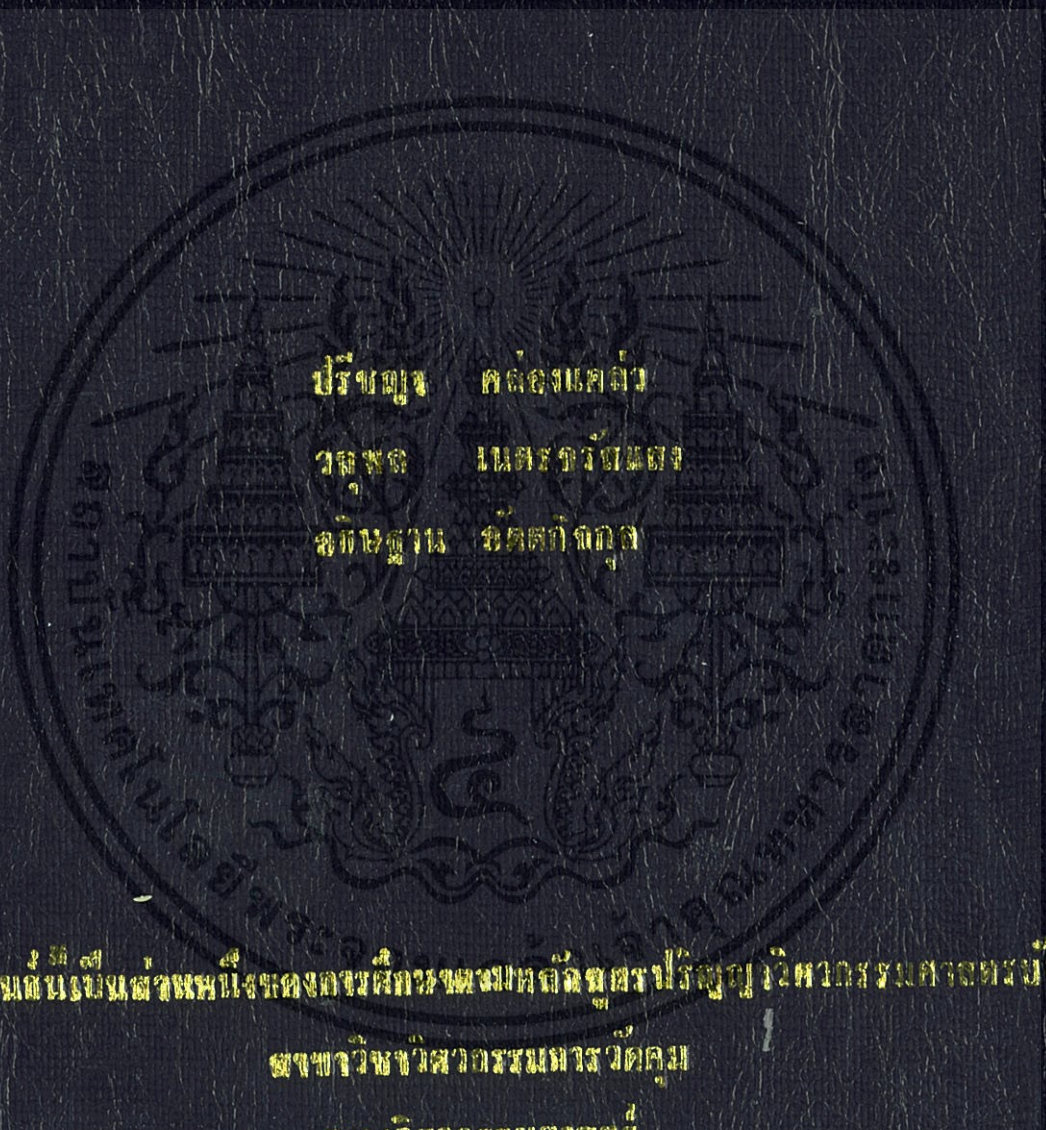
การทดสอบระบบต้นแบบสำหรับกระบวนการเผาขนมเค็มโดยใช้ LABVIEW

THE SYSTEM IDENTIFICATION FOR SMALL OVEN USING LABVIEW

ปรีชญา คล้องแคล่ว

วุฒิพล เนตรรัตนชัย

อภิษฐาน อัศตกิจกุล



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การหาเอกลักษณ์ของระบบสำหรับกระบวนการเตาอบขนาดเล็ก โดยใช้ LABVIEW

THE SYSTEM IDENTIFICATION FOR SMALL OVEN USING LABVIEW

ปรีชญา คล่องแคล่ว
วสุพล เนตรจรัสแสง
อธิษฐาน อัตตกิจกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE SYSTEM IDENTIFICATION FOR SMALL OVEN USING LABVIEW

PREECHAYA KLONGKAEW
WASUPOL NETCHARUSSAENG
ATHITTHARN ATTAKITKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERRING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG
ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ การหาเอกลักษณ์ของระบบสำหรับกระบวนการเตาอบขนาดเล็กโดยใช้ LabVIEW

THE SYSTEM IDENTIFICATION FOR SMALL OVEN USING LABVIEW

นักศึกษาผู้จัดทำ นายปรีชญา คล่องแคล่ว **รหัสนักศึกษา** 53010949
นายวสุพล เนตรจรัสแสง **รหัสนักศึกษา** 53011444
นายอธิษฐาน อัตตกิจกุล **รหัสนักศึกษา** 53011836

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2556

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์อาจินต์ น่วมสำราญ	
รองศาสตราจารย์ ดร.วิทยา ทิพย์สุวรรณพร	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การหาเอกลักษณ์ของระบบสำหรับกระบวนการเตาอบขนาดเล็กโดยใช้ LabVIEW The system identification for small oven using LabVIEW			
นักศึกษาผู้จัดทำ	นายปรีชญา	คล่องแคล่ว	รหัสนักศึกษา	53010949
	นายวสุพล	เนตรจรัสแสง	รหัสนักศึกษา	53011444
	นายอธิษฐาน	อัตตกิจกุล	รหัสนักศึกษา	53011836
อาจารย์ที่ปรึกษา	รองศาสตราจารย์อาจินต์		น่วมสำราญ	
	รองศาสตราจารย์ ดร.วิทยา		ทิพย์สุวรรณพร	
ปีการศึกษา	2556			

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอ ผลของการหาเอกลักษณ์ของระบบ (System Identification) และการควบคุมโดยศึกษาหาแบบจำลองทางคณิตศาสตร์ของระบบควบคุม โดยระบบควบคุมที่ทำการศึกษาคือกระบวนการควบคุมอุณหภูมิของเตาอบขนาดเล็ก โดยเริ่มต้นจากการศึกษาผลตอบสนองระบบควบคุมของกระบวนการเตาอบขนาดเล็ก นั้นจะทำโดยการเขียนโปรแกรมระบบแบบเปิด (Open Loop) เพื่อดูผลตอบสนองและแนวโน้มของกราฟด้วยโปรแกรม LabVIEW จากนั้นทำการหาเอกลักษณ์ของระบบ (System Identification) ด้วยวิธีการป้อนสัญญาณอินพุตแบบอิมพัลส์ (Impulse) ให้แก่ระบบและจะนำผลตอบสนองที่ได้ไปเปรียบเทียบกับผลตอบสนองจริงพบว่าผลตอบสนองทั้งสองเป็นไปตามข้อกำหนด จะได้เอกลักษณ์ของระบบ (Impulse response)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	The system identification for small oven using LabVIEW	
Authors	Mr. Preechaya	Klongkaew
	Mr. Wasupol	Netcharussaeng
	Miss Athittharn	Attakitkul
Thesis Advisor	Assoc.Prof.Arjin	Numsomran
	Assoc.Prof.Dr.Vittaya	Tipsuwanporn
Year	2013	

ABSTRACT

This thesis presents the study results of the identification and control by studying the mathematical model of control system. Beginning, observed the response of the temperature process control for the small oven. Study the response of the temperature process control for the small oven by Open Loop program for the responses and trends of the graphs with LabVIEW program after that, results of the System Identification by using impulse input signal to system which compares between actual responses

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยความกรุณาอนุเคราะห์เครื่องมือ อุปกรณ์ที่ใช้ในการดำเนินงานทดลอง ตลอดจนอบรมสั่งสอน แนะนำ และให้คำปรึกษาแนวทางในการดำเนินงานจากรองศาสตราจารย์ อาจินต์ น่วมสำราญ และรองศาสตราจารย์ ดร. วิทยา ทิพย์สุวรรณพร อาจารย์ผู้ควบคุมปริญญานิพนธ์ คณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาของท่านเป็นอย่างยิ่ง และขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ สาขาวิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆ อันเป็นประโยชน์อย่างยิ่งให้กับคณะผู้จัดทำ สามารถนำมาปรับใช้กับปริญญานิพนธ์นี้ได้เป็นอย่างดี

ขอกราบขอบพระคุณบิดา มารดา อันเป็นที่รักยิ่งของคณะผู้จัดทำทุกท่าน ที่เป็นกำลังใจสำคัญด้วยความรัก ความหวังดี และให้การสนับสนุนในทุกๆด้านเสมอ

ขอขอบคุณเพื่อนนักศึกษาทุกคน ที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

ประโยชน์ และคุณค่า รวมทั้งความดีทั้งปวง อันเกิดจากปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอมอบแด่บุพการี และผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VII
สารบัญภาพ VIII

บทที่ 1 บทนำ 1
1.1 ความสำคัญและที่มาของปริญญานิพนธ์ 1
1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1
1.3 ขอบเขตของปริญญานิพนธ์ 1
1.4 ขั้นตอนการศึกษา 2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากปริญญานิพนธ์ 2

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 3
2.1 กล่าวนำ 3
2.2 โมเดลหรือแบบจำลอง 3
2.2.1 แนวทางการสร้างแบบจำลองหรือโมเดล 3
2.2.2 ประเภทของโมเดลหรือแบบจำลอง 3
2.2.2.1 โมเดลเชิงแนวคิด (Conceptual Model) 4
2.2.2.2 โมเดลเชิงกายภาพ (Physical Model) 5
2.2.2.3 โมเดลเชิงคณิตศาสตร์และสถิติ (Mathematical and Static model) 5
2.2.2.4 โมเดลเชิงภาพเคลื่อนไหว (Visualization models) 7
2.2.2.5 โมเดลเชิงซิสเต็มไดนามิคส์ (System Dynamics model) 8
2.2.3 การทำงานของโมเดลสามารถอธิบายด้วยผังภาพ 3 ลักษณะ 8
2.3 โครงสร้างโมเดล (Model structure) 10
2.4 การสร้างแบบจำลองระบบไดนามิกส์ (Modeling a Dynamic system) 10
2.5 เอกลักษณ์ของระบบ (System Identification) 11
2.6 System Identification Procedure 12
2.7 ชนิดของ System Identification Method 12
2.8 Non-Parametric Methods 12
2.8.1 Step Response 12
2.8.2 Impulse Response 13

เอกสารนี้เป็นเอกสาร... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น...

สารบัญ (ต่อ)

หน้า

บทที่ 3 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์.....15

3.1 บทนำ..... 15

3.2 องค์ประกอบของฮาร์ดแวร์ 15

3.2.1 เตาอบขนาดเล็ก 15

3.2.2 RTD ชนิด PT100 16

3.2.3 RTD Transmitter : RT 95..... 18

3.2.4 Signal Converter : Model TW-4M-1-N 18

3.2.5 Solid-State Relay : SSR RM1E23AA25 19

3.2.6 Data Acquisition : DAQ-6008 20

3.3 โปรแกรมที่เกี่ยวข้องกับกระบวนการ 21

3.3.1 โปรแกรม LabVIEW 2011 21

3.3.1.1 ขั้นตอนการติดตั้งโปรแกรม LabVIEW 22

3.3.1.2 การใช้งานโปรแกรม LabVIEW 27

3.3.2 พื้นฐานการเขียนโปรแกรมภาษากราฟฟิก.....43

3.3.3 การสร้าง VI ด้วย VI แบบเร็ว (Express VI) 50

3.3.4 การแก้ไขและการดีบัคโปรแกรม 53

3.3.5 รูปแบบโครงสร้างการควบคุมการทำงานของโปรแกรม 59

3.3.6 พื้นฐานการใช้งานอาร์เรย์.....66

3.3.7 การสร้างอาร์เรย์ด้วยลูป.....69

3.3.8 ฟังก์ชันของอาร์เรย์.....71

3.3.9 พื้นฐานการใช้งานคลัสเตอร์ 75

3.3.10 พื้นฐานการใช้งาน Charts และ Graph 79

3.3.11 พื้นฐานเกี่ยวกับ String 82

3.3.12 การหาเอกลักษณ์ของกระบวนการโดย System identification toolkit.....84

3.3.12.1 SI Remove Trend VI..... 84

3.3.12.2 SI Estimate Transfer Function Model VI..... 85

3.3.12.3 SI Estimate Impulse Response VI..... 87

3.3.12.4 SI Model Simulation VI..... 89

3.4 สรุป..... 90

บทที่ 4 การทดลองและผลการทดลอง.....91

4.1 กล่าวนำ..... 91

4.2 กระบวนการที่ใช้ในการทดลอง..... 91

4.3 การจำลองกระบวนการ..... 92

4.4 กระบวนการหาเอกลักษณ์ของระบบ..... 96

4.5 ผลการทดลอง..... 97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.6 สรุป.....102

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ 103

5.1 สรุปผล..... 103

5.2 ปัญหาและอุปสรรค..... 103

5.3 ข้อเสนอแนะ..... 103

บรรณนานุกรม.....104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 ข้อมูลทางเทคนิคของเตาอบขนาดเล็ก 16

3.2 ข้อมูลทางเทคนิคของ RTD ชนิด PT100 17

3.3 ข้อมูลทางเทคนิคของ RT 95 18

3.4 ข้อมูลทางเทคนิคของ SSR 19

4.1 แสดงค่าของ Estimated Model และแสดงค่าความผิดพลาด 102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 แสดงตัวอย่างโมเดลเชิงแนวคิด....4
2.2 แสดงตัวอย่างโมเดลเชิงแนวคิด....4
2.3 แสดงตัวอย่างโมเดลเชิงกายภาพแบบใช้มาตราส่วนย่อขนาดจากจริง กรณีของจริงมีขนาดใหญ่....4
2.4 แสดงตัวอย่างโมเดลเชิงกายภาพแบบใช้มาตราส่วนย่อขนาดจากจริง กรณีของจริงมีขนาดเล็ก....5
2.5 แสดงตัวอย่างโมเดลเชิงกายภาพ....5
2.6 แสดงตัวอย่างโมเดลทางคณิตศาสตร์....6
2.7 แสดงตัวอย่างโมเดลทางสถิติ....6
2.8 แสดงตัวอย่างโมเดลเชิงภาพเคลื่อนไหว....7
2.9 แสดงตัวอย่างโมเดลเชิงภาพเคลื่อนไหว....7
2.10 วงจรย้อนกลับของระบบประชากร....9
2.11 ผังการไหลของระบบประชากร....9
2.12 วงจรย้อนกลับของระบบความจน....10
2.13 ผังรูปตาวัวของระบบความจน....10
2.14 แสดง Dynamic Model....11
2.15 แสดงแผนผังการหา System Identification....11
3.1 แสดงกระบวนการเตาอบขนาดเล็ก....15
3.2 ลักษณะของ RTD ชนิด PT100....17
3.3 ลักษณะของวงจรภายใน RT 95....18
3.4 ลักษณะของ Signal Converter....19
3.5 ลักษณะของ Solid-State Relay....19
3.6 ลักษณะของ DAQ-6008....21
3.7 แสดงหน้าต่าง Extract files ของโปรแกรม LabVIEW 2011....22
3.8 แสดงหน้าต่าง Extract files สำเร็จ....22
3.9 แสดงหน้าต่างเริ่มการติดตั้งโปรแกรม LabVIEW 2011....23
3.10 แสดงหน้าต่างการกรอกข้อมูล Full Name และ Organization....23
3.11 แสดงหน้าต่างการกรอกข้อมูล Serial Number....23
3.12 แสดงหน้าต่างการเลือกที่ติดตั้งโปรแกรม LabVIEW....24
3.13 แสดงหน้าต่างการเลือก Features ของโปรแกรม LabVIEW....24
3.14 แสดงหน้าต่างความต้องการทราบถึงการอัพเดทผลิตภัณฑ์ของ NI....24
3.15 แสดงหน้าต่างเงื่อนไขของลิขสิทธิ์....25
3.16 แสดงหน้าต่างเงื่อนไขของลิขสิทธิ์....25
3.17 แสดงหน้าต่างการแจ้งเริ่มการติดตั้งโปรแกรม....25
3.18 แสดงหน้าต่างการติดตั้งโปรแกรม LabVIEW 2011....26

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.19	แสดงหน้าต่างการเรียกหา Device Drivers	26
3.20	แสดงหน้าต่างการติดตั้งโปรแกรมเสร็จสมบูรณ์	26
3.21	แสดงหน้าต่างการโหลดเข้าสู่ตัวโปรแกรม LabVIEW 2011	27
3.22	แสดงหน้าต่างการเปิดหน้าที่ใช้โปรแกรม	27
3.23	แสดงหน้าต่างที่ใช้สำหรับการเขียนโปรแกรม	28
3.24	แสดงหน้าต่างพาเนล	29
3.25	แสดงหน้าต่างการเปลี่ยน Control และ Indicator ที่หน้าต่างพาเนล	29
3.26	แสดงหน้าต่างในส่วน Numeric Controls and Indicator	30
3.27	แสดงหน้าต่างในส่วนของกราฟ	31
3.28	แสดงหน้าต่างในส่วนของการปรับแต่งและตั้งค่ากราฟ	31
3.29	แสดงหน้าต่างบล็อกไดอะแกรม	32
3.30	แสดงส่วนประกอบต่างๆของบล็อกไดอะแกรม	33
3.31	แสดงการเชื่อมต่อสัญญาณ	35
3.32	การใช้โครงสร้างของคำสั่งแบบ Shortcut	36
3.33	แสดงหน้าต่างคำสั่งแบบ Pull Down	36
3.34	แสดง Tools Palette	38
3.35	แสดง Control Palette	39
3.36	แสดง Function Palette	40
3.37	แสดงทูลบาร์ในหน้าต่างพาเนล	40
3.38	แสดงทูลบาร์ในหน้าต่างบล็อกไดอะแกรม	41
3.39	แสดงการเริ่มต้นใช้งานโปรแกรม	42
3.40	แสดงไดอะล็อกบล็อกสำหรับเปิดใช้งานโปรแกรม LabVIEW	43
3.41	หน้าต่างฟร้อนท์พาเนล	44
3.42	หน้าต่างบล็อกไดอะแกรม	45
3.43	แสดงการสร้าง VI	46
3.44	แสดงการเลือกอุปกรณ์ไปวางที่ฟร้อนท์พาเนล	47
3.45	แสดงเทอร์มินอลที่เกิดขึ้นในบล็อกไดอะแกรม	47
3.46	แสดงฟังก์ชันในบล็อกไดอะแกรม	48
3.47	แสดงวิธีการลากสายสัญญาณ	49
3.48	แสดงการรันโดยใช้ Highlight Execution	49
3.49	แสดงการสร้าง Express VI	50
3.50	การเพิ่มคุณสมบัติไอคอนของ Express VI	51
3.51	การต่ออุปกรณ์ใช้งานร่วมกับ Express VI	52
3.52	แสดงหน้าต่าง Configure Simulate Signal	52

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.53	การสร้าง Control และ Indicator บนหน้าต่างบล็อกไดอะแกรม	53
3.54	ขั้นตอนการสร้าง Control และ Indicator บนหน้าต่างบล็อกไดอะแกรม	54
3.55	แสดงวิธีเลือกอุปกรณ์	54
3.56	แสดงวิธีการย้ายอุปกรณ์	55
3.57	แสดงการดีบัคโปรแกรม	56
3.58	เครื่องมือที่ใช้ตรวจสอบโปรแกรม	56
3.59	แสดงทิศทางการไหลของสัญญาณ	57
3.60	แสดง Tools Palette	58
3.61	แสดงการใช้ Breakpoint และ Probes	58
3.62	แสดงโครงสร้างการควบคุม	59
3.63	แสดงโครงสร้างของ For Loop	60
3.64	แสดงโครงสร้างของ While Loop	61
3.65	แสดงโครงสร้างของ Shift Register	61
3.66	แสดงการทำงานของ Shift Register	62
3.67	แสดงการใช้งาน Feedback Node	62
3.68	แสดงโครงสร้างของ Case Structure	63
3.69	แสดงโครงสร้างของ Flat และ Stack Seqence Structures	64
3.70	การ Add Sequence Local ของ Stack Sequence Structures	64
3.71	การสร้างโปรแกรมจากอุปกรณ์ทั่วไป	65
3.72	โครงสร้างและการใช้งาน Formula Node	65
3.73	แสดงการเพิ่มอินพุทและเอาท์พุทของ Formula Node	66
3.74	แสดง Array One Dimension (1-D)	66
3.75	การเรียกใช้งาน Array	67
3.76	แสดงอาร์เรย์สร้างจาก Control และ Indicator	68
3.77	ขั้นตอนการสร้างอาร์เรย์จาก Control และ Indicator	68
3.78	เทอร์มินอลของ array shell	68
3.79	แสดงวิธีการเพิ่มมิติของอาร์เรย์	69
3.80	การสร้างอาร์เรย์ด้วยฟังก์ชัน For Loop	70
3.81	การเปลี่ยนแปลงการแสดงผลจากอาร์เรย์เป็นแบบ Numeric	70
3.82	การต่อสายสัญญาณเมื่ออาร์เรย์อยู่ภายในลูป	71
3.83	การสร้างอาร์เรย์ 2 มิติ ด้วย For Loop	71
3.84	แสดงฟังก์ชันอาร์เรย์	72
3.85	แสดงการใช้งาน Array Size Function	73
3.86	แสดงการใช้งาน Initialize Array Function	73
3.87	แสดงการใช้งาน Build Array Function	74

เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.88 แสดงการใช้งาน Subset Array Function 74
3.89 แสดงการใช้งาน Index Array Function 75
3.90 แสดงการใช้งาน Index Array Function กับอาร์เรย์ 2 มิติ 75
3.91 แสดงการใช้งาน Cluster 76
3.92 แสดงการสร้างคลัสเตอร์ด้วย Control และ Indicator 76
3.93 แสดงการใช้งาน Bundle Function 77
3.94 แสดงการใช้งาน Bundle Function 78
3.95 แสดงการใช้งาน Unbundle Function 78
3.96 แสดงวิธีการเรียกใช้งาน Waveform Chart 79
3.97 แสดงคุณสมบัติของ Waveform Chart 80
3.98 แสดงคุณสมบัติของ Waveform Graph 80
3.99 แสดงคุณสมบัติของ Waveform Graph 81
3.100 Front Panel ของ XY Graph 81
3.101 Block Diagram ของ XY Graph 82
3.102 การเข้าถึง String controls และ String Indicators 83
3.103 การเพิ่ม Scrollbar ให้กับบล็อก String Indicator และ String Control 83
3.104 แสดงตัวอย่างการเข้ารหัสแบบต่างๆ 84
3.105 แสดง SI Remove Trend 84
3.106 แสดง SI Estimate Continuous Transfer Function Model 86
3.107 โครงสร้างของกรณีความสัมพันธ์ของ SI Estimate Impulse Response VI 87
3.108 SI Estimate Impulse Response 88
3.109 SI Model Simulation 89
4.1 แสดงกระบวนการที่ใช้ในการทดลอง 91
4.2 แสดงลักษณะการออกแบบกระบวนการ 92
4.3 แสดงโปรแกรมที่ใช้ควบคุมกระบวนการ 93
4.4 แสดงแสดงหน้าต่างพาเนล (Front Panel) 93
4.5 แสดงกราฟผลตอบสนองที่ MV = 75% 93
4.6 แสดงกราฟผลตอบสนองที่ MV = 40% 94
4.7 แสดงกราฟผลตอบสนองที่ MV = 40% 94
4.8 แสดงกราฟผลตอบสนองที่ MV = 65% 94
4.9 แสดงกราฟผลตอบสนองที่ MV = 30% 95
4.10 แสดงกราฟผลตอบสนองที่ MV = 50% 95
4.11 แสดงกราฟผลตอบสนองที่ MV = 55% 95
4.12 แสดงกราฟผลตอบสนองของ MV 96
4.13 แสดงโปรแกรมที่ใช้ในการหาเอกลักษณ์ของระบบ 96

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.14	แสดงหน้าต่างพาเนล(Front Panel) ของโปรแกรมที่ใช้ใน การหาเอกลักษณ์ของระบบ	97
4.15	แสดงผลของการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=1	97
4.16	แสดงผลของการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=1	98
4.17	แสดงผลของการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=2	98
4.18	แสดงผลของการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=2	99
4.19	แสดงผลของการทดลองสุ่ม num order (Zero)=2 และ den order (Pole)=2	99
4.20	แสดงผลของการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=3	100
4.21	แสดงผลของการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=3	100
4.22	แสดงผลของการทดลองสุ่ม num order (Zero)=2 และ den order (Pole)=3	101
4.23	แสดงผลของการทดลองสุ่ม num order (Zero)=3 และ den order (Pole)=3	101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปริญญานิพนธ์

ปัจจุบันในงานด้านวิศวกรรมการควบคุมและการวัดคุม ได้มีการใช้วิธีการหาเอกลักษณ์ของระบบ (System Identification) เพื่อหาแบบจำลองที่เหมาะสมสำหรับการออกแบบการคาดการณ์ขั้นตอนการทำงานหรือแบบจำลองของระบบ ที่ผ่านมาเทคนิคการหาเอกลักษณ์ของระบบ (System Identification) เป็นวิธีการที่มีประสิทธิภาพและเป็นเครื่องมือสำหรับข้อมูลในการสร้างแบบจำลองของระบบ โดยการหาเอกลักษณ์ของระบบนั้นสามารถทำได้หลากหลายวิธีไม่ว่าจะเป็น Parametric Methods หรือ Non-Parametric Methods โดยวิธีการหนึ่งที่สามารถให้ผลที่ถูกต้องที่นำมาใช้หาคือการศึกษาผลตอบสนองแบบอิมพัลส์ (Impulse Response) วิธีนี้เป็นวิธีที่ช่วยในการหาเอกลักษณ์ของระบบหรือหาแบบจำลองทางคณิตศาสตร์ของระบบแบบไดนามิกส์

โดยในปัจจุบันนี้มีการพัฒนาซอฟต์แวร์มากมายเพื่อช่วยในการออกแบบหรือควบคุมระบบในอุตสาหกรรมการผลิต ซึ่งซอฟต์แวร์หนึ่งที่ได้รับความนิยมมากในปัจจุบันคือ โปรแกรม LabVIEW เป็นโปรแกรมที่ใช้งานไม่ยากนัก ใช้งานได้หลากหลาย และมีประสิทธิภาพในการใช้งานสูง

งานวิจัยนี้ผู้วิจัยจึงได้ทำการนำโปรแกรม LabVIEW มาใช้ในการศึกษาและเขียนโปรแกรมเพื่อหาเอกลักษณ์ของระบบด้วยวิธีการศึกษาผลตอบสนองแบบอิมพัลส์ (Impulse Response) ของกระบวนการ ซึ่งกระบวนการที่ผู้วิจัยได้นำมาใช้ในการศึกษาคือกระบวนการเตาอบขนาดเล็ก โดยได้ทำการศึกษาทฤษฎีที่เกี่ยวข้อง การใช้งานโปรแกรม LabVIEW และทำการเขียนโปรแกรมเพื่อหาแบบจำลองทางคณิตศาสตร์ของกระบวนการเตาอบขนาดเล็ก

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษากระบวนการทำงานของเตาอบขนาดเล็ก
2. เพื่อศึกษาและดูผลตอบสนองของกระบวนการเตาอบขนาดเล็ก
3. เพื่อศึกษาเกี่ยวกับวิธีการหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็ก
4. เพื่อหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็กโดยใช้โปรแกรม LabVIEW

1.3 ขอบเขตของปริญญานิพนธ์

1. หาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็กโดยใช้วิธีการศึกษาผลตอบสนองแบบอิมพัลส์ (Impulse Response)
2. ใช้โปรแกรม LabVIEW ในการหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จัดทำรูปเล่มปริญญานิพนธ์

1.4 ขั้นตอนการศึกษา

ปริญญานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 5 บท โดยมีรายละเอียดของแต่ละบทดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึงความสำคัญของปริญญานิพนธ์ วัตถุประสงค์ของปริญญานิพนธ์ ขอบเขตของปริญญานิพนธ์ ขั้นตอนการดำเนินงาน และประโยชน์ที่คาดว่าจะได้รับจากปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับปริญญานิพนธ์ กล่าวถึงทฤษฎีทั้งหมดที่เกี่ยวข้องกับปริญญานิพนธ์ได้แก่ โมเดลหรือแบบจำลองประเภทต่างๆ ทฤษฎีเกี่ยวกับการหาเอกลักษณ์ของระบบ (System Identification) ด้วยวิธีการศึกษาผลตอบสนองแบบอิมพัลส์ (Impulse Response)

บทที่ 3 วิธีการดำเนินงาน กล่าวถึงองค์ประกอบและคุณสมบัติฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็ก กระบวนการการประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์ รวมทั้งการใช้งานและส่วนประกอบของซอฟต์แวร์ที่ใช้ในการออกแบบการทดลอง

บทที่ 4 การนำเสนอผลของข้อมูลและการวิเคราะห์ข้อมูล กล่าวถึงขั้นตอนในการหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็กโดยใช้โปรแกรม LabVIEW และนำเสนอผลการทดลอง

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ กล่าวถึงผลสรุปของการดำเนินงานเพื่อหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็ก

1.5 ประโยชน์ที่คาดว่าจะได้รับจากปริญญานิพนธ์

1. มีความรู้ความสามารถในการหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็ก
2. มีความรู้ความสามารถในการใช้งานโปรแกรม LabVIEW เพื่อหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็ก
3. สามารถนำความรู้ที่ได้ทั้งหมดไปประยุกต์กับกระบวนการทำงานอื่นๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 กล่าวนำ

เนื้อหาในบทที่ 2 นี้จะกล่าวถึงทฤษฎีทั้งหมดที่ใช้ในการทำปริญญานิพนธ์ เพื่อให้ทราบถึงแหล่งที่มาและวิธีการหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็ก โดยเนื้อหาประกอบด้วย ทฤษฎีของโมเดลหรือแบบจำลอง โครงสร้างโมเดล (Model Structure) ทฤษฎีการสร้างแบบจำลองระบบไดนามิกส์ (Modeling a Dynamic System) ทฤษฎีเอกลักษณ์ของระบบ (System Identification) ทฤษฎี System Identification Procedure ทฤษฎีชนิดของ System Identification Methods และ ทฤษฎี Non-Parametric Methods ในการนำมาออกแบบตัวควบคุมเพื่อให้สามารถออกแบบตัวควบคุมกระบวนการได้อย่างมีประสิทธิภาพ

2.2 โมเดลหรือแบบจำลอง (Model)

โมเดลหรือแบบจำลอง (Model) คือ สิ่งที่มนุษย์ได้สร้างขึ้นเพื่อใช้แทนของจริง เพื่อให้ง่ายต่อการศึกษา สามารถทำความเข้าใจการทำงานของระบบจริงได้ง่ายกว่าศึกษาจากระบบจริงโดยตรง

2.2.1 แนวทางการสร้างแบบจำลองหรือโมเดล

โมเดลอาจมีลักษณะ ขนาด และแบบแตกต่างกัน แต่จะมีลักษณะใน การพัฒนาโมเดลที่เหมือนกันคือ

- ข้อสมมติในการพัฒนาโมเดลจะต้องไม่ยุ่งยาก
- จะต้องกำหนดขอบเขตของโมเดลหรือสถานภาพเริ่มต้นให้ชัดเจน
- วัตถุประสงค์ของโมเดลที่ต้องการเรียนรู้

2.2.2 ประเภทของโมเดลหรือแบบจำลอง

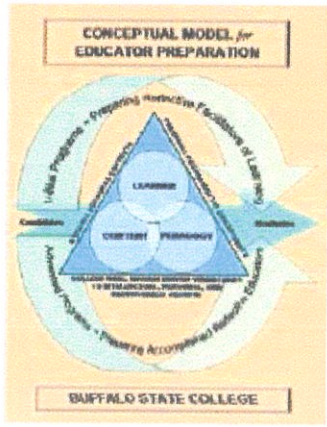
โดยทั่วไปสามารถแบ่งโมเดลตามลักษณะที่แตกต่างกันได้ 5 แบบ

1. โมเดลเชิงแนวคิด (Conceptual Model)
2. โมเดลเชิงกายภาพ (Physical Model)
3. โมเดลเชิงคณิตศาสตร์และสถิติ (Mathematical and statistical Model)
4. โมเดลเชิงภาพเคลื่อนไหว (Visualization Model)
5. โมเดลเชิงซิสเต็มไดนามิกส์ (System dynamics Model)

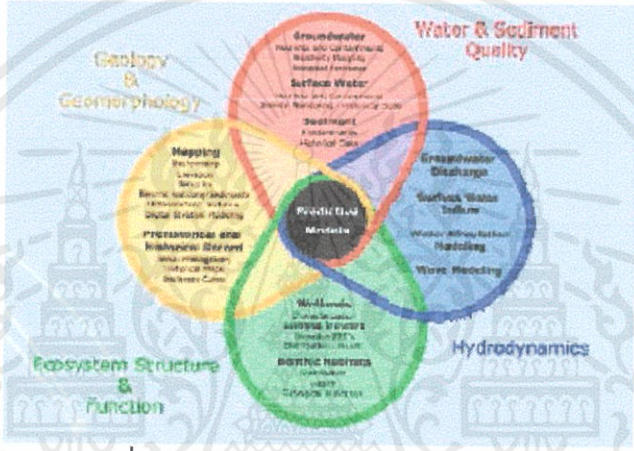
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.1 โมเดลเชิงแนวคิด (Conceptual Model)

เป็นแบบจำลองความคิด ซึ่งแสดงแนวคิดของระบบหรือกระบวนการโดยใช้ภาพเขียน หรือ ภาพวาด แสดงโครงสร้างของระบบ



รูปที่ 2.1 แสดงตัวอย่างโมเดลเชิงแนวคิด

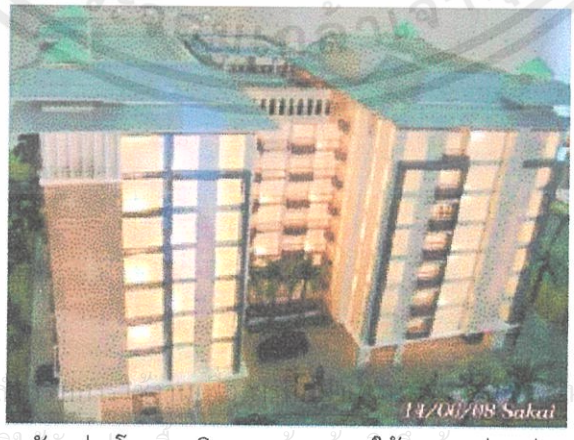


รูปที่ 2.2 แสดงตัวอย่างโมเดลเชิงแนวคิด

2.2.2.2 โมเดลเชิงกายภาพ (Physical Model)

คือแบบจำลองที่สร้างขึ้นมาเพื่อให้เห็นรูปทรง รูปร่างทางกายภาพ

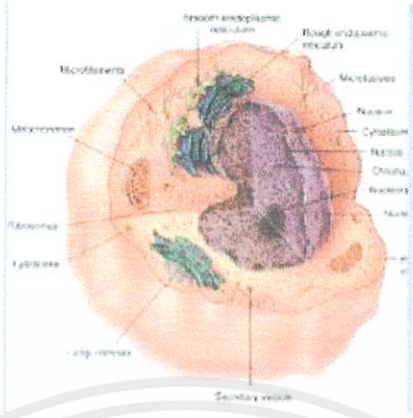
- ใช้มาตราส่วนที่ย่อขนาดจากของจริง ในกรณีที่ของจริงมีขนาดใหญ่



รูปที่ 2.3 แสดงตัวอย่างโมเดลเชิงกายภาพแบบใช้มาตราส่วนย่อขนาดจากจริง กรณีของจริงมีขนาดใหญ่

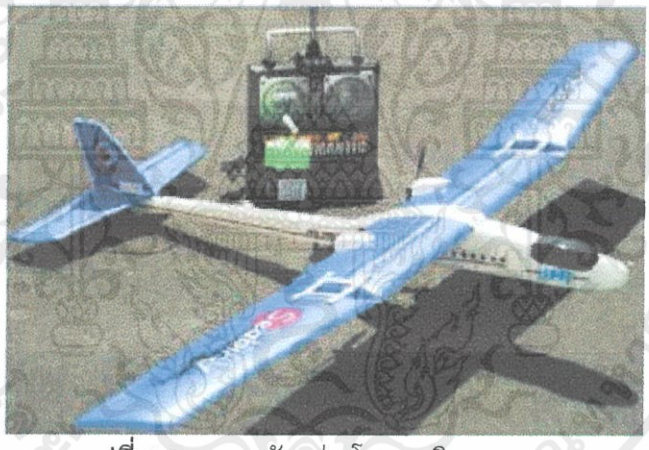
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

- ใช้มาตราส่วนขยายใหญ่จากของจริง ในกรณีที่ของจริงมีขนาดเล็กมาก



รูปที่ 2.4 แสดงตัวอย่างโมเดลเชิงกายภาพแบบใช้มาตราส่วนย่อขนาดจากจริง กรณีของจริงมีขนาดเล็ก

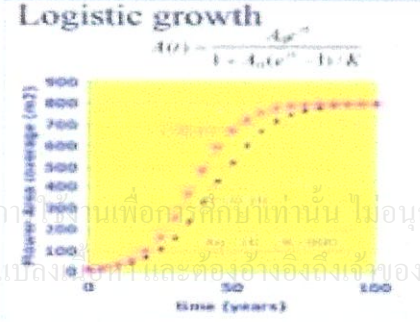
- แบบจำลองทางกายภาพอาจมีลักษณะที่นิ่งไม่สามารถเคลื่อนไหวได้ หรืออาจทำให้เคลื่อนไหวได้เหมือนของจริง



รูปที่ 2.5 แสดงตัวอย่างโมเดลเชิงกายภาพ

2.2.2.3 โมเดลเชิงคณิตศาสตร์และสถิติ (Mathematical and Static model)

1.) โมเดลทางคณิตศาสตร์ (Mathematical model)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.6 แสดงตัวอย่างโมเดลทางคณิตศาสตร์

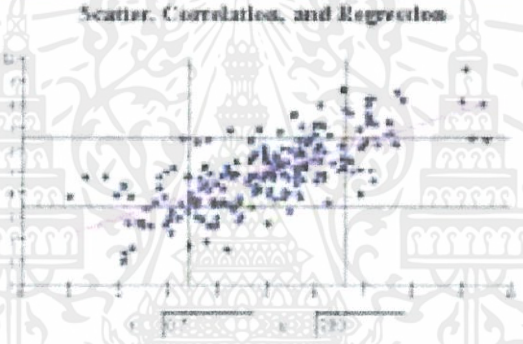
คือแบบจำลองที่ใช้ภาษาทางคณิตศาสตร์ในรูปของสมการ เพื่ออธิบายพฤติกรรมและความสัมพันธ์ระหว่างองค์ประกอบภายในระบบ อาจประกอบด้วยความสัมพันธ์ที่เป็นเส้นตรง เช่น $y = a+bx$ หรือความสัมพันธ์ระหว่างองค์ประกอบที่เป็นเส้นโค้ง เช่น $y_1 = y_0+ekt$

โมเดลทางคณิตศาสตร์แบ่งได้ 2 กลุ่มใหญ่ๆ

- กลุ่มที่หนึ่งอธิบายการทำงานของระบบภาพนิ่ง (static system) คือไม่นำเวลามาเกี่ยวข้อง

- กลุ่มที่สองอธิบายการทำงานของระบบพลวัตร สามารถทำให้คอมพิวเตอร์คำนวณการเปลี่ยนแปลงการทำงานของระบบตามกาลเวลาได้ โมเดลประเภทนี้เรียกว่า “computer simulation” เราสามารถใช้โมเดลทดลองการทำงานของระบบในสถานการณ์ต่างๆได้ และศึกษาผลลัพธ์ หากออกมาใกล้เคียงความจริง เราก็สามารถใช้โมเดลนี้ทำนายการเปลี่ยนแปลงในอนาคตได้

2.) โมเดลทางสถิติ (Statistical models)



รูปที่ 2.7 แสดงตัวอย่างโมเดลทางสถิติ

เป็นแบบจำลองที่ประกอบด้วยสมการสำหรับอธิบายความสัมพันธ์ระหว่างองค์ประกอบในระบบ เหมือนกับแบบจำลองทางคณิตศาสตร์ แต่จะต่างกันตรงที่ทางสถิตินี้ จะเกี่ยวข้องกับการวิเคราะห์ความผันแปร หรือความแตกต่างของปรากฏการณ์ และมุ่งเน้นที่จะอธิบายองค์ประกอบของความผันแปร

ตัวอย่างแบบจำลองทางสถิติได้แก่

- แบบจำลองสหสัมพันธ์ถดถอย (Multiple Regression model) เขียนในรูปของสมการได้ดังต่อไปนี้

$$Y = a + b_1X_1 + b_2X_2 + \ldots + b_nX_n + e$$

เมื่อ Y คือตัวแปรที่ต้องการให้อธิบายความแปรผัน

$X_1 \ldots X_n$ คือตัวแปรที่ใช้ในการอธิบายความแปรผันของ Y

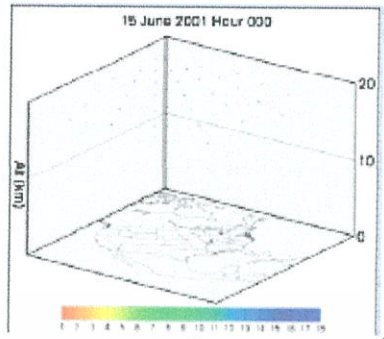
A คือค่าคงที่

$b_1 \ldots b_n$ คือตัวคูณ

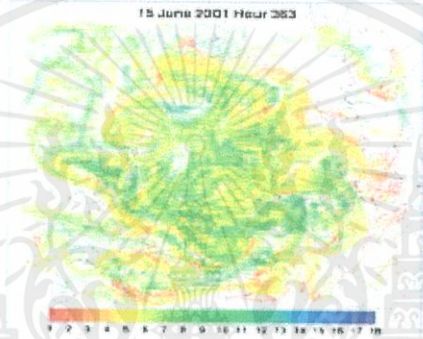
e คือค่าผิดพลาดของการอธิบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.4 โมเดลเชิงภาพเคลื่อนไหว (Visualization models)



รูปที่ 2.8 แสดงตัวอย่างโมเดลเชิงภาพเคลื่อนไหว



รูปที่ 2.9 แสดงตัวอย่างโมเดลเชิงภาพเคลื่อนไหว

Visualization หมายถึงกระบวนการที่ใช้เทคนิคทางเทคโนโลยีด้านการสร้างภาพ แปลงข้อมูลดิบให้เป็นภาพที่สามารถแสดงรูปทรง หรือกระบวนการที่เหมือนของจริงในระบบคอมพิวเตอร์ ภาพที่จะสร้างขึ้นมาอาจมี 2 มิติ หรือ 3 มิติ มีความเคลื่อนไหว แสดงความสัมพันธ์ระหว่างข้อมูล สามารถมองเห็นกระบวนการเปลี่ยนแปลง

โมเดลทางภาพ คือมีระบบการเปลี่ยนแปลงข้อมูลการทำงานของระบบต่างๆให้เกิดเป็นภาพเคลื่อนไหวหลายมิติเช่น โมเดลแสดงการทำงานของระบบหัวใจ โมเดลการเคลื่อนที่ของลมและการทำนายสภาพภูมิอากาศประโยชน์ของโมเดลประเภทนี้คือทำให้เข้าใจข้อมูล หรือเข้าใจระบบที่มีการทำงานสลับซับซ้อน นอกจากนี้เทคโนโลยีการสร้างภาพยังสามารถสร้างภาพในส่วนที่ไม่สามารถมองเห็นได้ด้วยตา เช่น การไหลเวียนของเลือดในหัวใจ

2.2.2.5 โมเดลเชิงซิสเต็มไดนามิคส์ (System Dynamics model)

เป็นโมเดล Computer Simulation Model คือสามารถคำนวณการทำงานของระบบในคอมพิวเตอร์ได้

ตัวแสดงหลักของระบบ

- สต็อค หมายถึง สิ่งที่สะสมและเปลี่ยนแปลงปริมาณได้ซึ่งอาจเป็นสิ่งมีชีวิต เช่น คน สัตว์ พืช หรือ สิ่งไม่มีชีวิต เช่น เงิน บ้าน น้ำ ความหิว ความรู้ เป็นต้น และสามารถวัดได้ทั้งเชิงปริมาณและคุณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โฟลว์ คือตัวที่กระทำให้สต็อคเปลี่ยนแปลง

ตัวแสดงประกอบของระบบ

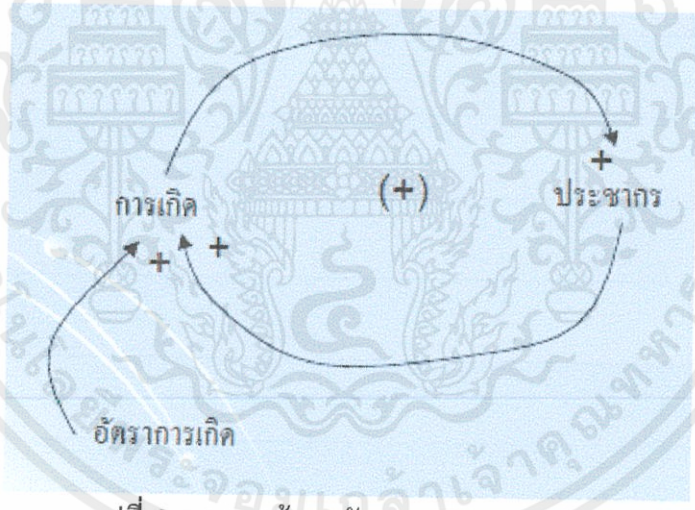
- ตัวแปลงค่า อาจเป็นค่าคงที่หรือตัวแปรที่ทำหน้าที่ขยายบทบาทของสต็อคและโฟลว์ให้เห็นภาพการทำงานของระบบดีขึ้น

- ตัวเชื่อม คือ ลูกศรแสดงทิศทางการทำงานและความสัมพันธ์ขององค์ประกอบภายในระบบ

2.2.3 การทำงานของโมเดลสามารถอธิบายด้วยผังภาพ 3 ลักษณะ

1) อธิบายด้วยผังวงจรย้อนกลับ (Feedback loop)

เป็นผังแสดงแนวคิดของระบบโดยมีลูกศรที่แสดงความสัมพันธ์ระหว่างองค์ประกอบมีทิศทางของหัวลูกศรไปในทางเดียวกัน ในผังจะประกอบไปด้วยตัวอักษรแทนองค์ประกอบลูกศรที่เชื่อมโยงระหว่างองค์ประกอบจะแสดงความสัมพันธ์ในเชิงเหตุและผล มีเครื่องหมาย บวกหรือลบกำกับที่หัวลูกศรแสดงความสัมพันธ์ระหว่างองค์ประกอบที่อยู่หัวและท้ายลูกศรว่าสอดคล้องหรือขัดแย้งกัน และยังมีเครื่องหมายบวกหรือลบในวงเล็บที่อยู่กลางวงจรจะอธิบายผลลัพธ์จากการทำงานของระบบ คือถ้าเป็นเครื่องหมายบวกแสดงว่าการทำงานของระบบจะให้ผลลัพธ์แบบเติบโตหรือเสื่อมถอย เรียกระบบนี้ว่าระบบปิดเชิงบวก ถ้าเป็นเครื่องหมายลบแสดงว่าการทำงานของระบบเป็นระบบปิดเชิงลบ หรือเรียกว่า ระบบสมดุล ดังรูปที่ 2.10

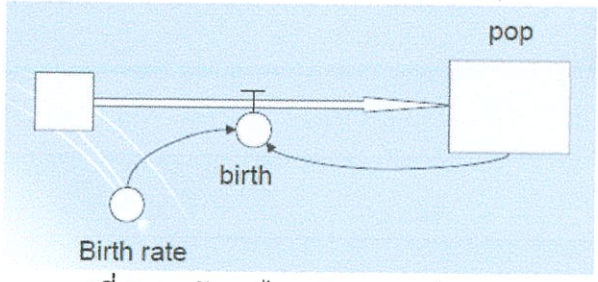


รูปที่ 2.10 วงจรย้อนกลับของระบบประชากร

2) อธิบายด้วยผังการไหลของระบบ (Flow diagram)

เป็นแผนผังที่แปลงแนวคิดของระบบ ที่จัดเรียงความสัมพันธ์ระหว่างองค์ประกอบในรูปของวงจรย้อนกลับ ให้อยู่ในรูปของสต็อค โฟลว์ และตัวแปลงค่า เพื่อใช้ในการสร้างสมการทางคณิตศาสตร์ หรือทางสถิติเพื่อคำนวณการทำงานของระบบตามกาลเวลา อาจเรียกว่าผังปฏิบัติการก็ได้ ดังรูปที่ 2.11

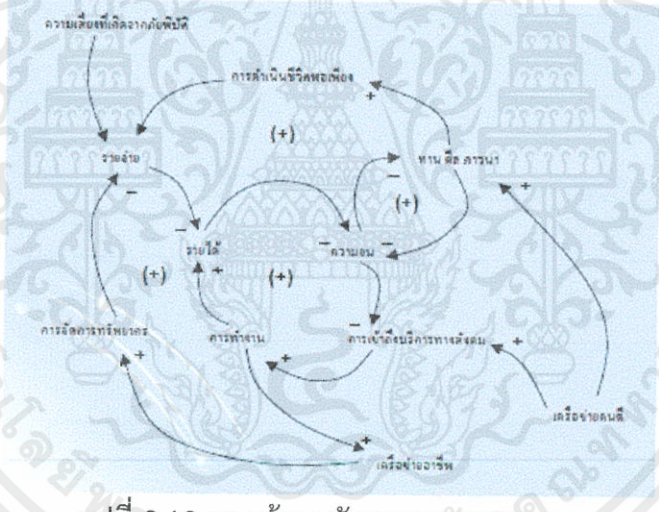
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ผังการไหลของระบบประชากร

3) อธิบายด้วยผังรูปตาวัว (Bull diagram)

เพื่อแสดงขอบเขตของโมเดล ในกรณีที่โมเดลแสดงระบบที่ง่ายๆ เราสามารถเห็นขอบเขตของโมเดลจากผังวงจรย้อนกลับ และผังการไหลของระบบได้ง่ายดังเช่นจากรูปที่ 2.10 องค์ประกอบที่อยู่ในระบบคือการเกิดและประชากร ซึ่งอาจเรียกอีกอย่างว่าตัวแปรภายในระบบ (endogenous variable) และอัตราการเกิดเป็นองค์ประกอบภายนอกระบบ เรียกอีกอย่างว่า ตัวแปรภายนอกระบบ (exogenous variable) แต่กรณีที่โมเดลเป็นระบบที่ซับซ้อนมีหลายสต็อค ดังรูป รูปที่ 2.12 การเขียนผังรูปตาวัวจะช่วยให้เห็นขอบเขตของระบบได้รวดเร็ว ดังรูปที่ 2.13



รูปที่ 2.12 วงจรย้อนกลับของระบบความจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.13 ผังรูปตาวัวของระบบความจน

2.3 โครงสร้างโมเดล (Model structure)

ในหลายๆกรณีขณะที่การจัดการกับปัญหาที่เกี่ยวข้องกับเอกลักษณ์ของระบบเราจะจัดการกับ Parametric models โดย model นี้แสดงลักษณะด้วย parameter vector ซึ่งจะแสดงด้วย θ โมเดลที่ตรงกันนี้ถูกแสดงด้วย $M(\theta)$ โดย θ จะมีค่าแตกต่างกันไปตามชุดของข้อมูลที่นำมาสร้างโครงสร้างของโมเดล (Model structure)

บางครั้งจะใช้ Non-parametric models เช่น model ที่อธิบายด้วยเส้นโค้ง, ฟังก์ชัน หรือตาราง เป็นต้น ตัวอย่างของ Non-parametric models เช่น Impulse response ซึ่งเป็นเส้นโค้งหรือฟังก์ชันที่ให้รายละเอียดข้อมูลเกี่ยวกับลักษณะหรือเอกลักษณ์ของระบบ

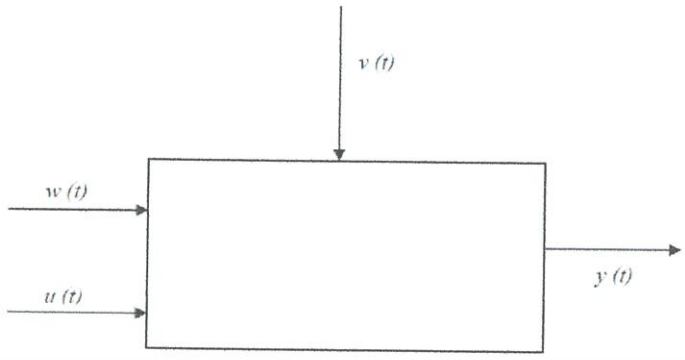
2.4 การสร้างแบบจำลองระบบไดนามิกส์ (Modeling a Dynamic system)

ระบบเป็นการนำตัวแปรที่แตกต่างกันมาปฏิสัมพันธ์กันแล้วทำการผลิตสัญญาณที่สังเกตได้ สัญญาณนี้เป็นสิ่งที่สนใจเรียกว่า เอาท์พุท

ระบบมักจะได้รับผลกระทบจากสิ่งเร้าภายนอก สัญญาณภายนอกที่สามารถจัดการโดยผู้สังเกตการณ์เรียกว่า อินพุท และสัญญาณอื่นๆจะเรียกว่า สัญญาณรบกวน ซึ่งแบ่งออกเป็นสัญญาณที่สามารถวัดได้โดยตรง กับ สังเกตผ่านผลกระทบที่เกิดกับเอาท์พุท ความแตกต่างระหว่างอินพุทกับสัญญาณรบกวนที่วัดได้มีความสำคัญน้อยในการสร้างแบบจำลองกระบวนการ

เอกลักษณ์ของระบบเป็นส่วนหนึ่งของการจำลองระบบไดนามิกส์ จากข้อมูลการทดลองระบบไดนามิกส์สามารถอธิบายได้ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดง Dynamic Model

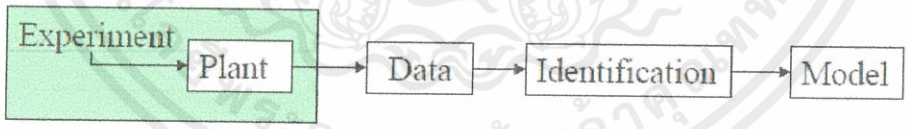
ระบบมี อินพุท แสดงด้วย u(t), เอาท์พุท แสดงด้วย y(t), สัญญาณรบกวนที่วัดได้ แสดงด้วย w(t), สัญญาณรบกวนที่ไม่สามารถวัดได้ แสดงด้วย v(t)

2.5 เอกลักษณ์ของระบบ (System Identification)

เอกลักษณ์ของระบบหรือ System Identification เป็นวิธีการของการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบแบบไดนามิกส์ซึ่งขึ้นอยู่ชุดข้อมูลของค่าสิ่งเร้าต่างๆที่วัดได้ และข้อมูลของการตอบสนอง

โดยสามารถนำเอกลักษณ์ของระบบไปใช้ในงานที่หลากหลายไม่ว่าจะเป็นงานด้าน วิศวกรรมเครื่องกล, ชีววิทยา, สรีรวิทยา, อุตุนิยมวิทยา, เศรษฐศาสตร์ และ การออกแบบรูปแบบการควบคุม (Model-based control design) เป็นต้น

ขั้นตอนการออกแบบโมเดลการควบคุมมักจะเกี่ยวข้องกับการหาโมเดลของแพลนท์, การวิเคราะห์และสังเคราะห์หาตัวควบคุมสำหรับแพลนท์, การจำลองระบบควบคุมแบบวงปิด และการปรับใช้ตัวควบคุมฮาร์ดแวร์แบบเวลาจริง (Real-time)



รูปที่ 2.15 แสดงแผนผังการหา System Identification

2.5.1 White-box identification

ประมาณค่าของแบบจำลองทางกายภาพจากข้อมูล

2.5.2 Gray-box identification

มีข้อมูลเกี่ยวกับโครงสร้างของแบบจำลองแล้วทำการประมาณค่าจากข้อมูลนั้น

2.5.3 Black-box identification

กำหนดรูปแบบโครงสร้างและประมาณการค่าจากข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 System Identification Procedure

ขั้นตอนการหาเอกลักษณ์ของระบบเป็นไปตามตรรกะธรรมชาติ คือเป็นแบบ step-by-step ตามขั้นตอน เริ่มจากเก็บข้อมูล จากนั้นทำการเลือกชุดของโมเดลและทำการเลือกชุดของโมเดลที่ดีที่สุด

โดยโมเดลที่ทำการเลือกครั้งแรกนั้นส่วนใหญ่มีแนวโน้มที่จะไม่ผ่านการ model validation tests โดยจะต้องทำการตรวจสอบขั้นตอนต่างๆของการดำเนินการ

โมเดลอาจจะบกพร่องเนื่องจากสาเหตุต่างๆ

1. ขั้นตอนที่เกี่ยวกับตัวเลขนั้นเกิดความผิดพลาดในการที่จะหา model ที่ดีที่สุดตามเกณฑ์
2. เกณฑ์ถูกเลือกมาไม่ดี
3. ชุดของโมเดลที่ได้มานั้นไม่เหมาะสม ทำให้ไม่สามารถอธิบายรายละเอียดของระบบได้
4. ชุดของข้อมูลให้ข้อมูลไม่เพียงพอที่จะช่วยในการเลือกโมเดลที่ดี

2.7 ชนิดของ System Identification Method

มี 2 วิธีหลักๆ คือ

2.7.1 Parametric Methods

เป็นวิธีการแมปปิ้งจากข้อมูลที่ทำการบันทึกเพื่อทำการประมาณค่า Parameter Vector และใช้ทำการประมาณ Model Parameter ขึ้นอยู่เฉพาะในชุดของข้อมูลที่สังเกตได้ที่ได้ทำการกำหนดไว้ ซึ่งมักเรียกว่า Parametric Identification

2.7.2 Non-Parametric Methods

ผลของหาแบบจำลองจะมีคุณสมบัติคืออยู่ในรูปของเส้นโค้งหรือฟังก์ชัน เป็นที่รู้จักกันว่าเป็นวิธีการแบบ Non-Parametric

2.8 Non-Parametric Methods

2.8.1 Step Response

การใช้ผลตอบสนองแบบขั้นบันได (Step response) ในการวิเคราะห์แบบชั่วขณะ (Transient analysis) เป็นอินพุทเพื่อให้เหมาะสมกับแบบจำลองแบบ simple-low order

โดยระบบอับดับหนึ่งและอันดับสองสามารถอธิบายโดยฟังก์ชันถ่ายโอน (Transfer function)

$$Y(s) = G(s)\, U(s)$$

เมื่อ Y(s) คือการแปลงลาปลาซ (Laplace transform) ของสัญญาณเอาท์พุท y(t)

U(s) คือการแปลงลาปลาซของอินพุท u(t)

G(s) คือฟังก์ชันถ่ายโอนของระบบ

เอกสารนี้เป็นของขวัญที่ผู้ใช้ไม่สามารถใช้งานเพื่อการค้าหรือแจกจ่ายต่อไปได้ ไม่ว่ากรณีใดๆ ทั้งสิ้น ผู้ใช้ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 Impulse Response

การตอบสนองแบบอิมพัลส์ของระบบ คือเมื่อเอาท์พุทออกมาเป็นสัญญาณสั้นๆ (Brief signal) ในขณะที่การกำหนดอิมพัลส์มีแนวคิดยากที่จะทำการคาดเดา มันแสดงให้เห็นถึงกรณีที่พัลส์สร้างระยะสั้นอนันต์ในเวลาขณะที่ยังคงรักษาพื้นที่ของมันหรือสมบูรณ์ (จึงทำให้เกิดส่วน high peak มากมาย)

พิจารณาระบบที่มีอินพุทแบบสเกลาและสัญญาณเอาท์พุท เป็น u(t) และ y(t) ตามลำดับ หากระบบเป็นเวลาคงที่ (Time invarant) การตอบสนองต่อสัญญาณเวลาที่แน่นอนไม่ได้ขึ้นอยู่กับช่วงเวลาที่แน่นอน มันเป็นเชิงเส้นถ้าการตอบสนองต่อเอาท์พุทนั้นมีผลรวมเชิงเส้นของอินพุทเหมือนกับผลรวมเชิงเส้นของผลตอบสนองเอาท์พุทของ individual อินพุท มันเป็น casual ถ้าเอาท์พุทในเวลาที่แน่นอนขึ้นอยู่กับอินพุทที่เวลานั้นเท่านั้น

เป็นที่ทราบกันดีว่า linear, time invariant, causal signal สามารถอธิบายโดยผลการตอบสนองแบบอิมพัลส์ของมันหรือด้วยฟังก์ชั่นแบบสัดส่วน (Weighting function), g(τ)

$$y(t) = \int_{\tau=0}^{\infty} g(\tau)u(t-\tau)d\tau$$

จากสมการข้างต้นทราบว่า $\{g(\tau)\}_{\tau=0}^{\infty}$ และรู้อินพุท u(s) เมื่อ s ≤ t เราสามารถคำนวณค่าเอาท์พุท y(s), s ≤ t สำหรับค่าอินพุทใดๆ ผลตอบสนองแบบอิมพัลส์นั้นจะเป็นคุณลักษณะที่สมบูรณ์ของระบบ

จากที่เห็นก่อนหน้านี้ ไดแร็กฟังก์ชันเดลต้า (Dirac function delta) $\delta(t)$ เป็นสิ่งที่จำเป็น เช่นอินพุทและเอาท์พุทจะเท่ากับ Weighting function, h(t) ของระบบ ซึ่งในทางปฏิบัตินั้นเป็นไปไม่ได้ที่จะเห็นค่า Ideal impulse จริง โดยเป็นแนวคิดที่เป็นประโยชน์เกี่ยวกับการ Idealization และใช้ในการประมาณอิมพัลส์

พิจารณาตัวอย่างต่อไปนี้เพื่อจะเข้าใจเกี่ยวกับการตอบสนองแบบอิมพัลส์

$$u(t) = \begin{cases} 1/\alpha & 0 \le t < \alpha \\ 0 & \alpha \le t \end{cases} \qquad (1)$$

อินพุทอยู่ในรูปปริพันธ์ $\int u(t)dt = 1$ เป็น Idealized Impulse

ข้อมูลข้างต้น (1) จะเกิดความผิดเพี้ยนค่าเอาท์พุทเท่าที่จะสามารถมองเห็นได้จากการคำนวณที่ง่าย

$$y(t) = \int_0^{\infty} h(s)u(t-s)ds = \frac{1}{\alpha}\int_{\max(0,t-\alpha)}^{t} h(s)ds \approx h(t)$$

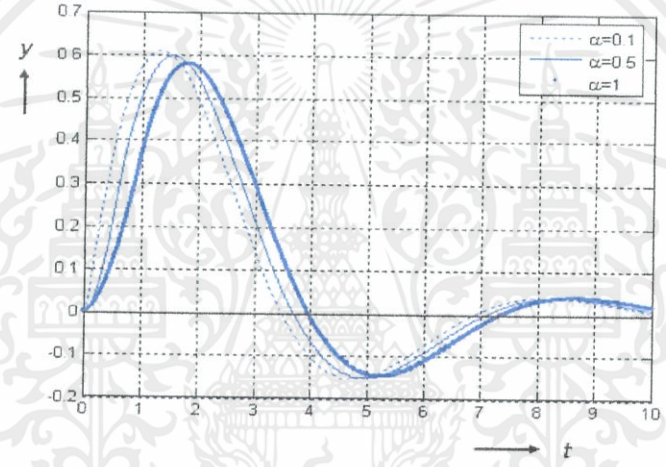
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากระยะเวลาของ α ของอิมพัลส์ (1) เป็นระยะสั้นเมื่อเทียบกับค่าคงที่เวลาแล้วความผิดเพี้ยนอาจจะเกิดเพียงเล็กน้อย แสดงให้เห็นได้ดังต่อไปนี้

พิจารณา Damped Oscillator ด้วยฟังก์ชันการถ่ายโอน (Transfer function)

$$G(s) = \frac{1}{s^2 + 0.8s + 1}$$

ดังนั้นจึงคำนวณ Weighting function และการตอบสนองต่อการประมาณอิมพัลส์ที่พิจารณาจากสมการข้างต้นและค่า Impulse duration, α ต่างๆ สามารถเห็นได้จากภาพที่การตอบสนองแบบอิมพัลส์จะเบี่ยงเบนน้อยมากจาก Weighting function ถ้า α มีขนาดเล็กเมื่อเทียบกับความผันผวน ค่าต่างๆของ α = 0.1, 0.5 และ 1 ตามลำดับซึ่งพล็อตในภาพด้านล่าง



รูปที่ 2.16 Weighting function and impulse responses of the damped oscillator excited by an approximate impulse

จากข้อมูลข้างต้น เป็นกล่าวถึงแนวคิดของการตอบสนองแบบอิมพัลส์ (Impulse Response) และผลลัพธ์ของ Weighting function สำหรับค่า α ที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์

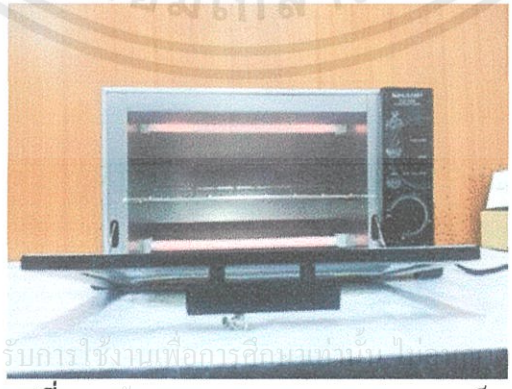
3.1 บทนำ

ในบทที่ 3 นั้นจะกล่าวถึง องค์ประกอบของกระบวนการเตาอบขนาดเล็ก ซึ่งเป็นกระบวนการที่ถูกนำมาใช้ในการการหาเอกลักษณ์ของกระบวนการ โดยใช้โปรแกรม LabVIEW ดังนั้นเพื่อให้การการหาเอกลักษณ์เป็นไปอย่างถูกต้อง โดยเกิดข้อผิดพลาดน้อยที่สุด จึงจำเป็นต้องศึกษาองค์ประกอบต่างๆ และพฤติกรรมของกระบวนการ ซึ่งประกอบไปด้วยองค์ประกอบด้านฮาร์ดแวร์และซอฟต์แวร์ ในส่วนของฮาร์ดแวร์จะกล่าวถึงส่วนประกอบต่างๆของกระบวนการเตาอบขนาดเล็ก ด้วยการแยกอธิบายอุปกรณ์แต่ละตัวโดยละเอียด และในส่วนของซอฟต์แวร์ จะอธิบายถึงซอฟต์แวร์ทั้งหมดที่เกี่ยวข้องกับการวิจัย รวมทั้งอธิบายวิธีการใช้งานโปรแกรมนั้นๆด้วย ซึ่งรายละเอียดต่างๆ จะอธิบายในแต่ละหัวข้อดังต่อไปนี้

3.2 องค์ประกอบของฮาร์ดแวร์

3.2.1 เตาอบขนาดเล็ก

เตาอบขนาดเล็ก เป็นอุปกรณ์ที่ใช้ในกระบวนการควบคุมอุณหภูมิ ซึ่งภายในเตาอบนั้นมีลักษณะโครงสร้างของเตาอบ แสดงได้ดังรูปที่ 3.1 เตาอบขนาดเล็ก มีลักษณะเป็นรูปทรงสี่เหลี่ยม มีฝาเปิด-ปิดเตาอยู่ด้านหน้า ในส่วนทางด้านขวามือมีปุ่มสวิชต์ตั้งเวลา และปุ่มเลือกหลอดความร้อน ภายในประกอบไปด้วยชั้นวางสำหรับใส่โหลด มีถาดใส่โหลด (น้ำ) พร้อมทั้งมีหลอดโซลิดสเตทซึ่งจะทำหน้าที่ให้ความร้อน จำนวนสองหลอด ให้กำลังงานสูงสุดหลอดละ 400 W



รูปที่ 3.1 แสดงกระบวนการเตาอบขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ข้อมูลทางเทคนิคของเตาอบขนาดเล็ก

รายละเอียด	คุณสมบัติ
เตาอบไฟฟ้า Sharp	KZ-50s
แรงเคลื่อนไฟฟ้า	220 V
ความถี่คลื่นไฟฟ้า	50 Hz
กำลังไฟฟ้าสูงสุด	800 W
ฮีทเตอร์บน	400 W
ฮีทเตอร์ล่าง	400 W
ขนาดภายนอก (กว้าง x ยาว x สูง)	244 x 390 x 213 mm
น้ำหนัก	3.3 Kg

3.2.2 RTD ชนิด PT100

RTD (Resistance Temperature Detector) คือ อุปกรณ์วัดอุณหภูมิโดย ใช้หลักการเปลี่ยนแปลงค่าความต้านทานของลวดโลหะไปตามอุณหภูมิที่เปลี่ยนไป ลวดโลหะที่นำมาใช้ผลิตRTD มีหลายชนิดเช่น ทองแดง ทังสเตน นิเกิล แพรทินัม เนื่องจาก RTD แบบแพรทินัม 100 โอห์ม เป็นที่นิยมสูงสุด เพราะมีความเป็นเชิงเส้นที่ดีจึงถูกกำหนดให้เป็นแบบมาตรฐานที่ใช้ในห้องปฏิบัติการทั่วไป สำหรับ RTD แบบแพรทินัม 100 โอห์ม จะเปลี่ยนค่าความต้านทานโดยเฉลี่ย 0.385 โอห์ม ต่อ 1 องศาเซลเซียสการใช้งานปกติจะมีแหล่งจ่ายกระแสคงที่ 1 mA ดังนั้นเมื่ออุณหภูมิเปลี่ยนไป 1 องศาเซลเซียส แรงดันจะเปลี่ยนไป 0.385 mV ซึ่งมากกว่า Thermocouple ถึง 10 เท่า ดังนั้นจึงเหมาะสำหรับวัดค่าอุณหภูมิที่ต้องการความละเอียดสูงและอุณหภูมิต่ำจนถึงติดลบ

ค่าความต้านทานของลวดโลหะจะเปลี่ยนแปลงตามอุณหภูมิดังสมการ

$$dR_t / dT = \alpha R_o$$

$$R_t = R_o (1 + \alpha T)$$

เมื่อ R_t = ค่าความต้านทานของลวดโลหะที่อุณหภูมิ t °C

R_o = ค่าความต้านทานของลวดโลหะที่อุณหภูมิ 0 °C

α = สัมประสิทธิ์ของการเปลี่ยนแปลงค่าความต้านทานไฟฟ้าต่ออุณหภูมิ 1 °C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากตัว RTD เป็นเพียงค่าความต้านทาน จึงต้องมีวงจรจ่ายกระแสเพื่อให้เกิดเป็นแรงดันที่เปลี่ยนไป แล้วจึงนำแรงดันนี้ไปใช้งาน แต่กระแสจำนวนนี้ก็เป็นการสร้างความร้อนขึ้นใน

ตัว RTD ด้วย ทำให้ค่าความต้านทานสูงขึ้น จึงจำเป็นที่จะต้องจำกัดไม่ให้กระแสเลี้ยง RTD นี้มีค่าสูงเกินไป ในส่วนของช่วงการใช้งานสูงสุดของ RTD ชนิด PT100 นั้น จะอยู่ที่ -250 ถึง 600 องศา – เซลเซียส ซึ่งช่วงการใช้งานและความไวในการวัดของ RTD แต่ละตัวจะแตกต่างกันตามแต่ละบริษัทจะกำหนด

รูปที่ 3.2 ลักษณะของ RTD ชนิด PT100

ตารางที่ 3.2 ข้อมูลทางเทคนิคของ RTD ชนิด PT100

รายละเอียด	คุณสมบัติ
RTD PCR-1	แบบมือถือ, ปลายกลมมน
ชนิดของไส้ (Element)	PT 100
ขนาดของปลอกป้องกัน	เส้นผ่านศูนย์กลาง 3 mm / ยาว 150 mm
ความยาวของสาย	1 m
ช่วงของอุณหภูมิที่ใช้งาน	0 ถึง 150 °C
การใช้งาน	วัดอุณหภูมิของเหลว, อากาศ

ข้อดีของ RTD ชนิด PT100

1. มีเสถียรภาพมาก
2. มีความเที่ยงตรงที่สูง
3. มีความเป็นเชิงเส้นมากกว่าเทอร์โมคัปเปิ้ล

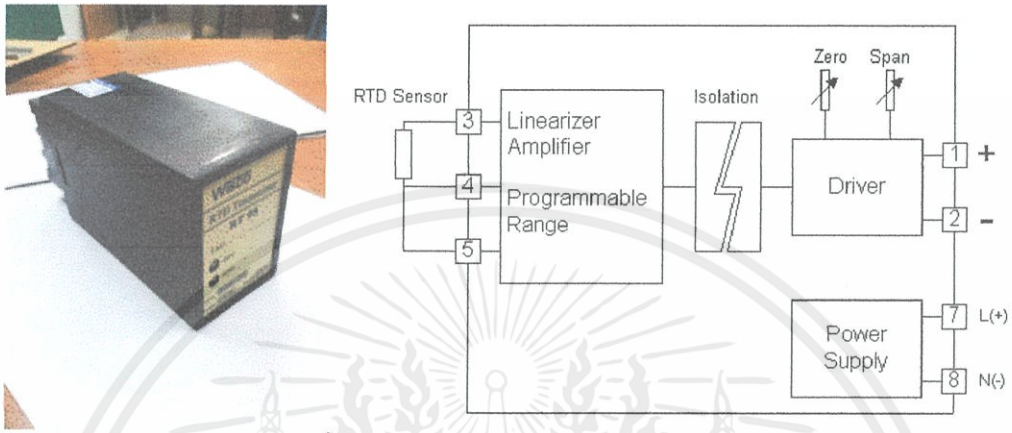
ข้อเสียของ RTD ชนิด PT100

1. ต้องการแหล่งจ่าย
2. มีราคาสูง
3. มีความร้อนเกิดขึ้นที่ตัวมันเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 RTD Transmitter : RT 95

RT 95 เป็นอุปกรณ์ทำหน้าที่แปลงอุณหภูมิจาก 0 ถึง 150 องศาเซลเซียส ให้อยู่ในรูปของความต่างศักย์ 0 ถึง 5 โวลท์ เพื่อส่งต่อไปยังDAQ (USB-6008) ของกระบวนการ สำหรับลักษณะของวงจรภายใน RT 95 แสดงในรูปที่ 3.3



รูปที่ 3.3 ลักษณะของวงจรภายใน RT 95

ตารางที่ 3.3 ข้อมูลทางเทคนิคของ RT 95

รายละเอียด	คุณสมบัติ
Input	0 ถึง 150 °C
Output	0 ถึง 5 V
Linearity	< $\pm$ 0.2% of span
Isolation	500 VAC
Power supply	220 VAC
Ambient temperature	0 ถึง 50 °C
Connection	Plug-in 11 pins socket
Mounting	Wall or DIN rail
Dimension	W50 x H70 x D130 mm

3.2.4 Signal Converter : Model TW-4M-1-N

Signal Converter เป็นอุปกรณ์ที่ทำหน้าที่แปลงความต่างศักย์จาก 0 ถึง 2.5 โวลท์ เป็นกระแสไฟฟ้า 4 ถึง 20 มิลลิแอมป์ เพื่อส่งต่อไปที่ Solid-State Relay ของกระบวนการ และลักษณะของ Signal Converter แสดงในรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 ลักษณะของ Signal Converter

3.2.5 Solid-State Relay : SSR RM1E23AA25

SSR เป็นอุปกรณ์ที่ทำหน้าที่ แปลงสัญญาณกระแสไฟฟ้าจาก 4 ถึง 20 มิลลิแอมป์ เป็นความต่างศักย์ 0 ถึง 220 โวลท์ เพื่อส่งต่อไปควบคุมฮีทเตอร์ของเตาอบขนาดเล็ก และลักษณะของ SSR แสดงในรูปที่ 3.5

รูปที่ 3.5 ลักษณะของ SSR

ตารางที่ 3.4 ข้อมูลทางเทคนิคของ SSR

รายละเอียด	คุณสมบัติ
SSR RM1E23AA25	1-Phase, Analog Switching
Input Control	4 ถึง 20 mA
Output	0 ถึง 220 VAC
Rated Operational Current	25 AACrms
Operating Temperature Range	0 ถึง 70 °C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 Data Acquisition : DAQ-6008

DAQ-6008 ก็คือ card interface เป็นแบบ plug-in เสียบเข้าใน computer bus ภายใน computer case ซึ่งทำให้ได้ความเร็วในการส่งถ่ายข้อมูลสูงขึ้น โดยการ์ดนี้ทำหน้าที่เชื่อมต่อกับคอมพิวเตอร์กับสัญญาณวัดที่ส่งมาจาก Signal conditioning module แต่ละการ์ดที่ติดตั้งในคอมพิวเตอร์จะมีแอดเดรสของตัวมันเองโดยระบุที่อยู่ในตำแหน่ง input/output memory map

โดยที่คุณสมบัติของ card interface จะถูกระบุไว้อยู่บนบอร์ด DAQ สำหรับตัวอย่างข้อกำหนดเบื้อต้นของบอร์ด DAQ ที่พบก็คือบอร์ด DAQ 16-bits resolution ADC, 100 kS/s sampling rate

- **จำนวนช่องสัญญาณอินพุท**

  Analog Input ของบอร์ด DAQ มักจะได้รับการระบุไว้ทั้ง 2 กรณี ดังนี้

  - Single-ended โดยอินพุทของบอร์ด DAQ ที่เป็น Single-ended นั้นจะอ้างอิงกับกราวด์จุดเดียวกันทั้งหมดและอินพุทแบบนี้จะใช้ในกรณีสัญญาณอินพุทมีระดับสูงมากกว่า 1 โวลท์ และใช้สายวัดสั้นๆ ไม่เกิน 15 ฟุต
  - Differential input แต่ละอินพุทจะมีกราวด์ของตัวเอง ซึ่งถือเป็นข้อดีเพราะเป็นการลดสัญญาณรบกวน common-mode ได้

- **อัตราการสุ่มวัด**

  เป็นพารามิเตอร์ที่จะกำหนดว่าบอร์ด DAQ จะสามารถวัดสัญญาณอินพุทได้ละเอียดมากเท่าใด โดยปกติการ์ด DAQ แบบ Plug-in จะมีอัตราการสุ่ม 30,000 ถึง 250,000 ครั้งต่อวินาที เนื่องจากการสุ่มวัดด้วยความถี่สูงๆจะทำให้สามารถติดตามการเปลี่ยนแปลงของสัญญาณอินพุทได้ในทันทีทันใด (ทฤษฎีของ Nyquist จะต้องทำการสุ่มอ่านค่าความถี่เสียง ด้วยความถี่อย่างน้อย 1.5 เท่าของความถี่สัญญาณอินพุท) ในทางกลับกันหากใช้อัตราการสุ่มวัดที่ไม่เหมาะสมแล้ว การตีความสัญญาณอินพุทก็จะผิดเพี้ยนไป โดยปกติแล้ว DAQ จะต้องทำการวัดสัญญาณอินพุทที่มีหลายช่องสัญญาณพร้อมๆ กัน บอร์ด DAQ จึงต้องมีวงจรมัลติเพล็กซ์เพื่อเลือกสัญญาณเข้ามาวิเคราะห์ทีละช่องสัญญาณ

- **ระดับการแยกแยะสัญญาณ**

  เกิดจากการแปลงสัญญาณวัดจากสัญญาณ Analog เป็น Digital ของบอร์ด DAQ ซึ่งจะใช้จำนวนบิตในการแทนค่าของสัญญาณ Analog ในแต่ละช่วงเวลา เช่น การแทนด้วยรหัสเพียง 3 บิต นั้น ให้ค่าความแยกแยะเพียง 8 ระดับ หากต้องการเพิ่มค่าความแยกแยะ เพื่อป้องกันข้อมูลของสัญญาณที่วัดได้ขาดหายไป ก็จะต้องเพิ่มการแทนบิตสูงขึ้น เช่น การแทนบิตด้วยรหัส 8 บิต จะสามารถแทนระดับสัญญาณอินพุทได้สูงถึง 255 ระดับ ซึ่งจะเป็นการเพิ่มความแม่นยำให้กับการวัดอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Communication-based**

  อุปกรณ์ Data Acquisition จะติดต่อกับคอมพิวเตอร์ผ่านทาง Communication port

  - Serial Communication ระยะทางไม่ไกลจะนิยมใช้ RS-232 โดยสามารถมีอัตราเร็วในการส่งถ่ายข้อมูล 115K baud (bits per second) ซึ่ง 7 หรือ 8 bits (on/off signal) จะเป็นสัญญาณที่แสดงถึงอักษรตามมาตรฐาน ASCII code ถ้าต้องการระยะไกลมากกว่า RS-232 ก็สามารถใช้ RS-422 และ RS-485 เพราะว่ามีความสามารถในการสื่อสารข้อมูลที่สูงกว่า

รูปที่ 3.6 ลักษณะของ DAQ-6008

3.3 โปรแกรมที่เกี่ยวข้องกับกระบวนการ

โปรแกรมเป็นส่วนประกอบหนึ่งที่สำคัญของปริญญานิพนธ์ ซึ่งเป็นตัวที่ทำหน้าที่ในการควบคุมกระบวนการให้เป็นตามที่ต้องการ ซึ่งโปรแกรมที่เลือกมาใช้ในครั้งนี้ ได้แก่ โปรแกรม LabVIEW 2011 ของบริษัท National Instrument เพื่อให้เป็นที่เข้าใจจะขอกล่าวถึงวิธีการใช้โปรแกรม LabVIEW และส่วนประกอบต่างๆในตัวของโปรแกรมที่เกี่ยวของกับปริญญานิพนธ์ ดังต่อไปนี้

3.3.1 โปรแกรม LabVIEW 2011

โปรแกรม LabVIEW เป็นเครื่องมือในการพัฒนาโปรแกรมชนิดหนึ่ง โดยการเขียนโปรแกรมในลักษณะของ Graphical-based Programming หรือเป็นการเขียนโปรแกรมเชิงรูปภาพ หรือกล่าวได้ว่าโปรแกรม LabVIEW เป็นภาษากราฟฟิกนั่นเอง นอกจากนั้นแล้วโปรแกรม LabVIEW ยังสามารถนำไปวัดค่าสัญญาณทางกายภาพ (Acquisition) จากเซ็นเซอร์นำมาทำการวิเคราะห์ (Analysis) แล้วก็ทำการแสดงผล (Presentation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

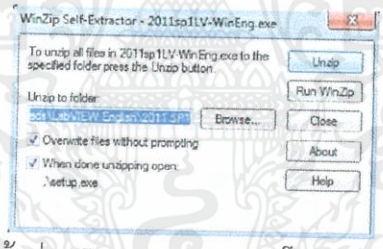
โปรแกรม LABVIEW เวอร์ชั่น 2011 สามารถเข้าไปดาวน์โหลดตามลิ้งค์ได้ที่ https://lumen.ni.com/nicif/us/evallvuser/content.xhtml ซึ่งต้องทำการลงทะเบียนกับทาง

บริษัท National Instrument ผ่านทางเว็บไซด์เสียก่อนจึงจะสามารถดาวน์โหลดโปรแกรม LABVIEW มาใช้ได้ เมื่อกรอกข้อมูลการลงทะเบียนเรียบร้อยแล้ว ให้ทำการยืนยัน แล้วจากนั้นให้ผู้ใช้ทำการเลือกคลิ๊ก Download LabVIEW 2011 ทางเว็บไซด์จะให้ทำการดาวน์โหลด 2011LV-WinEng_downloader.exe ทำการดับเบิ้ลคลิ๊ก 2011LV-WinEng_downloader.exe เพื่อดาวน์โหลดโปรแกรม LabVIEW 2011 ขนาด 1.03 GB ไฟล์ที่ดาวน์โหลดมาได้จะเป็นนามสกุล *.exe หลังจากนั้นให้ผู้ใช้ทำการดาวน์โหลดในหัวข้อ LabVIEW 2011 Modules and Toolkits ซึ่งทุก Toolkits จะมีสำคัญในการเขียนโปรแกรม ทำการดาวน์โหลดเช่นเดียวกับ 2011LV-WinEng_downloader.exe เมื่อได้ไฟล์โปรแกรม LabVIEW 2011 และ Toolkits ครบแล้ว จะเข้าสู่ขั้นตอนการติดตั้งต่อไป

3.3.1.1 ขั้นตอนการติดตั้งโปรแกรม LabVIEW

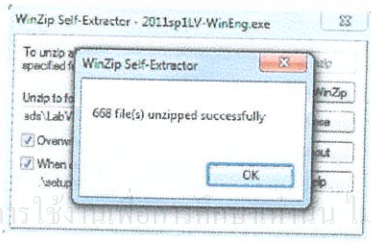
โปรแกรม LabVIEW ที่ใช้ในงานวิจัยนี้เป็นเวอร์ชั่น 2011 ซึ่งมีขั้นตอนในการติดตั้งโปรแกรมดังต่อไปนี้

1. ทำการดับเบิ้ลคลิ๊ก 2011sp1LV-WinEng ที่ดาวน์โหลดเสร็จสมบูรณ์แล้ว จะปรากฏหน้าต่างดังรูปที่ 3.7



รูปที่ 3.7 แสดงหน้าต่าง Extract files ของโปรแกรม LabVIEW 2011

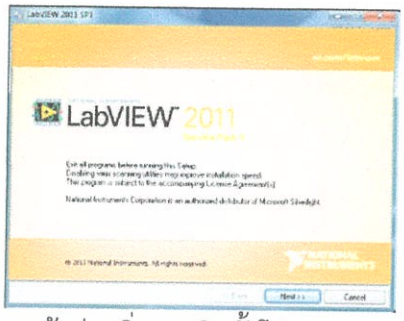
2. หลังจากขั้นตอนที่ 1 ให้ทำการคลิ๊กที่ Unzip หลังจากนั้นโปรแกรมจะทำการ Extract files โปรแกรม LabVIEW 2011 เมื่อ Extract files เสร็จเรียบร้อยแล้ว จะปรากฏหน้าต่างดังรูปที่ 3.8 จากนั้นให้คลิ๊กที่ OK



รูปที่ 3.8 แสดงหน้าต่าง Extract files สำเร็จ

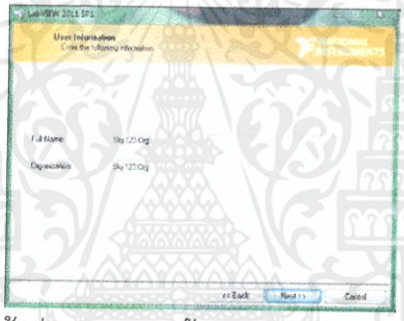
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

3. หลังจากคลิ๊ก OK แล้ว จะปรากฏหน้าต่างดังรูปที่ 3.9 เพื่อเข้าสู่ขั้นตอนลงโปรแกรม LabVIEW 2011 ให้ทำการคลิ๊กที่ Next เพื่อเริ่มขั้นตอนการลงโปรแกรม



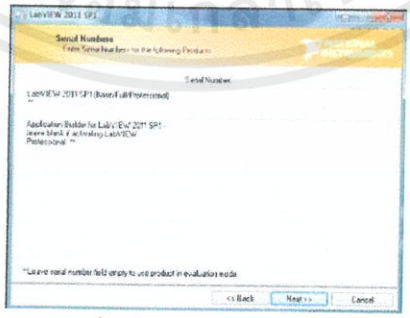
รูปที่ 3.9 แสดงหน้าต่างเริ่มการติดตั้งโปรแกรม LabVIEW 2011

4. หลังจากที่คลิ๊ก Next และจะปรากฏหน้าต่างดังรูปที่ 3.10 ซึ่งหน้าต่างนี้ ผู้ใช้จะต้องทำการกรอกข้อมูล คือ Full Name และ Organization ให้ครบ แล้วคลิ๊กที่ Next



รูปที่ 3.10 แสดงหน้าต่างการกรอกข้อมูล Full Name และ Organization

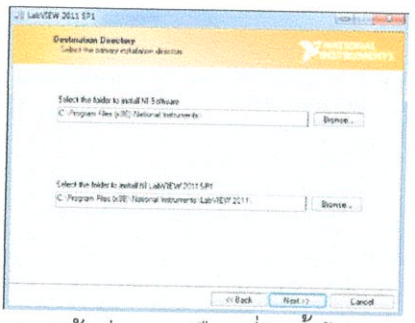
5. หลังจากคลิ๊ก Next ในขั้นตอนที่ 4 แล้ว จะปรากฏหน้าต่างดังรูปที่ 3.11 ซึ่งหน้าต่างนี้ ทำการกรอก Serial Number ถ้าในกรณีไม่มี Serial Number ผู้ใช้สามารถทดลองใช้งานได้ 30 วัน จากนั้นคลิ๊กที่ Next



รูปที่ 3.11 แสดงหน้าต่างการกรอกข้อมูล Serial Number

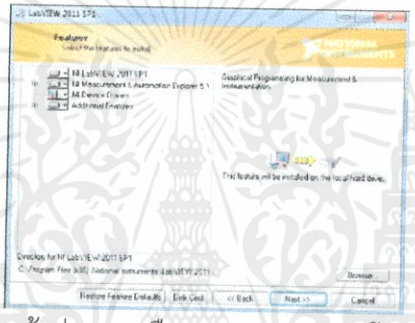
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หลังจากคลิ๊ก Next ในขั้นตอนที่ 5 จะปรากฏหน้าต่างดังรูปที่ 3.12 ซึ่งหน้าต่างนี้ ผู้ใช้จะต้องเลือกที่ที่จะติดตั้งโปรแกรม ปกติแล้วให้คลิ๊ก Next ได้เลย



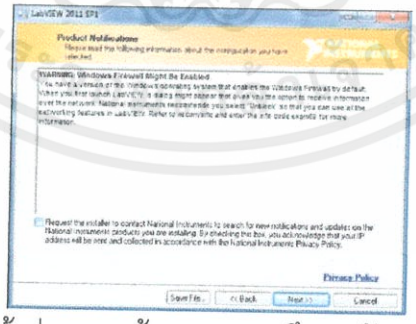
รูปที่ 3.12 แสดงหน้าต่างการเลือกที่ติดตั้งโปรแกรม LabVIEW

7. หลังจากคลิ๊ก Next ในขั้นตอนที่ 6 แล้ว จะปรากฏหน้าต่างดังรูปที่ 3.13 ซึ่งหน้าต่างนี้ จะให้ผู้ใช้ทำการเลือก Features ว่าจะลงหรือไม่ลงตัวไหน แล้วคลิ๊กที่ Next



รูปที่ 3.13 แสดงหน้าต่างการเลือก Features ของโปรแกรม LabVIEW

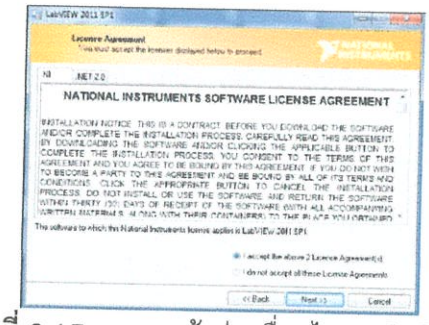
8. หลังจากคลิ๊ก Next ในขั้นตอนที่ 7 จะปรากฏหน้าต่างดังรูปที่ 3.14 ซึ่งหน้าต่างนี้ จะเป็นการเชื่อมต่อกับทางบริษัท National Instrument เพื่อแจ้งถึงการอัพเดทต่างๆของผลิตภัณฑ์ของ NI ซึ่งถ้าไม่จำเป็นแนะนำให้เอาเครื่องหมายติ๊กถูกออก แล้วคลิ๊กที่ Next



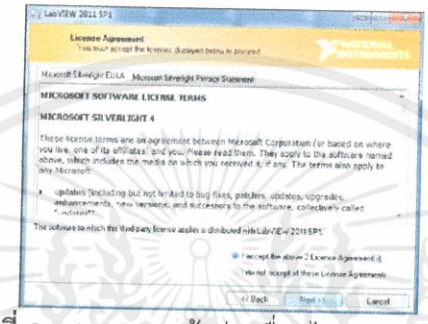
รูปที่ 3.14 แสดงหน้าต่างความต้องการทราบถึงการอัพเดทผลิตภัณฑ์ของ NI

9. หลังจากคลิ๊ก Next ในขั้นตอนที่ 8 จะปรากฏหน้าต่างดังรูปที่ 3.15 และรูปที่ 3.16 ซึ่งหน้าต่างนี้ จะเป็นการถามเกี่ยวกับเรื่องการยอมรับในเงื่อนไขเรื่องของลิขสิทธิ์ ผู้ใช้จะต้องเลือก I accept above 2 License Agreement(s). แล้วคลิ๊กที่ Next ทั้ง 2 หน้าต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

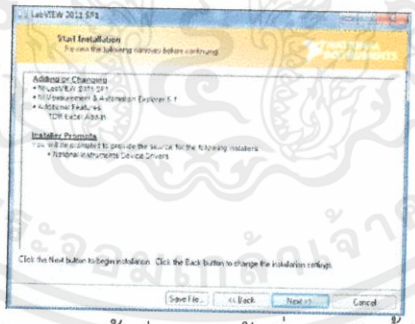


รูปที่ 3.15 แสดงหน้าต่างเงื่อนไขของลิขสิทธิ์



รูปที่ 3.16 แสดงหน้าต่างเงื่อนไขของลิขสิทธิ์

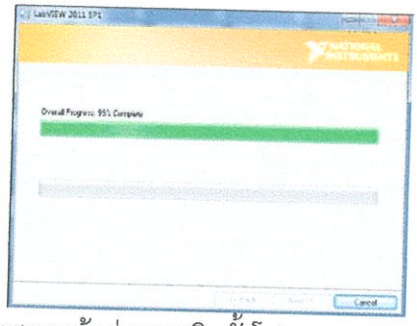
10. หลังจากคลิ๊ก Next ในขั้นตอนที่ 9 จะปรากฏหน้าต่างดังรูปที่ 3.17 ซึ่งหน้าต่างนี้ จะเป็นการแจ้งการเริ่มการติดตั้งโปรแกรมว่าจะมีการติดตั้งโปรแกรมอะไรลงไปบ้าง แล้วคลิ๊กที่ Next



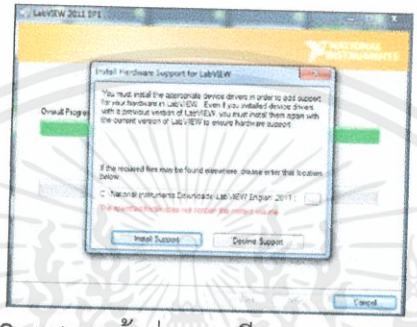
รูปที่ 3.17 แสดงหน้าต่างการแจ้งเริ่มการติดตั้งโปรแกรม

11. โปรแกรมจะเริ่มการติดตั้งดังหน้าต่างที่แสดงดังรูปที่ 3.18 เมื่อโปรแกรมติดตั้งเสร็จเรียบร้อยแล้ว จะมีการเรียกหา Device Drivers ดังหน้าต่างที่ปรากฏดังรูปที่ 3.19 ในกรณีนี้ให้เลือกคลิ๊กที่ Decline Support เนื่องจากไม่มี Device Drivers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

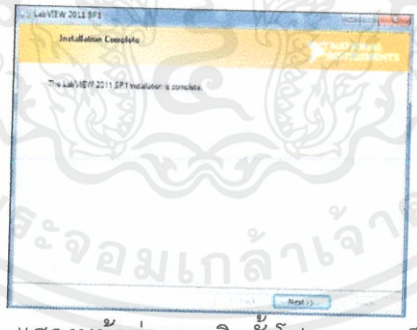


รูปที่ 3.18 แสดงหน้าต่างการติดตั้งโปรแกรม LabVIEW 2011



รูปที่ 3.19 แสดงหน้าต่างการเรียกหา Device Drivers

12. หลังจากคลิ๊กที่ Decline Support ในขั้นตอนที่ 11 แล้ว จะปรากฏหน้าต่างดังรูปที่ 3.20 นั่นแสดงว่าการติดตั้งเสร็จสมบูรณ์แล้ว จากนั้นให้ทำการรีสตาร์ทเครื่องคอมพิวเตอร์



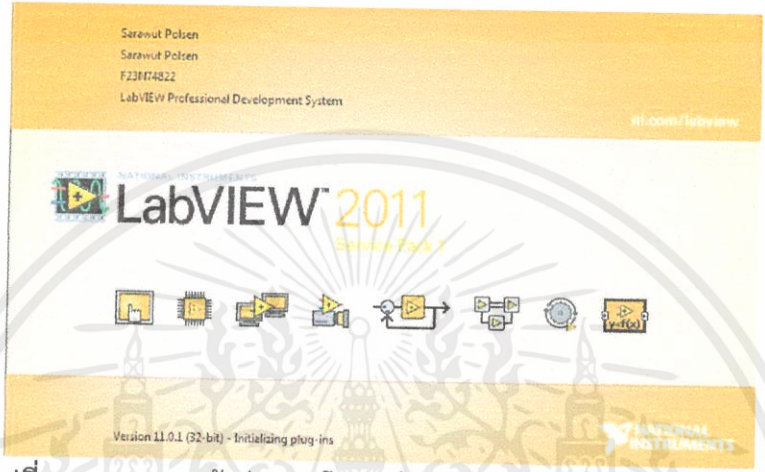
รูปที่ 3.20 แสดงหน้าต่างการติดตั้งโปรแกรมเสร็จสมบูรณ์

เมื่อทำการติดตั้งโปรแกรม LabVIEW 2011 เรียบร้อยแล้ว ผู้วิจัยจำเป็นที่จะต้องทำการติดตั้งตัว NIDAQ.exe ด้วย เพื่อที่โปรแกรม LabVIEW 2011 และคอมพิวเตอร์จะสามารถเชื่อมต่อกับ NI USB 6008 ได้ และใช้ในการรับส่งสัญญาณเพื่อจัดการและควบคุมกระบวนการต่อไป

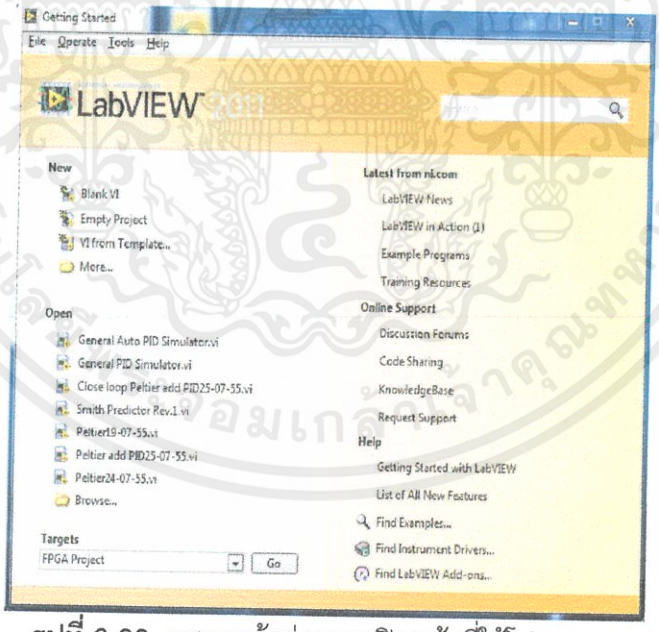
เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 การใช้งานโปรแกรม LabVIEW

เมื่อผู้ใช้ดับเบิ้ลคลิ๊กที่ไอคอนโปรแกรม LabVIEW 2011 จะปรากฏหน้าต่างแสดงได้ดังรูปที่ 3.21 หน้าต่างนี้ จะแสดงการโหลดเข้าสู่ตัวโปรแกรม LabVIEW 2011 เมื่อเข้าสู่ตัวโปรแกรมแล้วจะปรากฏหน้าต่างดังรูปที่ 3.22 ทำการเลือกคลิ๊กที่ Blank VI เพื่อเข้าสู่หน้าต่างการเขียนโปรแกรม LabVIEW 2011



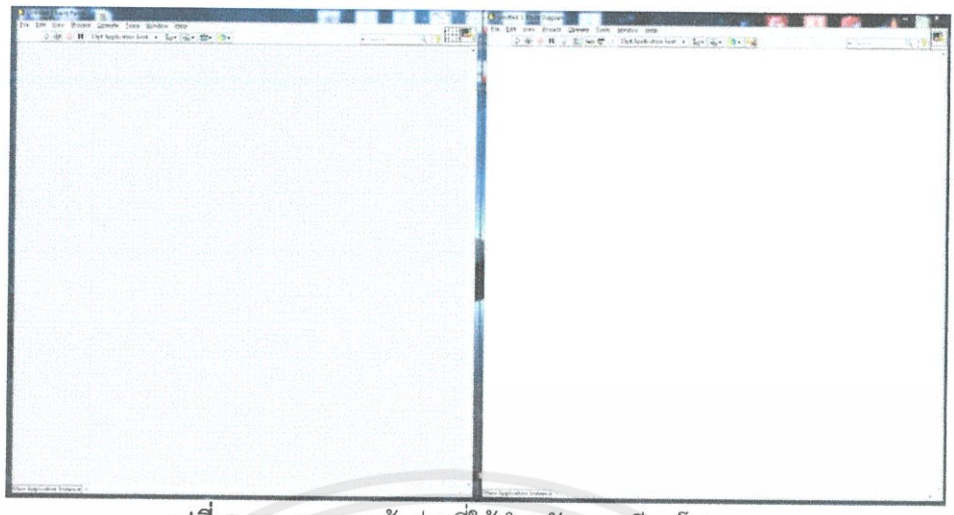
รูปที่ 3.21 แสดงหน้าต่างการโหลดเข้าสู่ตัวโปรแกรม LabVIEW 2011



รูปที่ 3.22 แสดงหน้าต่างการเปิดหน้าที่ใช้โปรแกรม

เมื่อคลิ๊กเลือก Blank VI แล้วจะปรากฏหน้าต่างที่ใช้สำหรับเขียนโปรแกรม LabVIEW 2011 ขึ้นมาพร้อมกัน 2 หน้าต่างดังรูปที่ 3.23 โดยด้านซ้ายเรียกว่า พาเนล และด้านขวาเรียกว่า บล็อกไดอะแกรม ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

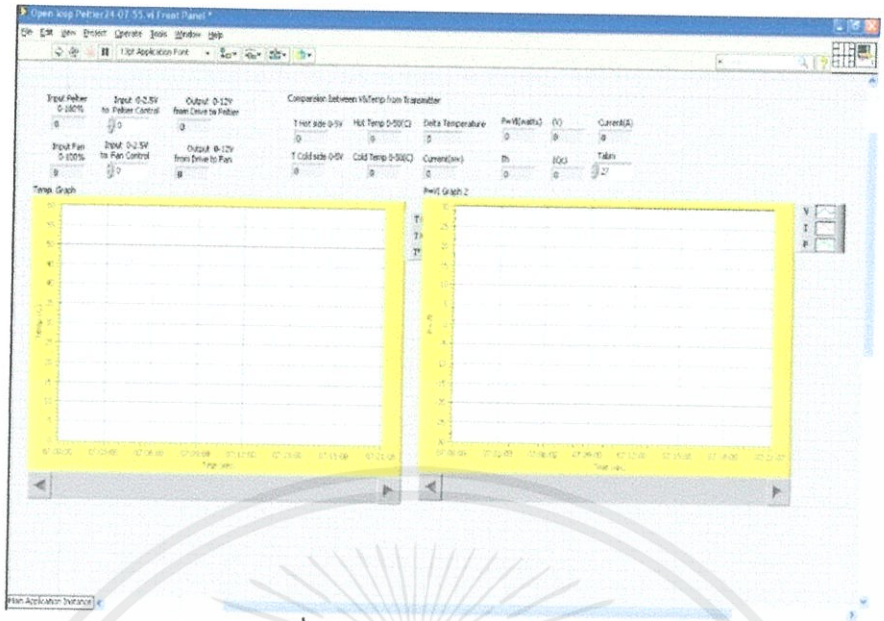


รูปที่ 3.23 แสดงหน้าต่างที่ใช้สำหรับการเขียนโปรแกรม

- **พาเนล**

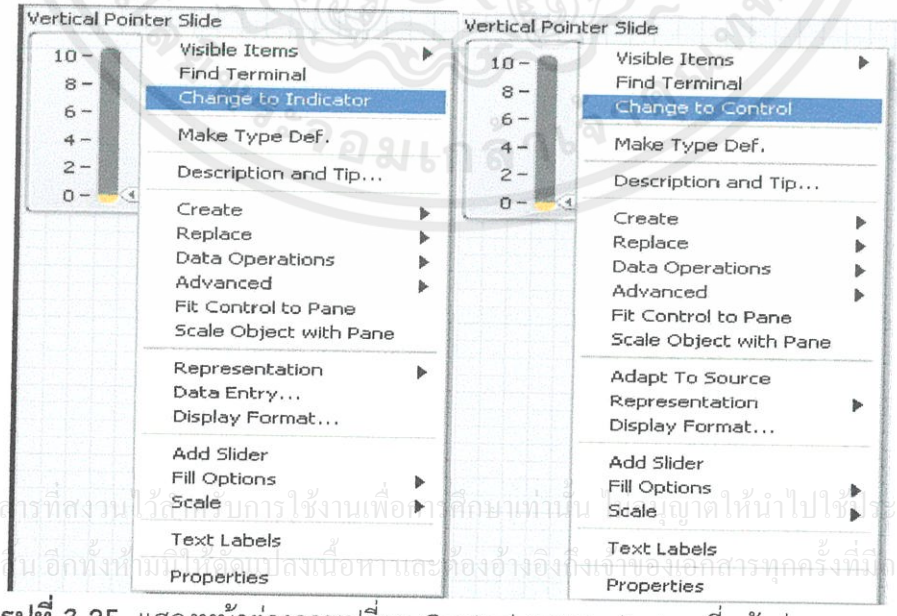
พาเนล เป็นส่วนที่ใช้ในการแสดงผลค่าต่างๆ ที่ผู้ใช้ต้องการทราบไม่ว่าจะเป็นการแสดงผลในเชิงตัวเลข ตัวอักษร กราฟ และอื่นๆ และใช้ในการควบคุมโดยการป้อนค่าที่ผู้ใช้ต้องการไม่ว่าจะเป็นตัวเลขหรือเป็นอักษรก็ตาม พูดง่ายๆก็คือพาเนลเป็นได้ทั้ง indicator และ control หรือเป็น Output และ Input นั่นเองสำหรับให้ผู้ใช้ สามารถสร้างรูปแบบขึ้นเอง ได้ตามความพอใจของผู้ใช้ ส่วนประกอบที่ของพาเนลมีอยู่ 3 ส่วน คือ พื้นที่สำหรับเขียนโปแกรม เมนูบาร์ และทูลบาร์ การเขียนโปรแกรมลงบนพาเนลนั้นไม่ใช่เรื่องยาก เพียงแต่ให้ผู้ใช้เข้าใจเสียก่อนว่า ผู้ใช้ต้องการเขียนอะไรลงบนพาเนล คือ ต้องการให้ตัวใดเป็นตัวแสดงผลหรือต้องการให้ตัวใดเป็นตัวควบคุม เพราะ Tool หนึ่งตัวสามารถทำหน้าที่ได้เพียงหน้าที่เดียว คือ ถ้าไม่ตัวควบคุมก็ต้องเป็นตัวแสดงผลในส่วนของพาเนล ผู้ใช้สามารถคลิ๊กเมาส์ทางด้านขวา บนหน้าต่างพาเนล จะปรากฏบล็อกทูลพาเลทขึ้นมา ซึ่งภายในทูลพาเลทจะประกอบด้วยตัวควบคุม และตัวแสดงผลแบบต่างๆ การวางอุปกรณ์ทำได้โดยการใช้เมาส์คลิ๊กค้างไว้ที่อุปกรณ์ที่ต้องการ จากนั้นลากไปวางในหน้าต่างพาเนลเป็นอันเสร็จเรียบร้อย หน้าต่างในส่วนของพาเนลแสดงดังรูปที่ 3.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แสดงหน้าต่างพาเนล

ซึ่งเมื่อผู้ใช้ทำการลากตัวอุปกรณ์ลงไปวางที่พาเนลเรียบร้อยแล้ว ผู้ใช้สามารถสั่งให้ตัวอุปกรณ์ตัวนั้นเป็นตัวควบคุม หรือตัวแสดงผลก็ได้ แต่ต้องเป็นอย่างใดอย่างหนึ่งเท่านั้น วิธีการ คือ คลิ๊กขวาที่ตัวอุปกรณ์ที่ผู้ใช้ต้องการจะเปลี่ยน ต่อจากนั้นจะเห็นคำสั่ง Change to Indicator หรือ Change to Control ดังรูปที่ 3.25 ซึ่งแตกต่างกันแล้วแต่ว่า ณ ตอนนั้นอุปกรณ์ที่ผู้ใช้ลากไปวางนั้นเป็นอย่างไร ถ้าเป็นประเภท Control เมื่อคลิ๊กขวาที่ตัวอุปกรณ์ จะเห็นเป็น Change to Indicator แต่ถ้าอุปกรณ์ ณ ตอนนั้นเป็นIndicator เมื่อคลิ๊กขวาที่ตัวอุปกรณ์ก็จะเห็นเป็น Change to Control ที่ได้กล่าวไปนั้นเป็นส่วนของพาเนลเท่านั้น



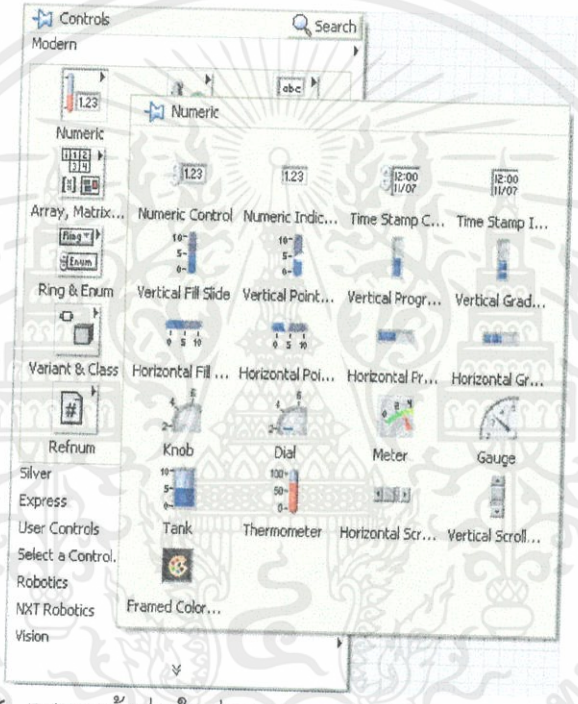
รูปที่ 3.25 แสดงหน้าต่างการเปลี่ยน Control และ Indicator ที่หน้าต่างพาเนล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเครื่องมือที่ใช้บ่อยในหน้าต่างพาเนล ได้แก่ Numeric และ Graph เป็นต้น

- **Numeric Controls and Indicator**

ใช้สำหรับเป็นตัวควบคุม และตัวแสดงผลที่มีลักษณะเป็นตัวเลข และกราฟฟิกเคลื่อนไหว ซึ่งจะสามารถเรียกใช้ในส่วนของหน้าต่าง Numeric โดยการคลิ๊กขวาที่พาเนลจะปรากฏ Control Palette คลิ๊กที่ Modern แล้ว Numeric จะปรากฏหน้าต่างของ Numeric ดังรูปที่ 3.26 ภายในก็จะมี Numeric Control, Numeric Indicator, Knob, Meter เป็นต้น ซึ่งค่าที่ควบคุม หรือค่าที่ใช้ในการแสดงผลล้วนแล้วแต่มีความสำคัญทั้งสิ้น เพราะค่าที่ผู้ใช้ควบคุมกระบวนการต้องมีการนำมาคำนวณหาค่าอื่นๆอีกในขั้นตอนต่อไป และค่าที่ใช้แสดงผลก็เช่นเดียวกัน

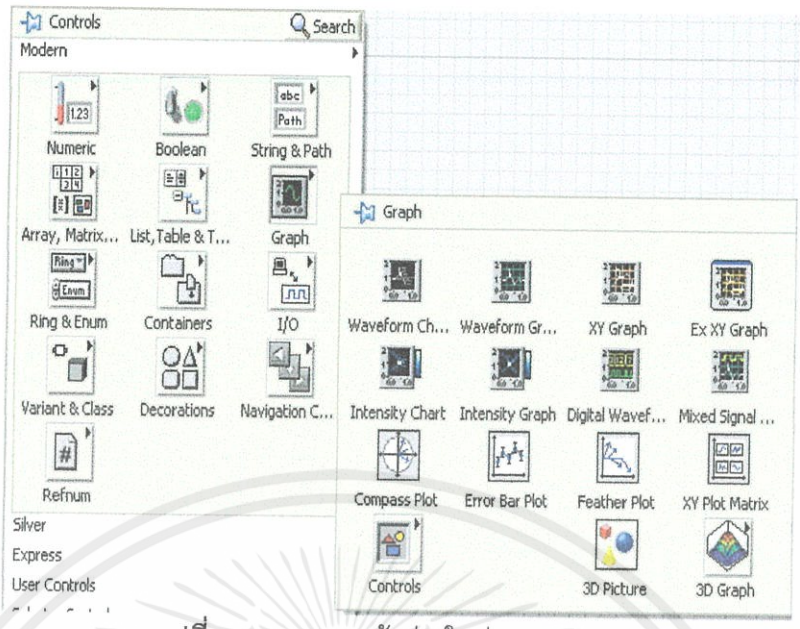


รูปที่ 3.26 แสดงหน้าต่างในส่วน Numeric Controls and Indicator

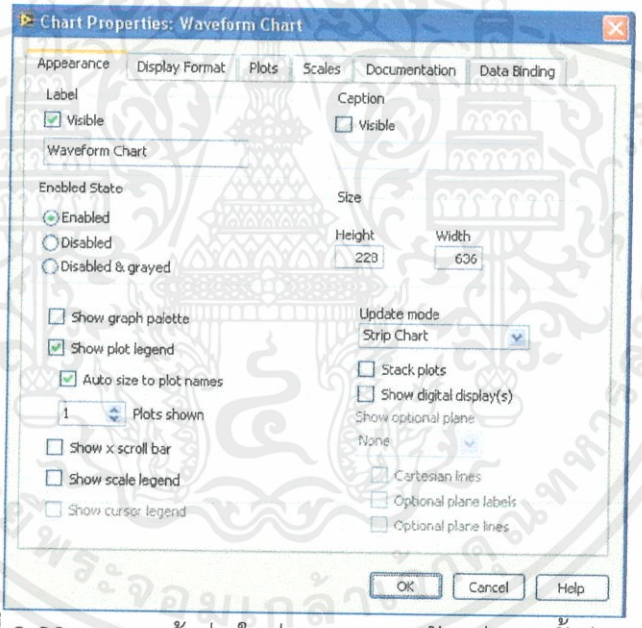
- **Graph**

ใช้สำหรับเป็นตัวแสดงผลมีลักษณะแบบเส้นกราฟ และสามารถแสดงเป็นตัวเลขได้อีกด้วยถ้าผู้ใช้ต้องการทราบ ซึ่งเส้นกราฟนั้นเป็นการพล็อตแบบเรียลไทม์ ผู้ใช้สามารถปรับแต่งหรือตั้งค่าที่กราฟได้ตามการใช้งานของผู้ใช้ ซึ่งการปรับแต่งหรือตั้งค่ากราฟ สามารถทำได้โดยการคลิ๊กขวาที่กราฟแล้วเลือก Properties ปรากฏหน้าต่างดังรูปที่ 3.28 ซึ่งผู้ใช้สามารถเรียกใช้ในส่วนของหน้าต่าง Graph โดยการคลิ๊กขวาที่พาเนลจะปรากฏ Control Palette คลิ๊กที่ Modern แล้ว Graph จะปรากฏหน้าต่างของ Graph ดังรูปที่ 3.27 ภายในก็จะมี Waveform Chart, Waveform Graph, XY Graph และอื่นๆ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 แสดงหน้าต่างในส่วนของกราฟ



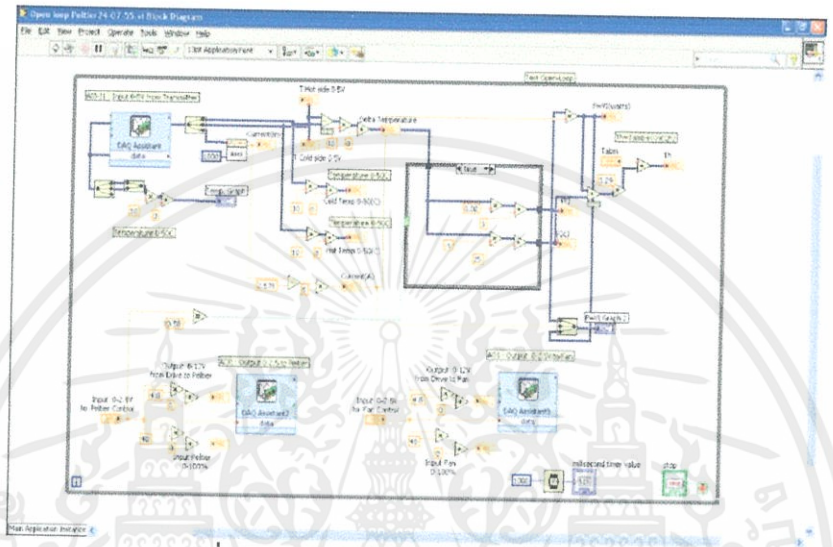
รูปที่ 3.28 แสดงหน้าต่างในส่วนของการปรับแต่งและตั้งค่ากราฟ

- **บล็อกไดอะแกรม**

บล็อกไดอะแกรม คือ ส่วนที่ใช้สำหรับการเขียนโปรแกรม และแสดงค่าต่างๆ ของการทำงานเช่น ถ้าผู้ใช้ตองการทราบค่าที่ออกมาจากบล็อกเครื่องมือต่างๆ ที่อยู่ในบล็อกไดอะแกรม ให้ใช้คำสั่ง Highligh Execution เป็นต้น นอกจากจะแสดงค่าต่างๆแล้ว ยังแสดงบล็อกของเครื่องมือต่างๆ ที่ใช้ในการเขียนโปรแกรมว่ามีการวางการเชื่อมโยงบล็อกนั้น มีการใช้สายไฟต่อระหว่างบล็อกกันอย่างไร ในเบื้องต้นนี้ผู้ใช้อาจมองบล็อกไดอะแกรมว่าเป็น Data Flow Chart และตัวโปรแกรมหรือ code ของ LabVIEW ก็ได้ การเขียนบล็อกไดอะแกรม คือ การเขียน code ในภาษา G นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ เรียกว่าการเขียนโปรแกรมด้วยภาษากราฟิก (G-Languages) ซึ่งก็เหมือนกับการเขียน code ในภาษา C นั่นเอง ความแตกต่างที่สำคัญระหว่าง code ในภาษา C กับ บล็อกไดอะแกรมใน LabVIEW ก็คือ บล็อกไดอะแกรมนั้นพร้อมที่จะสามารถทำการประมวลผลตลอดเวลา นั่นคือในระหว่างที่ผู้ใช้สร้างบล็อกไดอะแกรมอยู่ LabVIEW จะทำการตรวจสอบการทำงานของ VI อยู่ตลอดเวลา ซึ่งหน้าต่างในส่วนของบล็อกไดอะแกรมแสดงดังรูปที่ 3.29



รูปที่ 3.29 แสดงหน้าต่างบล็อกไดอะแกรม

เมื่อพิจารณาองค์ประกอบในบล็อกไดอะแกรม พบว่ามีส่วนประกอบที่สำคัญ 3 ส่วน คือ

- Terminal
- Node
- Wire

โดยทั้งสามส่วน จะมีหน้าที่หลัก คือ ทำหน้าที่ควบคุมการส่งผ่านข้อมูล หรือผู้ใช้อาจเรียกว่าการไหลของข้อมูล (Data Flow) นอกจากนั้นยังทำหน้าที่กำหนดวิธีการประมวลผลของข้อมูลอีกด้วย ซึ่งจะขออธิบายทั้ง 3 ส่วน ดังต่อไปนี้

- **Terminal**

ทุกครั้งที่ผู้ใช้ทำการสร้างตัวควบคุม หรือตัวที่ใช้แสดงผลบนพาเนล ในหน้าต่างของบล็อกไดอะแกรมจะปรากฏ Terminal ขึ้น Terminal ก็คล้ายกับสถานีของข้อมูล คือจะเป็นทั้งสถานีต้นทางของข้อมูล ถ้า Terminal นั้นเป็น Terminal ของตัวควบคุมและขณะเดียวกันจะเป็นทั้งสถานีปลายทางของข้อมูลถ้า Terminal นั้นเป็น Terminal ของตัวแสดงผล

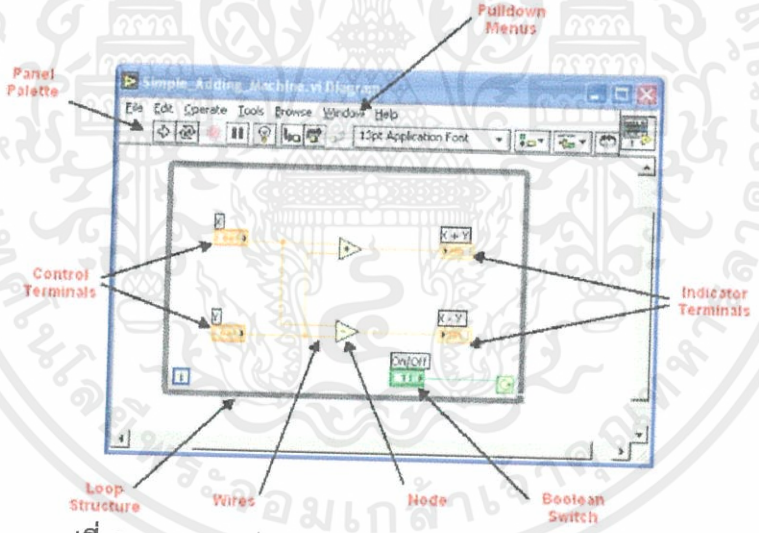
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่ควรเข้าใจอย่างหนึ่งก็คือ ถ้าหากผู้ใช้จะลบตัวควบคุม หรือตัวแสดงผลนั้นออกไปจากพาเนลแล้ว Terminal เหล่านี้ก็จะหายไปจากบล็อกไดอะแกรมเช่นกัน หรือถ้าหากผู้ใช้ลบ

Terminal ของตัวควบคุมหรือตัวแสดงผลออกจากบล็อกไดอะแกรม ตัวควบคุม หรือตัวแสดงผลเหล่านั้นก็จะหายไปจากพาเนลด้วยเช่นกัน

- **Node**

Node เป็นคำที่ใช้เรียก object ที่ทำกรรมวิธีใดๆ เพื่อที่จะทำการประมวลข้อมูลในบล็อกไดอะแกรม เช่นเดียวกับที่ผู้ใช้เขียนFlow Chart แล้วใช้สัญลักษณ์ต่างๆแทนวิธีการวิเคราะห์ข้อมูล เมื่อมีข้อมูลเข้าสู่ node สิ่งที่เกิดขึ้นภายใน node ก็จะขึ้นอยู่กับว่าจะกำหนดให้ข้อมูลที่ส่งเข้าไปนั้นจะมีการประมวลผลอย่างไร ซึ่งอาจจะเป็นการบวก ลบ คูณ หาร หารากยกกำลัง หรือเป็นประเภทของการเปรียบเทียบข้อมูลที่ได้รับ หรืออื่นๆ ซึ่งจะเป็นการประมวลผลทางคณิตศาสตร์ทั่วไป นอกเหนือจากนั้น node นี้จะมีส่วนที่เรียกว่า Function แบบต่างๆ ซึ่งจะเหมือนกับ Function สำเร็จรูป เช่น sine, cosine, log เป็นต้น ซึ่งก็จะเหมือนกับในภาษาที่เป็นตัวอักษรทั่วๆไป รูปที่ 3.30 แสดงถึงลักษณะของ Node และ Terminal และส่วนอื่นๆ ที่บรรจุอยู่ภายในบล็อกไดอะแกรมของโปรแกรม LabVIEW



รูปที่ 3.30 แสดงส่วนประกอบต่างๆของบล็อกไดอะแกรม

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Wires**

ขณะที่ผู้ใช้มีที่มาของข้อมูลส่วนประมวล หรือปรับแต่งข้อมูล และส่วนแสดงผลข้อมูล เรียบร้อยแล้ว ขั้นต่อไป คือ ผู้ใช้จะต้องสามารถควบคุมการส่งผ่านข้อมูลให้เป็นไปตามที่ผู้ใช้

ต้องการ อุปกรณ์ที่ใช้ใน LabVIEW ก็คือ การต่อสาย หรือ Wires ซึ่งจะเป็นการเชื่อมการส่งข้อมูลระหว่าง terminal หรือ node ต่างๆที่มีในบล็อกไดอะแกรมนี้เข้าด้วยกัน โดย wires นี้จะเป็นการกำหนดเส้นทางของข้อมูลว่าเมื่อออกจาก terminal หนึ่งแล้ว จะกำหนดการไหลไปที่ node ใดบ้าง มีลำดับเป็นอย่างไร และสุดท้ายจะให้แสดงผลที่ terminal ใดนั่นเอง ซึ่งการเชื่อมต่อสายนี้จะทำให้ผู้ใช้เข้าใจถึงหลักการของ Data Flow Programming ได้ดียิ่งขึ้น

เนื่องจากข้อมูลนั้นมีหลายแบบไม่ว่าจะเป็นเลขทศนิยม, เลขจำนวนจริง, ตัวอักษร หรือค่าจริง-เท็จ (Boolean) ดังนั้นเพื่อแสดงถึงความแตกต่างของข้อมูลแต่ละแบบ LabVIEW จึงกำหนดให้ลักษณะของ wires สำหรับข้อมูลแต่ละแบบมีลักษณะของเส้นและสีที่แตกต่างกัน โดยการใช้ Wiring Tool (รูปหลอดด้าย) ใน Tool Palette โดยที่

สีของ Control หรือ Indicator

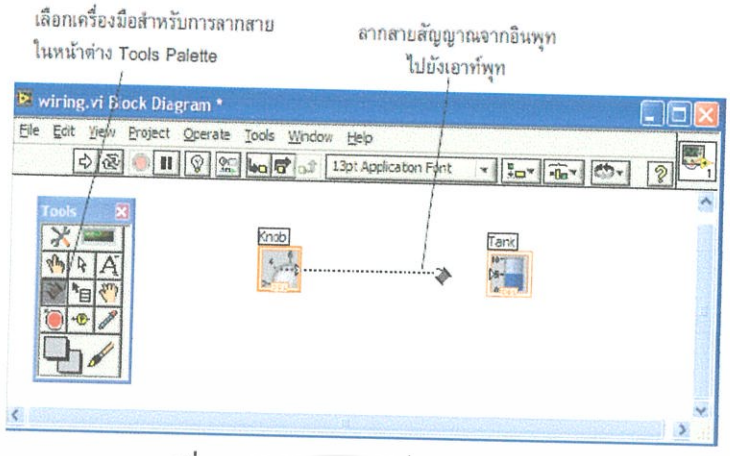
1. สีน้ำเงิน แทนตัวเลขที่เป็นจำนวนเต็ม
2. สีส้ม แทนตัวเลขที่มีค่าทศนิยม
3. สีเขียว แทน Logic หรือ Boolean
4. สีชมพู แทนค่าของ String

รูปแบบของWiring

1. เส้นบางแทนค่าจำนวนเดียว
2. เส้นหนาแทนค่าของอาร์เรย์ ขนาด 1 มิติ
3. เส้นคู่แทนค่าของอาร์เรย์ ขนาด 2 มิติ

การ Wiring ทำโดยใช้ Wiring Tool เลื่อนไปให้ตรง Terminals แรกที่ต้องการเชื่อมต่อสังเกตให้ Terminals กระพริบแล้วคลิ๊กเมาส์ จากนั้นให้ลากไปยัง Terminals ที่ต้องการเชื่อมต่อโดยจะเกิดเส้นประสีดำลากตาม cursor ให้คลิ๊กเมาส์บน Terminals ที่สอง ในขณะที่ Terminals นั้นกำลังกระพริบ ข้อแตกต่างระหว่าง Control กับ Indicator บนบล็อกไดอะแกรม คือ Control จะมีขอบเป็นเส้นหนา ส่วน Indicator จะมีขอบเป็นเส้นบาง แสดงดังรูปที่ 3.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 แสดงการเชื่อมต่อสัญญาณ

นอกจากนั้นข้อมูลแต่ละแบบดังกล่าวยังมีลักษณะเป็น scalars, 1-D array, 2-D array ได้ ซึ่งลักษณะของเส้นของข้อมูลแต่ละแบบก็จะแตกต่างออกไปอีก

ผู้ใช้จะเข้าใจหลักการของ Data Flow Programming ได้ในขณะที่ผู้ใช้ต่อสายระหว่าง terminal และ Node ต่างๆ เข้าด้วยกัน ซึ่งหลักการทำงานของ Data Flow Programming นั้นจะแตกต่างจากการเขียนโปรแกรมโดยใช้ตัวอักษร เพราะในโปรแกรมตัวอักษร คำสั่งจะถูกส่งเข้าสู่ส่วนประมวลผลทีละบรรทัด เพื่อการคำนวณตามลำดับบรรทัด แต่สำหรับใน Data Flow นั้นโปรแกรมจะคำนวณเมื่อมีข้อมูลส่งเข้ามาถึง Input ของ Node นั้นๆ ครบ และเมื่อข้อมูลส่งครบเข้าถึง Node นั้นแล้วจะมีการประมวลผล และส่งค่าที่ได้ออกไปตามการต่อเชื่อมสาย การคำนวณจะเสร็จสิ้นในแต่ละรอบเมื่อข้อมูลส่งข้ามไปถึง Terminal สุดท้าย การประมวลผลไม่ได้เป็นไปตามลำดับการจัดวาง คือ ไม่ได้ทำจากซ้ายไปขวาหรือบนลงล่าง แต่เป็นไปตามขั้นตอนการเดินทางของข้อมูล ซึ่งในการเขียนโปรแกรมในเบื้องต้นผู้ที่คุ้นเคยกับภาษาตัวหนังสือโดยทั่วไปอาจจะต้องใช้เวลาสักระยะเพื่อที่จะทำความเข้าใจการทำงานของ LabVIEW

- **ไอคอนและจุดเชื่อมต่อ**

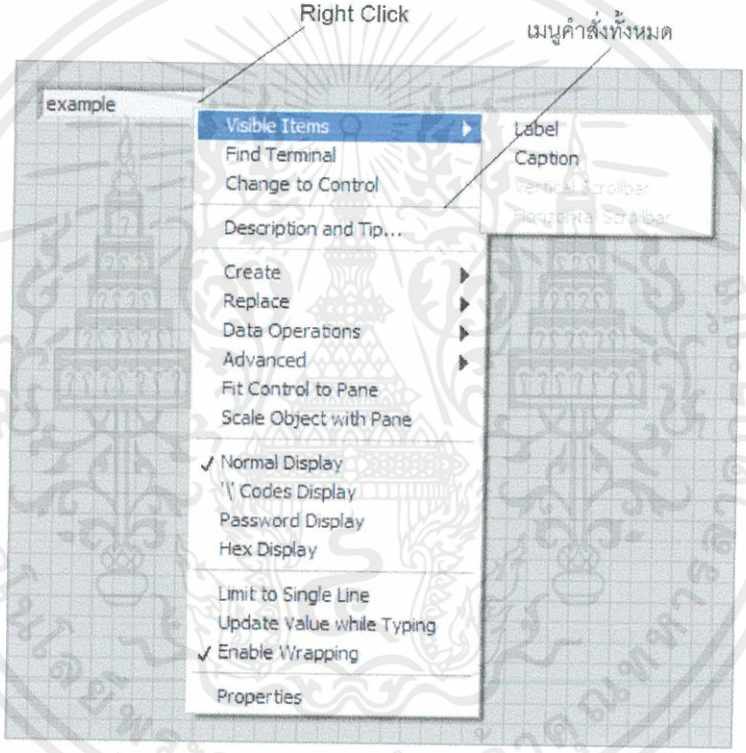
ไอคอน (Icon) และจุดเชื่อมต่อ (Connector) ในโปรแกรม LabVIEW แล้วก็คืออุปกรณ์ ตัวเดียวกัน เพียงแต่อุปกรณ์นั้นสามารถปรับเปลี่ยนคุณสมบัติให้เหมาะสมกับการใช้งานซึ่งโปรแกรมที่เห็น สัญลักษณ์ของอุปกรณ์โดยทั่วไปจะเรียกสัญลักษณ์เหล่านั้นว่า ไอคอน (Icon) เมื่อต้องการเชื่อมต่อ สายสัญญาณของไอคอนจำเป็นต้องรู้ว่าจุดเชื่อมต่อ (Connector) ของไอคอนนั้นอยู่ที่ตำแหน่งใดของไอคอน และมีคุณสมบัติอย่างไร จึงจำเป็นต้องเปลี่ยนคุณสมบัติของไอคอนให้เป็นคอนเน็คเตอร์ โดยคลิ๊กเมาส์ทางขวาที่รูปไอคอนแล้วเลือก Visible Item» Terminal ไอคอนก็จะเปลี่ยนไปเป็นเทอร์มินอลที่แสดงจุดเชื่อมต่ออย่างชัดเจน

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **เมนูคำสั่ง**

เมนูคำสั่งแบบ Shortcut

การเรียกใช้งานชุดคำสั่งในโปรแกรม LabVIEW มีอยู่สองวิธี คือ เรียกในเมนูบาร์ (Menu bar) และเรียกจากเมนูคำสั่งแบบย่อ โดยวิธีการเรียกใช้งานชุดคำสั่งแบบย่อ ทำได้โดยการลากเมาส์ ไปวางบริเวณอ๊อปเจ็ค บนหน้าต่างฟร้อนท์พาเนล หรือหน้าต่างบล็อกไดอะแกรม แล้วคลิ๊กเมาส์ทางด้านขวาซึ่งจะ ปรากฏชุดคำสั่งย่อขึ้นมาภายในคำสั่งก็จะมีชุดคำสั่งย่อย การเรียกใช้งานนั้นทำได้โดยการลากเมาส์ไปชี้ตรงคำสั่งที่ต้องการ ซึ่งจะเกิดแถบสีน้ำเงินทึบบริเวณที่เมาส์ชี้อยู่ถ้าต้องการใช้คำสั่งให้คลิ๊กเมาส์บริเวณคำสั่งที่ต้องการ ดังรูปที่ 3.32



รูปที่ 3.32 การใช้โครงสร้างของคำสั่งแบบ Shortcut

เมนูคำสั่งแบบ Pull Down

เมนูคำสั่งแบบ Pull Down ในหน้าต่างพาเนลและไดอะแกรมจะมีชุดคำสั่งที่เมนูบาร์อยู่ 8 คำสั่ง คือ File, Edit, View, Project, Operate, Tools, Window และ Help

File Edit View Project Operate Tools Window Help

รูปที่ 3.33 แสดงหน้าต่างคำสั่งแบบ Pull Down

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง File ในคำสั่ง File จะประกอบไปด้วยคำสั่งย่อยสำหรับการจัดการเกี่ยวกับการสร้าง VI การเปิดไฟล์ VI การปิด VI สำหรับหน้าต่างที่ผู้ใช้เปิดอยู่ การปิด VI ทุกหน้าต่างที่กำลังเปิดอยู่ การบันทึก การสร้าง Project การเปิดไฟล์ Project การตั้งค่าหน้ากระดาษ การปริ้นหน้าพาเนล หรือไดอะแกรม ดูคุณสมบัติของ VI ไฟล์ที่เปิดล่าสุดและสุดท้ายออกจากโปรแกรม อีกทั้งยังมีคำสั่งลัดที่สั่งเปิดจากคีย์บอร์ด เช่น Ctrl+N สำหรับสร้างหน้าต่าง VI ขึ้นมาใหม่ Ctrl+O เพื่อเปิดไฟล์ VI Ctrl+S เพื่อบันทึก VI Ctrl+P เพื่อปริ้นหน้าพาเนลหรือไดอะแกรมของ VI Ctrl+Q เพื่อปิด VI เป็นต้น

คำสั่ง Edit ในคำสั่ง Edit จะประกอบไปด้วยคำสั่งย่อยสำหรับการจัดการเกี่ยวกับการยกเลิก(Ctrl+Z) การทำซ้ำ(Ctrl+Shift+Z) การตัด(Ctrl+X) การคัดลอก(Ctrl+C) การวาง(Ctrl+V) การเลือกวัตถุทั้งหมด(Ctrl+A) การทำให้ค่าที่ผู้ใช้ต้องการนั้นเป็นค่าเริ่มต้น สร้าง SubVI การแทรกรูปภาพ การเอาสายที่ Broken ออก(Ctrl+B) คำสั่งย่อยในคำสั่ง Edit ที่ถูกใช้บ่อยครั้งที่สุดคือ คำสั่ง Undo Data Chang และ Redo ซึ่งคำสั่งนี้จะช่วยผู้ใช้ในเรื่องของการแก้ไขโปรแกรมหากการแก้ไขผิดพลาดหรือไม่ต้องการ ผู้ใช้สามารถกลับไปยังจุดที่ผู้ใช้เริ่มต้นแก้ไขได้โดยใช้คำสั่งนี้

คำสั่ง View ในคำสั่ง View จะประกอบไปด้วยชุดแสดงผลของโปรแกรม เช่น Tools Palette, Controls palette และ Function Palette และยังมีส่วนแสดงผลของค่าความผิดพลาด

คำสั่ง Project ในคำสั่ง Project ประกอบด้วยชุดคำสั่งพื้นฐานของโปรแกรมใช้สำหรับจัดการกับโปรแกรม ที่เปิดพร้อมกันหลายๆ โปรแกรม ให้เปิด ปิด บันทึก และสร้างโปรแกรม

คำสั่ง Operate ใช้สำหรับสั่งให้โปรแกรมทำงาน (Ctrl+R) หรือหยุดทำงาน (Ctrl+.) เปลี่ยนหรือเซตค่ามาตรฐานของ VI และสามารถเปลี่ยนโหมดการทำงาน จากโหมดรันไปเป็นโหมดแก้ไขได้ (Ctrl+M)

คำสั่ง Tools สำหรับเมนูนี้ ใช้เกี่ยวกับการติดต่อระหว่างโปรแกรมกับอุปกรณ์ภายนอก เปรียบเทียบ VI เช่น Instrument Real-Time Module FPGA Module DSC Module เป็นต้น การสร้างแอบพลิเคชันต่างๆ และติดต่อกับเว็บไซต์ของ NI ด้วย

คำสั่ง Window ใช้สำหรับเลือกการเปิดหน้าต่างพาเนลและไดอะแกรมสลับกันหรือเปิดทั้ง 2 หน้าจอพร้อมกันก็ได้ เช่น Show Block Diagram (Ctrl+E) แสดงหน้าต่างไดอะแกรม Tile Left and Right(Ctrl+T) แสดงหน้าต่างทั้งพาเนลและไดอะแกรมซ้ายและขวา เป็นต้น

คำสั่ง Help ใช้ในการแสดงรายละเอียดของ Tool แต่ละตัวและใช้สำหรับสืบค้นข้อมูลต่างๆ ภายในโปรแกรม LabVIEW เช่น ตัวอย่างโปรแกรม เป็นต้น

- **พาเลท**

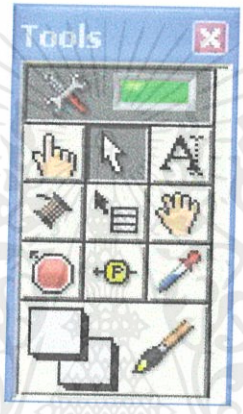
พาเลท เป็นหน้าต่างย่อยที่มีส่วนประกอบของอ๊อบเจ็ค ที่ใช้สำหรับในการเขียนโปรแกรม VI การนำไปใช้งานนั้นทำได้ง่าย เพียงแค่คลิ๊กเมาส์ทางด้านขวา บริเวณที่ว่างของหน้าต่างพาเนลหรือบล็อกไดอะแกรม ซึ่งจะปรากฏพาเลทขึ้นมา จากนั้นใช้เมาส์คลิ๊กที่อ๊อปเจ็คที่ต้องการแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

ลากไปวางบนหน้าต่างพาเนลหรือหน้าต่างไดอะแกรม พาเลทมีองค์ประกอบหลักๆ อยู่ 3 ส่วนคือ Tools, Control และ Function

- **Tools Palette**

ทูลพาเลท จะเป็นตัวบอก หรือกำหนดสถานะของเมาส์ว่าทำงานอยู่ในโหมดใด การเรียกใช้พาเลทนี้ทำได้โดยการเลือกที่เมนู View แล้วเลือกคำสั่งย่อย Tools Palette แล้วจะปรากฏ Tools Palette ขึ้นมาดังรูปที่ 3.34 ซึ่งการใช้งานนั้นเพียงแค่ผู้ใช้ นำเมาส์ไปคลิ๊กที่สัญลักษณ์ใน Tools Palette เคอร์เซอร์ของเมาส์ก็จะเปลี่ยนไปตามสัญลักษณ์ที่ทำการเลือกไว้

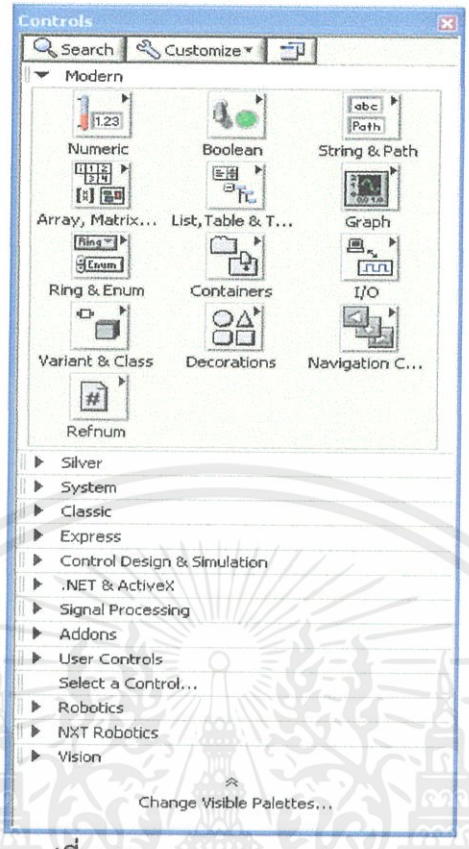


รูปที่ 3.34 แสดง Tools Palette

- **Control Palette**

มีส่วนประกอบของอุปกรณ์ที่เป็นตัวควบคุม และตัวแสดงผลที่ใช้สำหรับสร้าง VI โดยภายใน Control Palette ก็จะมีพาเลทย่อยๆ อีกเพราะว่าอุปกรณ์ใน Control Palette ก็จะมีอุปกรณ์ย่อยอีก แต่อย่างไรก็ตามสามารถเรียกดูอุปกรณ์ทั้งหมดที่อยู่ใน Control Palette ได้โดยคลิ๊กเมาส์ขวา ซึ่งจะรวมเอาอุปกรณ์ทั้งหมดไว้ในนี้ นอกจากนี้แล้วผู้ใช้ยังสามารถเปิดใช้งาน Control Palette ได้อีกทางหนึ่งคือเลือกที่เมนู View แล้วเลือก Control Palette ก็สามารถเลือกใช้ Control Palette ได้เช่นกัน ซึ่ง Control Palette แสดงดังรูปที่ 3.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

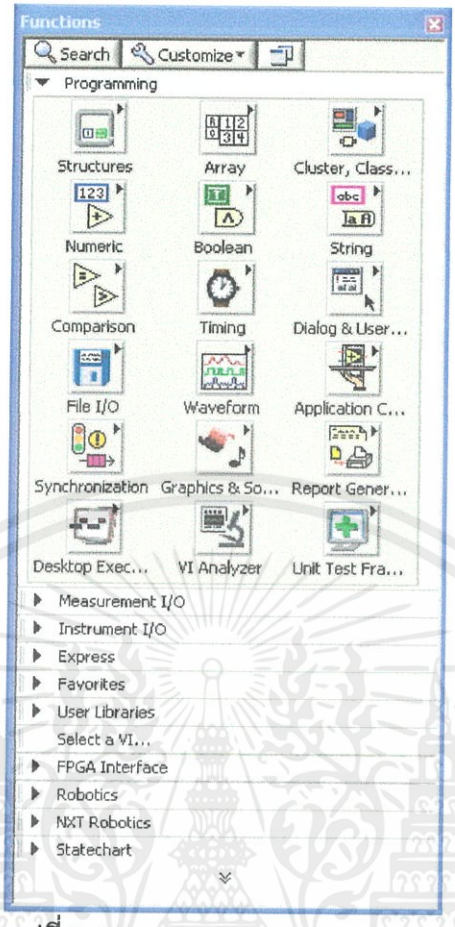


รูปที่ 3.35 แสดง Control Palette

- **Function Palette**

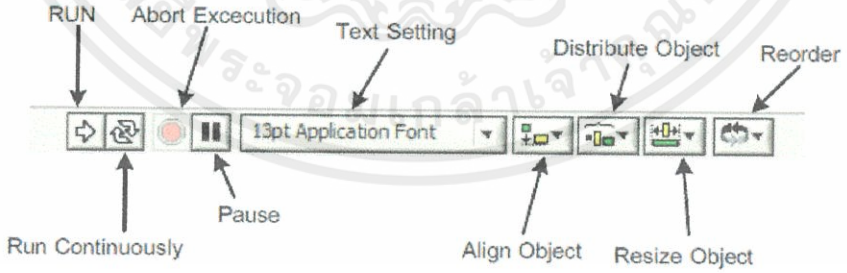
การใช้งานโดยทั่วไปของ Function Palette จะมีความคล้ายกันกับ Control Palette ภายในประกอบด้วยอุปกรณ์ในการสร้าง VI ตัวอย่างอุปกรณ์หลักๆ ใน Function Palette เช่น For Loops, While Loops และ Formula Nodes สามารถเรียกใช้งานได้ 2 วิธีเช่นเดียวกันกับการเรียกใช้งาน Control Palette คือ เลือกที่เมนู View ในหน้าต่างของบล็อกไดอะแกรม แล้วเลือก Function Palette หรือคลิ๊กเมาส์ทางด้านขวาบริเวณพื้นที่ว่างในหน้าต่างไดอะแกรม ก็จะปรากฏ Function Palette เช่นกัน ซึ่ง Function Palette แสดงดังรูปที่ 3.36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.36 แสดง Function Palette

- **เครื่องมือ(Tool)**
  - **เครื่องมือของหน้าต่างพาเนล**



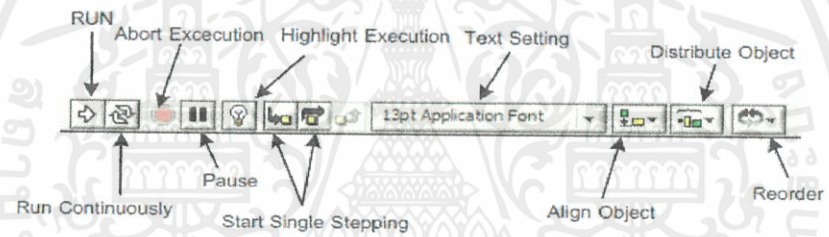
รูปที่ 3.37 แสดงทูลบาร์ในหน้าต่างพาเนล

RUN	ใช้สำหรับรันโปรแกรม
RUN Continuously	ใช้สำหรับรันโปรแกรมอย่างต่อเนื่อง
Abort Execution	หรือเรียกอีกอย่างหนึ่งว่าปุ่ม Stop ใช้สำหรับหยุดการรันโปรแกรม
Pause	ใช้สำหรับหยุดการรันโปรแกรมชั่วคราวเพื่อตรวจสอบการทำงานของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	โปรแกรม เมื่อต้องการรันโปรแกรมต่อให้กดปุ่มนี้ซ้ำอีกครั้งโปรแกรมก็จะรันต่อไป
Text setting	ใช้สำหรับจัดการกับตัวอักษรทั้งหมดไม่ว่าจะเป็น ฟร้อนท์ ขนาด รูปแบบตัวอักษรรวมไปถึงการกำหนดสีให้กับตัวอักษรด้วย
Align Object	ใช้จัดObject ต่างๆ ที่ใช้เขียนโปรแกรมให้อยู่ในแนวเดียวกันทั้งแนวตั้งและแนวนอนเพื่อความเป็นระเบียบ
Distribute Object	ใช้สำหรับการกำหนดระยะห่างระหว่าง Objects อัตโนมัติ
Resize Object	ใช้สำหรับปรับขนาดของ Object
Reorder	ใช้สำหรับกำหนดลำดับก่อนหลังสำหรับการวางซ้อนทับกันของ Object
Context Help	ใช้แสดงคุณสมบัติหรือรายละเอียดของ Object แต่ละตัว

- **เครื่องมือของหน้าต่างบล็อกไดอะแกรม**



รูปที่ 3.38 แสดงทูลบาร์ในหน้าต่างบล็อกไดอะแกรม

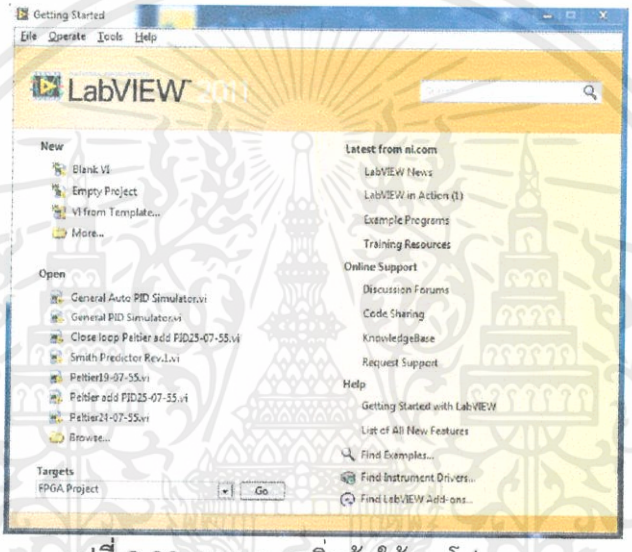
RUN	ใช้สำหรับรันโปรแกรม
RUN Continuously	ใช้สำหรับรันโปรแกรมอย่างต่อเนื่อง
Abort Execution	หรือเรียกอีกอย่างหนึ่งว่าปุ่ม Stop ใช้สำหรับหยุดการรันโปรแกรม
Pause	ใช้สำหรับหยุดการรันโปรแกรมชั่วคราวเพื่อตรวจสอบการทำงานของโปรแกรม เมื่อต้องการรันโปรแกรมต่อให้กดปุ่มนี้ซ้ำอีกครั้งโปรแกรมก็จะรันต่อไป
Highlight Execution	แสดงลำดับขั้นการทำงานของโปรแกรม โดยแสดงให้เห็นทิศทางการไหลของสัญญาณ
Start Single Step	ใช้สำหรับสั่งให้โปรแกรมทำงานทีละคำสั่ง
Text Setting	ใช้สำหรับจัดการกับตัวอักษรทั้งหมดไม่ว่าจะเป็น ฟร้อนท์ ขนาด รูปแบบตัวอักษร รวมไปถึงการกำหนดสีให้กับตัวอักษรด้วย
Align Object	ใช้จัด Object ต่าง ๆ ที่ใช้เขียนโปรแกรมให้อยู่ในแนวเดียวกันทั้งแนวตั้งและแนวนอนเพื่อความเป็นระเบียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Distribute Object	ใช้สำหรับการกำหนดระยะห่างระหว่าง Objects อัตโนมัติ
Reorder	ใช้สำหรับกำหนดลำดับก่อนหลังสำหรับการวางซ้อนทับกันของ Object
Context Help	ใช้แสดงคุณสมบัติหรือรายละเอียดของ Object แต่ละตัว

- **การเปิดและการบันทึกไฟล์**

การเปิดใช้งานและการบันทึกไฟล์ในโปรแกรมLabVIEW มีด้วยกันหลายวิธีด้วยกัน ซึ่งวิธีการเรียกใช้งาน และการบันทึกแต่ละแบบก็มีรูปแบบที่แตกต่างกันออกไป ซึ่งผู้ใช้งานสามารถเรียกใช้งานได้ตามความเหมาะสม เพราะไม่ว่าจะเปิดด้วยวิธีใดก็สามารถใช้งานได้เช่นเดียวกัน



รูปที่ 3.39 แสดงการเริ่มต้นใช้งานโปรแกรม

- **การเปิดไฟล์**

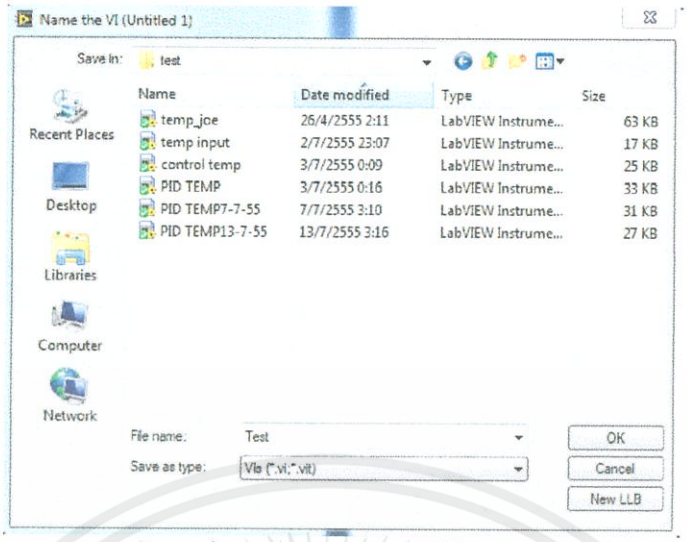
การเปิดไฟล์ใหม่ จากรูปที่ 3.39 คลิ๊กเลือกไปที่ Blank VI ก็จะได้หน้าต่างพาเนลและหน้าต่างไดอะแกรมใหม่ขึ้นมา

การเรียกโปรแกรมที่สร้างไว้แล้วมาใช้งานทำได้โดยเปิดโปรแกรม LabVIEW แล้วเลือกที่เมนู File»Open จะปรากฏไดอะล็อกบล็อกที่บันทึกโปรแกรมไว้แล้วทำการเลือกไฟล์ที่ต้องการก็จะได้โปรแกรมที่บันทึกไว้มาใช้งานได้

- **การบันทึกไฟล์**

การบันทึกไฟล์ VI สามารถทำได้โดยเลือก Save, Save As, Save All, หรือ Save with Option จากเมนูไฟล์ หลังจากนั้นจะปรากฏหน้าต่างสำหรับบันทึกขึ้นมา แล้วทำการเลือกโฟลเดอร์ ที่ต้องการบันทึกไฟล์ ตั้งชื่อไฟล์แล้วทำการบันทึก ไฟล์ที่บันทึกไว้จะมีนามสกุล *.vi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



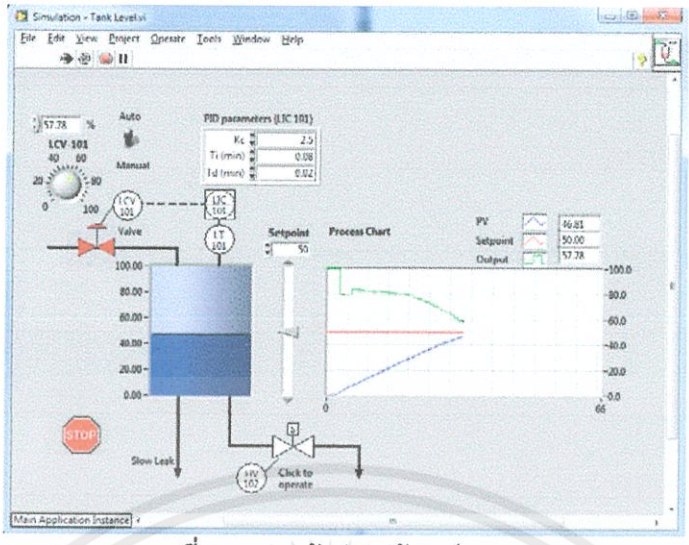
รูปที่ 3.40 แสดงไดอะล็อกบล็อกสำหรับเปิดใช้งานโปรแกรม LabVIEW

3.3.2 พื้นฐานการเขียนโปรแกรมภาษากราฟฟิก

- **ฟร้อนท์พาเนล**

หน้าต่างฟร้อนท์พาเนลสำหรับเขียนโปรแกรม LabVIEW นั้นจะเป็นหน้าต่างที่ใช้ติดต่อกับผู้ใช้งาน ซึ่งในหน้าต่างพาเนลจะประกอบไปด้วยอุปกรณ์อยู่ 2 ส่วน คือ อุปกรณ์สำหรับการควบคุม (Control) ยกตัวอย่างเช่น knobs, switches และอุปกรณ์อีกอย่างหนึ่งก็คืออุปกรณ์สำหรับแสดงผล (Indicator) เช่น LED, Text Setting, Graph และ Chart เป็นต้น ส่วนอุปกรณ์อื่นๆ ที่ไม่ใช่อุปกรณ์ควบคุมและแสดงผลจะแสดงไอคอนอยู่ในหน้าต่างบล็อกไดอะแกรม สำหรับอุปกรณ์ควบคุมในภาษากราฟฟิกจะหมายถึง “อินพุท” และอุปกรณ์แสดงผลจะหมายถึง “เอาท์พุท” การวางอุปกรณ์ควบคุม และอุปกรณ์แสดงผล ลงไปบนหน้าต่างฟร้อนท์พาเนลทำได้โดยการเลือกอุปกรณ์ที่ต้องการบนหน้าต่าง Control Palette แล้วใช้เมาส์ลากไปวางบนหน้าต่างพาเนล ซึ่งการเรียกหน้าต่าง Control Palette ทำได้โดยการคลิ๊กเมาส์ทางด้านขวาบริเวณพื้นที่ว่างของหน้าต่างพาเนลก็จะปรากฏหน้าต่าง Control Palette ที่ประกอบไปด้วยอุปกรณ์ควบคุมและอุปกรณ์แสดงผลขึ้นมา สำหรับการเรียกใช้งานหน้าต่าง Control Palette อีกวิธีหนึ่งก็คือ เลือกที่เมนู Window ของแท็บเมนูบาร์ แล้วเลือก Show Control Palette จะปรากฏหน้าต่างของ Controls Palette ขึ้นมาให้เลือกใช้งานเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.41 หน้าต่างฟร้อนท์พาเนล

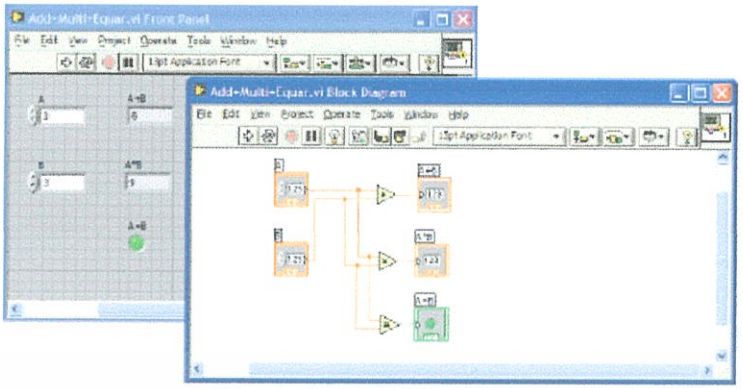
- **บล็อกไดอะแกรม**

ในหน้าต่างบล็อกไดอะแกรมจะประกอบไปด้วยอุปกรณ์ ที่ใช้สำหรับในการสร้างโปรแกรม ถ้าเปรียบเทียบการภาษาอื่นๆ อุปกรณ์ที่วางลงไปบนหน้าต่างบล็อกไดอะแกรมจะถือว่าเป็น Source Code การทำให้อุปกรณ์ต่างๆ ที่อยู่ในหน้าต่างของบล็อกไดอะแกรมทำงานสัมพันธ์กัน และเป็นไปตามที่ผู้เขียนต้องการนั้นสามารถทำได้ ด้วยการลากสายสัญญาณ จากจุดเชื่อมต่อของอุปกรณ์แต่ละตัวให้ถึงกัน และกำหนดค่าหรือคุณสมบัติของอุปกรณ์แต่ละตัว ก็จะทำให้โปรแกรมทำงานตามที่ผู้เขียนต้องการ คุณสมบัติของอุปกรณ์ที่อยู่บนหน้าต่างบล็อกไดอะแกรมมีอยู่ 3 คุณสมบัติ คือ Node, Terminal และ Wires

- Nodes

เป็นตัวกระทำให้โปรแกรมทำงานได้ตามที่ต้องการ ถ้าเปรียบเทียบกับโปรแกรมทั่วไป Nodes ก็เปรียบเสมือน statement, functions, และ subroutines โหนดในโปรแกรม LabVIEW มี 3 ชนิด คือ Function, Sub VI โหนด และ Structure โหนดแบบ Function จะสร้างขึ้นมาเพื่อให้สามารถทำงานขั้นพื้นฐานได้ เช่น การบวกเลข การคูณเลข เป็นต้น ดังเช่นการบวก และการคูณในรูปที่ 3.42 ก็เป็นฟังก์ชันโหนดแบบหนึ่งเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.42 หน้าต่างบล็อกไดอะแกรม

- **Terminal**

เป็นทางผ่านของข้อมูลระหว่างบล็อกไดอะแกรม กับพร้อนท์พาเนล และระหว่างโหนดแต่ละโหนดในบล็อกไดอะแกรม เทอร์มินอลเปรียบได้กับค่าพารามิเตอร์และค่าคงที่ในโปรแกรมภาษาอื่น เทอร์มินอลแต่ละชนิดก็มีความแตกต่างกันซึ่งในโปรแกรม LabVIEW นั้นจะมีเทอร์มินอล 4 ชนิดคือ Control and Indicator Terminals, node Terminal, Constants, และ เทอร์มินอลพิเศษ ซึ่งเทอร์มินอลแต่ละตัวสามารถลากสายสัญญาณเพื่อเป็นทางผ่านของข้อมูล สำหรับตัวอย่างในกรณีของ Control and Indicator Terminals เมื่อมีข้อมูลแบบตัวเลขผ่านเข้ายังบล็อกไดอะแกรมโดยผ่านช่องทาง Control Terminals แล้วทำการประมวลผลในหน้าต่างบล็อกไดอะแกรม เมื่อโปรแกรมทำการประมวลผลเสร็จเรียบร้อย ก็จะส่งข้อมูลจากหน้าต่างบล็อกไดอะแกรม ไปยังหน้าต่างฟร้อนท์พาเนล โดยผ่านช่องทาง Indicator Terminal

- **Wires**

เป็นทางผ่านของข้อมูลระหว่าง Terminal โดยการลากสายสัญญาณทำได้โดยการเลือกเครื่องมือที่ชื่อว่า connect wire ที่หน้าต่าง Tool Palette ที่มีลักษณะคล้ายกับหลอดด้าย เมาส์จะเปลี่ยนเป็นรูปหลอดด้าย จากนั้นลากไปวางตรงจุดต่อของเทอร์มินอลแรกที่ต้องการลาก และคลิ๊กเมาส์หนึ่งครั้งแล้วลากไปยังจุดต่อของเทอร์มินอลปลายทางที่ต้องการ คลิ๊กเมาส์อีกหนึ่งครั้งก็เสร็จการเชื่อมต่อสัญญาณ

- **การสร้าง VI**

ในส่วนของหัวข้อนี้จะกล่าวถึงขั้นตอน และวิธีการสร้าง VI โดยมีวัตถุประสงค์เพื่อให้สามารถบวกเลขและแสดงผลลัพธ์ได้, ให้สามารถคูณเลขและแสดงผลลัพธ์ได้ และเปรียบเทียบกันระหว่างอินพุทสองอินพุทถ้ามีค่าเท่ากันให้หลอด LED ติด เริ่มต้นพิจารณาที่หน้าต่าง ฟร้อนท์พาเนลในรูปที่ 3.42 ในหน้าต่างฟร้อนท์พาเนลจะมีอินพุทที่เป็นดิจิตอล 2 อินพุทกำหนดให้เป็น *A*

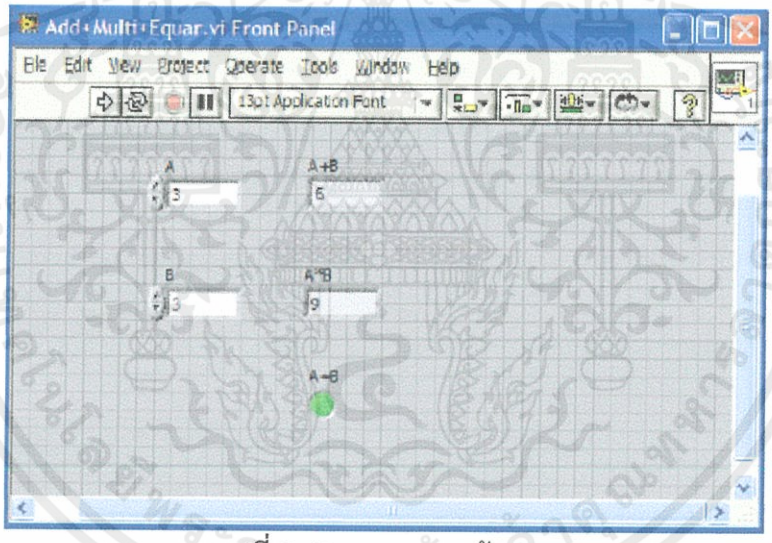
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ B และมีตัวแสดงผลแบบดิจิตอล 2 ตัว กำหนดให้เป็น $A + B$ และ $A \times B$ และมีตัวแสดงผลแบบ LED ซึ่งจะติดเมื่ออินพุท A กับ B มีค่าเท่ากันโดย ขั้นตอน และวิธีการสร้าง VI มีขั้นตอนดังต่อไปนี้

1. สร้างหน้าต่างฟร้อนท์พาเนลขึ้นมาใหม่โดยเลือกที่ file เลือก New

2. สร้าง Numerical Control และ Indicator สำหรับป้อนอินพุทและแสดงผลอย่างละ 2 ตัวและสร้างเครื่องมือในการบวก การคูณ และเปรียบเทียบค่า

- เลือก Numerical Control จาก Numerical Control Palette ซึ่งจะเป็นพาเลทย่อย ของ Controls Palette
- ใช้เมาส์คลิ๊กที่ Numeric Control แล้วลากไปวางหน้าต่างฟร้อนท์พาเนล
- ชื่อของอุปกรณ์จะมีชื่อว่า Numeric สามารถที่จะเปลี่ยนได้โดยดับเบิ้ลคลิ๊กเมาส์บริเวณตัวอักษรให้มีแถบสีดำทึบปรากฏขึ้น จากนั้นจึงจะสามารถเปลี่ยนชื่อได้ตามความต้องการ ถ้า Control หรือ Indicator ไม่ปรากฏแถบลาเบล ให้คลิ๊กเมาส์ทางด้านขวาบริเวณอุปกรณ์เลือกที่ Visible Item แล้ว เลือกที่ Label แถบลาเบลก็จะปรากฏขึ้นมา

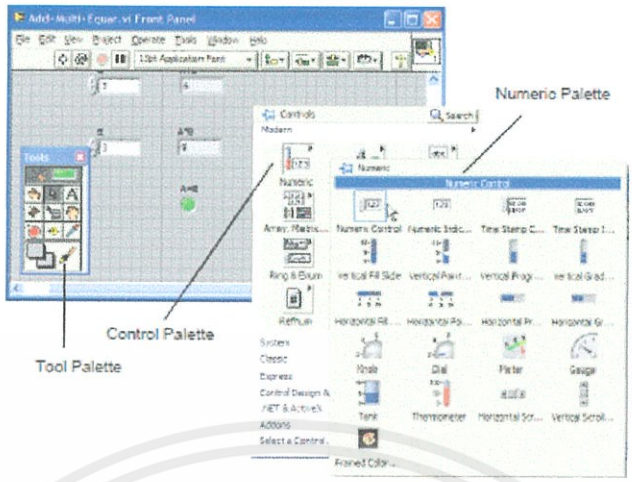


รูปที่ 3.43 แสดงการสร้าง VI

3. สร้างหลอด LED ถ้าต้องการให้หลอด LED ติดเมื่อค่าของอินพุททั้งสองมีค่าเท่ากันและดับเมื่ออินพุททั้งสองต่างกัน

- เลือก Round LED จาก LED Palette ซึ่งเป็นพาเลทย่อยของ Controls Palette คลิ๊กเมาส์ที่ Round LED แล้วลากไปวางบริเวณที่ว่างบนหน้าต่างฟร้อนท์พาเนล
- การตั้งชื่อสามารถทำได้เช่นเดียวกันกับการตั้งชื่อของ Numeric Control

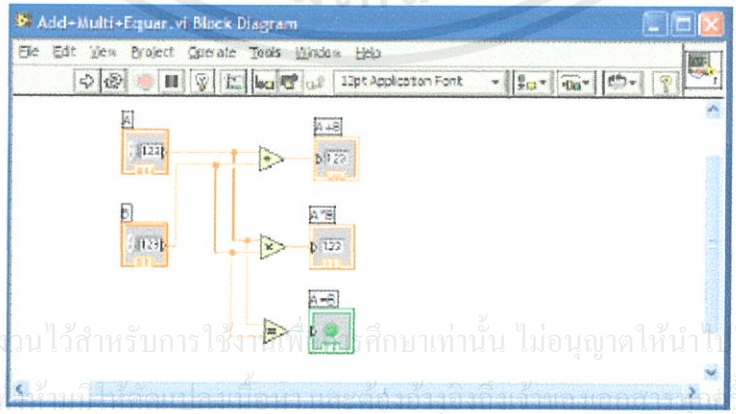
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.44 แสดงการเลือกอุปกรณ์ไปวางที่ฟร้อนท์พาเนล

ในการสร้างโปรแกรม VI เมื่อทำการลากอุปกรณ์ควบคุม และแสดงผลมาวางบนหน้าต่างฟร้อนท์พาเนล ในเวลาเดียวกันจะปรากฏเทอร์มินอลของอุปกรณ์นั้นๆ ขึ้นที่หน้าต่างบล็อกไดอะแกรม ซึ่งเทอร์มินอลที่เกิดขึ้นนี้เอง จะเป็นตัวที่ถูกนำมาลากสายสัญญาณเชื่อมต่อกัน ให้โปรแกรมทำงานเป็นไปตามเงื่อนไขที่ผู้เขียนโปรแกรมต้องการ

หลังจากที่ได้วางอุปกรณ์ควบคุม และแสดงผลลงบนหน้าต่างฟร้อนท์พาเนลเรียบร้อยแล้ว ต่อไปจึงจะมาพิจารณาในหน้าต่างบล็อกไดอะแกรม ดังที่กล่าวข้างต้นว่าอุปกรณ์ควบคุม และแสดงผลที่ได้วางลงไปบนหน้าต่างฟร้อนท์พาเนลนั้น จะทำให้เกิดเทอร์มินอลของอุปกรณ์เหล่านั้นที่หน้าต่างบล็อกไดอะแกรมด้วย แต่การสร้างโปรแกรมไม่ได้มีเพียงอุปกรณ์ควบคุม และแสดงผลเท่านั้น แต่ยังมีอุปกรณ์อื่นๆ ร่วมด้วย เช่น อุปกรณ์การกระทำทางคณิตศาสตร์ (บวก ลบ คูณ หาร ฯลฯ) ซึ่งอุปกรณ์เหล่านี้จะไม่สามารถสร้าง และไม่ปรากฏในหน้าต่างฟร้อนท์พาเนล แต่จะสร้างและปรากฏในหน้าต่างบล็อกไดอะแกรมเท่านั้น

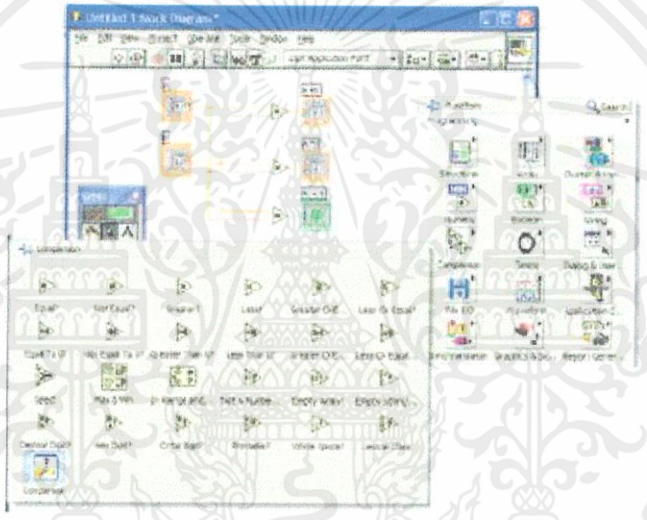


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.45 แสดงเทอร์มินอลที่เกิดขึ้นในบล็อกไดอะแกรม

1. การเปิดหน้าต่างบล็อกไดอะแกรมทำได้โดยเลือกที่เมนู Window แล้วเลือก Show Bolck Diagram จะปรากฏหน้าต่างไดอะแกรมขึ้นมา ซึ่งแสดงดังรูปที่ 3.45 เป็นหน้าต่างไดอะแกรมที่เขียนโปรแกรมเสร็จเรียบร้อยแล้ว

2. หน้าต่างบล็อกไดอะแกรมในครั้งแรกที่เปิดมาจะมีเพียงเทอร์มินอลของอุปกรณ์ควบคุมและแสดงผลเท่านั้น หากต้องการที่จะสร้างฟังก์ชันสำหรับการบวก การคูณ และเปรียบเท่ากัน ซึ่งจะทำได้โดยเลือกฟังก์ชันการบวก และการคูณจาก Function Palette» Numeric» Add, Multiply แล้วก็ลากฟังก์ชันไปวางในหน้าต่างไดอะแกรม ซึ่งจะสังเกตเห็นได้ว่าเมื่อวางฟังก์ชันลงไปบนหน้าต่างบล็อกไดอะแกรมแล้วจะไม่ปรากฏลาเบลของฟังก์ชันนั้นๆขึ้นมา สามารถเรียกลาเบลให้ปรากฏได้โดยคลิ๊กขวาที่ฟังก์ชันที่ต้องการจะปรากฏ pop-up ขึ้นมาเลือก Visible Item แล้วก็เลือกที่ Label ก็จะปรากฏลาเบลของฟังก์ชันดังกล่าวขึ้นมาสามารถที่จะตั้งชื่อฟังก์ชันได้แสดงให้เห็นดังรูปที่ 3.46



รูปที่ 3.46 แสดงฟังก์ชันในบล็อกไดอะแกรม

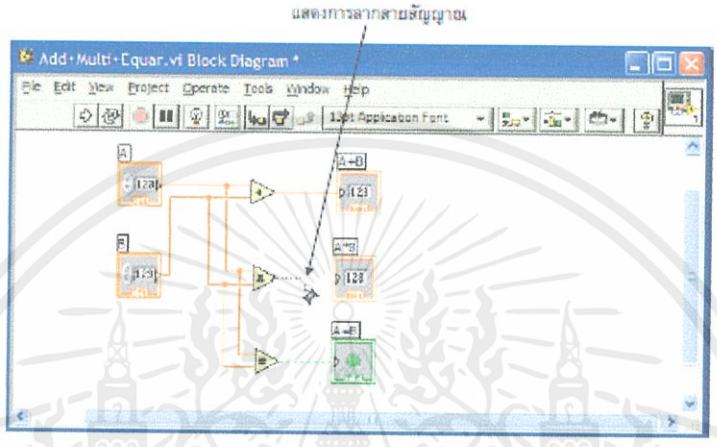
3. เลือกฟังก์ชันการเปรียบเทียบ (Equal) จาก FunctionPalette»Comparison»Equal ซึ่งเป็นตัวเปรียบเทียบระหว่างอินพุททั้งสองโดยถ้าอินพุททั้งสองมีค่าเท่ากันให้หลอด LED ติด และถ้าไม่เท่ากันให้หลอด LED จะดับ

4. หลังจากที่วางอุปกรณ์และฟังก์ชันต่างๆ ครบแล้วขั้นตอนต่อไปเป็นขั้นตอนในการลากสายสัญญาณเพื่อให้โปรแกรมทำงานตามเงื่อนไขที่กำหนดข้างต้นโดยเลือกเครื่องมือ Connect wire จาก Tool Palette หรือเลือกที่ Automatic Tool Select จาก Tool Palette เช่นกันหลังจากนั้นเริ่มการลากสายสัญญาณเริ่มจากนำเมาส์ไปวางที่คอนเน็คเตอร์ของ Control Terminal แล้วลากไปต่อกับอินพุทคอนเน็คเตอร์ของฟังก์ชัน จากนั้นทำการลากสายสัญญาณ จากเอาท์พุทคอนเน็คเตอร์ของฟังก์ชัน แล้วลากไปต่อกับอินพุทคอนเน็คเตอร์ของ Indicator Terminal ทำการลากสายสัญญาณทั้งหมดให้ครบตามเงื่อนไขซึ่งแสดงดังรูปที่ 3.46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หลังจากนั้นเรียกหน้าต่างฟร้อนท์พาเนลขึ้นมาเพื่อทำการบันทึกไฟล์ ซึ่งสำหรับตัวอย่างจะขอบันทึกชื่อไฟล์ Create VI.vi โดยเลือกจาก File» Save

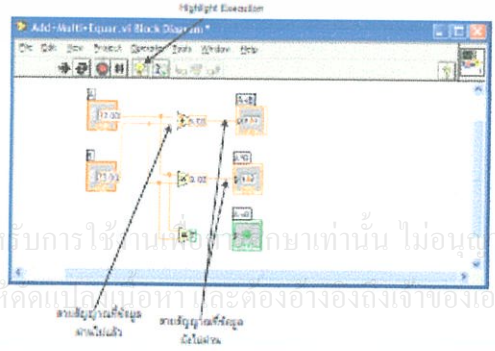
6. ทำการทดสอบโปรแกรมโดยสั่ง RUN โปรแกรมแบบต่อเนื่องแล้วเปลี่ยนแปลงอินพุททั้งสองไปแล้ว สังเกตว่าเอาท์พุทของการบวก และการคูณถูกต้องหรือไม่ และสังเกตถ้าอินพุททั้งสองมีค่าเท่ากันหลอด LED ติดหรือไม่ ถ้าตรวจเงื่อนไขต่างๆ ถูกต้องแล้ว ก็เสร็จสิ้นการสร้าง VI



รูปที่ 3.47 แสดงวิธีการลากสายสัญญาณ

■ **การไหลของข้อมูลใน VI**

สำหรับในโปรแกรม LabVIEW สามารถที่จะดูทิศทางการไหลของสัญญาณการทำงานของโปรแกรมที่ได้สร้างขึ้นมาได้โดยเครื่องมือที่ใช้ คือ Highlight Execution ซึ่งการใช้งานทำได้โดยการคลิ๊กที่สัญลักษณ์ของเครื่องมือ Highlight Execution (ลักษณะคล้ายหลอดไฟที่แถบ Tool bar) ในหน้าต่างบล็อกไดอะแกรมจากนั้นทำการกดปุ่มรันโปรแกรมจะสังเกตเห็นจุดกลมๆเล็กๆวิ่งจากอินพุทผ่านฟังก์ชันต่างๆ ตามลำดับการทำงานก่อนหลังยกตัวอย่างดังรูปที่ 3.48 แสดงฟังก์ชันการเปรียบเทียบ การคูณ และการบวก จะสังเกตเห็นว่าโปรแกรมจะทำการเปรียบเทียบอินพุทก่อนแล้วจึงทำการคูณและการบวกตามลำดับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.48 แสดงการรันโดยใช้ Highlight Execution

3.3.3 การสร้าง VI ด้วย VI แบบเร็ว (Express VI)

ในหัวข้อนี้จะกล่าวถึงการสร้างโปรแกรม VI ด้วย VI แบบเร็ว (Express VI) โดยต้องการที่จะสร้าง VI ที่สามารถสร้างสัญญาณคลื่นซายน์ที่สามารถกำหนดขนาดและความถี่ได้ และให้แสดงกราฟสัญญาณที่หน้าต่างฟร้อนท์พาเนลซึ่งในโปรแกรม LabVIEW มีรูปแบบที่สร้างไว้สำหรับช่วยให้ผู้เขียนโปรแกรมสะดวกในการใช้งาน

เลือกที่ New»VI from Template» Simulation» Generate and Display เพื่อที่จะสร้าง VI สำหรับสร้างสัญญาณคลื่นซายน์

1. คลิ๊ก OK เพื่อเปิดรูปแบบในการสร้างสัญญาณคลื่นซายน์

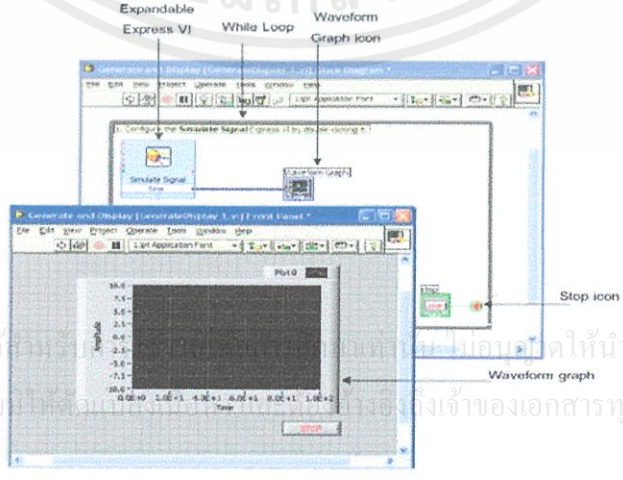
2. ในรูปที่ 3.49 สังเกตที่หน้าต่างฟร้อนท์พาเนลซึ่งจะสังเกตเห็นได้ว่าจะมีอุปกรณ์แสดงผลแบบกราฟ และปุ่มหยุดการทำงานปรากฏอยู่ และที่ Title bar จะแสดงชื่อของโปรแกรมชื่อ Generate and Display [Untitled] VI.

3. ที่หน้าต่างบล็อกไดอะแกรม แสดงในรูปที่ 3.49 จะประกอบไปด้วยไอคอนของ Simulate Signal Express VI, ไอคอน Waveform Graph, ไอคอน Stop Button และ While Loop ที่ Title bar จะมี ชื่อของ VI คือ Generate and Display [Untitled] VI. เช่นเดียวกับหน้าต่างฟร้อนท์พาเนล

4. ที่หน้าต่างฟร้อนท์พาเนลกดปุ่มรันโปรแกรมที่แถบ Tool bar ที่กราฟแสดงผลจะปรากฏเส้นสัญญาณคลื่นซายน์

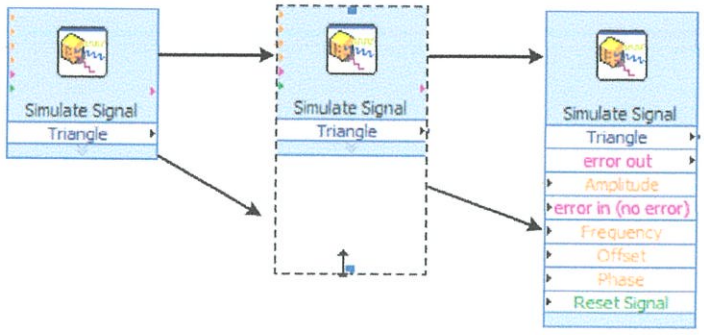
5. ถ้าต้องการหยุดการทำงานของโปรแกรมสามารถกดปุ่ม Stop ที่อยู่ทางด้านล่างขวาบนหน้าต่างฟร้อนท์พาเนล

6. สามารถเพิ่มอุปกรณ์อื่นเข้าไปในโปรแกรม เพื่อให้สามารถทำงานตามที่ต้องการได้มากขึ้น และสะดวกมากขึ้นในโปรแกรมนี้จะเพิ่มอุปกรณ์ควบคุมเข้าไป เพื่อปรับขนาดของสัญญาณได้ตามความต้องการ



รูปที่ 3.49 แสดงการสร้าง Express VI

เอกสารนี้เป็นเอกสารที่สงวนไว้ ... ตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม ... ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.50 การเพิ่มคุณสมบัติไอคอนของ Express VI

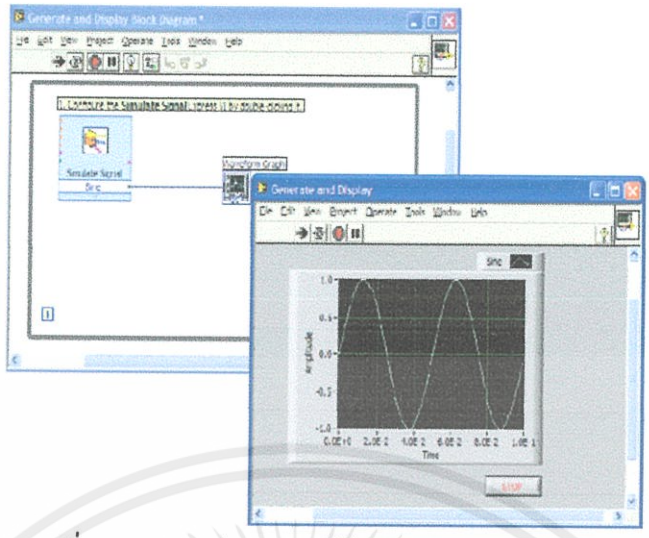
7. เลือกอุปกรณ์ knob จาก Numeric Controls Palette ซึ่งเป็นพาเลทย่อยของ Controls Palette ลากไปวางที่หน้าต่างฟร้อนท์พาเนลเปลี่ยนชื่อให้เป็น Amplitude

8. ที่ไอคอนของ Simulate signal Express VI ดังแสดงในรูปที่ 3.50 สามารถขยายไอคอนที่เพิ่ม คุณสมบัติในการใช้งานให้ได้มากยิ่งขึ้นซึ่งในไอคอนดังกล่าวจะมีฟังก์ชันการทำงานที่ซ่อนอยู่

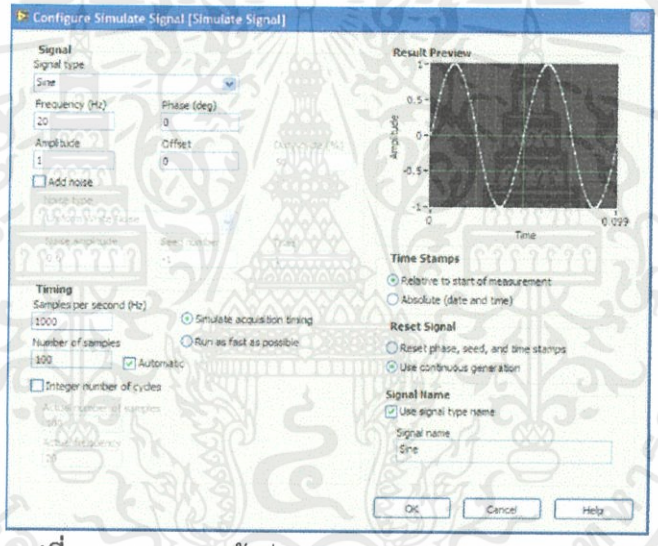
9. ในหน้าต่างบล็อกไดอะแกรม ลากสายสัญญาณจาก knob ไปต่อกับอินพุท Amplitude ของ Simulate Signal Express VI จากนั้นกลับไปยังหน้าฟร้อนท์พาเนล ทำการรันโปรแกรมแล้วทำการปรับค่าของแอมปลิจูดที่ปุ่ม knob ที่ค่าต่างๆ สังเกตผลทางด้านเอาท์พุทที่แสดงในกราฟสัญญาณค่าแอมปลิจูดจะเปลี่ยนค่าไปตามที่ปรับปุ่ม knob

10. ในการกำหนดคุณสมบัติของ Simulate signal Express VI ทำได้โดยคลิ๊กเมาส์ทางด้านขวาที่ ไอคอน Simulate signal Express แล้วเลือกที่เมนู Properties จะปรากฏหน้าต่างที่มีชื่อว่า Configure Simulate Signal [Simulate Signal] ขึ้นมาดังแสดงในรูปที่ 3.52 ซึ่งในหน้าต่างนี้ประกอบไปด้วย Signal Type ใช้สำหรับให้เลือกรูปแบบของสัญญาณซึ่งประกอบไปด้วย Sine, Square, Triangle, Sawtooth และ DC และมี Result Preview อยู่ทางด้านขวาซึ่งจะแสดงลักษณะของเส้นสัญญาณตามที่ เราเลือก Signal Type นอกจากนี้ยังมียังมี Frequency (Hz) สำหรับปรับความถี่และ Amplitude สำหรับปรับขนาดของ Amplitude

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.51 การต่ออุปกรณ์ใช้งานร่วมกับ Express VI



รูปที่ 3.52 แสดงหน้าต่าง Configure Simulate Signal

11. การใช้งาน Configure Simulate Signal [Simulate Signal] คือ เลือกชนิดของสัญญาณ กำหนดขนาดแอมปลิจูด และกำหนดความถี่ที่ตามความต้องการโดยสามารถดูลักษณะของกราฟที่กำหนดไว้ได้ที่ Result Preview ถ้าลักษณะของกราฟเป็นไปตามที่ต้องการแล้ว กดปุ่ม OK ซึ่งขณะนี้ที่ไอคอน Simulate Signal Express VI จะมีคุณสมบัติตามที่เรากำหนด

3.3.4 การแก้ไขและการดีบัคโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไปดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

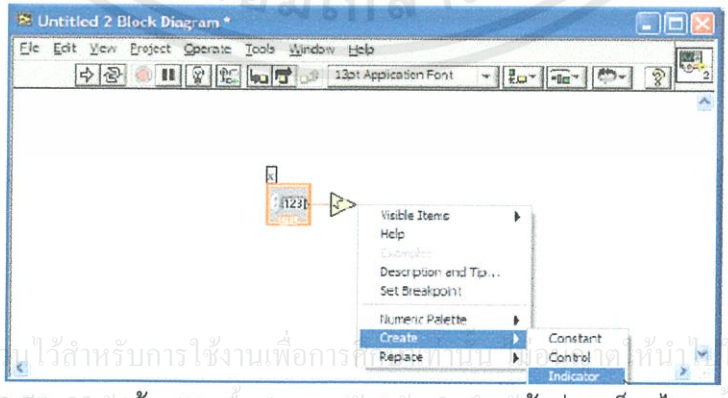
ในการเขียนโปรแกรมคอมพิวเตอร์ ไม่ว่าจะเขียนด้วยโปรแกรมภาษาใดก็ตามย่อมเกิดการผิดพลาดในการเขียนโปรแกรมได้ตลอดเวลาและต้องการการแก้ไขให้ถูกต้องซึ่งการแก้ไขและการตรวจสอบความผิดพลาดของแต่ละโปรแกรมก็จะแตกต่างกันไป สำหรับโปรแกรม LabVIEW ก็มีวิธีและเครื่องมือที่ช่วยใน การแก้ไขและตรวจสอบความผิดพลาดของโปรแกรม

ในหัวข้อนี้จะกล่าวถึงวิธีการสร้าง การเลือก การลบ การเคลื่อนย้าย และการจัดหมวดหมู่ของอุปกรณ์บนหน้าต่างฟร้อนท์พาเนล และหน้าต่างบล็อกไดอะแกรม ความสำคัญในการเลือก และลบสายสัญญาณ การตรวจสอบโปรแกรมด้วย Highlight Execution และ Single-Step Code

- **เทคนิคการแก้ไขโปรแกรม**
  - **การสร้าง Controls และ Indicator บนหน้าต่างบล็อกไดอะแกรม**

โดยปกติแล้วการสร้างตัวควบคุม และตัวแสดงผลซึ่งจะทำการสร้างจากหน้าต่างฟร้อนท์พาเนล ดังที่ได้กล่าวถึงวิธีการสร้างมาแล้วในหัวข้อการสร้าง VI สำหรับหัวข้อนี้ จะกล่าวถึงวิธีการสร้างตัวควบคุม และตัวแสดงผลในหน้าต่างบล็อกไดอะแกรม โดยจะสร้างโปรแกรมสำหรับหารากของตัวเลขดิจิตอลโดยมีวิธีการสร้างดังนี้

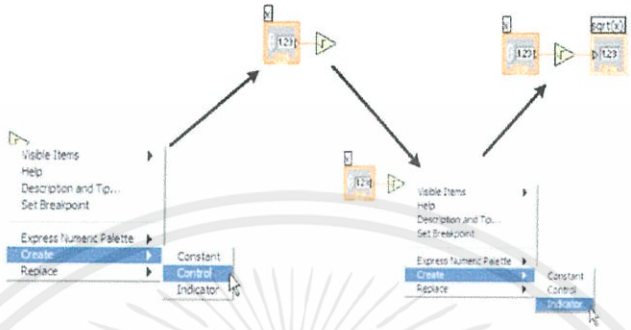
1. เปิดโปรแกรมสำหรับสร้าง VI แล้วเปิดหน้าต่างบล็อกไดอะแกรมขึ้นมา
2. คลิ๊กเมาส์ทางด้านขวาเลือก Function Palette» Numeric» Square Root แล้วลากมาวางที่หน้าต่างบล็อกไดอะแกรม
3. ตอนนี้ต้องการที่จะสร้าง Controls Terminal และ Indicator Terminal ให้กับฟังก์ชัน Square Root (แสดงดังรูปที่ 3.53) เริ่มต้นจากนำเมาส์ไปวางที่จุดเชื่อมต่อทางซ้ายมือของ ฟังก์ชัน Square Root แล้วคลิ๊กขวา เลือก Create แล้วเลือก control ก็จะได้ Controls Terminal ที่มีการต่อสายสัญญาณกับจุดต่อสัญญาณของฟังก์ชัน Square Root
4. จากนั้นทำการสร้าง Indicator Terminal โดยการนำเมาส์ไปวางที่จุดเชื่อมต่อทางด้านขวามือของฟังก์ชัน Square Root แล้วคลิ๊กขวา เลือก Create แล้วเลือก Indicator ก็จะได้ Indicator Terminal ที่มีการต่อสายสัญญาณกับจุดต่อสัญญาณของฟังก์ชัน Square Root
5. หลังจากทำขั้นตอนในข้อ 4 และข้อ 5 เสร็จแล้วจะได้โปรแกรมดังรูปที่ 3.53 ซึ่ง สังเกตว่าที่หน้าต่างฟร้อนท์พาเนลจะปรากฏอุปกรณ์ควบคุมและแสดงผลขึ้นเช่นกัน



รูปที่ 3.53 การสร้าง Control และ Indicator บนหน้าต่างบล็อกไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

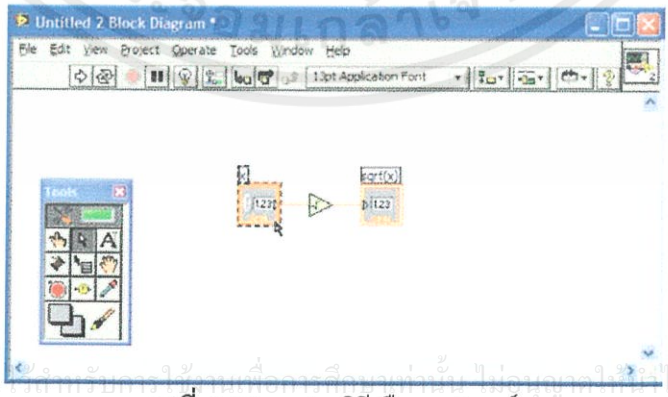
6. ทดสอบการทำงานของโปรแกรม กดปุ่มรันแบบต่อเนื่อง (Run Continuously) ที่หน้าต่างฟร้อนท์พาเนล แล้วทดลองป้อนตัวเลขที่ตัวควบคุมซึ่งผลลัพธ์ที่แสดงที่ตัวแสดงผลจะเป็นรากของตัวเลขที่ป้อนเข้าไป



รูปที่ 3.54 ขั้นตอนการสร้าง Control และ Indicator บนหน้าต่างบล็อกไดอะแกรม

- **การเลือกอุปกรณ์**

เครื่องมือที่ใช้สำหรับการเลือกอุปกรณ์ คือ Positioning Tool ซึ่งอยู่ใน Tool Palette มีลักษณะดังรูปด้านซ้ายมือวัตถุประสงค์ในการเลือกอุปกรณ์ คือ เพื่อย้ายตำแหน่ง ลด/ขยายขนาด และลบ เมื่อเลือก Positioning Tool แล้ว เมาส์ก็จะเปลี่ยนเป็นรูปลูกศร สำหรับการเลือกอุปกรณ์ทำได้โดยนำเมาส์ไปวางที่อุปกรณ์ที่ต้องการจะเลือกแล้วคลิ๊กเมาส์ทางด้านซ้ายจะเกิดเส้นประล้อมรอบอุปกรณ์นั้นแสดงว่าอุปกรณ์นั้นถูกเลือกแล้ว แต่ถ้าหากต้องการเลือกอุปกรณ์มากกว่าหนึ่งตัวแต่ไม่ต้องการเลือกทั้งหมดให้กดปุ่ม Shift ที่คีย์บอร์ด แล้วคลิ๊กเมาส์ที่อุปกรณ์ที่ต้องการเลือก หรือถ้าต้องการเลือกอุปกรณ์ทั้งหมดทุกตัวให้คลิ๊กเมาส์บริเวณที่ว่าง แล้วลากให้กรอบเส้นประสี่เหลี่ยมครอบอุปกรณ์ทุกตัวแล้วปล่อยเมาส์ อุปกรณ์ก็จะถูกเลือกทั้งหมด

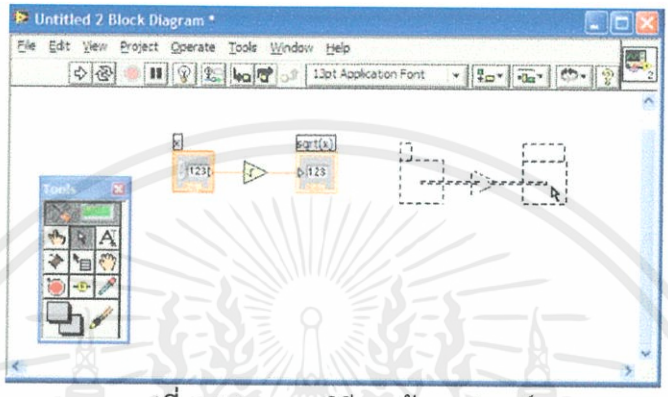


รูปที่ 3.55 แสดงวิธีเลือกอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การเคลื่อนย้ายอุปกรณ์**

การเคลื่อนย้ายอุปกรณ์ไปยังตำแหน่งต่างๆ ตามความต้องการ ทำได้โดยเลือกอุปกรณ์ที่ต้องการเคลื่อนย้ายซึ่งวิธีการเลือกได้กล่าวในหัวข้อที่ผ่านมา เมื่อเลือกอุปกรณ์ที่ต้องการแล้ว การเคลื่อนย้ายอุปกรณ์ทำได้โดยกดปุ่มลูกศรเลื่อนขึ้น/ลง และเลื่อนซ้าย/ขวา ที่คีย์บอร์ด โดยสามารถทำได้อีกหนึ่งวิธี คือ ใช้เมาส์ชี้ที่อุปกรณ์ที่เลือกไว้คลิ๊กค้างแล้วลากไปวางยังตำแหน่งที่ต้องการ



รูปที่ 3.56 แสดงวิธีการย้ายอุปกรณ์

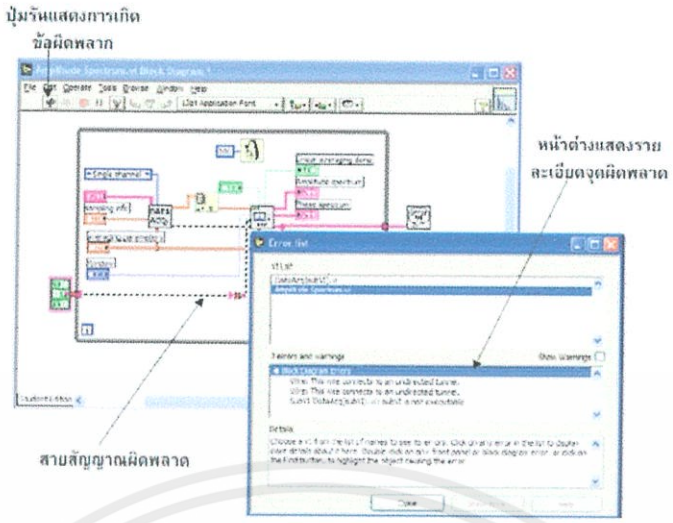
- **การลบและการคัดลอกอุปกรณ์**

การลบอุปกรณ์ ทำได้หลายวิธีด้วยกัน เริ่มต้นจากการเลือกอุปกรณ์ที่ต้องการลบที่คีย์บอร์ดสามารถเลือกได้ 2 ปุ่มคือ Backspace และ Delete อุปกรณ์ก็จะถูกลบ และที่หน้าต่างฟร้อนท์พาเนล หรือบล็อกไดอะแกรมเลือกที่เมนู Edit แล้วเลือก Cut หรือ Clear ส่วนวิธีการคัดลอกอุปกรณ์นั้นทำได้โดยเลือกอุปกรณ์แล้วเลือกเมนู Edit เลือก Copy หรือจะใช้ Ctri+C ที่คีย์บอร์ดก็ได้ การวางอุปกรณ์ทำได้โดยเลือกเมนู Edit เลือก Past หรือจะใช้ Ctri+V ที่คีย์บอร์ดก็ได้เช่นกัน

- **เทคนิคการดีบัคโปรแกรม**

ในหัวข้อนี้จะกล่าวถึงพื้นฐานในการตรวจสอบโปรแกรมของ LabVIEW ซึ่งจะมีเครื่องมือที่ช่วยในการตรวจสอบหาข้อผิดพลาดของโปรแกรม เช่น Highlight Execution, Single Stepping, breakpoints เป็นต้นซึ่งจะได้กล่าวถึงในหัวข้อนี้

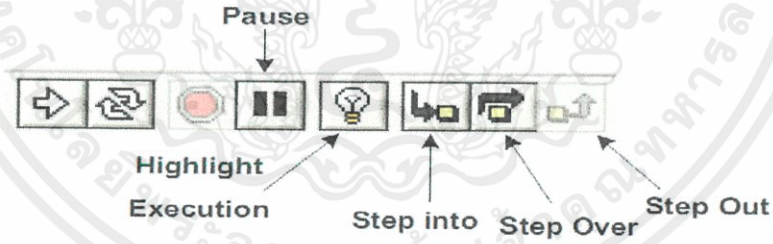
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.57 แสดงการดีบัคโปรแกรม

- **การตรวจหาข้อผิดพลาด**

เมื่อไม่สามารถรันโปรแกรมได้ สังเกตจากปุ่มรันซึ่งจะมีลักษณะแตกออกจากกัน นั่นแสดงว่าเกิดความผิดพลาดในการเขียนโปรแกรมขึ้น ความผิดพลาดส่วนใหญ่แล้ว จะเกิดจากการลากสายสัญญาณผิด ซึ่งจะต้องทำการแก้ไขโปรแกรมจนถูกต้อง จึงจะสามารถทำการรันโปรแกรมได้ การค้นหาจุดผิดพลาดทำได้โดยคลิ๊กเมาส์ที่ปุ่มรันจะปรากฏไดอะล็อกบล็อกขึ้นมา สำหรับรายละเอียดจุดผิดพลาดทั้งหมดดังแสดงในรูปที่ 3.57



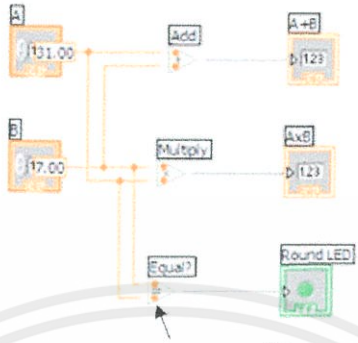
รูปที่ 3.58 เครื่องมือที่ใช้ตรวจสอบโปรแกรม

- Highlight Execution

สามารถที่จะดูทิศทางการไหลของสัญญาณ ลำดับขั้นการทำงานของโปรแกรมได้โดย กดปุ่ม Highlight Execution ที่ Tools bar ในหน้าต่างบล็อกไดอะแกรมแสดงดังรูปที่ 3.58 วัตถุประสงค์ในการตรวจสอบโปรแกรมแบบนี้ เพื่อดูลำดับขั้นการทำงาน และดูทิศทางการไหลของสัญญาณในหน้าต่างบล็อกไดอะแกรมแสดงดังรูปที่ 3.59 เมื่อทำการคลิ๊กที่ปุ่ม Highlight Execution แล้วกดปุ่มรันโปรแกรม จะเห็นจุดกลมๆเล็ก วิ่งจากจุดเริ่มต้นของโปรแกรมผ่านฟังก์ชันการทำงานต่างๆ ไปจนถึงเอาท์พุท และสิ้นสุดการทำงานจุดเล็กนี้จะแทนการไหลของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้มีการนำเนื้อหาในเอกสารนี้ไปเผยแพร่ต่อ

Highlight Execution ทำให้สามารถเห็นถึงขั้นตอนการทำงานของโปรแกรมและทำให้มีความเข้าใจมากยิ่งขึ้น ซึ่งมีความสำคัญเป็นประโยชน์ในการสร้างโปรแกรมเป็นอย่างมาก



รูปที่ 3.59 แสดงทิศทางการไหลของสัญญาณ

- **การตรวจสอบโปรแกรมแบบ Single Stepping**

สำหรับการตรวจหาข้อผิดพลาดของโปรแกรม แบบโหนดต่อโหนด เรียกวิธีการนี้ว่า Single Stepping สัญลักษณ์ของ Single Stepping แสดงดังรูปที่ 3.60 ซึ่งการตรวจสอบการทำงานของโปรแกรมจะใช้เมาส์คลิ๊กที่ปุ่ม Single Stepping ซึ่งประกอบไปด้วยปุ่ม Step into, Step over และ Step out แทนการกดปุ่มรัน โดยการสั่งให้โปรแกรมทำงานนั้นจะใช้ 2 ปุ่มคือ Step into และ Step over โดยการกดเมาส์ที่ปุ่มดังกล่าวโปรแกรมก็จะทำงานทีละโหนด จนกว่าจะจบการทำงาน ถ้าต้องการออกจากการรันโปแกรมแบบ Single Step ให้กดที่ปุ่ม Step out หรือที่ปุ่มหยุดการทำงานของโปรแกรม

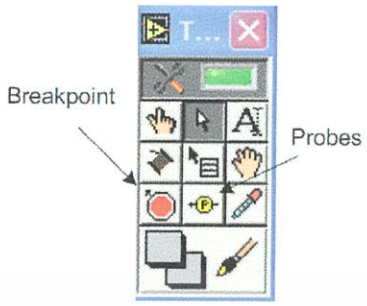
ถ้าต้องการดูทิศทางการไหลของข้อมูลแต่ละโหนด ก็สามารถทำได้โดยการเลือก Highlight Execution แล้วกดปุ่มรันแบบ Single Stepping ซึ่งการทำแบบนี้จะทำให้โปรแกรมรันทีละโหนดและเห็นทิศทางการไหลของข้อมูลแต่ละโหนดด้วย

- **Breakpoint and Probes**

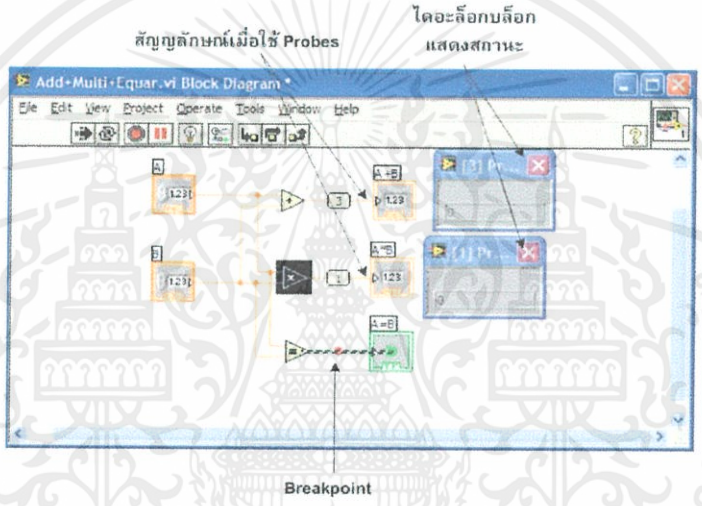
Breakpoint เป็นตัวที่ใช้ในการสั่งให้โปรแกรมหยุดทำงาน ณ ตำแหน่งที่ต้องการ ซึ่งลักษณะของ Breakpoint แสดงดังรูปที่ 3.60 การใช้งานทำได้โดยใช้เมาส์คลิ๊กที่ตัว Breakpoint เมาส์ก็จะเปลี่ยนเป็นรูปสัญลักษณ์ของ Breakpoint จากนั้นนำเมาส์ไปคลิ๊กในบริเวณสายสัญญาณที่ต้องการให้โปรแกรมหยุดทำงาน

Probes แสดงดังรูปที่ 3.60 เมื่อต้องการทราบค่าหรือสถานะของโปรแกรม ณ จุดใดของโปรแกรมก็ใช้เครื่องมือ Probes ไปคลิ๊กบริเวณนั้นซึ่งจะปรากฏไดอะล็อกบล็อกขึ้นมาแสดงค่าข้อมูลหรือสถานะของโปรแกรม ณ จุดนั้นให้ทราบการใช้งาน Breakpoint และ Probes สามารถแสดงได้ดังรูปที่ 3.61

เอกสารนี้เป็นเอกสารที่สงวน ไม่ว่ากรณีใดๆ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ทางการค้า



รูปที่ 3.60 แสดง Tools Palette

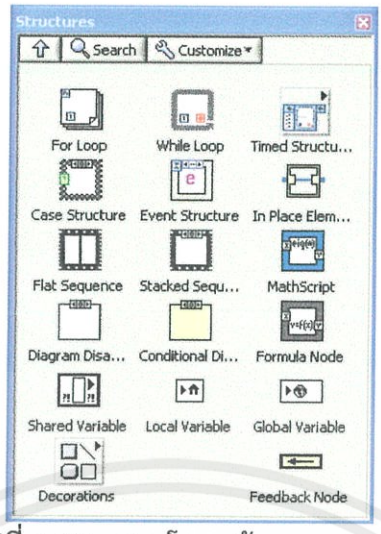


รูปที่ 3.61 แสดงการใช้ Breakpoint และ Probes

3.3.5 รูปแบบโครงสร้างการควบคุมการทำงานของโปรแกรม

ในหัวข้อนี้ จะกล่าวถึงโครงสร้างการควบคุมการทำงานของโปรแกรม VIs ซึ่งโครงสร้างในโปรแกรม LabVIEW ที่จะกล่าวถึงคือ For Loop, While Loop, Shift Register and Feedback Node, Case Structure, Flate and Strack Sequency, Formula Node, MathScrip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.62 แสดงโครงสร้างการควบคุม

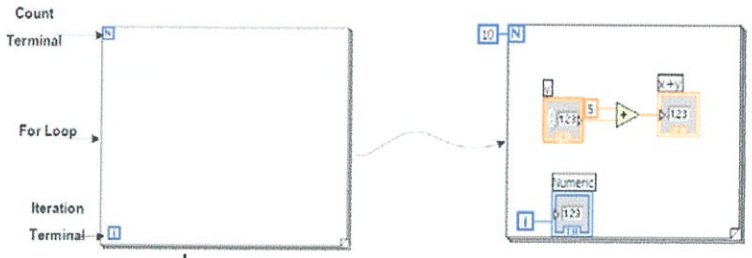
- **For Loop**

For Loop เป็นฟังก์ชันคำสั่งให้โปรแกรมทำงานซ้ำ จนกว่าการกระทำซ้ำนั้นจะเท่ากับจำนวน N เมื่อ N คือจำนวนครั้งที่ต้องการให้ทำซ้ำใน For Loop นั้นจะมีเทอร์มินอลอยู่ 2 จุด คือ count terminal และ iteration terminal สำหรับ count terminal นั้นมีหน้าที่เป็นตัวกำหนดว่าจะให้โปรแกรมทำงานซ้ำกี่รอบ ซึ่งสามารถจะกำหนดให้ปรับเปลี่ยนจำนวนรอบหรือเป็นค่าคงที่ก็ได้ ส่วน iteration terminal จะทำหน้าที่แสดงจำนวนรอบการทำงาน ว่าโปรแกรมทำงานในวงรอบไปกี่ครั้งแล้วตัวเลขที่แสดงของ iteration terminal จะมีค่าเท่ากับ N-1

การสร้าง For Loop ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรม แล้วเลือก Function » Structure แล้วใช้เมาส์เลือกที่ For Loop นำเมาส์ไปคลิ๊กที่หน้าต่างบล็อกไดอะแกรม แล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์จะได้โครงสร้างการทำงานของ For Loop นอกจากนี้แล้วยังสามารถทำการย่อ ขยาย ขนาดของ For Loop โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ For Loop จะปรากฏจุดสี่เหลี่ยมเล็กๆ บริเวณเส้นขอบของ For Loop ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิ๊กเมาส์ค้างแล้วลากย่อ หรือขยายได้ตามต้องการ

การทำงานของ For Loop นอกจากจะทำงานเพียงวงรอบเดี่ยวๆ แล้วยังสามารถทำงานซ้อนวงรอบได้ โดยการทำงานนั้น จะเริ่มทำงานวงรอบด้านในก่อนจนครบเงื่อนไขแล้วจึงมาทำวงรอบนอกหนึ่งรอบแล้วก็กลับไปทำงานในวงรอบในอีกครั้ง การทำงานจะทำจนกว่าวงรอบนอกครบตามเงื่อนไข โปรแกรมจึงจะหยุดทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.63 แสดงโครงสร้างของ For Loop

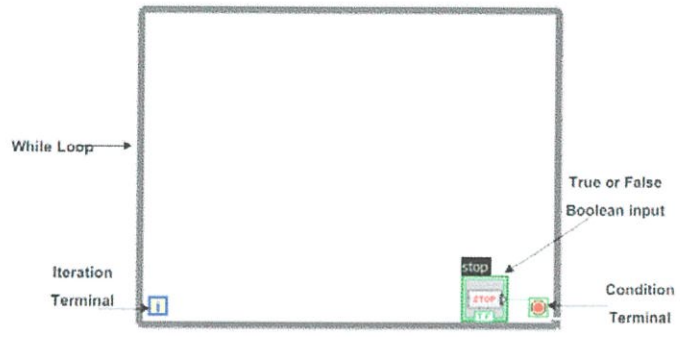
- **While Loop**

While Loop เมื่อเปรียบเทียบกับการเขียนโปรแกรมด้วยภาษาอื่นๆก็จะเป็นโครงสร้างของคำสั่ง Repeats Until นั่นเองซึ่งลักษณ์ของ While Loop แสดงดังรูปที่ 3.64 หลักการทำงานของ While Loop คือ จะทำงานภายในวงรอบไปจนกว่าเงื่อนไขที่ Condition Terminal จะเป็นจริงจึงจะหยุดทำงาน

สำหรับ While Loop นั้นจะมีเทอร์มินอลอยู่ 2 เทอร์มินอลเช่นเดียวกันกับ For Loop คือ Iteration Terminal และ Condition Terminal การทำงานของ Iteration Terminal จะทำงาน เหมือนกันกับ Iteration Terminal ของ For Loop คือ ทำหน้าที่แสดงจำนวนรอบที่โปรแกรมกระทำในวงรอบซึ่งแสดงผลออกมาในลักษณะของตัวเลขส่วน Condition Terminal เป็นการกำหนดเงื่อนไขให้โปรแกรมหยุดทำงาน ซึ่งเงื่อนไขในการหยุดทำงาน จะหยุดทำงานเมื่อเงื่อนไขเป็นจริง(True) หรือจะให้หยุดทำงานเมื่อเงื่อนไขเป็นเท็จ (False) ก็สามารถที่จะกำหนดได้ นอกจากนี้แล้วยังสามารถกำหนดเงื่อนไขในการหยุดทำงานได้อีกหนึ่งวิธีคือการนำเอาค่าเอาท์พุทของ Iteration Terminal มาเปรียบเทียบกับค่าที่ ผู้ใช้ต้องการไม่ว่าจะเป็นค่าคงที่หรือปรับเปลี่ยนได้ โปรแกรมจะทำงานวนในวงรอบและนำค่าเอาท์พุทของ Iteration Terminal มาเปรียบเทียบกับค่าที่ได้ทำการกำหนดไว้แล้วจะส่งเอาท์พุทที่ได้ไปยัง Condition Terminal เมื่อจำนวนรอบการทำงานมีค่าเท่ากับค่าที่กำหนด โปรแกรมจะหยุดทำงานในวงรอบนี้ทันที การกำหนดเงื่อนไขแบบนี้เปรียบได้กับการกำหนดค่า N ใน For Loop นั่นเอง

ในการสร้าง While Loop ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรม แล้วเลือก Function >> Structure แล้วใช้เมาส์เลือกที่ While Loop นำเมาส์ไปคลิ๊กที่หน้าต่างบล็อกไดอะแกรม แล้วลากให้มีขนาดที่ต้องการ แล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ While Loop นอกจากนี้ แล้วยังสามารถทำการย่อขยายขนาดของ While Loop ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ While Loop จะปรากฏจุดสี่เหลี่ยมเล็กๆ บริเวณเส้นขอบของ While Loop ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิ๊กเมาส์ค้างแล้วลากย่อหรือขยายได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

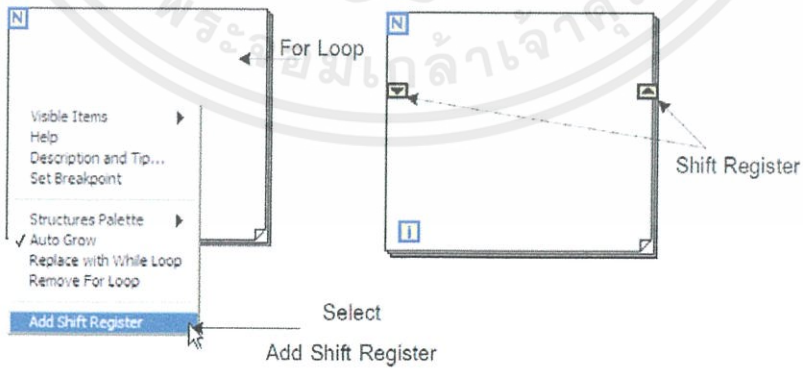


รูปที่ 3.64 แสดงโครงสร้างของ While Loop

- **Shift Register and Feedback Node**
  - **Shift Register**

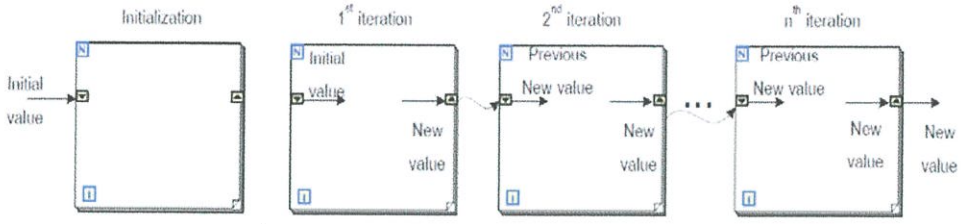
Shift Register ประกอบไปด้วยเทอร์มินอลสองตัวคือเทอร์มินอลทางด้านอินพุท และ เทอร์มินอลทางด้านเอาท์พุท Shift Register สามารถใช้ทำงานร่วมกับ For Loop และ While Loop เท่านั้นการสร้าง Shift Register ทำได้โดยการใช้เมาส์คลิ๊กขวาบริเวณเส้นขอบของ For Loop หรือ While Loop แล้วเลือก Add Shift Register ก็จะได้ Shift Register ที่อยู่บนเส้นขอบทางซ้าย และขวา ของ For Loop และ While Loop ดังแสดงให้เห็นในรูปที่ 3.66

การทำงานของ Shift Register คือ จะส่งค่าข้อมูลจากเทอร์มินอลทางซ้าย ไปยังเทอร์มินอลทางขวา ค่าเอาท์พุทของวงรอบที่ผ่านมาจะมาเป็นค่าเริ่มต้นของวงรอบ ค่าเริ่มต้นของ Shift Register สามารถกำหนดได้แต่ถ้าไม่ได้กำหนดค่าเริ่มต้น ก็จะมีค่ามาตรฐานที่มาจากโปรแกรม Shift Register สามารถใช้ได้กับข้อมูลทุกชนิด ไม่ว่าจะเป็นตัวเลข ตัวหนังสือลอจิกทั้งที่เป็นค่าเดี่ยว หรืออาร์เรย์



รูปที่ 3.65 แสดงโครงสร้างของ Shift Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

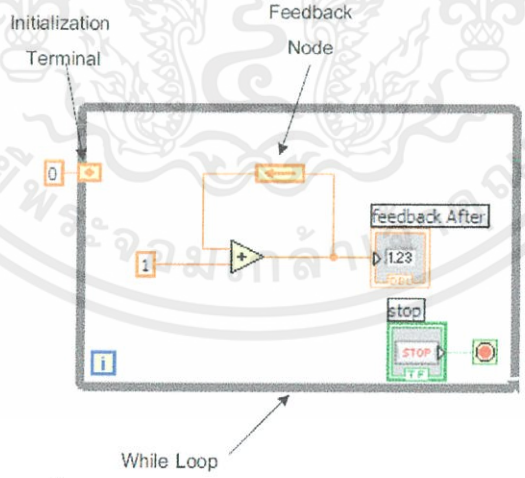


รูปที่ 3.66 แสดงการทำงานของ Shift Register

- **Feedback Node**

Feedback Node สามารถทำงานร่วมกับFor Loop และ While Loop ได้เช่นเดียวกันกับ Shift Register การทำงานของ Feedback Node คือจะเก็บข้อมูลไว้ที่ Iteration เมื่อโปรแกรมทำงานภายในวงรอบเสร็จแล้วจะส่งข้อมูลดังกล่าวไปที่ Iteration ของวงรอบต่อไป ทั้งนี้ Feedback Node ยังสามารถกำหนดค่าเริ่มต้นได้ด้วยและสามารถใช้ได้กับข้อมูลทุกชนิด ลักษณะการต่อใช้งาน Feedback Node แสดงดังรูปที่ 3.67

การสร้าง Feedback Node ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรม แล้วเลือก Function» Structure แล้วใช้เมาส์เลือกที่ Feedback Node (แสดงได้ดังรูปที่ 3.67) นำเมาส์ไปคลิ๊กที่หน้าต่างบล็อกไดอะแกรม (ที่มี For Loop หรือ While Loop อยู่แล้ว) แล้วลากเมาส์ไปวางภายในกรอบของ For Loop หรือ While Loop เท่านั้นเพราะถ้าวางเมาส์ภายนอกกรอบของ For Loop หรือ While Loop จะไม่สามารถวาง Feedback Node



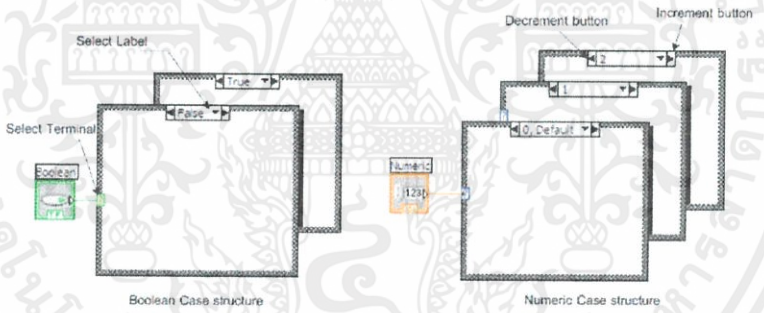
รูปที่ 3.67 แสดงการใช้งาน Feedback Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Case Structure**

Case Structure เป็นโครงสร้างการเขียนโปรแกรม ตามเงื่อนไขของตัวหนังสือ (Conditional text) ถ้าเปรียบเทียบกับการเขียนโปรแกรมภาษาอื่นๆ (text-based programming

languages) สามารถเปรียบเทียบได้กับโครงสร้างของคำสั่ง *If...Then...Else* การเรียกใช้งานฟังก์ชัน Case Structure ลงบนหน้าต่างบล็อกไดอะแกรมทำได้โดยเลือก Function» Structure แล้วใช้เมาส์เลือกที่ Case Structure (แสดงดังรูปที่ 3.68) นำเมาส์ไปคลิ๊กที่หน้าต่างบล็อกไดอะแกรมแล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ Case Structure นอกจากนี้แล้วยังสามารถทำการย่อขยายขนาดของ Case Structure ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ Case Structure จะปรากฏจุดสี่เหลี่ยมเล็กๆ บริเวณเส้นขอบของ While Loop ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิ๊กเมาส์ค้างแล้วลากย่อ หรือขยายได้ตามต้องการ สามารถแสดงลักษณะของ Case Structure ได้ดังรูปที่ 3.68 ซึ่งมีส่วนประกอบที่สำคัญอยู่ 2 ส่วนคือ Select Terminal และ Select Label โดยที่ Select Terminal จะทำหน้าที่รับอินพุทเข้ามาแล้วทำการตรวจสอบหรือเปรียบเทียบว่าข้อมูลที่รับเข้ามาเป็นข้อมูลชนิดใด และตรงกับเงื่อนไขใดใน Case Structure เพื่อทำให้โปรแกรมแกรมทำงานตามเงื่อนไขที่ต้องการ ส่วน Select Label จะทำหน้าที่สร้างหน้าต่างย่อยของ Case Structure ซึ่งสามารถเพิ่มและลดเงื่อนไขได้ถึง 2^{31}-1 เงื่อนไขการส่งข้อมูลออกเอาท์พุทนั้นทุกๆกรณี จะส่งข้อมูลออกที่เอาท์พุทเดียวกัน ซึ่งทางออกของสัญญาณเอาท์พุทเรียกว่า Output Tunnel



รูปที่ 3.68 แสดงโครงสร้างของ Case Structure

- **Flat and Stack Sequence Structures**

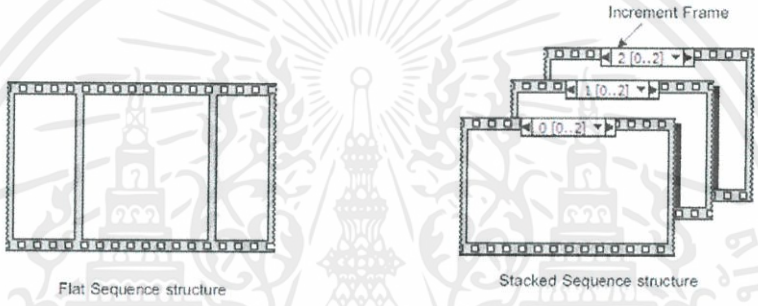
Sequence Structures เป็นการทำงานแบบลำดับขั้นรูปแบบของ Sequence Structures มีลักษณะเหมือนแผ่นฟิล์ม สำหรับ Sequence Structures มีทั้งหมด 2 แบบคือ Flat Sequence และ Stack Sequence ลักษณะของ Flat Sequence และ Stack Sequence แสดงได้ดังรูปที่ 3.69 ข้อแตกต่างของทั้ง 2 แบบ คือ Flat Sequence มีมีลักษณะของเฟรมต่อกันไปไม่ว่าจะเพิ่มหรือลดจำนวนเฟรมเราก็จะทั้งหมดว่ามีกี่เฟรมส่วน Stack Sequence จะมีลักษณะคล้ายกับ Case Structure กล่าวคือจะมี Select Label สำหรับเลือกเฟรมที่ซ่อนอยู่ภายในโดยจะตัวเลขบอกว่ากำลังเปิดเฟรมใดอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

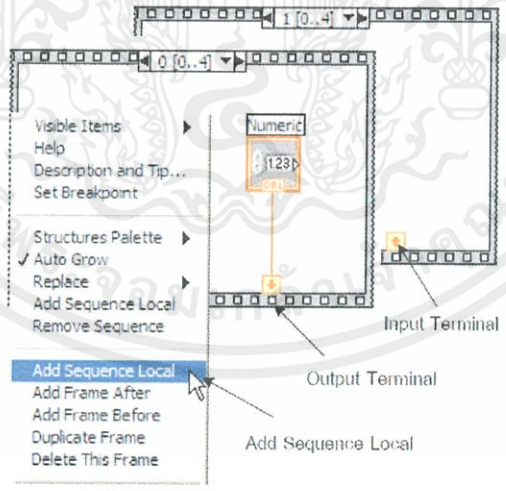
การสร้าง Flat และ Stack Sequence Structures ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function» Structure แล้วใช้เมาส์เลือกที่ Flat หรือ Stack

Sequence Structures ตามที่ต้องการ (แสดงดังรูปที่3.55) นำเมาส์ไปคลิ๊กหน้าต่างบล็อกไดอะแกรม แล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ ก็จะได้โครงสร้างการทำงานของ Flat และ Stack Sequence Structures

การที่จะนำข้อมูลของเฟรมหนึ่ง ไปยังอีกเฟรมหนึ่ง สามารถทำได้โดยคลิ๊กเมาส์ทางขวาที่บริเวณกรอบ Sequence Structures แล้วเลือก Add Sequence Local จะปรากฏเทอร์มินอลขึ้นที่ขอบของ Sequence Structures จะนั้นลากสายสัญญาณที่ต้องการเอาออกเอาท์พุทไปต่อกับเทอร์มินอล ซึ่งเทอร์มินอลดังกล่าว ก็จะเปลี่ยนเป็นลักษณะที่มีลูกศรพุ่งออกจากเฟรม ในขณะเดียวกันทุกเฟรมที่เหลือก็จะปรากฏเทอร์มินอลขึ้นเช่นกัน แต่จะมีลักษณะลูกศรชี้เข้าไปข้างในเฟรมแสดงดัง รูปที่ 3.70



รูปที่ 3.69 แสดงโครงสร้างของ Flat และ Stack Seqence Structures



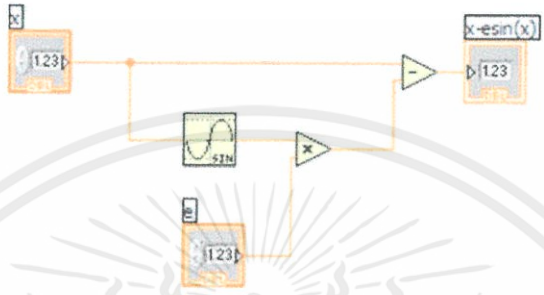
รูปที่ 3.70 การ Add Sequence Local ของ Stack Sequence Structures

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

- **Formula Node**

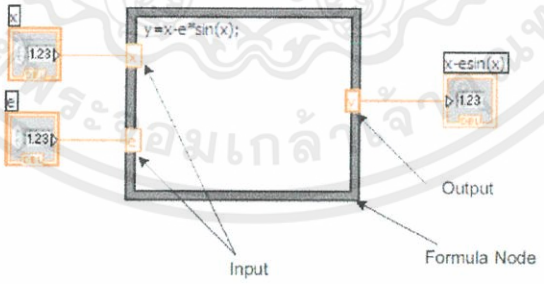
Formula Node เป็นโครงสร้างของคำสั่ง ที่มีความใกล้เคียงกับการเขียนโปรแกรมด้วยภาษาอื่นๆ ทั่วไปซึ่ง Formula Node จะถูกนำมาใช้เมื่อต้องการให้โปรแกรมทำงาน

ตามฟังก์ชันทางคณิตศาสตร์ที่มีความยุ่งยาก ของการใช้อุปกรณ์ ในการสร้างให้โปรแกรมทำงานตามฟังก์ชันทางคณิตศาสตร์ หนึ่งฟังก์ชันอาจต้องใช้อุปกรณ์หลายตัว ซึ่งอาจทำให้การตรวจสอบโปรแกรมเป็นไปด้วยความยุ่งยาก และโปรแกรมมีขนาดใหญ่เกินไป ยกตัวอย่างการสร้างโปรแกรมให้ทำงานตามสมการ $y = x - e\sin(x)$ ซึ่งถ้าเขียนโปรแกรมด้วยการใช้อุปกรณ์ทั่วไป จะได้ลักษณะของโปรแกรมดังรูปที่ 3.71 แต่หากใช้ Formula Node จะได้ลักษณะของโปรแกรมดังรูปที่ 3.72



รูปที่ 3.71 การสร้างโปรแกรมจากอุปกรณ์ทั่วไป

การทำงานของ Formula Node จะเป็นลักษณะของการกำหนดฟังก์ชันทางคณิตศาสตร์ลงไปใน Formula Node (แสดงดังรูปที่ 3.72) แล้วจึงกำหนดเทอร์มินอลเพื่อเชื่อมต่ออินพุท และเอาท์พุท การสร้าง Formula Node ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรม แล้วเลือก Function» Structure แล้วใช้เมาส์เลือกที่ Formula Node (แสดงดังรูปที่ 3.72) นำเมาส์ไปคลิ๊กที่หน้าต่างบล็อกไดอะแกรม แล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างของการทำงาน Formula Node

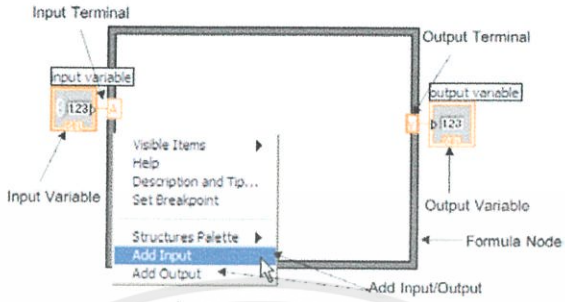


รูปที่ 3.72 โครงสร้างและการใช้งาน Formula Node

การกำหนดเทอร์มินอลอินพุท และเอาท์พุทสามารถทำได้โดยการคลิ๊กเมาส์ขวาที่ขอบของ Formula Node แล้วเลือก Add Input หรือ Add Output (แสดงดังรูปที่ 3.73) ที่เทอร์มินอลสามารถที่จะกำหนดชื่อได้โดยค่าอินพุทและเอาท์พุท จะเป็นข้อมูลแบบตัวเลข (Numeric) การกำหนดตัวแปรอินพุท สามารถกำหนดให้เป็นค่าคงที่ หรือค่าที่ปรับเปลี่ยนได้ส่วนเอาท์พุทจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

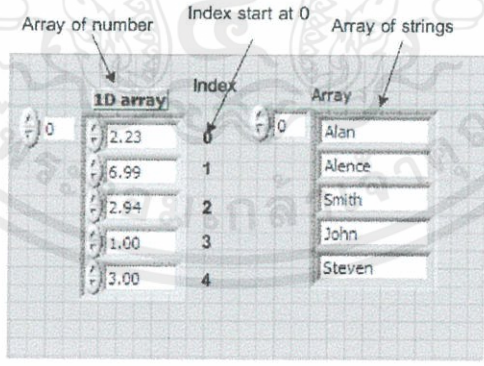
กำหนดให้เป็นตัวแสดงผล ชื่อที่เทอร์มินอลก็คือชื่อของตัวแปรของฟังก์ชันทางคณิตศาสตร์ ที่สร้างขึ้นใน Formula Node



รูปที่ 3.73 แสดงการเพิ่มอินพุทและเอาท์พุทของ Formula Node

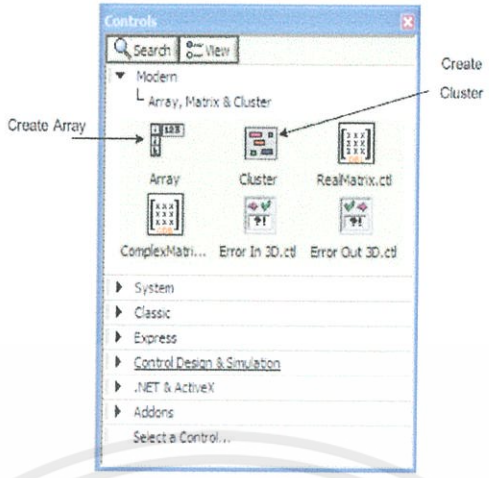
3.3.6 พื้นฐานการใช้งานอาร์เรย์

อาร์เรย์ เป็นการเก็บตัวแปร หรือข้อมูลชนิดเดียวกันหลายๆข้อมูลไว้ด้วยกัน สำหรับคลัสเตอร์จะเก็บข้อมูลที่ชนิดต่างกันไว้ด้วยกันได้ ปัญหาที่พบบ่อยคือ เวลาโปรแกรมทำงานจะมีข้อมูลเป็นจำนวนมาก ฉะนั้นอาร์เรย์หรือคลัสเตอร์จะช่วยในการเก็บข้อมูลอาร์เรย์สามารถสร้างได้หลายมิติ เริ่มตั้งแต่ 1 มิติ ซึ่ง ในอาร์เรย์ 1 มิติสามารถเก็บข้อมูลได้สูงถึง 2^{31}-1 ข้อมูลค่าสูงสุดที่จะเก็บได้ก็ขึ้นอยู่กับขนาดของหน่วยความจำอาร์เรย์ในโปรแกรม LabVIEW สามารถเก็บข้อมูลได้หลายชนิด ยกเว้น Chart และ Graph วิธีการอ้างถึงข้อมูลในอาร์เรย์เรียกว่าดัชนี (Index) ซึ่งมีค่าเริ่มต้นตั้งแต่ 0 ไปจนถึง N-1 เมื่อ N คือ จำนวนข้อมูลในอาร์เรย์ ในรูปที่ 3.74 เป็นอาร์เรย์ 1 มิติ โดยมีค่าดัชนีเริ่มต้นที่ 0



รูปที่ 3.74 แสดง Array One Dimension (1-D)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.75 การเรียกใช้งาน Array

- **การสร้างอาร์เรย์ด้วย Control และ Indicator**

สามารถสร้างอาร์เรย์ด้วย Control และ Indicator ของ Numeric, Boolean, Strings ซึ่งการสร้างอาร์เรย์จะประกอบไปด้วย Array Shell และ Matrix & Cluster อยู่ในพาเลทย่อย ของ Controls» Modern ดูได้จากรูปที่ 3.76 ขั้นตอนในการสร้างอาร์เรย์ด้วย Control และ Indicator มีดังนี้

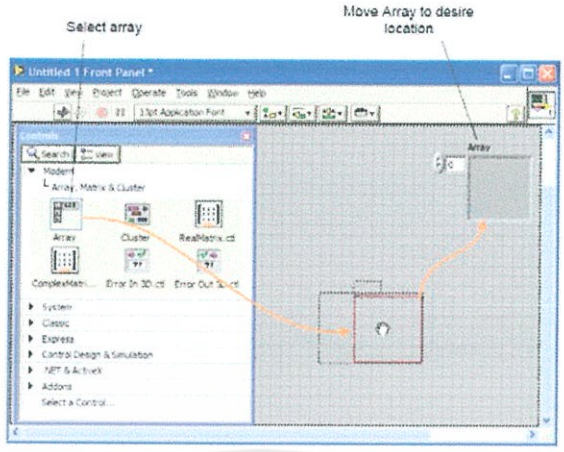
- เลือกบล็อกของอาร์เรย์จาก Array, Matrix & Cluster ในพาเลทย่อยของ Control» Modern พาเลท แล้วลากไปวางในหน้าต่างฟร้อนท์พาเนล ดังแสดงในรูปที่ 3.78

- เลือกข้อมูลที่ต้องการ เช่น Numeric, Boolean และ String แล้วลากไปวางไว้ในบล็อก อาร์เรย์ ดังแสดงในรูปที่ 3.76 โดยที่อาร์เรย์จะปรับขนาดให้เหมาะสมกับชนิดของข้อมูลที่นำมาวาง

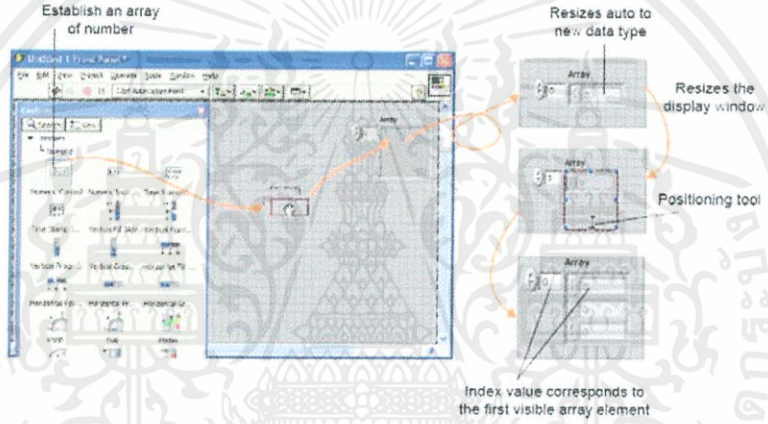
สามารถที่จะย่อหรือขยายอาร์เรย์ เพื่อดูค่าของสมาชิกที่อยู่ภายในอาร์เรย์ได้โดยใช้เมาส์ไปวางบริเวณอาร์เรย์ แล้วจะปรากฏจุดสี่เหลี่ยมเล็กๆ ขึ้นรอบๆ บล็อกของอาร์เรย์ จากนั้นใช้เมาส์ไปวางบริเวณจุดเหล่านั้นจุดใดก็ได้ตามต้องการ แล้วเมาส์จะเปลี่ยนไอคอน ไปเป็นลูกศรแล้วจึงทำการคลิ๊กเมาส์ค้างแล้วลากย่อขยายตามที่ต้องการ

ครั้งแรกที่ลากบล็อกของอาร์เรย์ไปวางในหน้าต่างฟร้อนท์พาเนล จะปรากฏบล็อกของ array shell ซึ่งจะเป็นแสดงชนิดของข้อมูลที่ไม่ทราบค่า แสดงดังรูปที่ 3.77 เมื่อเราลาก string, Boolean หรือ numeric ไปวางในบล็อกดังกล่าว ในหน้าต่างบล็อกไดอะแกรมจะสังเกตเห็นได้ว่าเทอร์มินอลจะเปลี่ยนเป็นสีส้ม แสดงดังรูปที่ 3.78

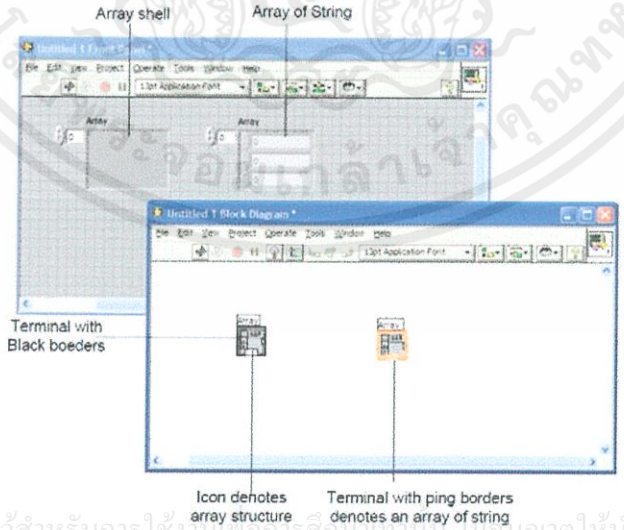
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.76 แสดงอาร์เรย์สร้างจาก Control และ Indicator



รูปที่ 3.77 ขั้นตอนการสร้างอาร์เรย์จาก Control และ Indicator



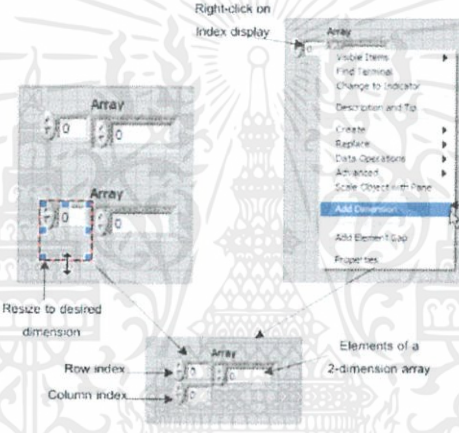
รูปที่ 3.78 เทอร์มินอลของ array shell

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **อาร์เรย์หลายมิติ**

อาร์เรย์ 2 มิติ (2D) จะมีตัวบอกคุณลักษณะอยู่ 2 ตัว คือ บอกแถวและบอกหลัก อาร์เรย์ 3 มิติก็จะมีตัวบอกคุณลักษณะ 3 ตัว ดังนั้นอาร์เรย์ n มิติ ก็จะมีตัวบอกคุณลักษณะ n ตัว การเพิ่มมิติของอาร์เรย์สามารถทำได้ 2 วิธีคือ ใช้ Positioning Tool เพื่อเพิ่มมิติของอาร์เรย์ และคลิ๊กเมาส์ขวาที่อาร์เรย์แล้วเลือกคำสั่ง Add Dimension จากเมนูย่อย ซึ่งวิธีการทั้งสองที่ใช้ในการเพิ่มมิติของอาร์เรย์สามารถแสดงได้ดังรูปที่ 3.79

การลดมิติของอาร์เรย์นั้น สามารถทำได้ 2 วิธี เช่นเดียวกันกับการเพิ่ม คือ ใช้ Positioning Tool เพื่อลดขนาดมิติของอาร์เรย์ และคลิ๊กเมาส์ขวาที่อาร์เรย์แล้วเลือกคำสั่ง Remove Dimension จากเมนูย่อย

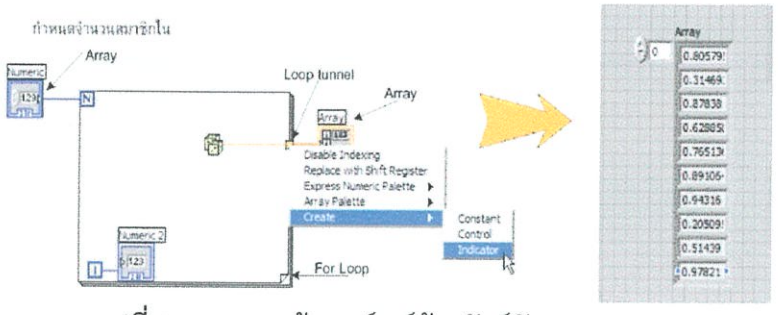


รูปที่ 3.79 แสดงวิธีการเพิ่มมิติของอาร์เรย์

3.3.7 การสร้างอาร์เรย์ด้วยลูป

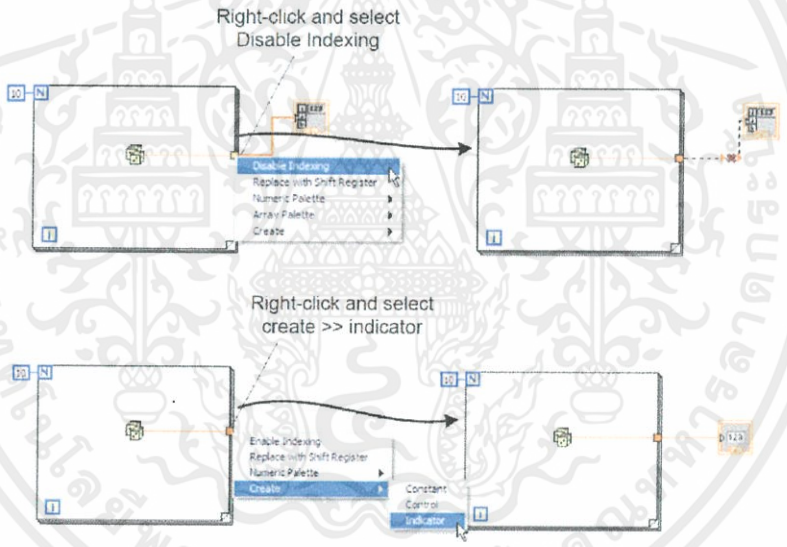
ในการสร้าง Array ด้วยลูปสามารถสร้างได้จาก For Loop และ While Loop ซึ่ง For Loop หรือ While Loop จะส่งข้อมูลของทุกลูปออกมาให้อัตโนมัติเรียกว่า Auto-indexing ใน For Loop สามารถสร้างอาร์เรย์ได้โดยจำนวนสมาชิกของอาร์เรย์ จะเท่ากับจำนวนลูปที่กำหนดไว้ การสร้างอาร์เรย์ 2 มิติโดยใช้ For Loop ทำได้โดยการใช้ For Loop ซ้อนกัน 2 ตัว For Loop วงนอกจะแทนจำนวนสมาชิกของแถว และ For Loop วงในจะแทนสมาชิกของหลัก วิธีการสร้างอาร์เรย์ด้วย For Loop แสดงดัง รูปที่ 3.80 ซึ่งเป็นวิธีการสร้างอาร์เรย์ด้วย For Loop เป็นอาร์เรย์ 1 มิติ ที่มีจำนวนสมาชิกในอาร์เรย์เท่ากับ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.80 การสร้างอาร์เรย์ด้วยฟังก์ชัน For Loop

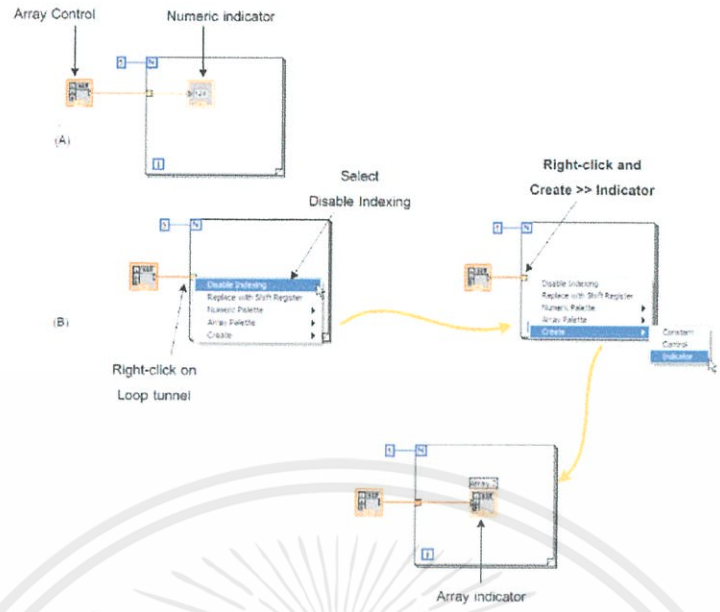
ถ้าหากไม่ต้องการที่จะสร้างอาร์เรย์ สามารถทำได้โดยการคลิ๊กเมาส์ขวาที่ tunnel (สี่เหลี่ยมเล็กๆ อยู่บนเส้นขอบของฟังก์ชัน For Loop เป็นทางผ่านของข้อมูล) แล้วเลือกคำสั่ง Disable» Indexing จากเมนูย่อย จากนั้นคลิ๊กขวาที่ tunnel เดิมแล้วเลือก Create» indicator จะปรากฏตัวแสดงผลที่เป็นแบบตัวเลขธรรมดา



รูปที่ 3.81 การเปลี่ยนแปลงการแสดงผลจากอาร์เรย์เป็นแบบ Numeric

ถ้ามีตัวแสดงผลแบบ Numeric อยู่ภายในฟังก์ชัน For Loop (ดังแสดงในรูปที่ 3.81) แล้วต้องการเปลี่ยนตัวแสดงผลให้เป็นอาร์เรย์สามารถทำได้โดยคลิ๊กเมาส์ขวาที่ tunnel แล้วทำการเลือกคำสั่ง Disable» Indexing จากเมนูย่อย จากนั้นคลิ๊กขวาที่ tunnel เดิมแล้วเลือก Create» indicator ตัวแสดงผลจะแสดงผลในรูปแบบของอาร์เรย์ แสดงดังรูปที่ 3.82

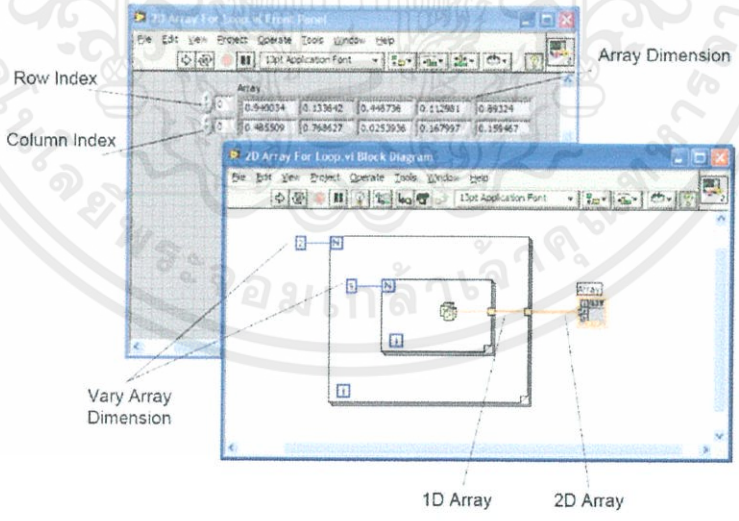
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.82 การต่อสายสัญญาณเมื่ออาร์เรย์อยู่ภายในลูป

- **การสร้างของอาร์เรย์ 2 มิติ**

การสร้างอาร์เรย์ 2 มิติสามารถสร้างจาก for Loopได้โดยใช้ For Loop สองตัว โดยที่ For Loop ลูปนอกเป็นตัวสร้างแถวของอาร์เรย์ และ For Loop ลูปในใช้สำหรับสร้างหลักของอาร์เรย์ อาร์เรย์ 2 มิติ ที่สร้างจาก For Loop แสดงได้ดังรูปที่ 3.83



รูปที่ 3.83 การสร้างอาร์เรย์ 2 มิติ ด้วย For Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

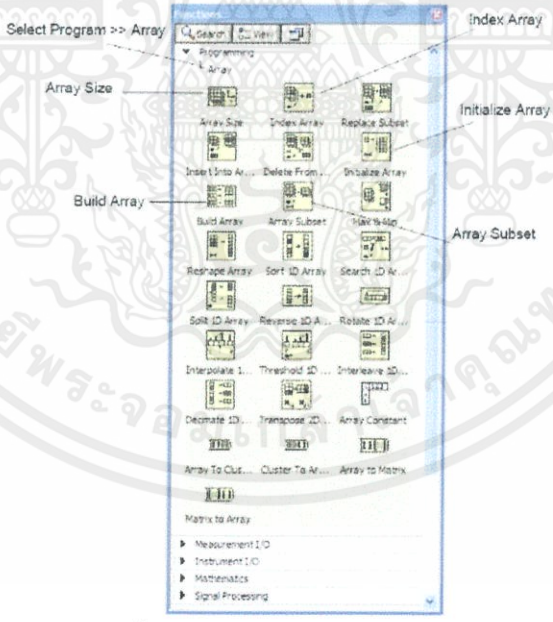
3.3.8 ฟังก์ชันของอาร์เรย์

- Array Size

ฟังก์ชัน Array Size จะเป็นตัวบอกขนาดของอาร์เรย์ว่ามีขนาดใด เช่น ถ้าอาร์เรย์อินพุทเป็นอาร์เรย์ 1 มิติ (4 แถว 1 หลัก) อาร์เรย์เอาท์พุทจะเป็นอาร์เรย์ 1 มิติ มีจำนวนสมาชิกทั้งหมด 4 ตัว หรือถ้าอาร์เรย์อินพุทเป็นอาร์เรย์ 2 มิติ (2 แถว 4 หลัก) อาร์เรย์เอาท์พุทจะแสดงเป็นอาร์เรย์ 2 มิติ (แสดงจำนวนของแถวและหลัก) แสดงดังรูปที่ 3.84

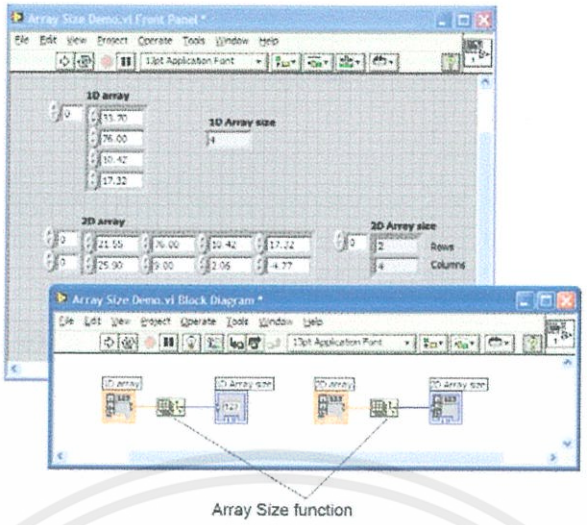
- **Initialize Array**

เป็นการสร้างอาร์เรย์ตามขนาด ที่กำหนดพร้อมทั้งสามารถกำหนด ค่าเริ่มต้นให้กับอาร์เรย์ที่สร้างได้อีกด้วย เช่น ต้องการสร้างอาร์เรย์ 1 มิติ มีจำนวนสมาชิกทั้งหมด 50 ตัว และกำหนดให้ค่าเริ่มต้นของอาร์เรย์เป็น 2.5 สามารถสร้างได้โดยคลิ๊กเมาส์ขวาที่เทอร์มินอล element ซึ่งอยู่ทางด้านบนซ้ายของ บล็อก Initialize Array แล้วเลือก create» constant และกำหนดค่าเริ่มต้นเป็น 2.5 ที่เทอร์มินอล dimension size ของบล็อก Initialize Array ที่อยู่ด้านล่างเทอร์มินอล element คลิ๊กเมาส์ขวาแล้วเลือก create» constant และกำหนดให้เป็น 50 (จำนวนสมาชิก) ถ้าต้องการเพิ่มมิติของอาร์เรย์ ก็สามารถทำได้โดยคลิ๊กเมาส์ขวาที่บล็อกของ Initialize Array แล้วเลือก Add dimension ซึ่งจะเพิ่มเทอร์มินอล เพื่อกำหนดขนาดของแถวและหลักแสดงดังรูปที่ 3.85

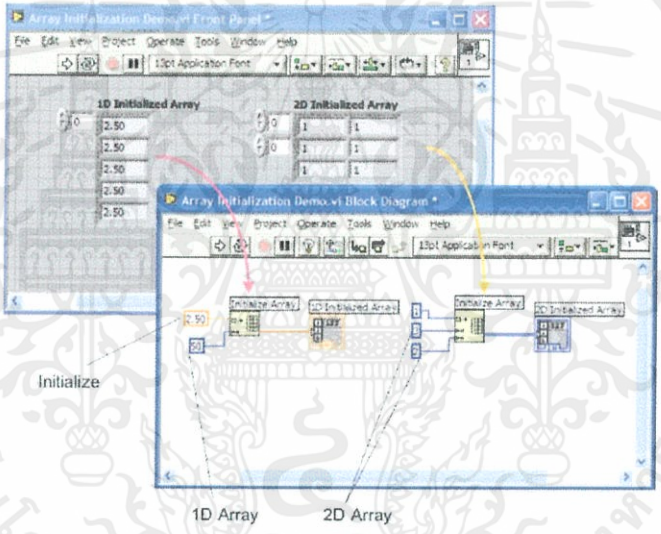


รูปที่ 3.84 แสดงฟังก์ชันอาร์เรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.85 แสดงการใช้งาน Array Size Function



รูปที่ 3.86 แสดงการใช้งาน Initialize Array Function

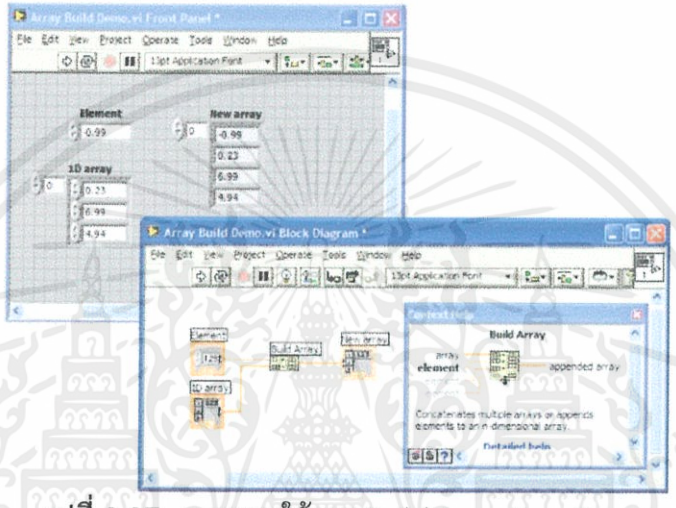
- **Build Array**

เป็นการนำเอาอาร์เรย์ หรือสมาชิกในอาร์เรย์หลายๆ ตัวมารวมกัน แล้วเกิดเป็นอาร์เรย์ใหม่ ซึ่งอินพุทสามารถที่จะเป็นทั้งอาร์เรย์และเป็นจำนวนสเกลาร์ โดยปกติแล้ว Build Array จะมีอินพุท 2 อินพุท ถ้าต้องการเพิ่มจำนวนอินพุทสามารถทำได้โดยการคลิ๊กเมาส์ขวาที่บล็อก Build Array แล้วเลือก Add Input หรือจะใช้ Positioning Tool คลิ๊กค้างแล้วลากขยายเพื่อเพิ่มอินพุท และถ้าต้องการลดจำนวนอินพุทก็สามารถทำได้โดยคลิ๊กเมาส์ขวา Build Array แล้วเลือก Remove Input ตัวอย่างการใช้งาน Build Array แสดงดังรูปที่ 3.87

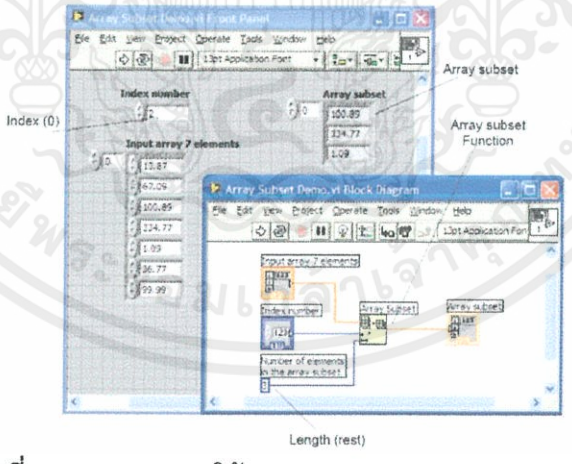
เอกสารนี้มีลิขสิทธิ์ ... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Array Subset**

เป็นวิธีการเลือกค่าของข้อมูลที่อยู่ในอาร์เรย์ไปใช้งาน สิ่งที่ต้องกำหนดอยู่ 2 อย่าง คือ Index (0) ซึ่งเป็นการกำหนดตำแหน่งเริ่มต้นที่เราต้องการ (ตำแหน่งแรกนับเป็นตำแหน่งที่ 0) และ Length (rest) เพื่อกำหนดจำนวนข้อมูลที่ต้องการ ดังตัวอย่างในรูปที่ 3.88 อาร์เรย์ 1 มิติ มีจำนวนข้อมูล 7 ข้อมูล ต้องการใช้ข้อมูลตั้งแต่ตำแหน่งที่ 3 ถึง 5 ทำได้โดยกำหนด Index (0) ให้มีค่าเท่ากับ 2 และ Length (rest) ให้มีค่าเท่ากับ 3 แสดงดังรูปที่ 3.88



รูปที่ 3.87 แสดงการใช้งาน Build Array Function



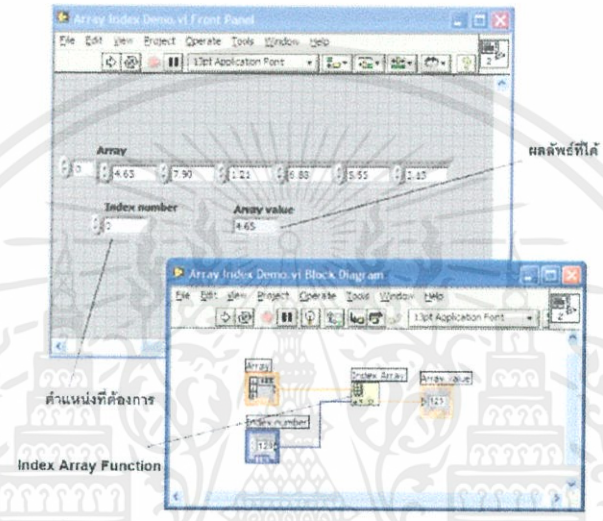
รูปที่ 3.88 แสดงการใช้งาน Subset Array Function

- **Index Array**

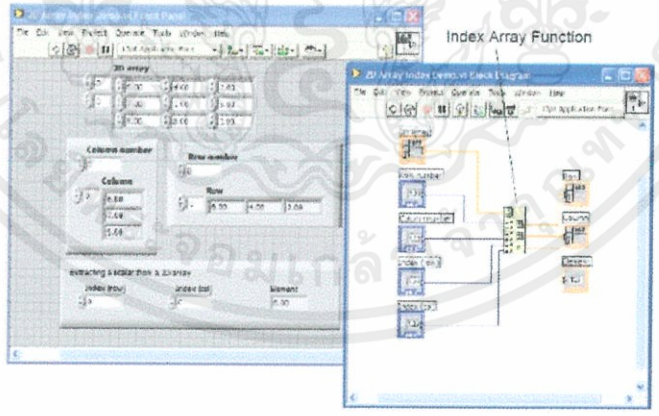
ฟังก์ชัน Index Array สามารถที่จะเลือกข้อมูลในอาร์เรย์ในตำแหน่งที่ต้องการมาใช้งานได้โดยการเลือกตำแหน่งของข้อมูลได้เลย ในกรณีที่เป็นอาร์เรย์ขนาด 2 มิติ ก็สามารถใช้ Index Array เลือกว่า จะเลือกข้อมูลออกเอาท์พุทที่ตำแหน่งแถวและหลัก แสดงดังรูปที่ 3.89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

มีอาร์เรย์ 1 มิติ มีจำนวนข้อมูล ทั้งหมด 6 ข้อมูล ต้องการเลือกข้อมูลที่ 1 ไปใช้งานสามารถทำได้โดยกำหนดให้ Index Number มีค่า เท่ากับ 0 (ตำแหน่งแรกนับเป็นตำแหน่งที่ 0) และในรูปที่ 3.90 เป็นการใช้ฟังก์ชัน Index Array ในการเลือกข้อมูลจากอาร์เรย์ 2 มิติ ขนาด 3 x 3 โดย Index Array จะต้องมี 2 ตัว เพื่อเลือกตำแหน่งแถวและตำแหน่งหลังที่ต้องการ ซึ่งการเพิ่มอินพุทของ Index Array สามารถทำได้โดยการใช้ Position Tool คลิ๊กที่ Index Array แล้วลากขยายเพื่อเพิ่มอินพุทให้กับ Index Array



รูปที่ 3.89 แสดงการใช้งาน Index Array Function



รูปที่ 3.90 แสดงการใช้งาน Index Array Function กับอาร์เรย์ 2 มิติ

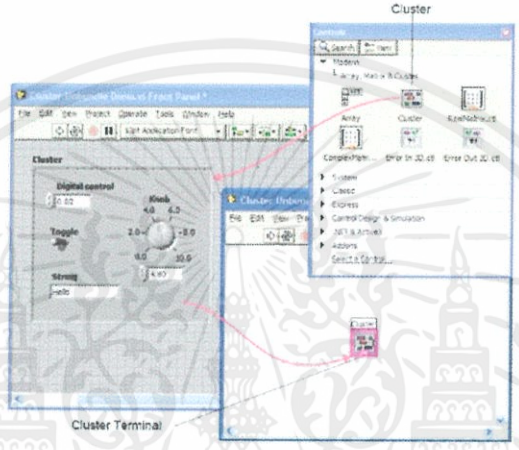
3.3.9 พื้นฐานการใช้งานคลัสเตอร์

คลัสเตอร์เป็นเครื่องมือ ที่ทำหน้าจัดเก็บข้อมูลคล้ายกับอาร์เรย์ เพียงแต่ว่าคลัสเตอร์สามารถเก็บข้อมูลที่ต่างชนิดกันได้ เช่นเก็บข้อมูลที่เป็น Numeric, Boolean และ Stringไว้ด้วยกันได้ วิธีการเรียกใช้งานคลัสเตอร์ทำได้โดยเลือกที่ Controls Palette» Modern» Array & Cluster» Cluster

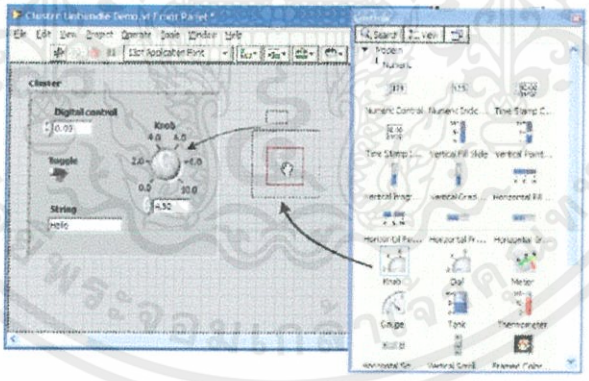
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การสร้างคลัสเตอร์ด้วย Control และ Indicator**

การสร้างคลัสเตอร์ด้วย Control และ Indicator นั้นสามารถที่จะกระทำได้เช่นเดียวกันกับการสร้างอาร์เรย์ แต่สำหรับการสร้างคลัสเตอร์จะแตกต่างจากการสร้างอาร์เรย์ตรงที่ข้อมูลในคลัสเตอร์ สามารถเป็นได้ทั้ง Numeric, String และ Boolean แต่ข้อมูลในอาร์เรย์จะเป็น Numeric เท่านั้นการใช้งานนั้นจะต้องมีฟังก์ชันที่ช่วยแยกชนิดของข้อมูลออกจากกันซึ่งจะได้กล่าวถึงในหัวข้อต่อไป



รูปที่ 3.91 แสดงการใช้งาน Cluster



รูปที่ 3.92 แสดงการสร้างคลัสเตอร์ด้วย Control และ Indicator

- **ฟังก์ชันของคลัสเตอร์**

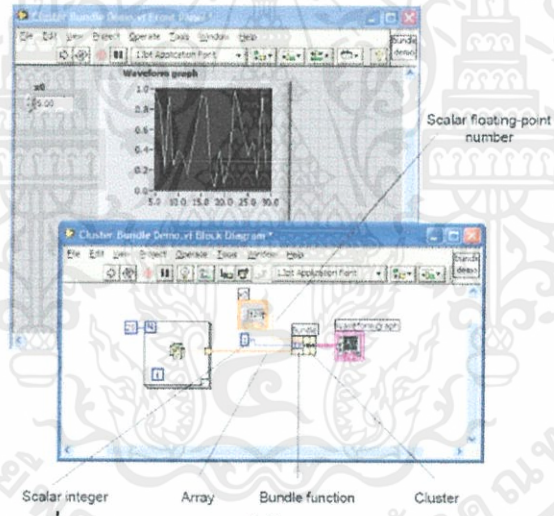
ฟังก์ชันที่ช่วยในการทำงานของคลัสเตอร์มี อยู่หลายแบบด้วยกัน ซึ่งในที่นี้จะขอกล่าวถึงฟังก์ชันที่ ถูกนำมาใช้งานบ่อยๆ 2 ฟังก์ชันคือ Bundle และ Unbundle ซึ่งวิที่เรียกใช้ฟังก์ชันทำโดยเลือกที่ Function Palette» All Function» Cluster แล้วเลือกฟังก์ชัน Bundle หรือ Unbundle ตามต้องการ

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ **Bundle Function**

BundleFunction เป็นฟังก์ชันใช้รวมเอาอินพุทหลายๆตัวมาสร้างเป็นคลัสเตอร์ใหม่ขึ้นมา โดยอินพุทนั้นสามารถเป็นได้ทั้ง ตัวเลขจำนวนจริง อาร์เรย์ หรือแม้กระทั่งคลัสเตอร์ เมื่อวางบล็อกของฟังก์ชัน Bundle ลงไปในหน้าต่างบล็อกไดอะแกรมฟังก์ชัน Bundle จะมีอินพุทเทอร์มินอล 2 เทอร์มินอลอยู่ทางซ้ายและเทอร์มินอลเอาท์พุท 1 เทอร์มินอลอยู่ทางขวาและอินพุทที่เป็นคลัสเตอร์จะอยู่ด้านบนของบล็อก สามารถที่จะเพิ่ม หรือลดจำนวนเทอร์มินอลอินพุทให้มีจำนวนเท่าที่จำเป็นต้องใช้งานโดยใช้เมาส์คลิ๊กที่บล็อก แล้วเลื่อนขึ้นลงในแนวตั้ง หรือใช้เมาส์ไปวางตำแหน่งตรงเทอร์มินอลอินพุทแล้วคลิ๊กเมาส์ขวา แล้วเลือก Add Input ถ้าต้องการเพิ่มเทอร์มินอล และเลือก Remove Input ถ้าต้องการลดเทอร์มินอล

ในรูปที่ 3.93 จะเป็นตัวอย่างการสร้างคลัสเตอร์ด้วย Bundle โดยมีอินพุท 3 ชนิดคือ floating-point number, Scalar integer และ array ที่สร้างจาก For Loop และมีเอาท์พุทที่แสดงผลด้วยกราฟ

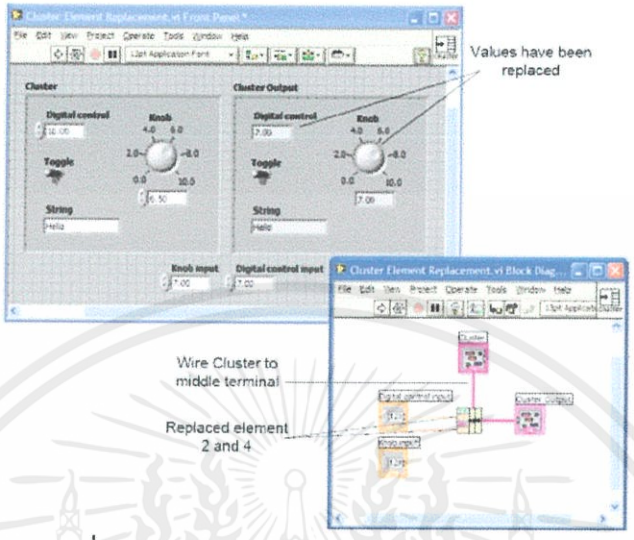


รูปที่ 3.93 แสดงการใช้งาน Bundle Function

การเพิ่มเทอร์มินอลอินพุทของฟังก์ชันBundle ทางซ้ายในขณะที่มีอินพุทที่เป็นคลัสเตอร์อยู่ด้วยโดยไม่มีผลกระทบต่อโปรแกรมเดิม ดังตัวอย่างในรูปที่ 3.93 ซึ่งมีคลัสเตอร์ 2 ตัว หนึ่งตัวเป็นอินพุทและอีกหนึ่งตัวเป็นเอาท์พุท การทำงานของโปรแกรม คือเอาท์พุทจะทำงานแปรตามอินพุท ถ้าต้องการควบคุมเอาท์พุทโดยที่ไม่ต้องไปแก้ไขอินพุท สามารถที่จะเพิ่มอินพุทให้กับ Bundle และสร้างอินพุทให้กับ Bundle ให้ตรงกับเอาท์พุทที่ต้องการควบคุมจากรูปที่ 3.94 การทำงานของโปรแกรมในขณะที่มีอินพุทที่เป็นคลัสเตอร์เพียงอย่างเดียว เอาท์พุทจะแปรตามอินพุท แต่เมื่อเพิ่มอินพุทแบบ knop และ อินพุท numeric control เข้าไปที่เทอร์มินอลอินพุทของ Bundle เมื่อทดสอบโปรแกรมจะเห็นได้ว่า เมื่อเปลี่ยนค่าอินพุทแบบ knop และอินพุท numeric control ในคลัสเตอร์อินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

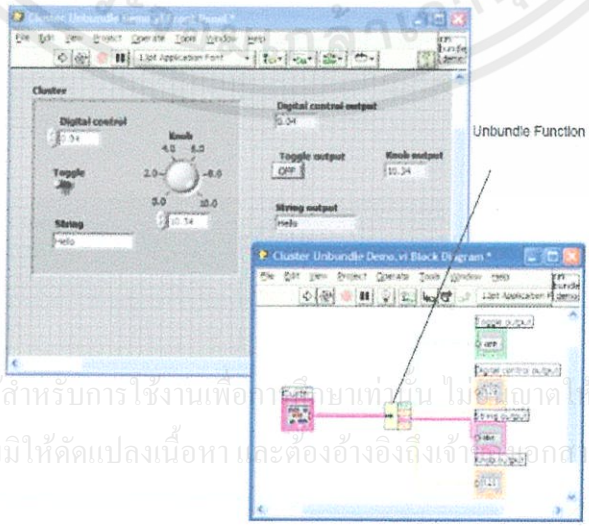
ค่าภายในคลัสเตอร์เอาท์พุทจะไม่เปลี่ยนแปลง แต่เมื่อเปลี่ยนค่าอินพุทที่เพิ่มเข้าไปค่าเอาท์พุทก็จะเปลี่ยนตาม



รูปที่ 3.94 แสดงการใช้งาน Bundle Function

- **Unbundle Function**

ในคลัสเตอร์นั้นจะประกอบไปด้วยชนิดของข้อมูล ที่มีความแตกต่างกันโดยส่วน Unbundle Function จะช่วยในการแยกข้อมูลอินพุทที่แตกต่างกัน ออกจากกันเพื่อจะนำเอาท์พุทนั้นไปใช้งานต่อไป ฟังก์ชันของ Unbundle ที่วางลงไปบนหน้าต่างบล็อกไดอะแกรมครั้งแรกจะมีอินพุท 1 อินพุท และมีเอาท์พุท 2 อินพุท เอาท์พุท Unbundle จะเปลี่ยนไปตามอินพุทที่เป็นคลัสเตอร์ตัวอย่างเช่น มีคลัสเตอร์อินพุทที่มีชนิดของข้อมูลแตกต่างกัน 4 ชนิด เมื่อต่อสายคลัสเตอร์เข้ากับอินพุทของ Unbundle เอาท์พุทของ Unbundle จะเปลี่ยนเป็น 4 เอาท์พุทโดยอัตโนมัติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

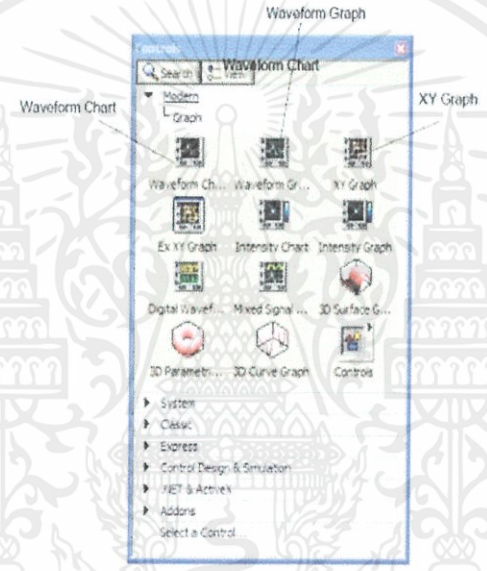
รูปที่ 3.95 แสดงการใช้งาน Unbundle Function

3.3.10 พื้นฐานการใช้งาน Charts และ Graph

- **Waveform Charts**

การแสดงข้อมูล ในลักษณะของ Chart เป็นการแสดงการพล็อตรูปคลื่นสัญญาณข้อมูลแบบหนึ่ง ที่สามารถพล็อตข้อมูลที่เป็นสเกลาร์และเป็นอาร์เรย์ได้ การเรียกใช้งาน คือ เลือกที่ Controls Palette » Graph » Waveform Chart

Waveform Chart ในโปรแกรม LabVIEW มีเพียงชนิดเดียวแต่มีคุณสมบัติในการแสดงข้อมูลที่ แตกต่างกัน 3 อย่างคือ Strip Chart, Scope Chart และ Sweep Chart ซึ่งข้อแตกต่างของ Chart ทั้ง 3 แบบ อธิบายได้ดังรูปที่ 3.96



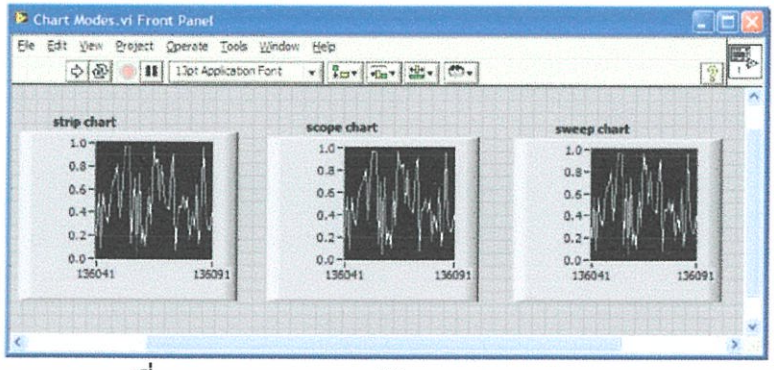
รูปที่ 3.96 แสดงวิธีการเรียกใช้งาน Waveform Chart

Strip Chart มีคุณสมบัติในการพล็อตข้อมูลไปจนกว่าจะหยุดพล็อตซึ่งจะสังเกตเห็นว่าเส้นของสัญญาณข้อมูลที่พล็อตนั้นจะเลื่อนจากขวาไปซ้ายต่อเนื่องกันไป

Scope Chart คุณสมบัติของการพล็อตแบบนี้ คือ จะแบ่งเป็นช่วงของข้อมูลโดยการกำหนดที่ แกนเวลาเมื่อโปรแกรมทำการพล็อตข้อมูลไปถึงค่าสุดท้ายของช่วงแรกแล้วการพล็อตข้อมูลของช่วงต่อไป โปรแกรมจะทำการลบกราฟในช่วงแรกไปแล้วเริ่มการพล็อตใหม่ โดยเวลาเริ่มต้นจะนับต่อจากค่าสุดท้ายของช่วงแรก

Sweep Chart จะทำงานคล้ายกับ Scope Chart แต่เมื่อเริ่มทำการพล็อตกราฟในช่วงต่อไปเส้นกราฟในช่วงแรกจะไม่ถูกลบจะทำงานในลักษณะพล็อตทับเส้นกราฟเดิน และจะมีเส้นสีแดงในแนวเลื่อนจากซ้ายไปขวา เพื่อแสดงถึงจุดการพล็อตกราฟครั้งล่าสุด คุณสมบัติทั้ง 3 ของ Waveform Chart แสดงดังรูปที่ 3.97 เป็นการแสดงค่าของข้อมูลที่เก็บอยู่ในอาร์เรย์เทียบกับเวลาที่รับเข้ามา

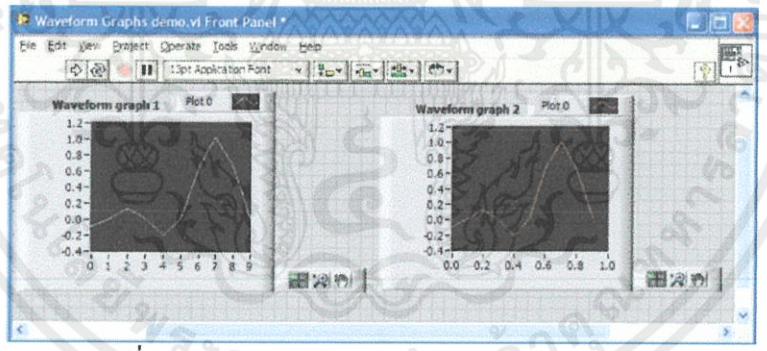
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.97 แสดงคุณสมบัติของ Waveform Chart

- Waveform Graphs

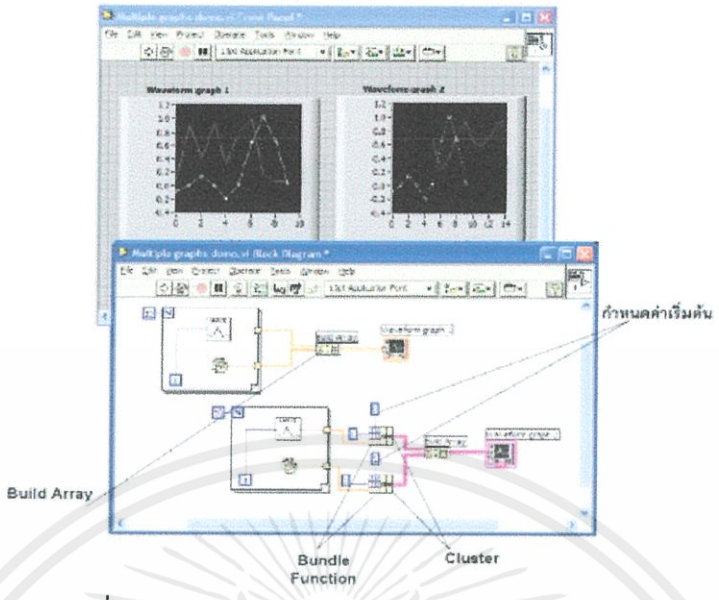
แตกต่างจาก Waveform Chart คือการพล็อตข้อมูลด้วย Waveform Graphs จะมีการกำหนด จำนวนของช่วงข้อมูลอย่างชัดเจน จำนวนข้อมูลทั้งหมดที่อยู่ในช่วงของกราฟจะถูกแสดงออกมาทั้งหมด การพล็อตกราฟหลายเส้นทำได้โดยการนำอาร์เรย์แต่ละตัวมาต่อรวม เพื่อสร้างอาร์เรย์ใหม่จำนวน เส้นกราฟที่แสดงจะเท่ากับมิติของอาร์เรย์ การเรียกใช้งาน Waveform Graphs ทำได้คือเลือกที่ Controls Palette» Modern» Graph» Waveform Graphs ลักษณะของการพล็อตกราฟลงบน waveform graph แสดงได้ดังรูปที่ 3.98



รูปที่ 3.98 แสดงคุณสมบัติของ Waveform Graph

นอกจากนี้แล้วยังสามารถใช้ Bundle และ Cluster มาช่วยในการกำหนดค่าเริ่มต้น และความถี่ในการแสดงข้อมูลแสดงดังรูปที่ 3.99 กราฟแรกจะแสดงการสร้างสัญญาณ 2 เส้นให้อยู่บนกราฟ เดียวกันโดยใช้ฟังก์ชัน Build Array เข้ามาช่วย กราฟที่สองจะเป็นการสร้างสัญญาณ 2 เส้นให้อยู่บนกราฟเดียวกัน และให้สามารถกำหนดค่าเริ่มต้นของกราฟแต่ละเส้นได้ ซึ่งจะนำเอา Bundle และ Cluster โดยกราฟเส้นแรกให้เริ่มต้นที่ค่า 0 แสดงค่าทั้งหมด 10 ค่า และอีกเส้นให้เริ่มต้นที่ 5 และค่าที่ แสดงทั้งหมด 10 ค่า เช่นกัน ดังแสดงในรูปที่ 3.99

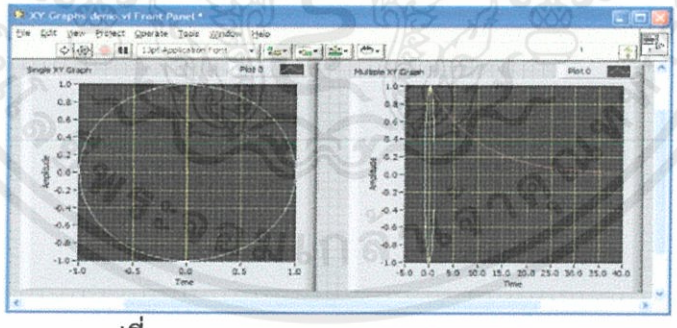
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.99 แสดงคุณสมบัติของ Waveform Graph

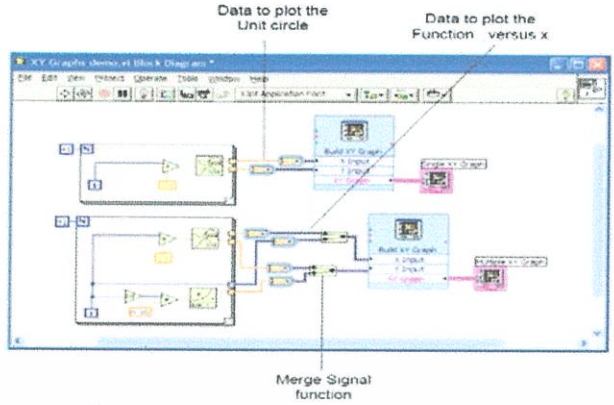
- **XY Graphs**

การพล็อตกราฟใน XY Graph เป็นการนำค่าของอาร์เรย์ 2 ชุดมาพล็อตในระนาบ XY โดย กำหนดให้อาร์เรย์ตัวที่ 1 และตัวที่ 2 เป็นค่าที่แสดงในแกน X และแกน Y ตามลำดับ โดยจะมีอุปกรณ์ที่ชื่อ Build XY Graph จะช่วยในการแปลงข้อมูลในอาร์เรย์ให้เป็นค่าใดๆรูปที่ 3.100 และรูปที่ 3.101



รูปที่ 3.100 Front Panel ของ XY Graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



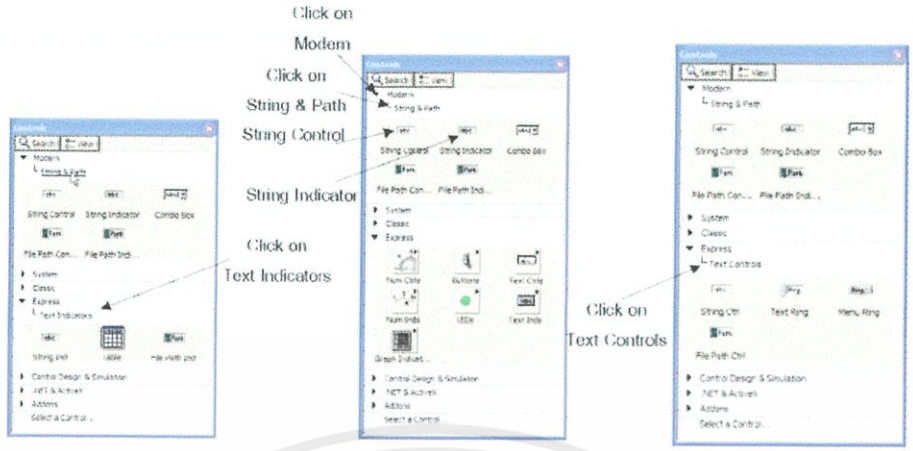
รูปที่ 3.101 Block Diagram ของ XY Graph

3.3.11 พื้นฐานเกี่ยวกับ String

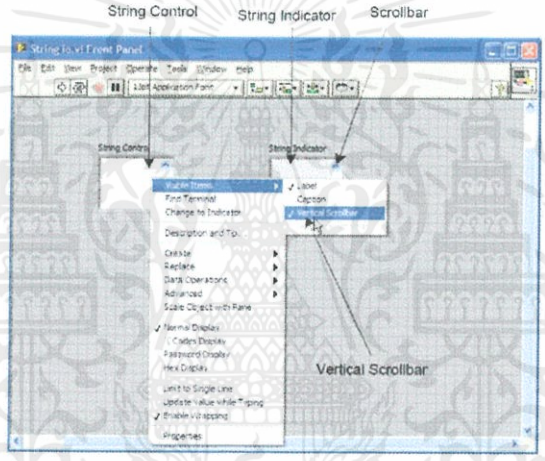
สตริงเป็นชนิดของข้อมูลที่อยู่ในรูปแบบของตัวอักษร ซึ่งสามารถที่จะกำหนดให้แสดงหรือไม่แสดงค่าก็ได้ หลายครั้งที่สตริงถูกใช้ในลักษณะของรหัส ASCII ในโปรแกรม LabVIEW สตริงจะถูกใช้ใน instrument control การใช้งานข้อมูลชนิดสตริง และข้อมูลชนิดตัวเลขจะต้องมีการเปลี่ยนชนิดของข้อมูล เพื่อให้ชนิดของข้อมูลเป็นชนิดเดียวกันก่อน ซึ่งก็จะมีวิธีการเปลี่ยนชนิดของข้อมูลซึ่งจะได้กล่าวในหัวข้อต่อไป ชนิดของข้อมูลก่อนที่จะบันทึกลงไฟล์บนฮาร์ดดิสก์จะต้องแปลงให้เป็นข้อมูลชนิดสตริงก่อน ในหัวข้อนี้จะพูดถึงสตริงและการใช้ไฟล์อินพุทไฟล์เอาท์พุท (File I/O)

การใช้งาน String Control และ String Indicator ที่อยู่ใน String & Path ซึ่งเป็นพาเลทย่อยของ Control» Modern ซึ่งแสดงในรูปที่ 3.102 ซึ่งเราสามารถเลือกใช้งานได้ทั้งแบบ Text Controls และ Text Indicators ซึ่งอยู่ในพาเลทย่อยของ Controls»Express การใช้งาน String Control และ String Indicator ทำได้โดยการคลิ๊กเมาส์ทางซ้ายค้างที่สัญลักษณ์ของ String Control และ String Indicator แล้วลากไปวางที่หน้าต่างฟร้อนท์พาเนล ซึ่งจะเห็นลักษณะของ String Control และ String Indicator เป็นบล็อกสี่เหลี่ยมสำหรับรับข้อมูลอินพุทและแสดงผลตามลำดับ ถ้าบล็อกนั้นมีขนาดเล็กไม่เพียงพอสำหรับข้อมูลที่ป้อนเข้าไปก็สามารถขยายได้ใช้เมาส์คลิ๊กแล้วลากขยายให้ใหญ่ขึ้น หรือถ้าขนาดของฟร้อนท์พาเนล มีขนาดจำกัดซึ่งทำให้ไม่สามารถขยายขนาดของบล็อกสตริงได้ก็สามารถเพิ่ม Scrollbar เข้าไปเพื่อให้สามารถกรอกข้อมูลชนิดสตริง ได้เป็นจำนวนมากขึ้น ซึ่งทำได้โดยคลิ๊กเมาส์ ทางขวาที่บริเวณบล็อกของ String Control หรือ String Indicator ที่ต้องการเพิ่ม Scrollbar แล้วเลือก Visible Item» Vertical Scrollbar ดังรูปที่ 3.102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.102 การเข้าถึง String controls และ String Indicators



รูปที่ 3.103 การเพิ่ม Scrollbar ให้กับบล็อก String Indicator และ String Control

สกอร์บาร์ที่เพิ่มเข้าไปใน String Control หรือ String Indicator ดังแสดงในรูปที่ 3.103 สกอร์บาร์ที่เพิ่มเข้าไปแสดงอยู่ในแนวตั้ง

การแสดงข้อมูลในบล็อกของ String Control และ String Indicator นั้นสามารถกำหนดให้แสดงได้ 4 รูปแบบ คือ

1. Normal Display จะแสดงข้อความเป็นแบบปกติตามที่ผู้ใช้ป้อนเข้าไป

2. “\” Codes Display จะแสดงข้อความในลักษณะของ Codes โดยมีเครื่องหมาย “\” เป็น ตัวขั้นแทนการเว้นวรรค

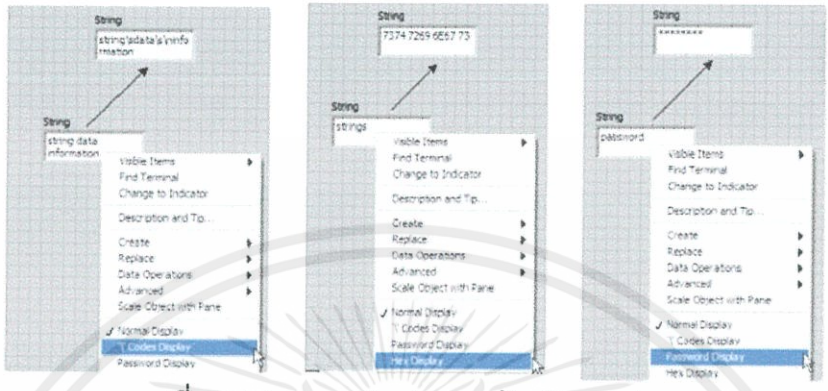
3. Password Display จะแสดงข้อความแบบรหัสผ่านแทนด้วย “*”

4. Hex Display จะแสดงข้อความในลักษณะของเลขฐาน 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ลักษณะของข้อความหลังจากที่ถูกเข้ารหัสในรูปแบบต่างๆ แล้วสามารถแสดงให้เห็นได้ ดังรูปที่ 3.104 ซึ่งในรูปจะแสดงวิธีการเข้ารหัสแบบ “\” Codes Display, Hex Display และ Password Display ตามลำดับ การกำหนดรูปแบบของชนิดของข้อมูลแบบสตริงว่าจะให้แสดง

อยู่ในรูปแบบใดนั้นสามารถทำได้โดยคลิ๊กเมาส์ทางขวาบริเวณบล็อกของ String Control และ String Indicator จะปรากฏเมนูย่อยขึ้นมาซึ่งในเมนูย่อยนี้จะมีชุดคำสั่งสำหรับกำหนดรูปแบบการแสดงผลอยู่ทั้งหมด 4 รูปแบบ

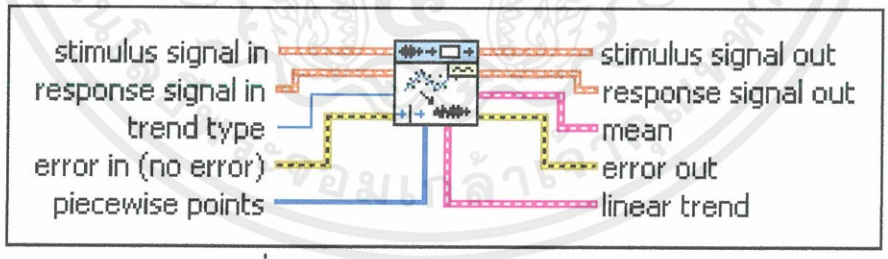


รูปที่ 3.104 แสดงตัวอย่างการเข้ารหัสแบบต่างๆ

3.3.12 การหาเอกลักษณ์ของกระบวนการโดย System identification toolkit

3.3.12.1 SI Remove Trend VI

เป็นการตัดค่าเฉลี่ย และ ค่าความโน้มเอียงในสภาวะอยู่ตัวแบบเชิงเส้น (linear steady-state trends) ออกจากสัญญาณกระตุ้น (stimulus signal) และสัญญาณตอบสนอง (response signal) ทำการเชื่อมสัญญาณอินพุทของ stimulus signal in และ response signal in เพื่อที่จะทำการวิเคราะห์หา กรณี เพื่อนำมาใช้หรือนำมาเลือกเอง



รูปที่ 3.105 แสดง SI Remove Trend

stimulus signal in คือ รูปสัญญาณอินพุทของสัญญาณกระตุ้น (stimulus signal)

response signal in คือ รูปสัญญาณอินพุทของสัญญาณตอบสนอง (response signal)

trend type คือ กระบวนการที่ใช้เพื่อตัด steady-state trends ออกจากสัญญาณกระตุ้นและสัญญาณตอบสนอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0 คือ **mean** (ค่า Default) - ค่าเฉลี่ยตัดออกจากสัญญาณ ถ้า piecewise point มี breakpoint ที่เท่ากับจำนวน sample แล้ว VI จะทำการตัดค่าเฉลี่ยออกจากแต่ละส่วนของข้อมูล มิฉะนั้น VI จะทำการหักค่าเฉลี่ยออกจากเซตเริ่มต้นของตัวอย่างข้อมูล

1 คือ **linear**—นำค่าความโน้มเอียงแบบเชิงเส้นอันดับหนึ่ง (first-order linear trend) ตัดออกจากสัญญาณหลังจากที่ least squares fit ถ้า piecewise point มี breakpoints ที่เท่ากับจำนวน sample แล้ว VI จะทำการตัดค่าทั้งค่า continuous, piecewise และ linear trends แต่ถ้าไม่ใช่ VI จะทำการตัดค่าของ single linear trend ออกจาก เซตเริ่มต้นของตัวอย่างข้อมูล

error in คือ เงื่อนไขของค่าความคลาดเคลื่อน ที่เกิดขึ้นก่อนที่โหนดนี้ทำงาน โดยอินพุทนี้จะทำให้ standard error in ทำงาน

piecewise points เป็นการระบุ breakpoints ที่จำนวน sample ในสัญญาณกระตุ้น และสัญญาณตอบสนอง

stimulus signal out จะส่งรูปคลื่นสัญญาณที่มีแนวโน้มการถูกขจัดของของสัญญาณกระตุ้นออกมา

response signal out จะส่งค่ารูปสัญญาณที่ทำการ detrend ของสัญญาณตอบสนองออกมา

mean จะส่งค่า mean ของสัญญาณก่อนที่จะทำการ remove trend ออกมา จะใช้ได้ก็ต่อเมื่อ ใช้ mean เป็น trend type

stimulus จะส่งค่า mean ของ แต่ละส่วนของสัญญาณกระตุ้น

response จะส่งค่า mean ของ แต่ละส่วนของสัญญาณตอบสนอง

error out ประกอบไปด้วยข้อมูลของค่าความคลาดเคลื่อน ซึ่งเอาท์พุตนี้จะใช้งาน standard error out

linear trend จะส่งค่า linear trend ของสัญญาณ ก่อนที่จะทำการ remove trend ออกมา จะใช้ได้ก็ต่อเมื่อ ใช้ linear เป็น trend type

stimulus slope จะส่งค่า slope ของแต่ละส่วนของ สัญญาณ stimulus ออกมา

stimulus intercept จะส่งค่า intercept ของแต่ละส่วนของ สัญญาณ stimulus ออกมา

response slope จะส่งค่า slope ของแต่ละส่วนของ สัญญาณ response ออกมา

response intercept จะส่งค่า intercept ของแต่ละส่วนของสัญญาณ response ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

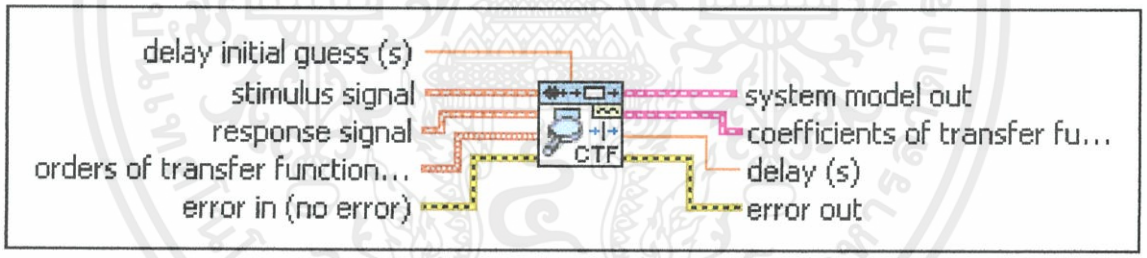
3.3.12.2 SI Estimate Transfer Function Model VI

เป็นการประมาณค่าของพารามิเตอร์ของ continuous transfer function model หรือ discrete transfer function model ของ transfer function model สำหรับระบบที่ไม่ทราบคุณสมบัติ โดย VI นี้ จะประมาณค่าเฉพาะ transfer function ระหว่าง stimulus และ response เท่านั้น ซึ่ง transfer function ระหว่าง noise และ response จะถูกสมมติเป็น 1 ซึ่งเราจะต้องทำการเลือกกรณีของรูปแบบเพื่อนำมาใช้

The SI Estimate Transfer Function Model VI รองรับระบบ SISO และระบบ MISO ของ continuous model ซึ่งจะสามารถใส่การประมาณเริ่มต้นเข้าไปในระบบ delay ได้โดยใช้อินพุท delay initial guess หรือ delay initial guesses

The SI Estimate Transfer Function Model VI จะรองรับเฉพาะระบบ SISO สำหรับ discrete model เท่านั้น ซึ่งจะมี 3 วิธี ในการประมาณค่าของ discrete models คือ direct indirect และ joint input-output ซึ่ง วิธี direct จะใช้สำหรับทั้ง ข้อมูลแบบวงเปิด (open-loop) และ วงปิด (close-loop) ส่วนวิธี indirect และ joint input-output จะใช้สำหรับข้อมูลแบบวงปิด (close-loop) เท่านั้น

ตัว VI นี้ยังมีวิธีกำลังสองเฉลี่ยน้อยที่สุด (least mean squares) เพื่อประมาณค่าของ continuous SISO model โดยการใช้สัญญาณ step response



รูปที่ 3.106 แสดง SI Estimate Continuous Transfer Function Model

delay initial guess (s) คือ ค่าประมาณการหน่วงเวลาเริ่มต้นของระบบ ในหน่วยวินาที โดยค่าตั้งต้น คือ NaN ซึ่งไม่มีการหน่วงเวลาใน transfer function

stimulus signal คือ ชุด array ที่ใช้แทนสัญญาณ stimulus

response signal คือ ชุด array ที่ใช้แทนสัญญาณ response

orders of transfer function model คืออันดับตัวเศษและอันดับตัวส่วน ของ transfer function model

num order คืออันดับตัวเศษของแบบจำลองของระบบ ซึ่งค่าตั้งต้น คือ 1

den order คืออันดับตัวส่วนของแบบจำลองของระบบ โดยค่าตั้งต้น คือ 2 ซึ่งค่าของ den order จะต้องมากกว่าค่าของ num order เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

error in คือ เงื่อนไขของค่าความคลาดเคลื่อน ที่เกิดขึ้นก่อนที่โหนดนี้ทำงาน โดยอินพุทนี้จะทำให้ standard error in ทำงาน

system model out จะส่งค่าข้อมูลต่างๆ เช่น โครงสร้างแบบจำลอง, nominal หรือ estimate parameters, identification result และอื่น ๆ โดยใช้ Model Management Vls เพื่อกู้ข้อมูล system model out กลับคืน

- **หมายเหตุ** สามารถใช้ customize system model probe ในการสังเกตข้อมูลที่ออกมาเมื่อทำการดีบัก block diagram ที่สร้างร่วมกับการใช้ System Identification Vls โดยคลิ๊กขวาที่ system model wire และเลือก Custom Probe >> SI system Model จากเมนู shortcut ในการใช้ System model probe

coefficients of transfer function model จะส่งค่าสัมประสิทธิ์ของแบบจำลองฟังก์ชันถ่ายโอน

numerator จะส่งค่า numerator coefficient ของ transfer function model แบบ ascending order ซึ่งค่าสัมประสิทธิ์จะเป็นตามรูปแบบ $b_0 + b_1 s + ... + b_m s^m$.

denominator จะส่งค่า denominator coefficients ของ transfer function model ในแบบ ascending order ซึ่งค่าสัมประสิทธิ์จะเป็นตามรูปแบบ $a_0 + a_1 s + ... + a_n s^n$.

delay (s) จะส่งค่าประมาณเวลาหน่วงของระบบในหน่วยวินาที

error out ประกอบไปด้วยข้อมูลของค่าความคลาดเคลื่อน ซึ่งเอาท์พุตนี้จะใช้งาน standard error out

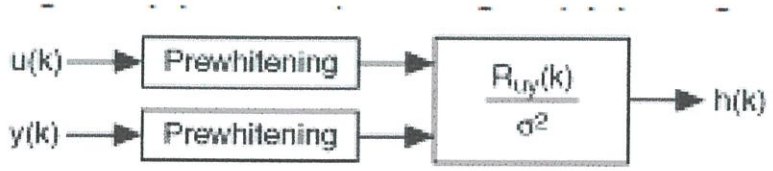
3.3.12.3 SI Estimate Impulse Response VI

การประมาณผลตอบสนองแบบอิมพัลส์ของระบบที่ไม่รู้จัก โดยใช้วิธีการวิเคราะห์ค่าสหสัมพันธ์แบบ prewhitening-based (prewhitening-based correlation analysis method) และวิธีกำลังสองเฉลี่ยน้อยที่สุด (least squares method)

กรณีความสัมพันธ์ของ SI Estimate Impulse Response VI ใช้ตัวกรองแบบอิมพัลส์จำกัด (FIR) ทำสัญญาณสีขาว และคำนวณค่าสัมประสิทธิ์ของตัวกรอง FIR จาก autoregressive (AR) โปรเซสเซอร์ ซึ่งอินพุต AR order กำหนด prewhitening behavior

รูปต่อไปนี้ แสดงให้เห็นถึงโครงสร้างของกรณีความสัมพันธ์ของ SI Estimate Impulse Response VI โดย VI นี้จัดการกระบวนการ prewhitening สัญญาณกระตุ้น u(k) และสัญญาณตอบสนอง y(k) โดยใช้คู่ของ identical AR processor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

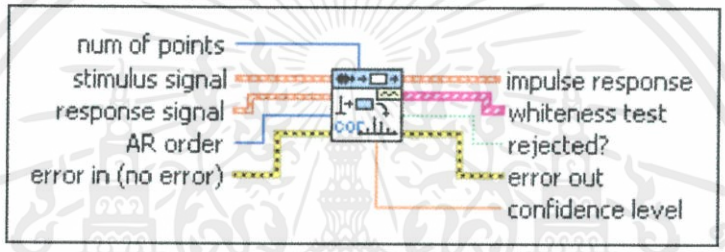


รูปที่ 3.107 โครงสร้างของกรณีความสัมพันธ์ของ SI Estimate Impulse Response VI

ในแผนภาพ ข้างต้น $R_{uy}(k)$ เป็น cross correlation ระหว่างสัญญาณกระตุ้นสีขาว u(k) และ สัญญาณตอบสนองสีขาว y(k)

σ คือค่าเบี่ยงเบนมาตรฐานของสัญญาณกระตุ้นสีขาว u(k)

$h(k)$ คือ ผลตอบสนองอิมพัลส์ของระบบ



รูปที่ 3.108 SI Estimate Impulse Response

num of points เป็นการระบุจำนวนของจุด ที่จะคำนวณผลตอบสนองแบบอิมพัลส์

stimulus signal เป็นการระบุรูปคลื่นอินพุทของสัญญาณกระตุ้น

response signal เป็นการระบุรูปคลื่นอินพุทของสัญญาณตอบสนอง

AR order เป็นการระบุออเดอร์ของแบบจำลอง AR ที่จะใช้ ในการประมานค่าตัวกรองสำหรับ prewhitening โดย AR ต้องมากกว่า หรือเท่ากับ 0

error in อธิบายถึงเงื่อนไขของค่าความคลาดเคลื่อน ที่เกิดขึ้นก่อนที่โหนดนี้ทำงาน โดยอินพุทนี้จะทำให้ standard error in ทำงาน

impulse response เปลี่ยนเป็นรูปคลื่นสัญญาณของค่าประมานของผลตอบสนองแบบอิมพัลส์ของระบบกลับไป

whiteness test แสดงสหสัมพันธ์ของตัวคลาดเคลื่อนของสัญญาณกระตุ้นที่ใกล้เคียงกับข้อมูล ที่ confidence region 99% หากสัญญาณกระตุ้นใกล้เคียงกับข้อมูลที่ส่งมาแล้วนั้น ความสัมพันธ์ระหว่างค่าความคลาดเคลื่อนที่มากที่สุด ของสัญญาณกระตุ้น ซึ่งยกเว้นจุดแรกของสัญญาณ จะอยู่ภายใน confidence region แต่ถ้าบางจุดที่เกินกว่า confidence region โดยเฉพาะอย่างยิ่งจุดที่สองและสาม จะเพิ่ม AR order เพื่อปรับปรุง prewhitening

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา รวมทั้งต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

rejected? แสดงให้เห็นถึงการประมาณค่าผลตอบสนองแบบอิมพัลส์ของ VI นี้มีความน่าเชื่อถือ ถ้า rejected? เป็น FALSE การประมาณค่าผลตอบสนองแบบอิมพัลส์นั้นมีความน่าเชื่อถือ

และความเสี่ยงของการยอมรับของการประมาณค่าผลตอบสนองแบบอิมพัลส์ที่ไม่ถูกต้องเท่ากับ 5 % ถ้า rejected? เป็น TRUE การประมาณค่าผลตอบสนองแบบอิมพัลส์นั้นจะไม่น่าเชื่อถือ ซึ่งการเพิ่ม AR order หรือการออกแบบการทดลองใหม่นั้น ปรับปรุงการประมาณค่าผลตอบสนองแบบอิมพัลส์

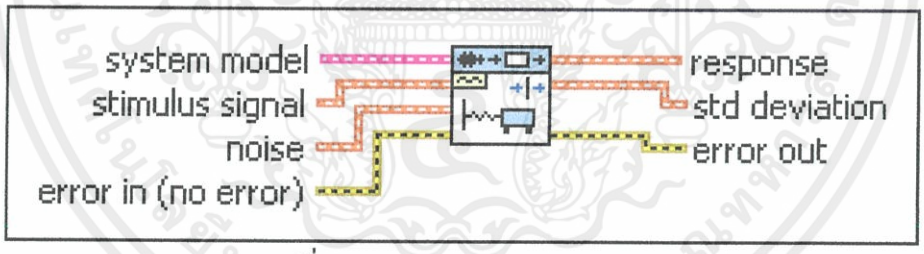
error out ประกอบด้วยข้อมูลที่คลาดเคลื่อน ซึ่งค่าเอาท์พุทนี้จะให้ค่า standard error out ในการทำงาน

confidence level ส่งค่าระดับความเชื่อมั่นของผลตอบสนองแบบอิมพัลส์ออกไป โดยค่าของผลตอบสนองแบบอิมพัลส์ระหว่างค่าบวกและค่าลบของระดับความเชื่อมั่นนั้น ไม่มีความเป็นนัยสำคัญกัน และสามารถพิจารณาให้มีค่าเท่ากับ 0

3.3.12.4 SI Model Simulation VI

เป็นการจำลองผลตอบสนองของ system model เข้าไปในสัญญาณอินพุท โดยทำการต่อสายอินพุตข้อมูลเข้ากับ stimulus signal และ noise เพื่อที่จะกำหนดรูปแบบของกรณีการใช้งานมาใช้งาน

- **หมายเหตุ** สำหรับ ARX (MIMO), state-space, transfer function และ zero-pole-gain model นั้น VI จะไม่คืนค่า std deviation ของการจำลองผลตอบสนอง และสำหรับ AR models, VI จะไม่สนใจสัญญาณ stimulus



รูปที่ 3.109 SI Model Simulation

system model จะประกอบไปด้วยข้อมูลของ โครงสร้างโมเดล, nominal parameter หรือ estimate parameter, identification result และอื่น ๆ โดยใช้ Model Management Vis ในการรับข้อมูลของระบบ

- **หมายเหตุ** สามารถใช้ customize system model probe ในการสังเกต ข้อมูลที่ออกมาจาก system model wires เมื่อทำการ debug block diagram ที่สร้างโดย System Identification Vis โดยคลิ๊กขวาที่ system model wire และ เลือก Custom Probe >> SI system model จาก shortcut menu ในการใช้ system model probe

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ ทุกครั้งที่มีการนำไปใช้

stimulus signal คือ รูปสัญญาณอินพุทของ stimulus signal

noise คือรูปสัญญาณ noise ที่ต้องการใช้ในระบบ

error in อธิบายถึงเงื่อนไขของค่าความคลาดเคลื่อน ที่เกิดขึ้นก่อนที่โหนดนี้ทำงาน โดยอินพุทนี้จะทำให้ standard error in ทำงาน

response จะส่งค่าจำลองของรูปสัญญาณ response ของระบบ

std deviation จะส่งค่า standard deviation ของ simulated system response ซึ่งแต่ละองค์ประกอบคือส่วนเบี่ยงเบนมาตรฐานของ corresponding value ใน response

error out ประกอบด้วยข้อมูลของข้อผิดพลาด เอาท์พุตนี้ให้ standard error out ในการทำงาน

3.4 สรุป

องค์ประกอบต่างๆ ที่เกี่ยวข้องกับกระบวนการควบคุมอุณหภูมิสำหรับเตาอบขนาดเล็ก ที่ได้กล่าวไปนั้นไม่ว่าจะเป็นอุปกรณ์ เครื่องมือ รวมทั้งโปรแกรม LabVIEW ตั้งแต่พื้นฐานเบื้องต้นและการนำมาหาเอกลักษณ์ของระบบที่นำมาใช้ในกระบวนการ ล้วนแล้วแต่มีความสำคัญเป็นอย่างยิ่ง จะขาดสิ่งหนึ่งสิ่งใดไปไม่ได้ เพราะอาจจะทำให้กระบวนการเตาอบขนาดเล็กนี้ทำงานไม่สมบูรณ์แบบ ตามที่ต้องการ สิ่งเหล่านี้จะเป็นประโยชน์เพื่อช่วยในการหาเอกลักษณ์กระบวนการควบคุมเตาอบขนาดเล็กด้วยวิธีผลตอบสนองแบบอิมพัลส์ได้อย่างแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การนำเสนอผลของข้อมูลและการวิเคราะห์ข้อมูล

4.1 กล่าวนำ

ในบทที่ 4 นั้นจะกล่าวถึง การนำเสนอผลของข้อมูลจากการศึกษาทฤษฎีที่เกี่ยวข้องทั้งหมด รวมทั้ง องค์ประกอบด้านฮาร์ดแวร์ และองค์ประกอบด้านซอฟต์แวร์ดังที่กล่าวมาแล้ว นำมาหาเอกลักษณ์ของระบบด้วยวิธีผลตอบสนองแบบอิมพัลส์ โดยใช้โปรแกรม LabVIEW ในการควบคุมกระบวนการ

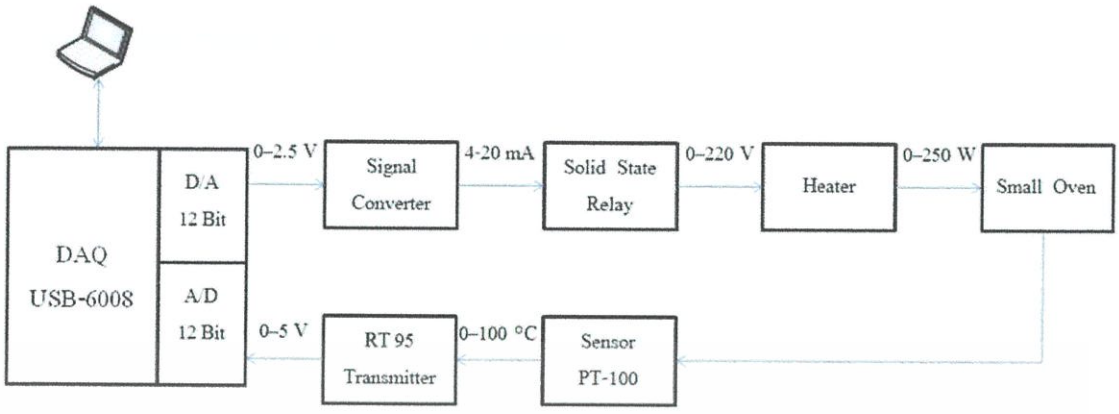
สำหรับการทดลอง จะกำหนดค่าควบคุมเอาท์พุต (Manipulated variable; MV) ที่ค่าต่างๆ จำนวน 6 ค่า เพื่อพิจารณาผลตอบสนองของกระบวนการและนำผลตอบสนองของกระบวนการที่ได้นำมาหาโมเดลทางคณิตศาสตร์ โดยใช้โปรแกรม LabVIEW ในการหาค่าประมานของฟังก์ชันถ่ายโอน

4.2 กระบวนการที่ใช้ในการทดลอง

รูปที่ 4.1 แสดงกระบวนการที่ใช้ในการทดลอง

จากรูปที่ 4.1 แสดงกระบวนการที่ใช้ในการทดลอง ซึ่งคือกระบวนการเตาอบขนาดเล็ก จากองค์ประกอบด้านฮาร์ดแวร์ที่กล่าวไว้ในบทที่ 3 สามารถนำมาออกแบบกระบวนการที่ใช้ในการทดลองได้ตามโครงสร้างไดอะแกรมของกระบวนการดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงลักษณะการออกแบบกระบวนการ

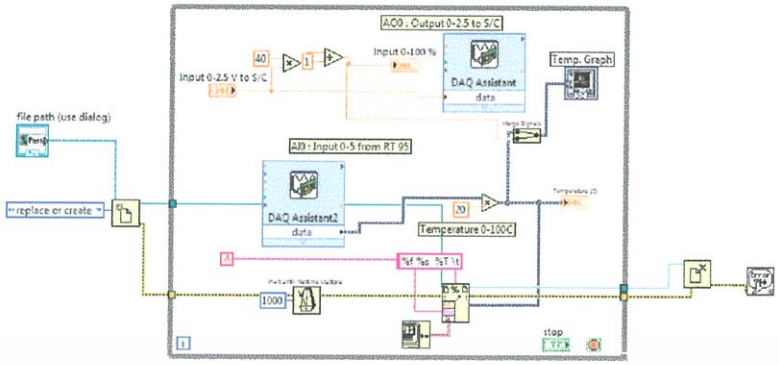
การออกแบบการทำงานของกระบวนการ แบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็นส่วนควบคุมและส่วนกระบวนการเตาอบ สามารถอธิบายการทำงานของกระบวนการ โดยเริ่มจากการกำหนดค่าเป้าหมาย (SP) ซึ่งกำหนดในโปรแกรม LabVIEW ได้โดยตรงให้คอนโทรลเลอร์ทำงานตามต้องการ จากนั้นสัญญาณ 0-2.5V จะเข้าไปที่ Signal Converter เพื่อทำหน้าที่แปลงสัญญาณ 0-2.5V เป็นสัญญาณมาตรฐาน 4-20 mA ส่งผ่านไปยัง Final Control Element (Heater) เพื่อจ่ายพลังงานความร้อนให้กับกระบวนการเตาอบ โดยการทำงานของตัวทำความร้อนถูกควบคุมโดย Solid State Relay (SSR) ซึ่งจะสร้างสัญญาณพัลส์ควบคุมตามสัญญาณที่ได้รับ (4-20 mA) ส่งผลให้อุณหภูมิของกระบวนการเตาอบจึงเกิดการเปลี่ยนแปลงขึ้น RTD (PT-100) จะทำการตรวจจับค่าความร้อนจากเตาอบในรูปของแรงดันไฟฟ้า เนื่องจากแรงดันที่ได้ไม่เหมาะสมกับการนำไปใช้งาน จึงต้องนำอุปกรณ์ RTD Transmitter ซึ่งเป็นตัวแปลงสัญญาณที่รับจาก RTD (PT-100) เพื่อแปลงให้เป็นสัญญาณมาตรฐาน 0-5V จากนั้นจึงนำสัญญาณที่ได้ส่งไปยังตัวควบคุม สัญญาณจะถูกส่งเข้าไปยังด้าน A/D ของ DAQ เพื่อแปลงสัญญาณที่ได้รับจากสัญญาณอนาล็อก ให้อยู่ในรูปของสัญญาณดิจิตอล และนำไปประมวลผลต่อไป

4.3 การจำลองกระบวนการ

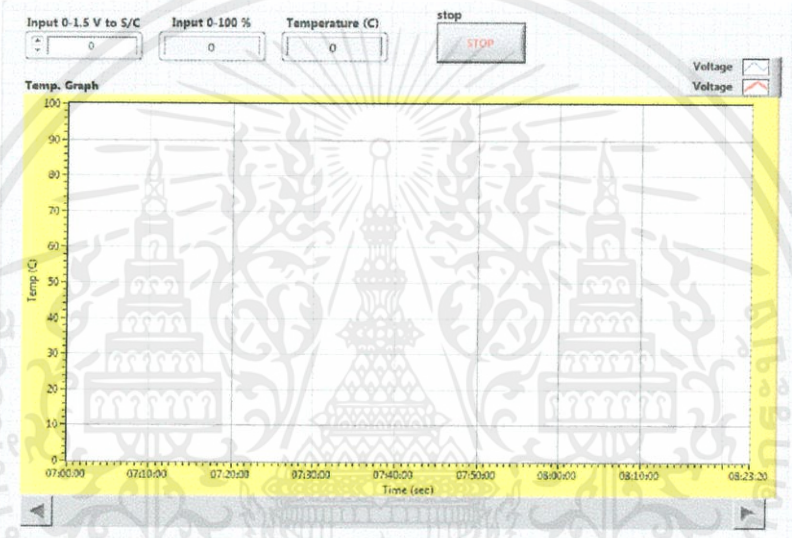
การทดลองควบคุมกระบวนการเตาอบขนาดเล็กวิธีวงรอบเปิด ในการควบคุมกระบวนการจะทำการป้อนค่า ควบคุมเอาท์พุต (Manipulated variable; MV) ให้ระบบค่าหนึ่งแล้วรอจนกว่าจะเข้าสู่สภาวะคงที่ ที่ค่าหนึ่ง ลงในโปรแกรม LabVIEW ที่เขียนขึ้นมามีขั้นตอนการทำงานดังนี้

1. เขียนโปรแกรม LabVIEW ในส่วนของ Block Diagram เพื่อใช้ในการควบคุมกระบวนการดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

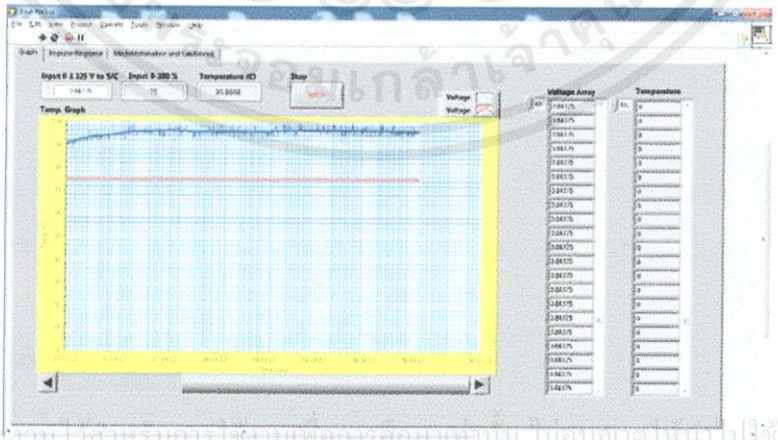


รูปที่ 4.3 แสดงโปรแกรมที่ใช้ควบคุมกระบวนการ



รูปที่ 4.4 แสดงแสดงหน้าต่างพาเนล (Front Panel)

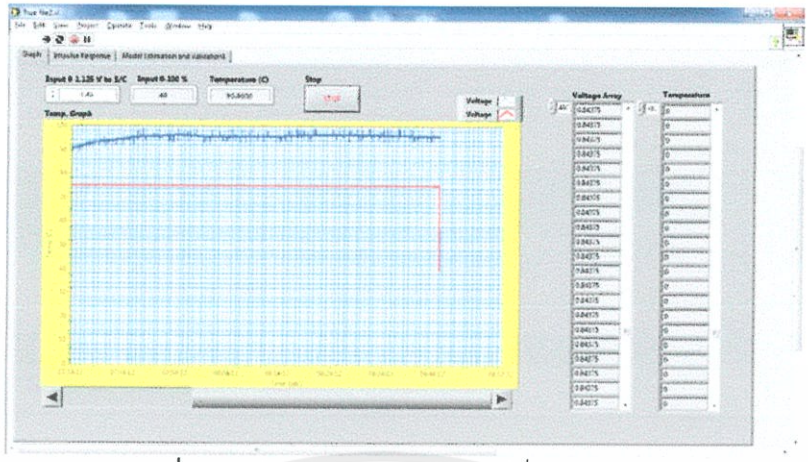
2. ใส่ค่า MV = 75% ลงในโปรแกรม LabVIEW จะได้กราฟผลตอบสนองดังรูปที่ 4.5



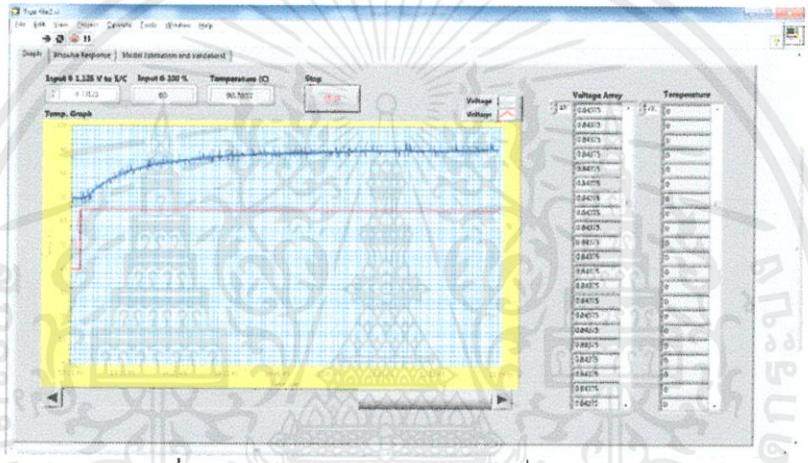
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 แสดงกราฟผลตอบสนองที่ MV = 75%

3. ปรับลดค่า MV = 40% ลงในโปรแกรม LabVIEW จะได้กราฟผลตอบสนองดังรูปที่ 4.6 และรูป ที่ 4.7

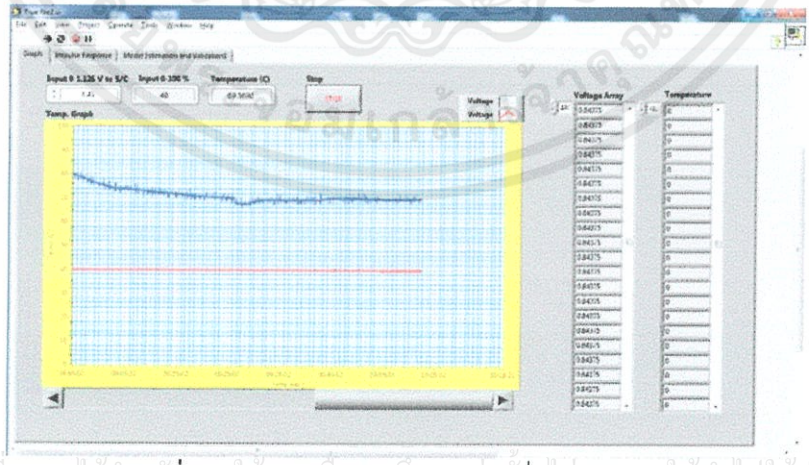


รูปที่ 4.6 แสดงกราฟผลตอบสนองที่ MV = 40%



รูปที่ 4.7 แสดงกราฟผลตอบสนองที่ MV = 40%

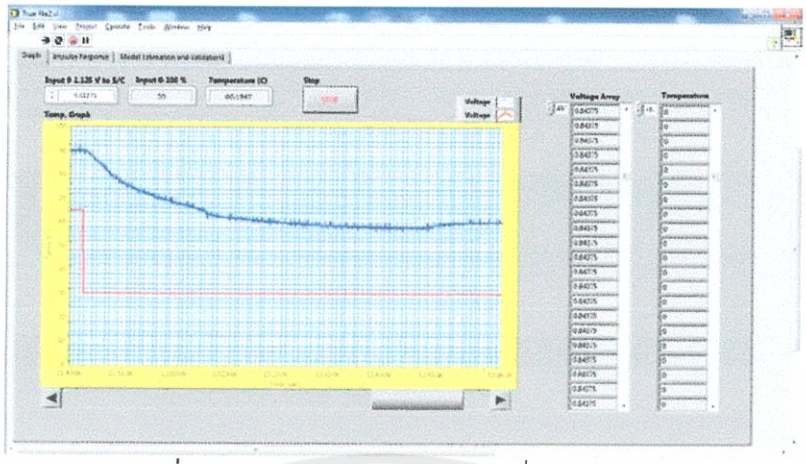
4. ปรับค่า MV = 65% ลงในโปรแกรม LabVIEW จะได้กราฟผลตอบสนองดังรูปที่ 4.8



รูปที่ 4.8 แสดงกราฟผลตอบสนองที่ MV = 65%

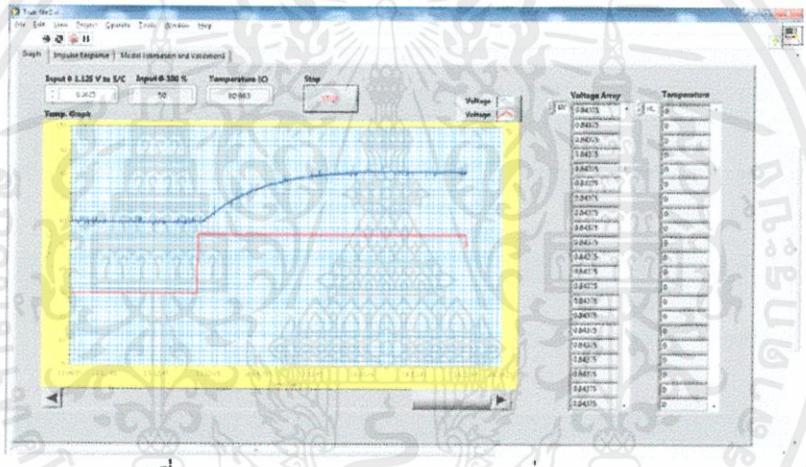
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ใส่ค่า MV = 30% ลงในโปรแกรม LabVIEW จะได้กราฟผลตอบสนองดังรูปที่ 4.9



รูปที่ 4.9 แสดงกราฟผลตอบสนองที่ MV = 30%

6. ใส่ค่า MV = 50% ลงในโปรแกรม LabVIEW จะได้กราฟผลตอบสนองดังรูปที่ 4.10

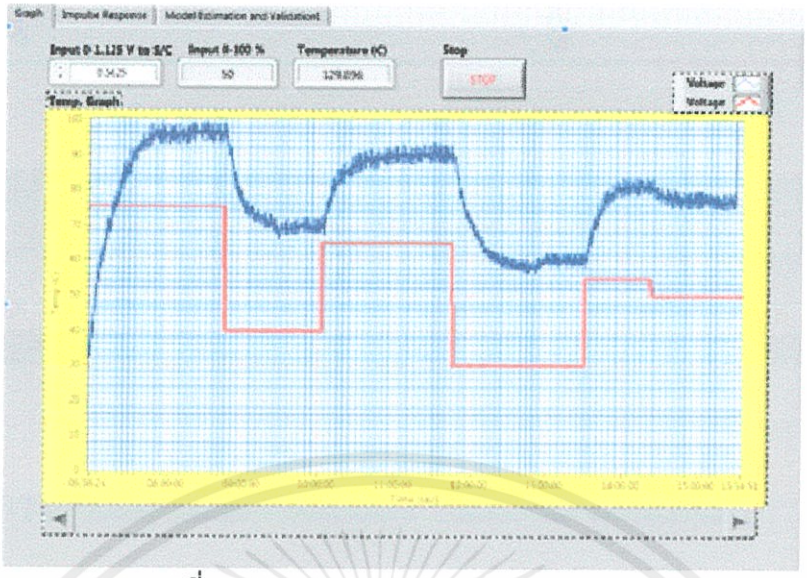


รูปที่ 4.10 แสดงกราฟผลตอบสนองที่ MV = 50%

7. ใส่ค่า MV = 55% ลงในโปรแกรม LabVIEW จะได้กราฟผลตอบสนองดังรูปที่ 4.11

รูปที่ 4.11 แสดงกราฟผลตอบสนองที่ MV = 55%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

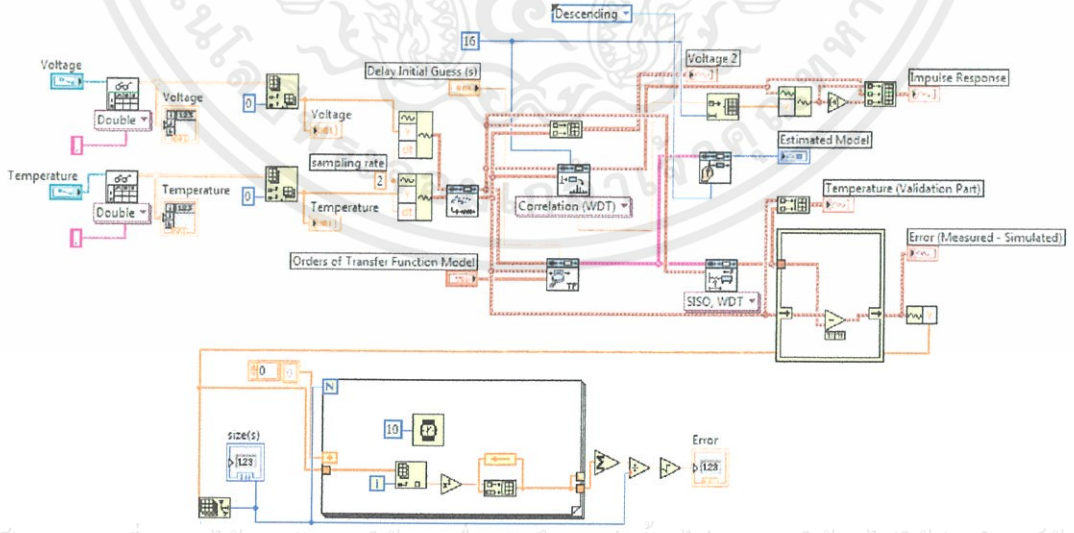


รูปที่ 4.12 แสดงกราฟผลตอบสนองของ MV

จากรูปที่ 4.12 แสดงกราฟผลตอบสนองที่ค่าควบคุมเอาท์พุต (Manipulated variable; MV) ค่าต่างๆโดยในการทดลองนั้นเราจะปรับค่า MV ที่ 75% 40% 65% 30% 50% และ 55% ตามลำดับ เพื่อนำค่าผลตอบสนองที่ได้นั้นไปทำการหาเอกลักษณ์ของระบบต่อไป

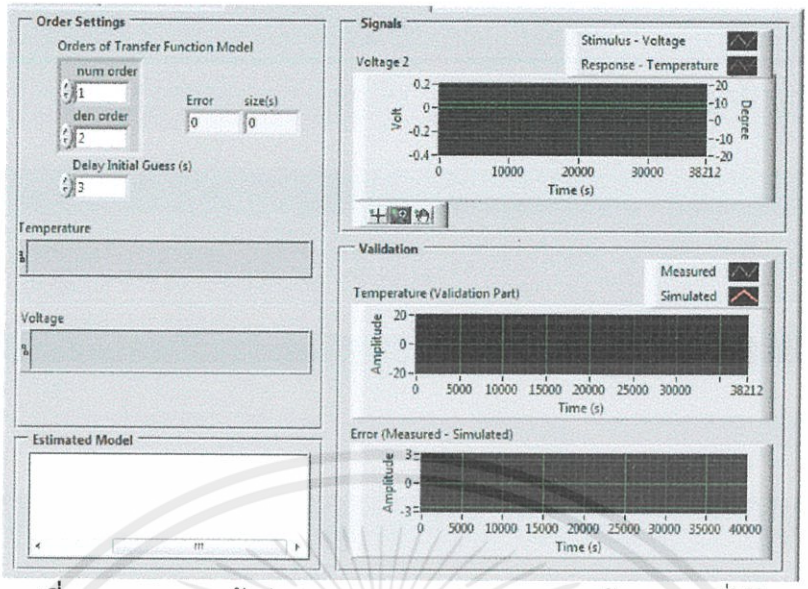
4.4 กระบวนการหาเอกลักษณ์ของระบบ

ในการหาเอกลักษณ์ของกระบวนเตาอบ ซึ่งจะเลือกใช้วิธีผลตอบสนองแบบอิมพัลส์โดยจะนำค่าผลตอบสนองที่ได้จากกระบวนการ มาเข้าสู่กระบวนการหาผลตอบสนองแบบอิมพัลส์ (Impulse Response) โดยใช้โปรแกรม LabVIEW



รูปที่ 4.13 แสดงโปรแกรมที่ใช้ในการหาเอกลักษณ์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

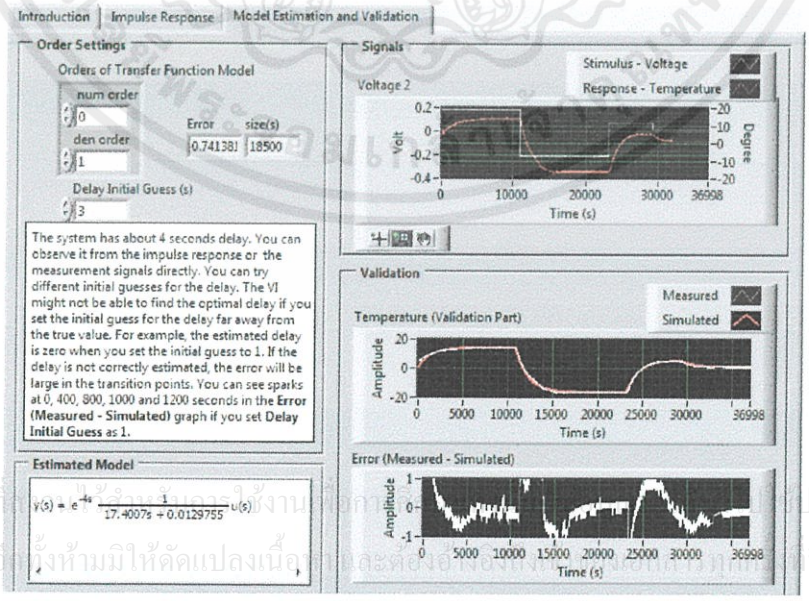


รูปที่ 4.14 แสดงหน้าต่างพาเนล (Front Panel) ของโปรแกรมที่ใช้ในการหาเอกลักษณ์ของระบบ

จากรูปที่ 4.13 แสดงโปรแกรม LabVIEW ที่ใช้ในการหาเอกลักษณ์ของกระบวนการเตาอบขนาดเล็กโดยวิธีการหาผลตอบสนองแบบอิมพัลส์ (Impulse Response) ในการทดลองจะทำการสุ่ม num order (Zero) และ den order (Pole) เพื่อที่จะประมาณของฟังก์ชันถ่ายโอนของระบบ (Estimated Model) โดย num order และ den order ชุดใดที่มีค่าความผิดพลาด (error) ต่ำที่สุด จะแสดงว่า Estimated Model ชุดนั้นๆสอดคล้องกับระบบ

4.5 ผลการทดลอง

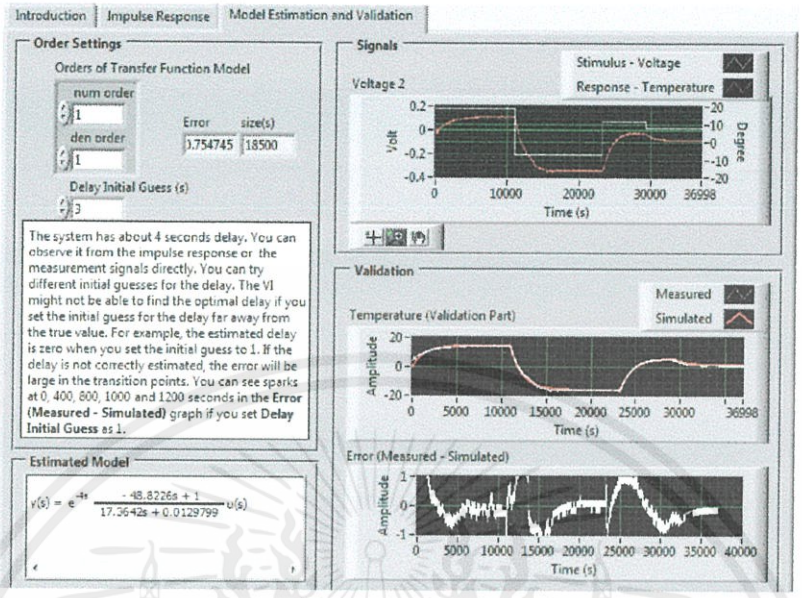
1. จากการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=1



รูปที่ 4.15 แสดงผลของการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=1

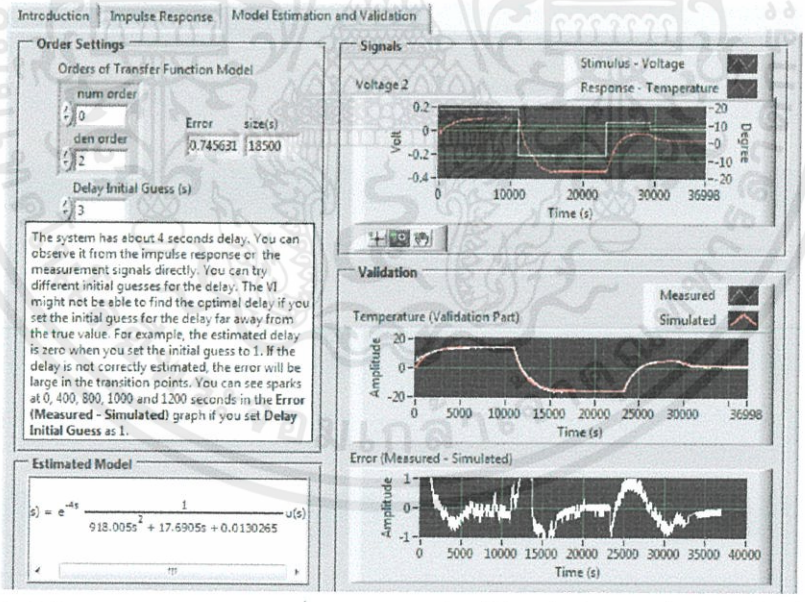
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

2. จากการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=1



รูปที่ 4.16 แสดงผลของการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=1

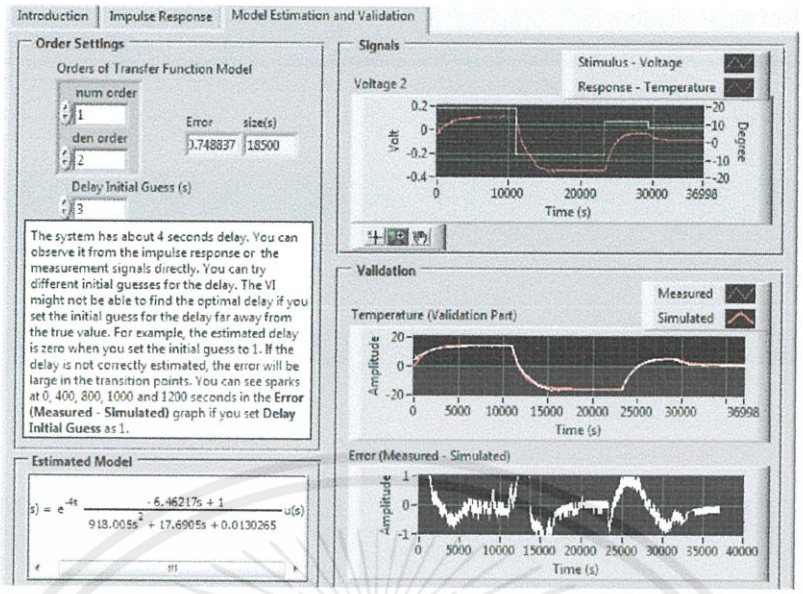
3. จากการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=2



รูปที่ 4.17 แสดงผลของการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=2

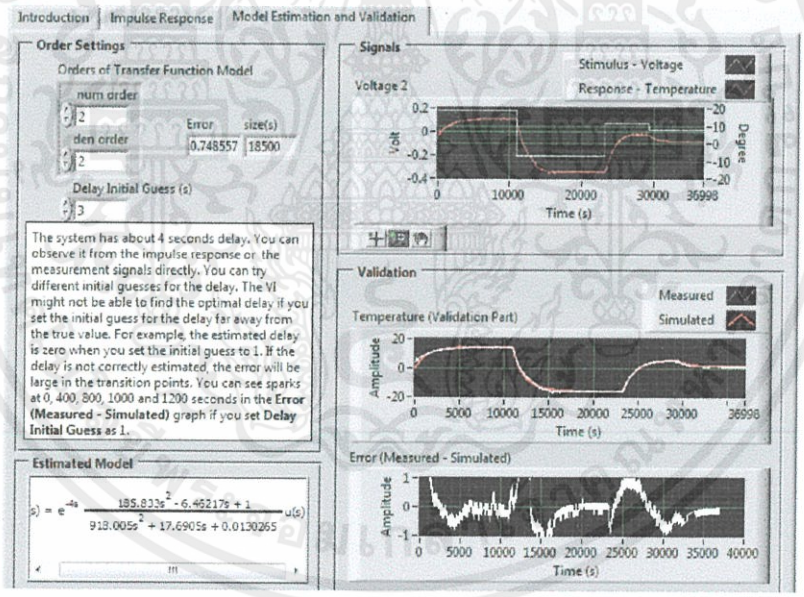
4. จากการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงผลของการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=2

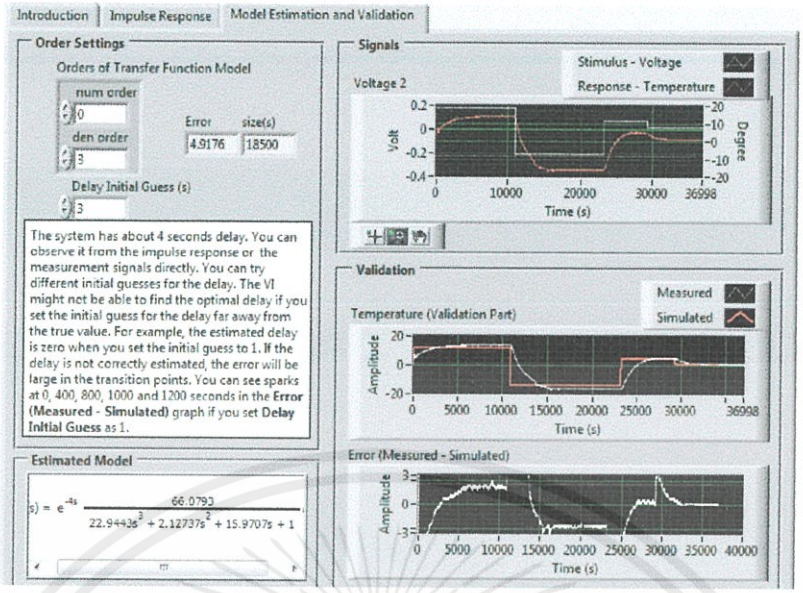
5. จากการทดลองสุ่ม num order (Zero)=2 และ den order (Pole)=2



รูปที่ 4.19 แสดงผลของการทดลองสุ่ม num order (Zero)=2 และ den order (Pole)=2

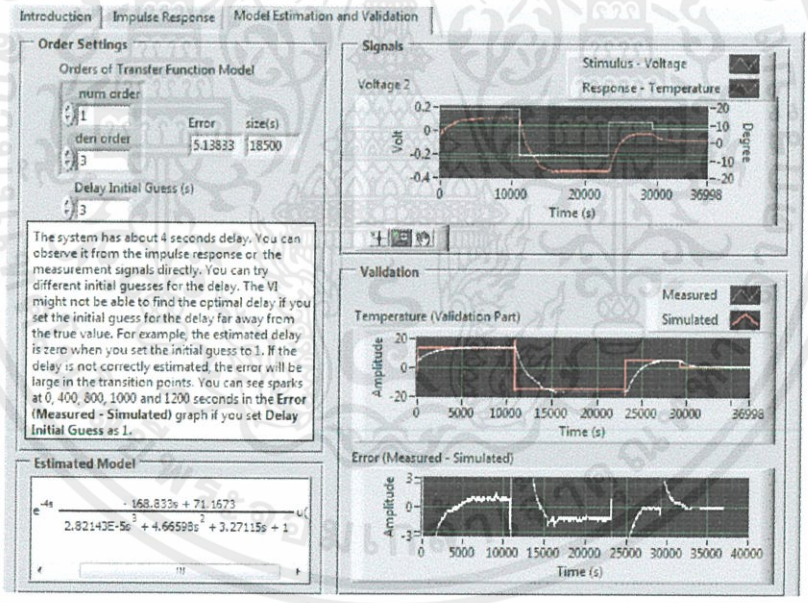
6. จากการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงผลของการทดลองสุ่ม num order (Zero)=0 และ den order (Pole)=3

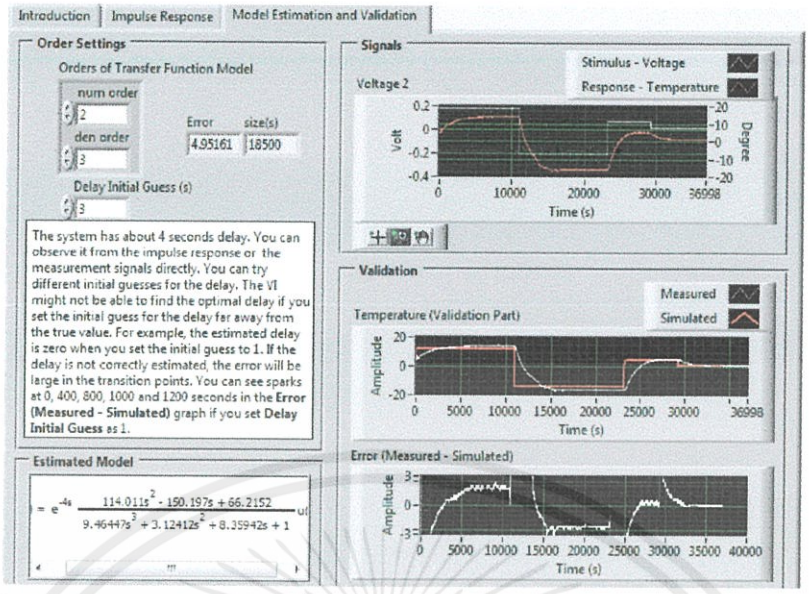
7. จากการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=3



รูปที่ 4.21 แสดงผลของการทดลองสุ่ม num order (Zero)=1 และ den order (Pole)=3

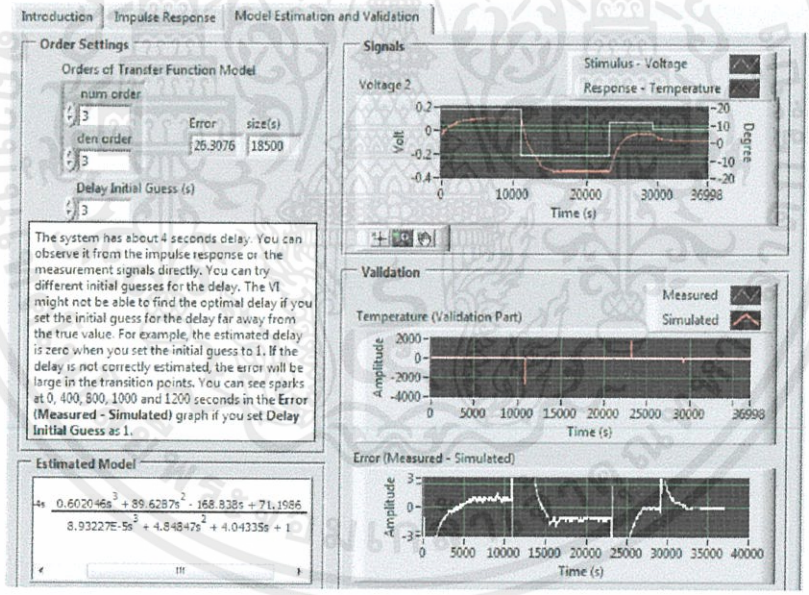
8. จากการทดลองสุ่ม num order (Zero)=2 และ den order (Pole)=3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 แสดงผลของการทดลองสุ่ม num order (Zero)=2 และ den order (Pole)=3

9. จากการทดลองสุ่ม num order (Zero)=3 และ den order (Pole)=3



รูปที่ 4.23 แสดงผลของการทดลองสุ่ม num order (Zero)=3 และ den order (Pole)=3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงค่าของ Estimated Model และแสดงค่าความผิดพลาด

Num order	Den order	Delay Initial Guess	Estimated Model	Error
0	1	2	$y(s) = e^{2.001s}\frac{1}{17.4007s + 0.0129755}u(s)$	0.741381
1	1	2	$y(s) = e^{-2s}\frac{-48.8226s + 1}{17.3642s + 0.0129799}u(s)$	0.754745
0	2	2	$y(s) = e^{-2s}\frac{1}{918.005s^2 + 17.6905s + 0.0130125}u(s)$	0.745631
1	2	2	$y(s) = e^{-2s}\frac{-6.46217s + 1}{918.005s^2 + 17.6905s + 0.0130265}u(s)$	0.748837
2	2	2	$y(s) = e^{-2s}\frac{185.833s^2 - 6.46217s + 1}{165.347s^2 + 17.8844s + 0.0130648}u(s)$	0.748557
0	3	2	$y(s) = e^{-2.001s}\frac{1}{22.9443s^3 + 2.12737s^2 + 15.9707s + 1}u(s)$	4.9176
1	3	2	$y(s) = e^{-2s}\frac{168.833s + 71.1673}{2.8214E - 5s^3 + 4.66598s^2 + 3.27115s + 1}u(s)$	5.13833
2	3	2	$y(s) = e^{-2s}\frac{114.011s^2 + 150.197s + 66.5152}{9.46477s^3 + 3.12412s^2 + 8.35942s + 1}u(s)$	4.95161
3	3	2	$y(s) = e^{-2.001s}\frac{0.602046s^3 + 89.6287s^2 - 168.838s + 71.986}{8.9322E - 5s^3 + 4.48484s^2 + 4.04335s + 1}u(s)$	26.3076

จากตาราง แสดงค่าของการ Estimated Model และค่าความผิดพลาด (Error) จากการประมาณของฟังก์ชันถ่ายโอน จะเห็นได้ว่าในการ สุ่ม num order (Zero) และ den order (Pole) (โดยที่ Zero ต้องน้อยกว่าหรือเท่ากับ Pole) ในแต่ละครั้ง จะได้ค่าความผิดพลาด (Error) ที่แตกต่างกันดังนั้นในการเลือก num order (Zero) และ den order (Pole) ต้องเลือกชุดที่มีความผิดพลาดน้อยที่สุด และจะเห็นได้ว่าค่าความผิดพลาด (error) ที่ num order (Zero)=0 และ den order (Pole)=1 มีค่าน้อยที่สุด ดังนั้นจึงเลือกฟังก์ชันถ่ายโอนของระบบที่มี num order (Zero)=0 และ den order (Pole)=1 และจะได้ฟังก์ชันถ่ายโอนโดยประมาณคือ

$$y(s) = e^{2.001s}\frac{1}{17.4007s + 0.0129755}u(s)$$

4.6 สรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานการวิจัยนี้ทุกครั้งที่มีการนำไปใช้

สำหรับในบทที่ 4 ได้อธิบายถึงกระบวนการที่ใช้ในการทดลองและการจำลองกระบวนการเพื่อหาฟังก์ชันถ่ายโอนของระบบ ซึ่งผลการทดลองแสดงให้เห็นว่าฟังก์ชันถ่ายโอนของระบบถูกเลือกมาจากชุดของการสุ่ม num order (Zero) และ den order (Pole) ที่มีความผิดพลาดน้อยที่สุด

บรรณานุกรม

กิจไพบูลย์ ชีวพันธุศรี. **LabVIEW ซอฟต์แวร์เพื่อการพัฒนาระบบการวัดและควบคุม**. กรุงเทพฯ: วี.พรินท์,2554

วิริยะ กองรัตน์ และคณะ. **ปฏิบัติการวิศวกรรมการวัดคุม** 4. พิมพ์ครั้งที่ 2. กรุงเทพฯ : แผนกตำรา คณะวิศวกรรมศาสตร์ สจล. ,2546

อาจินต์ น่วมสำราญ และทรงชัย วีระทวีมาศ. **ปฏิบัติการวิศวกรรมการวัดคุม 2**. พิมพ์ครั้งที่ 1 กรุงเทพฯ : แผนกตำรา คณะวิศวกรรมศาสตร์.สจล. ,2549

อาจินต์ น่วมสำราญ เอกสารประกอบการบรรยาย Control System

National Instruments. Corporation.**LabVIEW PID and Fuzzy Logic Toolkit User Manual**. Austin, Texas USA ,2009

จตุรพร จันทรโชติ. 2555 "การออกแบบตัวควบคุมอุณหภูมิแบบฟัซซี-พีไอดีสำหรับเตาอบขนาดเล็ก DESIGN OF TEMPERATURE FUZZY-PID CONTROLLER FOR SMALL OVEN" ปริญญานิพนธ์วิศวกรรมศาสตร์ สาขาวิศวกรรม การวัดคุม, สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้