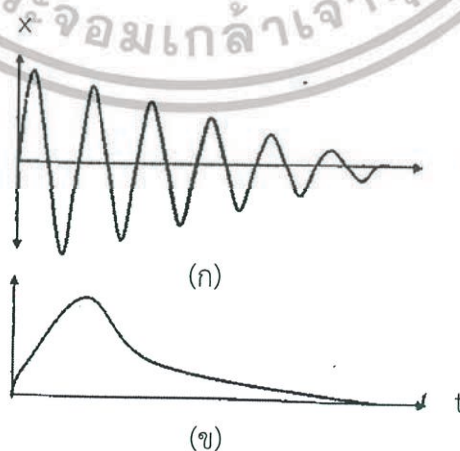


รูปที่ 2.4 การสั่นสะเทือนแบบสุ่ม (Random) ^[3]



รูปที่ 2.5 การสั่นสะเทือนที่เกิดจากแรงกระแทก (Impact) ^[3]

ระบบการสั่นสะเทือนที่มีตัวหน่วงการเคลื่อนที่เช่นการด้านการเคลื่อนที่ของอากาศ, ซอคแอบซอร์เบอร์ เป็นต้น ลักษณะการเคลื่อนที่จะคล้ายรูปที่ 2.6 ในรูปที่ 2.6 ก. สามารถหาค่าการเคลื่อนที่ได้แต่รูปที่ 2.6 ข จะไม่มีคาบการเคลื่อนที่และเรียกเคลื่อนที่ในรูปที่ 2.6 ข ว่า “การเคลื่อนที่ไม่เป็นรอบ” (non periodic หรือaperiodic)



รูปที่ 2.6 การสั่นสะเทือนของระบบที่มีตัวหน่วง^[3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบอกตำแหน่งหรือรูปร่างการเคลื่อนที่ของระบบอาจจำเป็นต้องใช้แกนอิสระหลายแกนจำนวน แกนอิสระเหล่านั้นจะเป็นตัวกำหนดจำนวนดีกรีอิสระของระบบเช่นในรูปที่ 2.1 จำนวนแกนอิสระเพียง แกนเดียวสามารถบอกตำแหน่งของการเคลื่อนที่ของระบบได้ดังนั้นจึงเป็นระบบที่มีดีกรีอิสระเดียวระบบที่มีดีกรีอิสระมากกว่า 2 ดีกรีขึ้นไปจะเรียกว่าระบบหลายดีกรีอิสระสำหรับระบบที่มีมวลและส่วนอิลาสติกกระจายสม่ำเสมอและต่อเนื่องจะมีจำนวนดีกรีอิสระจำนวนมากทั้งนี้เพราะระบบคานาระบบเพลลา ประกอบด้วยอิเลิเมนต์ของมวลและส่วนอิลาสติกจำนวนมากสำหรับระบบหลายดีกรีอิสระเงื่อนไขเริ่มต้น การเคลื่อนที่อาจทำให้การเคลื่อนที่ของระบบของระบบในแต่ละแกนอิสระเคลื่อนที่แบบฮาร์โมนิคและ เรียกลักษณะรูปร่างของการเคลื่อนที่ดังกล่าวว่า "โหมดหรือโหมดหลัก" (normal mode หรือ principle mode) ของการสั่นสะเทือนและจำนวนโหมดของการเคลื่อนที่ที่เท่ากับดีกรีอิสระของระบบ

โดยทั่วไปไม่ประสงค์จะให้เกิดการสั่นสะเทือนขึ้นเพราะการสั่นสะเทือนอาจจะทำให้เครื่องจักรกล ขำรดเสียหายแรงจากการสั่นสะเทือนอาจส่งผ่านฐานเครื่องหรือชิ้นส่วนที่อยู่ใกล้เคียงและถ้าขนาดของการ สั่นสะเทือนมีค่ามากก็อาจจะมีผลต่อการทำงานของชิ้นส่วนเครื่องจักรกลนอกจากนี้การสั่นสะเทือนยังทำให้ เกิดเสียงดังและทำให้ชิ้นส่วนสั่นคลอนอีกด้วยอย่างไรก็ตามในอุตสาหกรรมบางประเภทจำเป็นต้อง อาศัยการสั่นของเครื่องจักรกลเช่นเครื่องเขย่า (shaking) เป็นต้นเครื่องจักรกลหรือโครงสร้างอาจจะ ประกอบด้วยชิ้นส่วนที่ซับซ้อนแต่สามารถจำลองหรือแทนเครื่องจักรกลดังกล่าวด้วยระบบง่ายๆที่ ประกอบด้วยมวล, สปริงและตัวหน่วงและเรียกระบบที่จำลองขึ้นใหม่นี้ว่า "ระบบเสมือน" (Equivalent system) หรือ "ระบบสั่นสะเทือนจำลอง" (Vibration model) ผลของการวิเคราะห์จึงเป็นผลของระบบ จำลอง

2.1.2 การสั่นสะเทือนทางกล

การสั่นสะเทือนทางกลแต่ละชนิดจะมีรูปแบบที่ไม่เหมือนกันโดยจะมีการกระจัดความเร็วและความเร่งเป็นฟังก์ชันของเวลา ลักษณะโดยทั่วไปจะมีลักษณะเป็นคาบ มีการลดลงเนื่องจากตัวหน่วงการ เคลื่อนที่ และผลของความเสียดทาน สัญลักษณ์ตัวแปร คุณลักษณะที่สำคัญ ได้แก่

$$\text{แอมพลิจูดที่เวลา } t : \quad x = f(t) \quad (2.2)$$

$$\text{คาบของการสั่น :} \quad T = \frac{1}{f} \quad (2.3)$$

$$\text{ความเร็วเชิงมุม :} \quad \omega = 2\pi f = \frac{2\pi}{T} \quad (2.4)$$

$$\text{มุมเฟส :} \quad \gamma = \omega t + \gamma_0 \quad (2.5)$$

โดย x คือแอมพลิจูดการสั่นสะท้อนในรูปแบบความเร็ว(เมตรต่อวินาที) ความเร่ง(เมตรต่อวินาที²) และการกระจัด(เมตร)

t คือเวลา(วินาที)

f คือความถี่(รอบต่อวินาที)

T คือคาบเวลา(วินาที)

y_0 คือมุมเฟสที่ $t = 0$

y คือมุมเฟสที่ t ใดๆ

2.1.3 การสั่นสะท้อนเนื่องจากการหมุนรอบแกนของมวลไม่สมดุ

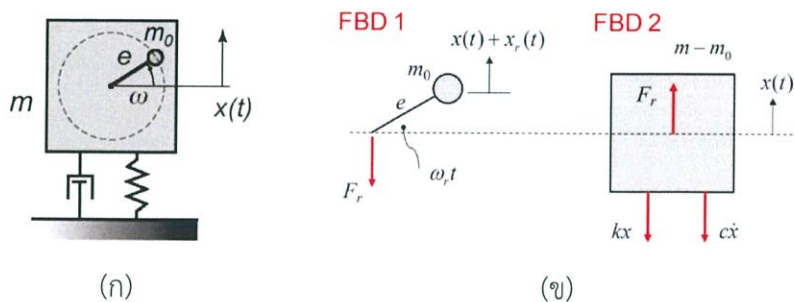
สำหรับชิ้นส่วนบางชิ้นในเครื่องจักรกลขณะทำงานที่มีการหมุน ซึ่งส่งผลให้เกิดการกระตุ้นอย่างต่อเนื่องต่อระบบการสั่นสะท้อนที่เกิดขึ้นจึงเป็นผลมาจากแรงหนีศูนย์กลางของมวลที่กำลังหมุนองค์ประกอบของแรงหนีศูนย์กลางในแนวการสั่นของระบบจะเปลี่ยนแปลงขนาดและทิศทางตามเวลาการสั่นสะท้อนนี้จึงจัดอยู่ในการสั่นสะท้อนแบบบังคับในรูปที่ 2.7 แสดงมวล m , ที่หมุนรอบจุดใดจุดหนึ่งบนเครื่องจักรกลซึ่งมีมวล M (ไม่รวมมวลที่กำลังหมุน) มวลทั้งหมดของระบบ $m = M + m_0$ ระบบติดตั้งไว้กับสปริงที่มีค่าคงตัว k และตัวหน่วงที่มีค่าคงตัว c ในภาพ (a) มวล m_0 หมุนรอบจุด O บนเครื่องจักรกลด้วยรัศมี (ระยะเบี่ยงศูนย์กลาง) เท่ากับ e หมุนด้วยอัตราเร็วเชิงมุมคงตัว ω ในทิศทวนเข็มนาฬิกา

เลือกระดับอ้างอิงเป็นตำแหน่งสมดุลสถิตของระบบ กล่าวคือ $kx_0 = mg$ และกำหนดระยะกระจัด (x) ความเร็ว (\dot{x}) และความเร่ง (\ddot{x}) มีทิศทางเป็นบวกดังแสดงใน FBD ของเครื่องจักรกลภาพ (ค) บน FBD จะมีแรงหนีศูนย์กลางกระทำต่อมวลหมุน $m_0e\omega^2$ ทำมุม ωt กับแนวระดับน้ำหนักของมวลหมุน m_0g น้ำหนักของเครื่องจักรกลไม่รวมมวลหมุน Mg แรงด้านสปริง $k(x - x_0)$ และแรงหน่วง $c\dot{x}$ พิจารณา FBD ด้วยกฎการเคลื่อนที่ข้อสองของนิวตัน

$$[\uparrow \sum F_x = m\ddot{x}] \text{ โดยที่ระบบมีมวลทั้งหมด } m = M + m_0$$

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \left(\frac{m_0e\omega^2}{m}\right)\sin\omega t \quad (2.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แบบจำลองการสั่นสะเทือนจากความไม่สมดุล และ FBD^[2]

โดยกำหนดให้ $c / m = 2\zeta\omega_n$ และ $\omega_n^2 = k / m$ นั่นคือ

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = \left(\frac{m_0e\omega^2}{m}\right)\sin\omega \tag{2.7}$$

โดยแทนค่า F_0 ด้วย $m_0e\omega^2$

$$\frac{X}{m_0e\omega^2/k} = \frac{1}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\zeta\frac{\omega}{\omega_n}\right)^2}} \tag{2.8}$$

มุมเฟสของผลตอบสนอง

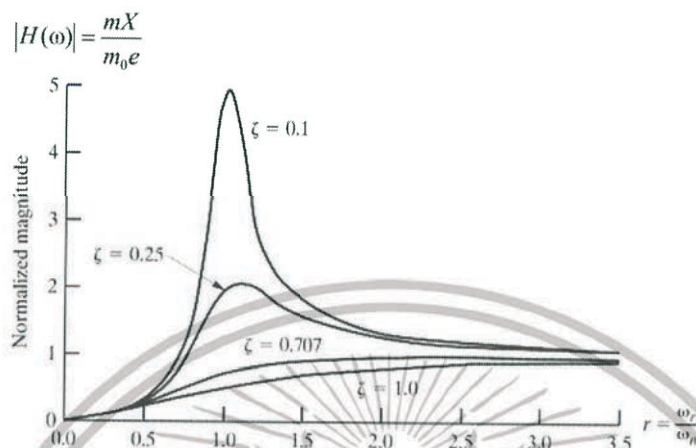
$$\phi = \tan^{-1}\left[\frac{2\zeta\frac{\omega}{\omega_n}}{1 - \left(\frac{\omega}{\omega_n}\right)^2}\right] \tag{2.9}$$

จัดสมการ (2.9) ใหม่โดยกำหนดให้ $= k / m$ จะได้

$$\frac{mX}{m_0e} = \frac{\left(\frac{\omega}{\omega_n}\right)^2}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\zeta\frac{\omega}{\omega_n}\right)^2}} \tag{2.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mX/m_0e เรียกว่าอัตราส่วนการขยาย (magnification ratio) ซึ่งพบว่าอัตราส่วนการขยายนี้ขึ้นอยู่กับอัตราส่วนความถี่ ω/ω_n ภายใต้อัตราส่วนการหน่วง (ζ) ที่กำหนดให้ค่าหนึ่ง ๆ เมื่อนำมาพล็อตกราฟระหว่าง mX/m_0e กับ ω/ω_n ที่ค่ากำหนดของ g จะได้ดังรูปที่ 2.8



รูปที่ 2.8 ผลตอบสนองเชิงความถี่ของปัญหาการสั่นสะเทือนจากความไม่สมดุล^[2]

ในรูปที่ 2.8 แสดงความสัมพันธ์ระหว่างอัตราส่วนการขยายและอัตราส่วนความถี่สำหรับระบบที่มีการหน่วงขนาดต่าง ๆ กราฟที่ได้มีลักษณะคล้ายคลึงกับความสัมพันธ์ระหว่างอัตราส่วนแอมพลิจูดกับความถี่ของแรงภายนอก จะแตกต่างกันเพียงเล็กน้อย คือกราฟในรูปที่ 2.8 ผ่านจุดกำเนิดและที่ $\omega \gg \omega_n$ นั้นแอมพลิจูดของผลตอบสนอง X เข้าใกล้ค่าของ $\frac{m_0e}{m}$ ส่วนกราฟของมุมเฟสจะเหมือนกันทุกประการ (ในกรณีของแรงฮาร์มอนิก $F(t) = F_0 \sin \omega t$)

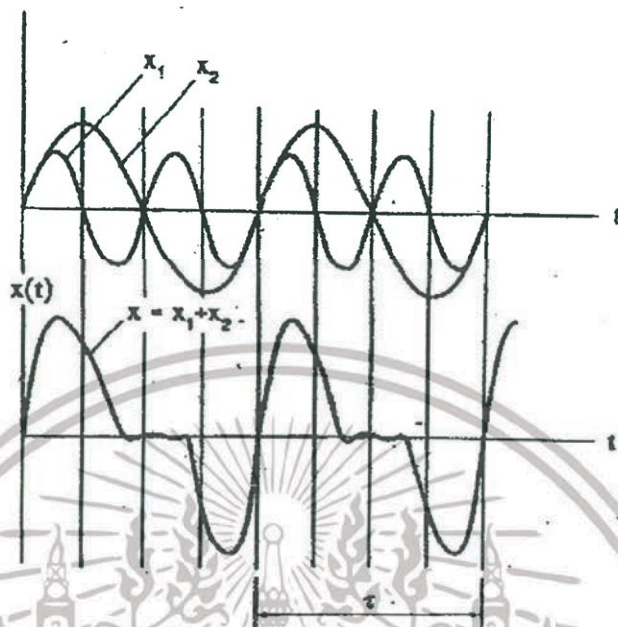
2.1.4 การเคลื่อนที่เป็นคาบ

การสั่นสะเทือนของเครื่องจักรกลหรือโครงสร้างอาจประกอบด้วยความถี่ของการสั่นที่มีหลาย ๆ ค่าจึงเกิดการเคลื่อนที่เป็นคาบหลาย ๆ ชุดฮาร์มอนิกแต่ละชุดจะมีความถี่ไม่เท่ากันอันประกอบด้วยความถี่มูลฐาน f และฮาร์มอนิกที่ความถี่เป็นจำนวนเท่าของความถี่มูลฐานซึ่งได้แก่ $2f, 3f, 4f, nf$ ตามลำดับ

ในการสั่นสะเทือนแบบอิสระที่มีหลาย ๆ ระดับชั้นความถี่จะประกอบด้วยความถี่ธรรมชาติของการสั่นหลาย ๆ ค่าพร้อมกันจึงได้รูปคลื่นเชิงซ้อน (complex waveform) โดยรูปคลื่นผลลัพธ์ที่ได้จะมีลักษณะหรือเหมือนกันในแต่ละคาบของคลื่นดังแสดงในรูปที่ 2.9 ซึ่งแสดงการเคลื่อนที่เป็นคาบทั้ง 2 ชุดที่มีความถี่ธรรมชาติเป็นจำนวนเท่า $f_2 = 2f_1$ แต่ละคนมีระยะกระจัด $x_1(t) = A_1 \sin(2\pi f_1 t)$ และ $x_2(t) = A_2 \sin(2\pi f_2 t)$ ตามลำดับ คลื่นผลลัพธ์ที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$x_1(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t) \quad (2.11)$$



รูปที่ 2.9 การเคลื่อนที่เป็นคาบด้วยคาบคงที่ T

โดยคาบมูลฐานของคลื่นผลลัพธ์ที่ได้ (T) พิจารณาที่ความถี่เดียวกันกับความถี่มูลฐานของคลื่นชุดแรก $\tau = \frac{1}{f_1}$ เจ. ฟูรีเยร์ (J. Fourier) นักคณิตศาสตร์ชาวฝรั่งเศสเสนอว่าการเคลื่อนที่เป็นคาบนั้นสามารถแทนได้ด้วยอนุกรมฟูรีเยร์ (Fourier Series) ซึ่งเป็นผลบวกของอนุกรมของไซน์และโคไซน์ที่มีความสัมพันธ์เชิงฮาร์มอนิกต่อกันถ้ากำหนดให้การเคลื่อนที่เป็นคาบที่มีระยะกระจัด $x(t)$ และคาบของการเคลื่อนที่เท่ากับ τ และสามารถเขียนในรูปสมการของอนุกรมฟูรีเยร์ที่ประกอบด้วยฟังก์ชันฮาร์มอนิก 2 ฟังก์ชันดังนี้

$$x(t) = \frac{a_0}{2} + a_1 \cos \omega t + a_2 \cos 2\omega t \dots + a_n \cos n\omega t + b_1 \cos \omega t + b_2 \cos 2\omega t \dots + b_n \cos n\omega t \quad (2.12)$$

สำหรับสัมประสิทธิ์ของอนุกรมฟูรีเยร์ $a_0, a_1, a_2, \dots, a_n$ และ $b_0, b_1, b_2, \dots, b_n$ จะอาศัยสมบัติของความสัมพันธ์เชิงตั้งฉากของฟังก์ชันในรูปของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\int_{-\tau/2}^{\tau/2} \cos \omega_n t \cos \omega_m t dt = \begin{cases} 0 & \text{สำหรับ } m \neq n \\ \frac{\tau}{2} & \text{สำหรับ } m = n \end{cases} \quad (2.13)$$

$$\int_{-\tau/2}^{\tau/2} \sin \omega_n t \sin \omega_m t dt = \begin{cases} 0 & \text{สำหรับ } m \neq n \\ \frac{\tau}{2} & \text{สำหรับ } m = n \end{cases}$$

$$\int_{-\tau/2}^{\tau/2} \sin \omega_n t \cos \omega_m t dt = \begin{cases} 0 & \text{สำหรับ } m \neq n \\ 0 & \text{สำหรับ } m = n \end{cases}$$

สัมประสิทธิ์ของฟูเรียร์ a_n พิจารณาจากสมการ (2.13) คูณด้วย $\cos \omega_m t$ แล้วหาปริพันธ์สมการในช่วงเวลา $-\tau/2$ ถึง $\tau/2$ จะได้

$$\int_{-\tau/2}^{\tau/2} x(t) \cos \omega_m t dt = \int_{-\tau/2}^{\tau/2} (\sum_{n=1}^{\infty} a_n \cos \omega_n t) \cos \omega_m t dt \quad (2.14)$$

$$= \int_{-\tau/2}^{\tau/2} (\sum_{n=1}^{\infty} b_n \sin \omega_n t) \cos \omega_m t dt$$

จากผลของสมบัติตั้งฉากกันในสมการ (2.13)

$$\int_{-\tau/2}^{\tau/2} x(t) \cos \omega_m t dt = a_m \left(\frac{\tau}{2} \right) + 0 \quad (2.15)$$

$$a_n = \frac{2}{\tau} \int_{-\tau/2}^{\tau/2} x(t) \cos \omega_n t dt \quad n=0,1,2,3,\dots \quad (2.16)$$

ในทำนองเดียวกัน จะหาค่าสัมประสิทธิ์ของฟูเรียร์ b_n ได้โดยการคูณสมการ (2.13) ด้วย $\sin \omega_n t$ แล้วหาปริพันธ์สมการ ผลที่ได้คือ

$$b_n = \frac{2}{\tau} \int_{-\tau/2}^{\tau/2} x(t) \sin \omega_n t dt \quad n=0,1,2,3,\dots \quad (2.17)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ทฤษฎีพื้นฐานการปรับความสมดุลวัตถุ

การปรับสมดุลเพื่อลดการสั่นสะเทือนในเครื่องจักร การปรับสมดุลจะแบ่งออกเป็นสองชนิด คือ

1) สมดุลแบบสถิตยศาสตร์ (static balancing)

การแก้ไขความไม่สมดุลแบบสถิตยศาสตร์จะมุ่งไปที่การแก้ไขในระนาบเดียวและเป็นการเพิ่มมวลที่เหมาะสมในระนาบเดียวกันเท่านั้น รูปแบบการหมุนที่มีศูนย์กลางของมวลอยู่ในระนาบเดียว หรือกล่าวได้ว่าถ้าวัตถุมีเส้นผ่านศูนย์กลางมากกว่าความยาว 7 ถึง 10 เท่าก็มักจะแก้ไขในระนาบเดียวหรือแบบสถิตยศาสตร์

2) สมดุลแบบพลศาสตร์ (Dynamics balancing)

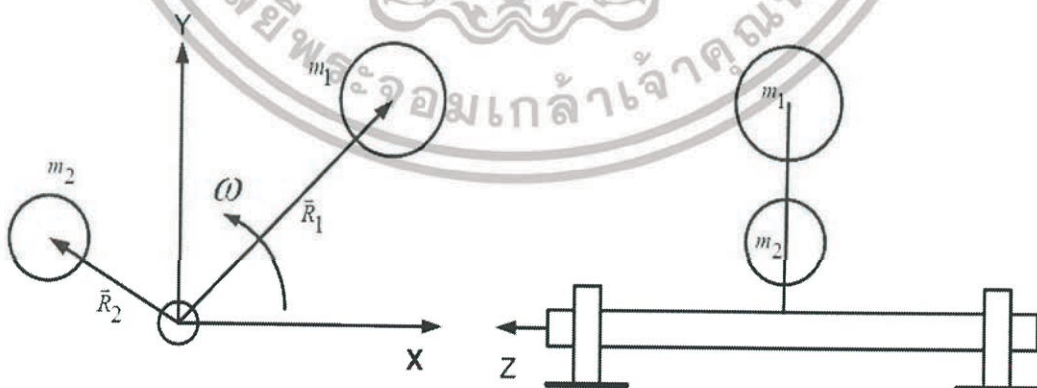
สำหรับการศึกษาพลศาสตร์สามารถแยกออกได้เป็น 2 ส่วน คือ

- คิเนเมติกส์ (Kinematics) เป็นการศึกษาถึงลักษณะการเคลื่อนที่ของวัตถุ การหาการจัดความเร็ว และความเร่ง โดยไม่มีการพิจารณาแรงกระทำบนวัตถุ
- จลศาสตร์ (Kinetics) เป็นการศึกษาและวิเคราะห์แรงกระทำที่ทำให้วัตถุเกิดการเคลื่อนที่

2.2.1 สมดุลแบบสถิตยศาสตร์ (static balancing) คือ ผลรวมแรงต้องมีค่าเท่ากับศูนย์

$$\Sigma \vec{F} = \vec{0}$$

(2.18)



รูปที่ 2.10 การปรับสมดุลแบบสถิตยศาสตร์⁽⁴⁾

จากรูปในระบบที่มีการหมุนของ m_1 และ m_2 รอบเพลา ผลจากการหมุนทำให้เกิดแรงหนีศูนย์กลาง(Centrifugal Force) ของมวลทั้งสอง เพราะที่ m_1 และ m_2 จะมีความเร่งสู่ศูนย์กลาง (Centripetal acceleration) โดยทำการเพิ่มมวล m_b เพื่อจะทำให้เกิดสมดุลแรงเหวี่ยงหนีศูนย์กลาง (Inertia forces) แทนค่าแรงทั้งสามของมวลทั้งสามลงในสมการที่ (2.18) จะได้

$$m_1 \vec{R}_1 \omega^2 + m_2 \vec{R}_2 \omega^2 + m_b \vec{R}_b \omega^2 = 0$$

หรือ

$$m_b \vec{R}_b = -m_1 \vec{R}_1 - m_2 \vec{R}_2 \quad (2.19)$$

แตกออกตามแนวแกน x และ y ได้เป็น

$$m_b R_{b_x} = -m_1 R_{1_x} - m_2 R_{2_x} \quad (2.20)$$

$$m_b R_{b_y} = -m_1 R_{1_y} - m_2 R_{2_y} \quad (2.21)$$

และสามารถหาทิศทางของ $m_b \vec{R}_b$ โดยการนำสมการ (2.20) หาด้วย (2.21) ได้ดังนี้

$$\theta_b = \tan^{-1} \left(\frac{m_b R_{b_y}}{m_b R_{b_x}} \right)$$

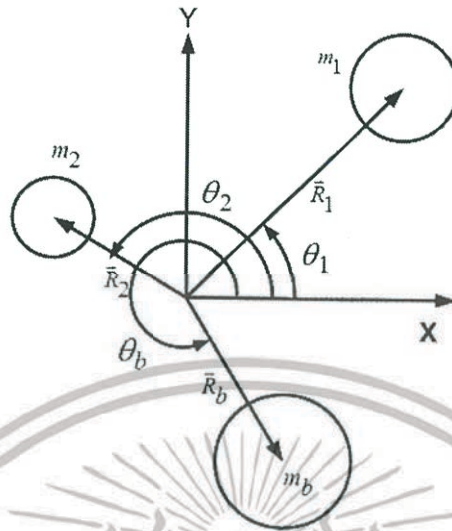
$$\theta_b = \tan^{-1} \left(\frac{-m_1 R_{1_y} - m_2 R_{2_y}}{-m_1 R_{1_x} - m_2 R_{2_x}} \right) \quad (2.22)$$

นอกจากนั้นจากรูปสามารถสรุปได้

$$R_b = \sqrt{R_{b_x}^2 + R_{b_y}^2}$$

$$m_b R_b = \sqrt{(m_b R_{b_x})^2 + (m_b R_{b_y})^2} \quad (2.23)$$

จากการปรับสมดุลตามสมการข้างต้น



รูปที่ 2.11 แสดงผลจากการปรับสมดุลสถิตยศาสตร์^๗

2.2.2 สมดุลแบบพลศาสตร์ (dynamics balancing) ในการปรับสมดุลสองระนาบต้องทำให้เงื่อนไขเป็นจริง

1. ผลรวมของแรงต้องมีค่าเป็นศูนย์

$$\Sigma \vec{F} = \vec{0} \quad (2.24)$$

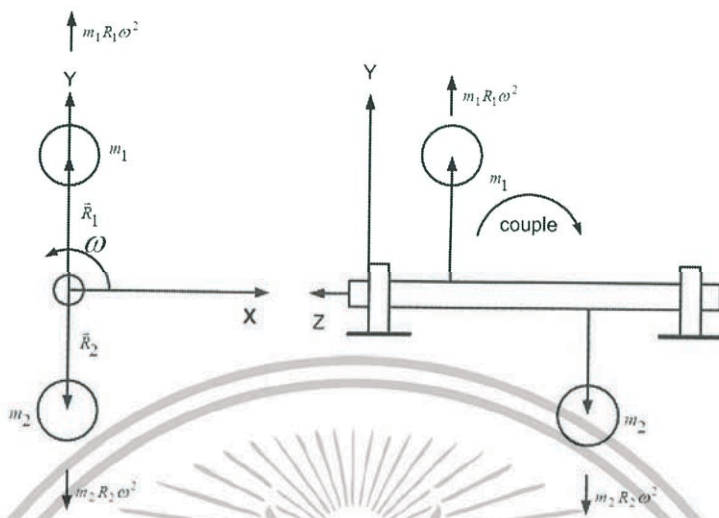
2. ผลรวมของโมเมนต์รอบจุดใดจุดหนึ่งต้องเป็นศูนย์

$$\Sigma \vec{M}_0 = \vec{0} \quad (2.25)$$

วัตถุหมุนมีความยาวมากตามแนวแกนการหมุนเมื่อเปรียบเทียบกับขนาดในแนวรัศมีจะต้องนำมาทำ Dynamic balancing

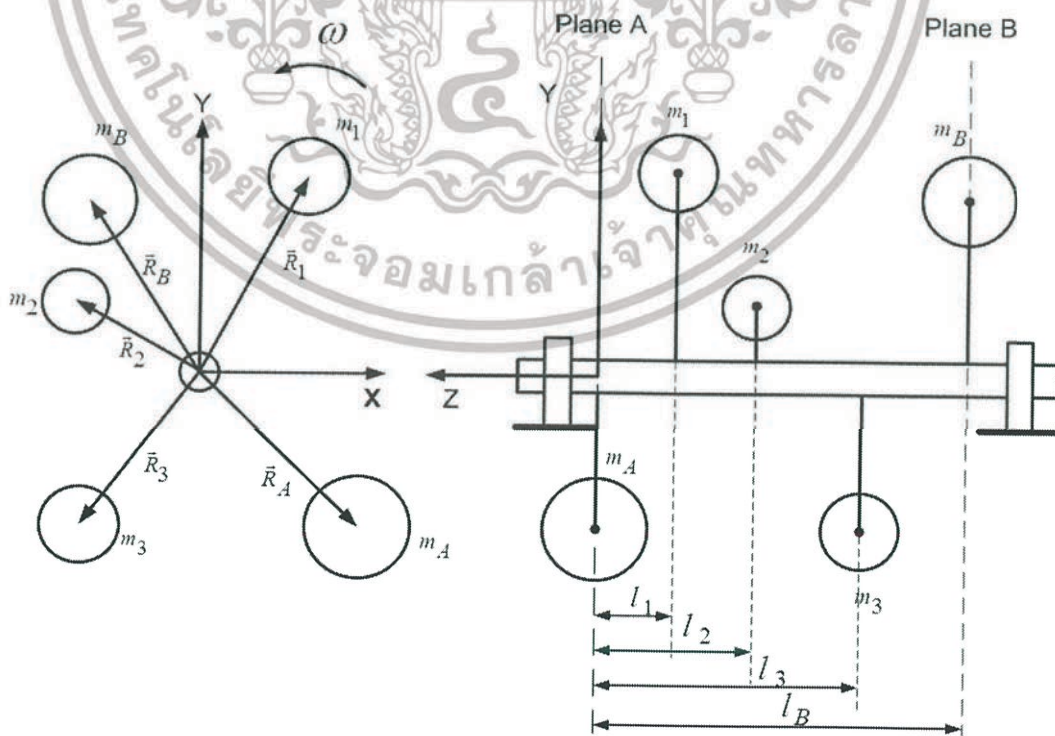
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามรูปมีคุณสมบัติเป็น Static balance แต่ไม่มีคุณสมบัติเป็น Dynamic balance



รูปที่ 2.12 การปรับสมดุลพลศาสตร์^[๓]

ดังนั้นการปรับสมดุลให้มันมีคุณสมบัติเป็น Dynamic balance ซึ่งทำได้โดยการเพิ่มมวลหรือลดมวลเข้าไปในระบบ บนสองระนาบเท่านั้น แต่สิ่งที่จะต้องหาคือตำแหน่งที่เหมาะสมที่จะทำให้ผลรวมของแรงและผลรวมของโมเมนต์มีค่าเป็นศูนย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 2.13 การสมดุลพลศาสตร์^[๓] ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณารูปข้างบนมวล m_1, m_2 และ m_3 คือมวลที่ทำให้เกิดความไม่สมดุลหรือเรียกว่าเป็น unbalance masses ทำโดยการเพิ่มมวล m_a และ m_b ไปบนระนาบ A และ B เพื่อทำให้มีความสมดุล

$$\Sigma \vec{F} = \vec{0} \quad (2.26)$$

หรือ

$$m_1 \vec{R}_1 \omega^2 + m_2 \vec{R}_2 \omega^2 + m_3 \vec{R}_3 \omega^2 + m_a \vec{R}_a \omega^2 + m_b \vec{R}_b \omega^2 = 0 \quad (2.27)$$

แตกสมการ (2.19) ออกตามแนวแกน x และ y ได้เป็น

$$m_a R_{ax} + m_b R_{bx} = -m_1 R_{1x} - m_2 R_{2x} - m_3 R_{3x} \quad (2.28)$$

$$m_a R_{ay} + m_b R_{by} = -m_1 R_{1y} - m_2 R_{2y} - m_3 R_{3y} \quad (2.29)$$

จากสมการ (2.20) และ (2.21) ค่าในเทอมด้านขวาของสมการจะเป็นค่าที่รู้ค่า แต่ต้องหาคือ

$$m_a R_{ax}, m_a R_{ay}, m_b R_{bx}, m_b R_{by}$$

หาสมมูลโมเมนต์รอบจุด O ซึ่งจะเขียนได้ดังนี้

$$(-l_1 \vec{k}) \times (m_1 \vec{R}_1 \omega^2) + (-l_2 \vec{k}) \times (m_2 \vec{R}_2 \omega^2) + (-l_3 \vec{k}) \times (m_3 \vec{R}_3 \omega^2) + (-l_b \vec{k}) \times (m_b \vec{R}_b \omega^2) = 0 \quad (2.30)$$

ทำการ cross vector จะได้โมเมนต์รอบแกน X และ y เท่านั้นซึ่งเทอมโมเมนต์รอบแกน y มีค่าดังนี้

$$m_b R_{bx} = \frac{-(m_1 R_{1x})l_1 - (m_2 R_{2x})l_2 - (m_3 R_{3x})l_3}{l_b} \quad (2.31)$$

เทอมโมเมนต์รอบแกน x มีค่าดังนี้

$$m_b R_{by} = \frac{-(m_1 R_{1y})l_1 - (m_2 R_{2y})l_2 - (m_3 R_{3y})l_3}{l_b} \quad (2.32)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นสามารถค่าในสมการ (2.24) และ (2.25) ไปแทนในสมการที่ (2.28) และ (2.29) เพื่อหาค่า $m_a R_{ax}$ และ $m_a R_{ay}$ นอกจากนั้นยังสามารถหาค่ามุมได้ดังนี้

$$\theta_a = \tan^{-1} \left(\frac{m_a R_{ay}}{m_a R_{ax}} \right) \quad (2.33)$$

$$\theta_b = \tan^{-1} \left(\frac{m_b R_{by}}{m_b R_{bx}} \right) \quad (2.34)$$

2.3 LabVIEW

LabVIEW คือชุดพัฒนาแอปพลิเคชัน (Development environment) จากเนชั่นแนลอินสตรูเมนต์ส (National Instruments) เพื่อใช้ในการสร้างระบบการวัด, ทดสอบ, และควบคุมโดยใช้การเขียนโปรแกรมด้วยภาษาโค้ดรูปภาพ (Graphical programming) และมีการต่อสายส่งค่าข้อมูลคล้ายกับการวาดโพลชาร์ตที่เข้าใจได้ง่ายซึ่งชุดพัฒนาจะมาพร้อมกับชุดฟังก์ชันทางคณิตศาสตร์และวิศวกรรมนับร้อยแบบสำหรับการวิเคราะห์ประมวลผล, และแสดงข้อมูลรวมถึงความสามารถต่าง ๆ ในการใช้งานร่วมกับฮาร์ดแวร์และซอฟต์แวร์อื่น ๆ ได้ดีทำให้ LabVIEW ซึ่งออกสู่ตลาดมาตั้งแต่ปี 1986 ถูกนำไปใช้งานอย่างกว้างขวางในแทบทุกส่วนของอุตสาหกรรมและการวิจัยที่ต้องการระบบอัตโนมัติในการวัดและการควบคุม อาทิเช่นในอุตสาหกรรมการผลิต, อิเล็กทรอนิกส์, ยานยนต์, การทหารและอากาศยาน, พลังงาน, การสื่อสาร, เครื่องมือแพทย์หรือแม้แต่ในของเด็กอย่างเล่นๆ ไปจนถึงเครื่องจักรที่ซับซ้อนที่สุดอย่าง เครื่องเร่งอนุภาค (LHC) ที่เชิร์นในปัจจุบันนอกจาก LabVIEW จะถูกใช้งานบนพีซีทั่วไปเพื่อการดึงสัญญาณหรือควบคุมเครื่องมือวัดแล้วแนวคิดของเนชั่นแนลอินสตรูเมนต์สที่เรียกว่า การออกแบบระบบกราฟิก (Graphical System Design) ยังขยายขีดความสามารถของแพลตฟอร์ม LabVIEW ให้สามารถทำการออกแบบได้ทั้งระบบตั้งแต่ส่วนของการออกแบบซอฟต์แวร์, การคำนวณไปจนถึงการดึงสัญญาณและการติดตั้งลงไบนารีในฮาร์ดแวร์ทั้งแบบวินโดวส์, เรียลไทม์, และ FPGA โดยใช้แพลตฟอร์ม LabVIEW เพียงแพลตฟอร์มเดียวจึงทำให้สามารถทำงานต่อยอดได้ทันทีเมื่อเวลาที่ต้องการย้ายจากการทำงานกับพีซีบนวินโดวส์ไปเป็นฮาร์ดแวร์แบบฝังตัวเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 โปรแกรม LabView

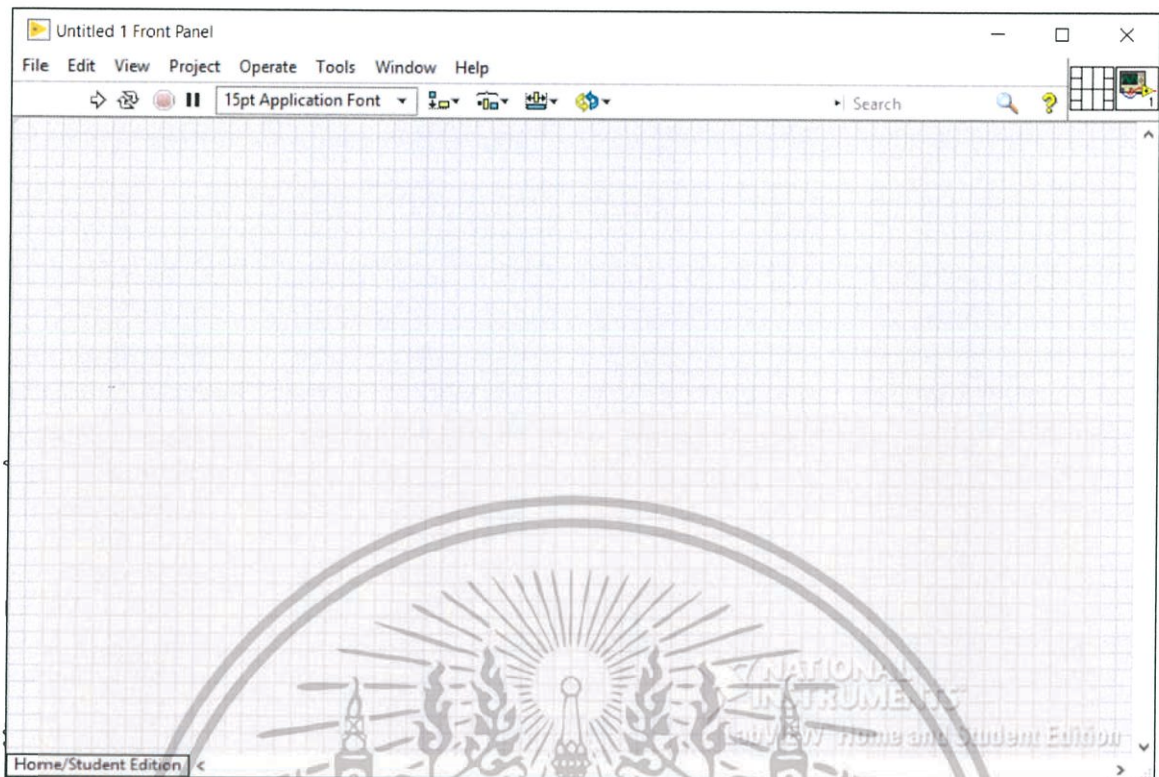
2.3.1 หลักการและส่วนประกอบในการเขียน LabVIEW

โปรแกรมที่เขียนขึ้นมาโดย LabVIEW จะเรียกว่า Virtual Instrument (VI) เพราะ ลักษณะที่ปรากฏทางจอภาพเมื่อผู้ใช้ใช้งานจะเหมือนกับเครื่องมือหรืออุปกรณ์ทางวิศวกรรม ใน ขณะเดียวกันหลังจากของอุปกรณ์เสมือนจริงเหล่านั้นจะเป็นการทำงานของ ฟังก์ชัน, Subroutines และโปรแกรมหลัก เหมือนกับภาษาทั่วไป สำหรับ VI หนึ่งๆ จะประกอบด้วยส่วนประกอบ 3 ส่วน คือ

- Front Panel
- Block Diagram
- Icon และ Connector

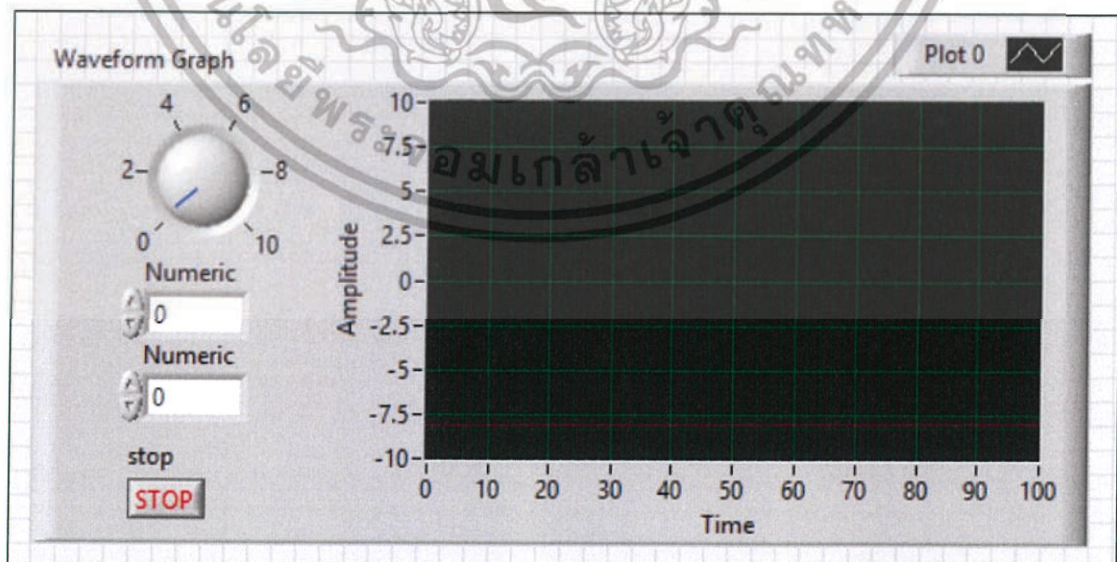
ทั้งสามส่วนนี้จะประกอบกันขึ้นมาเป็นอุปกรณ์เสมือนจริง ลักษณะและหน้าที่ของส่วนประกอบทั้ง สามมีดังต่อไปนี้

1) Front Panel หรือหน้าปัทม์ จะเป็นส่วนที่ใช้สื่อความกันระหว่างผู้ใช้กับโปรแกรม (หรือที่นิยมเรียก User Interface) โดยทั่วไปจะมีลักษณะเหมือนกับหน้าปัทม์ของของเครื่องมือหรือ อุปกรณ์ที่ใช้งานด้านการวัดทั่ว ๆ ไป โดยทั่วไปจะประกอบด้วย สวิตซ์ปิดเปิด, ปุ่มบิด, ฆุ่มกด จอแสดงผลหรือแม้แต่ค่าที่ผู้ใช้สามารถกำหนด สำหรับผู้ที่คุ้นเคยกับการเขียนโปรแกรมประเภท Visual ทั้งหลายคงจะเข้าใจกันดีว่า Front Panel นี้จะเปรียบเสมือนเป็น GUI ของโปรแกรมหรือ VI นั้นเอง



รูปที่ 2.15 Front Panel ของ LabVIEW

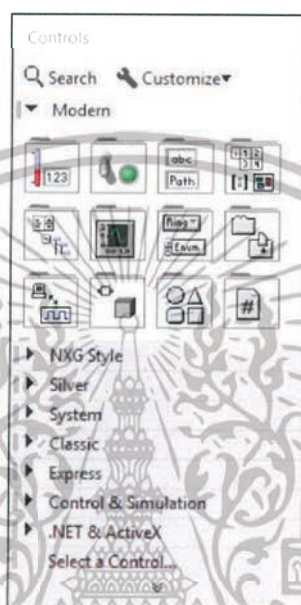
- Decorations เป็น Object ที่ไม่เกี่ยวข้องกับโปรแกรมและcode บน Block-Diagram เลย แต่มีไว้เพื่อความสวยงามเป็นระเบียบของ Front panel เท่านั้น ลักษณะของ Front Panel แสดงดังรูปต่อไปนี้เป็น



รูปที่ 2.16 Object ที่อยู่บน Front Panel ของ LabVIEW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือที่ใช้ออกแบบ Front Panel เครื่องมือที่ใช้ในการออกแบบ Front Panel จะประกอบไปด้วย Control Palette และ Tools Palette ซึ่ง LabVIEW มี Controls Palette ที่ใช้ในการออกแบบ Front Panel แสดงดังรูป 2.16 ซึ่งเป็น ส่วนที่ติดต่อกับผู้ใช้งาน (User Interface) โดยจะจัดเป็นกลุ่มต่างๆ เช่น กลุ่มของตัวเลข (Numeric) ซึ่งภายในกลุ่มจะมี Control และ Indicator ต่าง ๆ ที่เกี่ยวกับตัวเลข



รูปที่ 2.17 Controls Palette ที่ใช้ในการออกแบบ Front Panel

Tools Palette คือ เครื่องมือที่ใช้ในการพัฒนาโปรแกรม ซึ่งจะใช้ทั้งการออกแบบ Front Panel และ Block Diagram ในส่วนนี้จะกล่าวถึง Tools Palette สำหรับออกแบบ Front Panel



รูปที่ 2.18 Tools Palette ที่ใช้ในการออกแบบ Front Panel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) โหนดแผนภาพบล็อก (Block Diagram Node)

โหนด (Node) คือรูป ไอคอน(Icon) ที่อยู่บน แผนภาพบล็อก(Block Diagram) ซึ่งมี สิ่งที่ป้อนเข้า(Input) และ/หรือ ข้อมูลที่ส่งออกมา(Output) และจะทำงานตามหน้าที่เมื่อมีการรันโปรแกรมโดนแบ่งเป็นสามชนิดหลัก

- ฟังก์ชัน(Function) คือ โหนด(Node) ที่มีหน้าที่พื้นฐานของคอมพิวเตอร์ซึ่งไม่สามารถที่จะเจาะเข้าไปดูรายละเอียดภายในได้อีกเช่นการบวกการคูณ

- ซับวีไอ(SubVIs) หรือในภาษาทางซอฟต์แวร์อาจจะเรียกว่า ย่อย(Subroutine) หรือ โปรแกรมย่อย(Subprogram) คือโปรแกรมย่อยที่ถูกเขียนขึ้นมาเพื่อถูกนำมาเรียกใช้ในอีกโปรแกรมหนึ่งสามารถเปิดเข้าไปดู แผงด้านหน้า(Front panel) และ แผนภาพบล็อก(Block diagram) ได้เมื่อกดสองครั้ง (double click) ที่ ไอคอน(Icon) ของมัน

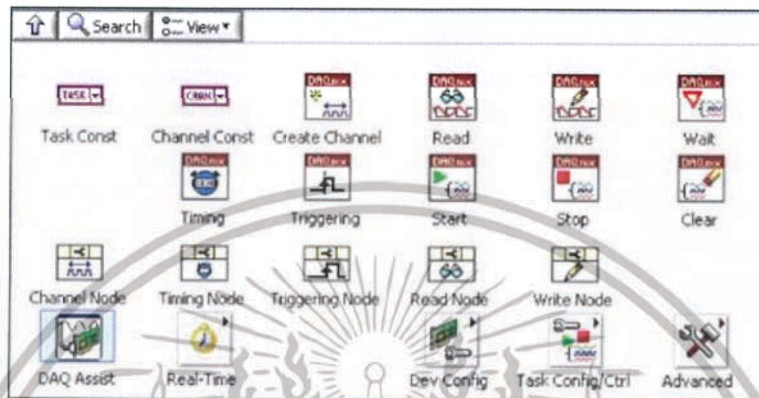
- เอ็กเพรสวีไอ(Express VIs) เป็น ซับวีไอ(SubVIs) ประเภทพิเศษคือเมื่อเลือก เอ็กเพรสวีไอ (Express VI) มาวางบน แผนภาพบล็อก(Block Diagram) มันจะปรากฏหน้าต่างองค์ประกอบ(Configuration) ขึ้นมาเพื่อนให้เข้าไปป้อนค่า ตัวแปร(Parameters) ต่างตามต้องการและเมื่อป้อนค่าเสร็จมันก็จะสร้างโค้ดไว้ภายในอัตโนมัติตามที่ได้อิงค่าไว้ซึ่งความสามารถของ เอ็กเพรสวีไอ (Express VIs) นี้ทำให้แทบไม่จำเป็นต้องต่อสาย เอ็กเพรสวีไอ(Input) เลยเพราะ ตัวแปร(Parameter) ทั้งหมดได้ถูกสร้างขึ้นมาแล้วถูกเก็บไว้ภายในเรียบร้อยแล้วจึงทำให้การเขียน LabVIEW ง่ายและเร็วขึ้นมากสังเกตง่าย ๆ เอ็กเพรสวีไอ(Express VIs) จะมี ไอคอน (Icon) ขนาดใหญ่ที่มีพื้นหลังเป็นสีฟ้า



รูปที่ 2.19 ตัวอย่าง Block Diagram Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือที่ใช้ในการเขียนโปรแกรมบน Block Diagram LabVIEW ใช้ Functions Palette ซึ่งจะมี Function และ SubVI ต่าง ๆ ที่มีอยู่แล้วให้ผู้ใช้ เลือกใช้โดย Function และ SubVI จัดเป็นกลุ่ม ๆ เช่น Numeric Function จะมี Function ต่าง ๆ เกี่ยวกับตัวเลข เช่น บวก ลบ คูณ หาร แสดงดังรูปที่ 2.19



รูปที่ 2.20 เครื่องมือสำหรับ Dam – Data Acquisition

Tools Palette สำหรับ Block diagram เปิด Tools Palette โดยการ Click ที่ Window >> Show Tools Palette หรือกด Shift Right Click แล้ว Click เลือก Tool ที่ต้องการใช้ (ถ้า Automatic Tools Selection เปิดอยู่แล้ว (LED) เป็นสีเขียว ให้ Click เพื่อปิด Automatic Tool Selection ซึ่งมีเฉพาะใน Version 6.1 ขึ้นไป) Tool ที่ใช้มีดังต่อไปนี้

1. Operation Tool ใช้ในการเปลี่ยนแปลงค่าหรือเลือกค่าคงที่ใน Block Diagram
2. Position/Size/Select ใช้ในการเลือก/เคลื่อนย้าย/จัดขนาดของสิ่งที่สร้างขึ้นบน Block Diagram
3. Edit Text Tool ใช้ในการแก้ไขข้อความที่เป็นตัวอักษร หรือเพิ่มข้อความลงบน Front Panel
4. Wiring Tool ใช้ในการโยงสาย (Wiring) ระหว่าง Terminal หรือ Node ซึ่งสายที่โยงนี้ จะเป็นเส้นทางเดินของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Icon และ Connector เปรียบเสมือนโปรแกรมย่อย Subroutine ในโปรแกรมปกติทั่ว ๆ ไป โดย Icon จะหมายถึง Block Diagram ตัวหนึ่งที่มีการส่งข้อมูลเข้าและออกผ่านทาง Connector ซึ่ง ใน LabVIEW จะเรียก Subroutine นี้ว่า SubVI ข้อดีของการเขียนโปรแกรมด้วยภาษา G นี้ก็คือ สามารถสร้าง VI ที่ละส่วนขึ้นมาให้ทำงานด้วยตัวเองได้อย่างอิสระ จากนั้นในภายหลังหากต้องการก็สามารถเขียนโปรแกรมอื่นขึ้นมาเพื่อเรียกใช้งาน VI ที่เคยสร้างขึ้นก่อนหน้านี้นี้ทีละตัว ซึ่งทำให้ VI ที่เขียนขึ้นก่อนกลายเป็น SubVI ไป การเขียนในลักษณะนี้เรียกว่า เขียนเป็น Module สำหรับลักษณะทั่วไปของ Icon และ Connector จะแสดงในรูปต่อไปนี้ จะเห็นว่าเมื่อแสดงในรูปของ Connector จะพบว่ามีช่องต่อข้อมูลหรือที่เรียกว่า Terminal ปรากฏให้เห็น



รูปที่ 2.21 แสดงลักษณะทั่วไปของ Icon และ Connector

คำศัพท์ต่าง ๆ ที่ใช้กันใน LabVIEW นี้ออกจะแตกต่างจากที่ใช้กันในภาษาการเขียนโปรแกรม ตัวหนังสือทั่วไปในหลายๆด้าน ดังนั้นเพื่อให้ผู้ที่เริ่ม LabVIEW เข้าใจถึงศัพท์ต่าง ๆ ที่ใช้ใน โปรแกรม จึงขอเปรียบเทียบศัพท์ที่ใช้ใน LabVIEW กับโปรแกรมพื้นฐานทั่ว ๆ ไปตามตารางที่ได้แสดงต่อไปนี้

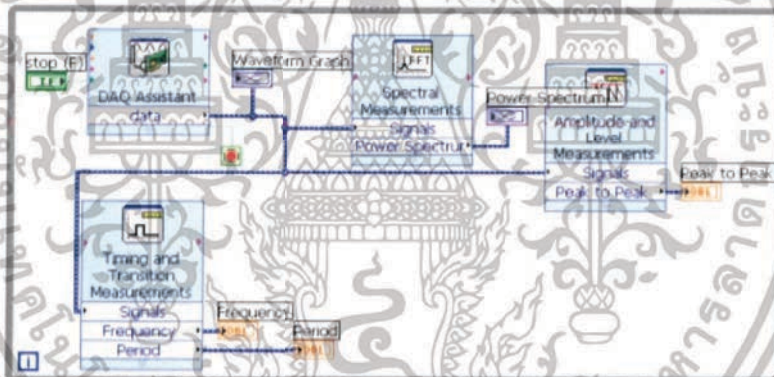
2.3.2 การไหลของข้อมูลและการเขียนโปรแกรม (Data flow and programming)

เนื่องจาก LABVIEW เป็นโปรแกรมที่ใช้รูปภาพหรือสัญลักษณ์แทนการเขียนด้วยตัวอักษรเหมือนโปรแกรมปกติทั่วไปซึ่งข้อดีข้อแรกก็คือการลดความผิดพลาดด้านการสะกดผิดหรือพิมพ์ผิดออกไปข้อแตกต่างอีกประการหนึ่งที่สำคัญของการเขียนโปรแกรมแบบ G กับการเขียนด้วยตัวหนังสือก็คือการเขียนด้วยภาษา G นี้เป็นการเขียนโดยใช้หลักการของการไหลของข้อมูล(Data Flow) ซึ่งเมื่อเริ่มส่งข้อมูลเข้าสู่โปรแกรมจะต้องกำหนดทิศทางไหลของข้อมูลว่าจะไปที่ส่วนใดผ่านการประเมินผลและคำนวณในส่วนใดบ้างและจะให้แสดงผลอย่างไรซึ่งลักษณะการเขียนภาษา G หรือ การไหลของข้อมูล(Data Flow) นี้จะมีลักษณะเหมือนกับการเขียน แผนภาพบล็อก(Block Diagram) ซึ่งทำให้ผู้เขียนโปรแกรมสามารถให้ความสนใจกับกาเคลื่อนที่และเปลี่ยนแปลงข้อมูลได้โดยไม่ต้องจดจำรูปแบบคำสั่งที่ยุ่งยากเนื่องจาก LabVIEW ใช้ลักษณะการเขียนแบบ แผนภาพบล็อก(Block Diagram) ซึ่งวิศวกรส่วนใหญ่มีความคุ้นเคยอยู่แล้วจึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

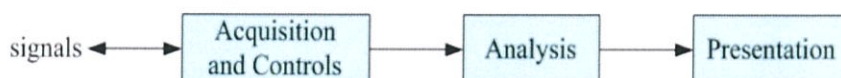
เป็นการง่ายที่จะทำความเข้าใจและนำไปพัฒนาใช้ต่อไปได้และถ้าหากจำได้ถึงขั้นตอนการเขียนโปรแกรมว่าก่อนที่จะเขียนโปรแกรมจะต้องเขียน ผังงาน(Flow Chart) ให้เสร็จสิ้นก่อนหลังจากตรวจสอบ ผังงาน (Flow Chart) เรียบร้อยแล้วจึงนำไปเขียนโปรแกรมซึ่งจะมีความสะดวกมากขึ้นถ้าหากการเขียน

ผังงาน (Flow Chart) ของ LabVIEW ก็คือการเขียนโปรแกรมนั่นเองซึ่งเป็นการลดขั้นตอนการทำงานลงไปได้เป็นอย่างมากแม้ว่าการเขียนโปรแกรมใน LabVIEW ไม่จำเป็นต้องมีความรู้ด้านการเขียนโปรแกรมใด ๆ มาก่อนเลยแต่การมีความรู้ด้านการเขียนโปรแกรมหรือใช้โปรแกรมสำเร็จรูปอื่น ๆ จะสามารถนำมาใช้ประโยชน์ได้เป็นอย่างดี LabVIEW จะมี แผงด้านหน้า(Front panel) ซึ่งเปรียบเสมือนได้กับสิ่งที่ผู้ใช้จะเห็นและควบคุมการทำงานผู้ใช้สามารถสร้างรูปแบบขึ้นเองได้อย่างรวดเร็วเพราะ LabVIEW มีส่วนประกอบต่าง ๆ ที่ใช้สำหรับออกแบบหน้าจอบางอย่างเช่นจอแสดงผลแบบออปติคอลโครบ, ปุ่มหมุน (Dial) และสวิตช์เป็นต้นโดย LabVIEW จะแสดงผลและควบคุมการทำงานผ่านทางคอมพิวเตอร์พื้นที่ส่วนเขียนโปรแกรมจะเรียกว่า แผนภาพบล็อก(Block Diagram) เปรียบเสมือนกับ ส่วนอุปกรณ์(Hardware) ภายในเครื่องมือวัดโดย LabVIEW จะเขียนโปรแกรมโดยอาศัยรูปภาพ



รูปที่ 2.22 ตัวอย่าง Block diagram ในโปรแกรม LABVIEW

LabVIEW อาศัยหลักการทำงานของเครื่องมือวัดหรือการวัดคุมทำให้ผู้ใช้สามารถออกแบบตามที่ต้องการหลักการดังกล่าวแบ่งออกเป็น 3 ส่วนใหญ่ๆคือ



รูปที่ 2.23 หลักการภายในโปรแกรม LABVIEW

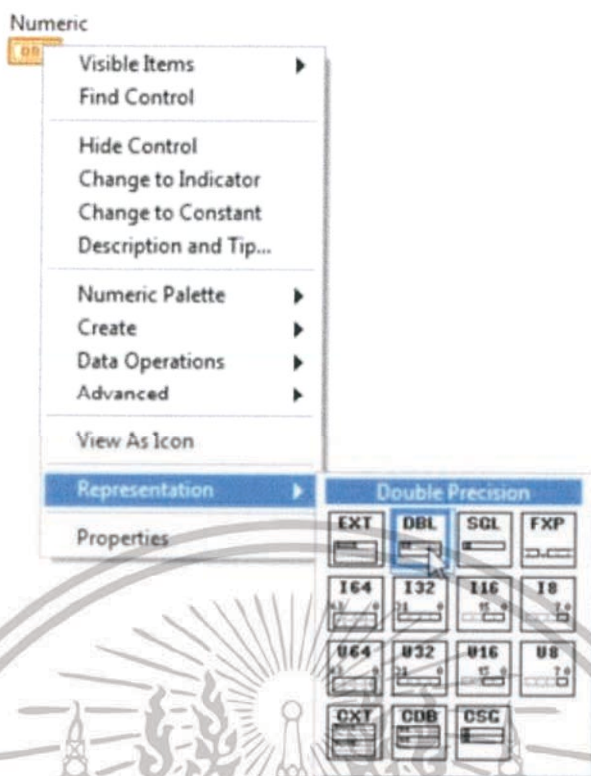
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สิ่งที่ได้มา(Acquisition) ซึ่งเป็นส่วนที่รับข้อมูล (Input) จากสิ่งแวดล้อมภายนอกเข้าสู่ระบบคือคอมพิวเตอร์โดยข้อมูลที่เข้าสู่ระบบนี้อาจมาจากการ์ด DAQ (สำหรับสัญญาณทางไฟฟ้า)
2. การวิเคราะห์(Analysis) หลังจากที่ได้รับข้อมูลแล้วอาจจะผ่านฟังก์ชันในการวิเคราะห์ข้อมูลซึ่งจะแสดงผลในรูปที่สื่อความหมายในสิ่งที่ผู้ใช้งานสามารถนำไปแสดงแทนสื่อที่วัดได้และใช้งานได้
3. นำเสนอ(Presentation) คือการแสดงผลในรูปแบบที่เป็นประโยชน์ต่อผู้ใช้งานโดยอาจแสดงบนหน้าจอคอมพิวเตอร์เช่น ดิจิตอลมัลติมิเตอร์(Digital Multimeter) แสดงผลเฉพาะที่วัดได้โดยไม่จำเป็นต้องรู้ความสำคัญกับเวลาหรือ การวิเคราะห์สเปกตรัม(Spectrum Analysis) จะแสดงสัญญาณในรูปความถี่หรือการพิมพ์ออกมาเป็นรายงานหรือเก็บข้อมูลในฮาร์ดดิสก์

2.3.3 รูปแบบข้อมูล (Data type)

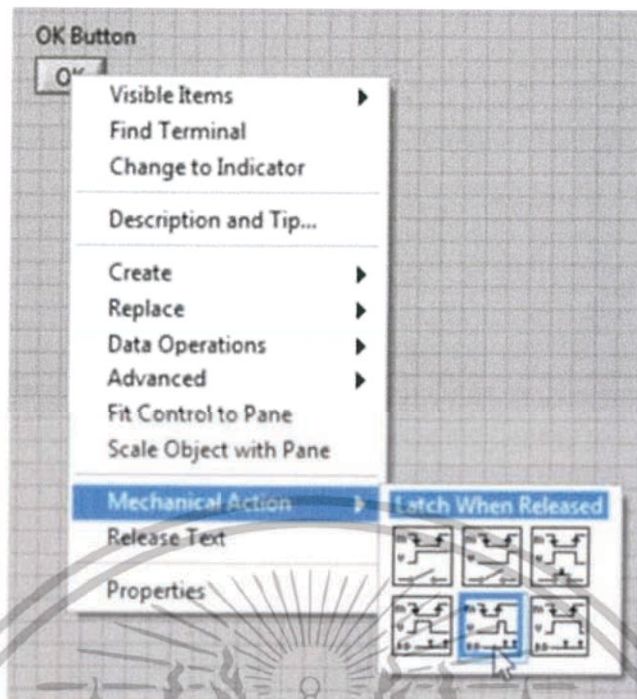
1) Numeric คือข้อมูลประเภทตัวเลข เมื่อสร้าง Numeric Control/Indicator/Constant VI ขึ้นมา ค่า เริ่มต้นเดิม (default) ของมันจะเป็นศูนย์ข้อมูลแบบ Numeric มีทั้งประเภทจำนวนเต็มซึ่งไอคอนและ สายใน Block Diagram จะเห็นเป็นสีน้ำเงิน และประเภทจำนวนที่มีทศนิยมซึ่งจะแสดงเป็นสีส้ม ว่าการเปลี่ยนประเภทของตัวเลข ให้คลิกขวาที่ Numeric Control/Indicator/Constant (บน Front Panel หรือ Block Diagram ก็ได้) แล้วเลือก Representation และเลือกประเภทตัวเลข ได้เลย ซึ่งแถบบนจะเป็นตัวเลขประเภททศนิยม (จำนวนจริง) แถวสองคือเลขจำนวนเต็ม (integer) แถวสาม คือจำนวนเต็มแบบไม่มีเครื่องหมาย (unsigned integer) และแถวล่างสุดคือจำนวนเชิงซ้อน (complex number)

สำหรับผู้ที่ไม่เคยเขียนโปรแกรมมาก่อน เบื้องต้นแนะนำว่าถ้าต้องการใช้ตัวเลขแบบมีทศนิยมให้เลือก DBL (Double Precision) และถ้าต้องการจำนวนเต็มให้เลือก I32 (Long) ไปก่อน ส่วนการกำหนดค่า ต่าง ๆ ของ Numeric Control/Indicator อย่างละเอียด เช่น การแสดงจำนวนจุดทศนิยม ค่าสูงสุด ต่ำสุด ฯลฯ ให้เลือกที่เมนู Properties ซึ่งจะมีหน้าต่างสำหรับให้ปรับแต่งรายละเอียดของ object นั้น



รูปที่ 2.24 แสดงข้อมูลประเภท Numeric

2) Boolean คือข้อมูลประเภทที่มีสองค่า คือ TRUE และ FALSE ค่า default เดิมของมันคือ FALSE. สำหรับบน Block Diagram จะแสดงสีของไอคอนและสายของข้อมูลประเภทนี้ด้วยสีเขียว และสำหรับ บน Front Panel ตัว Boolean Control จะมีคุณสมบัติสวิตช์ (Mechanical Action) ซึ่งก็มีหลายประเภท วิธีเปลี่ยนโหมดของสวิตช์ให้คลิกขวาที่ Boolean Control บน Front Panel แล้วเลือก Mechanical Action ซึ่งจะมีอยู่ 6 รูปแบบ



รูปที่ 2.25 แสดงข้อมูลประเภท Boolean

- Switch When Pressed คือสวิตช์แบบกดติด-กดดับ
- Switch When Released คือกดติด-กดดับเหมือนกัน แต่จะมีผลเมื่อปล่อยมือแล้ว
- Switch Until Released คือกดติด-ปล่อยดับ
- Latch When Pressed สวิตช์จะเปลี่ยนค่าทันทีเมื่อคลิก แล้วจะตั้งกลับเป็นค่าเดิมเองเมื่อโปรแกรมรับรู้ ถึงแม้ว่าจะยังไม่ปล่อยมือก็ตาม
- Latch When Released หลังจากกดแล้วสวิตช์ จะเปลี่ยนค่าก็ต่อเมื่อปล่อยมือ แล้วตั้งกลับ เป็นค่าเดิม
- Latch When Released หลังจากกดแล้วสวิตช์จะเปลี่ยนค่าก็ต่อเมื่อปล่อยมือ แล้วตั้งกลับ เป็นค่าเดิมอีกทีเมื่อโปรแกรมรับรู้
- Latch Until Released คล้ายกับกดติด-ปล่อยดับ แต่จะมีการรอให้โปรแกรมอ่านค่าตอน ปล่อยมือก่อน แล้วเปลี่ยนกลับเป็นค่าเดิม

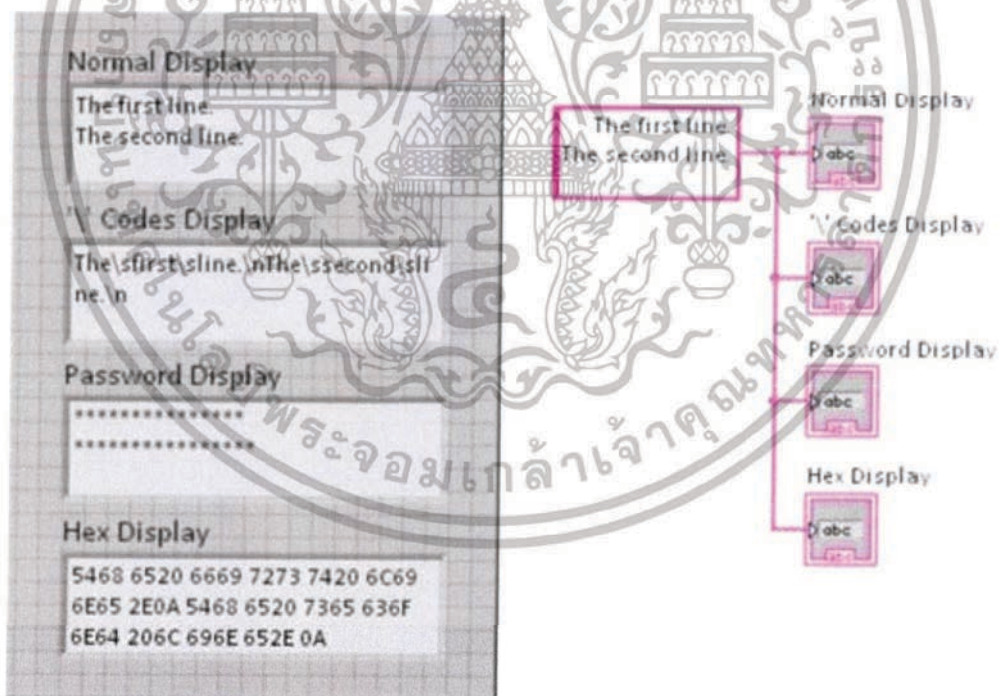
ทดลองเล่นแต่ละแบบดูได้โดยคลิกขวาที่ Boolean Control แล้วเลือก Properties แล้วเลือก

แท็บ Operation และทดลองกับปุ่มตัวอย่างบนส่วน Preview Selected Behavior ซึ่งปุ่มต่าง ๆ ที่ใช้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Windows อย่างปุ่ม OK, Cancel และ Help พวกนี้มี Mechanical Action เป็นแบบ Latch when released ทั้งสิ้น ซึ่ง Mechanical Action เป็นที่นิยมในการใช้ dialog ต่าง ๆ

3) String คือข้อมูลที่เป็นตัวอักษร ค่า default คือ ว่างเปล่า (empty string) ไอคอนและสายของ String จะเป็นสี่ขมพู สำหรับการแสดงผลของ String บน Front Panel หรือบน Block Diagram (กรณีที่เป็น String Constant) จะมีอยู่สี่รูปแบบ

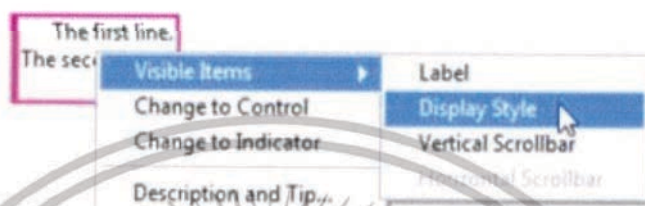
- Normal Display คือการแสดงผลแบบปกติ
- V Codes Display คือการแสดงผลแบบโค้ด มีประโยชน์สำหรับบ่อนตัวอักษรบางตัวที่ พิมพ์โดยตรงไม่ได้ หรือแสดงตัวอักษรที่มองไม่เห็น เช่น การเว้นวรรค (\s) แท็บ (\t) หรือตัว ขึ้นบรรทัดใหม่ (\n) เป็นต้น
- Password Display จะแทนตัวอักษรด้วย เครื่องหมายดอกจัน
- Hex Display แลดงผลเป็นรหัส ASCII ด้วยเลขฐานสิบหก



รูปที่ 2.26 การเปลี่ยนรูปแบบการแสดงผลของ String Control/Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

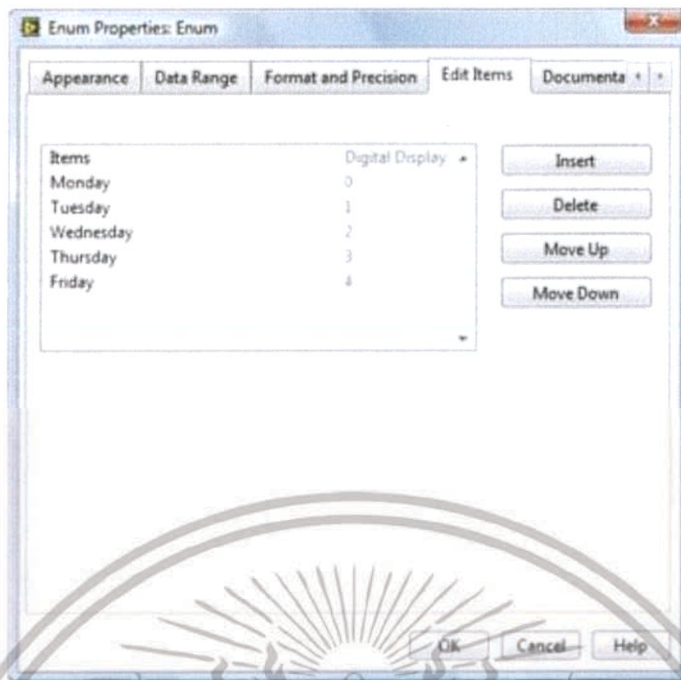
การเปลี่ยนรูปแบบการแสดงผลของ String Control/Indicator ให้คลิกขวาที่ String บน Front Panel แล้วเลือกเปลี่ยนเป็นแบบที่ต้องการจากเมนูได้ทันที สำหรับ String Constant ให้คลิกขวาบน Block Diagram ในการเขียนโค้ด อาจพิจารณาใส่สัญลักษณ์ radix กำกับไว้บน String Constant เพื่อระบุวิธีการแสดงผลและป้องกันการสับสนเวลาอ่านโค้ดวิธีการคือให้คลิกขวาที่ String แล้วเลือก Visible Items » Display Style



รูปที่ 2.27 String Constant เพื่อระบุวิธีการแสดงผลและป้องกันการสับสนเวลาอ่านโค้ด

จากนั้นจะเห็นสัญลักษณ์ของ Display Style ตัวเล็ก ๆ อยู่ตรงด้านซ้ายของ String ซึ่งจะมีสัญลักษณ์ n, \, *, และ x ตามลำดับ

4) Enum คือข้อมูลประเภทที่แสดงให้ผู้ใช้เห็นเป็นตัวหนังสือ แต่ค่าจริงของมันคือตัวเลขจำนวนเต็ม ดังนั้นบน Block Diagram จึงมองเห็น Terminal และสายของข้อมูลประเภทนี้เป็นสีน้ำเงินซึ่งเหมือนกับจำนวนเต็ม วิธีการสร้าง Enum Control คือให้เลือกมาจาก Controls Palette: Modem » Ring&Enun » Enum แล้ววางลงไปบน Front Panel แล้วคลิกขวาและเลือก Edit Items... จากนั้นที่หน้าต่าง Enum Properties ก็ให้พิมพ์ตัวหนังสือตรงคอลัมน์ซ้าย (Items) ซึ่งจะไปจับคู่กับตัวเลขทางคอลัมน์ขวา (Digital Display)



รูปที่ 2.28 หน้าต่างการตั้งค่า Enum

ซึ่งแต่ละไอเทมชื่อต้องไม่ซ้ำกันและถ้ามีไอเทมใดถูกเว้นว่างไว้ให้ลบทิ้งให้เรียบร้อย เมื่อสร้างเสร็จก็สามารถใช้งานได้ทันทีโดยใช้เมาส์เลือก

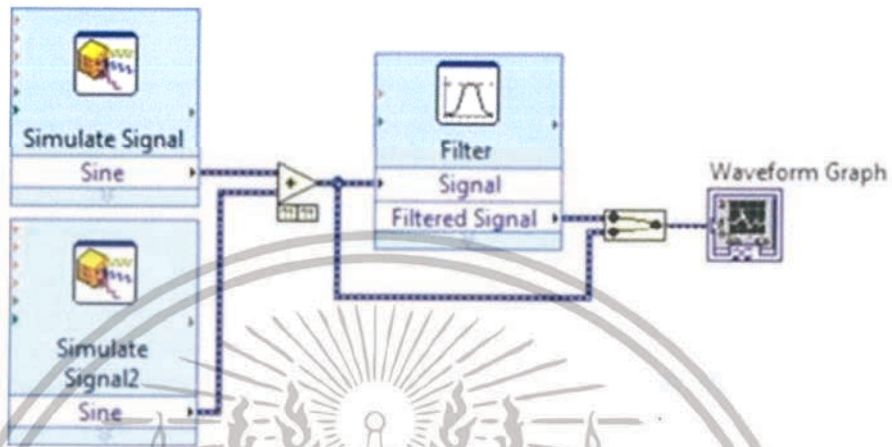


รูปที่ 2.29 การใช้งานของ Enum

Enum มีประโยชน์สำหรับการสร้าง Control ที่ต้องการจำกัด input ไม่ให้ผู้ใช้ใส่ข้อมูลอย่างอื่นนอกจากที่กำหนด ในกรณีที่ต้องการสร้าง Control ให้มีเมนูให้เลือกแบบ Enum แต่ยังไม่อนุญาตให้ผู้ใช้พิมพ์ข้อความอย่างอื่นเข้าไปเองด้วย ให้พิจารณาใช้ Combo Box ซึ่งเป็นข้อมูลแบบ String แทน (เลือกจาก Controls Palette: Modern » Siring & Path » Combo Box)

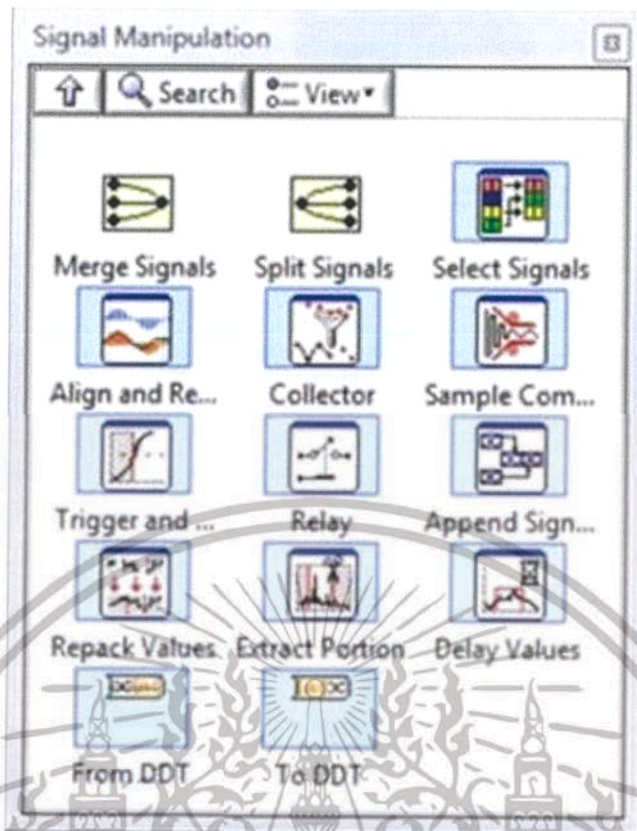
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) Dynamic (DDT) เป็นข้อมูลที่อยู่ในรูปของสัญญาณเวฟฟอร์ม สายบน Block Diagram ถูกแสดง ด้วยสีน้ำเงินเข้ม เส้นใหญ่ ซึ่งภายในประกอบด้วยข้อมูลหลายอย่าง เช่น Array ของเวฟฟอร์ม, Time Stamp ของสัญญาณ ฯลฯ ข้อมูลประเภท DDT นี้ส่วนใหญ่ใช้ใน Express VI สำหรับการอ่าน, กำเนิด และ วิเคราะห์สัญญาณ เป็นต้น



รูปที่ 2.30 การส่งข้อมูลแบบ DDT

สายข้อมูลแบบ DDT สามารถส่งข้อมูลหลาย ๆ ชานแนลได้ในเส้นเดียวโดยการ merge สัญญาณหลาย ช่องเข้าด้วยกัน เครื่องมือต่าง ๆ สำหรับการจัดการตัดแปลงแก้ไขข้อมูลแบบ DDT จะอยู่ใน Functions Palette: Express » Signal Manipulation ดังนี้

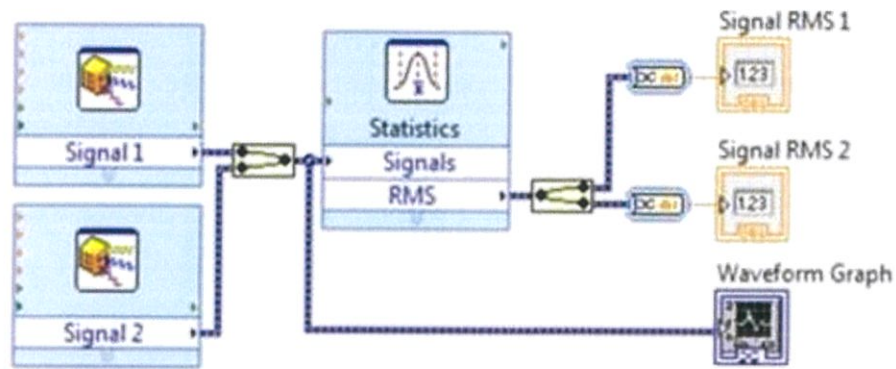


รูปที่ 2.31 เครื่องมือใน Signal Manipulation

ยกตัวอย่างเครื่องมือใน Signal Manipulation ที่ใช้บ่อยได้แก่

- Merge Signals: ใช้ในการรวมสัญญาณหลายช่องทางเข้าไว้ในเส้นเดียวกัน เช่นใช้ในการ พล็อตหลายสัญญาณในกราฟเดียวกัน
- Split Signals: ใช้ในการแยกสัญญาณ
- Sample Compression: ใช้สำหรับลดปริมาณจำนวน sample ของสัญญาณเวฟฟอร์ม
- From DDT: ใช้เปลี่ยน DDT ให้เป็นข้อมูลแบบปรกติ เช่น เปลี่ยนเป็น Numeric, Array. ฯลฯ ตามชนิดของข้อมูลนั้น ๆ เพื่อนำไปใช้ในส่วนที่ไม่ใช่ Express VI
- To DDT: ใช้แปลงข้อมูลแบบปรกติให้เป็น DDT เพื่อนำกลับไปใช้ใน Express VI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 ตัวอย่างการใช้เครื่องมือใน Signal Manipulation

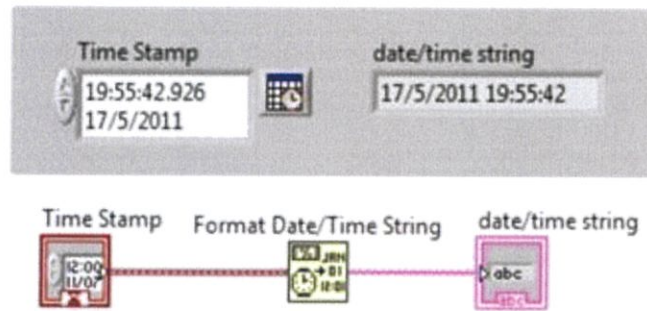
6) Time stamp เป็นข้อมูลที่ประกอบด้วยวันที่และเวลาที่มีความละเอียดถึงมิลลิวินาที LabVIEW คำนวณ Time Stamp โดยนับเป็นจำนวนวินาทีที่เริ่มตั้งแต่เที่ยงคืนวันที่ 1 มกราคม ค.ศ 1904 ในเวลามาตรฐาน หรือ UTC (เวลาในประเทศไทยเร็วกว่า UTC เจ็ดชั่วโมง) แล้วนำมาแปลงเป็นรูปแบบวันที่ และ เวลา ยกตัวอย่างเช่น ถ้าแปลงตัวเลข 3.4 พันล้านด้วยฟังก์ชัน To Time stamp ตั้งในภาพ วิธิตัดให้เริ่มนับจากเที่ยงคืน UTC วันที่ 1 มกราคม 1904 ไป 3.4 พันล้านวินาที ผลลัพธ์จะได้เป็นวันที่ 28 กันยายน ค.ศ. 2011 เวลาในประเทศไทย 3:26:40 นาฬิกา ดังนี้



รูปที่ 2.33 ตัวอย่างการใช้เครื่องมือใน Signal Manipulation

Time Stamp ยังสามารถนำมาแปลงให้เป็นวันที่และเวลาในรูปแบบ String ได้ด้วยฟังก์ชันใน Format Date/Time String ดังนี้

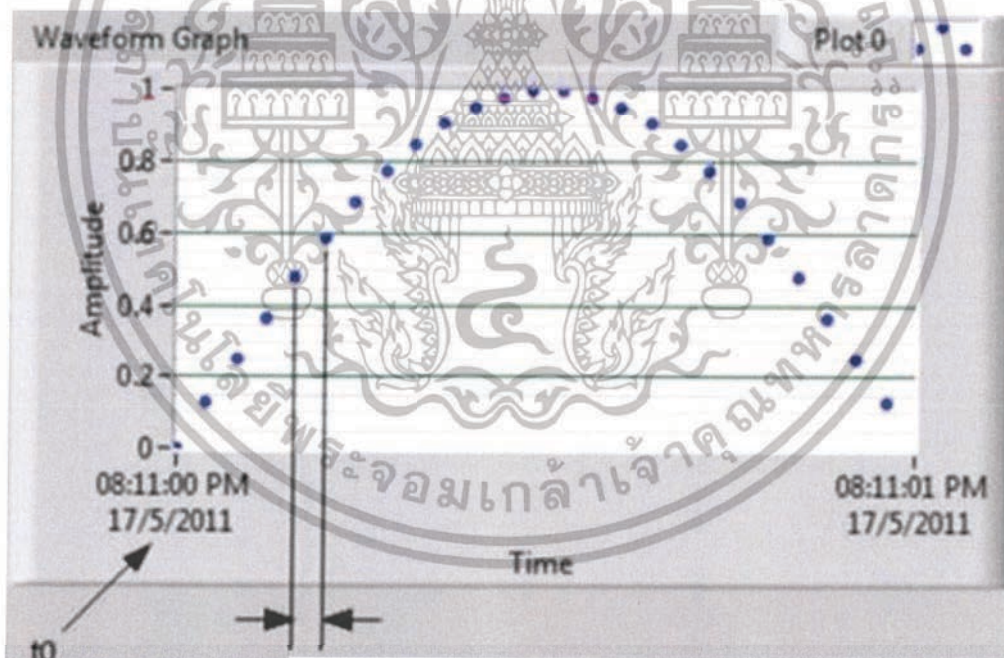
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 ตัวอย่างการใช้ Time Stamp

7) Waveform เป็นข้อมูลที่ประกอบด้วยข้อมูลย่อยอีกสามรายการหลัก คือ

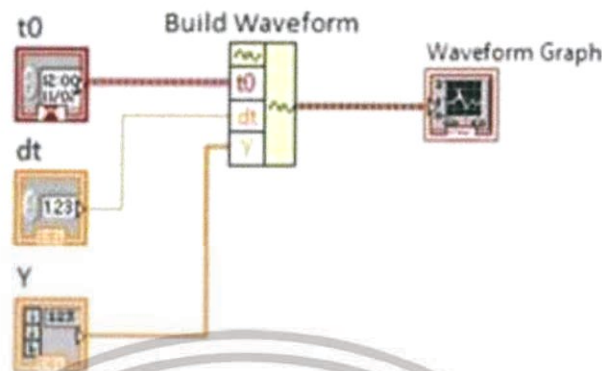
- Y: จุดของตัวเลขหลาย ๆ จุดประกอบเรียงกันเป็นเวฟฟอร์ม ซึ่งจะเรียกว่า Array
- dt: ข้อมูลที่ระบุว่าแต่ละจุดมีเวลาห่างกันกี่วินาที
- t0: ข้อมูลแบบ Time Stamp ที่ระบุว่าจุดแรกของชุดสัญญาณนี้ถูกสร้างขึ้นเมื่อเวลาใด



รูปที่ 2.35 ตัวอย่างการสร้าง Waveform graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถสร้าง Waveform ได้ด้วยบล็อก Build Waveform ที่เรียกมาจาก Functions Palette:
Programming » Waveform ดังนี้



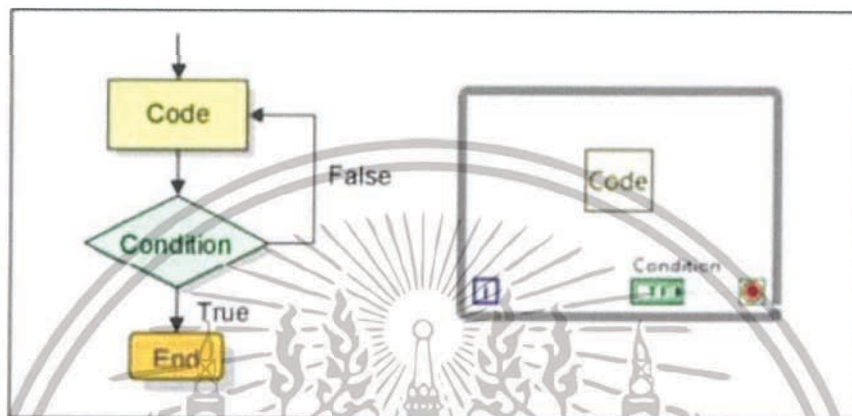
รูปที่ 2.36 ตัวอย่างการสร้าง Waveform ได้ด้วยบล็อก Build Waveform

2.3.4 ลูป (Loop)

1) While loop เมื่อเปรียบเทียบกับ การเขียนโปรแกรมด้วยภาษาอื่น ๆ ก็เปรียบได้กับโครงสร้างของคำสั่ง Repeats until นั่นเองซึ่งลักษณะของ While loop แสดงดังภาพที่ 13 หลักการทำงานของ While loop คือจะทำงานภายในวงรอบไปจนกว่าเงื่อนไข Condition terminal จะเป็นจริง จึงจะหยุดทำงานสำหรับ While loop นั้นจะมีเทอร์มินอลสอง 2 เทอร์มินอลเช่นเดียวกับ For loop คือ Iteration terminal และ Condition terminal การทำงานของ Iteration terminal จะทำงานเหมือนกับ Iteration terminal ของ For loop คือทำหน้าที่แสดงจำนวนรอบที่โปรแกรมกระทำในวงรอบซึ่งแสดงผลออกมาในลักษณะของตัวเลขส่วน Condition terminal เป็นการกำหนดเงื่อนไขให้โปรแกรมหยุดทำงานซึ่งเงื่อนไขในการหยุดทำงานคือจะหยุดทำงาน เมื่อเงื่อนไขเป็นจริง (True) หรือจะให้หยุดทำงานเมื่อเงื่อนไขเป็นเท็จ (False) ก็สามารถที่จะกำหนดได้นอกจากนี้แล้วยังสามารถกำหนดเงื่อนไขในการหยุดทำงานได้อีกหนึ่งวิธีคือการนำเอาค่าเอาต์พุตของ Iteration terminal มาเปรียบเทียบกับค่าที่ต้องการไม่ว่าจะเป็นค่าคงที่หรือปรับเปลี่ยนได้โปรแกรมจะทำงานวนในวงรอบและนำค่าเอาต์พุตของ Iteration terminal มาเปรียบเทียบกับค่าที่กำหนดไว้ แล้วจะส่งเอาต์พุตที่ได้ไปยัง Condition terminal เมื่อจำนวนรอบการทำงานมีค่าเท่ากับค่าที่กำหนดโปรแกรมจะหยุดทำงานในวงรอบนี้ทันทีที่กำหนดเงื่อนไขแบบนี้เปรียบได้กับการกำหนดค่า N ใน For loop นั่นเอง การสร้าง While loop ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function » Structure แล้วใช้เมาส์เลือกที่ While loop นำเมาส์ไปคลิกที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

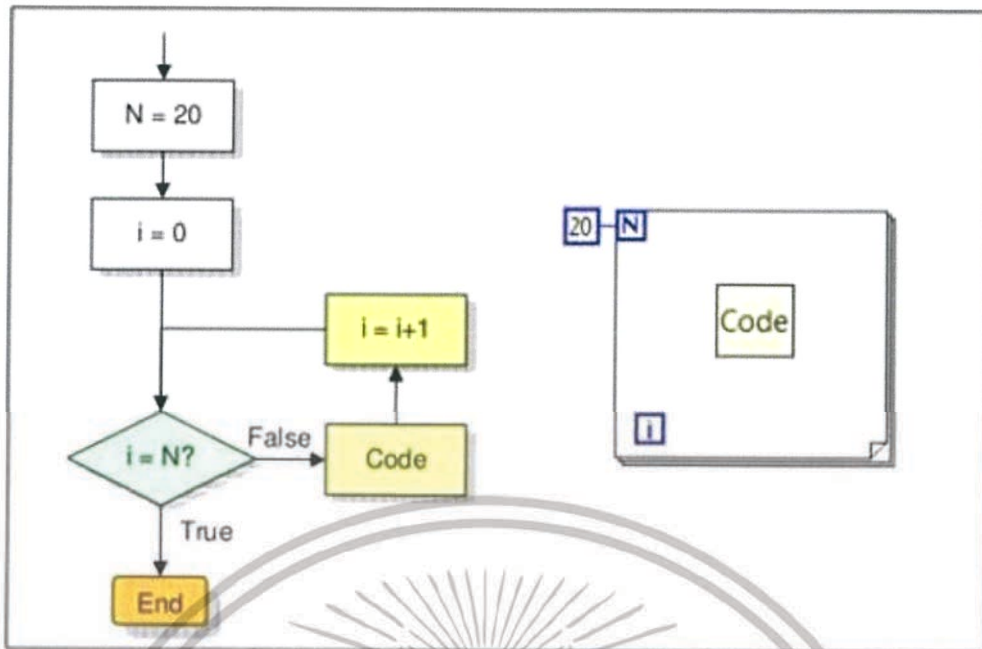
หน้าต่างบล็อกไดอะแกรมแล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ While loop นอกจากนี้แล้วยังสามารถทำการย่อขยายขนาดของ While loop ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ While loop จะปรากฏจุดสี่เหลี่ยมเล็ก ๆ บริเวณเส้นขอบของ While loop ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิกเมาส์ค้างแล้วลากย่อหรือขยายได้ตามต้องการ



รูปที่ 2.37 การสร้าง While loop

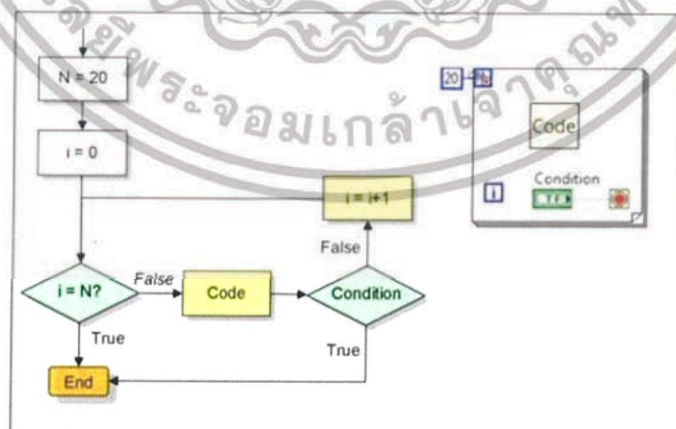
2) For Loop เป็นฟังก์ชันคำสั่งให้โปรแกรมทำงานซ้ำจนกว่าการกระทำซ้ำนั้นจะเท่ากับ จำนวน N เมื่อ N คือจำนวนครั้งที่ต้องการให้ทำซ้ำ ใน For loop นั้นจะมีเทอร์มินอลอยู่ 2 จุดคือ Count terminal และ iteration terminal สำหรับ Count terminal นั้นมีหน้าที่เป็นตัวกำหนดว่าจะให้ โปรแกรมทำงานซ้ำกี่รอบซึ่งสามารถจะกำหนดให้ปรับเปลี่ยนจำนวนรอบหรือเป็นค่าคงที่ก็ได้ ส่วน Iteration terminal จะทำหน้าที่แสดงจำนวนรอบการทำงานว่าโปรแกรมทำงานในวงรอบไป แล้วก็ครั้งตัวเลขที่แสดงของ iteration Terminal จะมีค่าเท่ากับ N-1 การสร้าง For loop ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function » Structure แล้วใช้เมาส์เลือกที่ For loop หลังจากนั้นนำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรม แล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ For loop นอกจากนี้แล้วยังสามารถทำการย่อขยายขนาดของ ' For loop ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ For Loop จะปรากฏจุดสี่เหลี่ยมเล็ก ๆ บริเวณเส้นขอบของ For loop ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิกเมาส์ค้างแล้วลากย่อหรือขยายได้ตามต้องการ การทำงานของ For loop นอกจากจะทำงานเพียงวงรอบเดียว ๆ แล้วยังสามารถทำงานซ้อนวงรอบได้ โดยจะทำงานวงรอบด้านในก่อนจนครบเงื่อนไขแล้วจึงมาทำวงรอบนอกหนึ่งรอบแล้วก็กลับไปทำงานในวงรอบในอีกครั้ง การทำงานจะทำจนกว่าวงรอบนอกครบตามเงื่อนไขโปรแกรมจึงจะหยุดทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.38 การสร้าง For loop

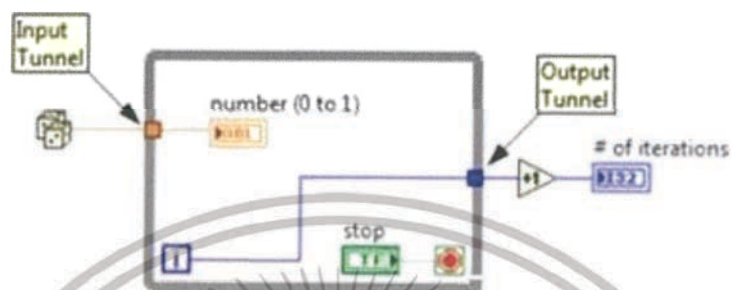
จะเห็นได้ว่าโค้ดใน For Loop มีโอกาสที่จะไม่ถูกรันเลยถ้า N มีค่าเป็นศูนย์ และ For Loop จะมีเงื่อนไขเดียวเท่านั้นที่จะจบได้ คือเมื่อรันครบ N ครั้งแล้วเท่านั้น For Loop ยังสามารถเพิ่มเงื่อนไขการจบให้เหมือนกับ While Loop ได้อีกด้วย โดยการคลิกขวาที่ขอบ ของ For Loop แล้วเลือก Conditional Terminal และสามารถจะให้ For Loop ที่มี เงื่อนไขการจบได้ ซึ่งเรียกรูปแบบนี้ว่า Conditional For Loop ซึ่งมี ประโยชน์สามารถใช้ทดแทน While Loop ได้ในหลายกรณี และช่วยในเรื่องการใช้หน่วยความจำ สรุปลการทำงานของ Conditional For Loop เป็นไฟล์ชาร์ตได้ดังนี้



รูปที่ 2.39 สรุปลการทำงานของ Conditional For Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งค่าเข้า-ออกจากรูปก็เป็นอีกสิ่งหนึ่งที่ต้องทำความเข้าใจ รูปเป็น Structure ซึ่งสามารถมองให้ เป็น Node ใหญ่อันหนึ่งที่สามารถมีได้ทั้ง input และ output, การที่โยงสายเข้าไปในรูปหรือออกจากรูป จะเห็นเกิดเป็นจุดสี่เหลี่ยมตันขึ้นที่ขอบของรูป จุดแบบนี้เรียกว่า Tunnel ซึ่งแสดงให้เห็นว่านี่คือ input หรือ output ของรูป



รูปที่ 2.40 input หรือ output ของรูป

และตามหลักการ Dataflow, รูป (Node) จะไม่สามารถเริ่มได้จนกว่าจะได้รับ input ครบ และรูปจะส่ง output ออกมาจาก Output Tunnel ได้ก็ต่อเมื่อรูปจบการทำงานแล้วเท่านั้น ซึ่งหลักการนี้สำคัญมากในการออกแบบโปรแกรมที่ต้องมีลำดับก่อนหลังในการรัน

2.3.5 การหน่วงเวลา

การสร้างรูปโดยที่ไม่ได้กำหนดการหน่วงเวลาให้ รูปจะรันด้วยความเร็วสูงสุดเท่าที่จะทำได้จน CPU มี การทำงานเต็มที่จนอาจไม่มีเวลาสำหรับงานอื่น ๆ สามารถกำหนดความเร็วของรูปทั้ง For Loop และ While Loop เพื่อให้รูปรันด้วยจังหวะที่สม่ำเสมอและไม่เร็วเกินความจำเป็นได้ด้วยฟังก์ชันหน่วงเวลา ซึ่งมีหลายแบบให้เลือกใช้ อยู่ใน Functions Palette: Programming » Timing

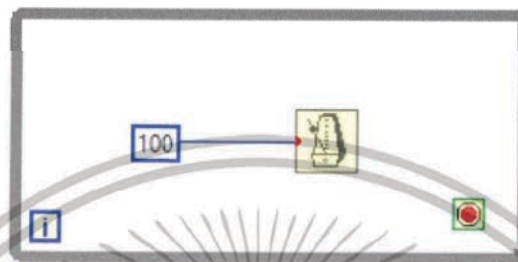


รูปที่ 2.41 แสดงการหน่วงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันหน่วงเวลานั้นใช้มาพิกจากคอมพิวเตอร์ซึ่งมีความละเอียดเป็นมิลลิวินาที (ms) มาอ้างอิง ซึ่ง นำฟังก์ชันนี้วางภายในลูปที่ต้องการจะหน่วงเวลา

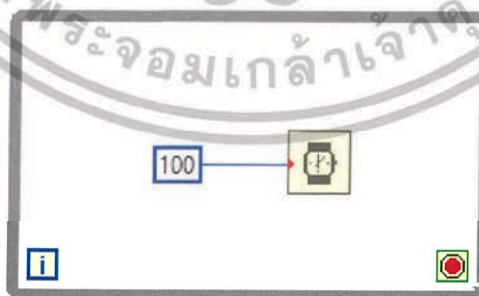
1) Wait Until Next ms Multiple ฟังก์ชันนี้ใช้เพื่อบังคับให้ลูปรันทุก ๆ มิลลิวินาทีที่ใส่ค่าเข้าไป ยกตัวอย่างเช่นใส่ Wait Until Next ms Multiple ให้มีค่าเป็น 100 ในลูปดังนี้



รูปที่ 2.42 ฟังก์ชันหน่วงเวลา Wait Until Next ms Multiple

สมมติว่าก่อนที่จะเข้าลูปจะต้องรันโค้ดที่เรียกว่า pre-loop ก่อนโดยใช้เวลา 3 ms หลังจากนั้นลูปจะ เริ่มต้น โดยโค้ดในลูปจะใช้เวลา 2 ms เมื่อลูปแรกจบ ลูปจะต้องรอโดยไม่มีการตอบสนอง (sleep) เป็นเวลา 95 ms เพื่อที่จะเริ่มลูปต่อไปที่มีลิวินาทีที่ 100 พอดี เพราะฉะนั้นลูปจะเริ่มใหม่ที่มีลิวินาทีที่ 100,200, 300 ไปเรื่อย ๆ

2) Wait (ms) ฟังก์ชันนี้จะหน่วงเวลาเท่ากับค่าที่ใส่ไว้ ยกตัวอย่างถ้าใช้ Wait (ms) ให้มีค่าเป็น 100ในลูปดังนี้



รูปที่ 2.43 ฟังก์ชันหน่วงเวลา Wait (ms)

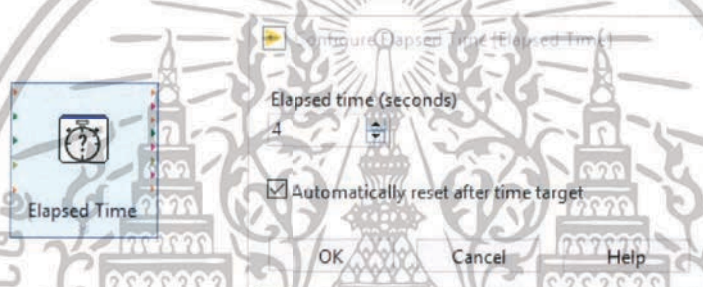
ทันทีที่ลูปแรกรันเสร็จไม่ว่าจะใช้เวลาไปแล้วกี่วินาที ลูปจะต้องรอต่อไปอีก 100 ms ถึงจะเริ่มลูปใหม่ ได้ แบบนี้ไปเรื่อย ๆ ทุกลูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Time Delay เป็น Express VI มีหน้าที่เหมือนกับฟังก์ชัน Wait (ms) ทุกประการ ต่างกันที่ Time Delay มีหน่วยเป็นวินาที

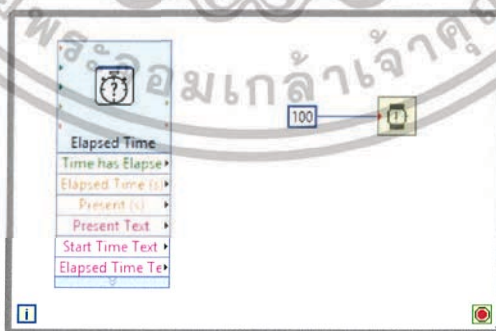
2.3.6 การจับเวลา

ประยุกต์ใช้ฟังก์ชันในกลุ่ม Timing เกี่ยวกับเวลา เช่นถ้าต้องการจับเวลา ซึ่งสามารถใช้ Elapsed Time มาช่วยได้ เมื่อวาง Express VI ตัวนี้ลงไปบน Block Diagram จะเกิดหน้าต่าง Configure Elapsed Time ให้เลือกว่าจะจับเวลานานแค่ไหน และจะรีเซ็ตเริ่มนับศูนย์ใหม่เมื่อครบกำหนดหรือไม่ ยกตัวอย่างว่า เมื่อต้องการให้โปรจับเวลาโดยทุกสองวินาทีให้ส่งสัญญาณออกมาทาง Boolean Indicator หนึ่งครั้ง กำหนดดังนี้



รูปที่ 2.44 การใช้ฟังก์ชัน การจับเวลา

และเมื่อรันโค้ดนี้ จะเห็นไฟ LED กระพริบเมื่อครบทุกๆวินาทีตามกำหนด

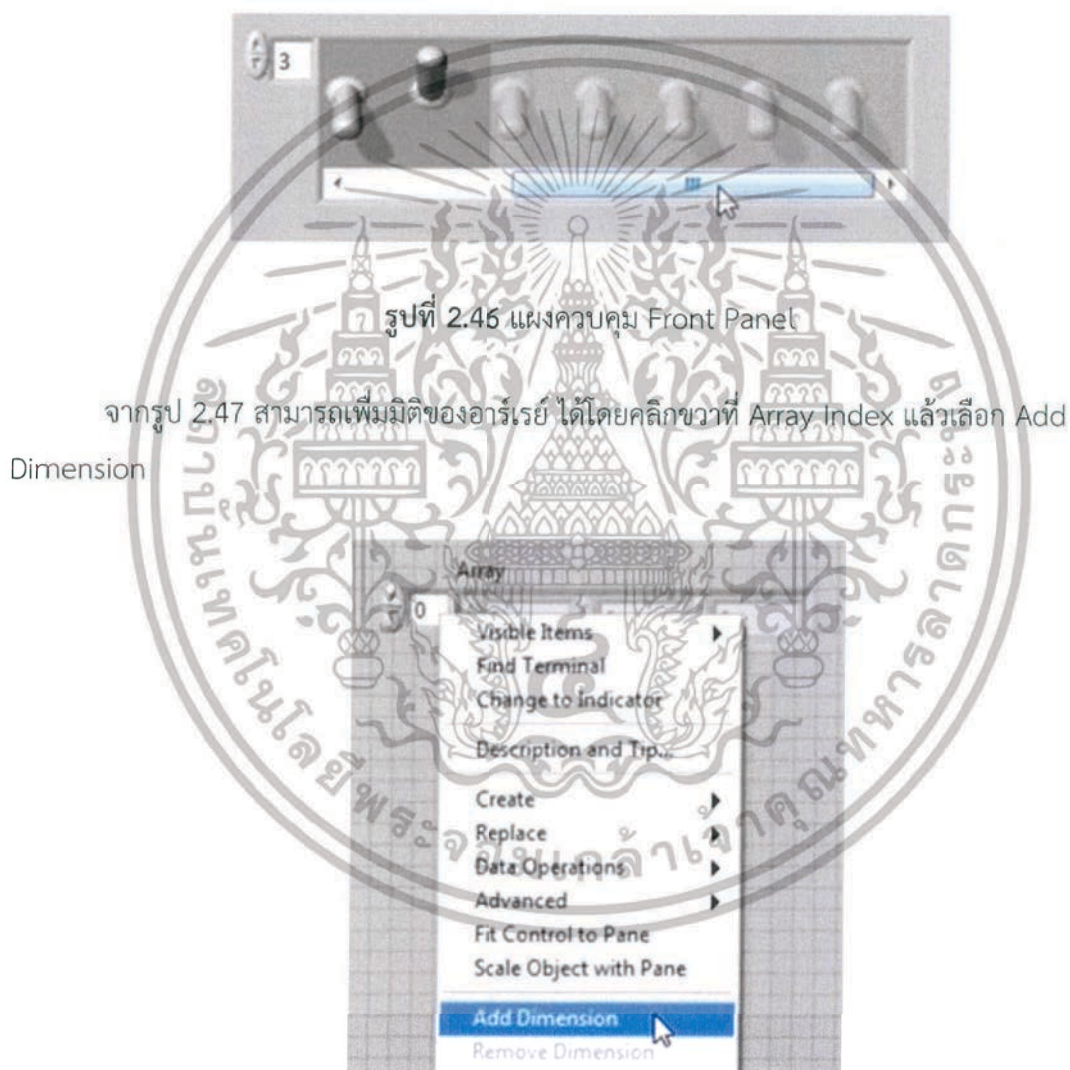


รูปที่ 2.45 ผลของโค้ดจากรูปที่ 2.43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7 ชุดตัวแปรแบบห้องแถว (Array)

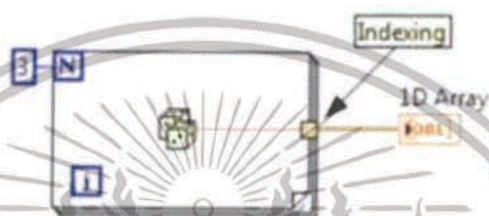
ชุดตัวแปรแบบห้องแถว คือกลุ่มของข้อมูลประเภทเดียวกันนำมาเรียงเป็นแถว เช่น อาร์เรย์ของตัวเลขจำนวนเต็ม หรืออาร์เรย์ของจำนวนจริง เป็นต้น การสร้างอาร์เรย์ Control/Indicator บน Front Panel โดยทำการเลือก Controls Palette แล้ว Modern >> Array Matrix & Cluster >> Array นำมาวางซึ่งจะได้กรอบเปล่า ๆ ของอาร์เรย์ จากนั้นให้นำข้อมูลที่ต้องการไปวางในกรอบอาร์เรย์นั้น เช่น ถ้าต้องการสร้างอาร์เรย์ของ Numeric Control ก็ให้นำ Numeric Control ไปวาง ถ้าลองเปลี่ยนอินเด็กซ์ (Index) จากศูนย์ให้เป็นสาม จะสังเกตว่าส่วนประกอบทุกตัวจะขยับเข้ามา



รูปที่ 2.47 การเพิ่มมิติของอาร์เรย์ (Add Dimension)

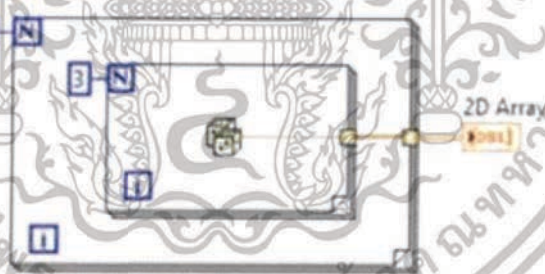
2.3.8 การสร้าง ชุดตัวแปรแบบห้องแถว (Array) ด้วยลูป

โดยสามารถใช้ For Loop ในการสร้างอาร์เรย์ได้ ซึ่งปกติ For Loop ถูกออกแบบมาให้ทำแบบนี้ อยู่แล้ว โดยโยนสายข้อมูล output ออกจาก For Loop สังเกตว่าจะเกิดจุดที่มีการทำ Indexing โดยอัตโนมัติ (ไม่ใช่จุดต้นแบบ Tunnel ธรรมดา) ซึ่งมีการเก็บค่านั้นตั้งแต่ลูปแรกจนถึงสุดท้ายไว้ในอาร์เรย์ แล้วส่งออกมาที่เดียวตอนลูปจบ ตัวอย่างเช่นการสร้างอาร์เรย์ของ Random Number ที่มีขนาด เท่ากับ สาม ทำได้ดังนี้



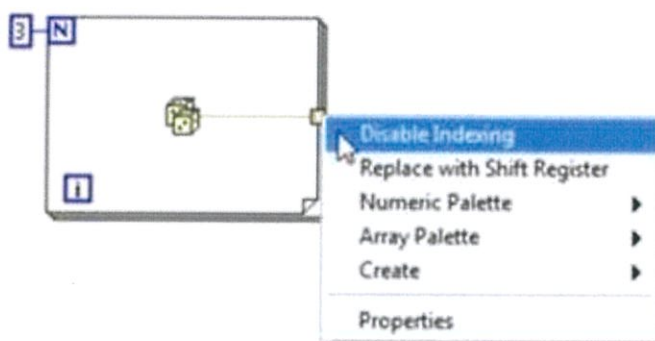
รูปที่ 2.48 For Loop ในการสร้างอาร์เรย์

หรือถ้าต้องการอาร์เรย์ แบบสองมิติที่มีขนาด 5×3 ก็ใส่ลูปเข้าไปอีกชั้น สังเกตที่ความหนาของเส้น ข้อมูลที่จะเป็นตัวบอกมิติของอาร์เรย์



รูปที่ 2.49 การสร้างอาร์เรย์ของ Random Number

แต่ถ้าไม่ต้องการ Indexing ของลูปก็สามารถเอาออกได้โดยคลิกขวาที่ Indexing แล้วเลือก Disable Indexing ซึ่งมันจะกลายเป็น Tunnel แบบปกติ นั่นก็หมายความว่าข้อมูลที่ออกมาจากลูป ผ่าน Tunnel นี้คือข้อมูลของลูปสุดท้ายตัวเดียวเท่านั้น



รูปที่ 2.50 การเลือกให้ข้อมูลออกจากกลุ่มผ่าน Tunnel

ในทางกลับกัน ถ้ามีอาร์เรย์เป็น input ของ For Loop และต้องการให้ลูปนำค่าของอาร์เรย์เข้าไปใน ลูปทีละ Element สามารถใช้ Indexing ได้ดังนี้



รูปที่ 2.51 การให้ลูปนำค่าของอาร์เรย์เข้าไปใน ลูปทีละ Element

2.3.9 กลุ่ม (Cluster)

กลุ่ม (Cluster) คือรูปแบบของข้อมูลที่สามารถมีสมาชิกข้อมูลย่อยหลาย Data type ผสมกันอยู่ภายใน Cluster เดียวกันได้ ซึ่งถือว่ากลุ่มเป็นอีกหนึ่ง Data type ที่สามารถกำหนดโครงสร้างของข้อมูลขึ้นเอง การสร้าง Cluster control/Indicator บน Front Panel นั้นก็คล้ายกับการสร้างอาร์เรย์คือให้เลือก Control Palette แล้ว Modern >> Array Matrix & Cluster>> Cluster เมื่อนำไปวางจะได้กรอบเปล่าของ Cluster จากนั้นก็นำสมาชิก Control / Indicator ที่ต้องการมาวางลงไปเรื่อย ๆ และปรับขยายขนาดได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.10 Error Cluster และการจัดการกับความผิดพลาด (Error)

ใน LabVIEW จะมี Cluster พิเศษแบบหนึ่งเรียกว่า Error Cluster มีไอคอนและสายเป็นสีน้ำตาล เหลือง ซึ่งมีเอาไว้ใช้สำหรับรับส่งสถานะของ SubVI หรือ Node ต่าง ๆ ว่าได้เกิดข้อผิดพลาดมาแล้วหรือไม่อย่างไร สร้าง Error Cluster ได้โดยดึงมาจาก Controls Palette: Modern >> Array, Matrix & Cluster >> Error In ซึ่งสมาชิกใน Error Cluster ประกอบไปด้วย

- Status คือข้อมูลแบบ Boolean จะมีค่าเป็น TRUE หากเกิด error ขึ้น
- Code เป็นตัวเลขแสดงรหัส error ที่ LabVIEW ได้กำหนดไว้
- Source คือตัวอักษรแสดงรายละเอียดเกี่ยวกับ error นั้น

2.3.11 การพล็อตข้อมูล

ถ้าลองสำรวจใน Controls Palette: Modern >> Graph จะพบกับอุปกรณ์พล็อตข้อมูลหลายแบบที่สามารถเลือกนำมาใช้ในการแสดงข้อมูลทั้งแบบสองและสามมิติในทันทีที่จะขออธิบายถึงการพล็อตแบบสองมิติที่ใช้อยู่ที่สุด

Waveform Chart เป็น Indicator แบบพิเศษที่สามารถรับข้อมูลตัวเลขในรูปแบบตัวเลขตัวเดียว (single-point), Array, DDT, Waveform, หรือ Cluster ตัวเลข (Cluster สีน้ำตาล) อย่างไม่อย่างหนึ่งการใช้ Waveform Chart คือการพล็อตโดยที่ยังคงต้องการแสดงข้อมูลชุดเก่า (history) ค้างไว้เพื่อให้สามารถเปรียบเทียบและดูแนวโน้มของข้อมูลจากอดีตจนถึงข้อมูลปัจจุบันได้เมื่อข้อมูลชุดใหม่เข้ามาก็จะพล็อตต่อท้ายไปเรื่อย ๆ ซึ่งจำนวนจุดทั้งหมดที่จะแสดงบนชาร์ตสามารถระบุเองได้

Waveform Graph สำหรับการพล็อตที่ต้องการแสดงเฉพาะข้อมูลชุดปัจจุบันให้ใช้ Waveform Graph ซึ่งจะไม่สามารถแสดงผลข้อมูลที่เป็นลักษณะ single-point ได้ดีนักเพราะจะไม่มี การเก็บลงบัฟเฟอร์จึงเหมาะกับการแสดงผลข้อมูลเวฟฟอร์มที่มาเป็น Array มากกว่ายกตัวอย่างเช่นการแสดงผล Sine wave ที่ถูกสร้างขึ้นมาจากบล็อก Simulate Signal โดยกำหนดให้สร้างรูปคลื่นหนึ่งไซเคิลที่มีความถี่ 1Hz แล้วนำมาพล็อตด้วย Waveform Graph และ Waveform Chart เพื่อเปรียบเทียบความแตกต่าง

2.3.13 โครงสร้างลำดับ (Sequence structure)

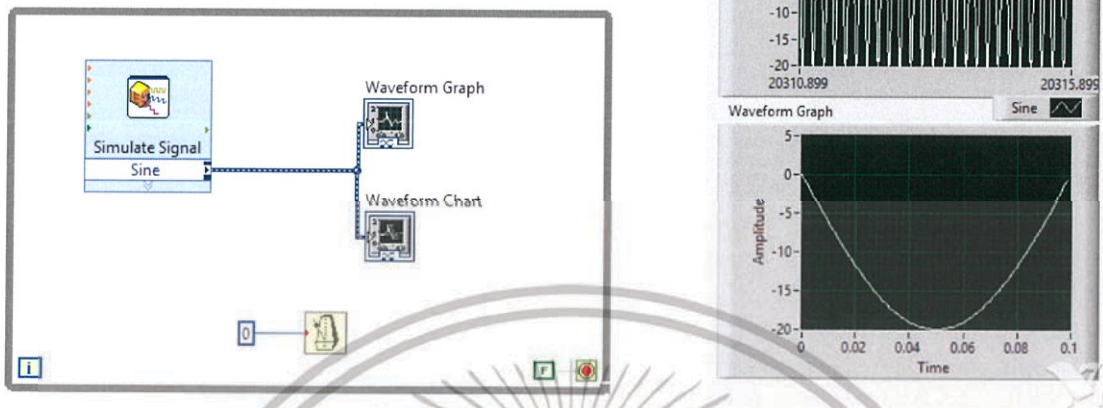
โค้ดใน LabVIEW สามารถรันแบบคู่ขนานตามหลัก Dataflow ดังที่ได้กล่าวไว้แต่ก็มีเทคนิคสามารถบังคับการรันให้เป็นไปตามลำดับที่ต้องการได้เหมือนกัน หากรันโค้ดนี้ส่วนบนที่เกี่ยวกับการเขียนไฟล์ก็จะทำงานตามปกติส่วนฟังก์ชันช่วงเวลาด้านล่างก็ทำงานอิสระเช่นกันซึ่งโค้ดทั้งสองส่วนไม่ได้เกี่ยวข้องกันเลยตอนนี้ถ้าต้องการให้มีการหน่วงเวลาระหว่างการเขียนไฟล์ครั้งที่หนึ่งและครั้งที่สองเนื่องจากฟังก์ชัน Wait ไม่มี Error Cluster ให้ต่อสามารถใช้ Flat Sequence Structure ครอบฟังก์ชัน Wait ไว้แล้วจึงโยนสายเข้าและออกทันทีดังนี้

Sequence Structure จึงเป็นตัวช่วยในการสร้าง Node ให้มี Input / output Tunnel ได้ตามแบบที่ต้องการนอกจากนี้ยังสามารถสร้างเฟรมเพิ่มภายใน Sequence Structure ด้วยการคลิกขวาที่ขอบของมันแล้วเลือก Add Frame ซึ่งสามารถเพิ่มได้ตามต้องการ

2.3.14 ฟอmula โหนด (Formula node)

ในระหว่างการพัฒนาโปรแกรมอาจจะต้องการพื้นที่บางส่วนบน Block Diagram สำหรับเป็นกระดาษทดที่สามารถวางอะไรลงไปก็ได้โดยไม่กระทบกับโปรแกรมสามารถใช้ Diagram Disable Structure โดยเลือกจาก Functions Palette: Programming >> Structures >> Diagram Disable Structure แล้วครอบส่วนของโค้ดที่ต้องการให้แยกออกจากโปรแกรมซึ่งส่วนนั้นไม่จำเป็นต้องถูกเขียนอย่างสมบูรณ์หรือลากสายให้ถูกต้องก็ยังสามารถรันโค้ดส่วนที่เหลือได้อย่างปรกติ Formula Node ในบางครั้งการเขียนฟังก์ชันทางคณิตศาสตร์จำนวนมาก ๆ ด้วยสัญลักษณ์กราฟิกก็ซับซ้อนและอ่านยากแต่ใน LabVIEW สามารถผสมเค็ดตัวหนังสือแบบ ANSI C ใช้ Formula Node ซึ่งมีบิวอินฟังก์ชันเป็นพื้นฐานของสมการทางคณิตศาสตร์ทั่วไป

ตั้งจากวงกรอบ Formula Node ลงใน Block Diagram แล้วให้ใส่ Input, Output Terminal โดยคลิกขวาที่ขอบของ Node แล้วเลือก Add Input ทางกรอบด้านซ้ายแล้วพิมพ์ชื่อตัวแปรเข้าไปได้เลย การวาง output ก็ควรแยกวางไว้ทางกรอบด้านขวาเพื่อความไม่สับสน



รูปที่ 2.52 กราฟแบบ-Waveform Graph และ Waveform Chart

จะเห็นได้ว่า Waveform Graph จะแสดงข้อมูลของรูปปล่าสุดเท่านั้นคือไซเคิลเดียวขณะที่ Waveform Chart จะนำข้อมูลมาเรียงต่อกันยาวไปเรื่อย ๆ ตามแต่จะกำหนด

2.3.12 โครงสร้างกรณี (Case structure)

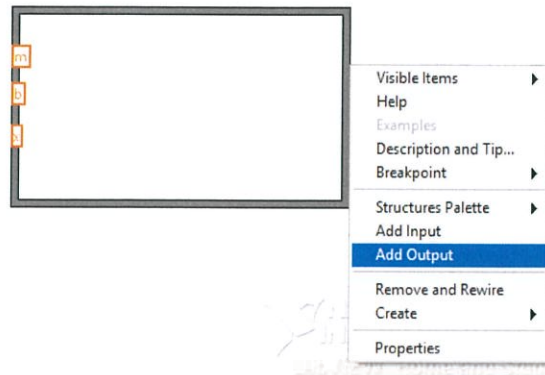
ใช้ในกรณีที่ต้องมีเงื่อนไขการตัดสินใจที่จะเลือกรันโค้ดบางส่วนสามารถเรียก Cast Structure มาจาก Functions Palette: Programming>> Structures, ส่วนประกอบของ Case Structure คือ Diagram ของเคสย่อยอย่างน้อยสองชั้นซ้อนทับกันอยู่ซึ่งจะมองเห็นได้แค่ครั้งละหนึ่งเคสและมีตัวเลือก (Case Selector Terminal) รับค่าว่าจะรันในเคสไหน



รูปที่ 2.53 Case Structure

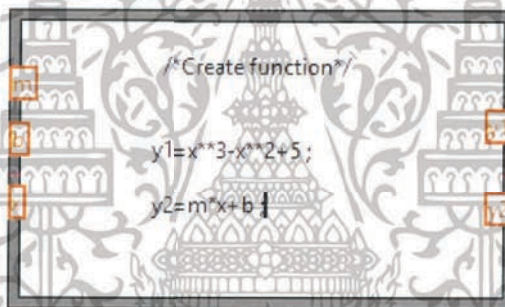
เมื่อสร้าง Case Structure ขึ้นมาใหม่ตัวเลือกเคสเริ่มต้นจะเป็นแบบ Boolean (เครื่องหมายคำถามสี่เหลี่ยม) ซึ่งหมายความว่า จะมีเพียงสองเคสให้ใช้เท่านั้นคือเคส True และเคส False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.54 การจัดวางกรอบ Formula Node

ส่วนประเภทข้อมูลของ input จะเปลี่ยนไปเองตามท่อสายให้ส่วนประเภทของ output จะเปลี่ยนไปตามที่ได้ประกาศไว้ในโค้ดแต่ถ้าไม่ได้ประกาศมันจะเป็นประเภทตัวเลขมีจุดทศนิยม (DBL หรือ float64) ให้เองอัตโนมัติ



รูปที่ 2.55 โค้ดที่ประกาศ

2.3.15 การจัดการกับโปรเจก (Project explorer)

เป็นเครื่องมือช่วยในการจัดการบริหารโปรเจกให้สะดวกยิ่งขึ้นเมื่อได้สร้างโปรเจกขึ้นมาใหม่แล้ว (Empty Project) สามารถ Add ไฟล์เข้าไปในโปรเจกแล้วแบ่งประเภทโดยแยกเป็นโฟลเดอร์ย่อยได้ตามต้องการซึ่งเหมาะกับโปรเจกที่มีไฟล์จำนวนมากไฟล์ทั้งไฟล์ของ LabVIEW เองและไฟล์ประกอบอื่น LabVIEW ซึ่งจะทำให้เห็นภาพรวมว่าทั้งโปรเจกมีไฟล์อะไรบ้างที่เกี่ยวข้องและแต่ละไฟล์มีหน้าที่อะไร เมื่อจะเริ่มสร้างโปรเจกขึ้นมาใหม่ให้คลิกที่ Empty Project ในหน้าแรกหรือให้เลือกเมนู File>> New Project จะได้ไฟล์โปรเจก (.lvproj) เปล่าซึ่งจะมองเห็นเป็น hierarchy ที่เริ่มต้นด้วย My Computer ซึ่งจะเป็นที่อยู่ของ VI, SubVI, และไฟล์ต่าง ๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ซับวีไอ (SubVI)

1) การสร้าง SubVI

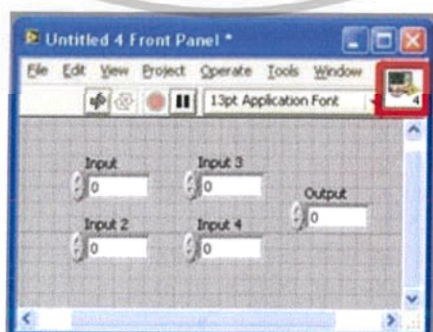
SubVI คือโปรแกรมย่อย หรือโมดูลซึ่งถูกเขียนไว้เพื่อนำมาเรียกใช้ในโปรแกรมหลัก โดยปกติ SubVI จะทำงานอยู่เบื้องหลังการทำงานของโปรแกรมหลัก แต่สามารถปรับแต่ให้สามารถปรากฏตัวอยู่เหนือโปรแกรมหลักได้ (SubVI Front panel) โดย SubVI แบบนี้จะเรียกอีกชื่อว่า “Dialog”

การสร้าง SubVI มีขั้นตอนดังต่อไปนี้ ดังรูป 2.55 ซึ่งแสดง VI ที่ไม่มีการใส่ลูก โดย SubVI มีหน้าคำนวณหาค่าเฉลี่ยจาก ข้อมูลขาเข้า (Input) ซึ่งเป็นตัวเลขสองตัว



รูปที่ 2.56 การสร้าง SubVI

จากรูป 2.56 หน้าต่าง Front Panel เป็นหน้าต่างสำหรับการตรวจสอบผลลัพธ์ของโค้ด โดยทางด้านซ้ายจะจัดเป็นฝั่ง Control และทางด้านขวาคือ Indicator โดยทำการกดที่ช่อง Connector ซึ่งอยู่ตรงมุมขวาด้านบนของหน้าต่าง

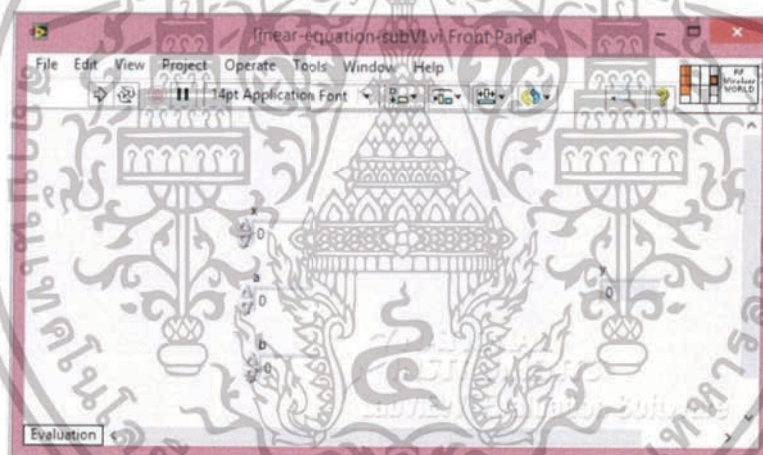


รูปที่ 2.57 หน้าต่าง Front Panel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ช่อง Connector จะมีตารางอยู่หลายช่องเรียกว่า Terminal ให้เลือกต่อสายซึ่งจะมีช่องฝั่งซ้ายขวาอย่างละสี่ช่อง และบนล่างอย่างละสองช่อง ซึ่งทุกๆ ช่องสามารถกำหนดให้เป็นได้ทั้งข้อมูลทางเข้า (Input) หรือข้อมูลขาออก (Output) แต่เพื่อความไม่สับสนควรจะแบ่งให้ฝั่งครึ่งซ้ายเป็นเฉพาะ Input Terminal และทางฝั่งขวาให้เป็นเฉพาะ Output Terminal โดยจะเริ่มกำหนดข้อมูลทางเข้า (Input) ตัวแรกโดยคลิกซ้ายที่ Terminal ช่องใดก็ได้ทางฝั่งซ้าย แล้วคลิกต่อที่บริเวณ Control ตัวแรกเพื่อเป็นการกำหนดความสัมพันธ์ว่า Terminal ช่องนี้เป็นของ control ตัวนี้

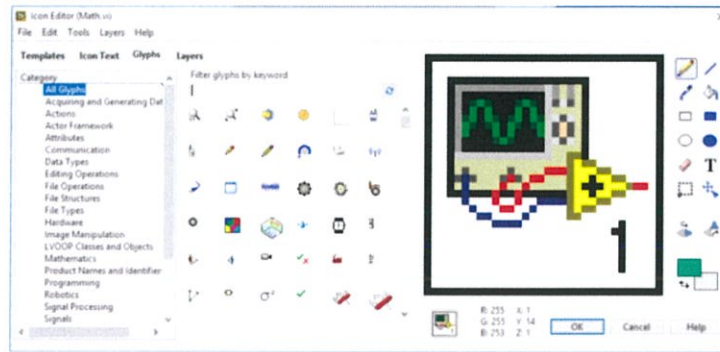
โดยทำแบบเดียวกันกับ Control ตัวต่อไปโดยให้เลือก Terminal ช่องที่ยังว่างอยู่ในขณะเดียวกันก็ทำแบบเดียวกันกับ Indicator ด้วยแต่ให้เลือก Terminal ทางฝั่งขวาจนครบทั้งสามตัวสังเกตว่าสีของ Terminal หลังจากกำหนด Control / Indicator ให้แล้วจะเปลี่ยนไปตามประเภทของข้อมูลของ Control / Indicator นั้น ดังรูป 2.57



รูปที่ 2.58 กำหนด Control / Indicator

ถ้าหาก SubVI มีจำนวนอินพุตและเอาพุต (input/output) มากเกินกว่าจำนวน Terminal ที่มีอยู่ให้ใช้การรวมชุดสายหลาย ๆ เส้นให้เป็นกลุ่มเดียวกันก่อนเพื่อลดจำนวนสายและเมื่อส่งกลุ่มเข้าไปใน SubVI แล้วจึงแตกกลุ่มสายนั้นออกด้วยฟังก์ชัน Unbundle

สำหรับขั้นตอนสุดท้ายคือการแต่งรูปให้กับไอคอนของ SubVI โดยคลิกขวาที่ไอคอนแล้วเลือก it Icon.. ซึ่งจะขึ้นหน้าต่าง Icon Editor ดังรูป 2.58



รูปที่ 2.59 หน้าต่าง Icon Editor

ใน Icon Editor จะมีเครื่องมือที่ช่วยในการแต่งภาพหลากหลาย โดยสามารถวางรูปหรือข้อความในลักษณะเป็นชั้น (layer) เพื่อให้สะดวกในการแก้ไขได้ที่ละหนึ่ง layer และยังมีคลังรูปภาพแบ่งไว้ตามหมวดให้สามารถลากไปใช้งานได้หลายแบบ หลังจากวาดรูปเสร็จแล้วสามารถบันทึก SubVI นี้ได้โดยใช้ชื่อที่สื่อความหมายของหน้าที่ของ SubV นี้ เช่น Average.vi เป็นต้น

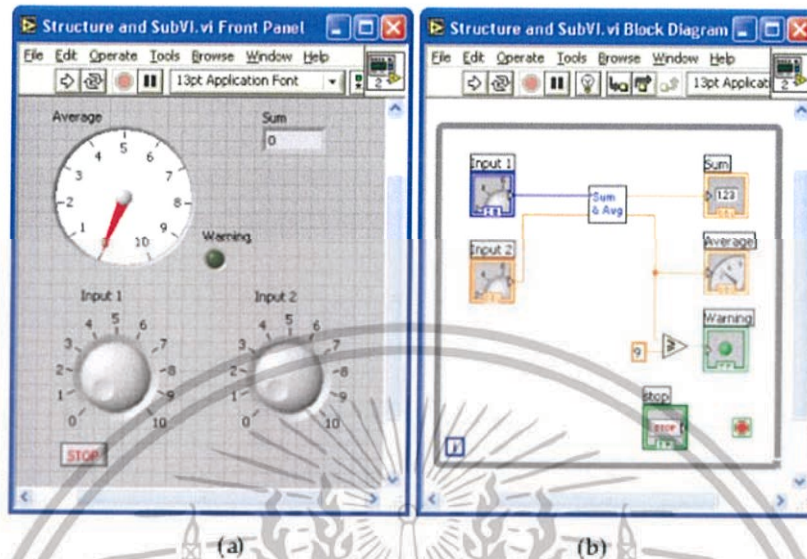


รูปที่ 2.60 ขั้นตอนการใช้งาน SubVI

ส่วนวิธีการเรียก SubVI มาใช้บนโปรแกรมหลักมีหลายวิธี ก่อนอื่นจะต้องสร้าง VI ใหม่ขึ้นมาซึ่งเรียกว่า MainVI เพื่อเป็นตัวเรียก SubVI มารัน จากนั้นลากไอคอนบน Front Panel ของ SubVI ลงไปวางบน Block Diagram ของ MainV1 นอกจากวิธีข้างต้นในการเรียก SubVI มาใช้บนโปรแกรมายังมีอีกวิธี คือการดึง SubVI ตัวไหนก็ได้ที่อยู่ในโปรเจกต์มาวางบน Block Diagram ของ MainVI เมื่อทำการวาง SubVI แล้วสามารถต่อสายใช้งานได้ทันที ซึ่งการสร้าง SubVI ไว้ใช้จะมีประโยชน์ตรงที่จะทำให้โค้ดใน MainVI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สะอาดขึ้น เพราะรายละเอียดและการคำนวณต่าง ๆ จะถูกซ่อนไว้ในกล่อง SubVI และสามารถเรียก SubVI นี้มาใช้ในส่วนต่าง ๆ ของโค้ดได้พร้อมกันหลาย ๆ จุดหรือนำไปใช้ใน VI อื่น ๆ ได้พร้อม ๆ กัน



รูปที่ 2.61 โครงสร้างและ SubVI (Structure and SubVI)

2) การสร้างไดอะล็อก (Dialog)

การสร้าง Dialog สำหรับ SubVI ซึ่งต้องการให้ Front Panel ปรากฏหน้าต่างขึ้นมาอยู่เหนือ MainVI ในขณะที่กำลังทำงานอยู่ การปรับ SubVI ให้เป็น Dialog โดยการเลือกเมนู File >> VI Properties ที่หน้าต่าง VI Properties จะมีให้เลือก Category โดยเลือกเป็น Window Appearance และเปลี่ยนจาก Default เป็น Dialog ซึ่งคุณลักษณะแบบ Dialog จะทำให้หน้าต่างของ SubVI เปลี่ยนไปในขณะรันโปรแกรม คือมองไม่เห็นเมนู แถบเครื่องมือ scrollbar และปุ่มสำหรับย่อขยายหน้าต่าง

ในกรณีที่ต้องการให้หน้าต่าง VI Properties โผล่ตรงกึ่งกลางของจอพอดี ซึ่งสามารถเลือกได้จาก Category เป็น Window Run-Time Position และเปลี่ยน Position จาก Unchanged เป็น Centered และบันทึกไฟล์

การทดลองใช้ Dialog โดยตั้งไปวางบน MainVI แล้วสร้างโค้ด โค้ดดังกล่าวจะทำการรัน dialog ก่อนเมื่อ dialog ตอบโต้กับผู้ใช้เสร็จก็จะส่งเอาพุตออกมาเข้า Case Structure ซึ่งในกรณีที่ output เป็น TRUE ก็จจะรันโค้ดในลูปแต่ถ้าเป็น FALSE ก็จะไม่ทำงานทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เทคนิคการออกแบบ VI

1) โค้ดรูปเดียว

โค้ดรูปเดียว คือรูปแบบการเขียนโค้ดที่มีรูปเดียว ซึ่งเป็นรูปแบบที่ใช้ในโปรแกรมทั่ว ๆ โดยจะมีตั้งแต่โค้ดแบบง่ายไปจนถึงโค้ดที่ซับซ้อน รูปแบบต่าง ๆ ของ Single Loop มีดังนี้

1.1) การเขียนแบบง่าย (Simple VI)

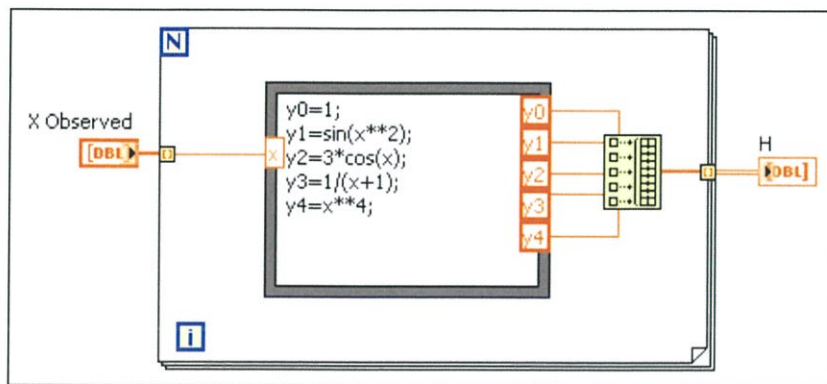
เป็นรูปแบบที่ง่ายที่สุด คือมีการรับค่าจาก source แล้วนำมาประมวลผลและส่งผลลัพธ์ไปที่ซิงค์ (sink) แล้วจบโปรแกรมลงทันทีโดยไม่ต้องมีการรอตอบโต้กับผู้ใช้ รูปแบบนี้เหมาะกับการเขียนโปรแกรมเพื่อการคำนวณสั้น ๆ หรือนำมาใช้เป็น SubVI เพื่อนำมาใช้ในโปรแกรมหลักต่อไป ตัวอย่างของการเขียนโค้ดแบบง่าย แสดงดังรูป 2.61



รูปที่ 2.62 การเขียนโค้ดแบบง่าย

1.2) การเขียนแบบทั่วไป (General VI)

เป็นรูปแบบที่พบบ่อย คือจะมี While Loop สำหรับรันโค้ดซ้ำไปเรื่อย ๆ จนกว่าผู้ใช้จะกดปุ่มให้หยุด โดยก่อนเข้าสู่ลูปอาจมีการรันโค้ดเพื่อการเตรียมค่าเริ่มต้น (Initialize) และพอจบลูปก็สามารถทำการจบการตั้งค่า (clean up) เช่นการปิดไฟล์หรือการถอนตัวออกจากหน่วยความจำเพื่อให้โปรแกรมปิดแหล่งที่ทุกอย่างลงอย่างสมบูรณ์ ตัวอย่างของการเขียนโค้ดแบบทั่วไป แสดงดังรูป 2.62



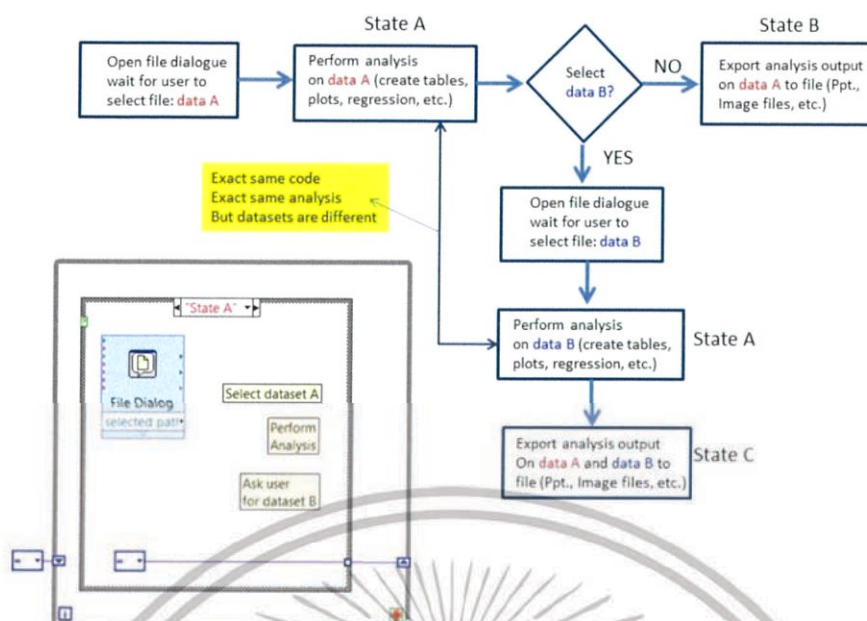
รูปที่ 2.63 การเขียนโค้ดแบบทั่วไป

1.3) Sequential Programming

เป็นการโปรแกรมที่มีลำดับของการรันของโค้ดที่ตายตัว โดยปกติโค้ดจะรันตามหลัก Dataflow อยู่แล้ว แต่ในบางกรณีก็ต้องใช้เครื่องมือช่วย เช่น โปรแกรม ดังรูป 2.69 ซึ่งมีโค้ดอยู่สามอันบน VI โดยโค้ดแต่ละอันถูกแทนด้วย One Button Dialog ซึ่งแต่ละอันไม่มีค่าข้อมูลเข้าและออก (Input/Output) ที่ต่อเข้าถึงกัน เมื่อกดรันจึงทำให้โค้ดทั้งสามรันพร้อม ๆ กัน โดยไม่มีลำดับก่อนหลัง หากต้องการให้โค้ดรันตามลำดับที่ต้องการ จะต้องใช้ Sequence Structure มาช่วย ดังรูป 2.70 การใช้ Sequence Structure มากเกินไปจะทำให้โค้ดมีขนาดใหญ่เกินความจำเป็นและจะอ่านลำบาก แต่สามารถทดแทนได้ด้วยการเขียนแบบ State Machine ดังที่จะกล่าวในหัวข้อถัดไป

1.4) State Machine VI

เป็นโปรแกรมที่มีการทำงานที่ยืดหยุ่นกว่าแบบ Sequential เพราะมีการเพิ่มส่วนของการตัดสินใจในขณะรัน (decision making) เช่นในกรณีที่ต้องการให้โค้ดบางตัวทำงานเฉพาะในบางเงื่อนไขเท่านั้น หรือต้องการเปลี่ยนลำดับการรัน หรืออาจจะหยุดการรันทันที โดยไม่ต้องรอให้จบทั้งลำดับ ซึ่งเราสามารถออกแบบการไหลของโปรแกรมให้ออกมาในรูปแบบ State diagram ซึ่งเป็นแผนภาพการรัน algorithm ที่เข้าใจง่ายและคล้ายกับ flowchart หลังจากนั้นจึงแปลงไปเป็นโค้ด State Machine VI ดังรูป 2.63



รูปที่ 2.64 แสดงการไหลของโปรแกรมให้ออกมาในรูปแบบ State diagram หลังจากนั้นจึงแปลงไปเป็นโค้ด State Machine VI

- 1) โค้ดสลับขานาน
- 2) การออกแบบ Front Panel
- 3) VI Properties
- 4) การจัดระเบียบ Block Diagram

2.6 การส่งข้อมูลระหว่างลูป

การส่งของมูระหว่างลูป สามารถแบ่งออกเป็น 3 รูปแบบด้วยกัน ดังต่อไปนี้

- 1) ตัวแปรเฉพาะที่ (Local Variable)

เป็นตัวแปรที่สามารถใช้ในรับส่งข้อมูลระหว่างลูปภายใน VI เดียวกัน ซึ่งการใช้ตัวแปรเฉพาะที่อาจเป็นวิธีเขียนโปรแกรมที่สะดวก แต่การใช้ Variable มีข้อเสียหลายด้าน เช่น การทำให้โปรแกรมทำงานได้ช้ากว่าการโยนสายธรรมดา และการแก้จุดบกพร่องโปรแกรมทำได้ยาก เพราะไม่มีสายแสดงที่มาของข้อมูล นอกจากนี้ต้องระวังเรื่องเงื่อนไขการแข่งขัน (Race Condition) คือการแย่งกันเข้าไปเขียนตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อม ๆ กัน ซึ่งจะทำให้ผลลัพธ์ของการทำงานของโปรแกรมไม่แน่นอน ดังนั้นหลักการใช้ตัวแปรที่ดีคือใช้เฉพาะในส่วนที่จำเป็น

2) ตัวแปรโดยรวม (Global Variable)

เป็นตัวแปรที่กว้างขึ้น นอกจากจะส่งข้อมูลระหว่างรูปภายใน VI เดียวกัน ยังสามารถส่งข้อมูลระหว่าง VI ได้ เช่นในกรณีที่มี VI ที่กำลังทำงานอยู่ 2 ตัว ในการที่จะควบคุมการหยุดลู่ออกอยู่ที่ VI ตัวแรกเท่านั้น การใช้งานตัวแปรโดยรวม จะสามารถสั่งให้ตัวแรกหยุด โดยที่ตัวที่สองก็จะหยุดตามด้วย

3) ตัวแปรแบบใช้ร่วมกัน (Shared Variable)

เป็นตัวแปรของที่มีความกว้างมากที่สุด คือ สามารถส่งข้อมูลระหว่างคอมพิวเตอร์ โดยสามารถแบ่งออกเป็น ตัวแปรแบบใช้ร่วมกัน โดยการแบ่งผ่านเน็ตเวิร์กสาธารณะ (Network-Published) มีไว้สำหรับการส่งข้อมูลระหว่างคอมพิวเตอร์ผ่านเน็ตเวิร์ค และการส่งข้อมูลแบบกระบวนการเดียว (Single-Process) ซึ่งใช้สำหรับการส่งข้อมูลภายในคอมพิวเตอร์เครื่องเดียวกัน ซึ่งจะเหมือนกับ ตัวแปรโดยรวม (Global Variable) ทุกประการเพียงแต่ การส่งข้อมูลแบบกระบวนการเดียว (Single-Process) จะดีตรงที่ไม่ต้องสร้างเป็นไฟล์ และสามารถเปลี่ยนมาเป็นการส่งแบบเน็ตเวิร์คได้โดยไม่ต้องแก้ไขโค้ด

2.7 การควบคุมหน้าต่างโปรแกรม (User Interface)

1) โหนดคุณสมบัติ (Property Node)

คุณสมบัติของอ็อบเจกต์ (Object) ที่สามารถอ่านค่าหรือเปลี่ยนค่า แทนค่าเก่าลงไปได้ซึ่งแต่ละอ็อบเจกต์ก็จะมีคุณสมบัติหลายรายการ ซึ่งเป็นของตัวเองเช่นค่า ตำแหน่ง, ขนาดบน Front Panel, สี, การกะพริบ เป็นต้น โดยสามารถแก้ไขคุณสมบัติของอ็อบเจกต์ นั้น ๆ ได้โดยการคลิกขวาที่ object แล้วเลือก Properties ซึ่งจะมีหน้าต่างที่สามารถเลือกดู และแก้ไขคุณสมบัติได้หลายอย่างตามต้องการ

2) โหนดการเรียกใช้ (Invoke Node)

อ็อบเจกต์ (Object) แต่ละประเภทจะมีความสามารถที่จะทำงานบางอย่างได้ ซึ่งเรียกว่าขั้นตอน (Method) ใน LabVIEW การสั่งให้อ็อบเจกต์ กระทำขั้นตอนต่าง ๆ จะใช้ Invoke Node เป็นการสั่งการแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าจำนวนรายการขั้นตอนของ อ็อบเจ็กต์จะมีไม่มากเท่ากับจำนวนคุณสมบัติ ยกตัวอย่างขั้นตอนที่ได้ใช้บ่อยๆ เช่น Reinitialize To Default

2.8 การบันทึกและการอ่านไฟล์

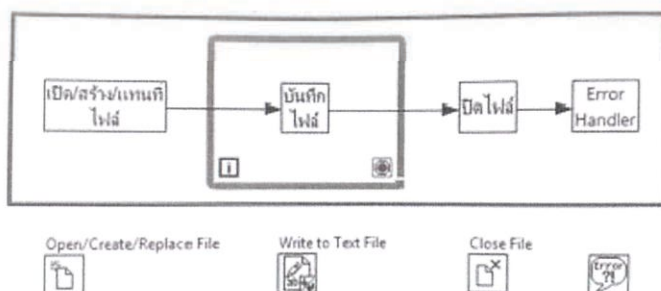
การบันทึกข้อมูลจากใน LabVIEW เก็บลงไฟล์และการอ่านข้อมูลจากไฟล์นำมาใช้ใน LabVIEW โดยการบันทึกข้อมูลเป็นไฟล์มี 2 รูปแบบ ดังต่อไปนี้

- การบันทึกเป็นตัวอักษร (Text) เป็นไฟล์ที่บันทึกเป็นภาษา ASCII มีนามสกุลไฟล์เป็น .txt
- การบันทึกเป็นภาษาคอมพิวเตอร์ (Binary) เหมาะกับการบันทึกหรืออ่านข้อมูลที่มีปริมาณสูงภายในเวลาจำกัด

การเขียนและอ่านไฟล์อย่างง่าย โดย LabVIEW จะมี VI สำเร็จรูปสำหรับการเขียนและอ่านไฟล์ โดยแบ่งออกเป็น 2 รูปแบบ คือ ระดับสูง (High level) ซึ่งเป็นการเขียนเบื้องต้น โดยไม่สนใจรายละเอียดเชิงลึก อีกรูปแบบคือ ระดับต่ำ (Low Level) ซึ่งมีการใช้ฟังก์ชันเกี่ยวกับไฟล์ที่คล่องตัวกว่า โดยมีรูปแบบเริ่มจากการเปิดไฟล์ หรือสร้างไฟล์ใหม่ จากนั้นเริ่มเขียนไปเรื่อยๆ และเมื่อจบก็ทำการปิดไฟล์และตรวจสอบความผิดพลาดที่อาจเกิดขึ้น

1) การบันทึกไฟล์แบบ Low Level

จากที่กล่าวมาเป็นการบันทึกไฟล์แบบเปิดไฟล์หรือสร้างไฟล์ใหม่ แล้วทำการเขียนไปเรื่อยๆ และเมื่อจบก็ทำการปิดไฟล์และตรวจสอบความผิดพลาด ซึ่งการบันทึกแบบนี้เรียกว่า Disk Streaming ซึ่งเป็นการเปิดไฟล์และปิดไฟล์เพียงครั้งเดียว ดังรูปที่แสดงต่อไปนี้



รูปที่ 2.65 การบันทึกไฟล์แบบ Low Level

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับฟังก์ชัน Write To Text File จะรับค่าเป็นแบบสายอักขระ (String) เท่านั้น ดังนั้นถ้าจะเขียนข้อมูลประเภทอื่นเข้าไปจำเป็นต้องแปลงให้เป็นอักขระ ก่อนซึ่งข้อมูลทุกชนิดสามารถแปลงเป็นอักขระได้ เช่นการบันทึกตัวเลขแบบสุ่มที่จะถูกสร้างขึ้นทุก ๆ ครึ่งวินาทีเก็บลงไปเรื่อย ๆ จนกว่าจะกดปุ่มหยุด ซึ่งวิธีการเขียนโปรแกรมเป็นดังนี้

- ใช้ฟังก์ชัน Open/Create/Replace File สิ่งที่ต้องใส่คือ file path และ operation เพื่อที่จะเปิดไฟล์ที่มีอยู่แล้วหรือสร้างไฟล์ใหม่
- ค่าที่ได้ (Output) ของฟังก์ชัน ซึ่งเป็นเส้นสีเขียวเรียกว่า refnum จะเป็นตัวเลขอ้างอิงซึ่งจะโยงไปหาฟังก์ชันอื่นๆ เกี่ยวกับไฟล์ เช่น การเขียน อ่าน ปิดไฟล์ เป็นต้น



รูปที่ 2.66 การบันทึกตัวเลขแบบสุ่ม

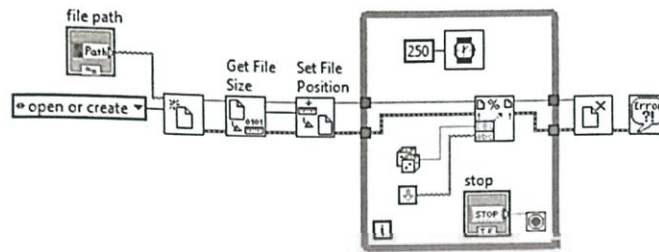
ส่วนการแปลงตัวเลขแบบมีทศนิยมเป็น อักขระ (string) โดยการใช้ฟังก์ชัน Number to Fractional String ซึ่งอยู่ใน Functions Palette : Programming >> string >> String/Number Conversion เมื่อเขียนเสร็จแล้ว ให้ทำการทดลองรันโปรแกรมสักครู่แล้วจึงหยุด เมื่อเปิดไฟล์ที่ถูกสร้างขึ้น จะเห็นเป็นตัวเลขที่บันทึกเรียงเป็นบรรทัดเดียวกัน ในกรณีที่ต้องการให้ตัวเลขในไฟล์ที่ถูกสร้างมา จัดเรียงเป็นบรรทัด สามารถทำได้โดย Function Palette: Programming >> String >> Line Feed Constant โดยทำการแทรกฟังก์ชัน Concatenate Strings ซึ่งเป็นการนำข้อมูลแบบอักขระ หลายๆตัวมาเรียงกัน

2) การบันทึกต่อจากไฟล์ที่มีอยู่แล้วแบบ Low-Level

เป็นการบันทึกข้อมูลต่อจากไฟล์เดิม โดยไม่เป็นการเขียนทับข้อมูลของไฟล์เดิม ซึ่งสามารถทำได้โดยเปลี่ยนตำแหน่งของตัวชี้ตำแหน่ง (Cursor) โดยการใช้ฟังก์ชัน Set File Position (Function Palette: Programming >> File I/O >> Advanced File Functions) หลังจากนั้นใส่ค่า offset เพื่อระบุว่าจะเลื่อน ตัวชี้ตำแหน่งไปวางตรงตำแหน่งไบต์ที่เท่าไรของไฟล์ ในกรณีที่ต้องการให้ตัวชี้ตำแหน่งอยู่ ณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งสุดท้ายของไฟล์ จำเป็นต้องทราบจำนวนไบต์ทั้งหมดของไฟล์ เพื่อเลือกตำแหน่งสุดท้าย โดยการหาจำนวนไบต์ของไฟล์นั้น สามารถใช้ฟังก์ชัน Get File Size โดยสามารถเขียนโค้ดได้ ดังรูป 2.67



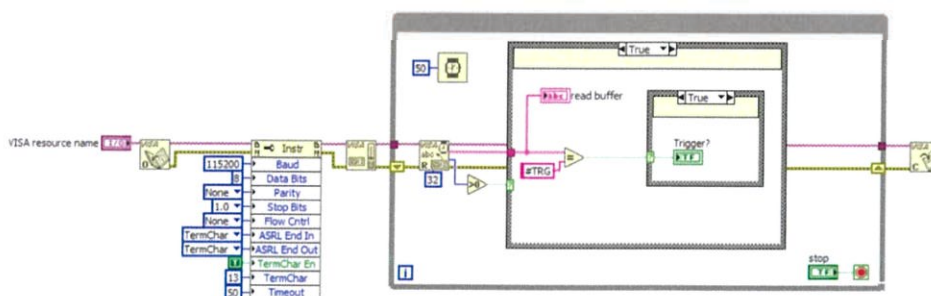
รูปที่ 2.67 แสดงการเขียนโค้ดการใช้ฟังก์ชัน Get File Size

2.9 การเขียนโปรแกรมแบบเหตุการณ์ (Event)

เหตุการณ์ (Event) ในที่นี้คือเหตุการณ์ที่เกิดจากการกระทำของผู้ใช้ผ่านทาง User Interface ที่ระบุไว้ ซึ่งจะทำให้โค้ดถูกรันเท่าที่จำเป็น จึงช่วยลดการทำงานของโปรแกรมและทำให้โปรแกรมมีประสิทธิภาพมากขึ้น

1) การสร้างโปรแกรมด้วยวิธีการหยั่งสัญญาณ (Polling)

เป็นการเขียนโปรแกรมที่ใช้นับจำนวนครั้งที่ผู้ใช้กดสวิตช์นับ (Count) โดยการตั้งค่า Mechanical Action แบบ Latch When Pressed โดยเมื่อลองเดินหน้าโปรแกรมจะพบว่า ลูปรันด้วยความเร็วสูงมาก และการทำงานของคอมพิวเตอร์ก็จะสูงตามไปด้วย เนื่องจากไม่ได้ใส่ตัวหน่วงเวลาไว้ในลูบ ดังนั้นควรที่จะเพิ่มฟังก์ชัน Wait (ms) เพื่อไม่ให้โปรแกรมใช้งานความจำคอมพิวเตอร์มากเกินไป โดยข้อเสียของการใส่ฟังก์ชันหน่วงเวลาคือ กรณีที่กดปุ่มนับเร็วกว่าเวลาที่ตั้งเอาไว้ในตัวหน่วงเวลา โปรแกรมจะจับการกดไม่ทันทั้งหมดทุกครั้ง แม้ในกรณีที่ไม่ได้กดปุ่มนับ ลูบก็ยังคงรันอย่างต่อเนื่องอยู่ตลอดเวลา จึงเป็นที่มาของชื่อวิธีการเขียนโปรแกรมแบบการหยั่งสัญญาณ (Polling)



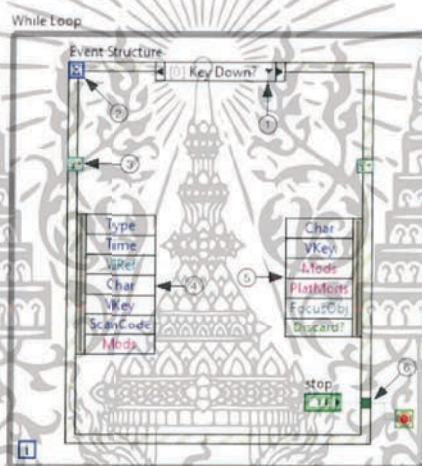
รูปที่ 2.68 การสร้างโปรแกรมด้วยวิธีการหยั่งสัญญาณ (Polling)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) โครงสร้างของเหตุการณ์ (Event structure)

โครงสร้างของเหตุการณ์ สามารถเขียนได้โดยการนำการเขียนโปรแกรมแบบหยั่งสัญญาณ (Event) มาทำการลบโหนด (Node) และสายภายในลูบออก โดยเหลือ Shift Register โดยเริ่มต้นเหตุการณ์ จะมีชื่อว่า Time out ซึ่งโครงสร้างของเหตุการณ์ (Event structure) จะเหมือนกับโครงสร้างของกรณี (Case structure) ตรงที่มีโค้ดสำหรับแต่ละเหตุการณ์วางซ้อนทับกันอยู่ ในกรณีที่ต้องมีการกดปุ่มนับ สามารถทำได้โดยการเพิ่มค่าใน Shift Register ที่ละหนึ่ง วิธีการคือ

- คลิกขวาที่ช่องของ Event Structure
- เลือก Edit Events Handled by This Case แล้วจะขึ้นหน้าต่าง Edit Event ดังรูป 2.69



รูปที่ 2.69 การเขียนโปรแกรมแบบหยั่งสัญญาณ (Event) มาทำการลบโหนด (Node)

- ในช่อง Event Sources ให้กดเลือก Controls >> Count (โดยสังเกตว่าจะมีรายชื่อของ Control/Indicator ทุกตัวในขณะนั้น)
- ในช่อง Event ทางขวามือ ให้เลือก Value Change แล้วคลิก OK
- จากนั้นจะได้เหตุการณ์ (Event) เคสหมายเลข 0 ชื่อว่า "Count": Value Change
- เขียนโค้ดโดยการเพิ่มค่า (Increment หรือบวกหนึ่ง) ของ Shift Register เมื่อค่าของสวิตช์ นับ มีการเปลี่ยนแปลง
- เพิ่มเหตุการณ์ (Event) อีกหนึ่งเคสสำหรับกรณีที่ปุ่มหยุด (Stop) ถูกกด โดยจะต้องส่งค่า TRUE ไปยังเงื่อนไขของลูบ ซึ่งทำโดยการ คลิกขวาที่ช่องของ Event Structure เลือก Add Event Case

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่มเหตุการณ์หมายเลข 1 แล้วเลือก control >> Stop
- เลือก Value Change นอกจากนี้สามารถเพิ่มเหตุการณ์ในเคสเดียวกันได้ โดยการกดปุ่ม Add Event

เมื่อทดลองรันโปรแกรมนี้ จะเห็นความแตกต่างจากโปรแกรมแบบเก่าตรงที่ ลูปจะไม่ทำงานตลอดเวลา แต่จะทำงานก็ต่อเมื่อเกิดเหตุการณ์เท่านั้น แต่ถ้ามีคำสั่งอื่นที่ไม่ได้ระบุใน โครงสร้างเหตุการณ์ (Event structure) ลูปจะไม่ทำงานและไม่ตอบสนองต่อคำสั่ง

2.10 การเก็บข้อมูล Data Acquisition

การเก็บข้อมูล (Data Acquisition) หรือ DAQ เป็นการเก็บรวบรวมข้อมูลโดยการอ่านค่าสัญญาณทางไฟฟ้า จากตัวรับรู้ข้อมูล (Sensor) ซึ่งทำหน้าที่แปลงสัญญาณทางกายภาพให้เป็นปริมาณทางไฟฟ้า แล้วแปลงสัญญาณให้อยู่ในรูปของสัญญาณดิจิทัล ซึ่งจะเก็บไว้ในหน่วยความจำคอมพิวเตอร์ เพื่อการวิเคราะห์ จัดเก็บ หรือแสดงผลต่อไป ขึ้นอยู่กับวัตถุประสงค์ในการใช้งานของแอปพลิเคชัน ระบบ DAQ ประกอบด้วยส่วนประกอบหลัก 4 ส่วน คือ

- 1) แหล่งสัญญาณ (Signals)
- 2) ส่วนปรับสภาพสัญญาณ (Signal Conditioning)
- 3) อุปกรณ์ Data Acquisition (DAQ)
- 4) ซอฟต์แวร์ประมวลผลข้อมูลบนคอมพิวเตอร์ (Application software)



รูปที่ 2.70 แสดงองค์ประกอบการเก็บข้อมูล (Data Acquisition หรือ DAQ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่วัดจากตัวรับข้อมูล อาจจะเป็นสัญญาณแอนะล็อกหรือสัญญาณดิจิทัล โดยสัญญาณเหล่านี้จะถูกส่งต่อไปยังส่วนปรับสภาพสัญญาณ หรือที่เรียกว่า Signal conditioning ซึ่งจะทำหน้าที่ในการปรับปรุงสัญญาณให้เหมาะสมก่อน ซึ่งจะใช้ในกรณีที่สัญญาณที่รับเข้ามามีขนาดใหญ่หรือเล็กเกินไป แต่ถ้าสัญญาณมีลักษณะเหมาะสมแล้ว ไม่จำเป็นต้องนำส่วนปรับสภาพสัญญาณมาใช้ เมื่อสัญญาณผ่านส่วนปรับสภาพสัญญาณเรียบร้อยแล้ว จะถูกส่งต่อไปยังอุปกรณ์ DAQ สำหรับอุปกรณ์ DAQ แบบพื้นฐานจะสามารถวัดสัญญาณแรงดันไฟฟ้าอยู่ในช่วง $\pm 10V$ เท่านั้น ซึ่งอุปกรณ์นี้จะถูกติดตั้งบนคอมพิวเตอร์ที่มีไดรฟ์เวอร์และมีแอปพลิเคชันที่เขียนด้วยซอฟต์แวร์สำหรับพัฒนาระบบเช่น LabVIEW

2.10.1 องค์ประกอบระบบการเก็บข้อมูล

1) แหล่งสัญญาณ (Signal)

การตรวจวัดสัญญาณทางกายภาพ (Transducer / Sensor / Detector) ในที่นี้จะใช้ตัวตรวจจับ (Detector) ตัวรับรู้ (Sensor) หรือตัวแปลงปริมาณ (Transducer) เพื่อแปลงปริมาณทางกายภาพให้เป็นสัญญาณทางไฟฟ้า

2) ส่วนปรับสภาพสัญญาณ (Signal Conditioning)

ส่วนนี้เป็นการปรับปรุงคุณภาพของสัญญาณไฟฟ้าที่มาจากแหล่งสัญญาณ หน้าที่เด่นของส่วนนี้คือการขยายขนาดหรือลดทอนสัญญาณ นอกจากนี้วงจรปรับสภาพสัญญาณยังใช้ในการแปลงสัญญาณไฟฟ้าที่ไม่ได้อยู่ในรูปของความต่างศักย์ไฟฟ้า ให้มาอยู่ในรูปของความต่างศักย์ไฟฟ้า เพื่อทำการแปลงสัญญาณนั้นให้เป็นสัญญาณดิจิทัล

3) อุปกรณ์ Data Acquisition

ส่วนควบคุมการเก็บข้อมูลและการเชื่อมต่อกับคอมพิวเตอร์ (Interface board) มีอยู่ 2 แบบคือแบบการ์ดเสียบและแบบเชื่อมต่อภายนอก ในส่วนนี้อาจกำหนดให้ทำหน้าที่อย่างเดียวหรือมากกว่าเพื่อรับค่าสัญญาณแอนะล็อกอินพุต ส่งค่าสัญญาณแอนะล็อกเอาต์พุตรับค่าสัญญาณดิจิทัลอินพุต ส่งค่าสัญญาณดิจิทัลเอาต์พุตและทำการนับจำนวนหรือจับเวลา

4) ซอฟต์แวร์ประมวลผลข้อมูลบนคอมพิวเตอร์

ส่วนนี้จะขึ้นกับการพัฒนาซอฟต์แวร์เพื่อใช้ในจุดประสงค์ใดโดยในการพัฒนาซอฟต์แวร์จะ

พิจารณาข้อจำกัด 3 ประการคือ

- ระบบปฏิบัติการที่ใช้ในเครื่องคอมพิวเตอร์ว่าเอื้อความสะดวกในการพัฒนาซอฟต์แวร์มากน้อยเพียงใด
- ความเชี่ยวชาญในการเขียนโปรแกรมและการใช้ซอฟต์แวร์
- ความสามารถเข้ากันได้ระหว่างซอฟต์แวร์และฮาร์ดแวร์

2.10.2 ลักษณะสัญญาณทางไฟฟ้าที่ใช้กับอุปกรณ์ Data Acquisition

ลักษณะสัญญาณในการเก็บข้อมูลของอุปกรณ์ Data Acquisition เปรียบระบบการเก็บข้อมูลเป็นเหมือนโมเดลทางคณิตศาสตร์ ที่เมื่อทำการป้อนข้อมูล (Input) ใดๆส่งเข้ามาที่โมเดลนี้ จะได้ข้อมูล (Output) ออกมา แบ่งเป็น 5 ลักษณะด้วยกัน คือ ค่าสถานะดิจิทัล (State), อัตราส่วน (Rate), ระดับสัญญาณ (Level), รูปร่าง (Shape) และความถี่ (Frequency) แสดงดังรูป

แหล่งกำเนิดสัญญาณประเภทไฟฟ้า หรือเรียกว่า Voltage source ซึ่งเป็นแหล่งสัญญาณที่พร้อมที่จะถูกวัดด้วยอุปกรณ์ Data Acquisition ซึ่งแบ่งออกเป็นสองชนิด ได้แก่

1) แหล่งสัญญาณแบบต่อกราวด์ (Grounded source)

สัญญาณที่ขาลบถูกอ้างอิงกับระบบกราวด์ เช่น กราวด์ของอาคารโดยปกติศักย์ทางไฟฟ้าของกราวด์ในแต่ละระบบจะมีค่าไม่เท่ากัน การต่อกราวด์ของสัญญาณเข้ากับกราวด์ของเครื่องมือวัดอาจจะทำให้เกิดกระแสรั่วระหว่างสองกราวด์ หรือที่เรียกว่า กราวด์ลูป (Ground loop) ซึ่งจะส่งผลทำให้ค่าที่วัดคลาดเคลื่อนได้

2) แหล่งสัญญาณแบบไม่ต่อกราวด์หรือแบบลอย (Floating source)

สัญญาณที่ไม่ได้ถูกอ้างอิงกับระบบกราวด์ใด ๆ ยกตัวอย่างเช่นเทอร์โมคัปเปิ้ล, แบตเตอรี่, สัญญาณขาออกจาก isolation amplifier เป็นต้น

2.10.3 ส่วนปรับสภาพสัญญาณ (Signal Conditioning)

ส่วนปรับสภาพสัญญาณ (Signal conditioning) มีหน้าที่ปรับสัญญาณให้เหมาะสมก่อนการวัดด้วยอุปกรณ์ Data Acquisition เพื่อให้ได้ผลการวัดที่เที่ยงตรงและปลอดภัยมากขึ้น ซึ่งตัวอย่างของส่วนปรับสภาพสัญญาณ ที่พบได้ทั่วไปมีดังนี้

- 1) Amplification คือการขยายสัญญาณที่เล็กเกินไปให้มีขนาดใหญ่ขึ้น ทำให้สามารถวัดการเปลี่ยนแปลงของสัญญาณได้ละเอียดมากขึ้น
- 2) Excitation คือการจ่ายไฟกระตุ้นให้กับตัวรับรู้ (Sensor) บางชนิด เช่น RTD เป็นต้น โดยจะทำการจ่ายไฟเลี้ยงให้ในรูปแบบและขนาดที่เหมาะสม
- 3) Bridge Configuration คือการต่อตัวต้านทานเพิ่มให้ครบสี่ตัวเป็นวงจรแบบบริดจ์ซึ่งใช้กับการวัดด้วยสเตรนเกจ (strain gauge)
- 4) Filtering คือการกรองสัญญาณในย่านความถี่ที่ไม่ต้องการออกจากระบบเพื่อให้สัญญาณมีคุณภาพดีขึ้น
- 5) Isolation คือการแยกตัวนำไฟฟ้าของระบบวัดสัญญาณสองส่วนออกจากกันในกรณีที่มีทรานเซียนขนาดใหญ่วิ่งเข้ามาในระบบ นอกจากนี้ยังสามารถแก้ปัญหาเรื่องกราวด์ลูป (Ground loop) ได้อีกด้วย

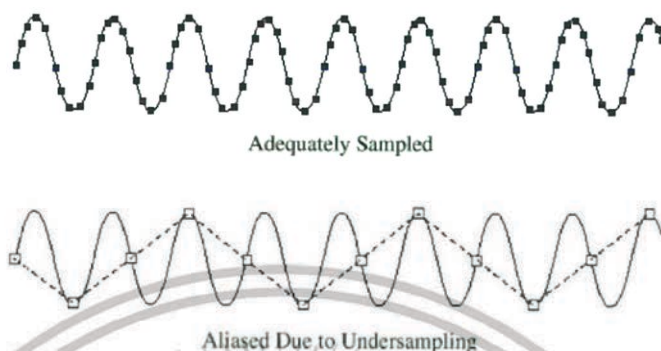
2.10.4 ขั้นตอนการแปลงสัญญาณจากอะนาล็อกเป็นดิจิทัล

สำหรับอุปกรณ์ Data Acquisition (DAQ) ทั่วไปสามารถวัดสัญญาณอะนาล็อกได้ โดยผ่านทางช่อง Analog input การนำสัญญาณอะนาล็อกเข้าสู่ระบบคอมพิวเตอร์ซึ่งเป็นฟอร์มแมตแบบดิจิทัลจำเป็นต้องผ่านอุปกรณ์การแปลงสัญญาณที่เรียกว่า การแปลงอะนาล็อกเป็นดิจิทัล (Analog to Digital converter หรือ ADC) ซึ่งเป็นกระบวนการในการนำค่าทางดิจิทัล มาเลียนแบบค่าทางอะนาล็อก ซึ่งขั้นตอนในการปรับแต่งค่าของอุปกรณ์ดังกล่าว มีดังนี้

1) การสุ่มสัญญาณ (Sampling)

กระบวนการนี้จึงเป็นการเปลี่ยนสัญญาณอะนาล็อกที่เป็นเส้นกราฟต่อเนื่องให้กลายเป็นจุดของข้อมูลดิจิทัลที่ไม่ต่อเนื่อง ด้วยวิธีการสุ่มสัญญาณหรือที่เรียกว่า Sampling ด้วยอัตราเร็วที่คงที่ โดยความ

ละเอียดของกราฟจะขึ้นอยู่กับค่า อัตราการสุ่มสัญญาณ (Sampling rate) ยิ่งมากยิ่งเก็บความละเอียดของเส้นกราฟได้ใกล้เคียงต้นฉบับ

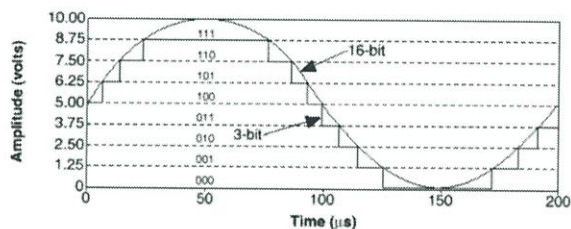


รูปที่ 2.71 แสดงขั้นตอนการวิธีการสุ่มสัญญาณ

การกำหนด อัตราการสุ่มสัญญาณในการแปลงสัญญาณที่เหมาะสม โดยยึดตามหลักการแปลงสัญญาณที่เรียกว่าทฤษฎี Nyquist โดยทฤษฎี กล่าวไว้ว่า สัญญาณจะต้องถูกสุ่มด้วยความเร็วอย่างน้อยสองเท่าของความถี่สูงสุดที่มีอยู่ในสัญญาณ เพื่อให้สามารถนำจุดข้อมูลเหล่านั้นมาสร้างเป็นเวฟฟอร์มกลับมาให้เหมือนกับสัญญาณต้นฉบับได้

2) การกำหนดความคมชัด (Resolution)

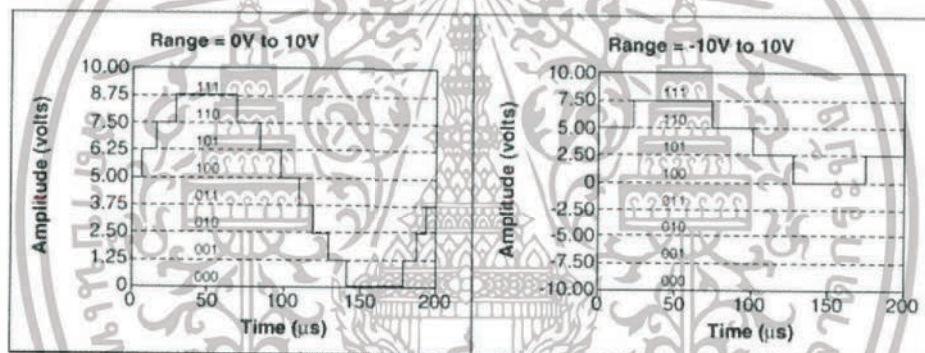
ขั้นตอนการสุ่มสัญญาณ (Sampling) คือการสุ่มสัญญาณทางแกนเวลา แต่สำหรับเป็นค่าของสัญญาณจะต้องทำการสุ่มเช่นกัน เพื่อเปลี่ยนให้เป็นค่าทางดิจิทัล ด้วยวิธีการแบ่งนับ (Quantization หรือ Digitization) โดยการทำการแบ่งนับขึ้นอยู่กับความละเอียดของ การแปลงอะนาล็อกเป็นดิจิทัล (Analog to Digital converter หรือ ADC) ถ้ายังมีความละเอียดมากก็ยังสามารถตรวจจับการเปลี่ยนแปลงของระดับสัญญาณอะนาล็อกได้ละเอียดยิ่งขึ้น ซึ่งความละเอียดของ ADC จะนับกันเป็นจำนวนบิต สำหรับอุปกรณ์ DAQ จะมีให้เลือกตั้งแต่ 12 บิต ไปจนถึง 24 บิต



รูปที่ 2.72 แสดงขั้นตอนการ Resolution

3) การกำหนดความกว้าง (Range)

การกำหนดความกว้างในขั้นตอนการแบ่งนับ ทำให้การวัดมีความเหมาะสม เพื่อที่จะได้ความละเอียดสูงสุดของ ADC ที่มีอยู่ โดยอุปกรณ์ DAQ จะอนุญาตค่า Range ได้จำกัดเฉพาะเพียงช่วงหนึ่ง และบางอุปกรณ์ไม่สามารถตั้งค่า Range อื่นได้ โดยค่า Range มาตรฐานคือ (± 10)



รูปที่ 2.73 แสดงขั้นตอนการกำหนดความกว้าง

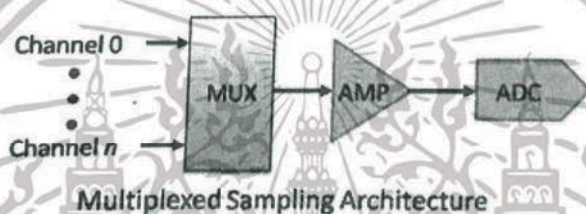
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.5 แซมปลิงอาร์คิเทคเจอร์ (Sampling Architecture)

อุปกรณ์ DAQ ทั่วไปจะมีช่องสำหรับ analog input จำนวนหลายช่องเพื่อให้สามารถวัดสัญญาณอะนาล็อกได้หลาย ๆ ช่องพร้อมกัน ซึ่งในทางการออกแบบฮาร์ดแวร์หรือ architecture แบ่งออกเป็น 2 วิธี ได้แก่

1) แซมปลิงอาร์คิเทคเจอร์แบบหลายทาง (Multiplexed Sampling Architecture)

เป็นการใช้ multiplexer (MUX) เข้ามาช่วยในการสวิตซ์สัญญาณที่ละช่องเข้ามาใน instrumentation amplifier (AMP) ซึ่งเป็นการสวิตซ์แบบกลับไปมาซึ่งรวดเร็วมาก ก่อนจะเข้าไปถึง ADC ดังภาพ

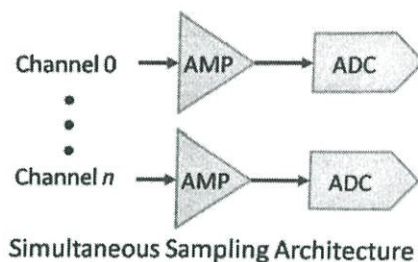


รูปที่ 2.74 แสดงหลักการของแซมปลิงอาร์คิเทคเจอร์แบบหลายทาง (Multiplexed Sampling Architecture)

ข้อดีของอุปกรณ์แบบนี้คือมีต้นทุนที่ต่ำกว่าเพราะใช้ ADC เพียงตัวเดียวก็สามารถวัดสัญญาณหลาย ๆ ช่องได้ในคราวเดียวกัน ข้อเสียคือ Sampling rate สูงสุดที่สามารถใช้ได้จะต้องถูกหารด้วยจำนวนช่องที่ใช้ นอกจากนี้คือเกิดการเกิด phase shift error เนื่องจากเป็นการวัดสัญญาณด้วยการสแกนสัญญาณเข้า ADC ที่ละช่อง ไม่ได้สุ่มสัญญาณพร้อมกันจริง ๆ จึงทำให้สัญญาณที่อ่านได้แต่ละช่องมีเฟสคลาดเคลื่อนออกไป ซึ่งจะเห็นชัดเจนมากขึ้นในกรณีที่สัญญาณมีความถี่สูง

2) แซมปลิงอาร์คิเทคเจอร์แบบพร้อมกัน (Simultaneous Sampling Architecture)

วิธีการนี้จะติดตั้ง ADC ประจำอยู่ในแต่ละช่องสัญญาณเพื่อให้สามารถใช้ sampling rate ได้เต็มที่และยังสามารถสุ่มสัญญาณได้พร้อมกันอย่างแท้จริง เพื่อรักษาความสัมพันธ์ทางเฟสของสัญญาณในแต่ละช่อง ซึ่งจะเหมาะกับการวิเคราะห์เสียงและความสั่นสะเทือน, แรงเครียดแบบไดนามิกสปีโนโครงสร้าง, การวิเคราะห์กำลังไฟฟ้า เป็นต้น

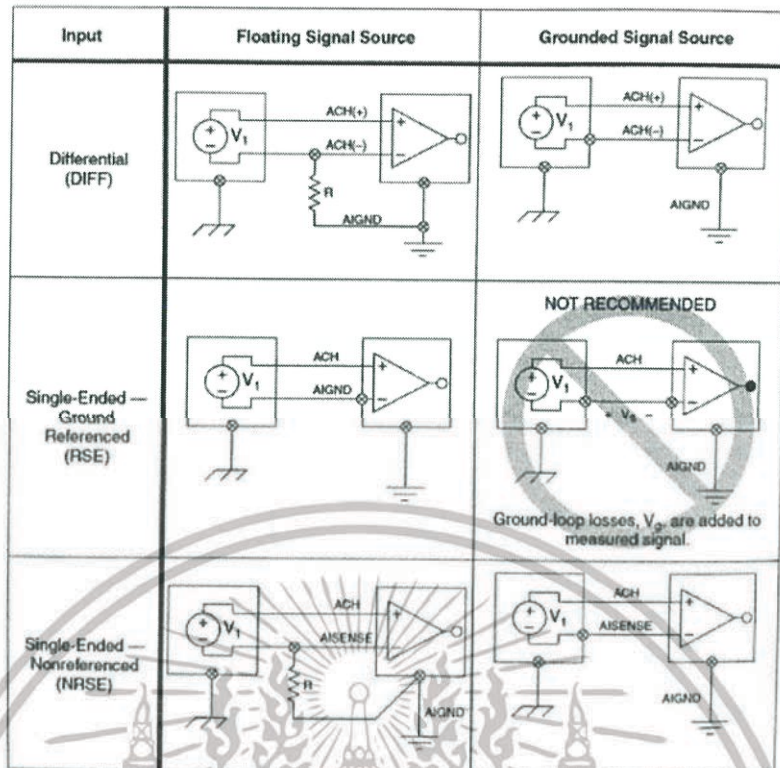


รูปที่ 2.75 แสดงหลักการของแซมปลิงอาร์คิเทคเจอร์แบบพร้อมกัน
(Simultaneous Sampling Architecture)

2.10.6 ประเภทของระบบการวัดสัญญาณ

เนื่องจากแหล่งสัญญาณซึ่งแบ่งออกเป็น 2 ประเภทที่ได้กล่าวมาข้างต้น คือ แหล่งสัญญาณแบบต่อกราวด์ (Grounded source) และ แหล่งสัญญาณแบบไม่ต่อกราวด์หรือแบบลอย (Floating Source) ซึ่งแต่ละประเภทมีวิธีการต่อสายเข้ากับอุปกรณ์ DAO ที่แตกต่างกันตามความเหมาะสมซึ่งสามารถแบ่งเป็น 3 วิธี ดังนี้

- 1) ดิฟเฟอเรนเชียล (Differential หรือ DIFF) คือ การวัดสัญญาณที่ขาบวกและขาลบของแหล่งสัญญาณไม่ได้ต่อกับจุดอ้างอิงใด ๆ เช่นกราวด์.
- 2) ซิงเกิล-เอนด์แบบเทียบกราวด์ (Referenced Single-Ended หรือ RSE) คือ การวัดสัญญาณที่มีการเทียบกับ AIGND (analog input ground) ซึ่งใช้เป็นจุดอ้างอิง โดยต่อขาบวกของช่องสัญญาณไปที่ช่อง CHX และต่อขาลบของทุกช่องร่วมกันที่ AIGND
- 3) ซิงเกิล-เอนด์แบบไม่เทียบกราวด์ (Non-Referenced Single-Ended หรือ NRSE) เป็นระบบที่คล้ายกับ RSE แต่จะใช้จุดอ้างอิงเป็น AISENSE (analog input sense) แทนซึ่งศักย์ไฟฟ้าที่จุดนี้จะแตกต่างกับ AIGND



รูปที่ 2.76 แสดงประเภทของระบบการวัดสัญญาณ

2.11 การสร้างแอปพลิเคชัน

2.11.1 การเตรียมโค้ดก่อนการสร้างแอปพลิเคชัน

ในการรันโปรแกรม VI บน LabVIEW จัดเป็นการรันบนระบบการพัฒนา (Development system) แต่เมื่อทำการรวบรวมเป็น EXE แล้ว จะเรียกระบบนี้ว่าระบบรันไทม์ (Run-time System) ซึ่งมีบางจุดที่แตกต่างกัน จำเป็นต้องเตรียมโค้ดให้เหมาะสมก่อนทำการรวบรวม ดังนี้

1.1) เส้นทางสัมพัทธ์ (Relative Path)

การใช้ Relative path ถือเป็นฟังก์ชันที่พบปัญหามากที่สุดในการสร้างแอปพลิเคชัน โดยการใช้ relative path ด้วยฟังก์ชัน Current VI's Path ซึ่งสามารถเลือกได้จาก Function Pallets: Programming >> File I/O >> File Constants ซึ่งฟังก์ชัน Current VI's Path เป็นฟังก์ชันที่มีประโยชน์มากในการระบุตำแหน่งของตัวไฟล์ VI ในปัจจุบัน เพราะจะให้ค่าเส้นทาง (Path) ปัจจุบันที่ไฟล์ VI ถูกบันทึกไว้ในดิสก์ และถ้ามีการย้ายไฟล์ เส้นทางก็จะเปลี่ยนตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีนี้ VI ถูกสร้างมาใหม่และยังไม่ถูกบันทึกมาก่อน ค่าเส้นทางจะเป็น <Not A Path> แต่เมื่อนำ VI มาสร้างเป็นแอปพลิเคชันแล้วบันทึกไว้ในโฟลเดอร์เดิม ดังรูป ทำการบันทึกชื่อเป็น Application.exe เมื่อนำมารัน ค่าเส้นทาง (Path) จะมีการเปลี่ยนแปลงดังที่แสดงในภาพ โดยจะถูกซ่อนด้วยโฟลเดอร์ที่ใช้ชื่อเดียวกับไฟล์ EXE อีกชั้นหนึ่งโดยที่ไม่มีโฟลเดอร์ชื่อนั้นอยู่จริง อย่างไรก็ตามสามารถกำจัดโฟลเดอร์เสมือนได้ โดยใช้ฟังก์ชัน Strip Path แต่อาจจะทำให้เกิดความยุ่งยาก เนื่องจากต้องกลับไปแก้ไขโค้ดใน LabVIEW

นอกจากนี้ยังมีทางเลือกสำหรับการเขียนโค้ดครั้งเดียว แต่สามารถรันได้ผล Relative path เหมือนกันทั้งระบบพัฒนา (Development System) บน LabVIEW และบน EXE คือการใช้ Property Node Path โดยเลือกจาก Function Palletts: Programming >> Application Control เพื่อตรวจสอบว่าขณะนี้ VI กำลังทำงานอยู่บนสิ่งแวดล้อมแบบไหน เมื่อขึ้นหน้าต่าง Property Node ที่อ่านค่า Application.Kind จะให้ค่าออกมาเป็น Enum ถ้า VI รันอยู่บนระบบรันไทม์ (Run-time System) จะต้องเพิ่มฟังก์ชัน Strip Path แต่ถ้ารันอยู่บนระบบพัฒนา (Development System) ไม่ต้องทำอะไร นอกจากนี้รูปที่ แสดงให้เห็นวิธีการเขียนโค้ดเพื่อหาค่าแห่งโฟลเดอร์ที่เก็บ VI หรือแอปพลิเคชัน

1.2) ปิดการทำงาน LabVIEW (Quit LabVIEW)

เมื่อกดปุ่มหยุดหรือออกจากแอปพลิเคชัน หน้าต่าง Top-Level VI จะยังคงถูกเปิดค้างทิ้งไว้ เพื่อให้ผู้ใช้งานปิดเอง แต่ผู้ใช้งานสามารถเลือกที่จะสั่งให้ปิดและออกจาก LabVIEW อย่างสมบูรณ์แบบได้ด้วยฟังก์ชัน Quit LabVIEW โดยเลือกได้จาก Function Palletts: Programming>> Application Control ซึ่งจะวางฟังก์ชันนี้ไว้เป็นตัวสุดท้ายก่อนจบ VI

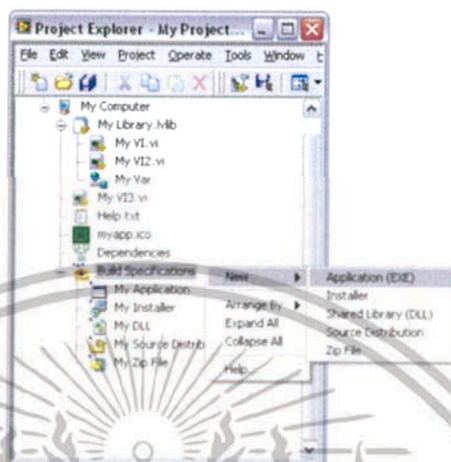
นอกจากนี้สามารถประยุกต์ใช้ Property Node ในการระบุว่าจะให้ใช้ Quit LabVIEW ในสิ่งแวดล้อมแบบไหน เพราะการที่ออกจาก LabVIEW ทุกครั้งที่ทดลองรัน VI จะทำให้เสียเวลา ดังนั้นอาจจะสั่งให้เรียก Quit LabVIEW เฉพาะตอนที่รวบรวมเป็น EXE แล้วเท่านั้น

2.11.2 การสร้าง EXE

การสร้าง EXE เมื่อพร้อมที่จะสร้าง EXE แล้วให้ตรวจสอบว่า LabVIEW ในเครื่องเป็นรุ่น Professional Development System และมี VI และ SubVI ทุกตัวรวมทั้งไฟล์ประกอบต่าง ๆ อยู่ในโปรเจกต์เรียบร้อยแล้วจึงเริ่มสร้าง Build Specification

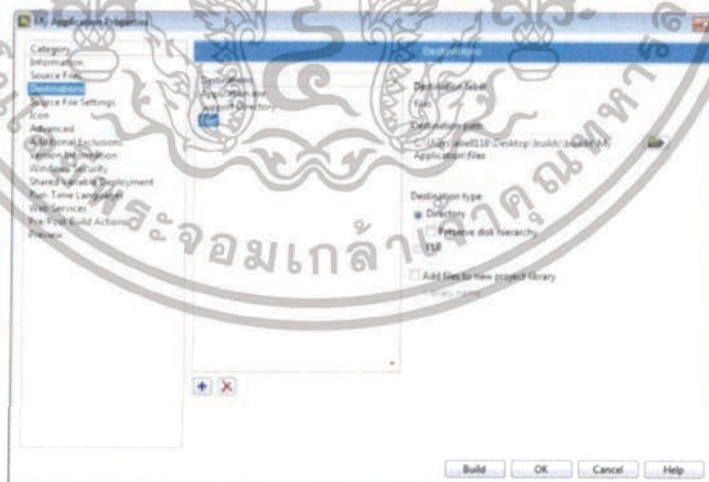
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง Build Specification เป็นการตั้งค่าเพื่อเตรียมสร้าง EXE, Installer หรือ DLL เป็นต้น ซึ่งสามารถเตรียม Build Specification ได้หลาย ๆ ตัวโดยใช้ชื่อต่างกัน แล้วบันทึกไว้ในโปรเจกต์โดยให้คลิกขวาที่ Build Specifications>> New>> Application (EXE) ดังรูป



รูปที่ 2.77 การสร้าง Build Specification

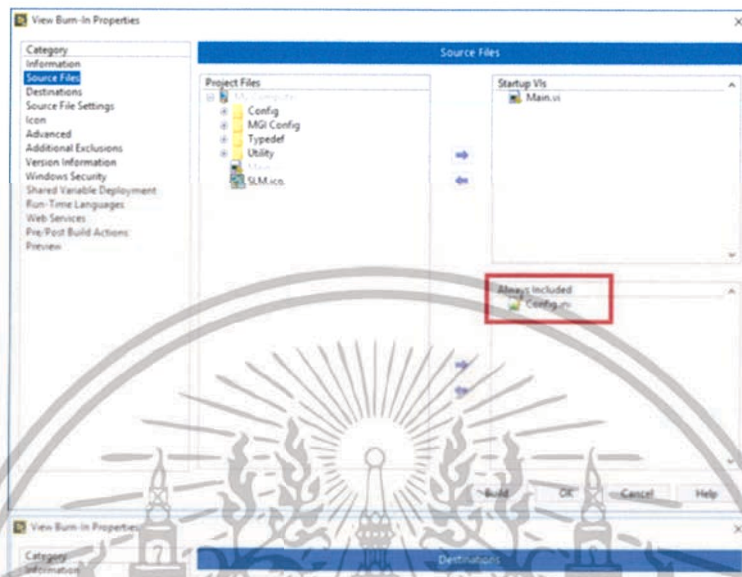
โดยจะปรากฏหน้าต่างสำหรับการตั้งค่าคุณสมบัติต่าง ๆ ซึ่งสามารถดูที่ช่อง Category โดยในหัวข้อแรกคือ Information ที่ช่อง Build Specification Name ไว้สำหรับใส่ ชื่อไฟล์ และรายละเอียดต่าง ๆ สำหรับช่อง Destination directory ไว้สำหรับระบุตำแหน่งโฟลเดอร์สำหรับเก็บไฟล์ ดังรูป



รูปที่ 2.78 หน้าต่างสำหรับการตั้งค่าคุณสมบัติต่าง ๆ

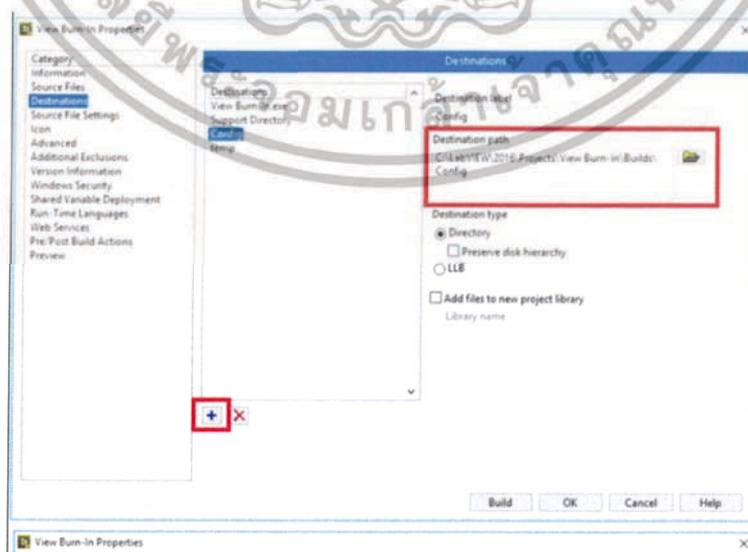
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถัดจาก Information จะเป็น Source Files เป็นส่วนไว้สำหรับบอกว่า VI ไหนในโปรเจกต์เป็น VI ตัวแรกที่จะรันและไฟล์ไหนเป็นไฟล์ประกอบ โดยปกติแล้ว VI หลัก (MainVI) จะอยู่ในช่อง Startup VIs และ SubMI ส่วนช่อง Always Included ไว้สำหรับไฟล์ประกอบอื่น ๆ ดังรูป



รูปที่ 2.79 Source Files เป็นส่วนไว้สำหรับบอกว่า VI

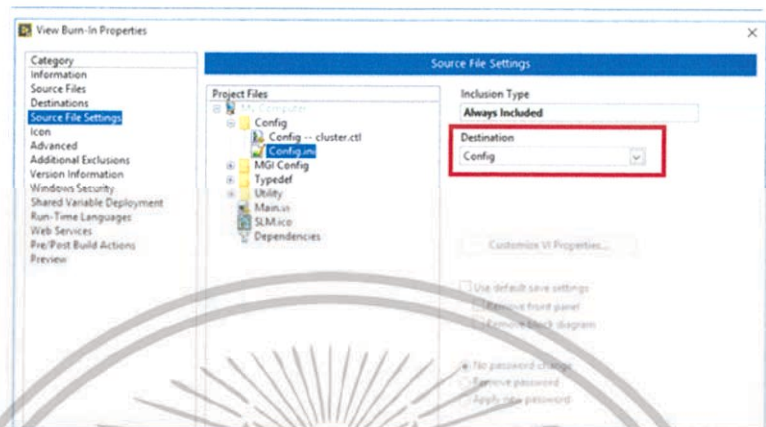
ถัดจาก Source Files จะเป็น Destinations ซึ่งไว้สำหรับระบุเส้นทาง (path) ของไฟล์ต่างๆที่จะสร้างขึ้นว่าจะถูกเก็บไว้ที่ใดบ้าง โดยเริ่มต้นจะมีให้ 2 เส้นทาง คือ เส้นทางของ Application.exe ที่เลือกไว้ในหัวข้อข้างต้นกับ Support Directory ถ้าต้องการเพิ่มสามารถเพิ่มได้โดยกดที่เครื่องหมายบวกด้านล่าง ดังรูป



รูปที่ 2.80 Destinations ซึ่งไว้สำหรับระบุเส้นทาง (path) ของไฟล์ต่างๆ

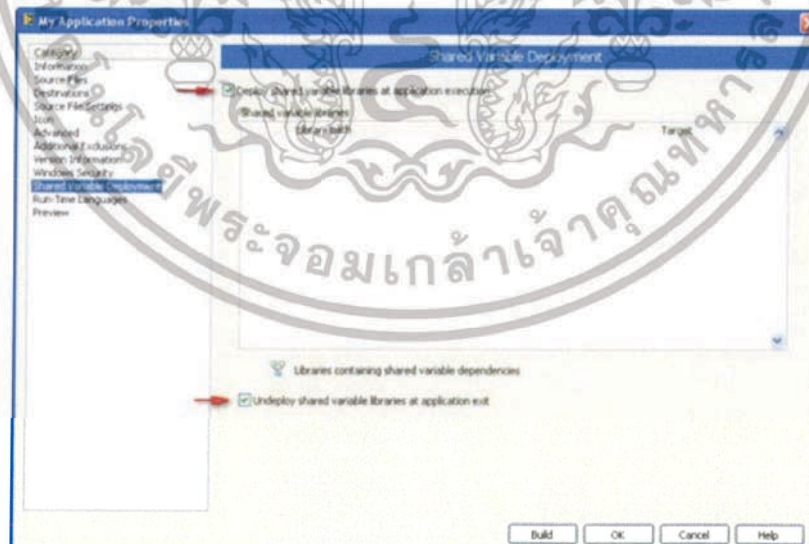
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Source File Settings ไว้สำหรับได้แก้ไข คุณสมบัติของ VI (VI Properties) ของ VI ทุกตัว โดยไม่จำเป็นที่ไฟล์ EXE จะต้องมีคุณสมบัติของ VI เหมือนกับไฟล์สกุล .vi ดังรูป



รูปที่ 2.81 Source File Settings

ในหัวข้อ Shared Variable Deployment ไว้สำหรับการติดตั้ง ตัวแปรแบบใช้ร่วมกัน (Shared Variable) เมื่อนำแอปพลิเคชันไปติดตั้งในคอมพิวเตอร์เครื่องอื่น ซึ่งจำเป็นจะต้องทำการปรับตัวแปร (deploy variable) ในเครื่องใหม่ด้วย ดังรูป



รูปที่ 2.82 Shared Variable Deployment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับหัวข้อ Preview จะใช้เมื่อตั้งค่าเสร็จเรียบร้อยแล้ว โดยปุ่ม Generate Preview ไว้สำหรับดูตำแหน่งไฟล์เดสก์ทอปที่ไว้เก็บไฟล์ที่สร้าง เมื่อทุกอย่างเรียบร้อยแล้วสามารถสร้างแอปพลิเคชันโดยการกดปุ่ม Build ได้ทันที

2.11.3 การสร้าง Installer

การสร้าง Installer สามารถใช้ Application Builder ใน LabVIEW สร้างตัว Installer เพื่อเป็นตัวติดตั้งแอปพลิเคชันที่สร้างขึ้นมาได้ และจะมีประโยชน์มากขึ้นถ้ารวม รันไทม์เอนจิน (Run-Time Engine) และไดร์ฟเวอร์ที่เกี่ยวข้องลงไปด้วย

นอกจากนี้ยังสามารถติดตั้งแอปพลิเคชันได้ภายในขั้นตอนเดียว โดยก่อนที่จะสร้าง Installer จะต้องสร้าง EXE ตามข้างต้นให้เรียบร้อยแล้ว จากนั้นเลือก Build Specification>> New>> Installer

จากนั้นขั้นตอนจะคล้ายกับการสร้าง EXE โดยให้ระบุชื่อและเส้นทาง (path) ในการเก็บ Installer ส่วนในหัวข้อ Source Files จะเป็นการระบุว่าจะไปติดตั้งอยู่ในส่วนไหนของฮาร์ดดิสก์ ในส่วน Additional Installers เป็นการใส่ รันไทม์เอนจิน (Run-Time Engine) หรือไดร์ฟเวอร์ที่เกี่ยวข้อง เพื่อเป็นการติดตั้งซอฟต์แวร์ทั้งหมดในการ install ครึ่งเดียวกัน เมื่อป้อนค่าทุกอย่างเรียบร้อยแล้วให้กดปุ่ม Build ซึ่งสามารถนำไฟล์ที่สร้างไปใส่ในดีวีดีเพื่อทำเป็น Installer ของผู้ใช้งานเองได้เลย

บทที่ 3

วิธีการดำเนินการวิจัย

ความไม่สมดุลของส่วนที่หมุนเป็นสาเหตุทำให้เครื่องจักรเกิดความสั่นสะเทือนมากที่สุด ซึ่งการผลิตใบพัดพัดลมจากการอัดฉีดพลาสติกเข้าไปในแบบหล่อ อาจทำให้เกิดความผิดพลาดจากการผลิต เช่น ความไม่สม่ำเสมอในเนื้อพลาสติกทั่วทั้งใบพัดทำให้ความหนาแน่นและน้ำหนักของเนื้อพลาสติกในส่วนต่างๆ ไม่เป็นไปตามที่ออกแบบ จึงทำให้เกิดการสั่นสะเทือนและส่งผลกระทบต่ออายุการใช้งานของพัดลม ความไม่สมดุลจะถูกนำมาพิจารณาเป็นหลัก

การปรับสมดุลของใบพัดพัดลมเป็นการป้องกัน และหลีกเลี่ยงความเสียหายที่เกิดจากความล้าของลูกปืนในตลับลูกปืน ช่วยให้พัดมียอายุการใช้งานที่ยาวขึ้น

3.1 บทนำ

ขั้นตอนการดำเนินงานจะเริ่มตั้งแต่ การวิเคราะห์ปัญหา จนกระทั่งถึงการวิเคราะห์ผลข้อมูลซึ่งประกอบไปด้วย วิธีการคำนวณค่าที่รับจากเซนเซอร์การสั่น จนถึงการคำนวณมวลที่ทำให้สมดุล ที่จะสามารถเพิ่มเข้าไปบนใบพัดพัดลม เป็นต้น

3.2 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงานจะเริ่มตั้งแต่การศึกษาพฤติกรรมกลไกการสั่นของพัดลมที่ความเร็วรอบการใช้งานต่างๆ เช่น 4000 รอบต่อนาที เป็นต้น และศึกษาองค์ประกอบของโครงสร้างพัดลมรวมถึงขนาดต่างๆ ซึ่งจะนำค่าข้อมูลเหล่านี้มาสร้างเป็นแบบจำลองทางคณิตศาสตร์ และสร้างแผนภาพวัตถุอิสระ (Free Body Diagram) เพื่อทำการคำนวณความเร่งจากตำแหน่งที่เซนเซอร์ความเร่งรับค่าการสั่นเข้ามา ไปสู่ความเร่งในตำแหน่งบนระนาบที่ต้องการจะทำการปรับสมดุลในโครงสร้างของพัดลม และทำการเขียนขั้นตอนการคำนวณลงในโปรแกรม LABVIEW ของบริษัท National Instrument (NI) เพื่อทำการแสดงผลในคอมพิวเตอร์เป็น น้ำหนักมวลที่จะทำให้สมดุลและตำแหน่งที่ต้องทำการติดตั้งมวล และทำการทดลองกับพัดลมเพื่อเปรียบเทียบค่าจากเครื่องที่มีอยู่ในภาคอุตสาหกรรม

- 1) เลือกชนิดใบพัดพัดลมที่จะนำมาสร้างเครื่องมือวัดการสั่นและศึกษาโครงสร้างของใบพัดพัดลม
- 2) ศึกษาแบบจำลองทางคณิตศาสตร์ของการสั่นทั้งสองระนาบของใบพัดลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) สร้างแบบจำลองทางคณิตศาสตร์ และสร้างแผนภาพวัตถุอิสระของใบพัดพัดลม
- 4) ศึกษาการใช้งานโปรแกรม LABVIEW และการสร้าง Virtual Instrument
- 5) ศึกษาการใช้งานการรับค่าจากเซอร์เซอร์โดยใช้ Data acquisition จาก NI
- 6) ออกแบบและสร้างวงจรการเก็บข้อมูลและการควบคุมการทำงานของอุปกรณ์
- 7) การสอบเทียบเซ็นเซอร์
- 8) ทำการวัดการสั่นจากใบพัดพัดลมตัวอย่างและนำผลที่ได้มาเปรียบเทียบกับค่าจากเครื่องมือที่ใช้ในภาคอุตสาหกรรม

3.3 Fourier Transform

การวิเคราะห์การสั่นสะเทือนซึ่งรับมาจากเซนเซอร์เป็นค่าความเร่ง (accelerometer) จะไม่สามารถหาขนาดของการสั่นได้โดยตรง จำเป็นต้องใช้การแปลงฟูเรียร์ (Fourier Transform) เพื่อแปลงค่าที่ได้รับมาซึ่งเป็นฟังก์ชันของเวลา (Time domain) ให้กลายเป็นฟังก์ชันของความถี่ (Frequency domain) จึงจะสามารถวิเคราะห์ขนาดของการสั่นได้จากความถี่ที่ต้องการวิเคราะห์ได้โดยตรง

3.3.1 Discrete Fourier Transform (DFT) เป็นวิธีการแปลงลำดับของตัวเลขทั้งที่เป็นจำนวนจริงและจำนวนเชิงซ้อนจำนวน N จุด ไปอยู่ในรูปของลำดับของ ตัวเลขเชิงซ้อนจำนวน N จุดอีกชุดหนึ่ง โดยมีวิธีการแปลงตามสมการที่ (3.1) ให้ $x(n), n = 0, 1, 2, \dots, N - 1$ เป็นลำดับของตัวเลขจำนวนจริงหรือลำดับของตัวเลขเชิงซ้อน และ $X(m), m = 0, 1, 2, \dots, N - 1$ เป็น DFT ของ $x(n)$

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nm/N}, j = \sqrt{-1} \quad (3.1)$$

อาจเรียก $X(m)$ วาเป็น DFT ขนาด N จุด (N-point DFT) ของ x เพราะจำนวนจุดที่ใช้ในการคำนวณ DFT นั้นมีผลกับการใช้งาน DFT ในการประมาณสเปกตรัมของสัญญาณ ในกรณีเช่นนี้จะแทน X ด้วย X_N

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าค่าของ DFT หนึ่งจุดคือ $X(m)$ นั้นได้จากการบวกกันของผลคูณของ $x(n)$ กับเลขเชิงซ้อน $e^{-j2\pi nm/N}$, $n = 0, 1, 2, \dots, N-1$ ซึ่งเป็นรากที่ N ของหนึ่ง (N^{th} root of unit)

3.3.2 Fast Fourier Transform (FFT) การใช้งาน Discrete Fourier Transform หรือ DFT นั้น จะต้องใช้ DFT ที่มีจำนวนจุด N สูงมากพอ แต่ปัญหาของ DFT คือ จำนวนครั้งในการคำนวณตามนิยามของ DFT นั้นแปรตาม N^2 ดังนั้นถ้าไม่มีวิธีการคำนวณ DFT ให้เร็วพอที่จะคำนวณ DFT ที่ N มีค่าสูง ๆ ได้ภายในเวลาอันสมควรแล้ว DFT ก็จะทำให้การแก้ไขความไม่สมดุลในใบพัดพัดลมในสายการผลิตเป็นไปได้โดยช้า จึงใช้วิธีการคำนวณโดยวิธี Fast Fourier Transform (FFT) โดยมีขั้นตอนดังนี้

ให้ $x(n)$ เป็นลำดับเลขเชิงซ้อน หรือลำดับเลขจำนวนจริง จำนวน $N = 2k$ จุด โดย $n = 0, 1, 2, \dots, N-1$ แล้ว $X(m)$ คือ DFT ของ $x(n)$ ที่จุด m โดย $m = 0, 1, 2, \dots, N-1$ จะได้

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nm/N} = \sum_{n=0}^{N-1} x(n) W_N^{mn} \quad (3.2)$$

$$W_N = e^{-j2\pi 1/N} \quad (3.3)$$

คือรากที่ N ของ 1 ตัวหลัก (a primitive N^{th} -root of unity) ถ้าแบ่งอินพุต $x(n)$ เป็นสองส่วน โดยส่วนที่หนึ่งเป็นส่วนที่ตัวชี้เป็นเลขคู่ และอีกส่วนหนึ่งเป็นเลขคี่ แล้วจะเขียน $X(m)$ ใหม่ได้ดังนี้

$$\begin{aligned} X(m) &= \sum_{n=0}^{N/2-1} x(2n) W_N^{2mn} + \sum_{n=0}^{N/2-1} x(2n+1) W_N^{(2n+1)n} \\ &= \sum_{n=0}^{N/2-1} x(2n) (W_N^2)^{mn} + \sum_{n=0}^{N/2-1} x(2n+1) W_N^m (W_N^2)^{mn} \end{aligned} \quad (3.4)$$

และเนื่องจาก

$$W_N^2 = e^{-j2\pi 2/N} = e^{-j2\pi \frac{1}{N/2}} = W_{N/2} \quad (3.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งคือรากที่ $N/2$ ของ 1 ตัวหลัก (a primitive $(N/2)^{th}$ - root of unity) จะได้

$$X(m) = \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{mn} + W_N^m \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{mn} \quad (3.6)$$

จะเห็นว่า $\sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{mn}$ และ $\sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{mn}$ คือการคำนวณ $(N/2)$ -point DFT ดังนั้นวิธีการนี้ได้แบ่งการคำนวณ N -point DFT ออกเป็นการคำนวณ $(N/2)$ -point DFT จำนวน 2 ส่วน โดยนำเอา W_N^{mn} ซึ่งเป็นเลขเชิงซ้อนตัวหนึ่งเรียกว่า twiddle factor ไปคูณกับผลที่ได้จากส่วนที่สอง แลวนำไปบวกกับส่วนที่ 1 สังเกตว่าในขั้นตอนแรกนี่เป็นการแบ่งตัวชี้ n ออกเป็นสองส่วน ทำนองเดียวกันสามารถแบ่งตัวชี้ m สามารถแบ่งตัวออกเป็น 2 ส่วนเช่นเดียวกัน โดยส่วนแรกคือ $m = 0, 1, 2, \dots, N/2 - 1$ ซึ่งใช้การคำนวณตามสมการที่ (3.7) และส่วนที่สองคือ $m = N/2, \dots, N - 1$ ซึ่งเท่ากับตัวชี้ส่วนนี้จะเป็ $m + N/2$ เมื่อ m เป็นตัวชี้ในส่วนแรก พิจารณา $X(m + N/2)$ โดยแทน m ด้วย $m + N/2$ ในสมการที่ (3.7) ดังนี้

$$\begin{aligned} X(m + N/2) &= \sum_{n=0}^{N/2-1} x(2n) W_N^{(m+N/2)n} + W_N^{m+N/2} \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{(m+N/2)n} \\ &= \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{mn} (W_{N/2}^{N/2})^n + W_N^m W_N^{N/2} \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{mn} (W_{N/2}^{N/2})^n \end{aligned} \quad (3.7)$$

และเนื่องจาก

$$W_{N/2}^{N/2} = e^{-j2\pi} = 1 \quad (3.8)$$

และ

$$W_N^{N/2} = e^{-j2\pi(N/2)/N} = e^{-j\pi} = -1 \quad (3.9)$$

จะได้

$$X(m + N/2) = \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{mn} + W_N^m \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{mn} \quad (3.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าในการคำนวณ $X(m + N/2)$ นั้น ต่างกับการคำนวณ $X(m)$ เพียงการนำส่วนหลังไปลบออกจากส่วนหน้า แทนการไปบวก ด้วยวิธีการนี้สามารถคำนวณ DFT โดย

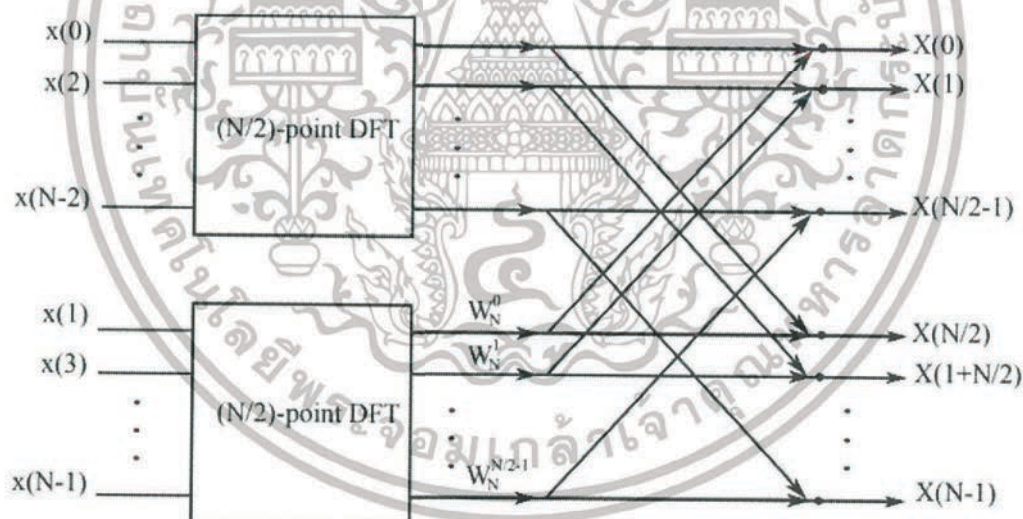
- 1) ให้ $m = 0$
- 2) คำนวณ $(N/2)$ -point DFT ที่จุด m ของอินพุต $x(n)$ ส่วนที่ n เป็นเลขคู่ ให้ผลลัพธ์คือ $X_0(m)$
- 3) คำนวณ $(N/2)$ -point DFT ที่จุด m ของอินพุต $x(n)$ ส่วนที่ n เป็นเลขคี่ ให้ผลลัพธ์คือ $X_1(m)$
- 4) คูณ $X_1(m)$ กับ twiddle factor W_N^m ได้ผลลัพธ์ $X_1(m)$
- 5) คำนวณ $X(m)$ และ $X(m + N/2)$ ดังนี้

$$X(m) = X_0(m) + W_N^m X_1(m) \quad (3.11)$$

$$X(m + N/2) = X_0(m) - X_1(m)$$

6) $m = m + 1$

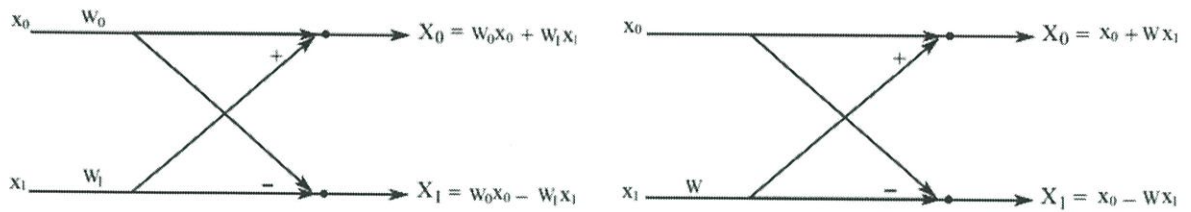
7) ถ้า $m = N/2$ จบ ถ้าไม่ใช่ให้กลับไปขั้นที่ 2



รูปที่ 3.1 ไดอะแกรมการไหล (dataflow diagram) ของการคำนวณ DFT

ตามสมการที่ (3.6) และ (3.10)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

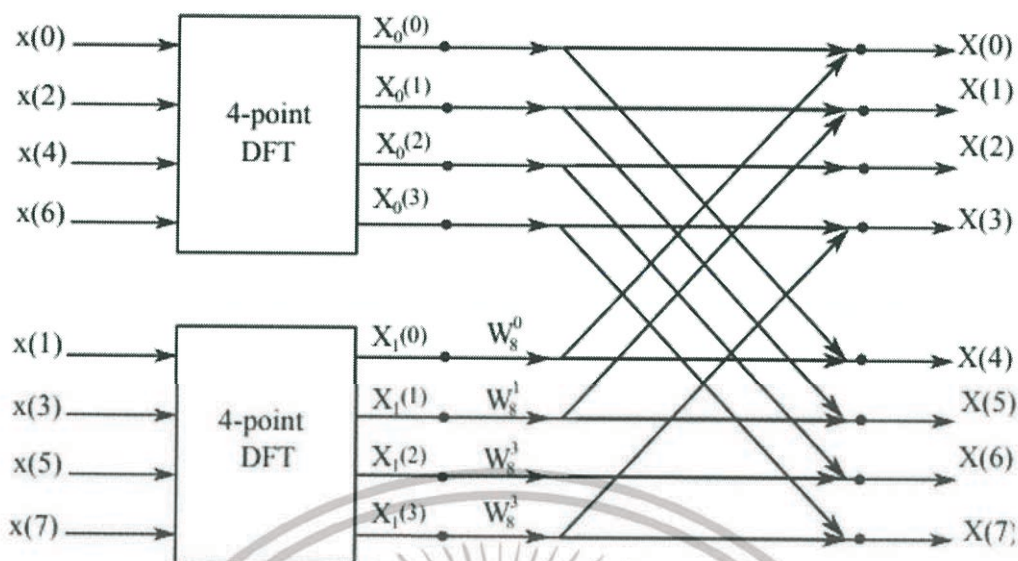


รูปที่ 3.2 Butterfly operations (ซ้าย) แบบtwiddle factorด้านเดียว และ
(ขวา)แบบtwiddle factor สองด้าน

รูปที่ 3.1 แสดงไดอะแกรมการไหล (dataflow diagram) ของการคำนวณ N-point DFT ตามอัลกอริธึมข้างบนหรือ ตามสมการสมการที่ (3.6) และ (3.10) เรียกการคำนวณในสมการที่ (3.11) และ (39) ว่า butterfly operation ซึ่งสามารถแสดงโดยไดอะแกรมการไหล (dataflow diagram) ได้ดังภาพที่ 3.2

Butterfly operations ในรูปทั่วไปจะมีการคูณด้วย twiddle factor ทั้งด้านอินพุต x_0 และอินพุต x_1 ดังแสดงในภาพที่ 3.2 ทางซ้าย แต่ในการคำนวณ FFT ของ Cooley-Tukey นั้น twiddle factor ที่คูณอยู่กับ x_0 คือ W_0 ในรูปจะมีค่าเป็น 1 ซึ่งจะได้อะไร butterfly operation ดังแสดง ภาพที่ 3.2 ทางขวา

หัวใจสำคัญของการคำนวณตามอัลกอริธึมข้างบนก็คือ สามารถใช้วิธีการนี้กับการคำนวณ (N/2)-point DFT ซึ่งแบ่งเป็นการคำนวณ (N/4)-point DFT สองส่วน และเนื่องจาก มี (N/2)-point DFT สองส่วน จะได้อะไรการคำนวณ(N/4)-point DFT ทั้งหมด 4 ส่วน และถ้า $N/4 \neq 2$ สามารถแบ่งต่อไปจนกว่า จะได้อะไรการคำนวณแต่ละส่วนเป็น 2-point DFT และ butterfly operations นอกจากนี้ 2-point DFT เองก็คำนวณด้วย butterfly operationsทำให้ผลจากการแบ่งจนถึง 2-point DFT จะได้อะไรการคำนวณ DFT จากการคำนวณ butterfly operations จำนวน $\frac{N/2}{\log_2 N}$ ครั้ง



รูปที่ 3.3 ไดอะแกรมการไหล (dataflow diagram) ของการคำนวณ 8-point DFT
หลังทำการแบ่งเป็น 4-point DFT สองส่วน

3.4 วิธีการคำนวณค่าความไม่สมดุล

เมื่อได้ค่าจากการทดสอบแล้ว ขั้นตอนต่อไปคือการคำนวณค่ามวลที่ใช้แก้ไขความไม่สมดุลและมุมที่ต้องติดตั้งมวลแก้ไขความไม่สมดุล โดยจะใช้โปรแกรม LABVIEW ที่เขียนขึ้นเพื่อใช้ในการคำนวณและแสดงผลตำแหน่งการปรับสมดุลโดยเฉพาะ

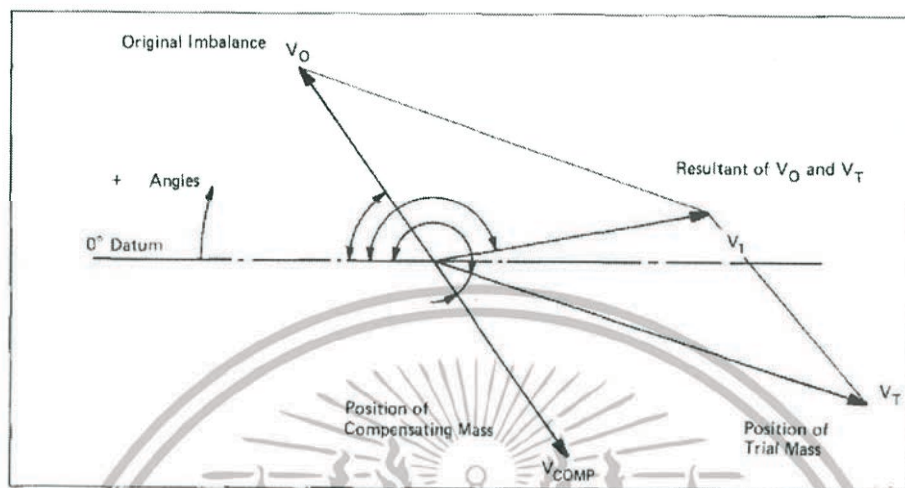
3.4.1 วิธีการปรับสมดุลระนาบเดียวแบบเวกเตอร์

วิธีเวกเตอร์สำหรับการปรับสมดุลแบบระนาบเดียว เพื่อแก้ไขความสมดุลสามารถกำหนดโดยการวัดเวกเตอร์

- 1) เวกเตอร์ \vec{V}_0 แสดงความไม่สมดุลของเครื่องใน เริ่มต้น ความยาวของเวกเตอร์ \vec{V}_0 จากรูปในเท่ากับขนาดของความสั่นสะเทือนและตำแหน่งความไม่สมดุล (Phase)
- 2) เวกเตอร์ \vec{V}_1 แสดงขนาดและตำแหน่งความไม่สมดุล (Phase) หลังจากติดมวลสุ่มไปแล้ว
- 3) เวกเตอร์ \vec{V}_0 และ \vec{V}_1 ต่อกันแล้วได้เป็นเวกเตอร์จะบอกขนาดและตำแหน่งความไม่สมดุล (Phase) \vec{V}_t เวกเตอร์จะหมายถึงเวกเตอร์ที่เกิดจากมวลสุ่มอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) เวกเตอร์ \vec{V}_{comp} ที่วาดต่อกับเวกเตอร์ \vec{V}_0 ด้วยขนาดและทิศทางเดียวกันแต่จุดเริ่มต้นของเวกเตอร์อยู่ที่จุดกำเนิด เวกเตอร์นี้เรียกว่าบีซึ่งแสดงตำแหน่งความไม่สมดุล (Phase) และขนาดของมวลที่ต้องการเพื่อแก้ไขความไม่สมดุลจากเดิม



รูปที่ 3.4 วิธีการปรับสมดุลระนาบเดียวแบบเวกเตอร์

3.4.2 วิธีการปรับสมดุลสองระนาบแบบเวกเตอร์

ใช้วิธีเวกเตอร์สำหรับการปรับสมดุลแบบสองระนาบในการคำนวณหาค่าความไม่สมดุลของวัตถุแบบสองระนาบ คำนวณค่าขนาดการสั่นและตำแหน่งมุมตอนเริ่มต้นของมวลที่ไม่สมดุลของวัตถุหมุนที่ได้จากการคำนวณค่าการสั่นของแต่ละระนาบ จะได้ดังนี้

| | เครื่องวัดความเร็ว A | เครื่องวัดความเร็ว B |
|-----------------------|----------------------|----------------------|
| ค่าเริ่มต้น | \vec{V}_A | \vec{V}_B |
| ติดมวลส่มที่ระนาบที่1 | \vec{V}_A' | \vec{V}_B' |
| ติดมวลส่มที่ระนาบที่2 | \vec{V}_A'' | \vec{V}_B'' |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงการติตมวลสุมที่ระนาบที่ 1 และระนาบที่ 2

- \vec{V}_A เวกเตอร์การสั่นสะเทือนที่เซนเซอร์วัดความเร่ง A
 \vec{V}_B เวกเตอร์การสั่นสะเทือนที่เซนเซอร์วัดความเร่ง B
 \vec{V}'_A เวกเตอร์การสั่นสะเทือนหลังจากติตมวลสุมระนาบที่ 1 ที่เซนเซอร์วัดความเร่ง A
 \vec{V}'_B เวกเตอร์การสั่นสะเทือนหลังจากติตมวลสุมระนาบที่ 1 ที่เซนเซอร์วัดความเร่ง B
 \vec{V}''_A เวกเตอร์การสั่นสะเทือนหลังจากติตมวลสุมระนาบที่ 2 ที่เซนเซอร์วัดความเร่ง A
 \vec{V}''_B เวกเตอร์การสั่นสะเทือนหลังจากติตมวลสุมระนาบที่ 2 ที่เซนเซอร์วัดความเร่ง B

เริ่มต้นยังไม่ติตมวลสุม

$$\vec{V}_A = \vec{A}_{A1}\vec{U}_1 + \vec{A}_{A2}\vec{U}_2 \quad (3.12)$$

$$\vec{V}_B = \vec{A}_{B1}\vec{U}_1 + \vec{A}_{B2}\vec{U}_2 \quad (3.13)$$

- \vec{U}_1 เวกเตอร์ของมวลที่ไม่สมดุลที่ระนาบ 1
 \vec{U}_2 เวกเตอร์ของมวลที่ไม่สมดุลที่ระนาบ 2
 \vec{A}_{A1} เวกเตอร์ความไม่สมดุลที่ระนาบ 1 จากเซนเซอร์วัดความเร่ง A
 \vec{A}_{A2} เวกเตอร์ความไม่สมดุลที่ระนาบ 2 จากเซนเซอร์วัดความเร่ง A
 \vec{A}_{B1} เวกเตอร์ความไม่สมดุลที่ระนาบ 1 จากเซนเซอร์วัดความเร่ง B
 \vec{A}_{B2} เวกเตอร์ความไม่สมดุลที่ระนาบ 2 จากเซนเซอร์วัดความเร่ง B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดตามลุ่มที่ระนาบที่ 1

$$\vec{V}_A' = \vec{A}_{A1}(\vec{U}_1 + \vec{W}_1) + \vec{A}_{A2}\vec{U}_2 \quad (3.14)$$

$$\vec{V}_B' = \vec{A}_{B1}(\vec{U}_1 + \vec{W}_1) + \vec{A}_{B2}\vec{U}_2 \quad (3.15)$$

$(\vec{U}_1 + \vec{W}_1)$ ผลรวมเวกเตอร์ของมวลที่ไม่สมดุลและมวลลุ่ม ที่ระนาบ 1

นำสมการ (3.12) - (3.14);

$$\vec{V}_A - \vec{V}_A' = -\vec{A}_{A1}\vec{W}_1$$

$$\vec{A}_{A1} = \frac{\vec{V}_A' - \vec{V}_A}{\vec{W}_1} \quad (3.16)$$

นำสมการ (3.13) - (3.15);

$$\vec{V}_B - \vec{V}_B' = -\vec{A}_{B1}\vec{W}_1$$

$$\vec{A}_{B1} = \frac{\vec{V}_B' - \vec{V}_B}{\vec{W}_1} \quad (3.17)$$

ติดตามลุ่มที่ระนาบที่ 2

$$\vec{V}_A'' = \vec{A}_{A1}\vec{U}_1 + \vec{A}_{A2}(\vec{U}_2 + \vec{W}_2) \quad (3.18)$$

$$\vec{V}_B'' = \vec{A}_{B1}\vec{U}_1 + \vec{A}_{B2}(\vec{U}_2 + \vec{W}_2) \quad (3.19)$$

$(\vec{U}_2 + \vec{W}_2)$ ผลรวมเวกเตอร์ของมวลที่ไม่สมดุลและมวลลุ่ม ที่ระนาบ 2

นำสมการ (3.12) - (3.18);

$$\vec{V}_A - \vec{V}_A'' = -\vec{A}_{A2}\vec{W}_2$$

$$\vec{A}_{A2} = \frac{\vec{V}_A'' - \vec{V}_A}{\vec{W}_2} \quad (3.20)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำสมการ (3.13) -(3.19);

$$\vec{V}_B - \vec{V}_B'' = -\vec{A}_{B2}\vec{W}_2$$

$$\vec{A}_{B2} = \frac{\vec{V}_B'' - \vec{V}_B}{\vec{W}_2} \quad (3.21)$$

แทนสมการลง (3.16),(3.17),(3.20),(3.21) ใน สมการ (3.12),(3.13)

$$\vec{U}_1 = \frac{\vec{A}_{A2}\vec{V}_B - \vec{A}_{B2}\vec{V}_A}{\vec{A}_{B1}\vec{A}_{A2} - \vec{A}_{A1}\vec{A}_{B2}} \quad (3.22)$$

$$\vec{U}_2 = \frac{\vec{A}_A\vec{V}_{B1} - \vec{A}_{A2}\vec{V}_B}{\vec{A}_{B1}\vec{A}_{A2} - \vec{A}_{A1}\vec{A}_{B2}} \quad (3.23)$$

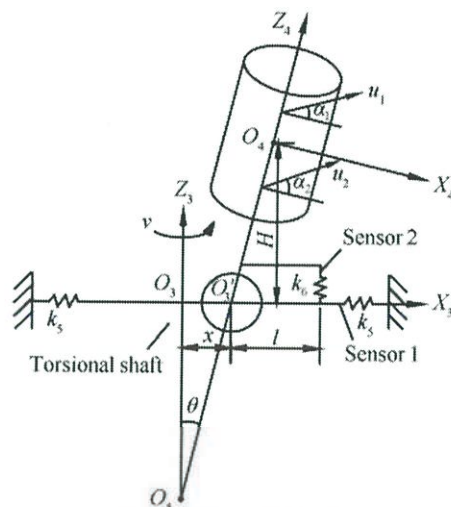
วัตถุประสงค์ทำให้สมดุลโดยการเพิ่มมวล ที่มีขนาดเท่ากับไว้ที่ด้านตรงข้ามในแต่ละระนาบ โดยเวกเตอร์มวลที่ต้องเติม คือ

$$\vec{M}_1 = -\vec{U}_1 \quad (3.24)$$

$$\vec{M}_2 = -\vec{U}_2 \quad (3.25)$$

3.5 สร้างแบบจำลองทางคณิตศาสตร์และสมการการเคลื่อนที่

ในขั้นตอนแรกศึกษาเครื่องมือวัดต้นแบบ และตำแหน่งการติดตั้งเซนเซอร์วัดความเร่งการสั่น (accelerometer) สร้างแผนภาพวัตถุอิสระและสร้างแบบจำลองทางคณิตศาสตร์โดยอ้างอิงสมการการเคลื่อนที่ของการสั่นสะเทือนแบบ harmonic pattern



รูปที่ 3.6 แผนภาพวัตถุอิสระของเครื่องมือวัดความไม่สมดุลของ
ใบพัดลมแบบสองระนาบ^[5]

u_1 คือ ขนาดของความไม่สมดุลในระนาบที่ 1 (g.mm)

u_2 คือ ขนาดของความไม่สมดุลในระนาบที่ 2 (g.mm)

α_1 คือ มุมของความไม่สมดุลในระนาบที่ 1 (g.mm)

α_2 คือ มุมของความไม่สมดุลในระนาบที่ 2 (g.mm)

H คือ ความสูงจากตัวรับรู้ 1 ถึง center of gravity

k_5 คือ ค่าความแข็งของสปริงตามแกนแนวนอน

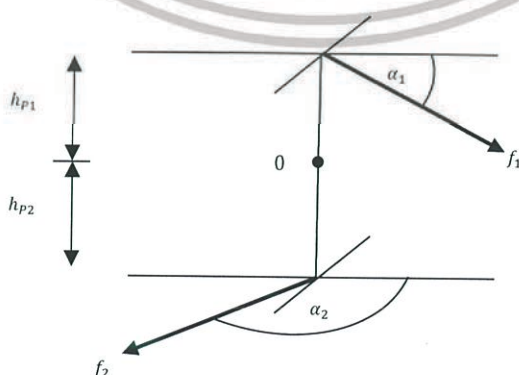
k_6 คือ ค่าความแข็งของสปริงตามแกนแนวตั้ง

x คือ ระยะที่สายตัวออกจากแนวแกนในระนาบเดียวกับตัวรับรู้ 1

l คือ ระยะจากตัวรับรู้ 2 ไปยังแกนหมุน

v คือ ความเร็วในการหมุนของวัตถุหมุน

θ คือ มุมที่สายตัวในแนวตั้งของวัตถุหมุน



รูปที่ 3.7 แผนภาพวัตถุอิสระแสดงตัวอย่างแรงภายในจากการสั่นทั้งสองระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7 แสดงถึงแรงภายในพัลลัมทั้งสองระนาบ ซึ่งต้องการย้ายแรงทั้งสองระนาบให้กระทำผ่านจุดเพียงหนึ่งจุด เมื่อ

$$f = mr\omega^2 \cos(\omega t + \alpha) \quad (3.26)$$

$$u = mr \quad (3.27)$$

จัดรูปสมการได้เป็น

$$f = u\omega^2 \cos(\omega t + \alpha) \quad (3.28)$$

จึงสามารถรวมแรงทั้งสองเป็นแรงที่กระทำผ่านจุด 0 และโมเมนต์คู่ควบที่จุด 0 ด้วยวิธีการทาง Vector ดังสมการต่อไปนี้

$$F = \begin{bmatrix} u_1\omega^2 \cos(\omega t + \alpha_1) + u_2\omega^2 \cos(\omega t + \alpha_2) \\ u_1\omega^2 \sin(\omega t + \alpha_1) + u_2\omega^2 \sin(\omega t + \alpha_2) \end{bmatrix} \quad (3.29)$$

$$T = \begin{bmatrix} 0 \\ 0 \\ h_{p1} \end{bmatrix} \times \begin{bmatrix} u_1\omega^2 \cos(\omega t + \alpha_1) \\ u_1\omega^2 \sin(\omega t + \alpha_1) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -h_{p2} \end{bmatrix} \times \begin{bmatrix} u_2\omega^2 \cos(\omega t + \alpha_2) \\ u_2\omega^2 \sin(\omega t + \alpha_2) \\ 0 \end{bmatrix} \quad (3.30)$$

โดยมีค่าความสูงทั้งสองระนาบจากจุดกึ่งกลาง h_{p1}, h_{p2} 8 มิลลิเมตร และตำแหน่งในการติดตามเพื่อปรับสมดุลวัตถุหมุน จะเป็นตำแหน่งที่มีเฟสตรงกันข้ามหรือเป็นดั่งสมการต่อไปนี้

$$\phi = \alpha + 180^\circ \quad (3.31)$$

เมื่อ α คือมุมเฟสของแรงภายในของวัตถุหมุนที่เกิดจากความไม่สมดุล
 ϕ คือมุมเฟสที่จะทำการติดตั้งมวลเพื่อปรับสมดุลของวัตถุหมุน

ความไม่สมดุล u_1 และ u_2 สามารถอธิบายได้โดยความไม่สมดุลสถิต(static unbalance) U และความไม่สมดุลคู่ควบ(couple unbalance) V ซึ่งมุมเฟสของทั้งสองแทนด้วย α_3 และ α_4 ตามลำดับ ให้ F แทนแรงเหวี่ยงหนีศูนย์กลาง(centrifugal force) และให้ T แทนแรงบิดพลิกกลับ(overturning torque) จากสมการที่ 3.29 กำหนดให้

$$U = \frac{|F|}{\omega^2} \text{ หรือ } U = \frac{\left[\left[u_1\omega^2 \cos(\omega t + \alpha_1) + u_2\omega^2 \cos(\omega t + \alpha_2) \right]^2 + \left[u_1\omega^2 \sin(\omega t + \alpha_1) + u_2\omega^2 \sin(\omega t + \alpha_2) \right]^2 \right]^{1/2}}{\omega^2} \quad (3.32)$$

และ

$$\alpha_3 = \tan^{-1} \left(\frac{u_1\omega^2 \cos(\omega t + \alpha_1) + u_2\omega^2 \cos(\omega t + \alpha_2)}{u_1\omega^2 \sin(\omega t + \alpha_1) + u_2\omega^2 \sin(\omega t + \alpha_2)} \right) \quad (3.33)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ 3.30 กำหนดให้

$$V = \frac{|T|}{\omega^2} \text{ หรือ } V = \frac{\left[\begin{array}{c} 0 \\ 0 \\ h_{p1} \end{array} \right] \times \left[\begin{array}{c} u_1 \omega^2 \cos(\omega t + \alpha_1) \\ u_1 \omega^2 \sin(\omega t + \alpha_1) \\ 0 \end{array} \right] + \left[\begin{array}{c} 0 \\ 0 \\ -h_{p2} \end{array} \right] \times \left[\begin{array}{c} u_2 \omega^2 \cos(\omega t + \alpha_2) \\ u_2 \omega^2 \sin(\omega t + \alpha_2) \\ 0 \end{array} \right]}{\omega^2} \quad (3.34)$$

และ

$$\alpha_4 = \tan^{-1} \left(\frac{h_{p1} u_1 \cos(\omega t + \alpha_1) - h_{p2} u_2 \cos(\omega t + \alpha_2)}{-h_{p1} u_1 \sin(\omega t + \alpha_1) + h_{p2} u_2 \sin(\omega t + \alpha_2)} \right) \quad (3.35)$$

ดังนั้น $F = U \omega^2 \cos(\omega t + \alpha_3)$ และ $T = V \omega^2 \cos(\omega t + \alpha_4)$ ซึ่งในระบบการสั่นนี้จะไม่คำนึงถึงผลจากแรงโน้มถ่วงและตัวหน่วงของระบบ(System damping) ดังนั้นจะสามารถตั้งสมการอนุพันธ์ได้ดังนี้

$$M\ddot{x} + mH\ddot{\theta} + 2k_5 = F \quad (3.36)$$

$$J\ddot{\theta} + m(\ddot{x} + H\ddot{\theta}) + k_6 l^2 \theta = F h_0 + T \quad (3.37)$$

เมื่อ M คือมวลรวมทั้งหมด(total mass), m คือมวลของวัตถุหมุน (rotational mass), J คือโมเมนต์ความเฉื่อย(moment of inertia) รอบจุดหมุนของการหมุน และ h_0 ระยะห่างระหว่างจุดความไม่สมดุลสถิต(static unbalance)เทียบกับศูนย์กลางแรงบิด O'_3 ซึ่งมีค่าดังตารางต่อไปนี้

| พารามิเตอร์ | ตัวแปร | ค่าของพารามิเตอร์ | หน่วย |
|---|------------------|---------------------------|----------------------------|
| มวลและชุดอุปกรณ์ | M | 0.1639 | กิโลกรัม |
| โมเมนต์ความเฉื่อย | J | 12.46627×10^{-6} | กิโลกรัม·เมตร ² |
| ค่าความแข็งของสปริงในแนวนอน | k_5 | 1936 | นิวตัน/เมตร |
| ค่าความแข็งของสปริงในแนวตั้ง | k_6 | 88000 | นิวตัน/เมตร |
| ระยะทางจากกึ่งกลางพัคลมถึงแกนหมุน | h_0 | 0.02196 | เมตร |
| ค่าความสูงทั้งสองระนาบจากจุดกึ่งกลาง | h_{p1}, h_{p2} | 0.008 | เมตร |
| ระยะทางจากกึ่งกลางพัคลมถึงเซนเซอร์ความเร่งในแนวตั้ง | H | 0.01196 | เมตร |
| ระยะทางจากแกนหมุนพัคลมถึงเซนเซอร์ความเร่งในแนวนอน | l | 0.02964 | เมตร |

ใช้การแปลงลาปลาซ (Laplace transform) กับสมการที่ 3.24 และ 3.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$(Ms^2 + 2k_5)X(s) + mHs^2\Theta(s) = F(s) \quad (3.38)$$

$$(Js^2 + mH^2s^2 + k_6l^2)\Theta(s) + mHs^2X(s) = F(s)h_0 + T(s) \quad (3.39)$$

ให้ความถี่ธรรมชาติของการเคลื่อนที่ตามแนวราบ (natural frequencies of the horizontal motion) คือ

$$\omega_{n1} = \sqrt{\frac{2k_5}{M}} \quad (3.40)$$

ให้ความถี่ธรรมชาติของบิดตัว (natural frequencies of the torsional motion) คือ

$$\omega_{n2} = \sqrt{\frac{k_6l^2}{J + mh_0^2}} \quad (3.41)$$

ดังนั้นสมการการเคลื่อนที่ของระบบสามารถเขียนได้เป็น

$$M(\omega_{n1}^2 - \omega^2)X(j\omega) - mH\omega^2\Theta(j\omega) = F(j\omega) \quad (3.42)$$

$$(\omega_{n1}^2 - \omega^2)(J + mH^2)\Theta(j\omega) - mH\omega^2X(j\omega) = F(j\omega)h_0 + T(j\omega) \quad (3.43)$$

เนื่องจากระบบมีตัวรองรับที่มีความแข็งสูง (stiffness) ดังนั้น $\omega_{n1} \gg \omega$, $\omega_{n2} \gg \omega$ ดังนั้นสามารถเขียนสมการ ได้เป็นรูปอย่างง่ายดังนี้

$$2k_5x = U\omega^2 \cos(\omega t + \alpha_3) \quad (3.44)$$

$$k_6l^2\theta = U\omega^2 \cos(\omega t + \alpha_3) + V\omega^2 \cos(\omega t + \alpha_4) \quad (3.45)$$

ค่าที่สอดคล้องกันสองค่า F_{n1} , F_{n2} ซึ่งสอดคล้องกับเซนเซอร์คือ

$$F_{n1} = 2k_5x \quad (3.46)$$

$$F_{n2} = k_6l^2\theta \quad (3.47)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นสมการความไม่สมดุลสถิต(static unbalance) U และความไม่สมดุลคู่ควบ(couple unbalance) V สามารถหาได้จากการรวมสมการที่ 3.44 , 3.45 , 3.46 และ 3.47 เข้าด้วยกัน และทำการจัดรูปใหม่ได้เป็น

$$U = \frac{2F_{n1}}{\omega^2 \cos(\omega t + \alpha_3)} \quad (3.48)$$

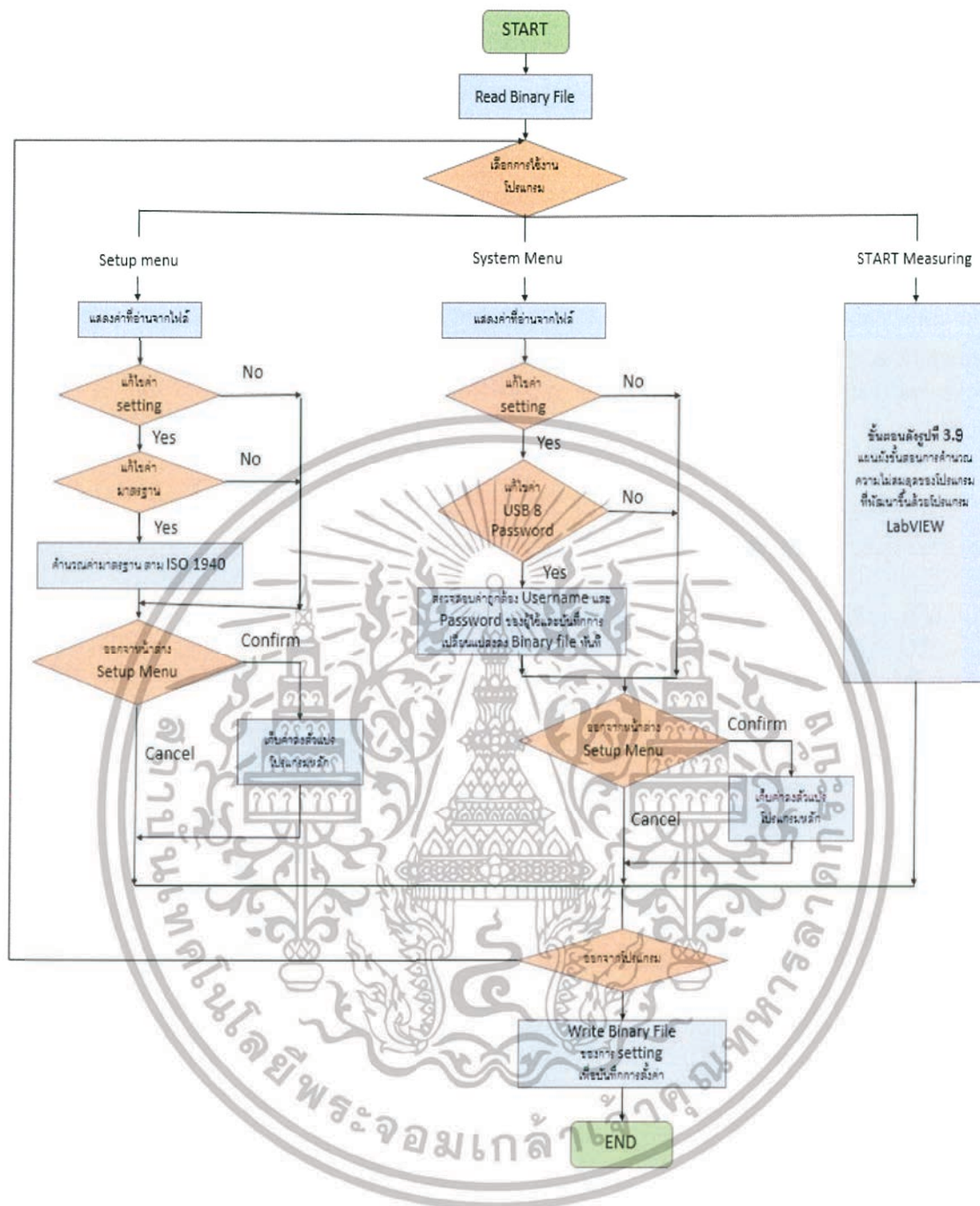
$$V = \frac{F_{n2}l - 2F_{n1}}{\omega^2 \cos(\omega t + \alpha_4)} \quad (3.49)$$

สมการที่ 3.48 และ 3.49 แสดงถึงความสัมพันธ์ระหว่างความไม่สมดุลในระนาบที่ต้องการทราบกับค่าจากเซนเซอร์

ดังนั้นหลังจากสามารถหาค่า F_{n1} , F_{n2} โดยการแทนค่าสมการจาก sensor ลงใน สมการที่ 3.46, 3.47 จึงทำให้ทราบค่า U และ V ดังสมการที่ 3.48, 3.49 และเนื่องจาก U และ V เป็น vector ที่ทำการยุบรวมจาก f_1, f_2 จึงสามารถหาค่าความไม่สมดุลทั้งสองระนาบโดยใช้โปรแกรม LabVIEW จากสมการที่ 3.27, 3.29 และ 3.30

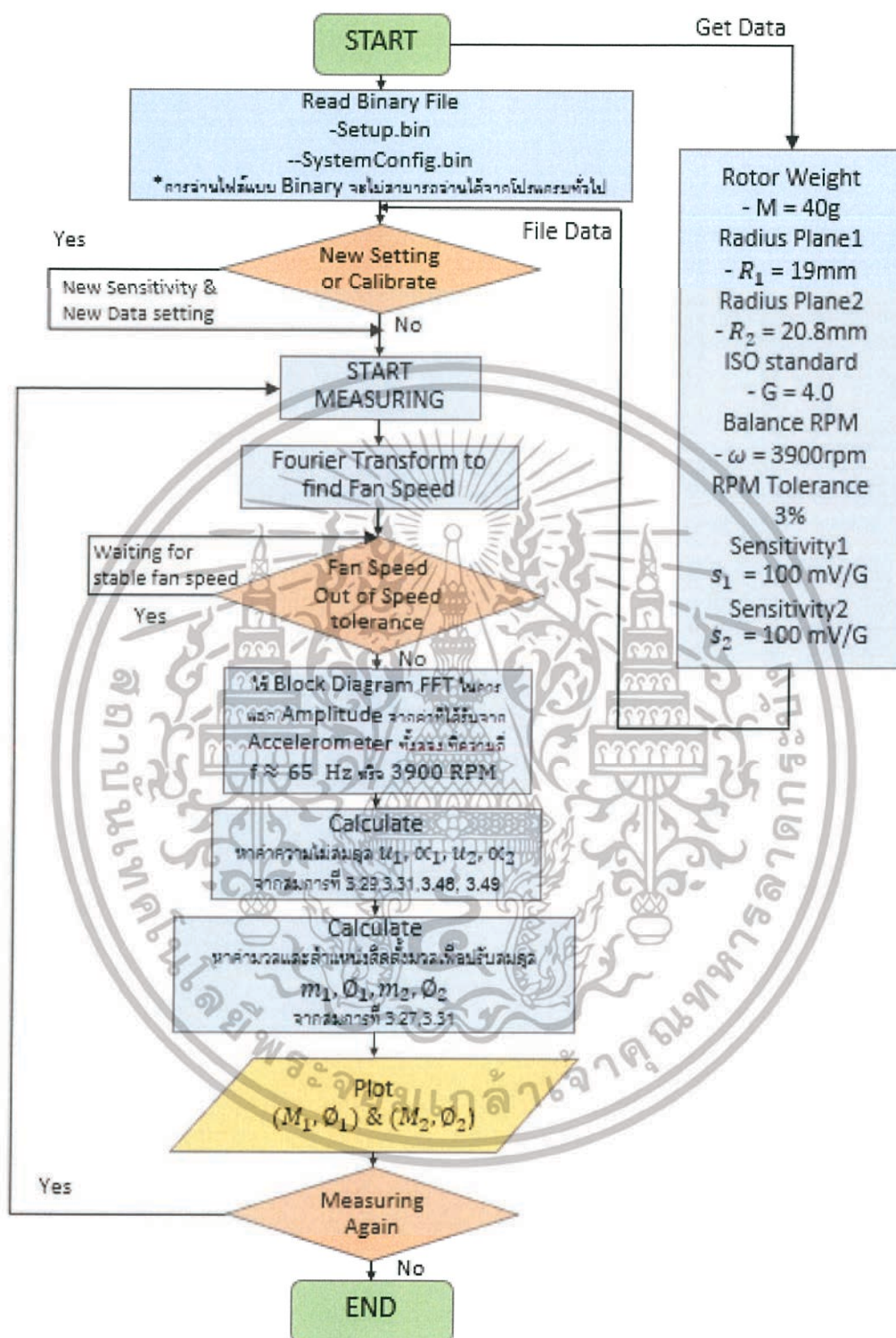
3.6 การออกแบบโปรแกรม

การออกแบบโปรแกรมทางผู้จัดทำได้ออกแบบให้ Front Panel ของโปรแกรมมีความเป็นระเบียบเรียบร้อย ผู้ใช้สามารถเข้าใจและใช้งานได้ง่ายโดยแบ่งออกเป็น 4 ส่วนหลัก โดยส่วนแรกคือส่วนควบคุม เป็นส่วนที่ใช้เลือกสัญญาณอินพุตที่เป็นสัญญาณจริงที่ได้จากชุดอุปกรณ์ รวมถึงส่วนที่ใช้ปรับค่าของสัญญาณจำลองด้วย ในส่วนที่ 2 จะเป็นการแสดงผลบนหน้าจอในรูปแบบการวิเคราะห์และคำนวณพารามิเตอร์ต่างๆ รวมถึงฟังก์ชันอื่นๆที่ใช้ในการวิเคราะห์ ส่วนที่ 3 เป็นส่วนที่ใช้ในการบันทึกข้อมูล ส่วนที่ 4 เป็นส่วนการคำนวณความไม่สมดุล (Unbalance) โดยแผนภาพการทำงานของโปรแกรม (Flow Chart) สามารถแสดงได้ดังนี้



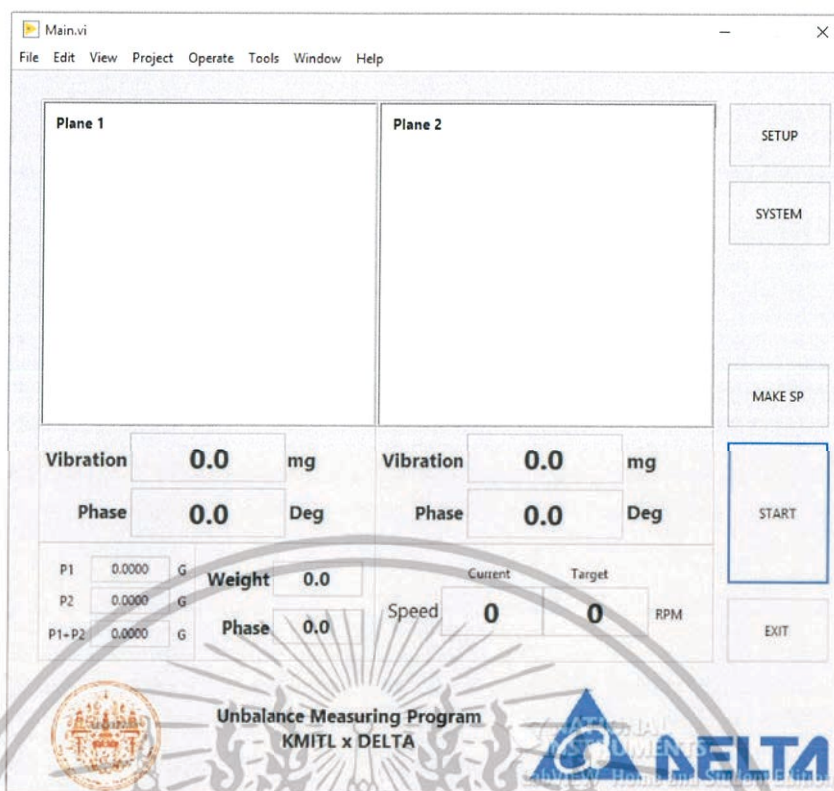
รูปที่ 3.8 แผนผังขั้นตอนการทำงานของโปรแกรมที่พัฒนาขึ้นด้วยโปรแกรม LabVIEW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แผนผังขั้นตอนการคำนวณความไม่สมดุลของโปรแกรมที่พัฒนาขึ้นด้วยโปรแกรม LabVIEW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ตัวอย่าง Front Panel ของโปรแกรมที่ออกแบบ

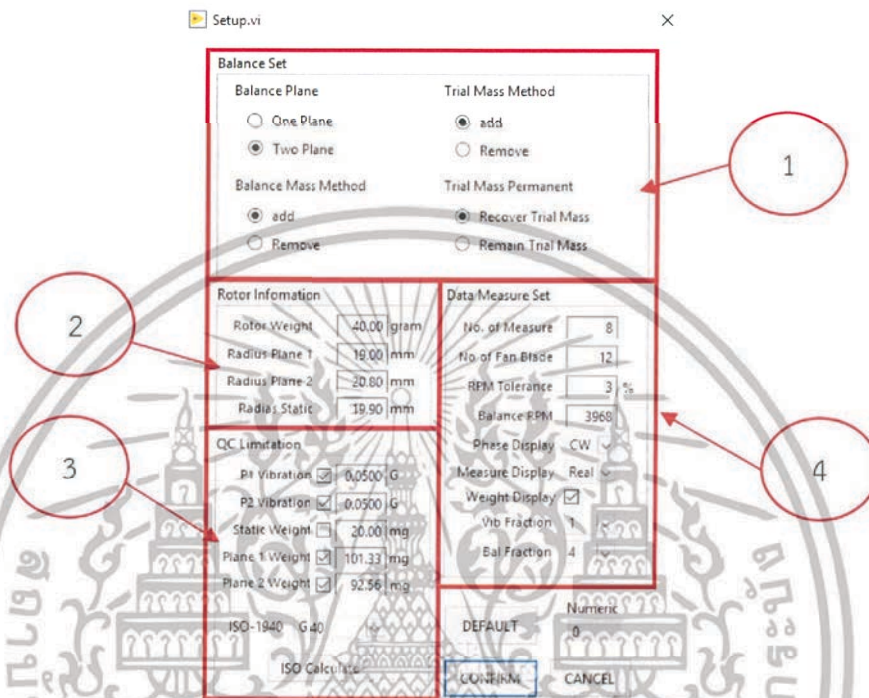


รูปที่ 3.11 ระนาบ 1 และระนาบ 2 บนพัคตลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.1 ส่วนควบคุม (Control)

ส่วนควบคุม (Control) ใช้ในการเลือกสัญญาณอินพุตระหว่างสัญญาณจริงที่ทำการวัดขึ้นมาด้วยโปรแกรม LabVIEW และส่วนที่ใช้ปรับค่าสัญญาณจำลองประกอบด้วยขนาดมุมเฟส ความถี่ของแรงดัน และกระแสไฟฟ้าสามเฟส

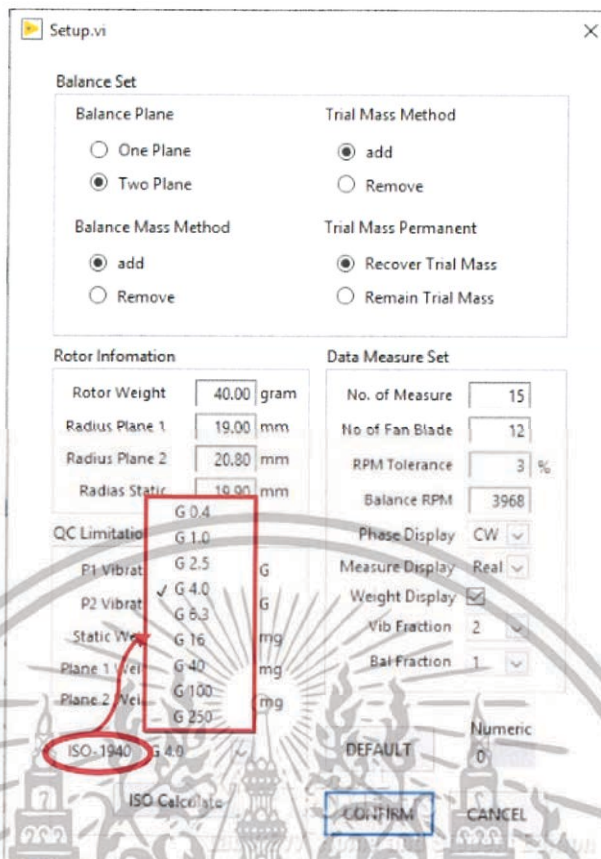


รูปที่ 3.12 การตั้งค่า (Set up) บน Front Panel

จากรูปที่ 3.8 แสดงการตั้งค่า (Set up) บน Front Panel ซึ่งสามารถกดปุ่ม set up ได้ที่ปุ่มด้านบนซ้ายสุดของหน้า front panel เมื่อเปิดหน้า set up ขึ้นมา จะพบข้อมูล 4 ส่วน ดังต่อไปนี้

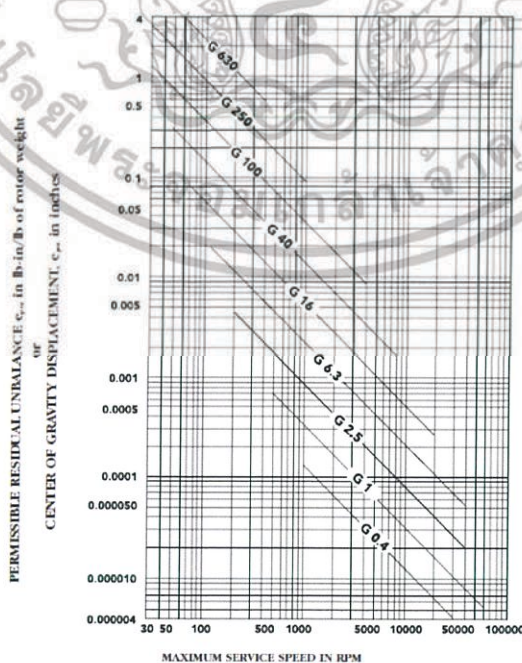
- 1) Balance set เป็นการตั้งค่าบาลานซ์ โดยแบ่งออกเป็น บาลานซ์ของระนาบและมวล ขั้นตอนการทดสอบมวล และมวลถาวร
- 2) Rotor Information เป็นข้อมูลมอเตอร์ของพัดลม
- 3) QC Limitation ขีดจำกัดของพัดลมที่ทดสอบ
- 4) Data Measure set ข้อมูลสำหรับการวัดค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



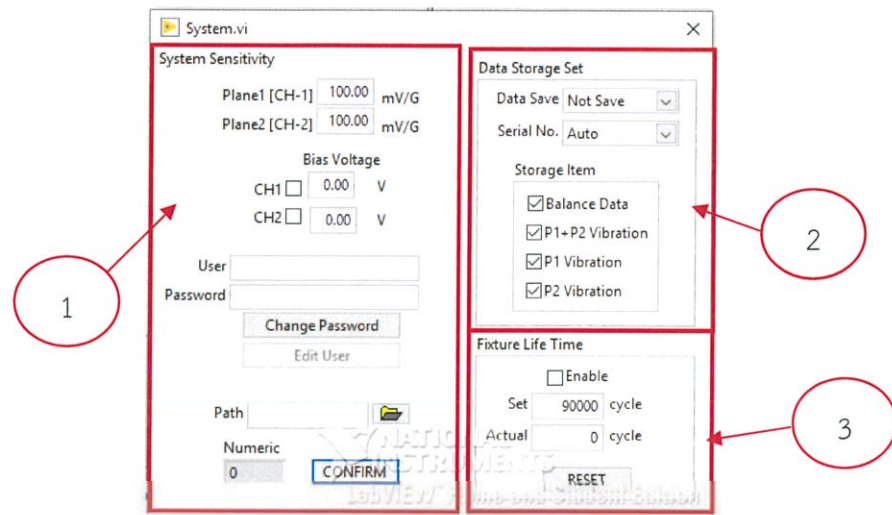
รูปที่ 3.13 การเลือกระดับของมาตรฐาน ISO-1940

ในส่วนของ QC Limitation ตรงส่วนของ ISO-1940 ซึ่งสามารถเลือกเป็นระดับของมาตรฐาน G ตั้งแต่ G 0.4 ถึง G 250 โดยในที่นี้จะเลือกค่า G4.0 ซึ่งระดับ G สามารถดูได้จากรูปที่ 3.10 ดังนี้



รูปที่ 3.14 ของมาตรฐาน ISO-1940^[12]

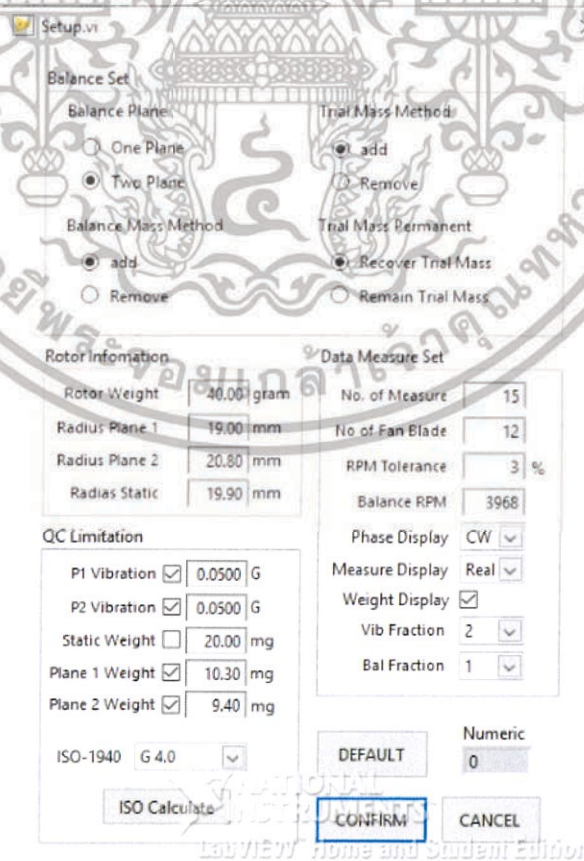
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 หน้าต่าง System บน Front Panel

จากรูปที่ 3.9 แสดงหน้าต่าง System บน Front Panel ซึ่งสามารถกดปุ่ม system ได้ที่ปุ่ม ด้านบนซ้ายสุดของหน้า front panel เมื่อเปิดหน้า system ขึ้นมา จะพบข้อมูล 3 ส่วน ดังต่อไปนี้

- 1) System Sensitivity เป็นการตั้งค่าความไว (Sensitivity) ของตัวรับรู้ (Sensor)
- 2) Data Storage set เป็นการตั้งค่าข้อมูลที่เก็บ
- 3) Fixture Life Time การตั้งค่ารอบในการวัด



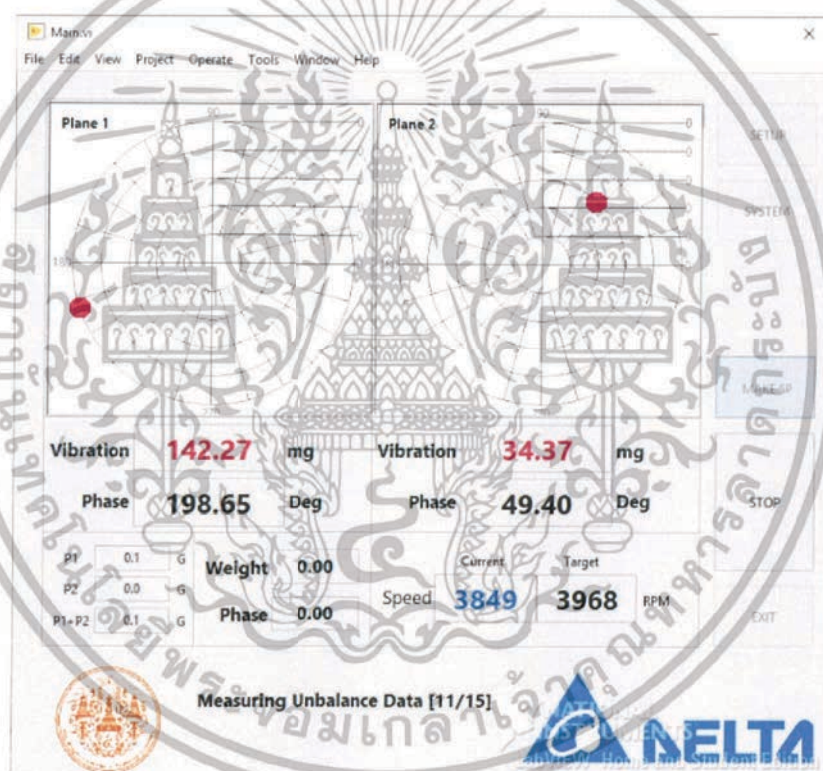
รูปที่ 3.16 แสดงค่าที่เลือกใช้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 หน้าจอแสดงผล (Monitoring)

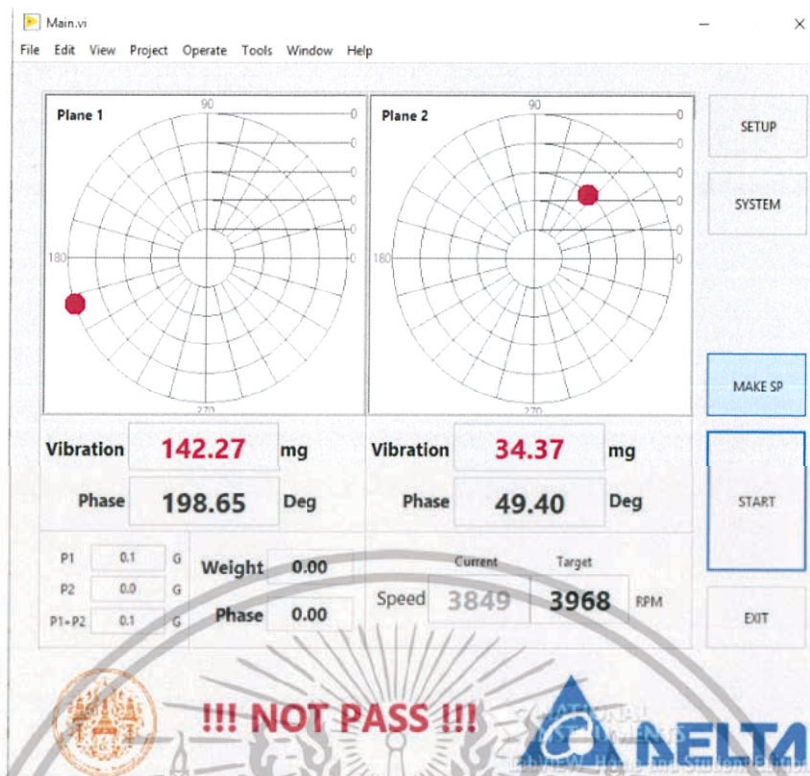
หน้าจอแสดงผลเป็นส่วนของการสร้างหน้าจอแสดงผลนั้น เป็นส่วนที่ใช้แสดงผลของการวิเคราะห์พารามิเตอร์ต่างๆ

3.6.2.1 หน้าจอหลัก Main Monitor ซึ่งเป็นส่วนที่ใช้แสดงผลการ unbalance และคำนวณหามุมเฟสและมวลที่ทำให้สมดุล จากรูปที่ 3.12 ในกรอบสีแดง คือช่อง No. of Measure (Number of Measure) สมมติในการประมวลผลของโปรแกรม ในกรณีที่การประมวลผลไม่ผ่าน หน้าต่างของ Front panel จะแสดงดังรูปที่ 3.13 และ รูปที่ 3.14



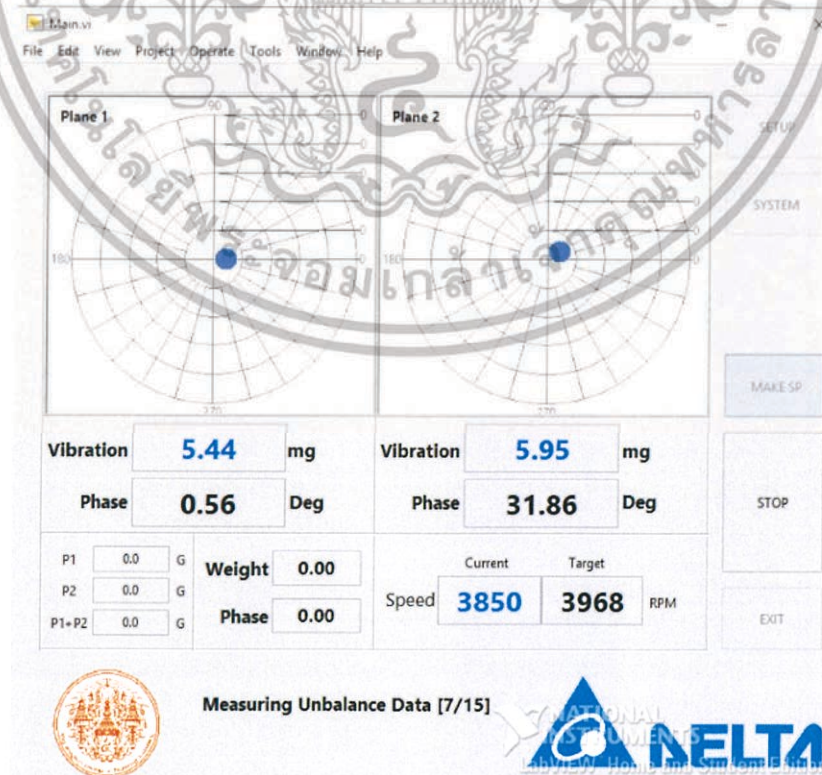
รูปที่ 3.17 หน้าจอ Main Monitor ขณะกำลังประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



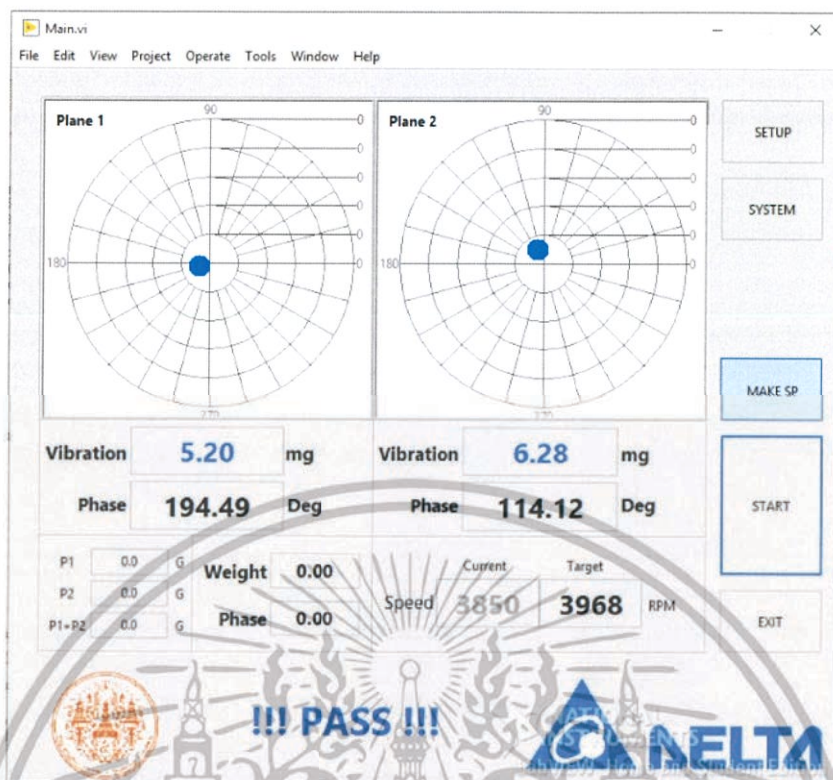
รูปที่ 3.18 หน้าจอ Main Monitor แสดงในกรณีที่ไม่ผ่าน

เมื่อหน้าต่าง Front panel แสดงผลการคำนวณไม่ผ่าน แสดงถึงการตั้งค่าที่ผิดพลาด ซึ่งจะต้องกลับไปแก้ไขยังส่วนของการตั้งค่า โดยในที่นี้ทางผู้จัดได้ยกตัวอย่างในกรณีที่การตั้งค่าผิดพลาดในส่วนของ No. of Measure ซึ่งเมื่อทำการปรับค่าในส่วนนี้เป็นที่เรียบร้อย ผลจากการประมวลผลจะได้ดังรูปต่อไปนี้



รูปที่ 3.19 หน้าจอ Main Monitor ขณะกำลังประมวลผล

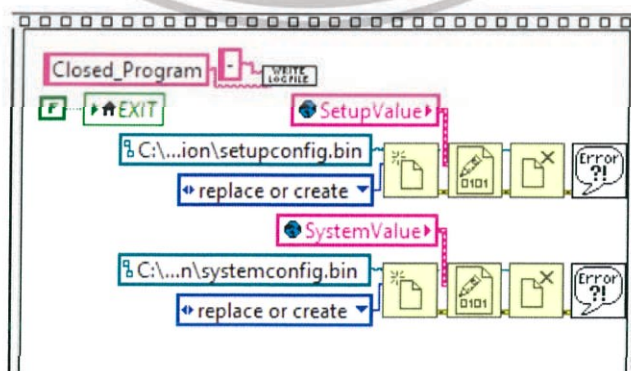
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 หน้าจอ Main Monitor แสดงในกรณีที่ผ่านมา

3.6.3 การบันทึกข้อมูล (Data storage)

ในการที่จะเริ่มโปรแกรมใหม่โดยการใช้การตั้งค่าเดิม หรือ การเก็บชื่อผู้ใช้และรหัสผ่าน จำเป็นต้องมีการเขียนไฟล์เก็บไว้เป็นแบบ Binary File เพื่อให้ไม่สามารกแก้ไขได้โดยง่ายโดยการใช้โปรแกรมภายนอก ในส่วนของกรเขียน Block Diagram ของหน้าจอ Main Monitor ดังรูปต่อไปนี้

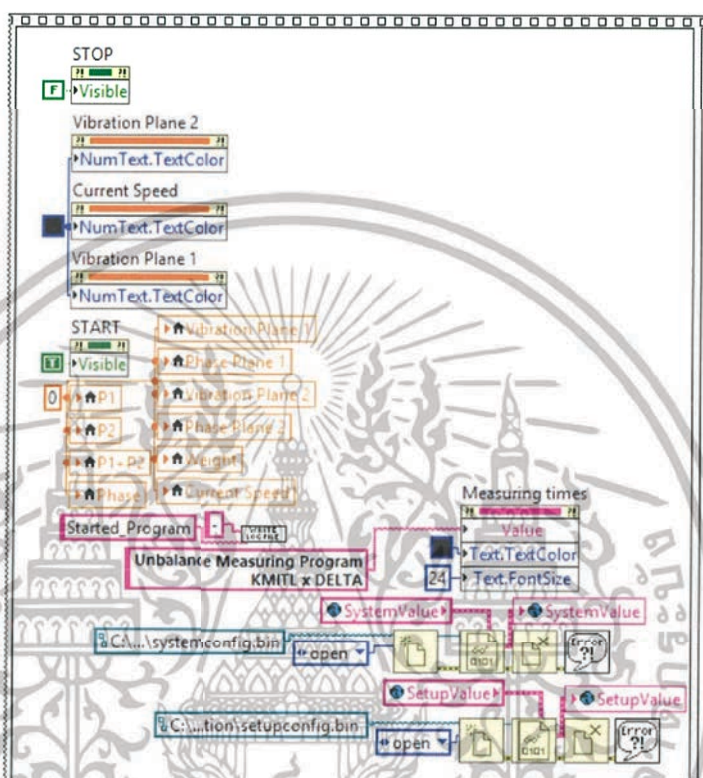


รูปที่ 3.21 Block Diagram ของการเขียนไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.4 การอ่านข้อมูลตั้งค่า (Read Setting Data)

ส่วนอ่านข้อมูลตั้งค่าที่ถูกบันทึกโดยผู้ใช้ครั้งล่าสุด เพื่อให้รวดเร็วและแม่นยำในการคิดคำนวณค่าที่เกี่ยวข้องในแต่ละครั้ง โดยโปรแกรม LabVIEW จะทำการอ่านค่าจากไฟล์ Binary และทำการดึงค่าที่อ่านเข้ามาในโปรแกรม โดยไฟล์แบบ Binary นั้นจะไม่สามารถทำการแก้ไขจากผู้ใช้ภายนอกได้โดยง่าย

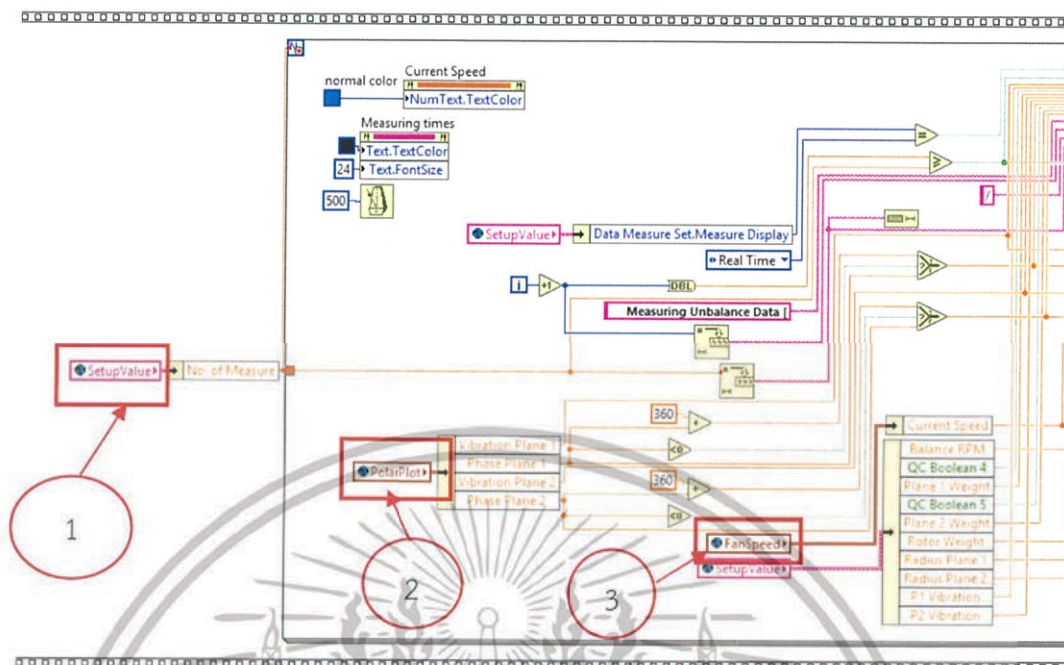


รูปที่ 3.22 Block Diagram โปรแกรมอ่านบันทึกการตั้งค่าจากไฟล์ binary

3.6.5 ส่วนการเตรียมตัวแปรต้นก่อนการคำนวณ (Variable Preparing)

ส่วนการเตรียมตัวแปรต้นนั้น สร้างขึ้นเพื่อนดึงตัวแปรที่อยู่ทั้งใน Local Variable และ Global Variable ให้สามารถพร้อมถูกใช้ในการคำนวณใน Formula Node การเตรียมตัวแปรนี้รวมถึงการดึงค่ามาตรฐานเพื่อกำหนดขอบเขตในการแสดงผลว่าผ่านหรือไม่ ค่าที่ได้จากตัวรับรู้ และความเร็วของวัตถุ หมุน ณ เวลานั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 Block Diagram โปรแกรมส่วนการเตรียมตัวแปรต้น

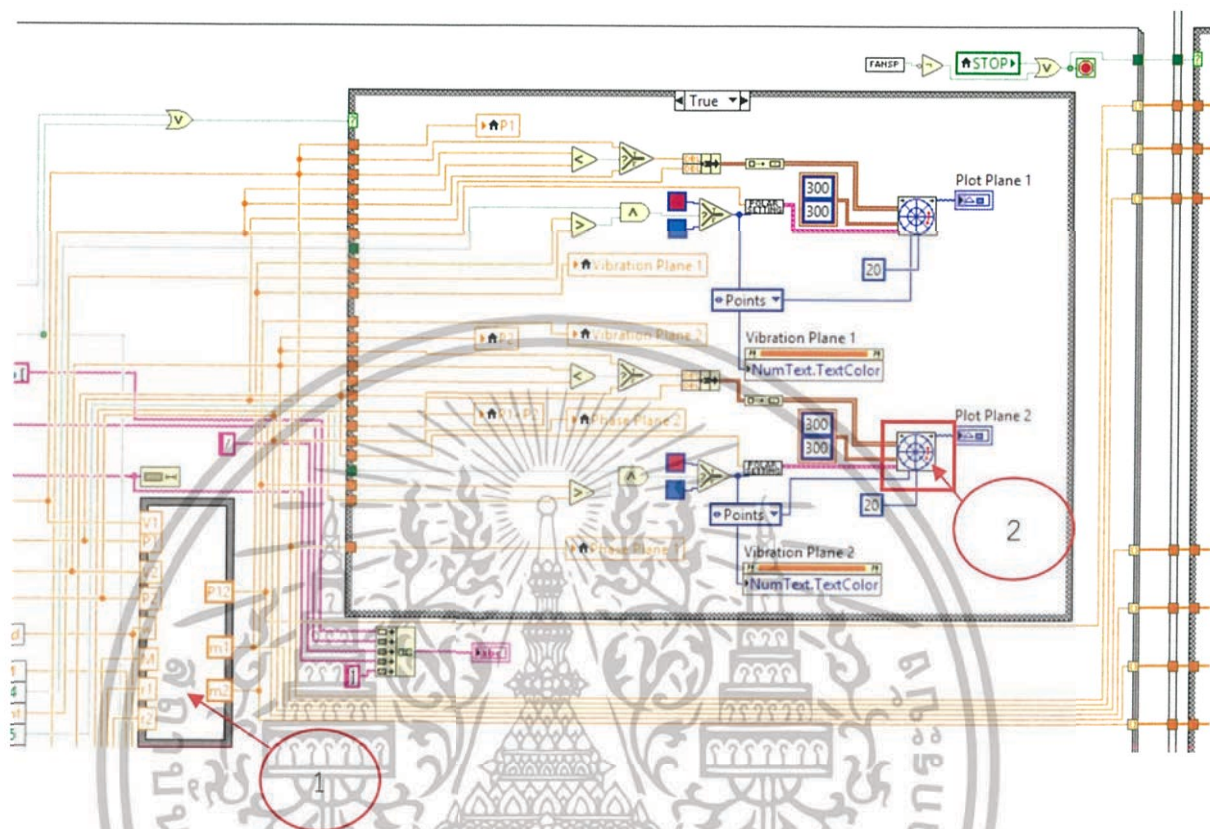
จากรูปที่ 3.21 แสดง Block Diagram โปรแกรมส่วนการเตรียมตัวแปรต้นเพื่อให้พร้อมถูกใช้งาน จะมีการนำค่าจาก Global Variable 3 ค่า ดังต่อไปนี้

- 5) SetupValue เป็นตัวแปรที่บรรจุค่าการตั้งค่าที่เก็บไว้ในหน้าต่างตั้งค่าของโปรแกรม โดยจะสามารถแยกออก เป็นค่ามาตรฐาน จำนวนครั้งการทดสอบ และอื่นๆ
- 6) PolarPlot เป็นตัวแปรที่บรรจุค่าที่ได้จากตัวรับรู้ ซึ่งผ่านการวัดความถี่และขนาดของความเร่ง ซึ่งหนึ่งรอบการหมุนจะได้ข้อมูลหนึ่งชุดสำหรับการคำนวณ
- 7) FanSpeed เป็นตัวแปรที่บรรจุค่าความเร็วของพัดลมโดยหาได้จากการแยกความถี่ของ Tachometer เพื่อหาความถี่เป็นรอบ/นาที

3.6.6 ส่วน Formular Node และการแสดงผลแบบทันที (Formular Node)

ในการคำนวณของโปรแกรม LabVIEW นั้นเพื่อให้ไม่ยุ่งยากต่อการเขียนโปรแกรมโดยใช้ Block Diagram จึงกำหนดให้ใช้ Formular Node (หมายเลข 1) โดยนำสมการที่ 3.38 และ 3.39 บรรจุลงไป เพื่อใช้ในการคิดคำนวณทางคณิตศาสตร์ และ การแสดงผลในตำแหน่งที่ต้องติดตาม โดยใช้ Polar Plot Diagram. (หมายเลข 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

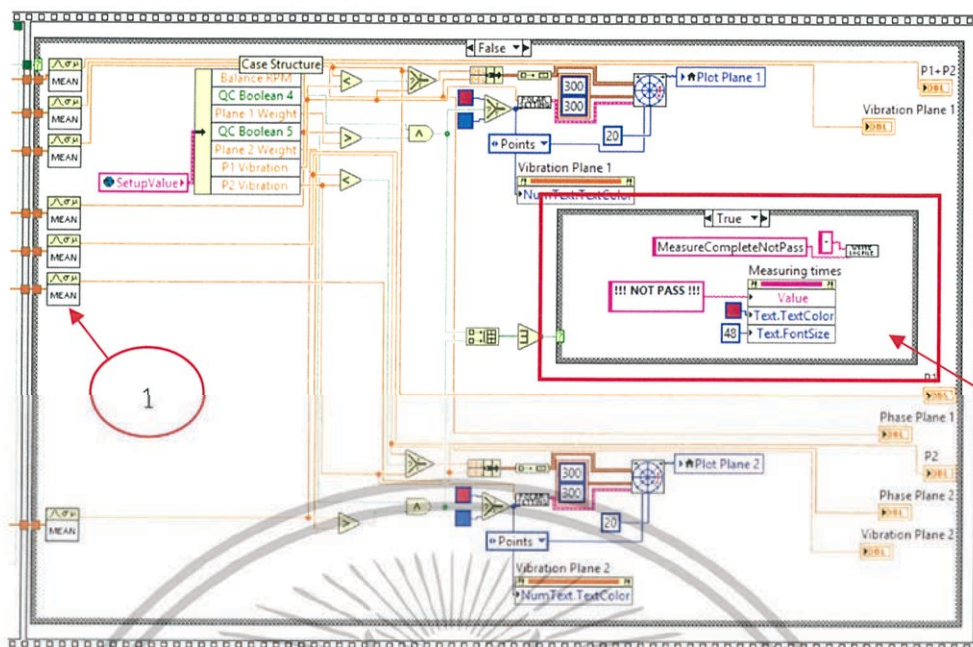


รูปที่ 3.24 Block Diagram Formutar Node และ Polar Plot

3.6.7 ส่วนการแสดงผลแบบเฉลี่ย (Monitor Meaning Result)

ในการใช้งานโปรแกรมสามารถทำการตั้งค่าว่า จะให้โปรแกรมแสดงค่าแบบทันทีในทุกรอบการวัด หรือ การแสดงผลแบบเฉลี่ยครั้งเดียวในตอนท้าย ซึ่งหากตัวเลือกดังกล่าวไปแล้วถูกตั้งค่าให้เปิดไว้ โปรแกรมจะมีลักษณะของ Block Diagram เป็นดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 Block Diagram Monitor Meaning Result

จากรูปที่ 3.23 แสดง Block Diagram ส่วนการแสดงผลแบบเฉลี่ย (Monitor Meaning Result) จะมีส่วนVI ที่สำคัญอยู่ 2 ตำแหน่งดังต่อไปนี้

- 1) Meaning Value เป็น VI ที่จะมีหน้าที่ในการรับค่าที่เป็น array และทำการหาค่าเฉลี่ยให้กับ array ชุดนั้นๆ
- 2) Pass Case Structure เป็น Case Structure ที่ถูกสร้างมาเพื่อการแสดงผลว่าวัตถุหมุนที่ทดสอบนั้นผ่านมาตรฐาน ISO - 1940 หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

เพื่อให้ชุดอุปกรณ์มีความน่าเชื่อถือและสามารถใช้งานได้จริง ผู้วิจัยจึงทำการทดลองและทดสอบความถูกต้องของซอฟต์แวร์ โดยนำค่าที่วัดได้จากการวิเคราะห์ค่าพารามิเตอร์การสั่นสะเทือนกับเครื่องมือวัดของทาง บริษัท เดลต้า อีเลคโทรนิคส์ (ประเทศไทย) จำกัด (มหาชน)

4.1 ผลการทดลองเพิ่มมวล 10 mg ถึง 50 mg

ทดลองเพิ่มมวล 5 กรณีเพื่อทดสอบแนวโน้มของความไว (Sensitivity) ของตัวรับรู้ ซึ่งมีผลต่อขนาดการสั่นซึ่งถูกวัดจากเซนเซอร์ทั้งสองระนาบ จากการเพิ่มมวลในกรณีต่างๆ บนพัลลคมตัวอย่างของทางบริษัททั้งสองเพลน จำนวนพัลลคม 5 ตัว





รูปที่ 4.1 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การเพิ่มมวลทดสอบ 5 กรัมทั้งสองระนาบ ของพัดลมหมายเลข 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



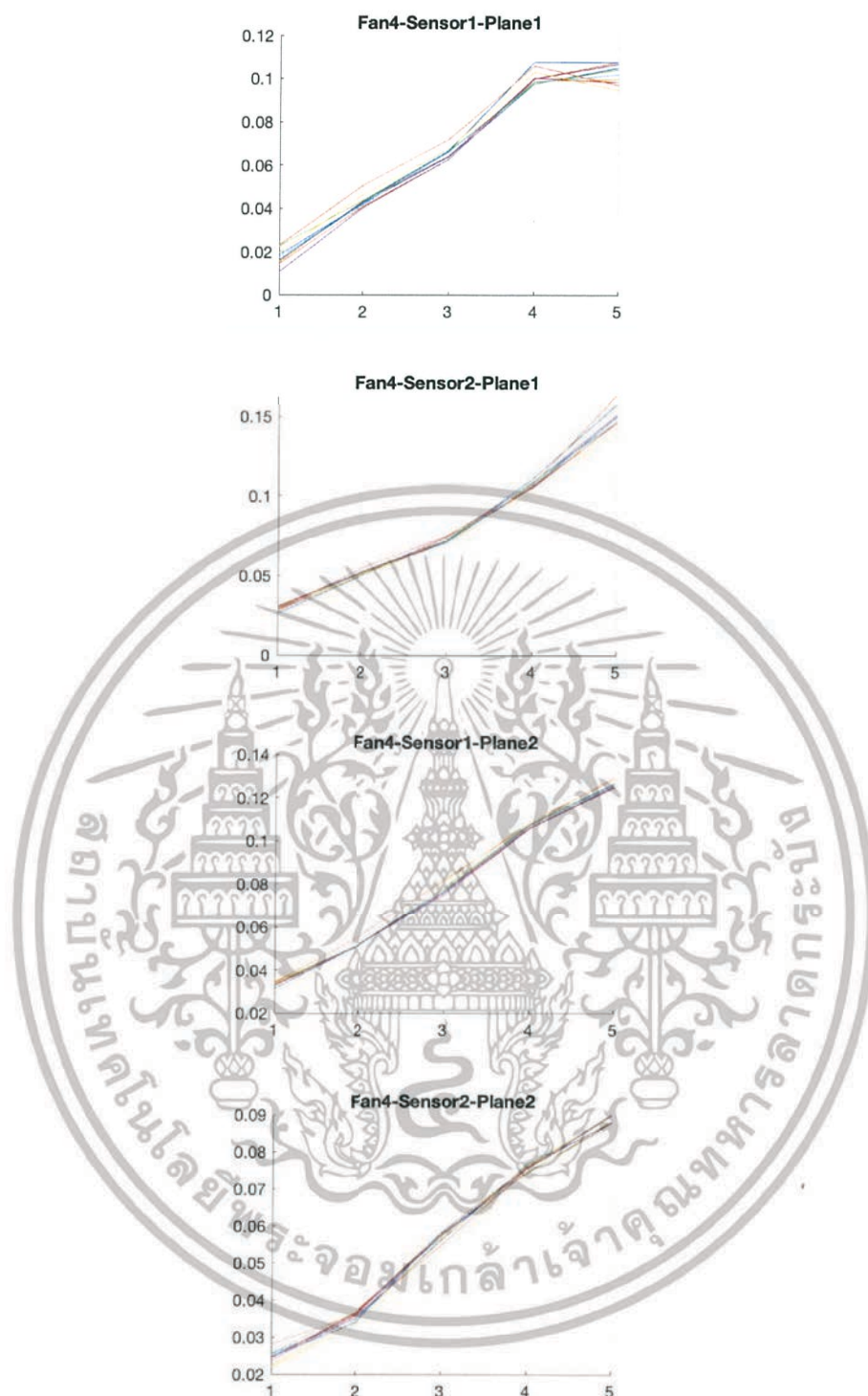
รูปที่ 4.3 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



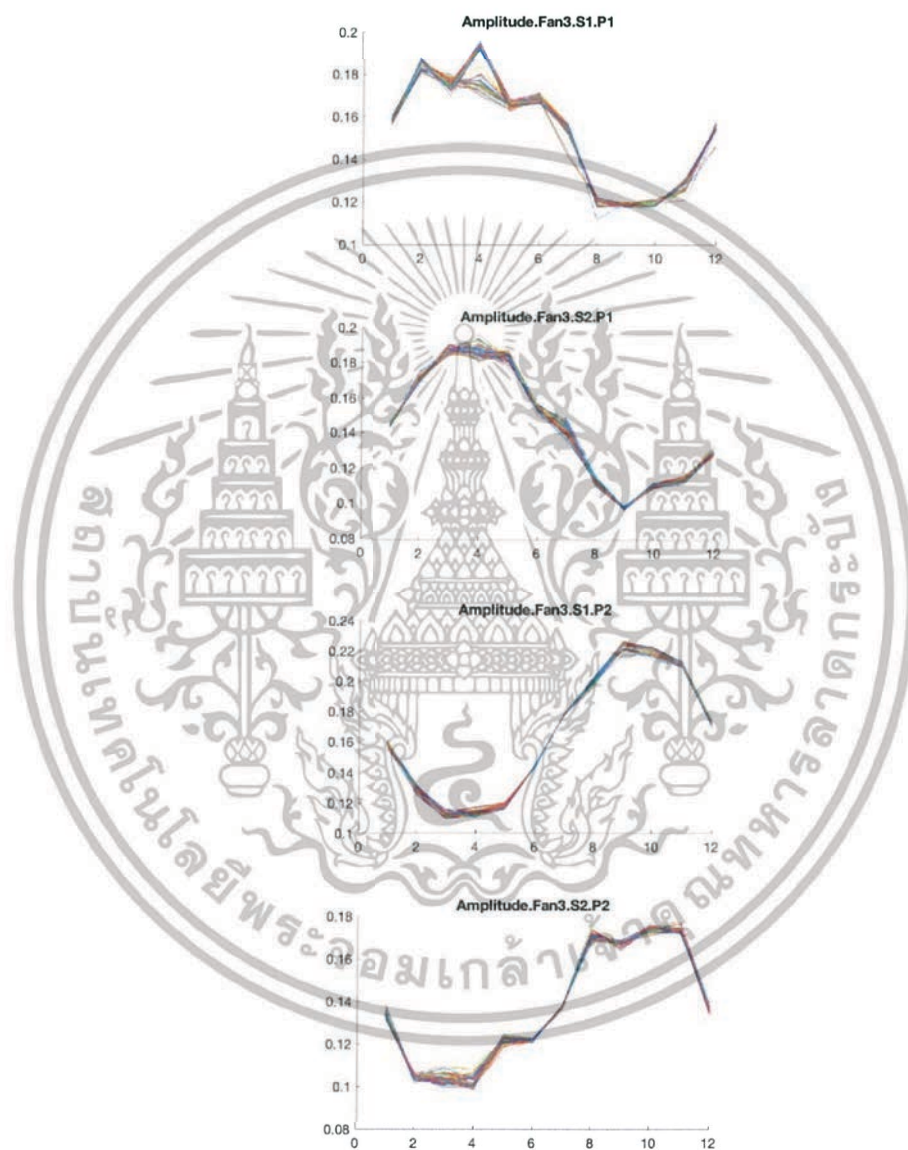
รูปที่ 4.5 การเพิ่มมวลทดสอบ 5 กรณีทั้งสองระนาบ ของพัดลมหมายเลข 4

จากรูปที่ 4.1, รูปที่ 4.2, รูปที่ 4.3, รูปที่ 4.4, รูปที่ 4.5 แสดงกราฟจะเห็นได้ว่าการเพิ่มขนาดมวลที่มากขึ้น ทำให้มีผลต่อเซ็นเซอร์ ตามมวลที่เพิ่มเป็นกราฟเส้นตรง ทำให้ทราบผลของเซ็นเซอร์ที่มีต่อความไม่สมดุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

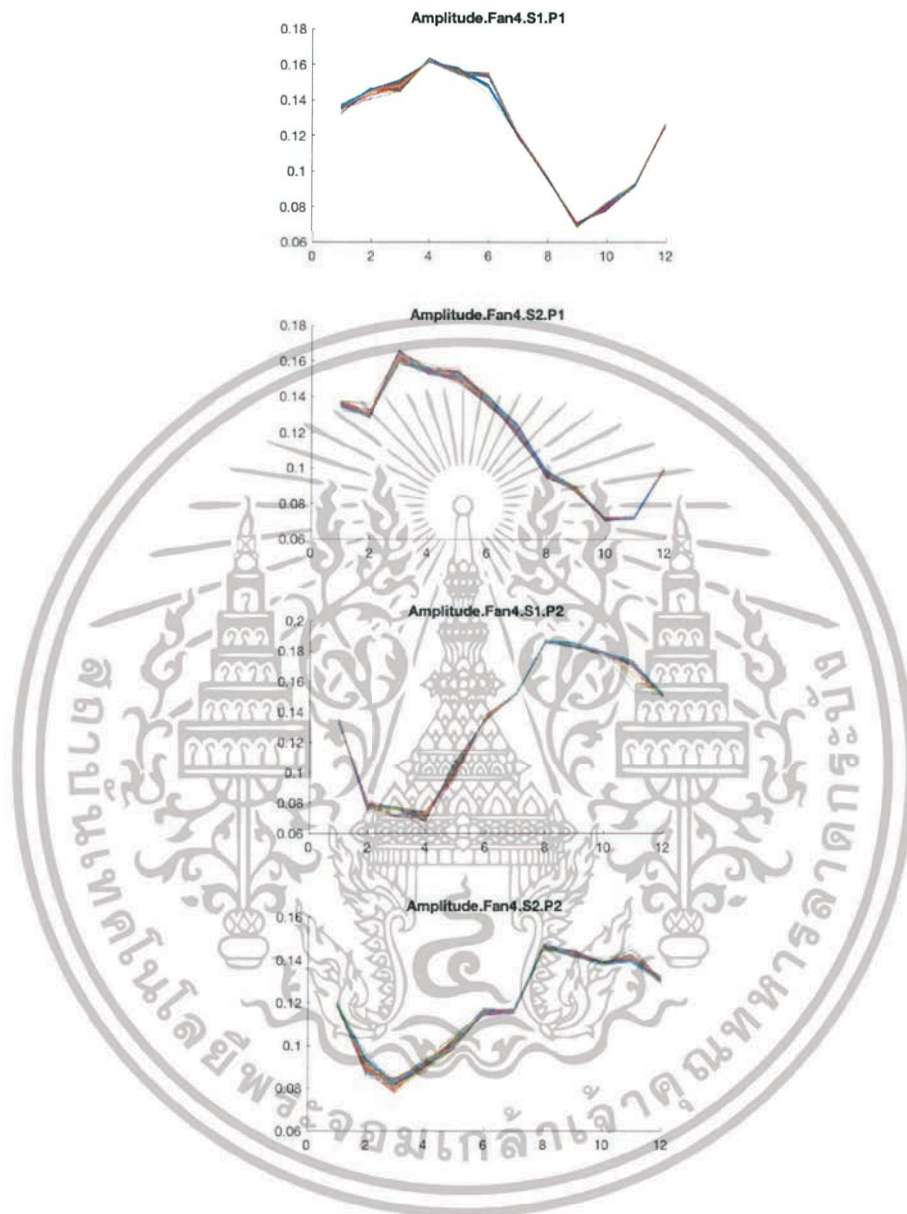
4.2 ผลการทดลองเพิ่มมวล 10 mg ในมุมที่แตกต่างกัน

ทดลองเพิ่มมวล 10mg เป็น 12 กรณี เพื่อทดสอบแนวโน้มของความไว (Sensitivity) ของตัวรับรู้ ซึ่งมีผลต่อมุมเฟสการสั่นซึ่งถูกวัดจากเซนเซอร์ทั้งสองระนาบ จากการเพิ่มมวลในกรณีต่างๆ บนพัลลมตัวอย่างของทางบริษัททั้งสองเพลน จำนวนพัลลม 2 ตัว



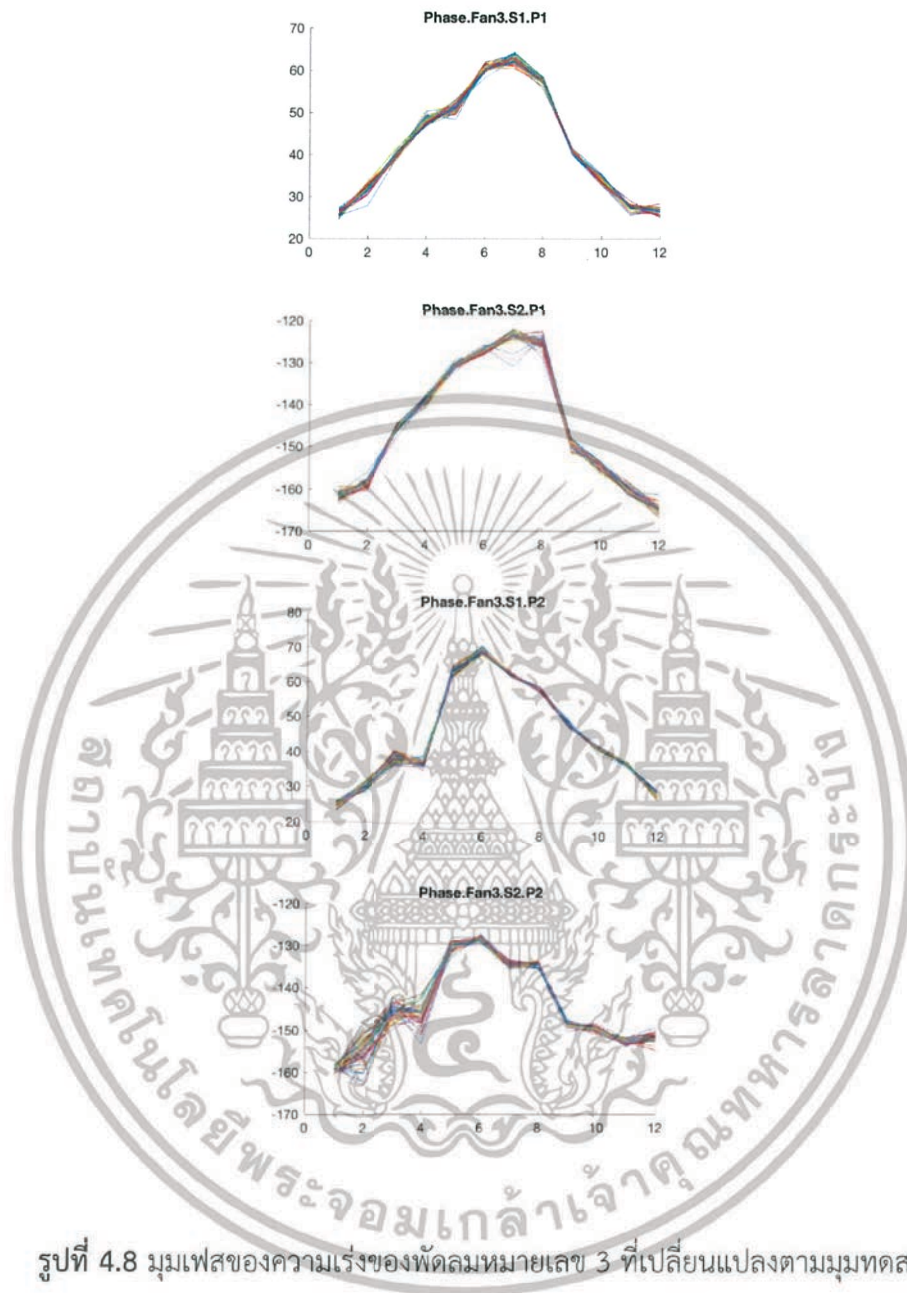
รูปที่ 4.6 ขนาดของความแรงของพัลลมหมายเลข 3 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



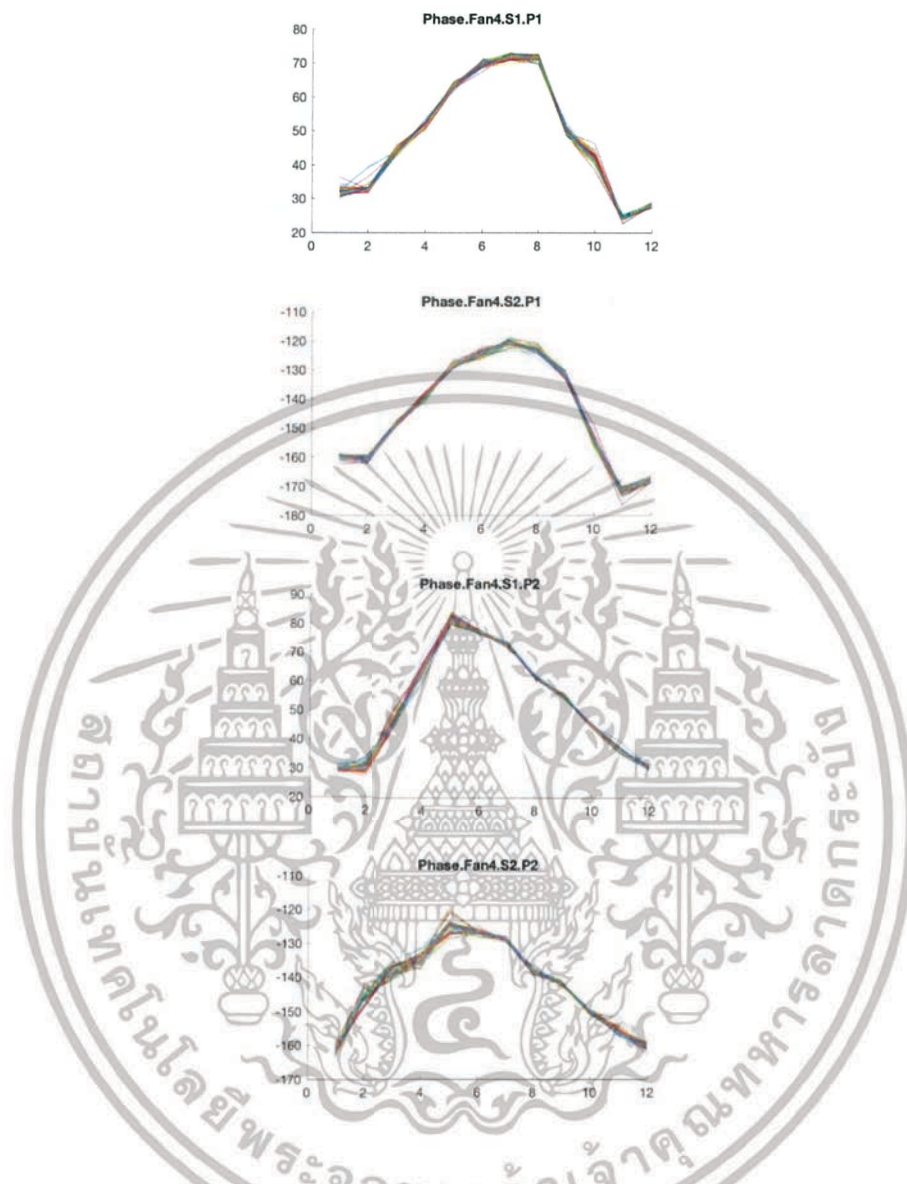
รูปที่ 4.7 ขนาดของความแรงของพัดลมหมายเลข 4 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 มุมเฟสของความเร่งของพัดลมหมายเลข 3 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ

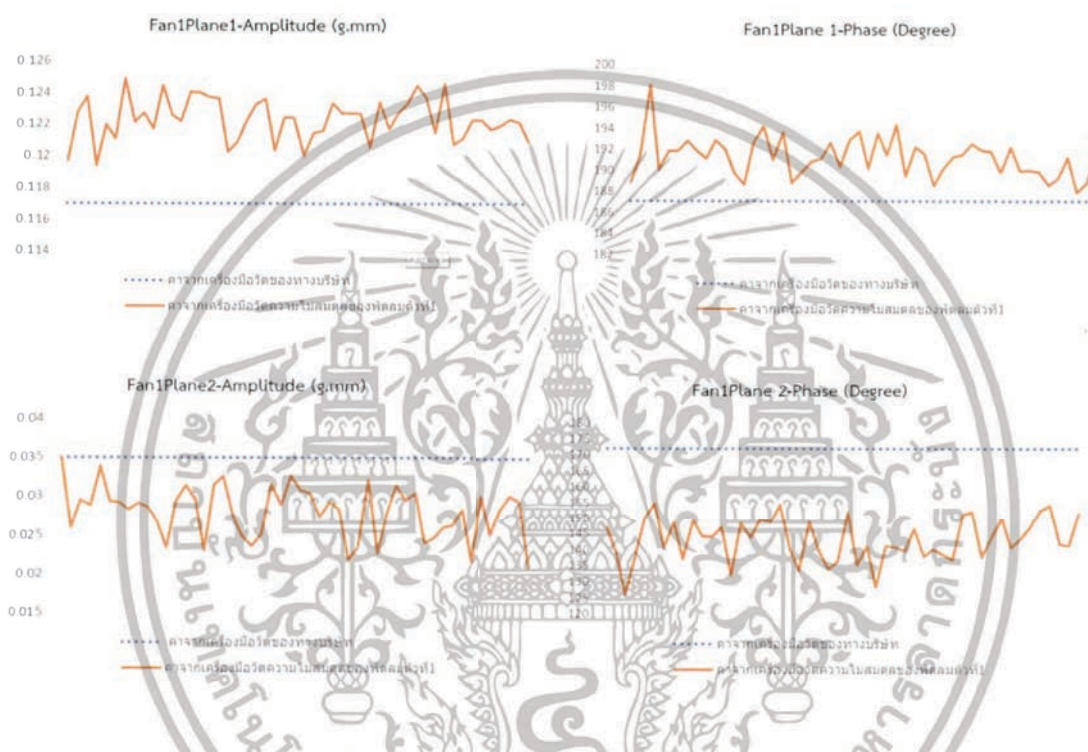
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 มุมเฟสของความแรงของพัดลมหมายเลข 4 ที่เปลี่ยนแปลงตามมุมทดสอบต่างๆ จากรูปที่ 4.6, รูปที่ 4.7, รูปที่ 4.8, รูปที่ 4.9, แสดงกราฟจะเห็นได้ว่าการเพิ่มมวลในตำแหน่งต่างๆจะทำให้มีผลต่อเซ็นเซอร์ ตามมุมตำแหน่งที่เติมลงไป ทำให้ทราบ ผลของเซ็นเซอร์ที่มีต่อความไม่สมดุล

4.3 ผลการทดสอบวัดค่าความไม่สมดุลจากพัดลมตัวที่ 1 กับเครื่องมือวัดเดิมของทางบริษัท

ผลการทดลองเปรียบเทียบกับเครื่องมือวัดของทางบริษัทที่ความ 65 Hz จำนวน 50 ครั้ง ที่ Sampling rate 102.4kS/sec



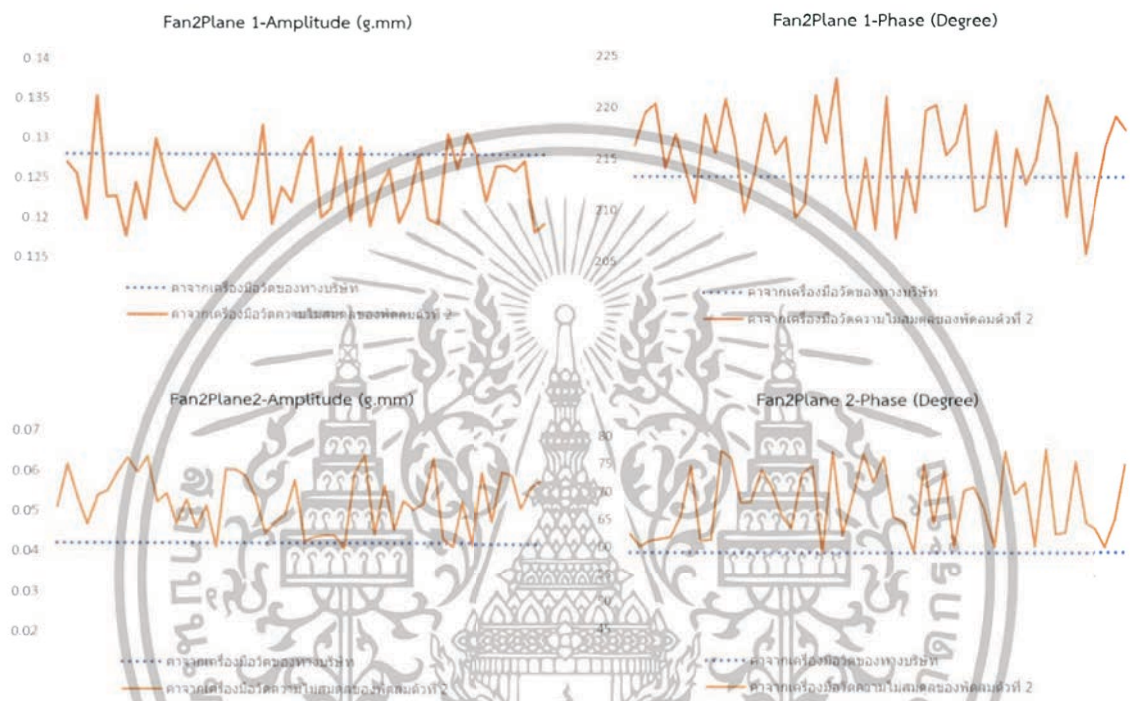
รูปที่ 4.10 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 1 กับเครื่องมือวัดเดิมของทางบริษัท

ผลจากการอ่านเก็บค่าจากพัดลมตัวที่ 1 แอมพลิจูด-เฟส ที่ระนาบที่ 1 จะใกล้เคียงกับค่าจริงมากกว่าระนาบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบวัดค่าความไม่สมดุลจากพัดลมตัวที่ 2 กับเครื่องมือวัดเดิมของทางบริษัท

ผลการทดลองเปรียบเทียบกับเครื่องมือวัดของทางบริษัทที่ความ 65 Hz จำนวน 50 ครั้ง ที่ Sampling rate 102.4kS/sec

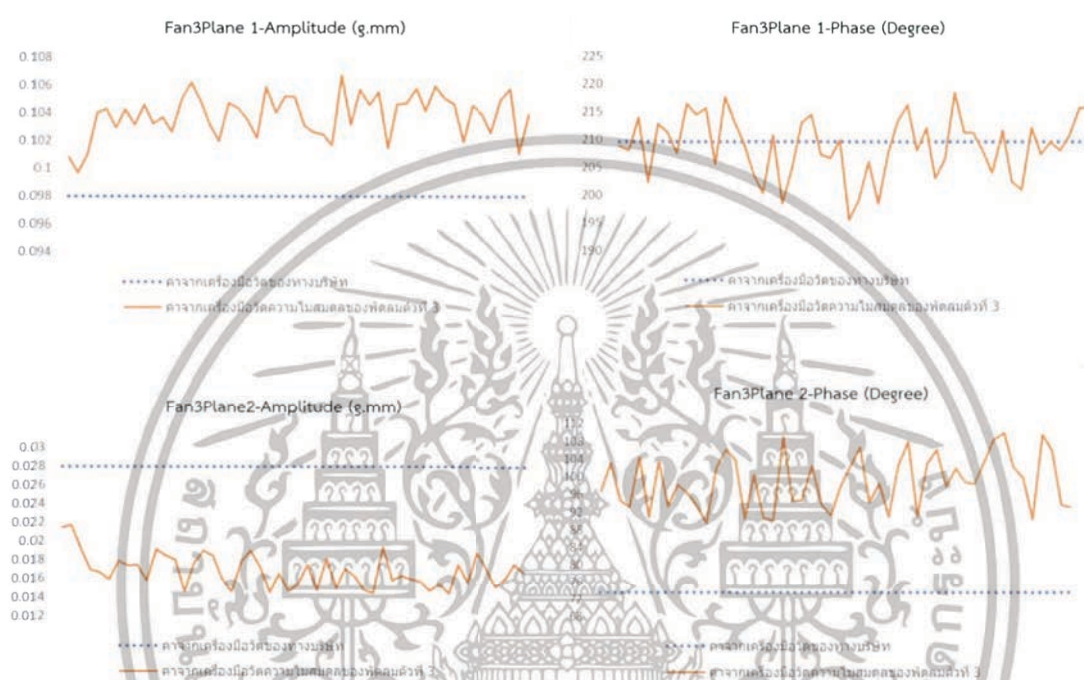


รูปที่ 4.11 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 2 กับเครื่องมือวัดเดิมของทางบริษัท

ผลจากการอ่านเก็บค่าจากพัดลมตัวที่ 2 แอมพลิจูด-เฟส ที่ระนาบที่ 1 จะใกล้เคียงกับค่าจริงมากกว่าระนาบที่ 2

4.5 ผลการทดสอบวัดค่าความไม่สมดุลจากพัดลมตัวที่ 3 กับเครื่องมือวัดเดิมของทางบริษัท

ผลการทดลองเปรียบเทียบกับเครื่องมือวัดของทางบริษัทที่ความ 65 Hz จำนวน 50 ครั้ง ที่ Sampling rate 102.4kS/sec



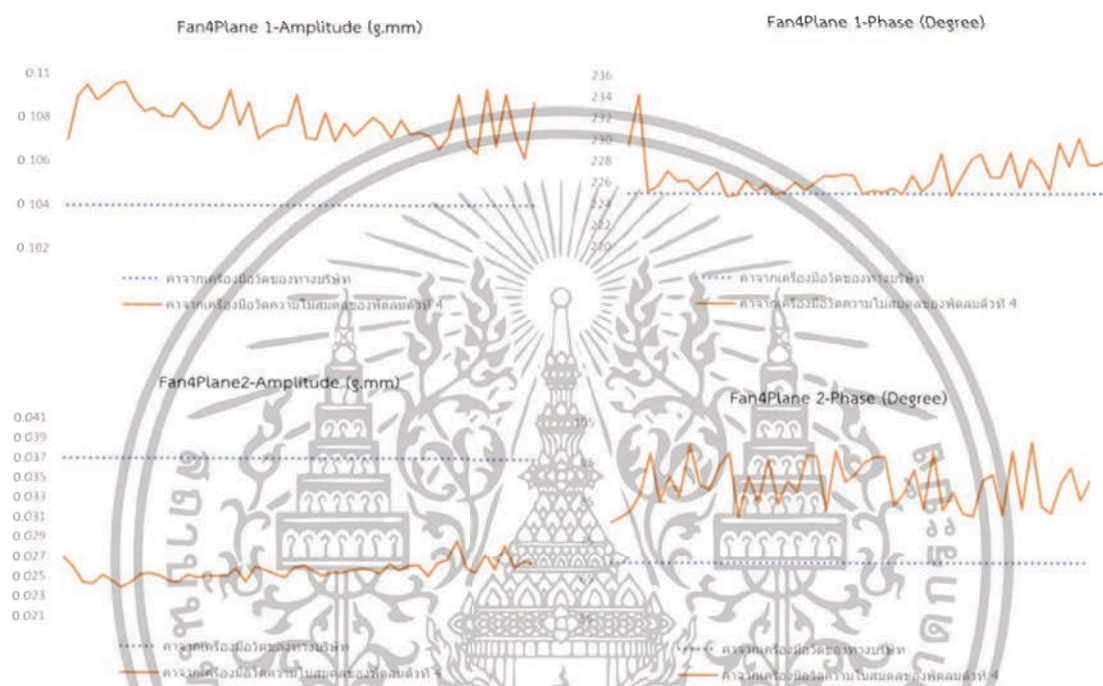
รูปที่ 4.12 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 3 กับเครื่องมือวัดเดิมของทางบริษัท

ผลจากการอ่านเก็บค่าจากพัดลมตัวที่ 3 แอมพลิจูด-เฟส ที่ระนาบที่ 1 จะใกล้เคียงกับค่าจริงมากกว่าระนาบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดสอบวัดค่าความไม่สมดุลจากพัดลมตัวที่ 1 กับเครื่องมือวัดเดิมของทางบริษัท

ผลการทดลองเปรียบเทียบกับเครื่องมือวัดของทางบริษัทที่ความ 65 Hz จำนวน 50 ครั้ง ที่ Sampling rate 102.4kS/sec



รูปที่ 4.13 กราฟแสดงค่าความไม่สมดุลจากพัดลมตัวที่ 4 กับเครื่องมือวัดเดิมของทางบริษัท

ผลจากการอ่านเก็บค่าจากพัดลมตัวที่ 4 แอมพลิจูด-เฟส ที่ระนาบที่ 1 จะใกล้เคียงกับค่าจริงมากกว่าระนาบที่ 2

4.7 ผลการทดสอบวัดค่าความไม่สมดุลจากพัดลมทั้ง 4 ตัวกับเครื่องมือวัดเดิมของบริษัท

| Fan no. | unbalance plane 1 | | | Phase plane 1 | | |
|---------|-------------------|------------|------------|---------------|------------|------------|
| | Delta | software | error(%) | Delta | software | error(%) |
| Fan1 | 0.117 | 0.12232601 | 4.5521444 | 187.10 | 191.25195 | 2.2191073 |
| Fan2 | 0.128 | 0.1241472 | -3.0099977 | 213.20 | 214.944718 | 0.81834797 |
| Fan3 | 0.098 | 0.10384739 | 5.9667256 | 209.70 | 208.846267 | -0.407121 |
| Fan4 | 0.104 | 0.10789556 | 3.74573313 | 225.00 | 226.676627 | 0.74516746 |

รูปที่ 4.14 แสดงค่าความไม่สมดุลของพัดลมทั้ง 4 ตัว ในระนาบที่ 1

| Fan no. | unbalance plane2 | | | phase plane 2 | | |
|---------|------------------|------------|------------|---------------|------------|------------|
| | Delta | software | error(%) | Delta | software | error(%) |
| Fan1 | 0.035 | 0.02779437 | -20.587523 | 172.2 | 143.26581 | -16.802665 |
| Fan2 | 0.042 | 0.05191272 | 23.6017028 | 59.10 | 67.6427598 | 14.4547543 |
| Fan3 | 0.028 | 0.01678895 | -40.039466 | 73.60 | 98.794628 | 34.2318315 |
| Fan4 | 0.037 | 0.02561665 | -30.765799 | 69.30 | 89.3303742 | 28.9038588 |

รูปที่ 4.15 แสดงค่าความไม่สมดุลของพัดลมทั้ง 4 ตัว ในระนาบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 บทนำ

ในวิทยานิพนธ์ฉบับนี้ ได้นำเสนอวิธีการวัดค่าความไม่สมดุลในพัดลมแบบสองระนาบโดยใช้โปรแกรมแลปวิวในการออกแบบ ถือเป็นงานนำเอาโปรแกรมที่ง่ายต่อการแก้ไขและพัฒนาเพื่อวัดและแสดงค่าความไม่สมดุลบนหน้าจอคอมพิวเตอร์ โดยอนาคตสามารถพัฒนาให้สามารถวัดและแสดงค่าจำนวนมากกว่าสองพัดลมต่อพัดลมหนึ่งเครื่อง ซึ่งจะเป็นการลดค่าใช้จ่ายของบริษัทในการพัฒนาซอฟต์แวร์

5.2 สรุปผลการทดลอง

1. จากการทดลองสามารถเห็นของความไม่สมดุลในพัดลม หรือ ระบบสมดุลลิจโรเตอร์ (rigid rotor) ในรูปแบบของความเร่งที่ส่งผลไปยังตัวรับรู้ทั้งสองตำแหน่ง
2. จากการศึกษาสามารถพัฒนาโปรแกรมคอมพิวเตอร์การวิเคราะห์การปรับเปลี่ยนสมดุสองระนาบของพัดลมระบายอากาศของเก้าอี้รถยนต์ โดยการแสดงข้อมูลสามารถทำได้ สองรูปแบบ รูปแบบแรกคือการแสดงผลแบบทันที ซึ่งจะแสดงผลทุกครั้งที่ครบจำนวนชุดข้อมูลหนึ่งชุด
3. โปรแกรมคอมพิวเตอร์การวิเคราะห์การปรับเปลี่ยนสมดุสองระนาบของพัดลมระบายอากาศของเก้าอี้รถยนต์สามารถนำมาประยุกต์ใช้กับพัดลม และง่ายต่อการพัฒนาหรือแก้ไข เนื่องจากโปรแกรมสามารถเพิ่มฟังก์ชันได้ทันที ทำให้โปรแกรมที่ได้ออกแบบสามารถเพิ่มฟังก์ชันเพิ่มเติมได้โดยง่าย

5.3 ปัญหาจากการศึกษา

1. การสร้างแบบจำลองทางคณิตศาสตร์ให้มีความแม่นยำนั้นไม่สามารถทำได้ภายใต้การตั้งสมการความสัมพันธ์จากความเร่งได้โดยตรงจำเป็นต้องใช้สมการที่มีความซับซ้อนมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เนื่องจากเป็นการศึกษาโดยมีความร่วมมือจากบริษัท Delta Electronics ทำให้ขั้นตอนการสั่งซื้อวัสดุอุปกรณ์จำเป็นต้องเผื่อเวลามากขึ้น
3. การนำพัสดุจากบริษัทฯ เข้ามาทดสอบเป็นระยะเวลาสั้นทำให้เกิด ค่า unbalance ที่เปลี่ยนแปลงไป เนื่องจากสาเหตุต่างๆเช่นการตกกระทบกับพื้น
4. การวัดความไม่สมดุลในปกติ บริษัทจะทำการวัดพัสดุที่เพิ่งทำการผลิต ดังนั้นในบางครั้งพัสดุที่นำมาทดสอบมีหน้าพัสดุที่ไม่สะอาดทำให้ความเร็วรอบวัดได้ไม่ตรงความเป็นจริง

5.4 ข้อเสนอแนะ

เนื่องจากการวัดค่าความไม่สมดุลโดยใช้โปรแกรม Labview ในการคำนวณ ได้ผลที่มีความคลาดเคลื่อนมากกว่าจะนำไปใช้ในภาคอุตสาหกรรมได้ ทางผู้วิจัยจึงแนะนำให้ทำการศึกษาทฤษฎีการสั่นและการแกว่งการทางคณิตศาสตร์ที่มีความซับซ้อนมากกว่าระดับปริญญาตรี จึงสามารถสร้าง Math model ที่สามารถแสดงค่าความไม่สมดุลทั้งสองระนาบได้อย่างแม่นยำมากขึ้น โดยสามารถพัฒนาจากโปรแกรมเวอร์ชันแรกนี้ ได้โดยง่าย



เอกสารอ้างอิง

- [1] R. C. Hibbeler, Kai Beng Yap. 2013. Mechanics for Engineers DYNAMICS. Jurong Singapore : Pearson
- [2] รศ. มนตรี พิรุณเกษตร. การสั่นสะเทือนทางกล (Mechanical Vibration) : 2005
- [3] นัฐพล ทรงพัฒน์โยธิน,ปกรณ พรติเม. การปรับสมดุลวัตถุหมุนเกร็ง.สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง : 2552
- [4] ผศ.ดร. มนต์ศักดิ์ พิมสาร. เอกสารประกอบการสอนวิชา กลศาสตร์เครื่องจักรกล (Mechanical of Machinery. ภาควิชาวิศวกรรมเครื่องกล สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง : 2015
- [5] Wang Qiuxiao, Wang Fei. 2014. A new vibration mechanism of balancing machine for satellite-borne spinning rotors. Beihang Chinese : Chinese Journal of Aeronautics
- [6] Victor Wowk ,Mechinery Vibration Balancing,McGraw-Hill,Inc.,1994
- [7] ABRO, Lecture Notes for Balancing Theory and Vibration Analysis” Access 28 November 2018. <http://abrobalancing.com>
- [8] Chulalongkorn university, vibration monitoring and analysis for rotating machines. Access 11 November 2018, <http://pioneer.netsew.chula.ac.th>
- [9] Brüel & KJAER ,Static and Dynamic Balancing. Access 12 November 2018, <https://www.bksv.com/>
- [10] วราคม เน็ดน้อย, พรชัย นิเวศน์รังสรรค์. เครื่องถ่วงสมดุลด้วยโปรแกรมLabVIEW พ.ศ.2552. มหาวิทยาลัยเทคโนโลยีมหานคร : 2552
- [11] กิจไพบูลย์ ชิวพันธุ์ศรี. 2011. LabVIEW ซอฟต์แวร์เพื่อการพัฒนาาระบบการวัดและควบคุม.
- [12] Balance Quality Requirements of Rigid Rotors The Practical Application of ISO 1940/1 : 2009



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียกแสดงผลและเปรียบเทียบข้อมูลด้วย MATLAB

```

%% Set
tic;
DataVersion=2;
switch DataVersion
    case 1
        MassCase=[10,20,30,40,50];
        FanNumList={'Fan0','Fan1','Fan2','Fan3','Fan4'};
        ConditionList={'D','A1','A2','A3','A4','A5','B1','B2','B3','B4','B5'};

        ConditionName={'Default','10mgPlane1','20mgPlane1','30mgPlane1','40mgPlane1','50mgPlane1','10mgPlane2','20mgPlane2','30mgPlane2','40mgPlane2','50mgPlane2'};

    case 2
        MassCase=[5,10,15,20,25];
        FanNumList={'Fan1','Fan2','Fan3','Fan4'};
        ConditionList={'D','A1','A2','A3','A4','A5','B1','B2','B3','B4','B5'};

        ConditionName={'Default','5mgPlane1','10mgPlane1','15mgPlane1','20mgPlane1','25mgPlane1','5mgPlane2','10mgPlane2','15mgPlane2','20mgPlane2','25mgPlane2'};

    case 3
        MassCase=[5,10,15,20,25];
        FanNumList={'Fan3','Fan4'};
        ConditionList={'D','A1','A2','A3','A4','A5','B1','B2','B3','B4','B5'};

        ConditionName={'Default','5mgPlane1','10mgPlane1','15mgPlane1','20mgPlane1','25mgPlane1','5mgPlane2','10mgPlane2','15mgPlane2','20mgPlane2','25mgPlane2'};

    case 4
        FanNumList={'Fan3','Fan4'};
        ConditionList={'D','A1','A2','A3','A4','A5','A6','A7','A8','A9','A10','A11','A12','B1','B2','B3','B4','B5','B6','B7','B8','B9','B10','B11','B12'};

        ConditionName={'Default','15DegPlane1','45DegPlane1','75DegPlane1','105DegPlane1','135DegPlane1','165DegPlane1','195DegPlane1','225DegPlane1','255DegPlane1','285DegPlane1','315DegPlane1','345DegPlane1','0DegPlane2','30DegPlane2','60DegPlane2','90DegPlane2','120DegPlane2','150DegPlane2','180DegPlane2','210DegPlane2','240DegPlane2','270DegPlane2','300DegPlane2','330DegPlane2'};

    case 5
        FanNumList={'Fan3','Fan4'};
        ConditionList={'D','A1','A2','A3','A4','A5','A6','A7','A8','A9','A10','A11','A12','B1','B2','B3','B4','B5','B6','B7','B8','B9','B10','B11','B12'};

        ConditionName={'Default','15DegPlane1','45DegPlane1','75DegPlane1','105DegPlane1','135DegPlane1','165DegPlane1','195DegPlane1','225DegPlane1','255DegPlane1','285DegPlane1','315DegPlane1','345DegPlane1','0DegPlane2','30DegPlane2','60DegPlane2','90DegPlane2','120DegPlane2','150DegPlane2','180DegPlane2','210DegPlane2','240DegPlane2','270DegPlane2','300DegPlane2','330DegPlane2'};

end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FanAmount=size(FanNumList,2);
AddmassPhase1=255;
AddmassPhase2=252;
ConditionAmount=size(ConditionList,2);
%% Import Data Script

if exist(['FanData_V' num2str(DataVersion) '.mat'],'File') == 2
    if exist('FanData','var')~=1
        load(['FanData_V' num2str(DataVersion) '.mat']);
    end
else
    for Condition=1:size(ConditionList,2)
        eval(['FanData_V' num2str(DataVersion) '(Condition).ConditionName=ConditionName(Condition);']);
    end
    for FanNum=1:FanAmount
        for Condition=1:size(ConditionList,2)

            FolderName=[FanNumList{FanNum} ' ' ConditionList{Condition}];
            PathP='./FanData_V'+num2str(DataVersion)+'/'+FolderName+'/Log102k_Prop.csv';
            % Path0='./FanData_V'+num2str(DataVersion)+'/'+FolderName+'/Log102k_ch0.csv';
            % Path1='./FanData_V'+num2str(DataVersion)+'/'+FolderName+'/Log102k_ch1.csv';
            % Path2='./FanData_V'+num2str(DataVersion)+'/'+FolderName+'/Log102k_ch2.csv';

            Data_prop=importdata(PathP);
            % Data_CH0=importdata(Path0);
            % Data_CH1=importdata(Path1);
            % Data_CH2=importdata(Path2);

            for DataSet=1:size(Data_prop,2)
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum}
(DataSet).TachroFreq=Data_prop(1,DataSet);]);
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum}
(DataSet).TachroPhase=Data_prop(2,DataSet);]);
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum} '(DataSet).Accel1=Data_prop(3,DataSet);]);
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum}
(DataSet).Accel1DiffPhase=Data_prop(4,DataSet);]);
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum} '(DataSet).Accel2=Data_prop(5,DataSet);]);
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum}
(DataSet).Accel2DiffPhase=Data_prop(6,DataSet);]);
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum}
(DataSet).Accel1Phase=Data_prop(7,DataSet);]);
                eval(['FanData_V' num2str(DataVersion) '(Condition)_' FanNumList{FanNum}
(DataSet).Accel2Phase=Data_prop(8,DataSet);]);
                Accel1DiffPhaseOffset(DataSet)=Data_prop(4,DataSet);
                Accel2DiffPhaseOffset(DataSet)=Data_prop(6,DataSet);
            end
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Accel1DiffPhaseOffset=median(Accel1DiffPhaseOffset);
Accel2DiffPhaseOffset=median(Accel2DiffPhaseOffset);
for DataSet=1:size(Data_prop,2)

    if Data_prop(4,DataSet)>(Accel1DiffPhaseOffset+90)
        eval(['FanData_V' num2str(DataVersion) '(Condition) ' FanNumList{FanNum}
(DataSet).Accel1DiffPhase=Data_prop(4,DataSet)-360;]);
    elseif Data_prop(4,DataSet)<(Accel1DiffPhaseOffset-90)
        eval(['FanData_V' num2str(DataVersion) '(Condition) ' FanNumList{FanNum}
(DataSet).Accel1DiffPhase=Data_prop(4,DataSet)+360;]);
    else

    end

    if Data_prop(6,DataSet)>(Accel2DiffPhaseOffset+90)
        eval(['FanData_V' num2str(DataVersion) '(Condition) ' FanNumList{FanNum}
(DataSet).Accel2DiffPhase=Data_prop(6,DataSet)-360;]);
    elseif Data_prop(6,DataSet)<(Accel2DiffPhaseOffset-90)
        eval(['FanData_V' num2str(DataVersion) '(Condition) ' FanNumList{FanNum}
(DataSet).Accel2DiffPhase=Data_prop(6,DataSet)+360;]);
    else

    end

end

% for DataSize=1:size(Data_CH0,2)
%     eval(['FanData_V' num2str(DataVersion) '(Condition) ' FanNumList{FanNum}
(DataSize).TachroVolt=Data_CH0(:,DataSize);]);
%     eval(['FanData_V' num2str(DataVersion) '(Condition) ' FanNumList{FanNum}
(DataSize).Accel1G=Data_CH1(:,DataSize);]);
%     eval(['FanData_V' num2str(DataVersion) '(Condition) ' FanNumList{FanNum}
(DataSize).Accel2G=Data_CH2(:,DataSize);]);
%     end
end
end

save(['FanData_V' num2str(DataVersion)],['FanData_V' num2str(DataVersion)]);
end

eval(['DataSize=size(FanData_V' num2str(DataVersion) '(1) ' FanNumList{1} ');2;]);
%% Generate Mass Vector
if exist(['MassVector_V' num2str(DataVersion) '.mat'],File) == 2
    if exist('MassVector','var')~=1
        load(['MassVector_V' num2str(DataVersion) '.mat']);
    end
else
    for Condition=2:size(ConditionList,2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

eval(['MassVector_V' num2str(DataVersion) '(Condition-1).ConditionName=ConditionName(Condition);]);
end
for FanNum=1:FanAmount
for Condition=2:size(ConditionList,2)
for DataSet=1:DataSetSize
eval(['A1_D=FanData_V' num2str(DataVersion) '(1).' FanNumList{FanNum} '(DataSet).Accel1;]);
eval(['P1_D=deg2rad(FanData_V' num2str(DataVersion) '(1).' FanNumList{FanNum} '(DataSet).Accel1DiffPhase;]);
eval(['A2_D=FanData_V' num2str(DataVersion) '(1).' FanNumList{FanNum} '(DataSet).Accel2;]);
eval(['P2_D=deg2rad(FanData_V' num2str(DataVersion) '(1).' FanNumList{FanNum} '(DataSet).Accel2DiffPhase;]);

eval(['A1_A=FanData_V' num2str(DataVersion) '(Condition).' FanNumList{FanNum} '(DataSet).Accel1;]);
eval(['P1_A=deg2rad(FanData_V' num2str(DataVersion) '(Condition).' FanNumList{FanNum} '(DataSet).Accel1DiffPhase;]);
eval(['A2_A=FanData_V' num2str(DataVersion) '(Condition).' FanNumList{FanNum} '(DataSet).Accel2;]);
eval(['P2_A=deg2rad(FanData_V' num2str(DataVersion) '(Condition).' FanNumList{FanNum} '(DataSet).Accel2DiffPhase;]);

eval(['MassVector_V' num2str(DataVersion) '(Condition-1).' FanNumList{FanNum} '(DataSet).Sensor1=A1_A*exp(P1_A*i)-
A1_D*exp(P1_D*i);]);
eval(['MassVector_V' num2str(DataVersion) '(Condition-1).' FanNumList{FanNum} '(DataSet).Sensor2=A2_A*exp(P2_A*i)-
A2_D*exp(P2_D*i);]);

% eval(['MassVector_V' num2str(DataVersion) '(Condition-1).' FanNumList{FanNum}
'(DataSet).Sensor1=A1_D*exp(P1_D*i);]); %A1_A*exp(P1_A*i)-
% eval(['MassVector_V' num2str(DataVersion) '(Condition-1).' FanNumList{FanNum}
'(DataSet).Sensor2=A2_D*exp(P2_D*i);]); %A2_A*exp(P2_A*i)-
%
end
end
end
save(['MassVector_V' num2str(DataVersion)], ['MassVector_V' num2str(DataVersion)]);
end
%% Compare Mass Vector Amplitude & Phase
for FanNum=1:FanAmount
for Case=1:5
for DataSet=1:DataSetSize
eval(['A.' FanNumList{FanNum} '(DataSet).A1(Case)=abs(MassVector_V' num2str(DataVersion) '(Case).' FanNumList{FanNum}
'(DataSet).Sensor1;]);
eval(['A.' FanNumList{FanNum} '(DataSet).A2(Case)=abs(MassVector_V' num2str(DataVersion) '(Case).' FanNumList{FanNum}
'(DataSet).Sensor2;]);
eval(['A.' FanNumList{FanNum} '(DataSet).A3(Case)=abs(MassVector_V' num2str(DataVersion) '(Case+5).'
FanNumList{FanNum} '(DataSet).Sensor1;]);
eval(['A.' FanNumList{FanNum} '(DataSet).A4(Case)=abs(MassVector_V' num2str(DataVersion) '(Case+5).'
FanNumList{FanNum} '(DataSet).Sensor2;]);
end
end

for DataSet=1:DataSetSize
subplot(4,FanAmount*2,FanNum*2-1)
hold on

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

eval(['plot(MassCase,A,' FanNumList{FanNum} '(DataSet).A1);']);
subplot(4,FanAmount*2,FanNum*2-1+FanAmount*2)
hold on
eval(['plot(MassCase,A,' FanNumList{FanNum} '(DataSet).A2);']);
subplot(4,FanAmount*2,FanNum*2-1+FanAmount*4)
hold on
eval(['plot(MassCase,A,' FanNumList{FanNum} '(DataSet).A3);']);
subplot(4,FanAmount*2,FanNum*2-1+FanAmount*6)
hold on
eval(['plot(MassCase,A,' FanNumList{FanNum} '(DataSet).A4);']);
end

subplot(4,FanAmount*2,FanNum*2-1)
grid on
title(['Amplitude.' FanNumList{FanNum} '.S1.P1'])
subplot(4,FanAmount*2,FanNum*2-1+FanAmount*2)
grid on
title(['Amplitude.' FanNumList{FanNum} '.S2.P1'])
subplot(4,FanAmount*2,FanNum*2-1+FanAmount*4)
grid on
title(['Amplitude.' FanNumList{FanNum} '.S1.P2'])
subplot(4,FanAmount*2,FanNum*2-1+FanAmount*5)
grid on
title(['Amplitude.' FanNumList{FanNum} '.S2.P2'])
end
for FanNum=1:FanAmount
for Case=1:5
for DataSet=1:DataSetSize
eval(['P.' FanNumList{FanNum} '(DataSet).P1(Case)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case).'
FanNumList{FanNum} '(DataSet).Sensor1));']);
eval(['P.' FanNumList{FanNum} '(DataSet).P2(Case)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case).'
FanNumList{FanNum} '(DataSet).Sensor2));']);
eval(['P.' FanNumList{FanNum} '(DataSet).P3(Case)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case+5).'
FanNumList{FanNum} '(DataSet).Sensor1));']);
eval(['P.' FanNumList{FanNum} '(DataSet).P4(Case)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case+5).'
FanNumList{FanNum} '(DataSet).Sensor2));']);

eval(['P01(DataSet)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case).' FanNumList{FanNum}
(DataSet).Sensor1));']);
eval(['P02(DataSet)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case).' FanNumList{FanNum}
(DataSet).Sensor2));']);
eval(['P03(DataSet)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case+5).' FanNumList{FanNum}
(DataSet).Sensor1));']);
eval(['P04(DataSet)=rad2deg(angle(MassVector_V' num2str(DataVersion) '(Case+5).' FanNumList{FanNum}
(DataSet).Sensor2));']);
end

P1PhaseOffset=median(P01);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

P2PhaseOffset=median(P02);
P3PhaseOffset=median(P03);
P4PhaseOffset=median(P04);

for DataSet=1:DataSetSize
    if P01(DataSet)>(P1PhaseOffset+90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P1(Case)=P01(DataSet)-360;']);
    elseif P01(DataSet)<(P1PhaseOffset-90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P1(Case)=P01(DataSet)+360;']);
    else

    end

    if P02(DataSet)>(P2PhaseOffset+90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P2(Case)=P02(DataSet)-360;']);
    elseif P02(DataSet)<(P2PhaseOffset-90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P2(Case)=P02(DataSet)+360;']);
    else

    end

    if P03(DataSet)>(P3PhaseOffset+90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P3(Case)=P03(DataSet)-360;']);
    elseif P03(DataSet)<(P3PhaseOffset-90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P3(Case)=P03(DataSet)+360;']);
    else

    end

    if P04(DataSet)>(P4PhaseOffset+90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P4(Case)=P04(DataSet)-360;']);
    elseif P04(DataSet)<(P4PhaseOffset-90)
        eval(['P.' FanNumList{FanNum} '(DataSet).P4(Case)=P04(DataSet)+360;']);
    else

    end

end
end

```

end

```

for DataSet=1:DataSetSize
    subplot(4,FanAmount*2,FanNum*2)
    hold on
    eval(['plot(MassCase,P.' FanNumList{FanNum} '(DataSet).P1;')]);
    subplot(4,FanAmount*2,FanNum*2+FanAmount*2)
    hold on
    eval(['plot(MassCase,P.' FanNumList{FanNum} '(DataSet).P2;')]);
    subplot(4,FanAmount*2,FanNum*2+FanAmount*4)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hold on
eval(['plot(MassCase,P.' FanNumList{FanNum} '(DataSet).P3;');
subplot(4,FanAmount*2,FanNum*2+FanAmount*6)
hold on
eval(['plot(MassCase,P.' FanNumList{FanNum} '(DataSet).P4;');
end

subplot(4,FanAmount*2,FanNum*2)
grid on
title(['Phase.' FanNumList{FanNum} '.S1.P1'])
subplot(4,FanAmount*2,FanNum*2+FanAmount*2)
grid on
title(['Phase.' FanNumList{FanNum} '.S2.P1'])
subplot(4,FanAmount*2,FanNum*2+FanAmount*4)
grid on
title(['Phase.' FanNumList{FanNum} '.S1.P2'])
subplot(4,FanAmount*2,FanNum*2+FanAmount*6)
grid on
title(['Phase.' FanNumList{FanNum} '.S2.P2'])
end

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้