การจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ เพื่อการควบคุมที่รวดเร็ว

FOUNDATION FIELDBUS SEGMENT SCHEDULING FOR FAST LOOP CONTROLS

ปิยพร หารธุจิต
PIYAPORN HARNTUJIT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2556
KMITL-2013-EN-M-257-081

FOUNDATION FIELDBUS SEGMENT SCHEDULING FOR FAST LOOP CONTROLS

PIYAPORN HARNTUJIT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN AUTOMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2013

KMITL-2013-EN-M-257-081

COPYRIGHT 2013

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์เพื่อการควบคุมที่รวดเร็ว
Thesis Title Foundation Fieldbus Segment Scheduling for Fast Loop Controls
นักศึกษา นางสาวปิยพร หารธุจิต
รหัสประจำตัว 54613407
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมอัตโนมัติ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ประสิทธิ์ จุลเสรีวงศ์
หมายเลขวิทยานิพนธ์ KMITL-2013-EN-M-257-081

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.ไสว	พงศ์สวัสดิ์	
ผศ.ดร.ธีรวัฒน์	เทพมณี	
รศ.ดร.จิระศักดิ์	ชาญวุฒิธรรม	
รศ.ดร.อัมพวัน	จุลเสรีวงศ์	
รศ.ประสิทธิ์	จุลเสรีวงศ์	

วัน / เดือน/ ปี ที่สอบ วันจันทร์ที่ 20 พฤษภาคม พ.ศ. 2556 เวลา 14.30-16.00 น.
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 1

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(ศาสตราจารย์ ดร.สุชัชวีร์ สุวรรณสวัสดิ์)
คณบดี คณะวิศวกรรมศาสตร์
วันที่ 20 พฤษภาคม พ.ศ. 2556

หัวข้อวิทยานิพนธ์	การจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์เพื่อการควบคุมที่รวดเร็ว
นักศึกษา	นางสาวปิยพร หารธุจิต
รหัสประจำตัว	54613407
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมอัตโนมัติ
พ.ศ.	2556
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ประสิทธิ์ จุลเสรีวงศ์

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอเทคนิคในการทำให้การควบคุมกระบวนการมีความรวดเร็วมากยิ่งขึ้น โดยอาศัยคุณสมบัติการควบคุมที่ระดับตัวอุปกรณ์หรือ Control in the Field ของเทคโนโลยีฟาวน์เดชันฟิลด์บัส การจัดมาโครไซเคิลของฟิลด์บัสเซกเมนต์ที่ประกอบด้วยวงควบคุมหนึ่งและสองวงควบคุมถูกทดสอบเพื่อเป็นตัวอย่างกรณีศึกษา ประสิทธิภาพในการทำงานของเทคนิคที่นำเสนอนี้ถูกศึกษาจากตารางเวลามาโครไซเคิลที่คำนวณได้ซึ่งถูกสร้างโดยระบบ DeltaV และ CentumVP สิ่งที่ทำให้เทคนิคที่นำเสนอนี้ได้เปรียบการควบคุมโดยการใช้ระบบดีซีเอสในแบบเดิมก็คือ วิธีการที่นำเสนอนี้มีการปรับปรุงการจัดตารางเวลาในการสื่อสารภายในเซกเมนต์ โดยการทำให้มาโครไซเคิลมีช่วงเวลาสำหรับ Scheduled Communications สั้นลง

Thesis	Foundation Fieldbus Segment Scheduling for Fast Loop Controls
Student	Ms. Piyaporn Harntujit
Student ID.	54613407
Degree	Master of Engineering
Program	Automation Engineering
Year	2013
Thesis Advisor	Assoc. Prof. Prasit Julsereewong

Abstract

In this thesis, a technique to obtain fast loop controls using Foundation Fieldbus-based control in the field strategy is presented. Scheduling segment macrocycle for running a single control loop and running two control loops on a segment is examined as illustrative case study. Performances of the proposed technique were studied through calculated macrocycle timing schedules produced by the DeltaV and CentumVP systems. Surpassing the traditional Distributed Control System (DCS) approach with control in the host, the proposed method offers significant improvement in segment schedules with minimizing network scheduled communications.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ. ประสิทธิ์ จุลเสรีวงศ์ ที่ได้ให้ความช่วยเหลือ ให้คำชี้แนะ ช่วยแก้ปัญหาตลอดจนให้ความรู้และถ่ายทอดประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอกราบขอบพระคุณ รศ.ดร.อัมพวัน จุลเสรีวงศ์ ที่คอยให้คำปรึกษา ให้แนวคิด ให้คำแนะนำตลอดจนให้การดูแลและความช่วยเหลือตั้งแต่เริ่มต้นจนวิทยานิพนธ์เล่มนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณคณาจารย์สาขาวิชาวิศวกรรมการวัดและควบคุม (หลักสูตรวิศวกรรมอัตโนมัติ) คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้ และให้ความอนุเคราะห์ในด้านต่าง ๆ

ขอขอบคุณหัวหน้างานและเพื่อนร่วมงานทุกท่าน ที่เข้าใจและให้โอกาส ให้ความช่วยเหลือในหน้าที่การงาน จนทำให้การทำวิทยานิพนธ์ของข้าพเจ้าเป็นไปอย่างราบรื่น

สุดท้ายขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยดูแลห่วงใย เป็นกำลังใจและให้การสนับสนุนข้าพเจ้ามาโดยตลอด รวมทั้งเพื่อน ๆ และญาติพี่น้องทุกคนที่คอยถามไถ่และเป็นกำลังใจให้เสมอมา

ปิยพร หารธุจิต

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....2

1.3 การนำเสนอหลักการใหม่ของวิทยานิพนธ์.....2

1.4 ขอบเขตวิทยานิพนธ์.....2

1.5 รายละเอียดของวิทยานิพนธ์.....3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....4

2.1 กล่าวนำ.....4

2.2 หลักการของตัวควบคุม.....4

2.2.1 กริยาของตัวควบคุม.....5

2.3 การควบคุมกระบวนการ.....9

2.3.1 ตัวควบคุมแบบป้อนกลับ.....9

2.3.2 ตัวควบคุมแบบคาสเคด.....10

2.3.3 ตัวควบคุมแบบป้อนไปข้างหน้า.....12

2.4 ระบบดีซีเอส.....13

2.4.1 หน่วยประมวลผลและส่วนอินพุตหรือเอาต์พุต.....14

2.4.2 หน่วยแสดงผล.....14

2.4.3 หน่วยเชื่อมต่อเครือข่าย.....15

2.5 ฟาวน์เดชันฟิลด์บัส.....15

2.5.1 ฟาวน์เดชันฟิลด์บัส H1.....17

2.5.2 ฟิลด์บัสบล็อก.....17

2.5.2.1 ทรานสดิวเซอร์บล็อก.....18

2.5.2.2 รีซอร์สบล็อก.....19

สารบัญ (ต่อ)

หน้า

2.5.2.3 ฟังก์ชันบล็อก....20
2.5.3 การสร้างวงควบคุมฟิลด์บัส....30
2.5.4 การบริหารจัดการความเสี่ยงของฟาวน์เดชันฟิลด์บัสเซกเมนต์....31
2.5.5 การสื่อสารข้อมูลบนเครือข่าย....33
2.5.5.1 Scheduled Communications....33
2.5.5.2 Unscheduled Communications....34
2.5.5.3 การจัดตารางมาโครไซเคิล....35

บทที่ 3 การจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์....38
3.1 กล่าวนำ....38
3.2 พลานต์โมเดลที่ใช้เป็นกรณีศึกษา....38
3.2.1 พลานต์โมเดล 1 38
3.2.2 พลานต์โมเดล 2....40
3.3 วงควบคุมฟิลด์บัสที่ใช้เป็นกรณีศึกษา....41
3.4 การสร้างฟิลด์บัสเซกเมนต์เพื่อศึกษาการจัดตารางเวลาในการสื่อสาร....42
3.4.1 การสร้างฟิลด์บัสเซกเมนต์โดยระบบ Centum VP 42
3.4.2 การสร้างฟิลด์บัสเซกเมนต์โดยระบบ DeltaV....43
3.5 ผลและการวิเคราะห์ตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ที่ได้จากกรณีศึกษา....44
3.5.1 การทดสอบกรณีศึกษาโดยใช้ระบบ Centum VP....45
3.5.1.1 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ภายในตัวอุปกรณ์....45
3.5.1.2 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ที่ดีซีเอส....48
3.5.2 การทดสอบกรณีศึกษาโดยใช้ระบบ DeltaV....51
3.5.2.1 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ภายในตัวอุปกรณ์....51
3.5.2.2 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ที่ดีซีเอส....54
3.6 การทดลองและผลการทดลองจากการทำงานของพลานต์โมเดลที่ใช้เป็นกรณีศึกษา....58

สารบัญ (ต่อ)

หน้า

บทที่ 4 สรุปผลการวิจัยและข้อเสนอแนะ.....65
4.1 สรุปผลการวิจัย.....65
4.2 ข้อเสนอแนะในการทำวิจัยต่อ.....65

เอกสารอ้างอิง.....67

ภาคผนวก.....68

ประวัติผู้เขียน.....77

สารบัญตาราง

ตารางที่ หน้า

2.1 ตัวอย่างข้อมูลบางส่วนที่อยู่ในทรานสดิวเซอร์บล็อก....18

2.2 ตัวอย่างข้อมูลบางส่วนที่อยู่ภายในรีซอร์สบล็อก....19

3.1 รายละเอียดของอุปกรณ์ฟาวน์เดชันฟิลด์บัสที่อยู่ภายในพลานต์โมเดล 1....39

3.2 รายละเอียดของอุปกรณ์ฟาวน์เดชันฟิลด์บัสที่อยู่ภายในพลานต์โมเดล 2....40

3.3 กรณีศึกษาที่ใช้ในวิทยานิพนธ์....44

3.4 สรุปค่าเวลาของมาโครไซเคิลซึ่งคำนวณโดยระบบ DeltaV....56

3.5 สรุปค่าเวลาของมาโครไซเคิลซึ่งคำนวณโดยระบบ Centum VP....56

3.6 ค่าประมาณการใช้แบนด์วิธสำหรับการสื่อสารแบบ Scheduled Communications 57

3.7 ฟังก์ชันบล็อกที่ใช้ในการกำหนดตั้งค่าสำหรับวงควบคุมในกรณี Control in the Field ใน Control Studio....59

3.8 ฟังก์ชันบล็อกที่ใช้ในการกำหนดตั้งค่าสำหรับวงควบคุมในกรณีที่กำหนดให้ตัวควบคุมอยู่ในดีซีเอส ใน Control Studio....60

3.9 ผลการทดลองการควบคุมพลานต์โมเดลในแต่ละกรณีศึกษา (CIF)....61

3.10 ผลการทดลองการควบคุมพลานต์โมเดลในแต่ละกรณีศึกษา (DCS)....61

3.11 Network Load ของการควบคุมในแต่ละกรณีที่ได้จากโปรแกรม NI-FBUS Monitor....62

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างพื้นฐานของตัวควบคุม PID 5
2.2 คุณสมบัติของกริยาควบคุมแบบ P 6
2.3 ผลตอบสนองของกริยาควบคุมแบบ I 7
2.4 ตัวอย่างผลตอบสนองของกริยาควบคุมแบบ D 8
2.5 ตัวอย่างผลตอบสนองของกริยาควบคุมแบบ PID (Direct Action).....9
2.6 บล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับ.....10
2.7 ตัวอย่างผลิตภัณฑ์ตัวควบคุมแบบคาสเคด.....10
2.8 ตัวควบคุมอุณหภูมิที่เป็นการควบคุมแบบคาสเคด (บริษัท Yokogawa).....11
2.9 ระบบควบคุมการแลกเปลี่ยนความร้อนโดยใช้การควบคุมแบบป้อนไปข้างหน้า.....12
2.10 ระบบควบคุมการแลกเปลี่ยนความร้อนโดยใช้การควบคุมแบบป้อนไปข้างหน้า ร่วมกับการชดเชยป้อนกลับ.....13
2.11 โครงสร้างพื้นฐานของระบบดีซีเอส.....14
2.12 โครงสร้างของระบบฟาวน์เดชันฟิลด์บัส.....16
2.13 ระบบเครื่องมือวัดทั้งสองแบบ.....16
2.14 โปรโตคอลของฟาวน์เดชันฟิลด์บัสเทียบกับโมเดลของ OSI 17
2.15 ชนิดของบล็อกที่ใช้ใน User Application Layer 17
2.16 โครงสร้างภายในของฟังก์ชันบล็อก.....20
2.17 ตัวอย่างการควบคุมที่ใช้ฟังก์ชันบล็อกที่อยู่ภายในตัวอุปกรณ์.....21
2.18 ฟังก์ชันบล็อก AI 21
2.19 โครงสร้างภายในฟังก์ชันบล็อก AI 22
2.20 ฟังก์ชันบล็อก AO 23
2.21 โครงสร้างภายในฟังก์ชันบล็อก AO 24
2.22 ฟังก์ชันบล็อก PID.....25
2.23 โครงสร้างภายในฟังก์ชันบล็อก PID.....26
2.24 หลักการของการเลือกและกำหนดขอบเขตของค่า SP.....27
2.25 ตัวอย่างแหล่งที่มาของ DD File และ CFF.....29
2.26 การสร้างวงควบคุมฟิลด์บัสสำหรับการควบคุมแบบ PID.....30
2.27 การสร้างวงควบคุมฟิลด์บัสสำหรับการควบคุมแบบ PID.....30
2.28 โครงข่ายการสื่อสารพื้นฐานของฟิลด์บัส.....33
2.29 การรับ-ส่งข้อมูลแบบ Scheduled Communications.....34

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.30 การรับ-ส่งข้อมูลแบบ Unscheduled Communications....35

2.31 ตัวอย่างการจัดมาโครไซเคิล....35

2.32 มาโครไซเคิลของการควบคุมแบบ Control in the Field ซึ่งประกอบด้วยวงควบคุมแบบ PID หนึ่งวง....37

3.1 พลานต์โมเดล 1....38

3.2 P&ID ของพลานต์โมเดล 1 ในกรณี Control in the Field....39

3.3 P&ID ของพลานต์โมเดล 1 ในกรณีควบคุมด้วยระบบดีซีเอส....39

3.4 พลานต์โมเดล 2....40

3.5 P&ID ของพลานต์โมเดล 2 ในกรณี Control in the Field....41

3.6 P&ID ของพลานต์โมเดล 2 ในกรณีควบคุมด้วยระบบดีซีเอส....41

3.7 วงควบคุมฟิลด์บัสสำหรับการควบคุมที่ระดับตัวอุปกรณ์....42

3.8 วงควบคุมฟิลด์บัสสำหรับการควบคุมที่ดีซีเอส....42

3.9 ตัวอย่างการสร้างฟิลด์บัสเซกเมนต์โดยระบบ Centum VP....43

3.10 ตัวอย่างการสร้างฟิลด์บัสเซกเมนต์โดยระบบ DeltaV....44

3.11 มาโครไซเคิลของกรณีที่ 1_CIF ที่ได้จากระบบ Centum VP....45

3.12 มาโครไซเคิลของกรณีที่ 2_CIF ที่ได้จากระบบ Centum VP....46

3.13 มาโครไซเคิลของกรณีที่ 3_CIF ที่ได้จากระบบ Centum VP....46

3.14 มาโครไซเคิลของกรณีที่ 4_CIF ที่ได้จากระบบ Centum VP....46

3.15 มาโครไซเคิลของกรณีที่ 5_CIF ที่ได้จากระบบ Centum VP....47

3.16 มาโครไซเคิลของกรณีที่ 6_CIF ที่ได้จากระบบ Centum VP....47

3.17 มาโครไซเคิลของกรณีที่ 7_CIF ที่ได้จากระบบ Centum VP....47

3.18 มาโครไซเคิลของกรณีที่ 8_CIF ที่ได้จากระบบ Centum VP....48

3.19 มาโครไซเคิลของกรณีที่ 1_DCS ที่ได้จากระบบ Centum VP....48

3.20 มาโครไซเคิลของกรณีที่ 2_DCS ที่ได้จากระบบ Centum VP....49

3.21 มาโครไซเคิลของกรณีที่ 3_DCS ที่ได้จากระบบ Centum VP....49

3.22 มาโครไซเคิลของกรณีที่ 4_DCS ที่ได้จากระบบ Centum VP....49

3.23 มาโครไซเคิลของกรณีที่ 5_DCS ที่ได้จากระบบ Centum VP....50

3.24 มาโครไซเคิลของกรณีที่ 6_DCS ที่ได้จากระบบ Centum VP....50

3.25 มาโครไซเคิลของกรณีที่ 7_DCS ที่ได้จากระบบ Centum VP....50

3.26 มาโครไซเคิลของกรณีที่ 8_DCS ที่ได้จากระบบ Centum VP....51

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.27 มาโครไซเคิลของกรณีที่ 1_CIF ที่ได้จากระบบ DeltaV....51

3.28 มาโครไซเคิลของกรณีที่ 2_CIF ที่ได้จากระบบ DeltaV....52

3.29 มาโครไซเคิลของกรณีที่ 3_CIF ที่ได้จากระบบ DeltaV....52

3.30 มาโครไซเคิลของกรณีที่ 4_CIF ที่ได้จากระบบ DeltaV....52

3.31 มาโครไซเคิลของกรณีที่ 5_CIF ที่ได้จากระบบ DeltaV....52

3.32 มาโครไซเคิลของกรณีที่ 6_CIF ที่ได้จากระบบ DeltaV....53

3.33 มาโครไซเคิลของกรณีที่ 7_CIF ที่ได้จากระบบ DeltaV....53

3.34 มาโครไซเคิลของกรณีที่ 8_CIF ที่ได้จากระบบ DeltaV....53

3.35 มาโครไซเคิลของกรณีที่ 1_DCS ที่ได้จากระบบ DeltaV....54

3.36 มาโครไซเคิลของกรณีที่ 2_DCS ที่ได้จากระบบ DeltaV....54

3.37 มาโครไซเคิลของกรณีที่ 3_DCS ที่ได้จากระบบ DeltaV....54

3.38 มาโครไซเคิลของกรณีที่ 4_DCS ที่ได้จากระบบ DeltaV....54

3.39 มาโครไซเคิลของกรณีที่ 5_DCS ที่ได้จากระบบ DeltaV....55

3.40 มาโครไซเคิลของกรณีที่ 6_DCS ที่ได้จากระบบ DeltaV....55

3.41 มาโครไซเคิลของกรณีที่ 7_DCS ที่ได้จากระบบ DeltaV....55

3.42 มาโครไซเคิลของกรณีที่ 8_DCS ที่ได้จากระบบ DeltaV....55

3.43 การใช้โปรแกรม NI-FBUS Monitor ในการทดลอง....58

3.44 ตัวอย่างการกำหนดตั้งค่าสำหรับวงควบคุมใน Control Studio....59

3.45 การทำงานของฟังก์ชันบล็อกในโหมด Online ของการควบคุมในกรณีที่ 7....64

3.46 แผนภาพกระบวนการทำงานของพลานต์โมเดล 2 ในการทดลองจริง....64

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีฟาวน์เดชันฟิลด์บัส (Foundation Fieldbus) ได้เข้ามามีบทบาทในการควบคุมกระบวนการผลิตแบบอัตโนมัติมากยิ่งขึ้น อันเนื่องมาจากความได้เปรียบทางด้านโครงสร้างทางกายภาพของระบบและความสามารถในการติดต่อสื่อสารข้อมูลที่มีความสะดวกและรวดเร็วกว่าการควบคุมแบบดั้งเดิม ดังจะเห็นได้จากการที่มีงานวิจัยต่าง ๆ ซึ่งมุ่งเน้นในการนำเอาศักยภาพทางด้านความเร็วในการสื่อสารข้อมูลของอุปกรณ์ฟิลด์บัสมาพัฒนาอย่างต่อเนื่อง R. William และคณะ [1] ได้ทำการศึกษาเกี่ยวกับวิธีการจัดมาโครไซเคิล (Macrocycle) ของฟาวน์เดชันฟิลด์บัสเพื่อเพิ่มประสิทธิภาพในการทำงาน โดยได้นำเสนอวิธีการจัดเรียงลำดับการดำเนินการของแต่ละฟังก์ชันบล็อก (Function Block) ตามลำดับความสำคัญ และวิธีการจัดวางบล็อกดำเนินการ (Execution Block) ให้ทำงานขนานกันไปในแต่ละอุปกรณ์ที่อยู่ภายในเซกเมนต์ (Segment) เพื่อลดช่วงเวลาว่างระหว่างการดำเนินการฟังก์ชันบล็อก (Block Execution) กับการรับ-ส่งข้อมูลระหว่างฟังก์ชัน (Block Publication) ซึ่งจะส่งผลให้มาโครไซเคิลมีช่วงเวลาสำหรับ Scheduled Communications สั้นลงและเป็นการเพิ่มช่วงเวลาสำหรับ Unscheduled Communications อีกด้วย นอกจากนี้ A. Julsereewong และคณะ [2] ได้ทำการทดลองการควบคุมระดับน้ำในถังโดยใช้การควบคุมแบบคาสเคด (Cascade) โดยกำหนดตำแหน่งของฟังก์ชันบล็อก PID ไปยังอุปกรณ์ต่าง ๆ ที่อยู่ภายในเซกเมนต์ เพื่อศึกษาคุณสมบัติในการกระจายฟังก์ชันการควบคุมไปยังตัวอุปกรณ์ของฟาวน์เดชันฟิลด์บัสตามแนวคิดของ Control in the Field ซึ่งเป็นจุดเด่นของเทคโนโลยีฟาวน์เดชันฟิลด์บัส จากการทดลองพบว่าการกำหนดให้ฟังก์ชันบล็อก PID ของตัวควบคุมหลัก (Master Controller) อยู่ภายในเครื่องวัดระดับ และฟังก์ชันบล็อก PID ของตัวควบคุมรอง (Slave Controller) อยู่ภายในวาล์วควบคุม จะให้ค่ามาโครไซเคิลสั้นที่สุดและมีการใช้แบนด์วิธ (Bandwidth) ของเครือข่ายน้อยที่สุด อย่างไรก็ตามในบทความวิจัย [2] นั้น เป็นการศึกษามาโครไซเคิลที่ได้จากการควบคุมกระบวนการโดยกำหนดให้ตัวควบคุมอยู่ภายในอุปกรณ์ฟิลด์บัสและเป็นการควบคุมสำหรับฟิลด์บัสเซกเมนต์ที่ประกอบไปด้วยวงควบคุม (Control Loop) หนึ่งวงควบคุมเท่านั้น

จากการติดตามผลงานวิจัยที่เกี่ยวข้องกับวิธีการจัดมาโครไซเคิลของฟิลด์บัสเซกเมนต์ที่ใช้เทคโนโลยีฟาวน์เดชันฟิลด์บัส พบว่ายังไม่มีการศึกษาถึงการจัดมาโครไซเคิลของฟิลด์บัสเซกเมนต์ที่ประกอบด้วยวงควบคุมสองวง และความสัมพันธ์ระหว่างตำแหน่งของตัวควบคุมกับเวลาที่ใช้ในการรับ-ส่งข้อมูลของ Scheduled Communications ซึ่งจัดเป็นภาระในการสื่อสารของเครือข่ายหรือ Network Load ซึ่งตามคำแนะนำของ Foundation Fieldbus End Users Council Australia Inc. [3] มาโครไซเคิลของแต่ละเซกเมนต์ควรจะมีช่วงเวลาสำหรับ Unscheduled Communications อย่างน้อย 60% เพื่อประสิทธิภาพในการควบคุมที่ดี

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้เป็นการนำเสนอแนวคิดในการนำคุณสมบัติการควบคุมที่ระดับตัวอุปกรณ์ หรือ Control in the Field ซึ่งเป็นคุณลักษณะพิเศษของเทคโนโลยีฟาวน์เดชันฟิลด์บัส และการจัดมาโครไซเคิล มาใช้ในการควบคุมกระบวนการโดยผ่านทางกรณีศึกษาซึ่งเป็นรูปแบบการควบคุมกระบวนการที่ใช้กันอย่างแพร่หลายในปัจจุบัน

วัตถุประสงค์ของการทำวิจัยคือ เพื่อศึกษาการจัดตารางเวลาในการสื่อสารของฟิลด์บัสเซกเมนต์ที่ใช้เทคโนโลยีฟาวน์เดชันฟิลด์บัสให้มีช่วงเวลาสำหรับ Scheduled Communications น้อยที่สุดเพื่อการควบคุมกระบวนการที่รวดเร็ว ในวิทยานิพนธ์ฉบับนี้นำเสนอการเปรียบเทียบมาโครไซเคิลที่ได้จากการควบคุมกระบวนการตามแนวคิด “Control in the Field” และการควบคุมกระบวนการโดยระบบดีซีเอส (DCS) ที่ใช้กันโดยทั่วไป

1.3 การนำเสนอหลักการใหม่ของวิทยานิพนธ์

ในวิทยานิพนธ์นี้ได้นำเสนอการจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ที่ประกอบด้วยหนึ่งวงควบคุมและสองวงควบคุม เพื่อให้การรับ-ส่งข้อมูลเป็นไปอย่างมีประสิทธิภาพเหมาะสำหรับการควบคุมกระบวนการที่ต้องการความรวดเร็ว โดยใช้การควบคุมกระบวนการตามแนวคิด Control in the Field เปรียบเทียบกับการควบคุมกระบวนการแบบดั้งเดิมที่ใช้ระบบดีซีเอส โดยมีการควบคุมแบบ PID และแบบคาสเคดเป็นกรณีศึกษา

1.4 ขอบเขตวิทยานิพนธ์

ในการจัดตารางเวลาในการสื่อสารของฟิลด์บัสเซกเมนต์ที่นำเสนอในวิทยานิพนธ์นี้ ได้ใช้พลานต์โมเดล (Plant Model) ที่ใช้เทคโนโลยีฟาวน์เดชันฟิลด์บัส จำนวน 2 ชุด ของสาขาวิชาวิศวกรรมการวัดและควบคุม (หลักสูตรวิศวกรรมอัตโนมัติ) คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และใช้ระบบ Centum VP และ DeltaV ซึ่งเป็นดีซีเอสของบริษัท Yokogawa และ Emerson ตามลำดับในการกำหนดวงควบคุมและสร้างตารางเวลาในการสื่อสารหรือที่เรียกว่ามาโครไซเคิล พร้อมทั้งได้ทำการทดลองการควบคุมพลานต์โมเดลเพื่อตรวจสอบ Network Load ของ Scheduled Communications ที่เกิดขึ้นจากการทำงานจริง โดยใช้ NI-FBUS Monitor ของบริษัท National Instruments เป็นตัวตรวจสอบการรับ-ส่งข้อมูล เพื่อเป็นการยืนยันผลการศึกษามาโครไซเคิลที่ได้จากการกำหนดตั้งค่าระบบ (System Configuration) แบบ Offline โดยระบบ CentumVP และ DeltaV

1.5 รายละเอียดของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 4 บทด้วยกัน คือ

บทที่ 1 บทนำ เป็นการกล่าวถึง ความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา การนำเสนอหลักการใหม่ของวิทยานิพนธ์ ขอบเขตวิทยานิพนธ์และรายละเอียดของวิทยานิพนธ์

บทที่ 2 เป็นการกล่าวถึง หลักการของตัวควบคุม การควบคุมกระบวนการ ระบบดีซีเอส และฟาวน์เดชันฟิลด์บัส

บทที่ 3 เป็นการกล่าวถึง พลานต์โมเดลและวงควบคุมฟิลด์บัสที่ใช้เป็นกรณีศึกษา การสร้างฟิลด์บัสเซกเมนต์เพื่อศึกษาการจัดตารางเวลาในการสื่อสาร ผลและการวิเคราะห์ตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ที่ได้จากกรณีศึกษาและการทดลองและผลการทดลองจากการทำงานของพลานต์โมเดลที่ใช้เป็นกรณีศึกษา

บทที่ 4 เป็นบทสุดท้ายซึ่งจะกล่าวถึงสรุปผลการวิจัยและข้อเสนอแนะ

ในส่วนสุดท้ายของวิทยานิพนธ์เป็นส่วนของภาคผนวก ซึ่งประกอบไปด้วยบทความวิจัยที่ได้รับการตีพิมพ์ ดังนี้

ภาคผนวก ก บทความวิจัยที่ได้รับการตีพิมพ์

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 กล่าวนำ

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีต่าง ๆ ตลอดจนงานวิจัยที่เกี่ยวข้องที่นำมาใช้เป็นแนวทางในการออกแบบการทดลองของวิทยานิพนธ์ฉบับนี้ ซึ่งประกอบด้วย หลักการของตัวควบคุมและกริยาควบคุมต่าง ๆ ที่ใช้ในระบบควบคุมทั่วไป ตลอดจนโครงสร้างของระบบ DCS ที่ถือเป็นหัวใจหลักในการควบคุมกระบวนการผลิตของโรงงานอุตสาหกรรมจำนวนมากในปัจจุบัน นอกจากนี้ยังกล่าวถึงหลักการต่าง ๆ ของฟาวน์เดชันฟิลด์บัสเทคโนโลยีที่อาจถือได้ว่าเป็นจุดประสงค์หลักของการศึกษานี้ เช่น วิธีการสื่อสารข้อมูลในรูปแบบต่าง ๆ การจัดตารางมาโครไซเคิล ตลอดจนคำแนะนำในการจัดอุปกรณ์ภายในเซกเมนต์เป็นต้น

2.2 หลักการของตัวควบคุม [4]

ตัวควบคุมในปัจจุบันสามารถแบ่งตามโครงสร้างการทำงานได้เป็น 3 ชนิดได้แก่ ตัวควบคุมแบบนิวเมติกส์ ทำงานโดยใช้สัญญาณลม ตัวควบคุมแบบอิเล็กทรอนิกส์ ทำงานโดยใช้วงจรอิเล็กทรอนิคส์เชิงเส้น (Linear Circuits) และตัวควบคุมแบบดิจิตอล ทำงานโดยการใช้ไมโครโปรเซสเซอร์ (Microprocessor) ในการสร้างสัญญาณควบคุม โดยตัวควบคุมที่ถูกนำมาใช้กันอย่างแพร่หลายที่สุดในวงการอุตสาหกรรมการผลิตก็คือตัวควบคุมแบบ PID

โดยส่วนใหญ่การควบคุมกระบวนการในอุตสาหกรรมการผลิตจะแบ่งออกเป็นสองลักษณะคือ การควบคุมแบบป้อนกลับ (Feedback Control) และการควบคุมแบบเรียงลำดับ (Sequence Control) ซึ่งในที่นี้จะกล่าวถึงระบบการควบคุมแบบป้อนกลับที่ถูกควบคุมด้วยตัวควบคุมแบบ PID ซึ่งมีฟังก์ชันถ่ายโอนคือ

$$G_c(S) = K_c\left(1+\frac{1}{T_i S}+T_d S\right) \qquad (2.1)$$

โดยที่ K_c คือ อัตราขยายของตัวควบคุม
T_i คือ ค่าอินทิกรัล (Integral) หรือ เวลาในการรีเซต (Reset Time)
T_d คือ ค่าอนุพันธ์ (Derivative) หรือ Rate Time

ค่าพารามิเตอร์ของตัวควบคุม PID ทั้ง 3 ค่านี้ จะใช้สำหรับปรับให้กับตัวควบคุมเพื่อควบคุมกระบวนการให้มีผลตอบสนองเป็นไปตามต้องการ ดังจะเห็นได้ว่าสัญญาณควบคุมหรือตัวแปรปรับกระบวนการ (Manipulated Variable : MV) ที่ได้จากตัวควบคุม PID นั้น จะถูกกำหนดโดยความสัมพันธ์ระหว่างสัญญาณขาเข้าหรือตัวแปรกระบวนการ (Process Variable : PV) กับสัญญาณอ้างอิงหรือค่าเป้าหมาย (Setpoint : SP) ซึ่งความสัมพันธ์ดังกล่าวนี้จะขึ้นอยู่กับกฎเกณฑ์ของการควบคุมที่ได้ถูกปรับแต่งไว้ล่วงหน้า ซึ่งก็คือการปรับแต่งค่าพารามิเตอร์ K_C, T_i และ T_d ของตัวควบคุม PID นั่นเอง และจากความสัมพันธ์ดังกล่าวทำให้สามารถหาสัญญาณควบคุมของตัวควบคุม PID ได้จากความสัมพันธ์คือ

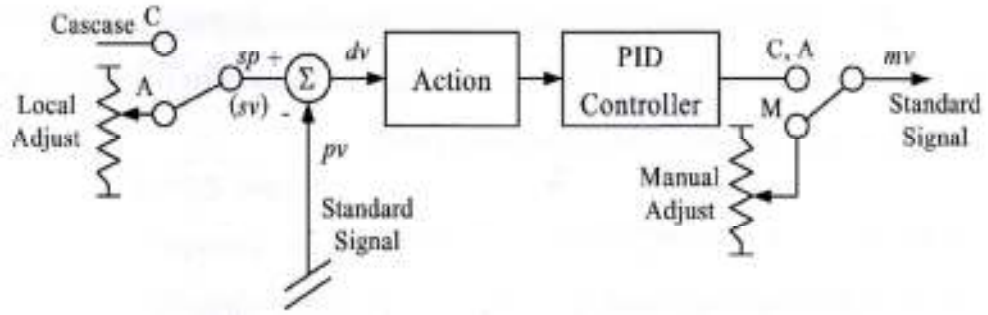
$$mv = \frac{100}{PB}\left[(sp - pv) + \frac{1}{T_i}\int_0^t (sp - pv)dt + T_d\frac{d(sp - pv)}{dt}\right] + b \qquad (2.2)$$

หรือ

$$mv = K_c\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right] + b \qquad (2.3)$$

โครงสร้างพื้นฐานของตัวควบคุม PID ที่ใช้ในงานอุตสาหกรรมแสดงดังรูปที่ 2.1 โดยที่

pv	คือ	ค่าสัญญาณที่อ่านได้จากเครื่องวัด
sp	คือ	สัญญาณอ้างอิงหรือค่าเป้าหมาย
dv	คือ	ค่าเบี่ยงเบน (Deviation) แต่ในทางทฤษฎีจะใช้ e หรือค่าความผิดพลาด (Error)
mv	คือ	ค่าสัญญาณควบคุม
C	คือ	โหมดคาสเคด (Cascade) หรือ Remote Set Mode
A	คือ	โหมดอัตโนมัติ (Automatic Mode)
M	คือ	โหมดการควบคุมด้วยมือ (Manual Mode)
Action	ได้แก่	การกระทำย้อนกลับ (Reverse Action) และ การกระทำโดยตรง (Direct Action)



รูปที่ 2.1 โครงสร้างพื้นฐานของตัวควบคุม PID

2.2.1 กริยาของตัวควบคุม

การออกแบบระบบควบคุมนั้นจะต้องพิจารณาถึงคุณลักษณะของกระบวนการที่ต้องการควบคุมเสียก่อน ซึ่งบางครั้งอาจจะพิจารณาในรูปแบบของแบบจำลองทางคณิตศาสตร์หรือฟังก์ชันถ่ายโอน (Transfer Function) ทั้งนี้ก็เพื่อที่จะได้เลือกรูปแบบของการควบคุมให้เหมาะสมเพื่อให้การควบคุมมีเสถียรภาพและเกิดประสิทธิภาพสูงสุด โดยสัญญาณที่ใช้ในการควบคุมนั้นจะเป็นไปตามลักษณะของการสร้างสัญญาณที่เรียกว่า กริยาการควบคุม (Control Action) นั่นเอง

1. การควบคุมแบบสัดส่วน

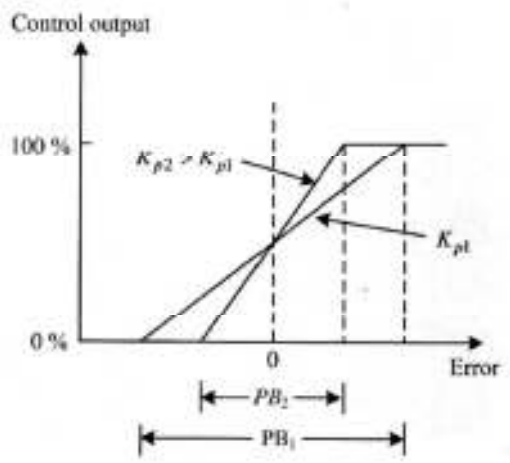
กริยาควบคุมแบบสัดส่วน (Proportional Control : P) นั้น ค่าเอาต์พุตของตัวควบคุมจะแปรผันตรงกับค่าความผิดพลาด e(t) กล่าวคือ ถ้าค่าความผิดพลาดมีค่ามากขึ้น ค่าเอาต์พุตของตัวควบคุมก็จะมีค่ามากขึ้น และถ้าค่าความผิดพลาดมีค่าน้อยลง ค่าเอาต์พุตของตัวควบคุมก็จะมีค่าน้อยลงตามไปด้วย กริยาควบคุมแบบ P สามารถเขียนสมการได้ดังนี้

$$m_p(t) = K_p e(t) + \bar{m} \tag{2.4}$$

เมื่อ $m_p(t)$ คือ ค่าเอาต์พุตของตัวควบคุมแบบ P
K_p คือ อัตราขยายของตัวควบคุมแบบ P
$\bar{m}$ คือ ค่าเอาต์พุตของตัวควบคุมที่มีค่าความผิดพลาดเท่ากับศูนย์

ตัวควบคุมแบบ P บางตัวอาจจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า K_p ซึ่ง PB คือ ช่วงของค่าความผิดพลาดระหว่างที่เอาต์พุตของตัวควบคุมมีค่า 0 - 100% ดังนี้

$$PB = \frac{100\%}{K_p} \tag{2.5}$$



รูปที่ 2.2 คุณสมบัติของกริยาควบคุมแบบ P

รูปที่ 2.2 แสดงคุณสมบัติของกริยาควบคุมแบบ P ซึ่งการควบคุมแบบนี้จะเหมาะกับกระบวนการที่มี Time Lag ไม่สูงมาก แต่การควบคุมแบบนี้มีจุดอ่อนก็คือการเกิดออฟเซต (Offset)

2. การควบคุมแบบอินทิกรัล

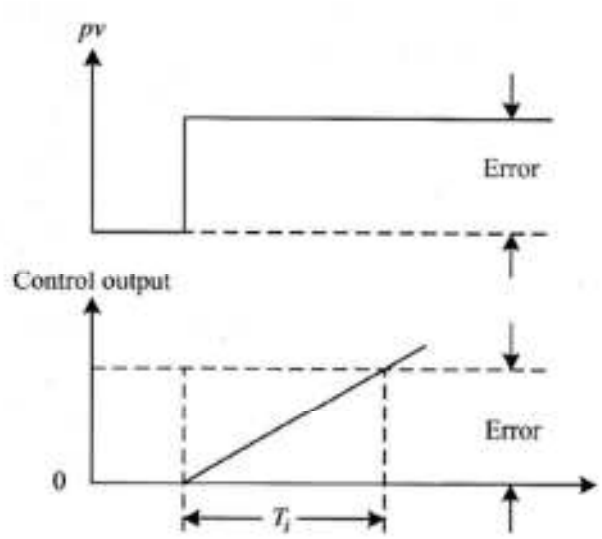
กริยาควบคุมแบบอินทิกรัล (Integral Control : I) หรือเรียกอีกอย่างหนึ่งว่าการควบคุมแบบรีเซต (Reset Control) ค่าเอาต์พุตของตัวควบคุมหาได้จากพื้นที่ทั้งหมดภายใต้กราฟของ

ค่าความผิดพลาดต่อเวลาคูณกับค่าคงที่ที่เรียกว่าอัตราขยายของตัวควบคุมแบบอินทิกรัล (Integral Gain) ดังนี้

$$m_I(t) = K_I \int_0^t e(t)dt + \bar{m}_I(0) \qquad (2.6)$$

เมื่อ $m_I(t)$ คือ ค่าเอาต์พุตของตัวควบคุมแบบ I
K_I คือ อัตราขยายของตัวควบคุมแบบ I
$\int_0^t e(t)dt$ คือ พื้นที่ทั้งหมดของค่าความผิดพลาด
$\bar{m}_I(0)$ คือ ค่าเอาต์พุตของตัวควบคุมที่เวลา t เท่ากับศูนย์

ผลของกริยาควบคุมแบบ I นี้จะทำให้ไม่เกิดออฟเซตขึ้นในระบบ และลดค่าพุ่งเกิน (Overshoot) ของระบบลงได้ แต่ถ้ากริยาควบคุมมีค่าสูงเกินไปจะทำให้ผลตอบสนองของกระบวนการช้าลง ผลตอบสนองของกริยาควบคุมแบบ I แสดงดังรูปที่ 2.3



รูปที่ 2.3 ผลตอบสนองของกริยาควบคุมแบบ I

จากรูปที่ 2.3 จะเห็นว่าสัญญาณเอาต์พุตของตัวควบคุมจะเพิ่มขึ้นจนมีค่าเท่ากับค่าสัญญาณผิดพลาดที่เวลาเท่ากับเวลาในการอินทิกรัลหรือเวลาในการรีเซต (T_i)

3. การควบคุมแบบอนุพันธ์

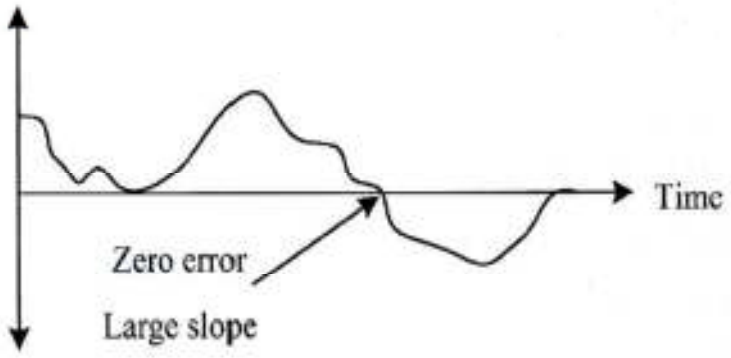
กริยาควบคุมแบบอนุพันธ์ (Derivative Control : D) เรียกอีกชื่อหนึ่งว่าการควบคุมแบบอัตราส่วน โดยสัญญาณเอาต์พุตของตัวควบคุมจะขึ้นอยู่กับอัตราการเปลี่ยนแปลงของค่าความผิดพลาดต่อเวลา จะเห็นว่าค่าความผิดพลาดนี้มีโอกาสเป็นศูนย์ได้ และค่าเอาต์พุตก็สามารถเปลี่ยนแปลงให้มีค่าสูงขึ้น เมื่อความผิดพลาดเปลี่ยนแปลง ซึ่งเรียกการกระทำดังกล่าวว่า อัตราการกระทำ (Rate Action) ดังนี้

$$m_D(t) = K_D \frac{de(t)}{dt} \quad (2.7)$$

เมื่อ $m_D(t)$ คือ ค่าเอาต์พุตของตัวควบคุมแบบ D

K_D คือ อัตราขยายของตัวควบคุมแบบ D

$\frac{de(t)}{dt}$ คือ ค่าความผิดพลาดที่เวลา t



รูปที่ 2.4 ตัวอย่างผลตอบสนองของกริยาควบคุมแบบ D

รูปที่ 2.4 แสดงตัวอย่างของผลตอบสนองของกริยาควบคุมแบบ D ซึ่งผลของกริยาควบคุมแบบ D นี้จะทำให้ผลตอบสนองของกระบวนการเร็วขึ้นแต่ระบบจะไวต่อการรบกวน

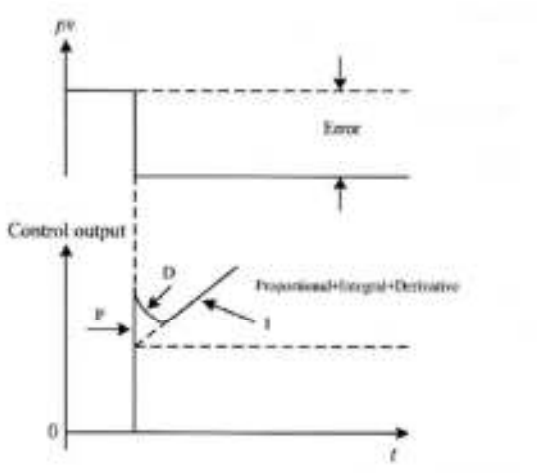
4. การควบคุมแบบ PID

จากที่ได้กล่าวมาแล้ว เพื่อให้ผลตอบสนองของระบบควบคุมมีสมรรถนะเป็นไปตามที่ต้องการจึงใช้กริยาควบคุมทั้ง 3 แบบร่วมกัน (Proportional + Integral + Derivative) ซึ่งจะทำให้ได้กริยาควบคุมแบบ PID ที่มีสมการคือ

$$m_{PID}(t) = \bar{m} + K_P e(t) + K_P K_I \int_0^t e(t)dt + K_P K_D \frac{de(t)}{dt} \quad (2.8)$$

หรือ

$$m_{PID}(t) = \overline{m} + K_C e(t) + \frac{K_C}{T_i} \int_0^t e(t)dt + K_C T_d \frac{de(t)}{dt} \quad (2.9)$$



รูปที่ 2.5 ตัวอย่างผลตอบสนองของกริยาควบคุมแบบ PID (Direct Action)

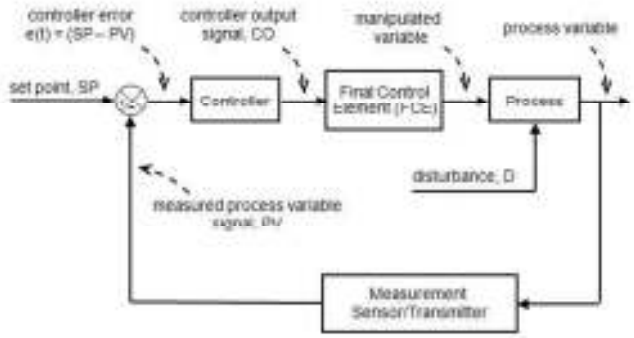
รูปที่ 2.5 แสดงตัวอย่างผลตอบสนองของกริยาควบคุมแบบ PID ซึ่งการควบคุมแบบนี้จะเหมาะกับกระบวนการที่มี Time Delay มาก ๆ เช่นการควบคุมอุณหภูมิเป็นต้น แต่ถ้าเราปรับค่า D และค่า I ไม่เหมาะสมก็จะทำให้เกิดการแกว่งได้เช่นเดียวกัน

2.3 การควบคุมกระบวนการ [4]-[5]

ในการควบคุมกระบวนการ (Control Loop) นั้น โดยทั่วไปมักจะเป็นการควบคุมแบบธรรมดาไม่ยุ่งยากซับซ้อนมากนักแต่สามารถใช้งานได้อย่างมีประสิทธิภาพ อย่างไรก็ตามในกระบวนการควบคุมบางกระบวนการจะต้องทำการรวมตัวควบคุมอัตโนมัติเข้าด้วยกัน เพื่อปรับปรุงการควบคุมให้ดีขึ้น ซึ่งตัวควบคุมเหล่านี้ได้แก่ตัวควบคุมแบบป้อนกลับ (Feedback Control) ตัวควบคุมแบบคาสเคด (Cascade Control) และตัวควบคุมแบบป้อนไปข้างหน้า (Feedforward Control) เป็นต้น

2.3.1 ตัวควบคุมแบบป้อนกลับ

ระบบควบคุมแบบป้อนกลับ หรือ Feedback Control หมายถึงระบบควบคุมที่ต้องการควบคุมให้สัญญาณด้านออกหรือผลตอบสนองของระบบมีค่าตามที่ต้องการ โดยการนำเอาสัญญาณด้านออกป้อนกลับมาเปรียบเทียบกับสัญญาณด้านเข้า ซึ่งการป้อนกลับนี้จะต้องเป็นการป้อนกลับแบบลบ (Negative Feedback) จากนั้นนำค่าความแตกต่างระหว่างด้านเข้ากับด้านออก (Error) ส่งต่อไปยังส่วนสร้างสัญญาณของตัวควบคุม เพื่อสร้างสัญญาณควบคุมที่เหมาะสมที่จะทำให้สัญญาณด้านออกของระบบที่ถูกควบคุมมีค่าตามต้องการ ระบบควบคุมป้อนกลับนี้อาจเรียกอีกอย่างหนึ่งว่าระบบควบคุมแบบลูปปิด (Closed Loop Control) หรือระบบควบคุมอัตโนมัติ รูปที่ 2.6 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับโดยทั่วไป

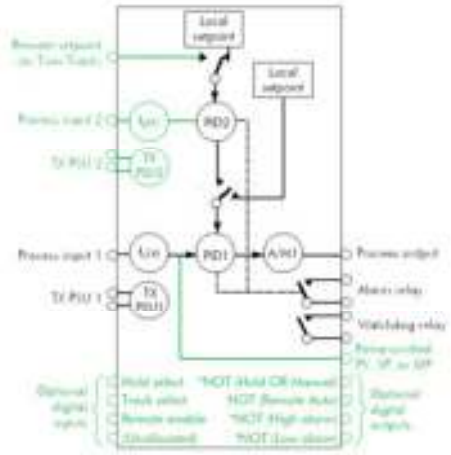


รูปที่ 2.6 บล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับ

ข้อดีของการควบคุมแบบป้อนกลับนี้คือ เป็นวิธีการง่าย ๆ ที่สามารถชดเชยการรบกวนระบบได้ในทุกรูปแบบ โดยเมื่อมีการรบกวนต่าง ๆ เข้ามาในระบบและส่งผลให้ค่าตัวแปรกระบวนการเบี่ยงเบนออกจากค่าเป้าหมาย ตัวควบคุมจะทำการเปลี่ยนแปลงค่าเอาต์พุตให้กลับเข้าสู่ค่าเป้าหมายโดยไม่สนใจว่ามีการรบกวนใดที่เข้ามาสู่ระบบบ้าง ส่วนข้อเสียของการควบคุมแบบนี้คือ ระบบจะสามารถชดเชยการรบกวนต่าง ๆ ได้ก็ต่อเมื่อค่าตัวแปรกระบวนการมีการเบี่ยงเบนออกจากค่าเป้าหมายแล้วเท่านั้น นั่นก็หมายความว่าการรบกวนต่าง ๆ ได้แพร่เข้าไปในกระบวนการผลิตแล้วก่อนที่การควบคุมแบบป้อนกลับจะเริ่มต้นทำงานเพื่อชดเชยการรบกวนเหล่านั้น

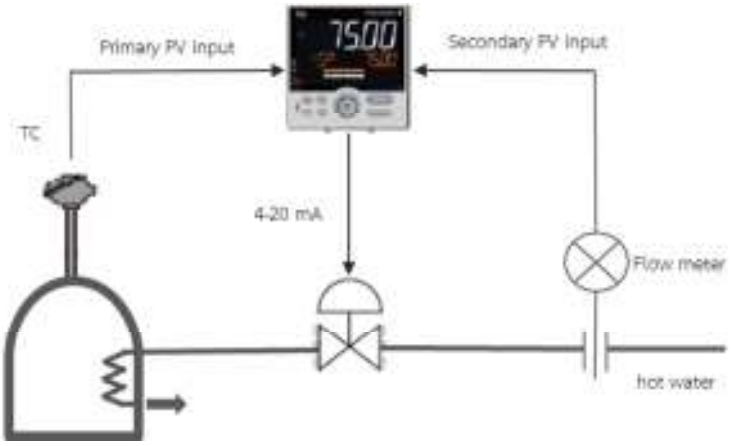
2.3.2 ตัวควบคุมแบบคาสเคด

ระบบควบคุมแบบคาสเคด ประกอบไปด้วยตัวควบคุมหลัก PID1 (Master Controller) และตัวควบคุมรอง PID2 (Slave Controller) ดังรูปที่ 2.7 ตัวควบคุมหลักจะทำหน้าที่รักษาค่าของตัวแปรกระบวนการให้เป็นไปตามต้องการ ส่วนตัวควบคุมรองจะทำหน้าที่ควบคุมตัวแปรอื่น ๆ ที่สามารถส่งผลให้ตัวแปรกระบวนการเกิดการเปลี่ยนแปลงได้ ดังนั้นสัญญาณควบคุมของตัวควบคุมหลักจะทำหน้าที่เป็นค่าเป้าหมายให้กับตัวควบคุมรอง หรือก็คือการทำหน้าที่ในการควบคุมการทำงานของวาล์วควบคุมทางอ้อมนั่นเอง



รูปที่ 2.7 ตัวอย่างผลิตภัณฑ์ตัวควบคุมแบบคาสเคด

จุดมุ่งหมายของการควบคุมแบบคาสเคดนี้จะเหมือนกับการควบคุมแบบลูปเดี่ยวทั่วไป แต่ที่แตกต่างกันก็คือ การควบคุมแบบคาสเคดจะเป็นการควบคุมในลักษณะของสองลูปที่ทำงานเกี่ยวเนื่องกัน โดยลูปที่สองจะมีผลทำให้ lag ของกระบวนการมีค่าลดลง ตัวควบคุมรองที่อยู่ภายในลูปที่สองนี้ถือได้ว่าเป็นอุปกรณ์ควบคุมสุดท้ายที่มีการทำงานที่ละเอียดมากเนื่องจากถูกควบคุมโดยตัวควบคุมหลักอีกทีหนึ่ง เช่นตัวอย่างในรูปที่ 2.8 ก็คือการควบคุมอัตราการไหลนั่นเอง เพราะว่าตัวควบคุมหลักจะไม่ได้ควบคุมหรือกำหนดตำแหน่งของวาล์วโดยตรงแต่จะควบคุมอัตราการไหลแทน



รูปที่ 2.8 ตัวควบคุมอุณหภูมิที่เป็นการควบคุมแบบคาสเคด (บริษัท Yokogawa)

จากรูปที่ 2.8 ผลิตภัณฑ์จะถูกส่งไปยังที่เตาอบความร้อนซึ่งความต้องการก็คือต้องการควบคุมผลิตภัณฑ์นี้ให้ได้อุณหภูมิตามต้องการ โดยที่กระบวนการนี้สิ่งที่เราต้องการควบคุมก็คืออุณหภูมิของผลิตภัณฑ์ ซึ่งจะต้องควบคุมให้อยู่ในช่วงเกณฑ์ที่กำหนดไว้ แต่การควบคุมนี้จะทำได้ค่อนข้างยากเพราะความร้อนจะเปลี่ยนแปลงอยู่ตลอดเวลา (เนื่องจากวัสดุของผลิตภัณฑ์ที่ถูกควบคุมกับน้ำร้อนที่ถูกจ่ายเข้ามา) เมื่อความร้อนทางเข้าเพิ่มขึ้น ความร้อนของเตาก็จะเพิ่มมากขึ้น นั่นก็คืออุณหภูมิของผลิตภัณฑ์จะสูงเกินกว่าที่ต้องการได้ ในกรณีที่ถ้ากระบวนการนี้มีตัวควบคุมเพียงตัวเดียวนั้น ตัวควบคุมจะพยายามแก้ไขให้อุณหภูมิของผลิตภัณฑ์เป็นไปตามต้องการ โดยควบคุมการไหลข้าวของน้ำร้อนให้ลดลง และถ้าในเวลาเดียวกันนี้อุณหภูมิของน้ำร้อนลดลงมาด้วย ผลที่เกิดขึ้นก็คือจะทำให้อุณหภูมิของผลิตภัณฑ์เปลี่ยนแปลงไปเกินกว่าช่วงที่ยอมรับได้ (เพราะว่าตัวควบคุมเพียงลูปเดียวจะต้องใช้เวลาอยู่ชั่วขณะหนึ่งกว่าจะควบคุมอุณหภูมินี้ได้) นั่นก็คือกระบวนการแบบนี้จะต้องใช้ระบบควบคุมแบบคาสเคด โดยที่ตัวควบคุมการไหลจะทำหน้าที่ป้องกันกระบวนการจากการเปลี่ยนแปลงของผลข้างเคียงที่เกิดขึ้นจากการเปลี่ยนแปลงของอุณหภูมิของน้ำร้อน และทำให้ตัวควบคุมหลักทำการควบคุมอุณหภูมิของผลิตภัณฑ์ได้อย่างมีเสถียรภาพมากขึ้น

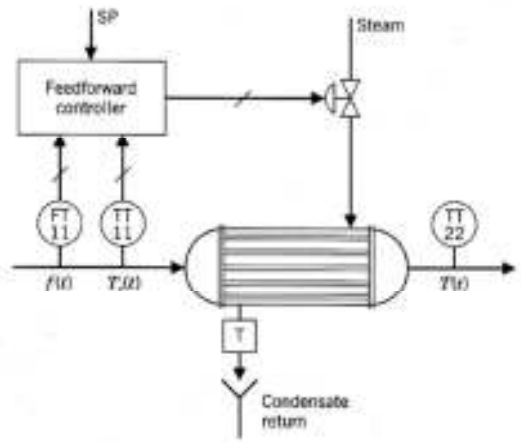
การวัดอัตราการไหลของน้ำร้อนนั้นทำได้โดยการวัดความดันตกคร่อมแผ่นออริฟิซและใช้เครื่องมือวัดความดันแตกต่างส่งสัญญาณที่ถอดค่ารากที่สองแล้วไปยังตัวควบคุมรอง ซึ่งทำหน้าที่ควบคุมอัตราการไหลของน้ำร้อน สำหรับการวัดอุณหภูมิจะวัดด้วยเซนเซอร์วัดอุณหภูมิแล้วจะส่งสัญญาณค่าอุณหภูมินี้ไปยังตัวควบคุมหลัก (เนื่องจากตัวควบคุมที่เป็นแบบอิเล็กทรอนิคส์จะมีวงจรแปลงสัญญาณของอุณหภูมิอยู่ในตัวแล้ว) เพื่อใช้เปรียบเทียบกับค่าเป้าหมายของอุณหภูมิผลิตภัณฑ์ที่

ต้องการ และสัญญาณควบคุมเอาต์พุตของตัวควบคุมหลักจะถูกส่งไปยังตัวควบคุมรอง และสัญญาณนี้จะเป็นค่าเป้าหมายของตัวควบคุมรองนั่นเอง

ถ้าวิเคราะห์ระบบด้วยสมการทางคณิตศาสตร์จะสามารถแสดงให้เห็นว่า ระบบควบคุมแบบคาสเคดนั้นจะทำให้ค่า Time Constants บางค่ามีค่าลดลง ซึ่งเป็นผลดีต่อระบบควบคุม ยิ่งไปกว่านั้นระบบควบคุมแบบคาสเคด ยังทำให้ลดผลกระทบจากการรบกวนต่าง ๆ ลงได้ การปรับค่าต่าง ๆ ในกรณีที่มีสิ่งรบกวนเข้ามาในกระบวนการทำได้แม่นยำกว่า และสามารถกำหนดขีดจำกัดบน (High Limit) และขีดจำกัดล่าง (Low limit) ในตัวควบคุมรองได้

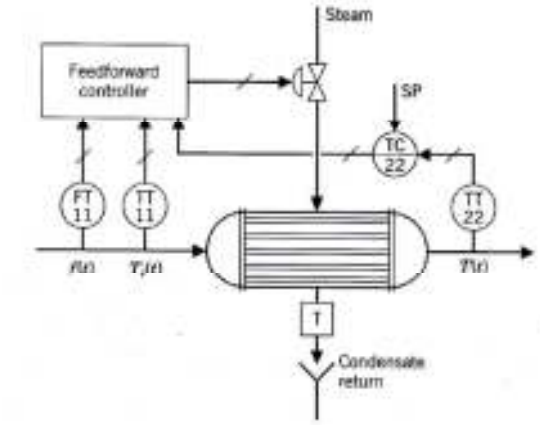
2.3.3 ตัวควบคุมแบบป้อนไปข้างหน้า

เนื่องจากการควบคุมแบบป้อนกลับนั้นจะส่งผลให้เกิดการเบี่ยงเบนชั่วคราวของค่าตัวแปรกระบวนการ ดังนั้นการควบคุมแบบป้อนไปข้างหน้าจึงเข้ามามีบทบาทในการแก้ปัญหานี้ โดยหลักการของการควบคุมแบบป้อนไปข้างหน้าคือเพื่อทำการชดเชยการรบกวนต่าง ๆ ก่อนที่จะส่งผลกระทบต่อระบบ การควบคุมแบบป้อนไปข้างหน้าจะมีการเรียกร้องให้มีการวัดค่าของการรบกวนต่าง ๆ ก่อนที่จะผ่านเข้าสู่กระบวนการและมีการคำนวณค่า MV ที่ต้องการเพื่อรักษาให้ค่าตัว PV มีค่าอยู่ในช่วงของค่า SP ซึ่งถ้าการคำนวณเป็นไปอย่างถูกต้องเหมาะสม ค่า PV ก็จะไม่เกิดการเบี่ยงเบน ดังนั้นการควบคุมแบบป้อนไปข้างหน้าจึงอาจกล่าวได้ว่าเป็นกลวิธีการควบคุมแบบเชิงรุกนั่นเอง



รูปที่ 2.9 ระบบควบคุมการแลกเปลี่ยนความร้อนโดยใช้การควบคุมแบบป้อนไปข้างหน้า

ตัวอย่างของการควบคุมแบบป้อนไปข้างหน้าแสดงดังรูปที่ 2.9 ซึ่งเป็นกระบวนการแลกเปลี่ยนพลังงานความร้อน สมมุติว่าการรบกวนหลักของระบบคืออุณหภูมิขาเข้า T_i (t) และอัตราการไหลของของเหลว f(t) ในการดำเนินการของการควบคุมแบบป้อนไปข้างหน้า การรบกวนสองอย่างนี้เป็นสิ่งแรกที่จะต้องทำการวัดและตัดสินใจว่าจะปรับวาล์วควบคุมอย่างไรให้สามารถชดเชยการรบกวนเหล่านี้ เพื่อที่จะควบคุมให้ตัวค่า PV คงอยู่ที่ SP ซึ่งจากรูปที่ 2.9 การชดเชยจะเกิดขึ้นเฉพาะกับการรบกวนเพียงสองตัวแปรเท่านั้น ในกรณีที่มีการรบกวนมากกว่านี้เข้ามาในกระบวนการ การควบคุมแบบนี้จะไม่สามารถชดเชยได้ ซึ่งจะส่งผลให้เกิดการเบี่ยงเบนของค่า PV ออกจาก SP อย่างถาวร ดังนั้นเพื่อที่จะหลีกเลี่ยงการเบี่ยงเบนนี้ การชดเชยแบบป้อนกลับจึงถูกนำมาเพิ่มลงในการควบคุมแบบป้อนไปข้างหน้า ดังแสดงในรูปที่ 2.10

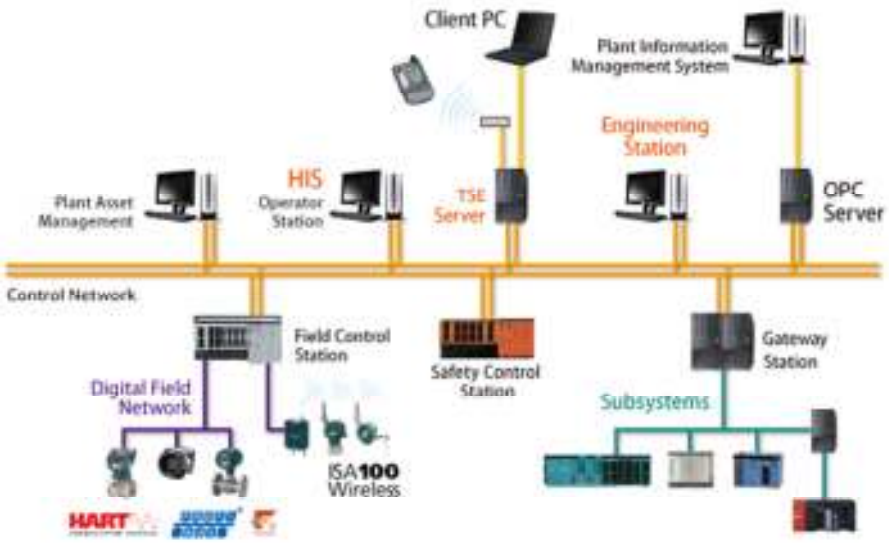


รูปที่ 2.10 ระบบควบคุมการแลกเปลี่ยนความร้อนโดยใช้การควบคุมแบบป้อนไปข้างหน้าร่วมกับการชดเชยป้อนกลับ

จากรูปที่ 2.10 การควบคุมแบบป้อนไปข้างหน้าจะทำหน้าที่ในการชดเชยการรบกวนที่เกิดจากสิ่งรบกวนหลัก ในขณะที่การควบคุมแบบป้อนกลับจะทำหน้าที่ในการชดเชยการรบกวนจากปัจจัยอื่น ๆ ซึ่งจะทำให้ค่าตัวแปรกระบวนการไม่เกิดการเบี่ยงเบนออกจากค่าเป้าหมาย ส่งผลให้เสถียรภาพของกระบวนการดีขึ้นนั่นเอง

2.4 ระบบดีซีเอส [6]

ดีซีเอส (DCS) ย่อมาจาก Distributed Control System หรือระบบควบคุมแบบกระจายส่วน เป็นเครื่องมือสำหรับควบคุมกระบวนการผลิตในโรงงานอุตสาหกรรม ดีซีเอสเป็นระบบควบคุมหลักที่ถูกเลือกใช้งานเป็นส่วนมากในปัจจุบัน เนื่องจากมีการพัฒนาให้มีขีดความสามารถและสมรรถนะในการทำงานเพิ่มขึ้นอย่างมากจากการใช้งานในอดีต อีกทั้งยังมีความสะดวกในการเขียนโปรแกรมในการควบคุมและแสดงผล นอกจากนั้นยังสามารถเลือกใช้อุปกรณ์ต่าง ๆ ภายในระบบได้จากหลากหลายผู้ผลิตไม่ว่าจะเป็นชิ้นส่วนทางด้านฮาร์ดแวร์หรือซอฟท์แวร์ ซึ่งอุปกรณ์ต่าง ๆ เหล่านี้จะเป็นอุปกรณ์มาตรฐานของเครื่องคอมพิวเตอร์หรือแม้กระทั่งเครือข่ายการสื่อสารต่าง ๆ ดีซีเอสเป็นระบบควบคุมที่มีหน่วยประมวลผล (Controller) อยู่หลายชุดแยกกัน และในการใช้งานมักจะถูกใช้ในการควบคุมกระบวนการแยกกันออกไปตามหน่วยการผลิต โดยหน่วยประมวลผลเหล่านี้จะเชื่อมต่อกันผ่านเครือข่ายสื่อสารหลักของระบบ เพื่อทำการสื่อสารและแลกเปลี่ยนข้อมูลกับหน่วยควบคุมอื่น ๆ เช่น หน่วยแสดงผล (Human Machine Interface) หน่วยทางวิศวกรรม (Engineering Unit) และหน่วยพิมพ์รายงาน เป็นต้น โครงสร้างพื้นฐานโดยทั่วไปของระบบดีซีเอสแสดงดังรูปที่ 2.11



รูปที่ 2.11 โครงสร้างพื้นฐานของระบบดีซีเอส

สถาปัตยกรรมดีซีเอสของผู้ผลิตในแต่ละบริษัทอาจมีโครงสร้างแตกต่างกันไปตามการออกแบบของผู้ผลิต แต่ทั้งนี้ทั้งนั้นส่วนประกอบของระบบดีซีเอสทุกระบบจะประกอบด้วยหน่วยการทำงานหลักที่คล้ายคลึงกันดังต่อไปนี้

2.4.1. หน่วยประมวลผลและส่วนอินพุตหรือเอาต์พุต

เป็นส่วนที่ทำหน้าที่ในการประมวลผลตามคำสั่งที่ถูกโปรแกรมไว้เพื่อใช้ในการควบคุมการผลิต โดยแบ่งออกเป็น 2 ส่วนคือ

1. หน่วยเชื่อมต่อกระบวนการ (Process Interface Module) ใช้สำหรับติดต่อระหว่างดีซีเอสกับกระบวนการผลิต ทำหน้าที่รับสัญญาณวัดจากเครื่องมือวัดที่อยู่ภายในกระบวนการผลิตให้กับดีซีเอส และส่งสัญญาณควบคุมจากดีซีเอสไปยังอุปกรณ์ที่อยู่ภายในกระบวนการผลิต ซึ่งหน่วยเชื่อมต่อกระบวนการของดีซีเอสจะประกอบไปด้วย หน่วยรับสัญญาณอนาล็อก (Analog Input Module) หน่วยส่งสัญญาณอนาล็อก (Analog Output Module) หน่วยรับสัญญาณดิจิตอล (Digital Input Module) และหน่วยส่งสัญญาณดิจิตอล (Digital Output Module)

2. หน่วยควบคุมกระบวนการ (Process Control Module) เป็นอุปกรณ์หลักของดีซีเอสสำหรับควบคุมกระบวนการผลิต โดยรับข้อมูลของกระบวนการผลิตจากหน่วยเชื่อมต่อกระบวนการเพื่อคำนวณค่าของสัญญาณควบคุมและส่งกลับไปยังหน่วยเชื่อมต่อกระบวนการเพื่อควบคุมกระบวนการผลิตอีกทีหนึ่ง

2.4.2 หน่วยแสดงผล

หน่วยแสดงผล (Human Machine Interface) เป็นส่วนที่ใช้ในการติดต่อสื่อสารกับผู้ปฏิบัติงาน ดีซีเอสอาจแยกหน่วยแสดงผลออกเป็น 2 ชุด ได้แก่หน่วยติดต่อและปฏิบัติงานของพนักงาน (Operator Workstation) และหน่วยปฏิบัติการทางวิศวกรรม (Engineering Workstation) หรือจะใช้อุปกรณ์ชุดเดียวกันก็ได้ หน่วยติดต่อและปฏิบัติงานของพนักงานเป็นส่วนที่ใช้ในการตรวจสอบและควบคุมกระบวนการผลิตผ่านทางจอแสดงผล ส่วนหน่วยปฏิบัติการทาง

วิศวกรรมจะใช้สำหรับการจัดโครงสร้างของระบบควบคุม การเชื่อมต่อหน่วยต่าง ๆ ภายในระบบดีซีเอส การกำหนดรายละเอียดและลำดับการแสดงแผนภาพกระบวนการผลิต การเก็บบันทึกข้อมูลประวัติการทำงานและแนวโน้มของกระบวนการ ตลอดจนข้อมูลการแจ้งเตือนต่าง ๆ ของดีซีเอสเป็นต้น

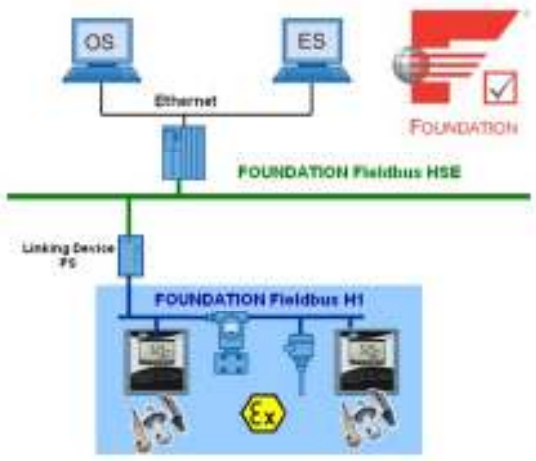
2.4.3 หน่วยเชื่อมต่อเครือข่าย

หน่วยเชื่อมต่อเครือข่าย (Communication Module) เป็นส่วนที่ทำหน้าที่ในการสื่อสารข้อมูลระหว่างหน่วยแสดงผล หน่วยควบคุมกระบวนการและส่วนอื่น ๆ ที่เกี่ยวข้องกับการควบคุม

สำหรับการใช้งานในอดีตที่ผ่านมา ระบบดีซีเอสจะถูกออกแบบจากผู้ผลิตที่เฉพาะเจาะจงกับระบบของตนเองทั้งในส่วนของฮาร์ดแวร์ (Hardware) และซอฟท์แวร์ (Software) ทำให้ไม่สามารถใช้งานร่วมกับอุปกรณ์ที่ผลิตจากผู้ผลิตรายอื่นได้ แต่ในปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์และการสื่อสารมีการพัฒนาไปอย่างรวดเร็ว จนทำให้เทคโนโลยีต่าง ๆ เหล่านี้เข้ามามีบทบาทในอุปกรณ์ต่าง ๆ ในระบบดีซีเอสมากขึ้น อาทิเช่น หน่วยแสดงผลที่ในอดีตต้องเลือกใช้อุปกรณ์และโปรแกรมการแสดงผลจากบริษัทผู้ผลิตโดยตรง แต่ในปัจจุบันสามารถเลือกใช้เป็นคอมพิวเตอร์ส่วนบุคคลคุณภาพสูงร่วมกับระบบปฏิบัติการวินโดวส์ (Windows) ได้ ซึ่งนอกจากจะสามารถติดตั้งโปรแกรมแสดงผลตัวแปรต่าง ๆ ที่ได้รับมาจากหน่วยประมวลผลแล้ว ยังสามารถติดตั้งโปรแกรมการใช้งานอื่น ๆ ที่ต้องการสำหรับการควบคุมได้อีกด้วย เครือข่ายการสื่อสารก็เป็นอีกส่วนหนึ่งที่มีการพัฒนาให้สามารถเลือกใช้ได้จากหลายผู้ผลิต โดยเครือข่ายการสื่อสารที่นิยมใช้กันในปัจจุบันได้แก่ เครือข่ายอีเทอร์เนต (Ethernet) ซึ่งมีความน่าเชื่อถือและสามารถเชื่อมต่อกับอุปกรณ์ได้หลากหลายประเภท จึงทำให้โครงสร้างของดีซีเอสมีความเป็นระบบเปิดมากขึ้นและสามารถนำไปเชื่อมต่อเข้ากับอุปกรณ์จากผู้ผลิตรายอื่น ๆ ได้สะดวกมากยิ่งขึ้น

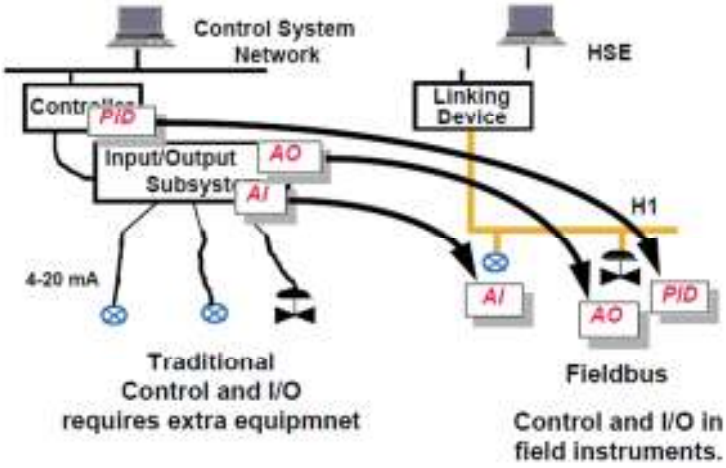
2.5 ฟาวน์เดชันฟิลด์บัส [6]-[8]

ฟาวน์เดชันฟิลด์บัสเป็นการสื่อสารแบบอนุกรม (Series) ด้วยสัญญาณดิจิตอลแบบสองทิศทางระหว่างเครื่องมือวัดกับระบบควบคุมหรือระหว่างเครื่องมือวัดเอง สามารถแบ่งตามความเร็วในการสื่อสารข้อมูลได้เป็นสองชนิดคือ H1 Fieldbus ซึ่งมีความเร็วในการรับส่งข้อมูลที่ระดับ 31.25 กิโลบิตต่อวินาที (kbit/s) เป็นเครือข่ายที่ใช้ในการเชื่อมต่อระหว่างอุปกรณ์เครื่องมือวัด เช่น เซนเซอร์ (Sensor) และ Actuator กับหน่วยอินพุตหรือเอาต์พุต และ HSE Fieldbus หรือระบบสื่อสารความเร็วสูง (High Speed Ethernet) มีความเร็วในการรับส่งข้อมูลที่ระดับ 100 เมกะบิตต่อวินาที (Mbit/s) เป็นบัสที่ใช้ในการเชื่อมต่อระหว่างหน่วยควบคุมที่มีความเร็วในการทำงานสูงเช่น พีแอลซี (PLC), ระบบรองของเอชวัน (H1 Subsystem), หน่วยเก็บข้อมูลระบบ (Data Server) และหน่วยเชื่อมต่อกับผู้ใช้งาน (Workstation) เป็นต้น โครงสร้างของระบบฟาวน์เดชันฟิลด์บัสแสดงดังรูปที่ 2.12



รูปที่ 2.12 โครงสร้างของระบบฟาวน์เดชันฟิลด์บัส

ฟาวน์เดชันฟิลด์บัสเป็นโปรโตคอลที่มีคุณสมบัติเด่นคือ สามารถกระจายหน่วยควบคุมไปยังส่วนต่าง ๆ ที่อยู่ภายในเครือข่ายได้ รูปที่ 2.13 แสดงการเปรียบเทียบระบบเครื่องมือวัดที่ใช้ในการควบคุมแบบเก่าและระบบฟาวน์เดชันฟิลด์บัส ซึ่งจะเห็นว่าระบบเครื่องมือวัดแบบดั้งเดิมที่ใช้สัญญาณกระแส 4-20 มิลลิแอมป์ จะมีศูนย์กลางการควบคุมอยู่ที่ตัวควบคุม (Controller) และมีสายสัญญาณเชื่อมต่อระหว่างเครื่องมือวัดไปยังหน่วยอินพุตและเอาต์พุตแบบตัวต่อตัว การควบคุมจะถูกสั่งการมาจากตัวควบคุมทั้งหมด ส่วนระบบฟาวน์เดชันฟิลด์บัสจะใช้เพียงสายสัญญาณเส้นเดียวจากตัวควบคุมกลางหรือโฮส (Host) ไปยังเครื่องมือวัดตัวแรกและต่อไปยังตัวต่อ ๆ ไป โดยจะมีการกำหนดชุดคำสั่งควบคุมหรือฟังก์ชันบล็อกในแต่ละวงควบคุม (Control Loop) ให้กระจายไปอยู่ภายในตัวอุปกรณ์แต่ละตัว ซึ่งคุณสมบัติในการกำหนดชุดคำสั่งควบคุมเหล่านี้จะขึ้นอยู่กับความสามารถของอุปกรณ์นั้น ๆ



รูปที่ 2.13 ระบบเครื่องมือวัดทั้งสองแบบ

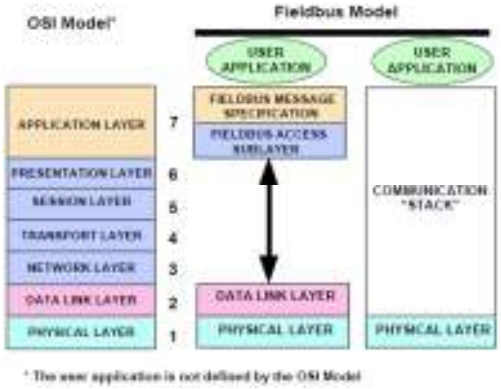
2.5.1 ฟาวน์เดชันฟิลด์บัส H1

เทคโนโลยีในการสื่อสารของฟาวน์เดชันฟิลด์บัส H1 จะใช้มาตรฐานของ OSI (Open Systems Interconnect Model) เป็นโมเดลในการสื่อสารระหว่างอุปกรณ์ต่าง ๆ ที่ต่ออยู่บนเครือข่ายดังรูปที่ 2.14 ซึ่งประกอบไปด้วยส่วนสำคัญ 3 ส่วน คือ

1. Physical Layer เป็นส่วนของโครงสร้างทางด้านฮาร์ดแวร์ ทำหน้าที่ในการเชื่อมต่ออุปกรณ์ทุกตัวในเครือข่ายเข้าด้วยกันเพื่อใช้เป็นเส้นทางเดินของสัญญาณ ซึ่งสามารถเทียบได้ Layer 1 ของ OSI Model

2. Communication Stack เป็นส่วนที่ใช้ในการสื่อสารระหว่าง Physical Layer ของระบบกับผู้ใช้งาน ประกอบไปด้วย Data Link Layer (DLL) ซึ่งเทียบได้กับ Layer 2 ของ OSI Model และ Fieldbus Message Specification (FMS) ซึ่งเทียบได้กับ Layer 7 ของ OSI Model

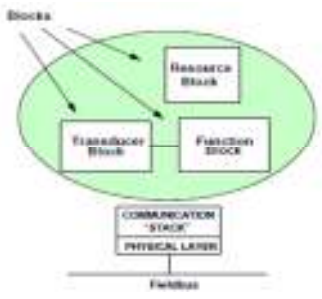
3. User Application Layer เป็นส่วนที่ฟาวน์เดชันฟิลด์บัสใช้ติดต่อกับผู้ใช้งาน เป็นชั้นที่ไม่ได้อยู่ใน OSI Model เนื่องจากเป็นลักษณะเฉพาะที่ถูกกำหนดขึ้นโดย Fieldbus Foundation ซึ่งทำให้มีความแตกต่างจากโมเดลอื่น ๆ



รูปที่ 2.14 โปรโตคอลของฟาวน์เดชันฟิลด์บัสเทียบกับโมเดลของ OSI

2.5.2 ฟิลด์บัสบล็อก

Fieldbus Foundation ได้กำหนดมาตรฐานของ User Application Layer โดยการใช้ฟิลด์บัสบล็อก (Fieldbus Blocks) เป็นตัวแทนของแอพพลิเคชันแต่ละชนิดซึ่งมีหน้าที่แตกต่างกันออกไป ชนิดของฟิลด์บัสบล็อกที่ใช้ในระดับชั้นนี้แสดงดังรูปที่ 2.15



รูปที่ 2.15 ชนิดของบล็อกที่ใช้ใน User Application Layer

จากรูปที่ 2.15 อุปกรณ์ที่อยู่ในเครือข่ายจะถูกกำหนดค่าต่าง ๆ ที่เป็นลักษณะเฉพาะของแต่ละอุปกรณ์โดยใช้รีซอร์สบล็อก (Resource Block) และทรานสดิวเซอร์บล็อก (Transducer Block) ส่วนทฤษฎีหรือหลักการที่ใช้ในการควบคุมกระบวนการจะถูกสร้างขึ้นโดยฟังก์ชันบล็อก (Function Block)

2.5.2.1 ทรานสดิวเซอร์บล็อก

ทรานสดิวเซอร์บล็อกจะติดต่อกับหน่วยอินพุตหรือเอาต์พุตภายในกระบวนการผ่านทางฟังก์ชันบล็อกโดยที่ทรานสดิวเซอร์บล็อกจะมีหน้าที่ในการอ่านค่าจากเซนเซอร์และเขียนคำสั่งการทำงานส่งกลับไปยังอุปกรณ์ที่เกี่ยวข้อง ซึ่งบล็อกนี้จะได้รับอนุญาตให้ทำงานได้บ่อยครั้งตามที่ต้องการเพื่อให้ได้รับค่ากระบวนการที่ดีที่สุดจากเซนเซอร์ และเพื่อให้มั่นใจว่าคำสั่งควบคุมที่ส่งออกไปมีความถูกต้องเหมาะสมโดยปราศจากการเพิ่มภาระงานให้กับฟังก์ชันบล็อก ซึ่งจะทำหน้าที่ในการอ่านข้อมูลเหล่านั้น ทรานสดิวเซอร์บล็อกจะแยกฟังก์ชันบล็อกออกจากลักษณะเฉพาะของอินพุตหรือเอาต์พุตที่ถูกกำหนดโดยผู้ผลิตแต่ละบริษัท ซึ่งภายในทรานสดิวเซอร์บล็อกจะประกอบไปด้วยข้อมูลต่าง ๆ เช่น ชนิดของเซนเซอร์และวันที่ทำการสอบเทียบเครื่องมือวัดเป็นต้น ตัวอย่างข้อมูลบางส่วนที่อยู่ภายในทรานสดิวเซอร์บล็อกแสดงดังตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างข้อมูลบางส่วนที่อยู่ในทรานสดิวเซอร์บล็อก

Transducer Channel #	Input (I) Output (O)	Function	Notes
0		No connection	
1	O	Open/Close	Readback = isOpened/isClosed
2	O	Open	Readback = isOpened
3	O	Close	Readback = isClosed
4	O	Stop	Readback = isStopped
5	O	Open/Close/Stop	Readback = Opening, isOpened, Closing, isClosed, isStopped
6	I	Valve is Opened/Valve is Closed	
7	I	Valve is opened	
8	I	Valve is closed	
9	I	Auxiliary limit switch input	
10	I	Cycle Time Open Alarm (CTOA)	
11	I	Cycle Time Closed Alarm (CTCA)	
12	I	Cycle Count Alarm (CCA)	
13	I	Masked Alarm	CTOA I CTCA I CCA| TMPH (Temperature High) ITMPL (Temperature Low)

2.5.2.2 รีซอร์สบล็อก

รีซอร์สบล็อกประกอบไปด้วยคุณลักษณะเฉพาะทางฮาร์ดแวร์ของอุปกรณ์ฟิลด์บัสแต่ละตัว ในบล็อกนี้จะไม่มีตัวแปรอินพุตหรือเอาต์พุต ชุดคำสั่งที่อยู่ภายในบล็อกจะเกี่ยวข้องกับการตรวจสอบหรือการควบคุมการดำเนินการโดยทั่วไปในส่วนฮาร์ดแวร์ของตัวอุปกรณ์ การทำงานของชุดคำสั่งจะขึ้นอยู่กับคุณลักษณะทางกายภาพของตัวอุปกรณ์ซึ่งถูกกำหนดโดยบริษัทผู้ผลิต ในอุปกรณ์ฟิลด์บัสแต่ละตัวจะมีรีซอร์สบล็อกได้เพียงหนึ่งบล็อกเท่านั้น ซึ่งถ้ากรณีที่รีซอร์สบล็อกอยู่ในโหมด Out of Service ก็จะส่งผลกระทบต่อบล็อกอื่น ๆ ที่อยู่ใน User Application Layer ด้วย ตัวอย่างข้อมูลบางส่วนที่อยู่ภายในรีซอร์สบล็อกแสดงดังตารางที่ 2.2

ตารางที่ 2.2 ตัวอย่างข้อมูลบางส่วนที่อยู่ภายในรีซอร์สบล็อก

ST_REV	The revision level of the static data associated with the function block. To support tracking changes in static parameter attributes, the associated block's static revision parameter will be incremented each time a static parameter attribute is written but the value is not changed.
TAG_DESC	The user description of the intended application of the block.
STRATEGY	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
MODE_BLK	The actual, target, permitted, and normal modes of the block.
BLOCK_ERR	This parameter reflects the error status associated with the hardware of software components associated with a block. It is a bit string, so that multiple errors may be shown.
RESOURCE STATE	State of the function block application state machine.
TEST READ WRITE	Read/write test parameters - used only for conformance testing.
DD_RESOURCE	String identifying the tag of the resource which contains the Device Description for this resource.
MANUFAC_ID	Manufacturer identification number - used by an interface device to locate the DD file for the resource.
DEVICE TYPE	Manufacturer's model number associated with the resource - used by interface devices to location the DD file for the resource.
DD REVISION	Revision of the DD associated with the resource - used by an interface device to locate the DD file for the resource.
GRANT_DENY	Options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.
HARD_TYPES	The types of hardware available as channel numbers.
RESTART	Allows a manual restart to be initiated. Several degrees of restart are possible. They are 1: Run, 2: Restart resource, 3: Restart with defaults, 4: Restart processor; and 5: Restart with factory defaults.
FEATURES	Used to show supported resource block options.
FEATURE SELECTION	Used to select resource block options. Enable 0 x 32 bit to allow readbacks from function blocks.
CYCLE_TYPE	Identifies the block execution methods available for this resource.
CYCLE SELECTION	Used to select the block execution method for this resource.
MINIMUM CYCLE TIME	Time duration of the shortest cycle interval of which the resource is capable.
FREE_SPACE	Percent of memory available for further configuration. Zero in a preconfigured resource.
FREE_TIME	Percent of the block processing time that is free to process additional blocks.
SHED REMOTE CASCASE	Time duration at which to give up on computer writes to function block RCAS locations. Shed from RCAS shall never happen when SHED REMOTE CASCASE = 0.

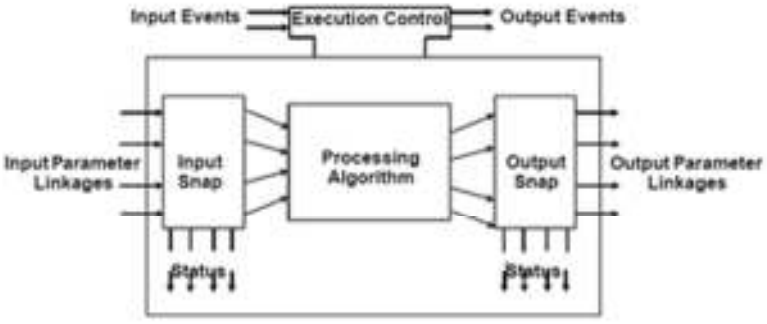
2.5.2.3 ฟังก์ชันบล็อก

ฟังก์ชันบล็อกภายในอุปกรณ์ฟิลด์บัสประกอบไปด้วยฟังก์ชันต่าง ๆ ที่จำเป็นสำหรับการควบคุมกระบวนการ เนื่องจากระบบแต่ละระบบมีความแตกต่างกัน ดังนั้นการผสมผสานและการกำหนดฟังก์ชันต่าง ๆ ในการทำงานจึงมีความแตกต่างกัน Fieldbus Foundation จึงได้มีการออกแบบกลุ่มของฟังก์ชันบล็อกให้แต่ละบล็อกสามารถตอบสนองความต้องการในการใช้งานที่แตกต่างกัน

ฟังก์ชันบล็อกประกอบด้วยฟังก์ชันในการควบคุมกระบวนการ เช่นอนาล็อกอินพุต (AI) อนาล็อกเอาต์พุต (AO) และ PID ฟังก์ชันบล็อกมาตรฐานจะประกอบด้วยโครงสร้างทั่วไปสำหรับการกำหนดฟังก์ชันบล็อกอินพุต เอาต์พุต ตัวแปรควบคุม เหตุการณ์ การแจ้งเตือน และโหมดการควบคุม และทำการรวบรวมบล็อกเหล่านั้นไปยังกระบวนการ ซึ่งสามารถดำเนินการภายในอุปกรณ์เพียงตัวเดียวหรือดำเนินการบนเครือข่ายฟิลด์บัสก็ได้ นอกจากนี้การดำเนินการของแต่ละฟังก์ชันบล็อกยังสามารถกำหนดตารางการทำงานที่แน่นอนได้

ฟาวน์เดชันฟิลด์บัสอนุญาตให้บริษัทผู้ผลิตสามารถเพิ่มตัวแปรต่าง ๆ ของตนเองเข้าไปภายในฟังก์ชันบล็อกได้โดยการนำเข้าและการกำหนดกลุ่มย่อยของแต่ละหมวดหมู่ขึ้นมา ซึ่งเป็นการทำให้นิยามของฟังก์ชันบล็อกกว้างขึ้นส่งผลให้สามารถรองรับเทคโนโลยีใหม่ ๆ ได้มากขึ้นด้วย

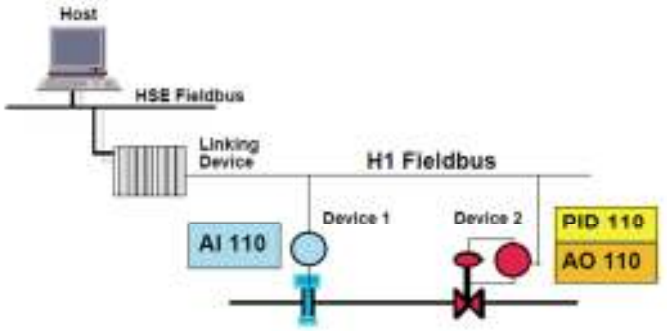
รูปที่ 2.16 แสดงตัวอย่างของโครงสร้างภายในฟังก์ชันบล็อก เมื่อการดำเนินการเริ่มต้นขึ้น ค่าตัวแปรอินพุตจากบล็อกอื่น ๆ จะถูก Snap โดยบล็อก ซึ่งขั้นตอนการ Snap อินพุตนี้เป็นไปเพื่อทำให้แน่ใจว่าค่าพารามิเตอร์จะไม่มีการเปลี่ยนแปลงระหว่างที่ฟังก์ชันบล็อกกำลังดำเนินการ ค่าของตัวแปรที่เกิดการเปลี่ยนแปลงถัดมาจะไม่มีผลต่อค่าที่ถูก Snap และจะไม่ถูกนำมาใช้ในขณะที่ฟังก์ชันบล็อกกำลังดำเนินการ



รูปที่ 2.16 โครงสร้างภายในของฟังก์ชันบล็อก

เมื่อค่าอินพุตถูก Snap จะถูกดำเนินการโดยสมการต่าง ๆ ที่อยู่ภายในบล็อกเพื่อทำให้ได้ค่าเอาต์พุต การทำงานของสมการต่าง ๆ ถูกควบคุมโดยการกำหนดค่าตัวแปรที่บรรจุอยู่ภายในฟังก์ชันบล็อก ซึ่งพารามิเตอร์เหล่านี้จะไม่ปรากฏให้เห็นในช่วงของตัวแปรอินพุตหรือเอาต์พุตปกติ อย่างไรก็ตามพารามิเตอร์เหล่านี้สามารถถูกเข้าถึงและเปลี่ยนแปลงได้ตามที่ถูกระบุโดยฟังก์ชันบล็อก ฟังก์ชันบล็อกสามารถเก็บข้อมูลของตัวมันได้เป็นระยะเวลาสั้น ๆ และมีหน่วยจัดเก็บสำหรับการตรวจสอบพฤติกรรมของตัวมันเอง

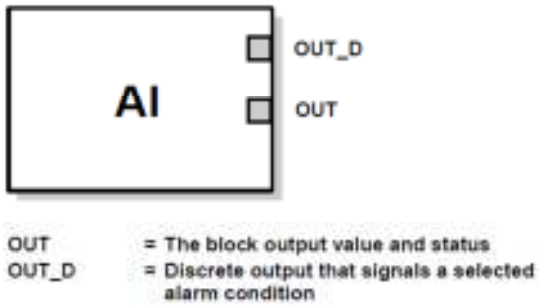
ฟังก์ชันบล็อกเหล่านี้สามารถกำหนดให้อยู่ภายในตัวอุปกรณ์ฟิลด์บัสได้ โดยจะขึ้นอยู่กับความสามารถของอุปกรณ์ฟิลด์บัสนั้น ๆ ตัวอย่างเช่นในรูปที่ 2.17 อุปกรณ์วัดอุณหภูมิสามารถประกอบไปด้วยฟังก์ชันบล็อก AI วาล์วควบคุมจะประกอบไปด้วยฟังก์ชันบล็อก PID และฟังก์ชันบล็อก AO ดังนั้นจะเห็นได้ว่าวงควบคุมที่สมบูรณ์สามารถสร้างได้จากอุปกรณ์พื้นฐานเพียงสองตัว คือทรานสมิตเตอร์และวาล์วควบคุมเท่านั้น



รูปที่ 2.17 ตัวอย่างการควบคุมที่ใช้ฟังก์ชันบล็อกที่อยู่ภายในตัวอุปกรณ์

อย่างไรก็ตาม Fieldbus Foundation ได้กำหนดกลุ่มของฟังก์ชันบล็อกมาตรฐานสำหรับการควบคุมพื้นฐานออกเป็นหลายกลุ่มด้วยกัน ซึ่งในที่นี้จะยกตัวอย่างเฉพาะบล็อกที่ใช้ในวิทยานิพนธ์ฉบับนี้ ดังต่อไปนี้

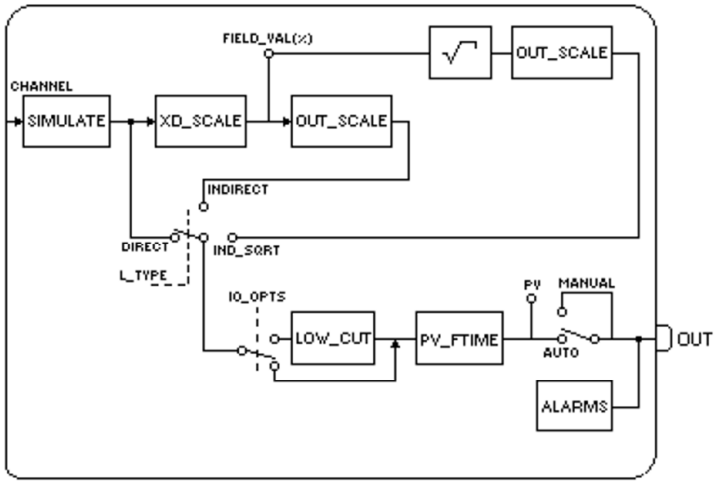
1. ฟังก์ชันบล็อก AI



รูปที่ 2.18 ฟังก์ชันบล็อก AI

ฟังก์ชันบล็อก AI ทำหน้าที่เกี่ยวกับการวัดของอุปกรณ์เครื่องมือวัดและทำให้อุปกรณ์เหล่านั้นสามารถใช้ได้กับฟังก์ชันบล็อกอื่น ๆ ค่าตัวแปรเอาต์พุตจากฟังก์ชันบล็อก AI จะอยู่ในรูปของหน่วยทางวิศวกรรมและประกอบด้วยสถานะที่ระบุถึงคุณภาพของการวัด อุปกรณ์เครื่องมือวัดมักจะวัดค่าตัวแปรได้หลายค่าหรือได้รับค่าตัวแปรที่สามารถนำมาใช้ได้มาจากช่องทาง (Channel) อื่น ๆ ดังนั้นจึงจะต้องมีการกำหนดหมายเลขช่องทางของตัวแปรที่จะนำมาใช้ในกระบวนการให้ถูกต้อง

ฟังก์ชันบล็อก AI รองรับฟังก์ชันสัญญาณเตือนต่าง ๆ การปรับสัญญาณ การกรองสัญญาณ การคำนวณสถานะของสัญญาณ โหมดในการควบคุมและการจำลองการทำงาน (Simulation) ในโหมดอัตโนมัติ (Automatic Mode) บล็อกของตัวแปรเอาต์พุต (OUT) จะสะท้อนให้เห็นถึงสถานะของ PV ในโหมดการควบคุมด้วยมือ (Manual Mode) ค่า OUT จะถูกกำหนดโดยผู้ใช้ Manual Mode จะถูกสะท้อนให้เห็นจากสถานะของเอาต์พุต ในบางผู้ผลิตจะมีการเพิ่มฟังก์ชันของค่าเอาต์พุตที่ไม่ต่อเนื่อง (OUT_D) ซึ่งใช้สำหรับแสดงผลเมื่อเงื่อนไขของระบบแจ้งเตือนต่าง ๆ มีการทำงาน การตรวจจับของระบบแจ้งเตือนจะขึ้นอยู่กับค่า OUT และการกำหนดขอบเขตการแจ้งเตือนของผู้ใช้ รูปที่ 2.19 แสดงตัวอย่างวงจรภายในฟังก์ชันบล็อก AI



รูปที่ 2.19 โครงสร้างภายในฟังก์ชันบล็อก AI

ลักษณะต่าง ๆ ที่สำคัญของฟังก์ชันบล็อก AI ซึ่งจะต้องมีการกำหนดค่าให้ถูกต้องเหมาะสม เพื่อให้การควบคุมกระบวนการเป็นไปอย่างมีประสิทธิภาพ มีดังต่อไปนี้

- Filtering

ลักษณะของฟังก์ชัน Filtering คือผู้ใช้สามารถเปลี่ยนค่าเวลาในการตอบสนองของตัวอุปกรณ์เครื่องมือวัดให้มีการเปลี่ยนแปลงอย่างราบเรียบในการอ่านค่าเอาต์พุต ซึ่งเป็นผลมาจากการที่ค่าอินพุตมีการเปลี่ยนแปลงอย่างรวดเร็ว ผู้ใช้สามารถปรับเปลี่ยนค่าคงที่ในการฟิลเตอร์ผ่านทางตัวแปร PV_FTIME และสามารถยกเลิกการใช้ฟังก์ชันฟิลเตอร์ได้โดยการกำหนดค่าเวลาในการฟิลเตอร์ให้เป็นศูนย์

- Direct

การแปลงสัญญาณแบบ Direct จะอนุญาตให้สัญญาณสามารถผ่านไปยังช่องทางของค่าตัวแปรอินพุตได้โดยตรง โดยที่ค่า PV จะมีค่าเท่ากับค่าของช่องสัญญาณนั้น โดยที่

PV = Channel Value

- Indirect

การแปลงสัญญาณแบบ Indirect จะทำการเปลี่ยนสัญญาณให้เป็นเชิงเส้นไปยังช่องทางของค่าตัวแปรอินพุต โดยจะเปลี่ยนจากค่าสัญญาณจริงในช่วงที่ตัวอุปกรณ์วัดได้เป็นสัญญาณที่อยู่ในช่วงและหน่วยของค่าตัวแปรกระบวนการและค่าเอาต์พุต (OUT_SCALE) โดยที่

$$PV = \left(\frac{FIELD_VAL}{100}\right) x\left(EU^{**}@100\% - EU^{**}@0\%\right) + EU^{**}@0\% \qquad (2.9)$$

**OUT_SCALE values

- Modes

อนาล็อกอินพุตฟังก์ชันบล็อกรองรับโหมดการดำเนินการทั้งหมด 3 โหมดด้วยกัน ซึ่งกำหนดโดยตัวแปร MODE_BLK ดังนี้

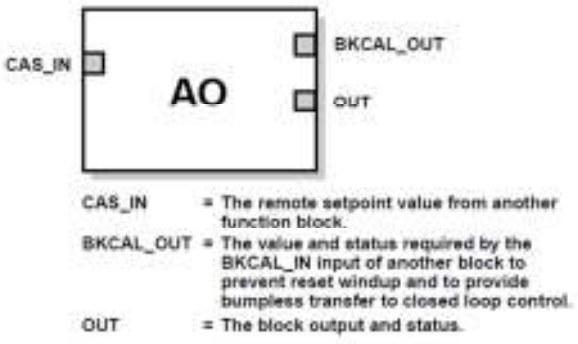
1.) Manual (Man) ค่า OUT จะถูกกำหนดค่าโดยผู้ใช้งาน

2.) Automatic (Auto) ค่า OUT เป็นผลมาจากค่าอนาล็อกอินพุตที่ได้จากการวัดหรือค่าที่ได้จากการจำลองเมื่ออยู่ในโหมด Simulation

3.) Out of Services (O/S) บล็อกจะไม่มีการดำเนินการ ค่า FIELD_VAL และค่า PV ไม่มีการอัพเดตและสถานะของ OUT ถูกกำหนดเป็น Bad : Out of Service เมื่อตัวแปร BLOCK_ERR แสดงสถานะ Out of Service ในโหมดนี้ผู้ใช้สามารถเปลี่ยนแปลงการกำหนดค่าต่าง ๆ ของตัวแปรทั้งหมดที่ถูกกำหนดไว้ก่อนหน้าได้

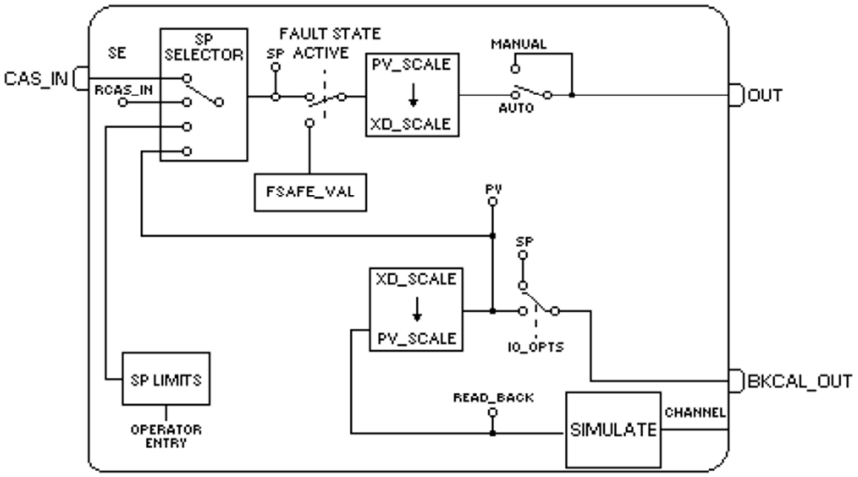
โหมดเป้าหมายของบล็อกอาจถูกจำกัดให้เลือกได้เพียงหนึ่งหรือมากกว่าภายในจำนวนโหมดที่รองรับ

2. ฟังก์ชันบล็อก AO



รูปที่ 2.20 ฟังก์ชันบล็อก AO

ฟังก์ชันบล็อก AO ทำหน้าที่กำหนดค่าเอาต์พุตไปยังตัวอุปกรณ์ที่อยู่ในกระบวนการผ่านช่องทางที่กำหนด อนาล็อกเอาต์พุตบล็อกรองรับโหมดการควบคุม การคำนวณสถานะของสัญญาณและการจำลองการทำงาน รูปที่ 2.21 แสดงตัวอย่างวงจรภายในฟังก์ชันบล็อก AO



รูปที่ 2.21 โครงสร้างภายในฟังก์ชันบล็อก AO

เพื่อที่จะกำหนดค่าเอาต์พุตสำหรับฟังก์ชันบล็อก AO สิ่งแรกคือเราจะต้องเลือกโหมดการทำงานเพื่อกำหนดลักษณะภายในบล็อกซึ่งจะมีผลต่อการกำหนดค่า SP ของบล็อก ในโหมด Manual ลักษณะของค่า OUT จะถูกกำหนดโดยผู้ใช้งานและมีความเป็นอิสระจากค่าเป้าหมาย ในโหมดอัตโนมัติ ค่า OUT จะถูกกำหนดโดยอัตโนมัติขึ้นอยู่กับค่าที่ถูกระบุโดยค่า SP ในรูปของหน่วยทางวิศวกรรมและการเลือกลักษณะของอินพุตหรือเอาต์พุต (IO_OPTS) นอกจากนี้เรายังสามารถกำหนดขอบเขตของค่า SP และอัตราการเปลี่ยนแปลงของค่าเป้าหมายซึ่งถูกส่งไปยัง OUT ได้

ในโหมดคาสเคด การเชื่อมต่อคาสเคดอินพุต (CAS_IN) ถูกใช้ในการอัพเดตค่า SP ค่าเอาต์พุตป้อนกลับ (BKCAL_OUT) ถูกเชื่อมต่อกับค่าอินพุตป้อนกลับ (BKCAL_IN) ของบล็อกก่อนหน้าซึ่งเป็นบล็อกที่ป้อนค่า CAS_IN ฟังก์ชันนี้จะเตรียมวงจรลดการกระแทก (Bumpless Transfer) ในโหมดที่มีการเปลี่ยนแปลงและ Windup Protection ที่บล็อกต้นทาง ลักษณะของ OUT หรือค่าอนาล็อกที่อ่านกลับมาได้ เช่น ตำแหน่งของวาล์ว ถูกแสดงโดยค่า PV ในหน่วยทางวิศวกรรม

ลักษณะต่าง ๆ ที่สำคัญของฟังก์ชันบล็อก AO ซึ่งจะต้องมีการกำหนดค่าให้ถูกต้องเหมาะสม เพื่อให้การควบคุมกระบวนการเป็นไปอย่างมีประสิทธิภาพ มีดังต่อไปนี้

- Mode

ฟังก์ชันบล็อก AO รองรับการทำงานของโหมดดังต่อไปนี้

1.) Manual (Man) ผู้ใช้สามารถกำหนดค่าเอาต์พุตไปยังช่องของ IO ได้ด้วยตัวเองโดยผ่านทางคุณสมบัติของ OUT โหมดนี้ใช้เป็นโหมดเริ่มต้นเมื่อมีการการซ่อมบำรุงหรือแก้ไขปัญหาต่าง ๆ

2.) Automatic (Auto) ค่า OUT เป็นผลมาจากเป้าหมายของการควบคุมกระบวนการซึ่งถูกกำหนดโดยคุณสมบัติของค่า SP

3.) Cascade (Cas) คุณสมบัติของค่า SP จะถูกกำหนดโดยฟังก์ชันบล็อกอื่นผ่านทางการเชื่อมต่อไปยัง CAS_IN ค่า SP จะถูกใช้ในการกำหนดคุณสมบัติของ OUT โดยอัตโนมัติ

4.) Remote Cascade (RCas) ค่า SP ของโหมดนี้จะกำหนดโดยเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นโฮส โดยการเขียนข้อมูลไปยังตัวแปร RCAS_IN ค่า SP จะถูกใช้ในการกำหนดคุณสมบัติของ OUT โดยอัตโนมัติ

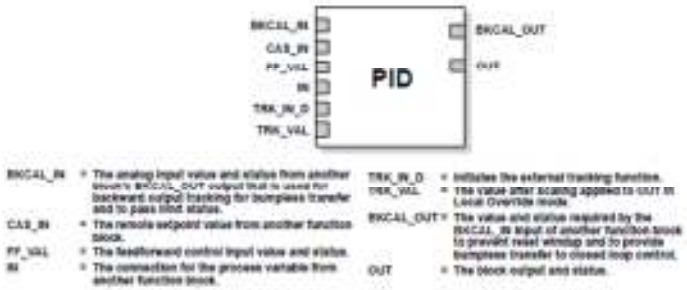
5.) Out of Service (O/S) หมายถึงบล็อกไม่มีการดำเนินการ ช่องของเอาต์พุตจะคงค่าของการดำเนินการสุดท้ายเอาไว้และสถานะของ OUT จะถูกกำหนดให้เป็น Bad: Out of Service คุณสมบัติของ BLOCK_ERR จะแสดงข้อความ Out of Service

6.) Initialization Manual (Iman) ฟังก์ชันนี้จะถูกแสดงเมื่อเส้นทางเชื่อมต่อไปยังเอาต์พุตไม่สามารถใช้งานได้และค่าเอาต์พุตจะถูกคงไว้ที่ค่าตำแหน่งสุดท้าย

7.) Local Override (LO) เอาต์พุตของบล็อกนี้จะไม่ตอบสนองต่อค่า OUT เนื่องจากรีซอร์สบล็อกอยู่ในโหมด LO หรือมีการกระทำที่ผิดไปจากสถานะการใช้งานปกติ

โหมดเป้าหมายของบล็อกอาจถูกจำกัดให้เลือกได้เพียงหนึ่งหรือมากกว่าของโหมดดังนี้ Man, Auto, Cas, RCas, หรือ O/S

3. ฟังก์ชันบล็อก PID



รูปที่ 2.22 ฟังก์ชันบล็อก PID

ฟังก์ชันบล็อก PID ประกอบด้วยตรรกะสำคัญทั้งหมดที่จำเป็นต้องใช้ในการดำเนินการควบคุมแบบ P แบบ I และแบบ D ฟังก์ชันบล็อก PID รองรับโหมดการควบคุม การปรับสัญญาณและการกำหนดขอบเขต การควบคุมแบบป้อนไปข้างหน้า การ Override Tracking การตรวจจับการแจ้งเตือนและการแพร่ขยายสถานะของสัญญาณ

บล็อกนี้รองรับสมการ PID สองรูปแบบคือ รูปแบบมาตรฐานและแบบอนุกรม ผู้ใช้งานสามารถเลือกสมการที่เหมาะสมกับความต้องการใช้งานโดยผ่านทางตัวแปรที่ชื่อว่า FORM สมการ PID ตามมาตรฐาน ISA จะถูกกำหนดให้เป็นค่าเริ่มต้น โดย

$$\text{Standard Out} = \text{GAIN x e x } \left(1+\frac{1}{T_r S+1}+\frac{T_d S}{\alpha \text{ x } T_d S}+1\right)+F \qquad (2.10)$$

$$\text{Series Out} = \text{GAIN x e x } \left[\left(1+\frac{1}{T_r S}\right)+\left(\frac{T_d S+1}{\alpha+T_d S+1}\right)\right]+F \qquad (2.11)$$

ซึ่ง GAIN คือ ค่าอัตราขยายแบบ P

T_r คือ ค่าอินทิกรัลหรือเวลาในการรีเซต

S คือ การดำเนินการลาปลาซ

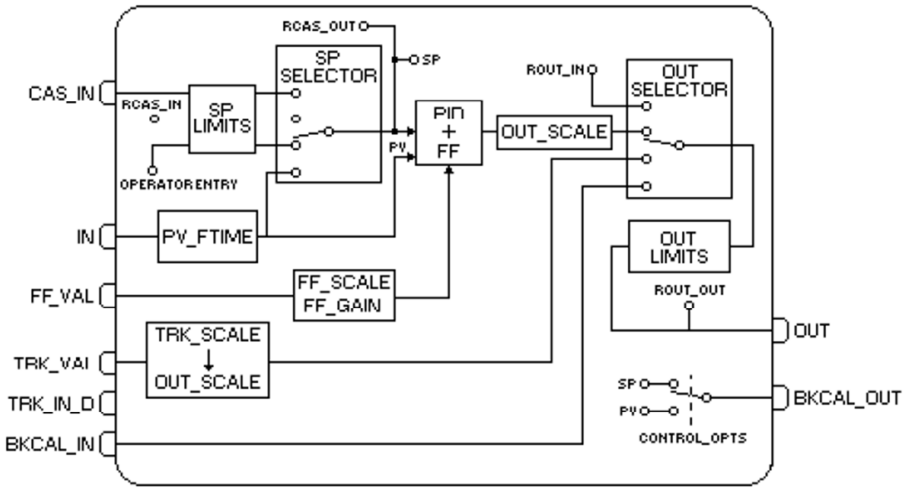
T_d คือ ค่าอนุพันธ์หรือ Rate Time

α คือ ค่าคงที่มีค่าเท่ากับ 0.1

F คือ รูปแบบการควบคุมแบบป้อนไปข้างหน้า

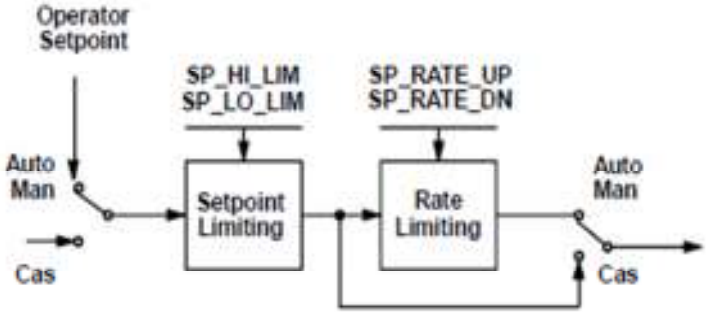
e คือ ค่าผิดพลาดระหว่างค่าตัวแปรกระบวนการกับค่าเป้าหมาย

เพื่อปรับแต่งบล็อกให้ตรงกับความต้องการในการใช้งาน ผู้ใช้สามารถกำหนดค่าการกรองสัญญาณ ค่าอินพุตของการควบคุมแบบป้อนไปข้างหน้า ค่า tracking input ค่าเป้าหมายและขอบเขตของค่าเอาต์พุต โครงสร้างของสมการ PID และการตอบสนองของเอาต์พุตบล็อก รูปที่ 2.23 แสดงตัวอย่างวงจรภายในฟังก์ชันบล็อก PID



รูปที่ 2.23 โครงสร้างภายในฟังก์ชันบล็อก PID

ค่าเป้าหมายของฟังก์ชันบล็อก PID จะถูกกำหนดโดยโหมด ผู้ใช้สามารถกำหนดค่าตัวแปร SP_HI_LIM และ SP_LO_LIM เพื่อกำหนดขอบเขตของค่า SP ในโหมดคาสเคดหรือรีโมตคาสเคด ค่า SP จะถูกปรับโดยฟังก์ชันบล็อกอื่นหรือโดยโฮส และค่าเอาต์พุตจะถูกคำนวณโดยขึ้นอยู่กับค่า SP ในโหมดอัตโนมัติ ค่า SP จะถูกกำหนดโดยผู้ใช้งานโดยตรงและค่าเอาต์พุตจะถูกคำนวณโดยขึ้นอยู่กับค่า SP เช่นเดียวกับในสองโหมดแรก ในโหมดนี้ผู้ใช้งานสามารถปรับเปลี่ยนขอบเขตและอัตราการเปลี่ยนแปลงของค่า SP โดยผ่านทางตัวแปร SP_RATE_UP และ SP_RATE_DN ในโหมดการควบคุมด้วยมือ ค่าเอาต์พุตจะถูกกำหนดโดยการป้อนค่าของผู้ใช้งานและเป็นอิสระจากค่า SP ในโหมดรีโมตเอาต์พุต ค่าเอาต์พุตจะถูกป้อนมาจากโฮสและเป็นอิสระจากค่า SP เช่นเดียวกับในโหมด Manual รูปที่ 2.24 แสดงตัวอย่างหลักการของการเลือกและกำหนดขอบเขตของค่า SP



รูปที่ 2.24 หลักการของการเลือกและกำหนดขอบเขตของค่า SP

ลักษณะต่าง ๆ ที่สำคัญของฟังก์ชันบล็อก PID ซึ่งจะต้องมีการกำหนดค่าให้ถูกต้องเหมาะสม เพื่อให้การควบคุมกระบวนการเป็นไปอย่างมีประสิทธิภาพ มีดังต่อไปนี้

- Bumpless Transfer and Setpoint Tracking

ผู้ใช้สามารถกำหนดการตั้งค่าวิธีการในการ tracking หรือการติดตามค่าเป้าหมาย โดยการกำหนดค่าตัวเลือกในการควบคุม (CONTROL_OPTS) ดังนี้

1.) SP-PV Track in MAN – ฟังก์ชันนี้จะอนุญาตให้ค่าเป้าหมายเปลี่ยนแปลงตามค่าตัวแปรกระบวนการ เมื่อโหมดเป้าหมายของบล็อกถูกเซตเป็น Man

2.) SP-PV Track in LO or IMAN – ฟังก์ชันนี้จะอนุญาตให้ค่าเป้าหมายเปลี่ยนแปลงตามค่าตัวแปรกระบวนการ เมื่อโหมดของบล็อกในขณะนั้นคือ Local Override (LO) หรือ Initialization Manual (IMan)

เมื่อหนึ่งในตัวเลือกเหล่านี้ถูกกำหนด ค่าเป้าหมายจะถูกกำหนดไปยังค่าตัวแปรกระบวนการในขณะที่มีการกำหนดโหมด ซึ่งผู้ใช้งานสามารถเลือกค่าที่อุปกรณ์ควบคุมหลักใช้สำหรับทำการ Tracking ได้โดยการกำหนดให้ใช้ PV ในตัวเลือกการควบคุมแบบ BKCAL_OUT ซึ่งค่า BKCAL_OUT นี้จะ ติดตามค่า PV BKCAL_IN บนตัวควบคุมหลักจะถูกเชื่อมต่อกับ BKCAL_OUT บนฟังก์ชันบล็อก PID ในการควบคุมแบบคาสเคดจะผลักดันให้ค่า OUT ของมันสอดคล่องกับค่า BKCAL_IN ดังนั้นการ Tracking ค่า PV จากบล็อกควบคุมรองไปยังการเชื่อมต่ออินพุตแบบคาสเคดของตัวมัน (CAS_IN) ถ้าการใช้ค่า PV ในตัวเลือก BKCAL_OUT ไม่ถูกกำหนด ค่า SP ในการทำงาน (SP_WRK) จะถูกใช้สำหรับ BKCAL_OUT

ผู้ใช้สามารถจะสามารถเลือกรูปแบบการควบคุมได้ก็ต่อเมื่ออยู่ในโหมด Manual หรือ Out of Service เท่านั้น เมื่อโหมดการทำงานถูกเลือกให้เป็น Auto ค่าเป้าหมายจะคงสถานะสุดท้ายเอาไว้

- PID Equation Structure

เป็นการกำหนดค่าตัวแปร STRUCTURE เพื่อเลือกโครงสร้างสมการ PID โดยผู้ใช้สามารถเลือกหนึ่งในตัวเลือกต่อไปนี้

1.) PI Action on Error, D Action on PV

2.) PID Action on Error

3.) I Action on Error, PD Action on PV

ผู้ใช้สามารถกำหนดค่า RESET ให้เป็นศูนย์เพื่อที่จะกำหนดให้ฟังก์ชันบล็อก PID ดำเนินการควบคุมเพียงหนึ่งเดียวโดยไม่คำนึงถึงการเลือกตัวแปร STRUCTURE เมื่อตัวแปร RESET ถูกกำหนดให้เป็นศูนย์ สมการจะถูกลดรูปไปอยู่ในรูปของสมการการหาร ดังแสดงในสมการที่ 2.12

$$\frac{\text{GAIN x e(s)}}{\text{s}} \qquad (2.12)$$

เมื่อ GAIN คือ ค่าอัตราขยายแบบ P

e คือ ค่าผิดพลาด

s คือ ค่าการดำเนินการของลาปลาซ

- Mode

พีไอดีฟังก์ชันบล็อกรองรับโหมดการทำงานดังต่อไปนี้

1.) Manual (MAN) ในโหมดนี้ค่า OUT จะถูกกำหนดโดยผู้ใช้งานเอง

2.) Automatic (Auto) ในโหมดนี้ค่า SP จะถูกกำหนดโดยผู้ใช้งานและค่าเอาต์พุตจะได้มาจากการคำนวณตามสมการของบล็อก

3.) Cascade (Cas) ในโหมดนี้ค่า SP จะถูกคำนวณมาจากบล็อกอื่นและถูกส่งมายัง ฟังก์ชันบล็อก PID ผ่านทางการเชื่อมต่อไปยัง CAS_IN

4.) RemoteCascade (RCas) ในโหมดนี้ค่า SP จะถูกกำหนดโดยโฮสซึ่งจะเขียนคำสั่งไปยังตัวแปรที่ชื่อว่า RCAS_IN

5.) RemoteOutput (Rout) ในโหมดนี้ค่าเอาต์พุตจะถูกกำหนดโดยโฮสซึ่งจะเขียนคำสั่งไปยังตัวแปรที่ชื่อว่า ROUT_IN

6.) Local Override (LO) ในโหมดนี้ฟังก์ชัน Tracking จะมีการทำงาน ค่าเอาต์พุตจะถูกกำหนดโดยตัวแปรที่ชื่อว่า TRK_VAL และตัวแปรที่ชื่อว่า BLOCK_ERR จะแสดงข้อความ Local Override

7.) Initialization Manual (IMan) โหมดนี้แสดงถึงความไม่สมบูรณ์ของเส้นทางเดินของสัญญาณเอาต์พุต ในโหมดนี้ค่าเอาต์พุตเปลี่ยนแปลงตามตัวแปร BKCAL_IN

8.) Out of Service (O/S) คือบล็อกไม่มีการดำเนินการใด ๆ สถานะของเอาต์พุตจะถูกกำหนดให้เป็น Bad: Out of Service และตัวแปรที่ชื่อว่า BLOCK_ERR จะแสดงข้อความ Out of Service

ผู้ใช้สามารถกำหนดโหมดการทำงานทั้งในแบบ Man, Auto, Cas และ O/S ให้เป็นโหมดที่ได้รับการอนุญาตสำหรับการเข้าดำเนินการของพีไอดีฟังก์ชันบล็อกได้

ในการเพิ่มฟังก์ชันบล็อกมาตรฐานและคำจำกัดความของพฤติกรรมต่าง ๆ ของอุปกรณ์ฟิลด์บัสภายในระบบเพื่อให้สามารถทำงานร่วมกันได้ตามคุณลักษณะความเป็น Interoperability ของเทคโนโลยีฟาวน์เดชันฟิลด์บัส เทคโนโลยีของ Device Description (DD) ได้ถูกนำมาใช้ โดย DD จะเตรียมข้อมูลที่จำเป็นสำหรับระบบควบคุมหรือโฮสให้สามารถเข้าใจความหมายของข้อมูลใน Visual Field Device (VFD) ซึ่งประกอบไปด้วยการเชื่อมต่อกับผู้ใช้งานสำหรับฟังก์ชันต่าง ๆ เช่น

การสอบเทียบ (Calibration) และการวินิจฉัยต่าง ๆ (Diagnostics) ลักษณะของ DDs นี้จะเหมือนกับไดร์ฟเวอร์ (Driver) ของเครื่องคอมพิวเตอร์ที่ใช้ในการทำให้อุปกรณ์ต่าง ๆ ที่ต่อร่วมสามารถทำงานได้ ซึ่ง DD File ของตัวอุปกรณ์ที่ได้รับการลงทะเบียนแล้วนั้น จะสามารถดาวน์โหลดได้จากเว็บไซต์ www.fieldbus.org ดังแสดงในรูปที่ 2.25

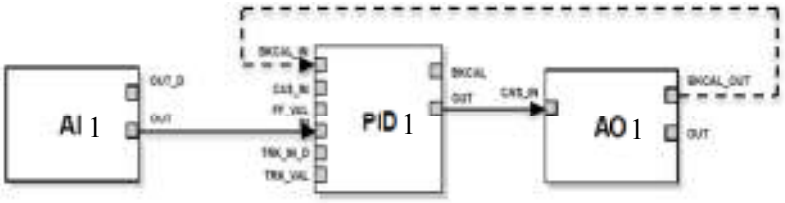
รูปที่ 2.25 ตัวอย่างแหล่งที่มาของ DD File และ CFF

ไฟล์อีกประเภทหนึ่งที่มาพร้อมกับตัวอุปกรณ์ฟิลด์บัสทุกตัวก็คือ Capability Files ซึ่งเป็นไฟล์ที่ทำให้ระบบดีซีเอสรู้ว่าในอุปกรณ์แต่ละตัวมีส่วนประกอบใดบ้างในเทอมของฟังก์ชันบล็อกและ VCRs ซึ่งไฟล์นี้จะทำให้โฮสสามารถกำหนดการตั้งค่าของอุปกรณ์ฟิลด์บัสภายในระบบได้ถึงแม้ว่าจะยังไม่ได้ต่อตัวอุปกรณ์เข้ากับระบบก็ตาม ซึ่งก็คือการกำหนดการตั้งค่าแบบ Offline นั่นเอง

ซึ่ง Capability Files นี้จะเรียกว่า Common File Format (CFF) ซึ่งสามารถดาวน์โหลดได้จากเว็บไซต์ของ Fieldbus Foundation ได้เช่นเดียวกับ DD File

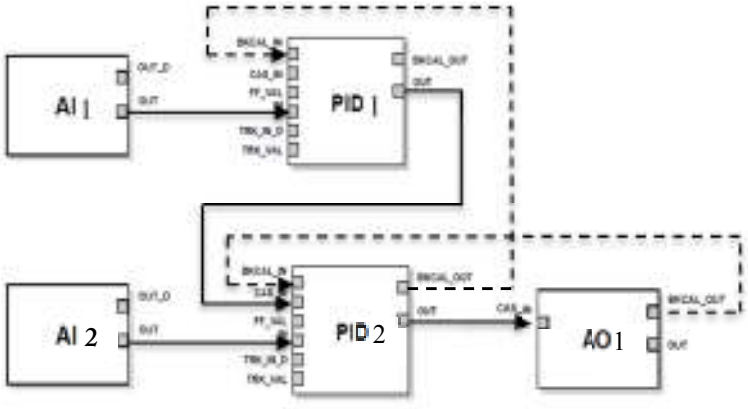
2.5.3 การสร้างวงควบคุมฟิลด์บัส

ในการสร้างวงควบคุมฟิลด์บัส (Creating Fieldbus Control Loop) ที่ใช้ในการควบคุมจะต้องมีการเลือกและเชื่อมต่อฟังก์ชันบล็อกเข้าด้วยกันดังที่ได้กล่าวมาในหัวข้อก่อนหน้านี้ รูปที่ 2.26 แสดงตัวอย่างวิธีการเชื่อมต่อฟังก์ชันบล็อกของการควบคุมแบบ PID หนึ่งวงควบคุม โดยอันดับแรกจะต้องทำการเชื่อมต่อส่วน OUT ของฟังก์ชันบล็อก AI ไปยังส่วน IN ของฟังก์ชันบล็อก PID จากนั้นเชื่อมต่อส่วน OUT ของฟังก์ชันบล็อก PID เข้ากับส่วน CAS_IN ของฟังก์ชันบล็อก AO ซึ่งจะเป็นการทำให้ค่าเอาต์พุตของฟังก์ชันบล็อก PID สามารถส่งค่าไปควบคุมอุปกรณ์สุดท้ายได้ นอกจากนี้ยังมีการเชื่อมต่อส่วน BKCAL_OUT จากฟังก์ชันบล็อก AO ไปยังส่วน BKCAL_IN ภายในฟังก์ชันบล็อก PID เพื่อนำไปคำนวณหาค่า error ซึ่งจะใช้ในการพิจารณาปรับปรุงกระบวนการควบคุมของฟังก์ชันบล็อก PID นั่นเอง



รูปที่ 2.26 การสร้างวงควบคุมฟิลด์บัสสำหรับการควบคุมแบบ PID

ในทำนองเดียวกันการเชื่อมต่อฟังก์ชันบล็อกของการควบคุมแบบคาสเคดหนึ่งวงควบคุมในรูปที่ 2.27 จะมีขั้นตอนในการเชื่อมต่อฟังก์ชันบล็อกเหมือนกับในกรณีของการควบคุมแบบ PID แต่จะมีการเพิ่มฟังก์ชันบล็อก PID และฟังก์ชันบล็อก AI ของวงควบคุมรองซึ่งใช้ค่า PV ของวงควบคุมหลักเป็นค่า SP ในการควบคุม ดังนั้นจึงต้องมีการเพิ่มการเชื่อมต่อส่วน OUT ของ ฟังก์ชันบล็อก PID1 ไปยังส่วน CAS_IN ของฟังก์ชันบล็อก PID2 เพื่อดำเนินการควบคุมแบบคาสเคดนั่นเอง



รูปที่ 2.27 การสร้างวงควบคุมฟิลด์บัสสำหรับการควบคุมแบบ PID

ในการเลือกตำแหน่งของฟังก์ชันบล็อก AI และ PID ในวงควบคุมสำหรับการควบคุมแบบ Control in the Field นั้น วงควบคุมแบบ PID ฟังก์ชันบล็อก AI จะถูกกำหนดให้อยู่ภายในตัวทรานสมิตเตอร์ และฟังก์ชันบล็อก PID จะถูกกำหนดให้อยู่ภายในวาล์วควบคุม ในทำนองเดียวกันสำหรับวงควบคุมแบบคาสเคด ฟังก์ชันบล็อก AI จะถูกกำหนดให้อยู่ภายในทรานสมิตเตอร์ ส่วนฟังก์ชันบล็อก PID ที่ทำหน้าที่เป็นตัวควบคุมหลักและตัวควบคุมรอง จะถูกกำหนดให้อยู่ที่ทรานสมิตเตอร์ของเครื่องวัดระดับ และอยู่ที่วาล์วควบคุมตามลำดับ ทั้งนี้เนื่องมาจากตำแหน่งของการจัดวางฟังก์ชันบล็อกในลักษณะดังที่กล่าวมานี้ จะให้ค่ามาโครไซเคิลที่สั้นที่สุดนั่นเอง [2]

2.5.4 การบริหารจัดการความเสี่ยงของฟาวน์เดชันฟิลด์บัสเซกเมนต์ [10]

การจัดอันดับของการประเมินความเสี่ยง (Risk) สำหรับในแต่ละเซกเมนต์จะต้องถูกบันทึกลงในแผนการประเมินความเสี่ยง การออกแบบโครงสร้างของระบบจะต้องคำนึงถึงการลดจุดเสี่ยงในการเกิดความผิดพลาด หลักการจัดลำดับความเสี่ยงของวาล์วและภาระในการทำงานของเซกเมนต์ควรจะถูกนำมาประยุกต์ใช้ วาล์วควบคุมและเครื่องมือวัดที่เกี่ยวข้องซึ่งมีความสำคัญต่อกระบวนการจะต้องได้รับการกำหนดไว้ให้เป็นภาระงานของฟิลด์บัสเซกเมนต์

ฟาวน์เดชันฟิลด์บัสเซกเมนต์ควรจะจัดแบ่งออกตามอันดับความเสี่ยงซึ่งมีอยู่ด้วยกันทั้งหมด 3 ลำดับ โดยขึ้นอยู่กับผลกระทบที่จะเกิดขึ้นเมื่อตัวอุปกรณ์หรือเซกเมนต์เกิดความผิดพลาดในระหว่างการทำงาน การจัดอันดับต่อไปนี้จะถูกกำหนดลงไปยังวาล์วแต่ละตัวและในแต่ละเซกเมนต์

1. เซกเมนต์และอุปกรณ์ควบคุมสุดท้ายระดับ 1

ความผิดพลาดของเซกเมนต์หรือวาล์วควบคุมจะส่งผลให้ระบบหยุดทำงานในทันทีทันใด ซึ่งเป็นสาเหตุทำให้กระบวนการถูกปิดลง (Shutdown) ส่งผลให้กระบวนการผลิตเกิดความสูญเสีย สำหรับในกลุ่มนี้วาล์วจะกลับเข้าสู่สถานะปกติเมื่อเกิดความผิดพลาดในการควบคุม

ข้อจำกัดในการออกแบบเซกเมนต์ระดับ 1 คือวาล์วควบคุมและอุปกรณ์เครื่องมือวัดที่เกี่ยวข้องจะต้องอยู่ภายในเครือข่ายซึ่งใช้เฉพาะสำหรับการควบคุมระดับ 1 และภายในเซกเมนต์จะประกอบด้วยวาล์วควบคุมและทรานสมิตเตอร์อย่างละ 1 ตัวเท่านั้น ในส่วนของการ์ดที่ใช้เชื่อมต่อระหว่างโฮสกับอุปกรณ์ในวงควบคุมระดับ 1 จะต้องได้รับการทดสอบการทำงานร่วมกันโดยอิสระ แผนภาพของเซกเมนต์จะต้องแสดงระดับความเสี่ยงและจะต้องมีการแสดงสัญลักษณ์ให้เห็นอย่างชัดเจนว่าห้ามมีการเพิ่มอุปกรณ์ลงในเซกเมนต์นี้

2. เซกเมนต์และอุปกรณ์ควบคุมสุดท้ายระดับ 2

ความผิดพลาดของเซกเมนต์หรือวาล์วควบคุมไม่ส่งผลให้เกิดความเสี่ยงในระยะสั้นๆของกระบวนการจนกระทั่งทำให้เกิดการปิดตัวลงหรือสูญเสียการทำงาน วาล์วระดับ 2 สามารถเปลี่ยนไปยังตำแหน่งผิดพลาดของมันโดยไม่ต้องการการทำงานแบบทันทีทันใด

ข้อจำกัดในการออกแบบเซกเมนต์ระดับ 2 คือวาล์วควบคุมสามารถกำหนดให้อยู่ภายในการ์ดหรือเซกเมนต์ที่มีวาล์วควบคุมระดับ 2 ตัวอื่น ๆ อยู่ด้วยได้ เครือข่ายที่มีการควบคุมในระดับ 2 สามารถใช้อุปกรณ์ที่มาจากต่างบริษัทผู้ผลิตซึ่งประกอบด้วยเครื่องมือวัดในระดับ 3 ได้ อุปกรณ์และการ์ดเชื่อมต่อจะถูกเก็บรักษาไว้ในระดับที่แก้ไขได้ตามอายุการใช้งานของเครือข่าย แผนภาพเครือข่ายหรือกระบวนการจะต้องแสดงระดับความอันตรายให้เห็นชัดเจน

3. เซกเมนต์ระดับ 3

อุปกรณ์ระดับ 3 จะเป็นอุปกรณ์ที่ทำหน้าที่ในการตรวจสอบกระบวนการเท่านั้น จะไม่สามารถนำมาใช้ในการควบคุมหรือทำฟังก์ชันที่มีการประสานงานกัน (Interlocking) ได้

ข้อจำกัดในการออกแบบเซกเมนต์ระดับ 3 คืออุปกรณ์ที่อยู่ในระดับนี้สามารถกำหนดให้อยู่ภายในเซกเมนต์ได้เต็มจำนวนตามข้อจำกัดของแบนด์วิธ (bandwidth) ของเครือข่าย และข้อจำกัดทางกายภาพ อุปกรณ์และการ์ดเชื่อมต่อจะถูกเก็บรักษาไว้ในระดับที่แก้ไขได้ตามอายุการใช้งานของเครือข่าย แผนภาพเครือข่ายหรือกระบวนการจะต้องแสดงระดับความเสี่ยงให้เห็นชัดเจน

4. การควบคุมแบบ PID บนวงควบคุมวงเดี่ยว

สำหรับการควบคุมแบบวงเดี่ยว ฟังก์ชันบล็อกทั้งหมดที่อยู่ภายในวงควบคุมจะต้องถูกกำหนดให้อยู่ภายในเซกเมนต์เดียวกัน

5. การควบคุมแบบ PID บนวงควบคุมคาสเคด

สำหรับการควบคุมแบบคาสเคดในกรณีที่เซกเมนต์ถูกกำหนดให้อยู่ในระดับที่ 2 วงควบคุมหลักและวงควบคุมรองจะต้องถูกกำหนดให้อยู่ภายในเซกเมนต์เดียวกัน และในกรณีที่เซกเมนต์ถูกกำหนดให้อยู่ในระดับที่ 1 อุปกรณ์เครื่องมือวัดและฟังก์ชันบล็อกที่อยู่ภายในวงควบคุมหลักและวงควบคุมรองจะต้องถูกกำหนดให้อยู่คนละเซกเมนต์

6. สรุปการจัดอุปกรณ์ฟาวน์เดชันฟิลด์บัสไปยังเซกเมนต์

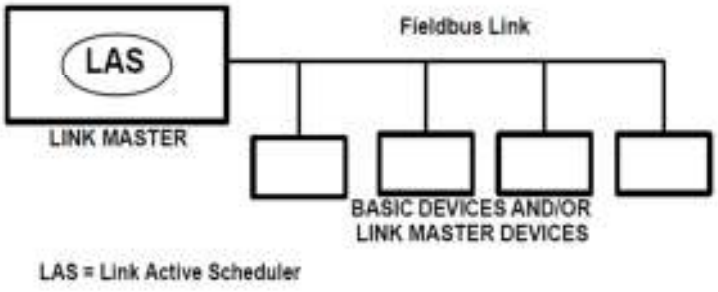
ข้อสรุปต่อไปนี้เป็นข้อจำกัดที่สามารถนำไปใช้เมื่อมีการจัดอุปกรณ์ฟาวน์เดชันฟิลด์บัสไปยังเซกเมนต์ ดังนี้

- เซกเมนต์ระดับ 1 สามารถมีวาล์วควบคุมภายในเซกเมนต์ได้เพียง 1 ตัวเท่านั้น
- เซกเมนต์ระดับ 2 สามารถมีวาล์วควบคุมภายในเซกเมนต์ได้เพียง 2 ตัวเท่านั้น
- เซกเมนต์ระดับ 2 สามารถมีอุปกรณ์ฟาวน์เดชันฟิลด์บัสภายในเซกเมนต์ได้ไม่เกิน 8 ตัวในกรณีที่มีวาล์วควบคุมต่อร่วมด้วย
- ในเซกเมนต์ระดับ 3 ซึ่งใช้สำหรับการตรวจสอบเท่านั้น สามารถมีอุปกรณ์ฟาวน์เดชันฟิลด์บัสภายในเซกเมนต์ได้ไม่เกิน 12 ตัว
- ในแต่ละเซกเมนต์สามารถมีอุปกรณ์ DVC ได้ไม่เกิน 6 ตัว เนื่องจากข้อจำกัดในเรื่องภาระงานของเซกเมนต์
- ความยาวของสายสัญญาณจากบล็อกเชื่อมต่อถึงอุปกรณ์ฟาวน์เดชันฟิลด์บัสที่อยู่ภายในกระบวนการจะต้องยาวไม่เกิน 60 เมตร

หลักเกณฑ์เหล่านี้สามารถนำมาใช้ได้กับทั้งระบบที่เป็น Intrinsically Safe (I.S) และแบบ Non-Intrinsically Safe (non-I.S)

2.5.5 การสื่อสารข้อมูลบนเครือข่าย

รูปที่ 2.28 แสดงตัวอย่างของโครงข่ายฟิลด์บัสโดยทั่วไป ซึ่งประกอบไปด้วยเซกเมนต์เพียงหนึ่งเซกเมนต์ดังรูป



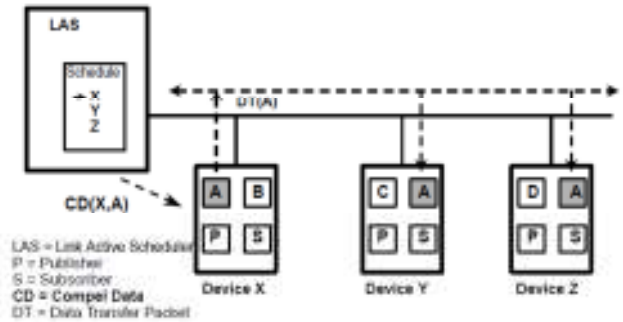
รูปที่ 2.28 โครงข่ายการสื่อสารพื้นฐานของฟิลด์บัส

อุปกรณ์ฟาวน์เดชันฟิลด์บัสที่อยู่ภายในแต่ละเซกเมนต์มีการติดต่อสื่อสารแลกเปลี่ยนข้อมูลกันผ่านทางฟิลด์บัสลิงค์ ซึ่งรูปแบบหรือลำดับในการสื่อสารสามารถแบ่งออกได้เป็นหลายลักษณะดังนี้

2.5.5.1 Scheduled Communications

การรับ-ส่งข้อมูลแบบ Scheduled Communications นี้ ข้อมูลต่าง ๆ จะถูกรับ-ส่งระหว่างตัวอุปกรณ์กับฟิลด์บัสโดยใช้รูปแบบการติดต่อสื่อสารที่เรียกว่า Publisher/Subscriber รูปแบบการสื่อสารชนิดนี้ถูกใช้ในการส่งข้อมูลที่มีความสำคัญในกระบวนการ เช่นตัวแปรกระบวนการ เมื่อ Publisher ได้รับ Compel Data จะทำการประกาศข้อมูลลงในตัวบัฟเฟอร์ (Buffer) ซึ่งจะถูกส่งต่อไปยัง Subscriber ตัวบัฟเฟอร์จะประกอบไปด้วยข้อมูลที่ถูกทำการคัดลอกไว้เพียงข้อมูลเดียวเท่านั้น ข้อมูลใหม่ที่เข้ามาจะถูกเขียนทับลงบนข้อมูลก่อนหน้า การรับส่งข้อมูลในลักษณะนี้จะสามารถกำหนดตารางเวลาในการทำงานได้อย่างแม่นยำ

รูปที่ 2.29 แสดงแผนภาพของการรับ-ส่งข้อมูลแบบ Scheduled Communications ซึ่งการรับ-ส่งข้อมูลแบบนี้จะใช้สำหรับวงจรการส่งข้อมูลปกติของข้อมูลกระบวนการระหว่างอุปกรณ์บนฟิลด์บัส LAS (Link Active Scheduler) ซึ่งทำหน้าที่เปรียบเสมือนผู้ควบคุมตารางเวลาในการทำงานของอุปกรณ์ภายในเซกเมนต์ ประกอบไปด้วยลิสต์เวลาในการส่งข้อมูลสำหรับทุก Publisher ในทุกตัวอุปกรณ์ซึ่งต้องการส่งข้อมูลไปยังบัส เมื่อถึงเวลาที่ตัวอุปกรณ์จะต้องส่งข้อมูล LAS จะส่งข้อความ Compel Data (CD) ไปยังตัวอุปกรณ์ เมื่อได้รับ CD แล้ว อุปกรณ์ฟิลด์บัสจะประกาศข้อมูลไปยังอุปกรณ์ทุกตัวที่อยู่บนบัส ซึ่งอุปกรณ์ที่ถูกกำหนดให้เป็นตัวรับข้อมูลเหล่านั้นจะเรียกว่า Subscriber



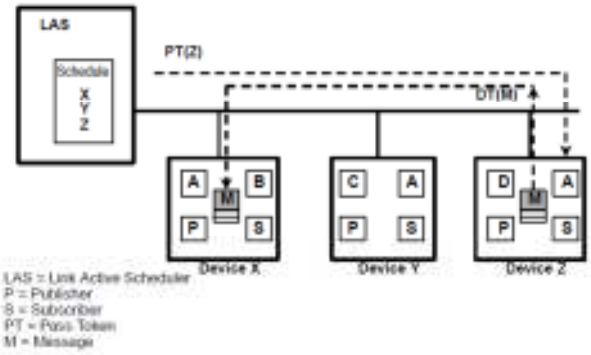
รูปที่ 2.29 การรับ-ส่งข้อมูลแบบ Scheduled Communications

2.5.5.2 Unscheduled Communications

การรับ-ส่งข้อมูลแบบ Unscheduled หรือการสื่อสารที่ไม่มีกำหนดเวลาแน่นอน จะใช้สำหรับการรับ-ส่งข้อมูลประเภทที่เป็นการเปลี่ยนแปลงค่าตัวแปรต่าง ๆ โดยผู้ใช้งาน ประกอบไปด้วย การเปลี่ยนค่า SP การเปลี่ยนโหมดการทำงาน การปรับค่าพารามิเตอร์ และการอัพโหลด (Upload) หรือดาวน์โหลด (Download) การสื่อสารข้อมูลแบบ Unscheduled นี้จะใช้รูปแบบการสื่อสารทั้งแบบ Report Distribution หรือ Client/ Server ซึ่งมีลักษณะดังนี้

1.) Report Distribution รูปแบบการสื่อสารข้อมูลชนิดนี้ จะถูกใช้ในการรับและส่งรายงานเหตุการณ์ต่าง ๆ และแนวโน้มของกระบวนการ ที่อยู่ปลายทางที่จะทำการส่งข้อมูลไปจะถูกกำหนดไว้ล่วงหน้า ดังนั้นข้อมูลรายงานทั้งหมดจะถูกส่งไปยังที่อยู่เดียวกันหรืออาจจะจัดไว้แยกออกจากกันในแต่ละรายงานก็ได้ การส่งข้อมูลชนิดนี้จะเป็นไปในลักษณะของการเรียงแถว (Queue) ซึ่งข้อมูลจะถูกส่งไปยังตัวรับที่อยู่ในลำดับของการส่ง ถึงแม้ว่าอาจจะเกิดช่องว่างซึ่งเกิดจากความเสียหายของข้อมูลในระหว่างถ่ายโอนก็ตาม การถ่ายโอนข้อมูลชนิดนี้ไม่มีกำหนดเวลาที่แน่นอนและจะเกิดขึ้นในช่วงที่มีการการถ่ายโอนข้อมูลที่มีการกำหนดเวลาแน่นอนโดยเรียงตามลำดับความสำคัญ

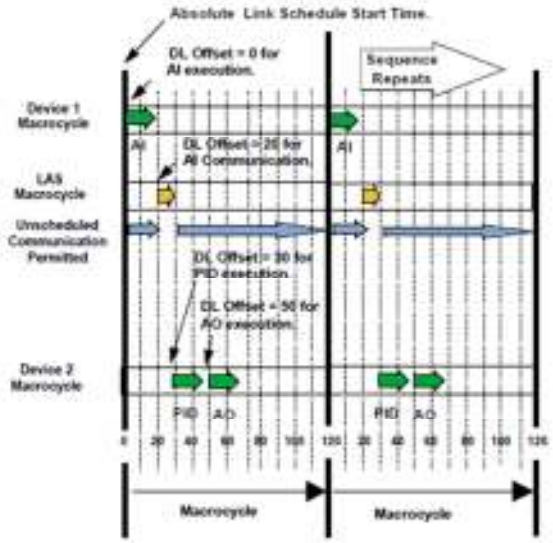
2.) Client/ Server รูปแบบของการสื่อสารข้อมูลชนิดนี้ถูกใช้สำหรับการร้องขอหรือตอบสนองการแลกเปลี่ยนข้อมูลระหว่างคู่ของอุปกรณ์ เช่นเดียวกับการสื่อสารในรูปแบบของ Report Distribution การสื่อสารจะเป็นไปในลักษณะของการเรียงแถว (Queue) ไม่มีกำหนดเวลาที่แน่นอนและจะเรียงตามลำดับความสำคัญ Queue ในที่นี้หมายถึงการที่ข้อความถูกส่งและรับตามลำดับในการส่งข้อมูลซึ่งถูกกำหนดโดยการจัดลำดับความสำคัญของข้อมูล และจะไม่มีการเขียนข้อมูลทับลงบนข้อความก่อนหน้า อย่างไรก็ตามสิ่งที่แตกต่างจาก Report Distribution ก็คือการส่งข้อมูลในลักษณะนี้จะถูกควบคุมการส่งและมีขั้นตอนในการส่งข้อมูลใหม่เพื่อกู้คืนความเสียหายที่อาจเกิดขึ้น แผนภาพการรับ-ส่งข้อมูลแบบ Unscheduled Communications แสดงดังรูปที่ 2.30



รูปที่ 2.30 การรับ-ส่งข้อมูลแบบ Unscheduled Communications

2.5.5.3 การจัดตารางมาโครไซเคิล

มาโครไซเคิล คือช่วงเวลาในการดำเนินการของฟังก์ชันบล็อกทั้งหมดที่อยู่ภายในเซกเมนต์แต่ละเซกเมนต์ รูปที่ 2.31 แสดงตัวอย่างของ Link Schedule ซึ่งเมื่อระบบถูกสร้างขึ้นและฟังก์ชันบล็อกถูกเชื่อมต่อเข้าด้วยกัน ตารางเวลาในการสื่อสารหลักจะถูกสร้างขึ้นสำหรับ LAS ซึ่งจะทำหน้าที่เป็นตัวกำหนดลำดับเวลาในการทำงานให้กับอุปกรณ์ที่อยู่บนระบบ อุปกรณ์แต่ละตัวจะรักษาส่วนของตัวมันเองภายในตารางเวลา หรือที่เรียกว่า Function Block Scheduled ซึ่งจะแสดงให้เห็นเมื่อฟังก์ชันบล็อกสำหรับตัวอุปกรณ์มีการดำเนินการ เวลาในการดำเนินการของแต่ละฟังก์ชันบล็อกถูกแทนด้วยค่าออฟเซต (offset) จากตำแหน่งเวลาเริ่มต้นของมาโครไซเคิล



รูปที่ 2.31 ตัวอย่างการจัดมาโครไซเคิล

จากรูปที่ 2.31 จะเห็นว่าที่จุดเริ่มต้นออฟเซต 0 ฟังก์ชันบล็อก AI ที่อยู่ภายในเครื่องมือวัดตัวแปรกระบวนการ จะทำการประมวลผลหาค่าตัวแปรที่วัดได้จากกระบวนการ จากนั้นที่ออฟเซต 20 ตัวควบคุมเวลาการทำงานของระบบหรือ LAS จะกำหนดให้ข้อมูลอินพุตที่ได้จากการประมวลผล AI ส่งข้อมูลออกไปยังระบบผ่านเครือข่ายหลัก ที่ออฟเซต 30 ฟังก์ชันบล็อก PID ที่อยู่ใน

วาล์วควบคุมทำการประมวลผลจากข้อมูลอินพุตที่ได้รับ และจากนั้นฟังก์ชันบล็อก AO ที่อยู่ในวาล์วควบคุมเช่นเดียวกันจะทำการประมวลผลที่ออฟเซต 50 ซึ่งลำดับการประมวลผลของอุปกรณ์เหล่านี้จะมีตารางเวลาที่แน่นอน เพื่อทำให้การควบคุมมีการทำงานถูกต้องและมีความน่าเชื่อถือ ในช่วงเวลาที่ฟังก์ชันบล็อกกำลังประมวลผลนั้น LAS จะอนุญาตให้ข้อมูลที่เป็นแบบ Unscheduled Communications สามารถส่งข้อมูลเข้าไปยังระบบผ่านทางเครือข่ายหลักได้ ซึ่งวิธีการก็คือ ในระหว่างที่ฟังก์ชันบล็อกกำลังดำเนินการ LAS จะส่ง Pass Token Message ไปยังอุปกรณ์ทุกตัวที่อยู่บนบัส ทำให้อุปกรณ์เหล่านั้นสามารถส่งข้อมูลที่เป็น Unscheduled Messages เช่นการแจ้งเตือนต่าง ๆ หรือการเปลี่ยนแปลงค่า SP ออกมาบนบัสได้ จากในตัวอย่างนี้ ช่วงเวลาที่ฟิลด์บัสไม่สามารถใช้สำหรับส่งข้อมูลที่เป็น Unscheduled Messages ได้อยู่ในช่วงออฟเซต 20 ถึงออฟเซต 30 หรือเมื่อข้อมูลจากฟังก์ชันบล็อก AI เริ่ม Publish ไปยังฟิลด์บัสนั่นเอง

- เวลาในการดำเนินการของเซกเมนต์ (Segment Execution Time) [3]

ค่าเริ่มต้นของเวลาในการดำเนินการสำหรับทุกเซกเมนต์ควรจะมีค่ามากที่สุดไม่เกิน 1 วินาที ค่าเวลาในการดำเนินการที่ต้องการของแต่ละเซกเมนต์ควรจะมีการตรวจสอบในระหว่างขั้นตอนการออกแบบเพื่อที่จะทำให้มั่นใจได้ว่าการทำงานของแต่ละเซกเมนต์มีความสอดคล้องกับฟังก์ชันการควบคุมกระบวนการที่ต้องการในแต่ละวงควบคุม

ในกรณีที่วงควบคุมต้องการเวลาในการตอบสนองของระบบเร็วกว่า 1 วินาที ผู้ออกแบบควรจะปฏิบัติตามแนวทางดังต่อไปนี้

1.) สำหรับเซกเมนต์ที่ใช้สำหรับอ่านค่าเพียงอย่างเดียว อุปกรณ์ที่อยู่ภายในเซกเมนต์จะสามารถมีได้ไม่เกิน 12 ตัว

2.) วงควบคุมที่ต้องการเวลาในการดำเนินการที่ 1 วินาที จะจำกัดจำนวนอุปกรณ์ให้มีได้ไม่เกิน 8 ตัวต่อเซกเมนต์

3.) วงควบคุมที่ต้องการเวลาในการดำเนินการที่ 0.5 วินาที จะจำกัดจำนวนอุปกรณ์ให้มีได้ 4 - 8 ตัวต่อเซกเมนต์

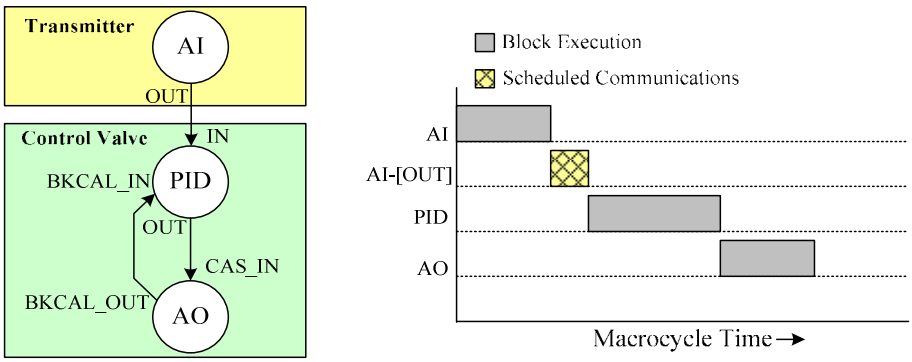
4.) วงควบคุมที่ต้องการเวลาในการดำเนินการที่ 0.25 วินาที จะจำกัดจำนวนอุปกรณ์ให้มีได้ไม่เกิน 3 ตัวต่อเซกเมนต์และให้มีอุปกรณ์ควบคุมสุดท้ายได้มากที่สุดเพียงหนึ่งตัวเท่านั้น

ความถี่ในการดำเนินการของฟังก์ชันบล็อกจะต้องสอดคล้องกันกับทั้งภาระงานของระบบและจุดประสงค์ของการควบคุมกระบวนการ และมาโครไซเคิลควรจะมีช่วงเวลาว่างจากการดำเนินการสำหรับการส่งข้อความที่เป็น Unscheduled Massages อย่างน้อย 60%

- การจัดตารางเวลาของเซกเมนต์

ลำดับการทำงานของฟังก์ชันบล็อกจะถูกกำหนดโดยอัตโนมัติ ขึ้นอยู่กับการเชื่อมต่อระหว่างฟังก์ชันบล็อกบนฟิลด์บัสเซกเมนต์ การคำนวณพื้นที่ว่างสำหรับการสื่อสารแบบ Unscheduled Communications บนบัสขึ้นอยู่กับเวลาที่ต้องการสำหรับการติดต่อสื่อสารและเวลาในการดำเนินการของแต่ละบล็อก

- ตัวอย่างการคำนวณมาโครไซเคิล



(a) PID Control in the Field (b) Macrocycle Schedule

รูปที่ 2.32 มาโครไซเคิลของการควบคุมแบบ Control in the Field ซึ่งประกอบด้วยวงควบคุมแบบ PID หนึ่งวง

รูปที่ 2.32 แสดงการควบคุมกระบวนการโดยการใช้ฟังก์ชันบล็อก PID ที่อยู่ภายในอุปกรณ์เครื่องวัด โดยรูปที่ 2.32 (a) แสดงการกำหนดการเชื่อมต่อฟังก์ชันบล็อกของวงควบคุม และรูปที่ 2.32 (b) แสดงถึงมาโครไซเคิลสำหรับวงควบคุมซึ่งประกอบด้วยฟังก์ชันบล็อก AI ฟังก์ชันบล็อก PID และฟังก์ชันบล็อก AO แต่ละฟังก์ชันบล็อกในวงควบคุมถูกดำเนินการที่ความเร็วที่ถูกระบุอยู่ภายในอุปกรณ์ฟิลด์บัส เวลาของแต่ละมาโครไซเคิลสามารถประมาณได้ในระหว่างขั้นตอนการกำหนดการรวมเข้าด้วยกันของฟังก์ชันบล็อกภายหลังจากที่มีการกำหนดฟิลด์บัสเซกเมนต์ เช่นจากตัวอย่างวงควบคุมในรูปที่ 2.32 สมมติให้ฟังก์ชันบล็อก AI ใช้เวลาในการดำเนินการเท่ากับ 75 มิลลิวินาที และใช้เวลาในการ Publish ข้อมูลไปบนเครือข่าย (AI-[OUT]) เท่ากับ 30 มิลลิวินาที ฟังก์ชันบล็อก PID ใช้เวลาในการดำเนินการเท่ากับ 130 มิลลิวินาที และฟังก์ชันบล็อก AO ใช้เวลาในการดำเนินการภายเท่ากับ 75 มิลลิวินาทีตามลำดับ ดังนั้นค่าเวลาของมาโครไซเคิลสำหรับวงควบคุมนี้จะมีค่าเท่ากับ 75+30+130+75 = 310 มิลลิวินาที อย่างไรก็ตามการคำนวณมาโครไซเคิลขึ้นอยู่กับหลากหลายตัวแปรในการตั้งค่าระบบที่จะต้องนำมาพิจารณา เช่นจำนวนและประเภทของอุปกรณ์ฟิลด์บัส เวลาในการดำเนินการของแต่ละฟังก์ชันบล็อก ลำดับการทำงานของฟังก์ชันบล็อก และเวลาที่ต้องการในการรับ-ข้อมูลระหว่างบล็อกเป็นต้น

บทที่ 3
การจัดตารางเวลาในการสื่อสารของ ฟาวน์เดชันฟิลด์บัสเซกเมนต์

3.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ที่ประกอบด้วยหนึ่งและสองวงควบคุม โดยใช้แนวคิด Control in the Field เปรียบเทียบกับการควบคุมในระบบดีซีเอส โดยมีการควบคุมแบบ PID และการควบคุมแบบคาสเคดเป็นกรณีศึกษาผ่านทางพลานต์โมเดล 2 ชุด ตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ซึ่งต่อจากนี้จะถูกแทนด้วยคำว่า “มาโครไซเคิล” เพื่อความกระชับในการอ้างถึง จะถูกสร้างขึ้นโดยระบบดีซีเอส Centum VP และ DeltaV ตามลำดับ นอกจากนี้ยังมีการตรวจสอบผลการศึกษาการจัดมาโครไซเคิลที่ได้จากการกำหนดตั้งค่าแบบ Offline โดยการใช้ NI-FBUS Monitor ในการตรวจสอบการรับ-ส่งข้อมูลของ Schedule Communications จากการทำงานจริงอีกด้วย

3.2 พลานต์โมเดลที่ใช้เป็นกรณีศึกษา

วิทยานิพนธ์ฉบับนี้ใช้พลานต์โมเดลทั้งหมด 2 ชุดที่ประกอบไปด้วยอุปกรณ์ฟาวน์เดชันฟิลด์บัสจากหลากหลายบริษัทผู้ผลิตทำงานร่วมกัน เพื่อศึกษาการจัดมาโครไซเคิลและ Network Load ของระบบ ดังแสดงในรูปที่ 3.1 และ 3.4 ตามลำดับ

3.2.1 พลานต์โมเดล 1

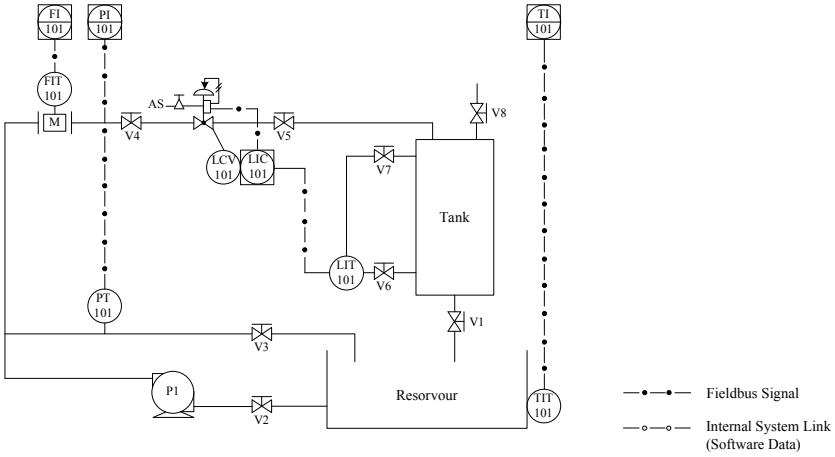
พลานต์โมเดล 1 เป็นกระบวนการควบคุมระดับน้ำในถังเปิด ประกอบด้วยอุปกรณ์เครื่องมือวัดได้แก่ เครื่องมือวัดระดับ (Level Transmitter) ทำหน้าที่ในการวัดระดับของน้ำในถัง เครื่องมือวัดอัตราการไหล (Flow Transmitter) ทำหน้าที่วัดอัตราการไหลของน้ำที่จะเข้าสู่ถัง เครื่องมือวัดอุณหภูมิ (Temperature Transmitter) ทำหน้าที่วัดอุณหภูมิของน้ำภายในกระบวนการ และวาล์วควบคุม (Control Valve) ทำหน้าที่ในการควบคุมปริมาณการไหลเข้าของน้ำในถังให้ได้ระดับตามที่กำหนด คุณสมบัติเฉพาะของอุปกรณ์ฟิลด์บัสแต่ละตัวภายในพลานต์แสดงดังตารางที่ 3.1

รูปที่ 3.1 พลานต์โมเดล 1

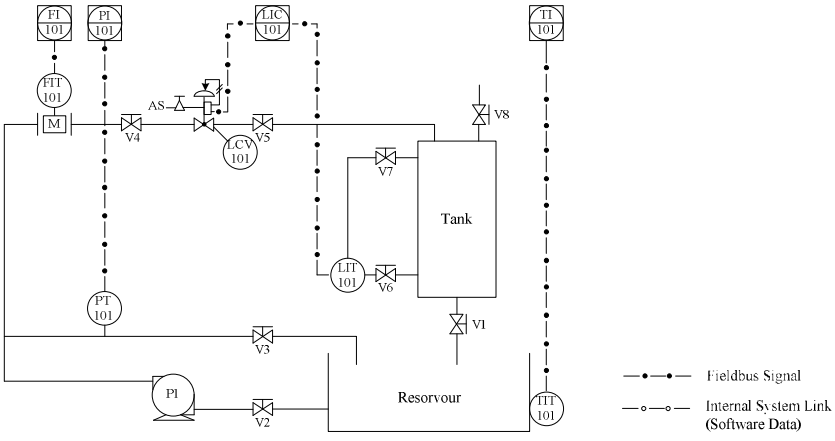
ตารางที่ 3.1 รายละเอียดของอุปกรณ์ฟาวน์เดชันฟิลด์บัสที่อยู่ภายในพลานต์โมเดล 1

Tag	Model	Rev.	AI count	AI time	AO count	AO time	PID count	PID time
TIT_A101	3144P	2	3	60 ms	N/A	N/A	2	90 ms
LCV_A101	AVP303	2	N/A	N/A	1	75 ms	2	125 ms
FIT_A101	8732E	2	1	10 ms	N/A	N/A	1	10 ms
LIT_A101	EJX110A	3	3	30 ms	N/A	N/A	1	45 ms

จากพลานต์โมเดลในรูปที่ 3.1 สามารถนำมาเขียนในลักษณะของแผนภาพกระบวนการผลิต (Piping and Instrument Diagram : P&ID) ได้ดังรูปที่ 3.2 และ 3.3 ตามลำดับ โดยในรูปที่ 3.2 นั้นเป็น P&ID ของพลานต์โมเดล 1 ในกรณีที่กำหนดให้ตัวควบคุมอยู่ภายในตัวอุปกรณ์หรือ Control in the Field ส่วนรูปที่ 3.3 เป็น P&ID ของพลานต์โมเดล 1 ในกรณีที่ควบคุมด้วยระบบดีซีเอสหรือ Control in the Host



รูปที่ 3.2 P&ID ของพลานต์โมเดล 1 ในกรณี Control in the Field



รูปที่ 3.3 P&ID ของพลานต์โมเดล 1 ในกรณีควบคุมด้วยระบบดีซีเอส

3.2.2 พลานต์โมเดล 2

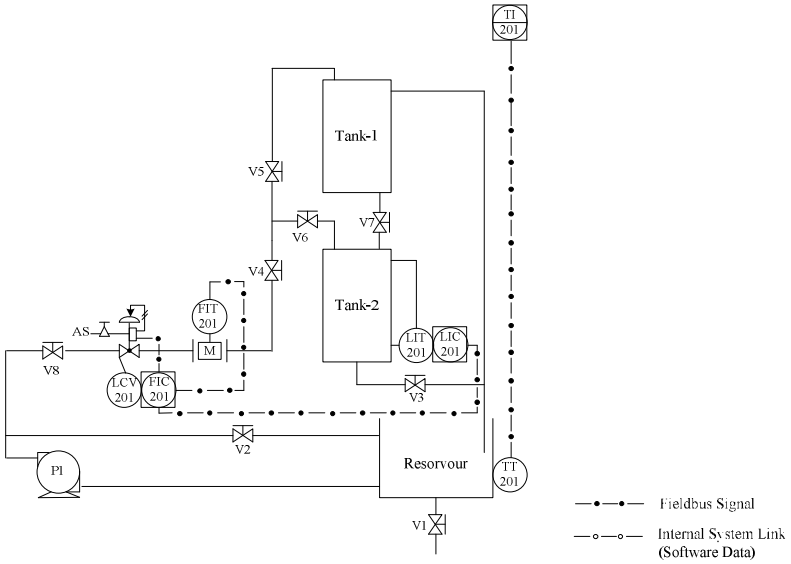
พลานต์โมเดล 2 เป็นกระบวนการควบคุมระดับน้ำในถังปิด ประกอบด้วยอุปกรณ์เครื่องมือวัด ได้แก่ เครื่องมือวัดระดับ ทำหน้าที่ในการวัดระดับของน้ำในถัง เครื่องมือวัดอัตราการไหล ทำหน้าที่ในการวัดอัตราการไหลของน้ำที่จะเข้าสู่ถัง เครื่องมือวัดอุณหภูมิ ทำหน้าที่ในการวัดอุณหภูมิของน้ำในกระบวนการ และวาล์วควบคุม ทำหน้าที่ในการควบคุมปริมาณการไหลเข้าของน้ำในถัง คุณสมบัติเฉพาะของอุปกรณ์ฟิลด์บัสแต่ละตัวภายในพลานต์แสดงดังตารางที่ 3.2

รูปที่ 3.4 พลานต์โมเดล 2

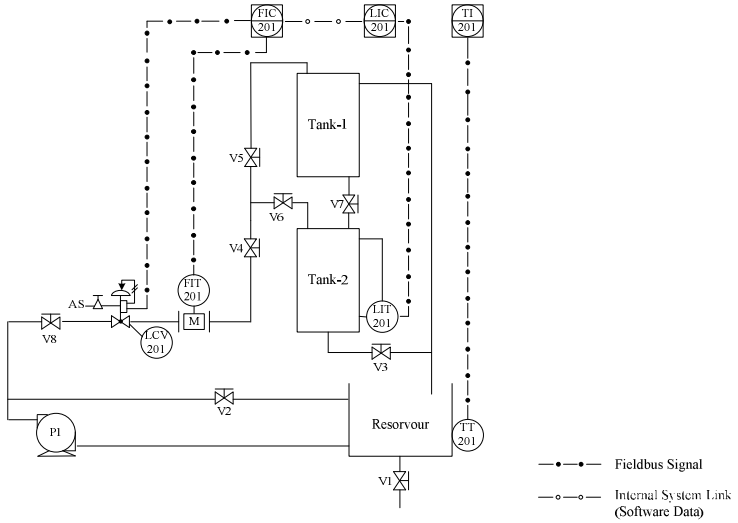
ตารางที่ 3.2 รายละเอียดของอุปกรณ์ฟาวน์เดชันฟิลด์บัสที่อยู่ภายในพลานต์โมเดล 2

Tag	Model	Rev.	AI count	AI time	AO count	AO time	PID count	PID time
TT_A201	STT35F	4	1	23 ms	N/A	N/A	1	90 ms
LCV_A201	DVC6000F	1	1	25 ms	1	25 ms	1	30 ms
FIT_A201	AXF	1	1	30 ms	N/A	N/A	1	50 ms
LIT_A201	DSTJ	1	2	75 ms	N/A	N/A	1	125 ms

จากพลานต์โมเดลในรูปที่ 3.4 สามารถนำมาเขียนในลักษณะของแผนภาพกระบวนการผลิต ได้ดังรูปที่ 3.5 และ 3.6 ตามลำดับ โดยในรูปที่ 3.5 นั้นเป็น P&ID ของพลานต์โมเดล 2 ในกรณีที่กำหนดให้ตัวควบคุมอยู่ภายในตัวอุปกรณ์หรือ Control in the Field ส่วนรูปที่ 3.6 เป็น P&ID ของพลานต์โมเดล 2 ในกรณีที่ควบคุมด้วยระบบดีซีเอสหรือ Control in the Host



รูปที่ 3.5 P&ID ของพลานต์โมเดล 2 ในกรณี Control in the Field

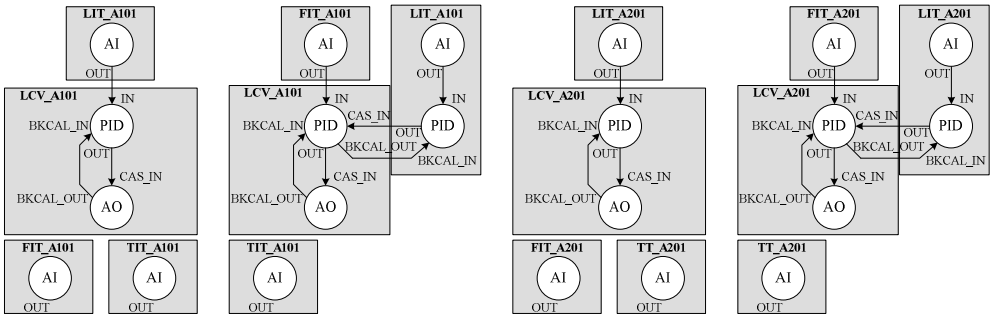


รูปที่ 3.6 P&ID ของพลานต์โมเดล 2 ในกรณีควบคุมด้วยระบบดีซีเอส

3.3 วงควบคุมฟิลด์บัสที่ใช้เป็นกรณีศึกษา

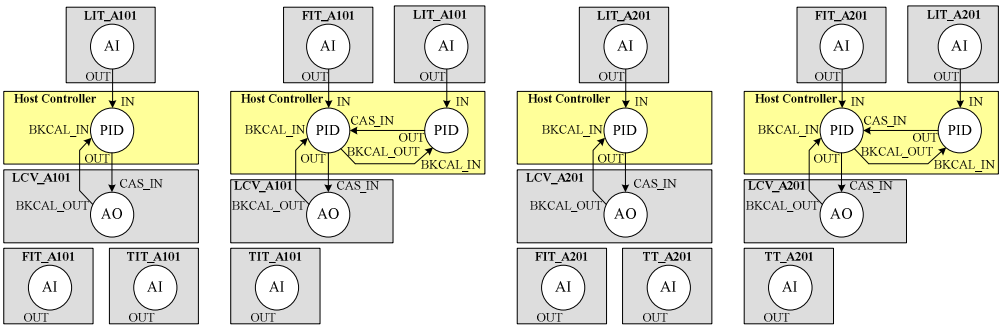
วงควบคุมฟิลด์บัสสร้างขึ้นจากการเลือกและเชื่อมต่อฟังก์ชันบล็อกที่ต้องการเข้าด้วยกัน ซึ่งจะส่งผลให้ค่าตัวแปรต่าง ๆ ภายในกระบวนการสามารถส่งจากบล็อกหนึ่งไปยังบล็อกต่อ ๆ ไปได้ รูปที่ 3.7 และ 3.8 แสดงการกำหนดวงควบคุมฟิลด์บัสที่ใช้สำหรับการทดลองนี้ โดยในแต่ละวงควบคุมจะประกอบไปด้วยส่วนหลัก ๆ ที่สำคัญ ได้แก่ ฟังก์ชันบล็อก AI ทำหน้าที่ในการรับค่าที่อ่านได้จากทรานสดิวเซอร์บล็อกภายในเครื่องวัด ฟังก์ชันบล็อก AO ทำหน้าที่ในการส่งค่าที่ได้จากการประมวลผลของตัวควบคุมไปยังทรานสดิวเซอร์บล็อกภายในวาล์วควบคุม ฟังก์ชันบล็อก PID ทำหน้าที่ในการนำค่าต่าง ๆ ที่ได้จากกระบวนการมาประมวลผลเพื่อส่งค่าออกไปควบคุมระบบ ซึ่งในวิทยานิพนธ์ฉบับนี้จะทำการทดลองการควบคุมกระบวนการโดยอาศัยคุณสมบัติ Control in the

Field ของฟาวน์เดชันฟิลด์บัสเปรียบเทียบกับการควบคุมโดยระบบดีซีเอสแบบดั้งเดิม เพื่อศึกษาการจัดมาโครไซเคิลของการควบคุมในแต่ละรูปแบบ



a) 1A_CIF b) 1B_CIF c) 2A_CIF d) 2B_CIF

รูปที่ 3.7 วงควบคุมฟิลด์บัสสำหรับการควบคุมที่ระดับตัวอุปกรณ์



a) 1A_DCS b) 1B_DCS c) 2A_DCS d) 2B_DCS

รูปที่ 3.8 วงควบคุมฟิลด์บัสสำหรับการควบคุมที่ดีซีเอส

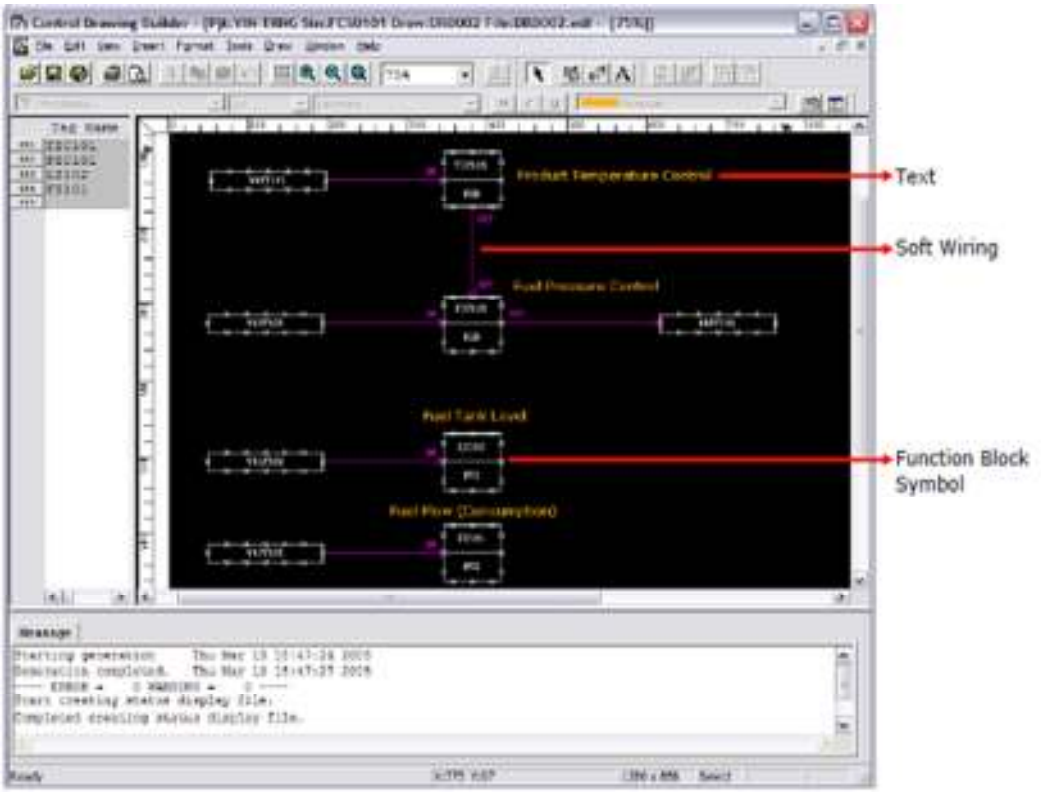
3.4 การสร้างฟิลด์บัสเซกเมนต์เพื่อศึกษาการจัดตารางเวลาในการสื่อสาร

ในการศึกษาการจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ อันดับแรกคือจะต้องสร้างและเชื่อมต่อฟังก์ชันบล็อกเข้าด้วยกันดังที่ได้กล่าวมาแล้วในบทที่ 2 จากนั้นโปรแกรมที่อยู่ภายในดีซีเอสซึ่งทำหน้าที่เป็นโฮสของระบบจะทำการสร้างตารางมาโครไซเคิลขึ้นมาโดยอัตโนมัติ ทำให้เราสามารถคำนวณหาภาระงานในการสื่อสารของระบบในแต่ละกรณีได้ ซึ่งในงานวิจัยนี้ได้ใช้ระบบดีซีเอสจากสองบริษัทผู้ผลิตด้วยกันคือระบบ Centum VP ซึ่งเป็นของบริษัท Yokogawa และระบบ Delta V ซึ่งเป็นของบริษัท Emerson ตามลำดับ

3.4.1 การสร้างฟิลด์บัสเซกเมนต์โดยระบบ Centum VP

ในการกำหนดฟังก์ชันการควบคุมของระบบ Centum VP นั้น ได้ใช้โปรแกรมที่มีชื่อว่า Control Drawing Builder ซึ่งภายในโปรแกรมประกอบไปด้วยฟังก์ชันบล็อกมาตรฐานของระบบ

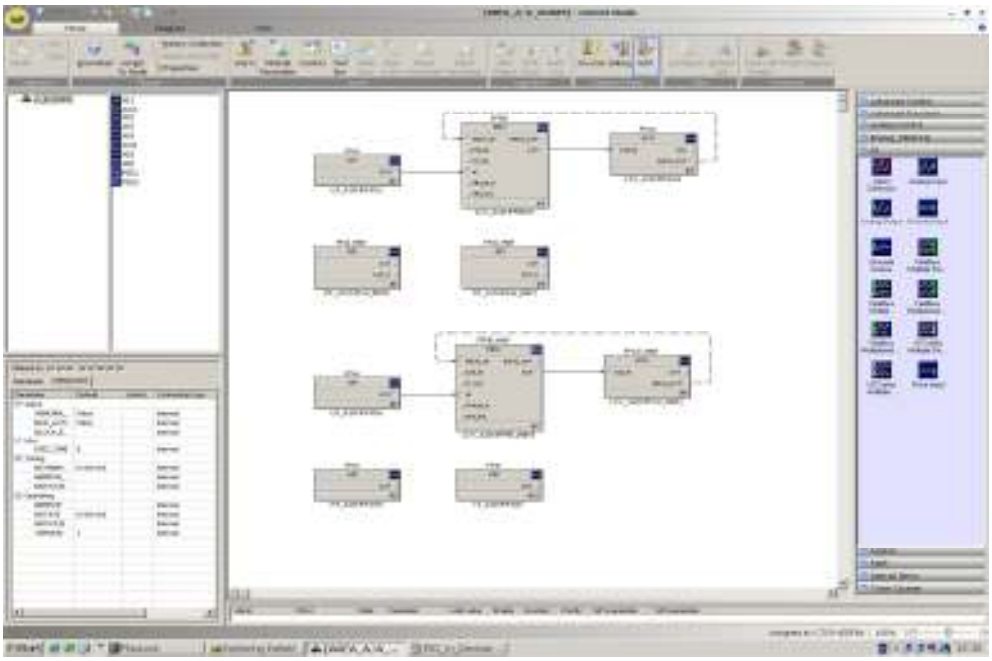
ดีซีเอสที่ถูกกำหนดโดยผู้ผลิต และฟังก์ชันบล็อกของอุปกรณ์ฟิลด์บัสในการกำหนดตั้งค่าแบบ Offline จากข้อมูลที่ได้จากไฟล์ CFF ที่ถูกติดตั้งในโฮส ผู้ใช้สามารถเลือกและเชื่อมต่อฟังก์ชันบล็อกตามที่ต้องการได้ ซึ่งลำดับของการจัดวางและการเชื่อมต่อฟังก์ชันบล็อกในแต่ละกรณีศึกษานั้น ควรจะมีลำดับขั้นตอนเป็นไปตามลักษณะการดำเนินการของระบบควบคุมโดยทั่วไป เช่น ฟังก์ชันบล็อก AI ควรจะถูกจัดวางก่อนฟังก์ชันบล็อกPID และฟังก์ชันบล็อก PID ก็ควรจะถูกจัดวางก่อนฟังก์ชันบล็อก AO เป็นต้น ทั้งนี้เนื่องจากลำดับของการจัดวางบล็อกแต่ละบล็อกในขณะที่ทำการสร้างวงควบคุมฟิลด์บัสนั้น จะเกี่ยวเนื่องกับลำดับการดำเนินการของแต่ละฟังก์ชันบล็อกภายในตารางเวลาในการสื่อสารข้อมูลฟิลด์บัส ซึ่งจะส่งผลต่อมาโครไซเคิลนั่นเอง [2] รูปที่ 3.9 แสดงตัวอย่างการสร้างฟิลด์บัสเซกเมนต์โดยระบบ Centum VP



รูปที่ 3.9 ตัวอย่างการสร้างฟิลด์บัสเซกเมนต์โดยระบบ Centum VP

3.4.2 การสร้างฟิลด์บัสเซกเมนต์โดยระบบ DeltaV

ในการกำหนดฟังก์ชันการควบคุมของระบบ DeltaV นั้น ได้ใช้โปรแกรมที่มีชื่อว่า Control Studio ซึ่งภายในโปรแกรมประกอบไปด้วยฟังก์ชันบล็อกมาตรฐานของระบบดีซีเอสที่ถูกกำหนดโดยผู้ผลิต และฟังก์ชันบล็อกของอุปกรณ์ฟิลด์บัสในการกำหนดตั้งค่าแบบ Offline จากข้อมูลที่ได้จากไฟล์ CFF ที่ถูกติดตั้งในโฮส ผู้ใช้สามารถเลือกและเชื่อมต่อฟังก์ชันบล็อกตามที่ออกแบบไว้ได้ โดยมีข้อกำหนดต่าง ๆ เช่นเดียวกับการสร้างฟิลด์บัสเซกเมนต์โดยระบบ Centum VP ตัวอย่างการสร้างฟิลด์บัสเซกเมนต์โดยระบบ DeltaV แสดงดังรูปที่ 3.10



รูปที่ 3.10 ตัวอย่างการสร้างฟิลด์บัสเซกเมนต์โดยระบบ DeltaV

3.5 ผลและการวิเคราะห์ตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ที่ได้จากกรณีศึกษา

วิทยานิพนธ์ฉบับนี้ได้ทำการศึกษาการจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์ที่ประกอบด้วยหนึ่งวงควบคุมและสองวงควบคุม โดยมุ่งความสนใจไปที่ความสัมพันธ์ระหว่างตำแหน่งของตัวควบคุมกับเวลาที่ใช้ในการรับ-ส่งข้อมูลของ Schedule Communications ซึ่งจัดเป็นภาระในการสื่อสารบนเครือข่าย ในการทดลองได้ทำการแบ่งกรณีศึกษาออกเป็น 8 กรณีที่แตกต่างกัน โดยอุปกรณ์ที่อยู่ในพลานต์โมเดล 1 และ พลานต์โมเดล 2 ถูกจัดให้อยู่บนฟิลด์บัสเซกเมนต์ 1 และเซกเมนต์ 2 ตามลำดับ และใช้การควบคุมแบบ PID และการควบคุมแบบคาสเคดในการควบคุมกระบวนการ กรณีศึกษาที่ใช้ในงานวิจัยฉบับนี้แสดงดังตารางที่ 3.3

ตารางที่ 3.3 กรณีศึกษาที่ใช้ในวิทยานิพนธ์

ตำแหน่งตัวควบคุม	กรณีที่	วงควบคุม	รายละเอียด
CIF	1	1A	การควบคุมแบบ PID ของพลานต์ 1 บนเซกเมนต์ 1
	2	2A	การควบคุมแบบ PID ของพลานต์ 2 บนเซกเมนต์ 2
	3	1B	การควบคุมแบบคาสเคดของพลานต์ 1 บนเซกเมนต์ 1
	4	2B	การควบคุมแบบคาสเคดของพลานต์ 2 บนเซกเมนต์ 2
	5	1A + 2A	การควบคุมแบบ PID ของพลานต์ 1 และ 2 บนเซกเมนต์ 2
	6	1B + 2B	การควบคุมแบบคาสเคดของพลานต์ 1 และ 2 บนเซกเมนต์ 2
	7	1A + 2B	การควบคุมแบบ PID ของพลานต์ 1 และการควบคุมแบบคาสเคดของพลานต์ 2 บนเซกเมนต์ 2
	8	1B + 2A	การควบคุมแบบคาสเคดของพลานต์ 1 และการควบคุมแบบ PID ของพลานต์ 2 บนเซกเมนต์ 2

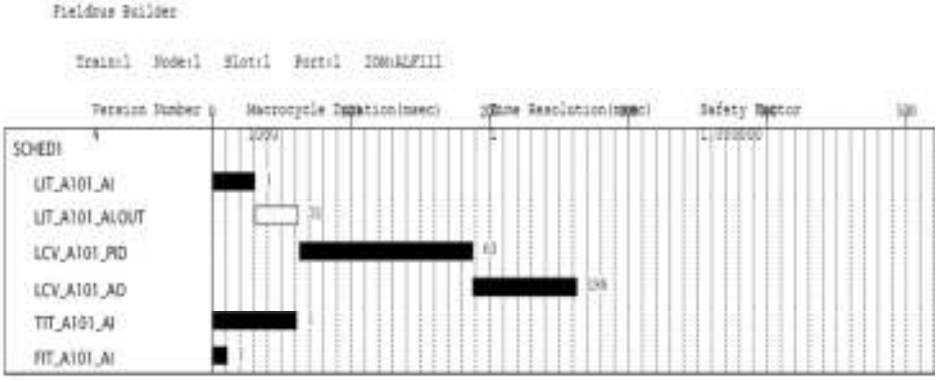
DCS	1	1A	การควบคุมแบบ PID ของพลานต์ 1 บนเซกเมนต์ 1
	2	2A	การควบคุมแบบ PID ของพลานต์ 2 บนเซกเมนต์ 2
	3	1B	การควบคุมแบบคาสเคดของพลานต์ 1 บนเซกเมนต์ 1
	4	2B	การควบคุมแบบคาสเคดของพลานต์ 2 บนเซกเมนต์ 2
	5	1A + 2A	การควบคุมแบบ PID ของพลานต์ 1 และ 2 บนเซกเมนต์ 2
	6	1B + 2B	การควบคุมแบบคาสเคดของพลานต์ 1 และ 2 บนเซกเมนต์ 2
	7	1A + 2B	การควบคุมแบบ PID ของพลานต์ 1 และการควบคุมแบบคาสเคดของพลานต์ 2 บนเซกเมนต์ 2
	8	1B + 2A	การควบคุมแบบคาสเคดของพลานต์ 1 และการควบคุมแบบ PID ของพลานต์ 2 บนเซกเมนต์ 2

จากตารางที่ 3.3 จะเห็นว่ากรณีศึกษาแต่ละกรณีจะมีความแตกต่างกันทั้งในเรื่องตำแหน่งของตัวควบคุมและวิธีการที่ใช้ในการควบคุม ซึ่งแน่นอนว่ามาโครไซเคิลที่ถูกสร้างขึ้นโดยระบบ Centum VP และ DeltaV ของแต่ละกรณีนั้นย่อมให้ผลที่แตกต่างกันด้วย การจัดมาโครไซเคิลที่ได้จากการทดสอบกรณีศึกษามีดังต่อไปนี้

3.5.1 การทดสอบกรณีศึกษาโดยใช้ระบบ Centum VP

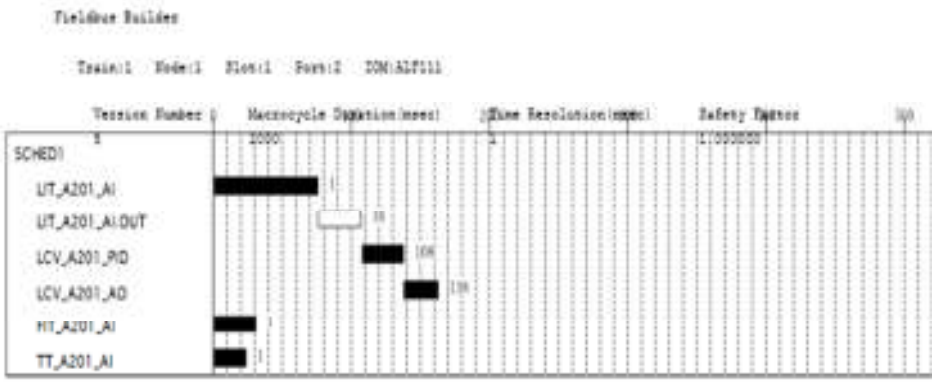
3.5.1.1 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ภายในตัวอุปกรณ์

<u>กรณีที่ 1</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 263 มิลลิวินาที



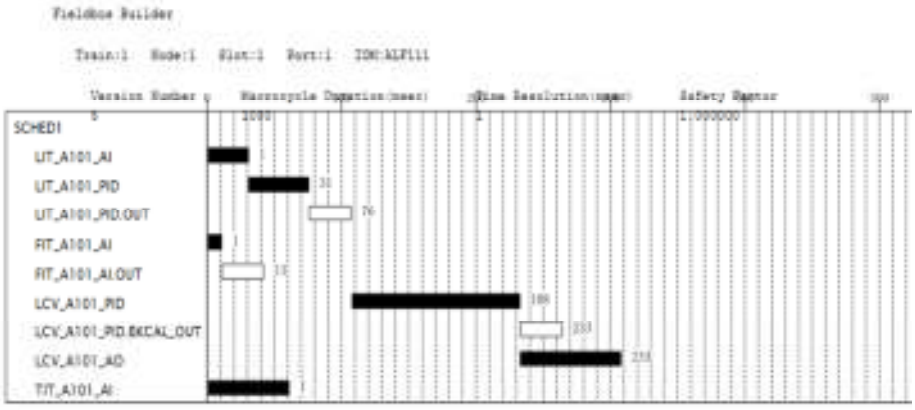
รูปที่ 3.11 มาโครไซเคิลของกรณีที่ 1_CIF ที่ได้จากระบบ Centum VP

<u>กรณีที่ 2</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 163 มิลลิวินาที



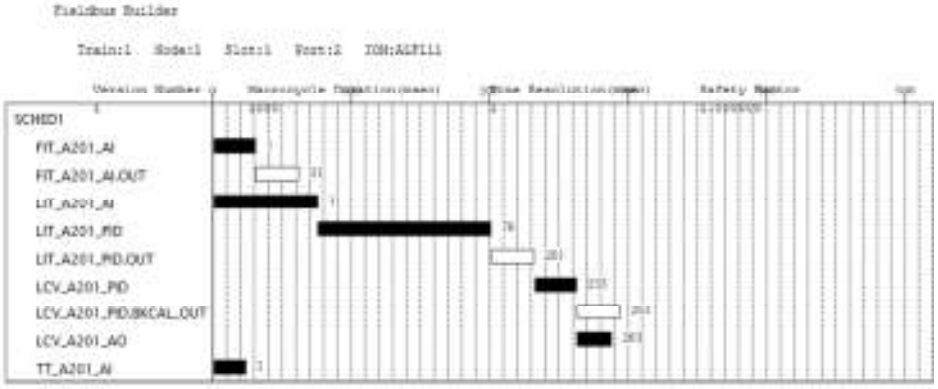
รูปที่ 3.12 มาโครไซเคิลของกรณีที่ 2_CIF ที่ได้จากระบบ Centum VP

<u>กรณีที่ 3</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 308 มิลลิวินาที



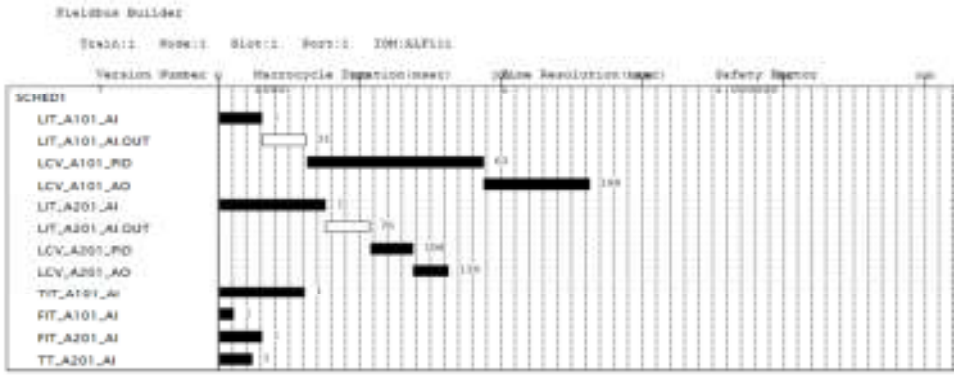
รูปที่ 3.13 มาโครไซเคิลของกรณีที่ 3_CIF ที่ได้จากระบบ Centum VP

<u>กรณีที่ 4</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 294 มิลลิวินาที



รูปที่ 3.14 มาโครไซเคิลของกรณีที่ 4_CIF ที่ได้จากระบบ Centum VP

<u>กรณีที่ 5</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 263 มิลลิวินาที

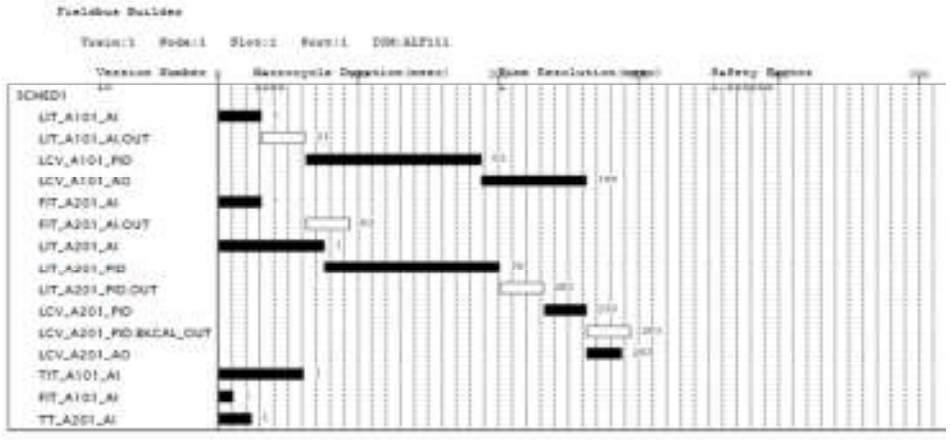


รูปที่ 3.15 มาโครไซเคิลของกรณีที่ 5_CIF ที่ได้จากระบบ Centum VP

<u>กรณีที่ 6</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 308 มิลลิวินาที

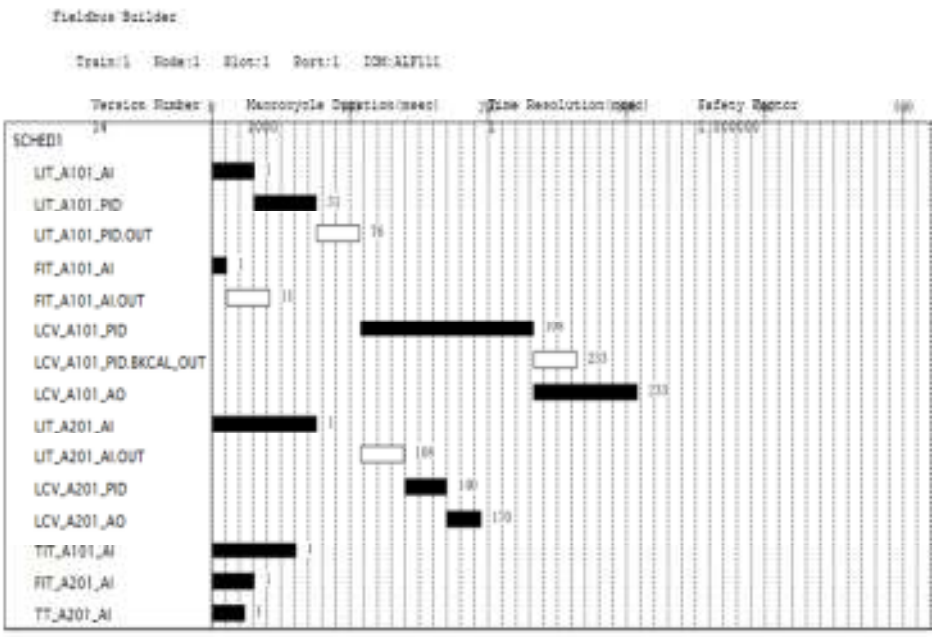
รูปที่ 3.16 มาโครไซเคิลของกรณีที่ 6_CIF ที่ได้จากระบบ Centum VP

<u>กรณีที่ 7</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 294 มิลลิวินาที



รูปที่ 3.17 มาโครไซเคิลของกรณีที่ 7_CIF ที่ได้จากระบบ Centum VP

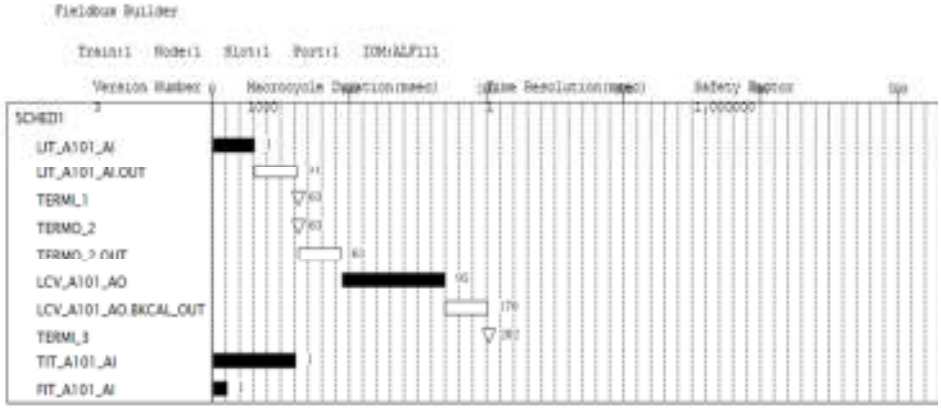
<u>กรณีที่ 8</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 308 มิลลิวินาที



รูปที่ 3.18 มาโครไซเคิลของกรณีที่ 8_CIF ที่ได้จากระบบ Centum VP

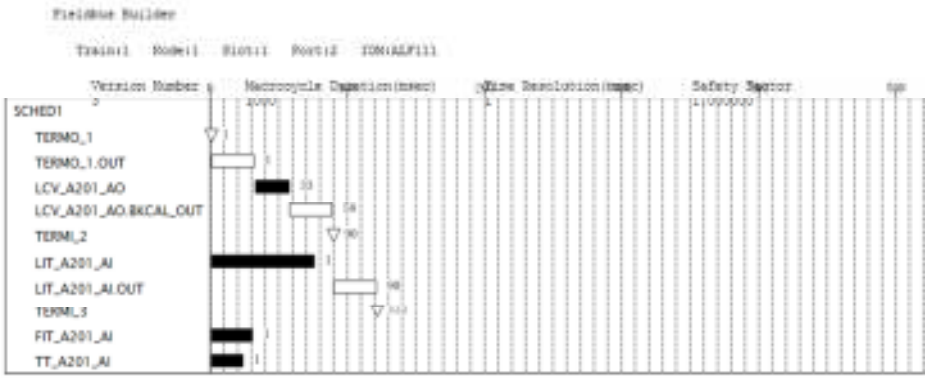
3.5.1.2 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ที่ดีซีเอส

<u>กรณีที่ 1</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 202 มิลลิวินาที



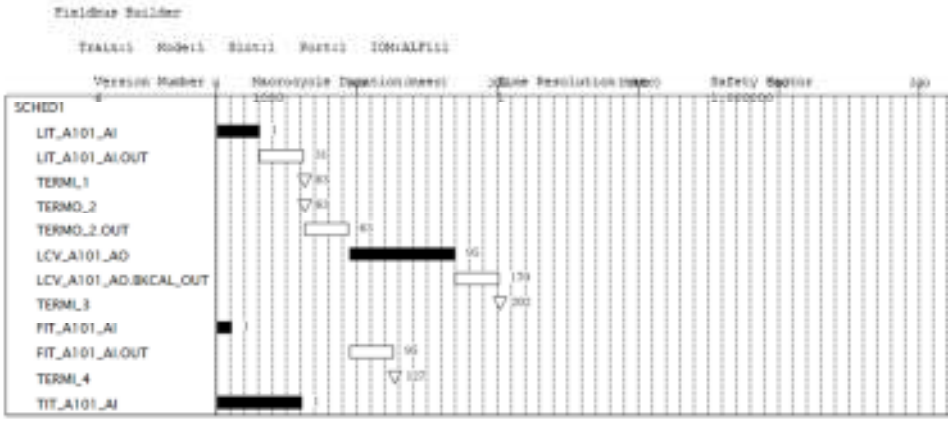
รูปที่ 3.19 มาโครไซเคิลของกรณีที่ 1_DCS ที่ได้จากระบบ Centum VP

<u>กรณีที่ 2</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 122 มิลลิวินาที



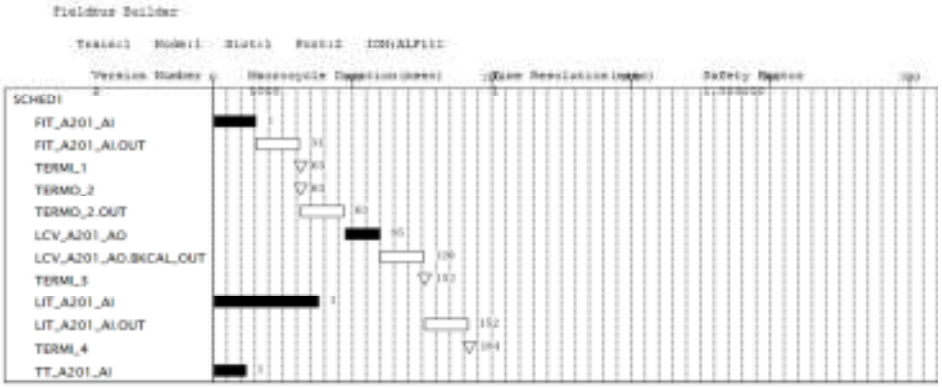
รูปที่ 3.20 มาโครไซเคิลของกรณีที่ 2_DCS ที่ได้จากระบบ Centum VP

<u>กรณีที่ 3</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 202 มิลลิวินาที



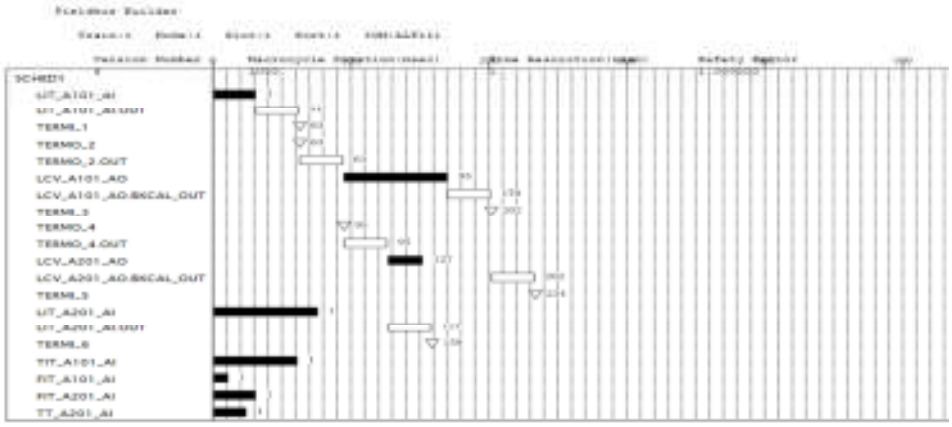
รูปที่ 3.21 มาโครไซเคิลของกรณีที่ 3_DCS ที่ได้จากระบบ Centum VP

<u>กรณีที่ 4</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 184 มิลลิวินาที



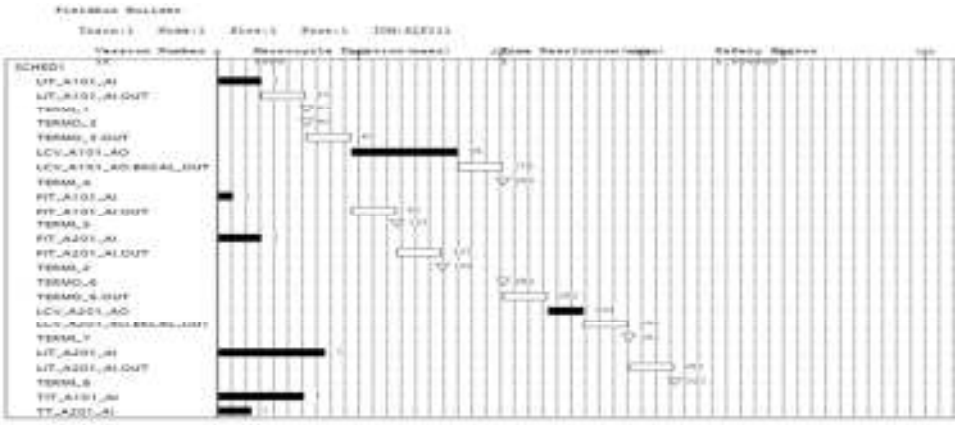
รูปที่ 3.22 มาโครไซเคิลของกรณีที่ 4_DCS ที่ได้จากระบบ Centum VP

<u>กรณีที่ 5</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 234 มิลลิวินาที



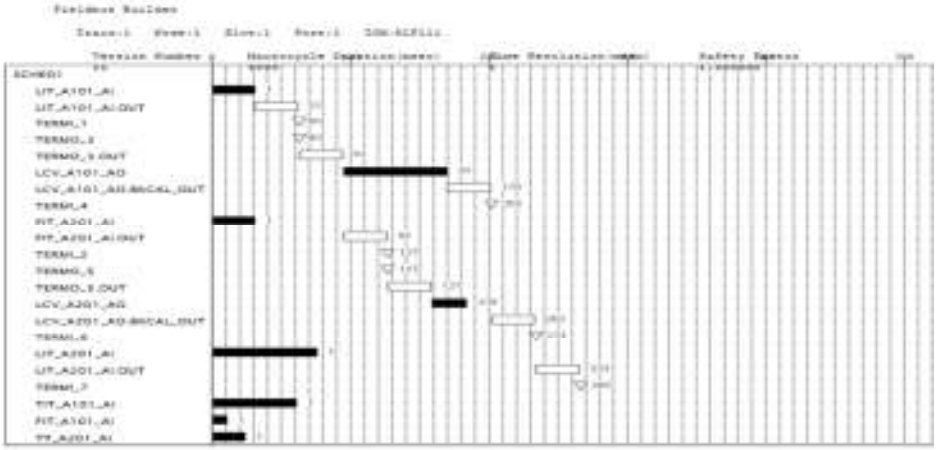
รูปที่ 3.23 มาโครไซเคิลของกรณีที่ 5_DCS ที่ได้จากระบบ Centum VP

<u>กรณีที่ 6</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 323 มิลลิวินาที



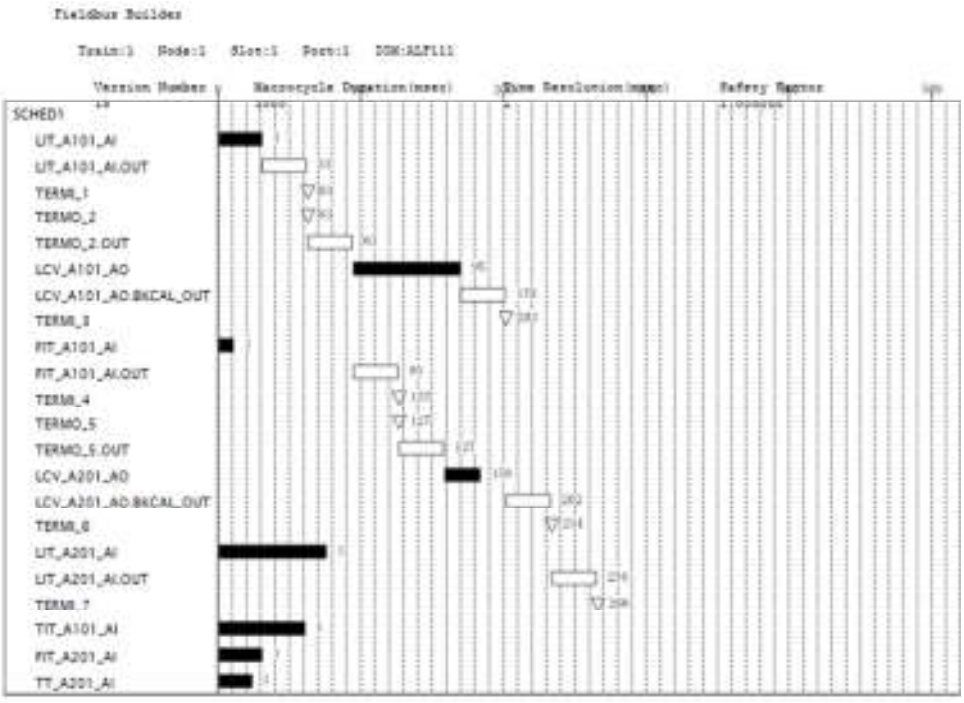
รูปที่ 3.24 มาโครไซเคิลของกรณีที่ 6_DCS ที่ได้จากระบบ Centum VP

<u>กรณีที่ 7</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 266 มิลลิวินาที



รูปที่ 3.25 มาโครไซเคิลของกรณีที่ 7_DCS ที่ได้จากระบบ Centum VP

<u>กรณีที่ 8</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 266 มิลลิวินาที



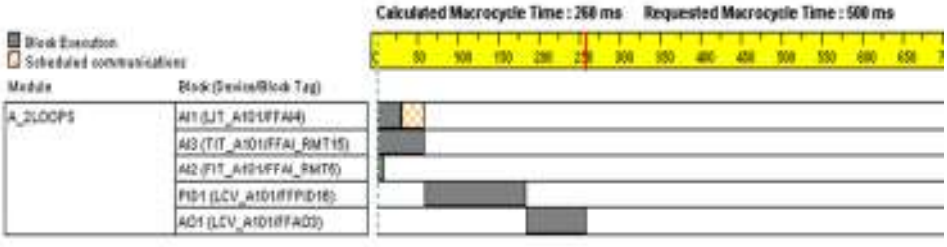
รูปที่ 3.26 มาโครไซเคิลของกรณีที่ 8_DCS ที่ได้จากระบบ Centum VP

3.5.2 การทดสอบกรณีศึกษาโดยใช้ระบบ DeltaV

ระบบ DeltaV รองรับการกำหนดเวลาของมาโครไซเคิลในสองรูปแบบด้วยกันคือ แบบที่กำหนดโดยผู้ใช้งานและแบบที่กำหนดโดยระบบ เมื่อค่า Requested Macrocycle Time ถูกกำหนดโดยผู้ใช้งาน บล็อกทุกบล็อกที่อยู่ในมาโครไซเคิลจะทำงานที่อัตราการสแกน (Scan Rate) เดียวกัน เพื่อให้ได้ค่ามาโครไซเคิลที่เกิดขึ้นจริง ระบบ DeltaV จะใช้ค่าที่ดีกว่าระหว่างค่ามาโครไซเคิลที่กำหนดโดยผู้ใช้งานกับค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบ ซึ่งก็หมายความว่าค่ามาโครไซเคิลที่เกิดขึ้นจริงจะเท่ากับ Requested Macrocycle Time ถ้าค่า Calculated Macrocycle Time มีค่าน้อยกว่าหรือเท่ากับค่า Requested Macrocycle Time นั่นเอง

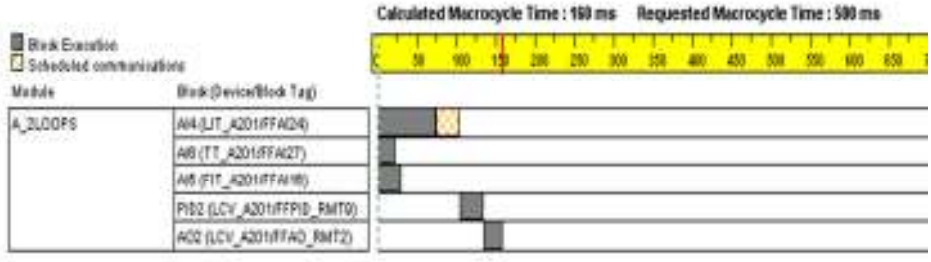
3.5.2.1 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ภายในตัวอุปกรณ์

<u>กรณีที่ 1</u> : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 260 มิลลิวินาที



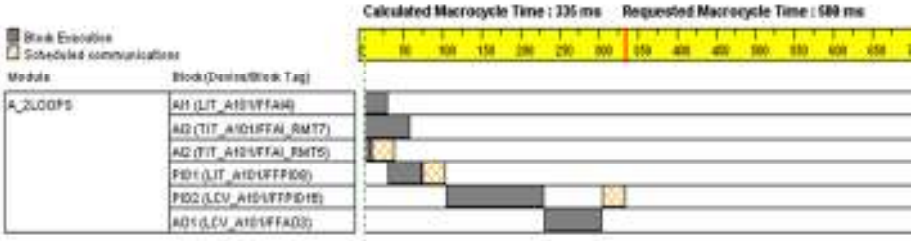
รูปที่ 3.27 มาโครไซเคิลของกรณีที่ 1_CIF ที่ได้จากระบบ DeltaV

กรณีที่ 2 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 160 มิลลิวินาที



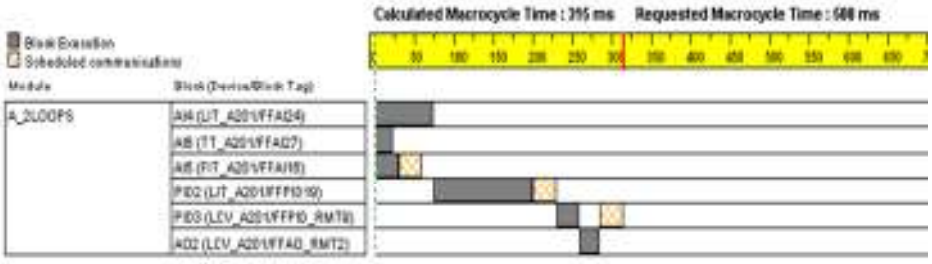
รูปที่ 3.28 มาโครไซเคิลของกรณีที่ 2_CIF ที่ได้จากระบบ DeltaV

กรณีที่ 3 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 335 มิลลิวินาที



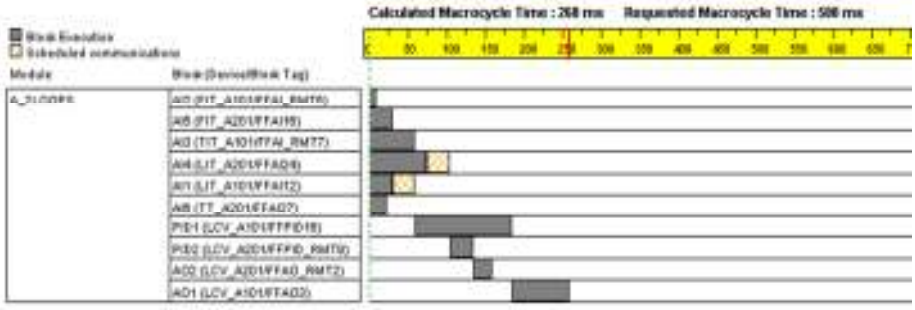
รูปที่ 3.29 มาโครไซเคิลของกรณีที่ 3_CIF ที่ได้จากระบบ DeltaV

กรณีที่ 4 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 315 มิลลิวินาที



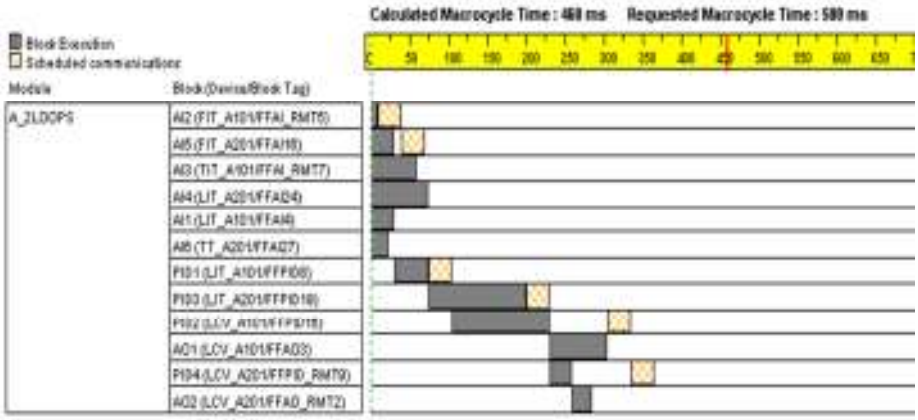
รูปที่ 3.30 มาโครไซเคิลของกรณีที่ 4_CIF ที่ได้จากระบบ DeltaV

กรณีที่ 5 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 260 มิลลิวินาที



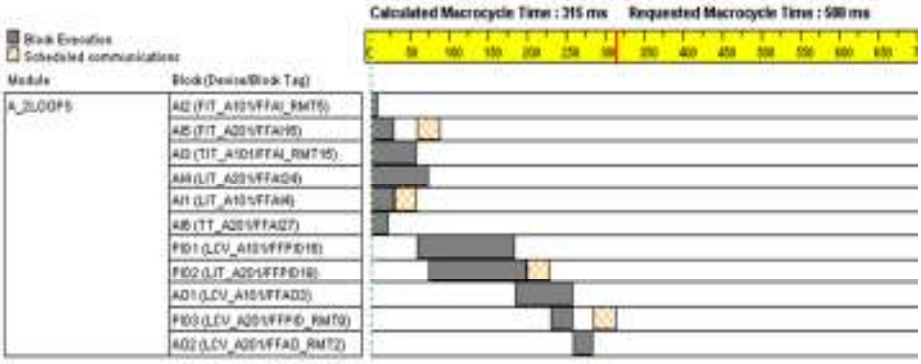
รูปที่ 3.31 มาโครไซเคิลของกรณีที่ 5_CIF ที่ได้จากระบบ DeltaV

กรณีที่ 6 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 460 มิลลิวินาที



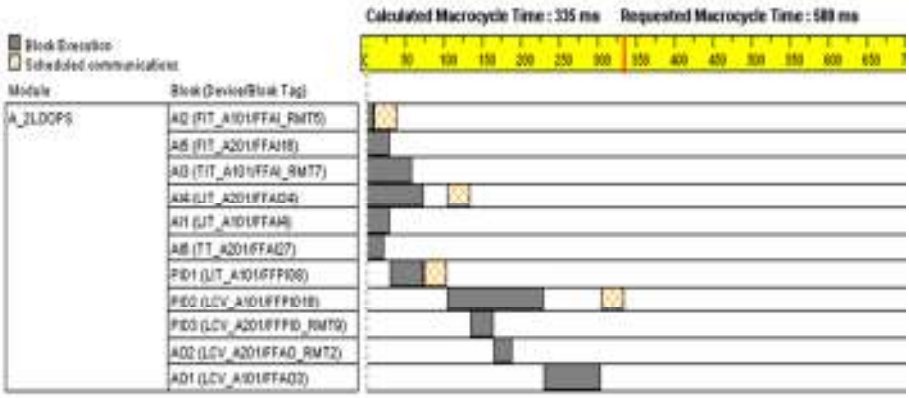
รูปที่ 3.32 มาโครไซเคิลของกรณีที่ 6_CIF ที่ได้จากระบบ DeltaV

กรณีที่ 7 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 315 มิลลิวินาที



รูปที่ 3.33 มาโครไซเคิลของกรณีที่ 7_CIF ที่ได้จากระบบ DeltaV

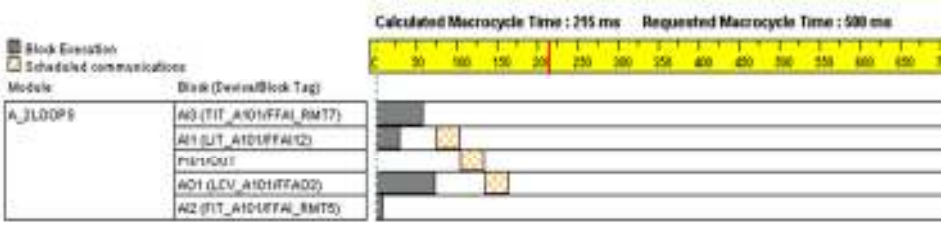
กรณีที่ 8 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 335 มิลลิวินาที



รูปที่ 3.34 มาโครไซเคิลของกรณีที่ 8_CIF ที่ได้จากระบบ DeltaV

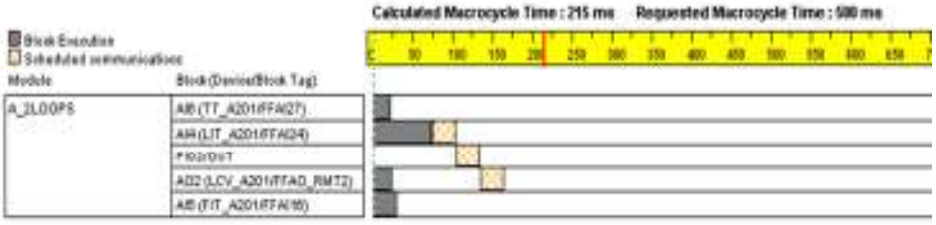
3.5.2.2 การทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ที่ดีซีเอส

กรณีที่ 1 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 215 มิลลิวินาที



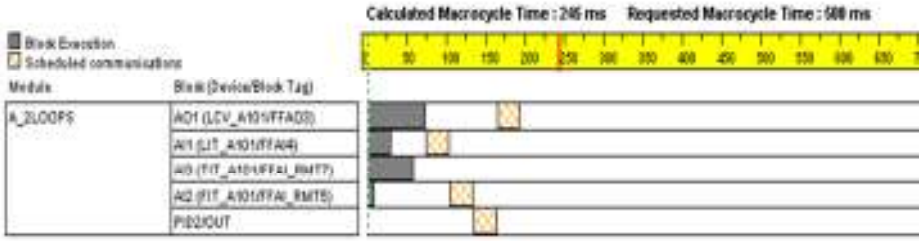
รูปที่ 3.35 มาโครไซเคิลของกรณีที่ 1_DCS ที่ได้จากระบบ DeltaV

กรณีที่ 2 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 215 มิลลิวินาที



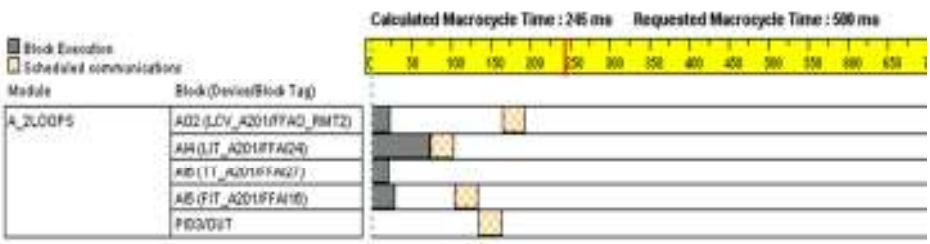
รูปที่ 3.36 มาโครไซเคิลของกรณีที่ 2_DCS ที่ได้จากระบบ DeltaV

กรณีที่ 3 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 245 มิลลิวินาที



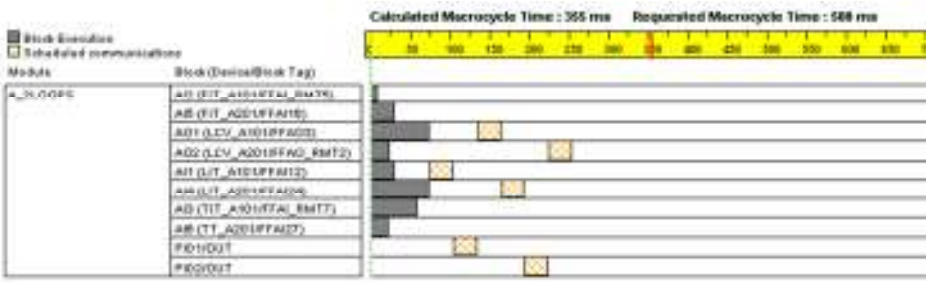
รูปที่ 3.37 มาโครไซเคิลของกรณีที่ 3_DCS ที่ได้จากระบบ DeltaV

กรณีที่ 4 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 245 มิลลิวินาที



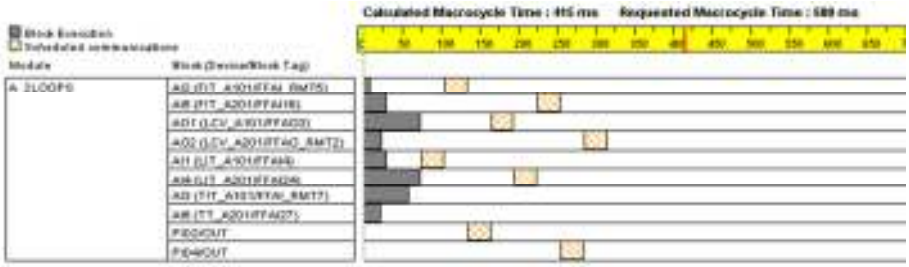
รูปที่ 3.38 มาโครไซเคิลของกรณีที่ 4_DCS ที่ได้จากระบบ DeltaV

กรณีที่ 5 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 355 มิลลิวินาที



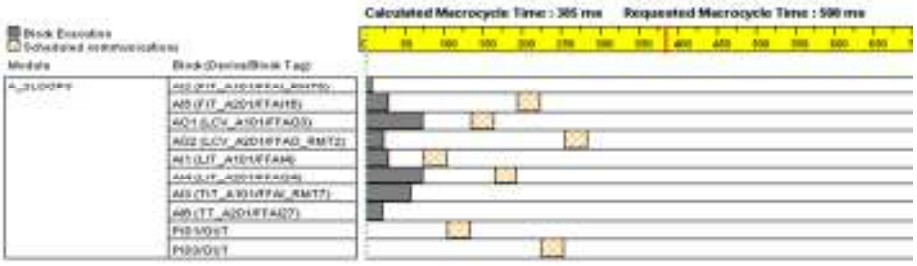
รูปที่ 3.39 มาโครไซเคิลของกรณีที่ 5_DCS ที่ได้จากระบบ DeltaV

กรณีที่ 6 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 415 มิลลิวินาที



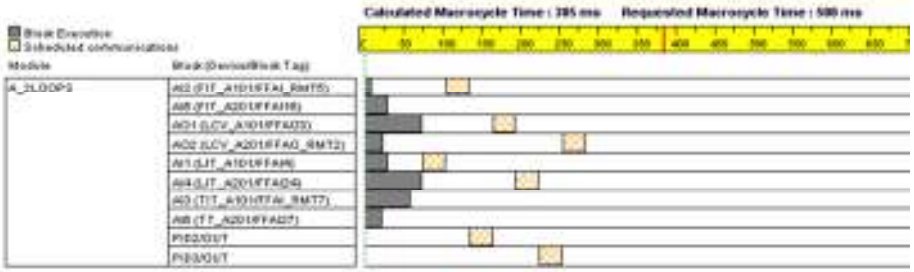
รูปที่ 3.40 มาโครไซเคิลของกรณีที่ 6_DCS ที่ได้จากระบบ DeltaV

กรณีที่ 7 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 385 มิลลิวินาที



รูปที่ 3.41 มาโครไซเคิลของกรณีที่ 7_DCS ที่ได้จากระบบ DeltaV

กรณีที่ 8 : ค่ามาโครไซเคิลที่ได้จากการคำนวณโดยระบบเท่ากับ 385 มิลลิวินาที



รูปที่ 3.42 มาโครไซเคิลของกรณีที่ 8_DCS ที่ได้จากระบบ DeltaV

จากการทดลองสามารถสรุปค่าเวลาของมาโครไซเคิลซึ่งถูกคำนวณโดยดีซีเอสทั้ง 2 ระบบ ได้ดังตารางที่ 3.4 และ 3.5 ตามลำดับ อย่างไรก็ตามการคำนวณเวลาของมาโครไซเคิลสำหรับการควบคุมบนดีซีเอสดังแสดงในตารางที่ 3.5 นั้นจะไม่รวมเวลาในการดำเนินการของตัวควบคุม ในทำนองเดียวกันการดำเนินการของฟังก์ชัน PID บนดีซีเอสโดยทั่วไปก็จะไม่ถูกซิงโครไนซ์ (Synchronized) กับการดำเนินการของมาโครไซเคิลบนฟิลด์บัสเซกเมนต์ ซึ่งก็หมายความว่าการเปลี่ยนแปลงทางด้านเอาต์พุตของตัวควบคุมไปยังวาล์วควบคุมจะไม่ถูกส่งออกไปจนกว่าจะถึงมาโครไซเคิลถัดไปนั่นเอง

ตารางที่ 3.4 สรุปค่าเวลาของมาโครไซเคิลซึ่งคำนวณโดยระบบ DeltaV

Case	Control loop(s) in the Fieldbus segment	Number of scheduled comm.		Calculated macrocycle time	
		CIF	DCS	CIF	DCS
1	1A_CIF	1	3	260 ms	215 ms
2	2A_CIF	1	3	160 ms	215 ms
3	1B_CIF	3	4	335 ms	245 ms
4	2B_CIF	3	4	315 ms	245 ms
5	1A_CIF + 2A_CIF	2	6	260 ms	355 ms
6	1B_CIF + 2B_CIF	6	8	460 ms	415 ms
7	1A_CIF + 2B_CIF	4	7	315 ms	385 ms
8	1B_CIF + 2A_CIF	4	7	335 ms	385 ms

ตารางที่ 3.5 สรุปค่าเวลาของมาโครไซเคิลซึ่งคำนวณโดยระบบ Centum VP

Case	Control loop(s) in the Fieldbus segment	Number of scheduled comm.		Calculated macrocycle time	
		CIF	DCS	CIF	DCS
1	1A_DCS	1	3	263 ms	202 ms
2	2A_ DCS	1	3	163 ms	122 ms
3	1B_ DCS	3	4	308 ms	202 ms
4	2B_ DCS	3	4	294 ms	184 ms
5	1A_ DCS + 2A_ DCS	2	6	263 ms	234 ms
6	1B_ DCS + 2B_ DCS	6	8	308 ms	323 ms
7	1A_ DCS + 2B_ DCS	4	7	294 ms	266 ms
8	1B_ DCS + 2A_ DCS	4	7	308 ms	266 ms

จากตารางมาโครไซเคิลที่ได้จากการทดลอง เราสามารถคำนวณหาค่าประมาณการใช้แบนด์วิธ หรือ Network Load สำหรับการรับ-ส่งข้อมูลแบบ Scheduled Communications ของแต่ละเซกเมนต์ได้จากความสัมพันธ์ดังนี้

$$\text{Network Consumed} = \frac{\text{Total Publication Time}}{\text{Calculated Macrocycle}} \times 100\% \qquad (3.1)$$

เมื่อ Total Publication Time สามารถคำนวณได้จากการนำจำนวนครั้งของ Scheduled Communications คูณด้วยค่าเวลาที่ระบบดีซีเอสใช้ในการรับ-ส่งข้อมูลแบบ Scheduled

Communications โดยที่ระบบ DeltaV และ Centum VP ใช้เวลา 30 และ 31 มิลลิวินาที ตามลำดับ ตัวอย่างเช่นจากกรณีที่ 3 ในตารางที่ 3.4 จะได้ค่า Total Publication Time ของการควบคุมแบบ CIF และ DCS โดยระบบ Delta V เท่ากับ (3 x 30) = 90 มิลลิวินาที และ (4 x 30) = 120 มิลลิวินาที ตามลำดับ ดังนั้นค่า Network Consumed ของการควบคุมแบบ CIF และ DCS ที่คำนวณได้จากระบบ DeltaV คือ $\frac{90}{335}$ x 100= 26.866% และ $\frac{120}{245}$ x 100= 48.980% ในทำนองเดียวกันจากกรณีที่ 3 ในตารางที่ 3.5 จะได้ค่า Total Publication Time ของการควบคุมแบบ CIF และ DCS โดยระบบ Centum VP เท่ากับ (3 x 31) = 93 มิลลิวินาที และ (4 x 31) = 124 มิลลิวินาที ตามลำดับ ดังนั้นค่า Network Consumed ของการควบคุมแบบ CIF และ DCS ที่คำนวณได้จากระบบ Centum VP คือ $\frac{93}{308}$ x 100= 30.195% และ $\frac{124}{202}$ x 100= 61.386% ตามลำดับ จากตัวอย่างข้างต้น ค่าประมาณของการใช้แบนด์วิธจากการกำหนดตั้งค่าระบบแบบ Offline ด้วยระบบ DeltaV และ Centum VP สามารถสรุปได้ดังตารางที่ 3.6

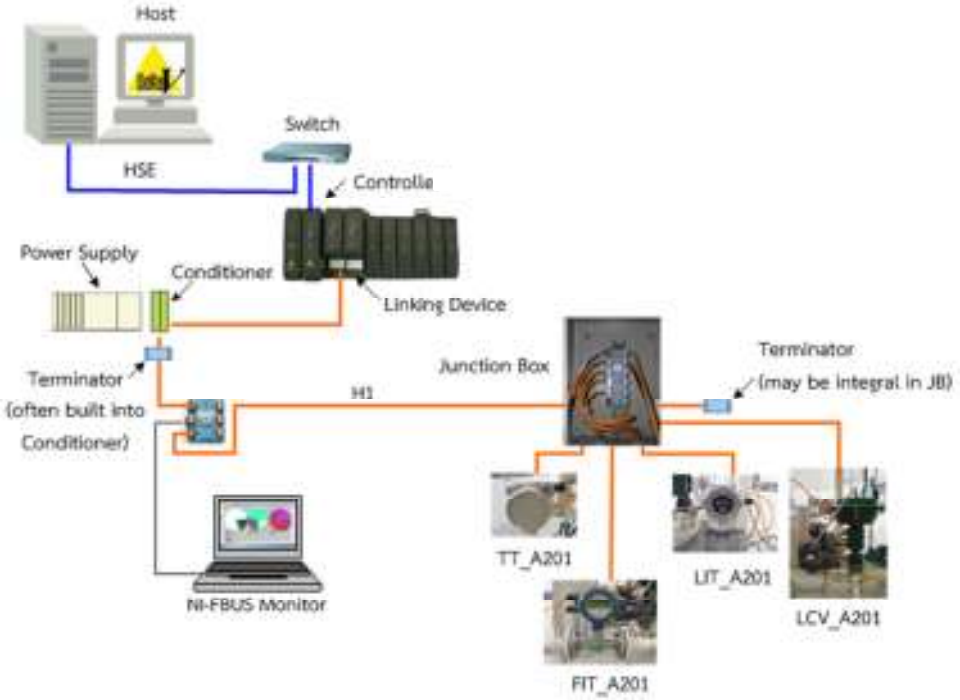
ตารางที่ 3.6 ค่าประมาณการใช้แบนด์วิธสำหรับการสื่อสารแบบ Scheduled Communications

Case	Number of scheduled communications		Consumed Bandwidth			
			DeltaV		CentumVP	
	CIF	DCS	CIF	DCS	CIF	DCS
1	1	3	11.538 %	41.860 %	11.787 %	46.040 %
2	1	3	18.750 %	41.860 %	19.018 %	76.230 %
3	3	4	26.866 %	48.980 %	30.195 %	61.386 %
4	3	4	28.571 %	48.980 %	31.633 %	67.391 %
5	2	6	23.077 %	50.704 %	23.574 %	79.487 %
6	6	8	39.130 %	57.831 %	60.39 %	76.780 %
7	4	7	38.095 %	54.545 %	42.177 %	81.579 %
8	4	7	35.821 %	54.545 %	40.26 %	81.579 %

จากตารางที่ 3.6 ซึ่งเป็นการสรุปค่าประมาณการใช้แบนด์วิธของการรับ-ส่งข้อมูลแบบ Scheduled Communications ภายในหนึ่งมาโครไซเคิล จะเห็นได้ว่าการควบคุมกระบวนการโดยระบบดีซีเอสแบบดั้งเดิมนั้น มาโครไซเคิลที่สร้างขึ้นจากระบบ Centum VP และระบบ DeltaV มีการใช้แบนด์วิธสำหรับ Scheduled Communications ที่มากกว่าเนื่องจากมีจำนวนครั้งของ Scheduled Communications ในแต่ละมาโครไซเคิลมากกว่า ซึ่งอาจส่งผลต่อระยะเวลาในการตอบสนองของระบบควบคุมได้ ซึ่งแนวคิด Control in the Field สามารถช่วยให้เครือข่ายการติดต่อสื่อสารมีประสิทธิภาพมากขึ้น เนื่องจากมีจำนวนครั้งของ Scheduled Communications ที่น้อยกว่า ซึ่งการจัดการให้มีส่วนของมาโครไซเคิลว่างจาก Scheduled Communications นั้นจะเป็นการทำให้ระบบสามารถใช้ช่วงเวลาเหล่านั้นในการเตรียมฟังก์ชันการติดต่อสื่อสารอื่น ๆ เช่น การเปลี่ยนแปลงการกำหนดตั้งค่าโดยผู้ใช้งานและการอัพเดตข้อมูลต่าง ๆ ซึ่งจะส่งผลให้การควบคุมระบบเป็นไปอย่างมีประสิทธิภาพมากยิ่งขึ้น

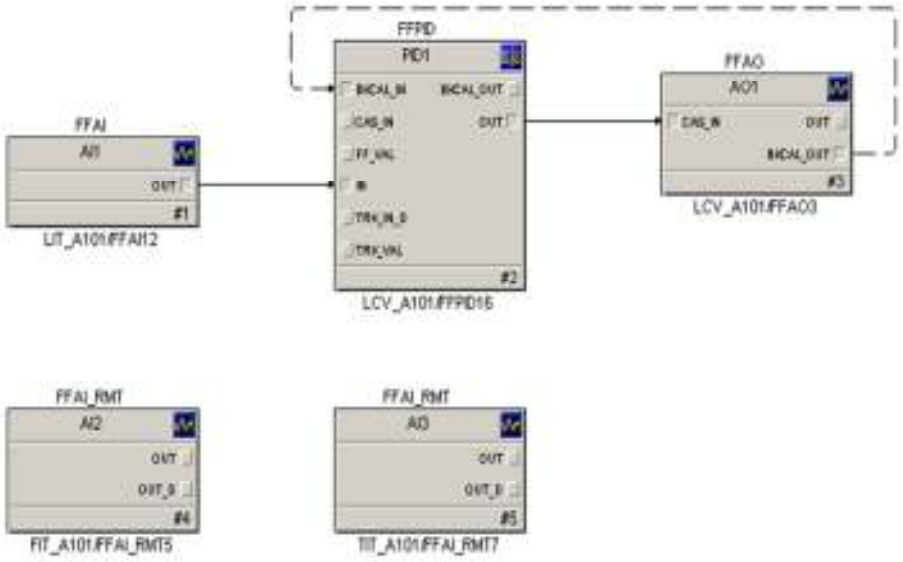
3.6 การทดลองและผลการทดลองจากการทำงานของพลานต์โมเดลที่ใช้เป็นกรณีศึกษา

จากกรณีศึกษาที่กล่าวมาข้างต้น วิทยานิพนธ์ฉบับนี้ได้มีการนำเอากรณีศึกษาในบางกรณีมาทำการทดลองการควบคุมพลานต์โมเดลเพื่อตรวจสอบ Network Load ของ Scheduled Communications ที่เกิดขึ้นจากการทำงานจริง โดยใช้ระบบ DeltaV ทำหน้าที่เป็นโฮส และใช้โปรแกรม NI-FBUS Monitor ในการตรวจสอบการรับ-ส่งข้อมูลของ Scheduled Communications บนเครือข่ายการติดต่อสื่อสาร ในรูปที่ 3.43 แสดงการเชื่อมต่อเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรม NI-FBUS Monitor เข้ากับฟาวน์เดชันฟิลด์บัสเซกเมนต์และโฮสในการทดลอง



รูปที่ 3.43 การใช้โปรแกรม NI-FBUS Monitor ในการทดลอง

การทดลองนี้นำเอากรณีศึกษาที่ 1, 2, 4, 5 และ 7 ซึ่งเป็นการควบคุมทั้งในแบบ PID และแบบคาสเคดมาใช้ควบคุมพลานต์โมเดล 1 และ 2 ซึ่งเป็นกระบวนการควบคุมระดับน้ำในถังปิด โดยกำหนดให้ค่า SP ของทุกกรณีมีค่าเท่ากับ 50 และใช้โปรแกรม NI-FBUS Monitor ในการตรวจสอบค่า Network Load ในช่วงที่ระบบเข้าสู่ SP ตัวอย่างการกำหนดตั้งค่าสำหรับวงควบคุมในโปรแกรม Control Studio แสดงดังรูปที่ 3.44 ซึ่งฟังก์ชันบล็อกที่ใช้ในการกำหนดตั้งค่าสำหรับวงควบคุมในแต่ละกรณีแสดงดังตารางที่ 3.7 และ 3.8 ตามลำดับ



รูปที่ 3.44 ตัวอย่างการกำหนดตั้งค่าสำหรับวงควบคุมใน Control Studio

ตารางที่ 3.7 ฟังก์ชันบล็อกที่ใช้ในการกำหนดตั้งค่าสำหรับวงควบคุมในกรณี Control in the Field ใน Control Studio

Case	Function Block	Field Device
1	AI1	LIT_A101_AI
	AI2	FIT_A101_AI
	AI3	TIT_A101_AI
	PID1	LCV_A101_PID
	AO1	LCV_A101_AO
2	AI1	LIT_A201_AI
	AI2	FIT_A201_AI
	AI3	TT_A201_AI
	PID1	LCV_A201_PID
	AO1	LCV_A201_AO
4	AI1	LIT_A201_AI
	AI2	FIT_A201_AI
	AI3	TT_A201_AI
	PID1	LIT_A201_PID
	PID2	LCV_A201_PID
	AO1	LCV_A201_AO
5	AI1	LIT_A101_AI
	AI2	FIT_A101_AI
	AI3	TIT_A101_AI
	PID1	LCV_A101_PID
	AO1	LCV_A101_AO
	AI4	LIT_A201_AI

	AI5	FIT_A201_AI
	AI6	TT_A201_AI
	PID2	LCV_A201_PID
	AO2	LCV_A201_AO
7	AI1	LIT_A101_AI
	AI2	FIT_A101_AI
	AI3	TIT_A101_AI
	PID1	LCV_A101_PID
	AO1	LCV_A101_AO
	AI4	LIT_A201_AI
	AI5	FIT_A201_AI
	AI6	TT_A201_AI
	PID2	LIT_A201_PID
	PID3	LCV_A201_PID
	AO2	LCV_A201_AO

ตารางที่ 3.8 ฟังก์ชันบล็อกที่ใช้ในการกำหนดตั้งค่าสำหรับวงควบคุมในกรณีที่กำหนดให้ตัวควบคุมอยู่ในดีซีเอส ใน Control Studio

Case	Function Block	Field Device
1	AI1	LIT_A101_AI
	AI2	FIT_A101_AI
	AI3	TIT_A101_AI
	PID1	PID
	AO1	LCV_A101_AO
2	AI1	LIT_A201_AI
	AI2	FIT_A201_AI
	AI3	TT_A201_AI
	PID1	PID
	AO1	LCV_A201_AO
4	AI1	LIT_A201_AI
	AI2	FIT_A201_AI
	AI3	TT_A201_AI
	PID1	PID
	PID2	PID
	AO1	LCV_A201_AO
5	AI1	LIT_A101_AI
	AI2	FIT_A101_AI
	AI3	TIT_A101_AI
	PID1	PID
	AO1	LCV_A101_AO
	AI4	LIT_A201_AI
	AI5	FIT_A201_AI
	AI6	TT_A201_AI

	PID2	PID
	AO2	LCV_A201_AO
7	AI1	LIT_A101_AI
	AI2	FIT_A101_AI
	AI3	TIT_A101_AI
	PID1	PID
	AO1	LCV_A101_AO
	AI4	LIT_A201_AI
	AI5	FIT_A201_AI
	AI6	TT_A201_AI
	PID2	PID
	PID3	PID
	AO2	LCV_A201_AO

การทดลองการควบคุมพลานต์โมเดลในทุกกรณีนั้น ได้ใช้การปรับค่า Gain และ Reset ซึ่งเป็นพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีการลองผิดลองถูก (Trial and Error) เพื่อให้ค่า PV เข้าสู่ค่า SP ที่ต้องการ ผลการทดลองแยกตามกรณีต่าง ๆ แสดงดังตารางที่ 3.9 และ 3.10 ตามลำดับ

ตารางที่ 3.9 ผลการทดลองการควบคุมพลานต์โมเดลในแต่ละกรณีศึกษา (CIF)

Case	PID Tuning Parameter						% Error
	PID 1		PID 2		PID 3		
	Gain	Reset	Gain	Reset	Gain	Reset	
1	0.5	50	-	-	-	-	1%
2	0.5	100	-	-	-	-	1%
4	0.7	150	0.75	15	-	-	1%
5	0.5	50	0.5	100	-	-	0.5%
7	0.5	50	0.6	160	0.75	15	1%

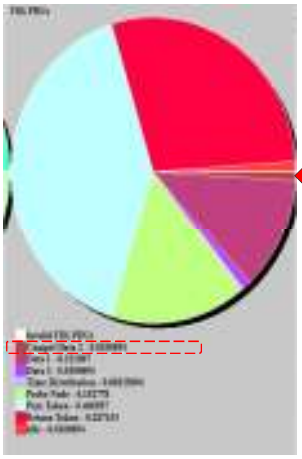
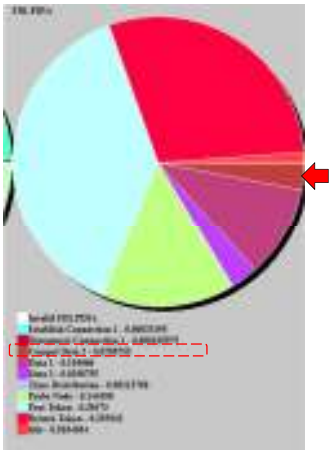
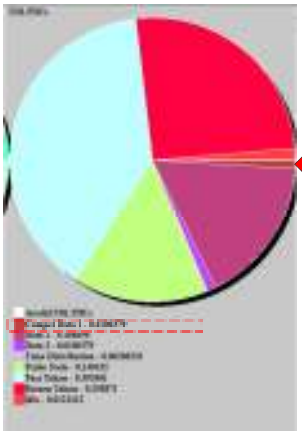
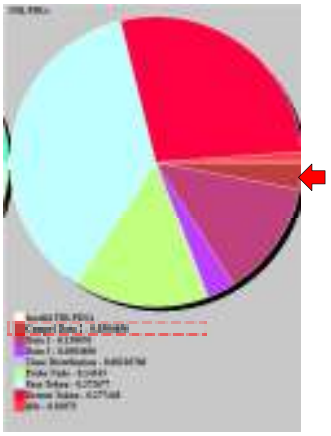
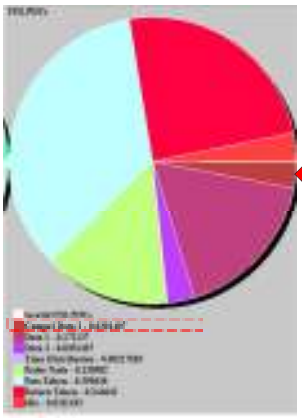
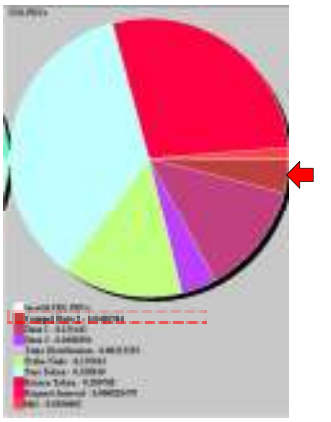
ตารางที่ 3.10 ผลการทดลองการควบคุมพลานต์โมเดลในแต่ละกรณีศึกษา (DCS)

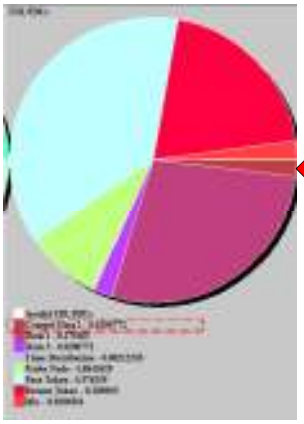
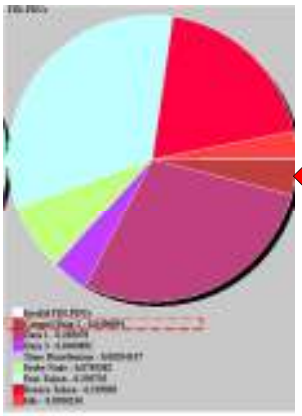
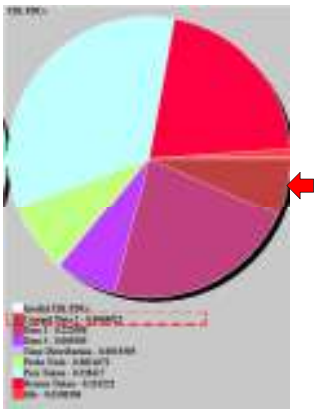
Case	PID Tuning Parameter						% Error
	PID 1		PID 2		PID 3		
	Gain	Reset	Gain	Reset	Gain	Reset	
1	0.5	100	-	-	-	-	1.5%
2	1.0	100	-	-	-	-	1%
4	0.5	270	1.2	25	-	-	2.5%
5	0.5	100	0.9	100	-	-	0.5%
7	0.5	100	0.5	270	1.2	25	1%

จากตารางที่ 3.9 และ 3.10 จะเห็นได้ว่าการควบคุมทั้งในแบบ Control in the Field และแบบที่กำหนดให้ตัวควบคุมอยู่ภายในดีซีเอส สามารถนำมาใช้ในการควบคุมกระบวนการได้อย่างมี

ประสิทธิภาพ ดังจะเห็นได้จากค่าความผิดพลาด (% error) มีค่าไม่เกิน 3% ซึ่งเป็นช่วงที่ยอมรับได้สำหรับในการทดลองนี้ ซึ่งค่า Network Load ของการควบคุมทั้งสองแบบแสดงดังตารางที่ 3.11

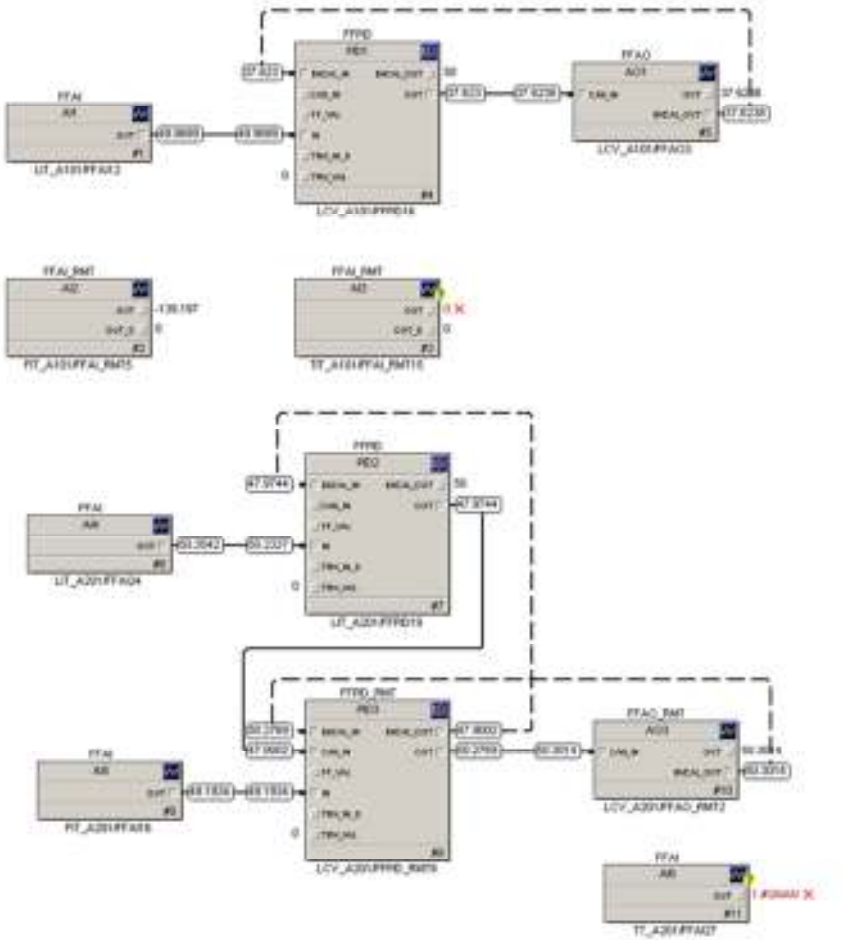
ตารางที่ 3.11 Network Load ของการควบคุมในแต่ละกรณีที่ได้จากโปรแกรม NI-FBUS Monitor

Case	Screen of NI-FBUS Monitor		Network Load (*)	
	CIF	DCS	CIF	DCS
1			10.07%	29.88%
2			10.04%	30.47%
4			30.14%	40.04%

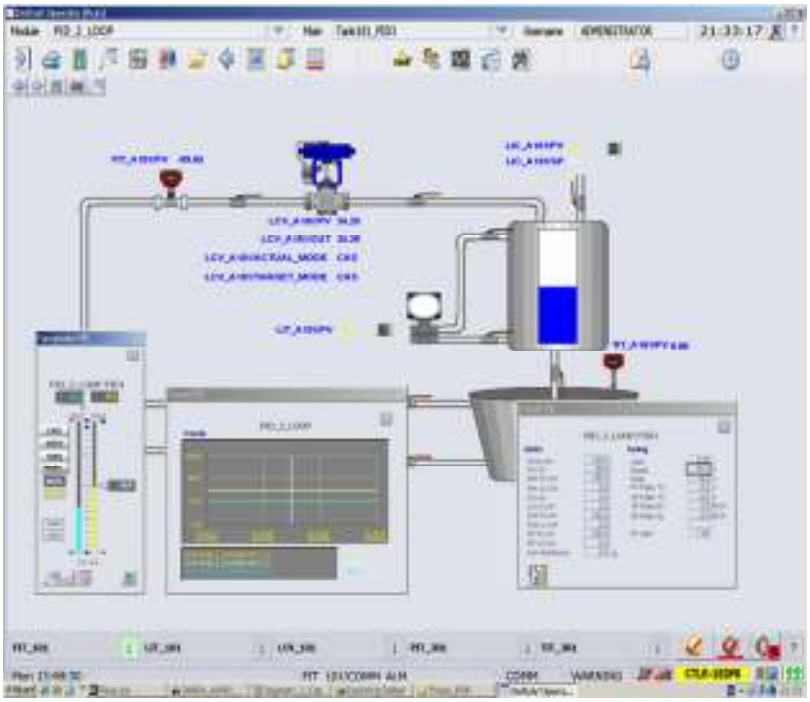
5			20.08%	60.51%
7			40.09%	66.60%

(*) ค่า Network Load คำนวณได้จากค่า Compel Data ที่โปรแกรม NI-FBUS Monitor สามารถตรวจจับได้ในขณะที่พลานต์โมเดลมีการทำงานจริง

จากตารางที่ 3.11 จะเห็นว่า Network Load สำหรับการรับ-ส่งข้อมูลแบบ Scheduled Communications ของการทดสอบกรณีศึกษาในกรณีที่กำหนดให้ตัวควบคุมอยู่ภายในอุปกรณ์ฟิลด์บัสหรือ Control in the Field จะมีการใช้แบนด์วิธบนเครือข่ายการติดต่อสื่อสารที่น้อยกว่ากรณีที่กำหนดให้ตัวควบคุมอยู่ภายในดีซีเอส และเมื่อนำไปเปรียบเทียบกับผลการทดลองในตารางที่ 3.6 ซึ่งได้จากการกำหนดตั้งค่าระบบแบบ Offline จะเห็นได้ว่ามีแนวโน้มไปในทิศทางเดียวกัน นอกจากนี้ จากกรณีศึกษาที่ 5 และ 7 ซึ่งเป็นการทดสอบการจัดมาโครไซเคิลของฟิลด์บัสเซกเมนต์ที่ประกอบด้วยสองวงควบคุมพบว่า การรับ-ส่งข้อมูลของ Scheduled Communications ที่ถูกควบคุมโดยระบบดีซีเอสจะให้ค่า Network Load ที่มากกว่า 60% ซึ่งถือเป็นค่าที่ไม่เหมาะสมตามคำแนะนำของ Foundation Fieldbus End Users Council Australia Inc. [3] ที่กำหนดให้จะต้องมีช่วงเวลาสำหรับ Unscheduled Communications อย่างน้อย 60% ซึ่งในกรณีนี้สามารถแก้ไขได้โดยการเพิ่มเวลาของมาโครไซเคิลให้มากกว่า 1 วินาที แต่ก็จะส่งผลทำให้ความเป็นเวลาจริง (Real Time) ของระบบลดลงเนื่องจากเวลาในการทำงานของวงควบคุมในแต่ละรอบเพิ่มขึ้นนั่นเอง ตัวอย่างผลการทดลองในการควบคุมการทำงานของพลานต์โมเดลแสดงดังรูปที่ 3.45 และ 3.46 ตามลำดับ



รูปที่ 3.45 การทำงานของฟังก์ชันบล็อกในโหมด Online ของการควบคุมในกรณีที่ 7



รูปที่ 3.46 แผนภาพกระบวนการทำงานของพลานต์โมเดล 2 ในการทดลองจริง

บทที่ 4

สรุปผลการวิจัยและข้อเสนอแนะ

4.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้นำเสนอเทคนิคในการนำคุณสมบัติการควบคุมที่ระดับตัวอุปกรณ์หรือ Control in the Field ซึ่งเป็นคุณลักษณะพิเศษของเทคโนโลยีฟาวน์เดชันฟิลด์บัส และการจัดมาโครไซเคิลมาใช้ในการควบคุมกระบวนการโดยผ่านทางกรณีศึกษาที่มีการควบคุมแบบ PID และแบบคาสเคดทั้งหมด 8 กรณีด้วยกัน ในขั้นแรกได้ทำการศึกษาการจัดมาโครไซเคิลเพื่อคำนวณหา Network Load ของการรับ-ส่งข้อมูลที่เป็นแบบ Scheduled Communications ของแต่ละกรณีศึกษาผ่านทางการกำหนดการตั้งค่าระบบแบบ Offline โดยระบบดีซีเอส Centum VP และ DeltaV ตามลำดับ ขั้นตอนต่อมาเป็นการทดลองการควบคุมพลานต์โมเดลเพื่อตรวจสอบ Network Load ที่เกิดขึ้นจากการทำงานจริงผ่านทางโปรแกรม NI-FBUS Monitor เพื่อเป็นการยืนยันผลการศึกษาการจัดมาโครไซเคิลที่ได้จากการกำหนดตั้งค่าระบบแบบ Offline ซึ่งผลจากการทดลองทั้งในแบบ Offline และการทดลองควบคุมพลานต์โมเดลพบว่า การควบคุมที่ระดับตัวอุปกรณ์ของฟาวน์เดชันฟิลด์บัส ใช้ช่วงเวลาในการรับ-ส่งข้อมูลแบบ Scheduled Communications ที่สั้นกว่าการควบคุมที่ดีซีเอส ดังตัวอย่างเช่นผลของการทดลองในกรณีที่ 5 และกรณีที่ 7 ซึ่งประกอบด้วยวงควบคุมสองวงบนฟาวน์เดชันฟิลด์บัสเซกเมนต์ Network Load ของการควบคุมแบบ Control in the Field มีค่าเท่ากับ 20.08% และ 40.09% ตามลำดับ ในขณะที่ Network Load ของการควบคุมบนดีซีเอส มีค่าเท่ากับ 60.51% และ 66.60% ตามลำดับ ซึ่งก็หมายความว่าช่วงเวลาว่างสำหรับการรับ-ส่งข้อมูลแบบ Unscheduled Communications ของมาโครไซเคิลที่เกิดจากการควบคุมแบบ Control in the Field จะมากกว่าการควบคุมที่ดีซีเอสด้วย ดังนั้นจึงถือได้ว่าการควบคุมแบบ Control in the Field ของกระบวนการที่ใช้เทคโนโลยีฟาวน์เดชันฟิลด์บัสนั้นสามารถเพิ่มประสิทธิภาพให้กับระบบควบคุมได้ เนื่องจากจะทำให้เกิดการปรับปรุงการจัดตารางเวลาในการสื่อสารของฟาวน์เดชันฟิลด์บัสเซกเมนต์โดยการทำให้มาโครไซเคิลมีช่วงเวลาสำหรับ Scheduled Communications สั้นลงนั่นเอง

4.2 ข้อเสนอแนะในการทำวิจัยต่อ

เทคนิคที่นำเสนอในวิทยานิพนธ์เล่มนี้ เป็นการศึกษาการจัดมาโครไซเคิลโดยผ่านทางการควบคุมกระบวนการในสองรูปแบบคือการควบคุมแบบ PID และการควบคุมแบบคาสเคดซึ่งเป็นรูปแบบที่เหมาะสมกับการควบคุมแบบ Control in the Field ตามคำแนะนำของ Fieldbus Foundation เท่านั้น ซึ่งในความเป็นจริงแล้วระบบควบคุมอัตโนมัติที่ใช้ในการควบคุมกระบวนการในงานอุตสาหกรรมโดยทั่วไปยังมีรูปแบบการควบคุมในลักษณะอื่น ๆ อีกหลายวิธี ซึ่งการจัดมาโครไซเคิลของการควบคุมโดยระบบดีซีเอสของรูปแบบการควบคุมเหล่านั้นก็เป็นอีกแนวทางหนึ่งในการปรับปรุงประสิทธิภาพของระบบควบคุมที่น่าศึกษา นอกจากนี้แนวทางในการเลือกใช้และการกำหนดจำนวนของอุปกรณ์เครื่องมือวัดภายในวงควบคุมแต่ละวง เพื่อปรับปรุงความเร็วในการทำงานให้

เหมาะสมกับระดับความสำคัญของวงควบคุมแต่ละวงภายในกระบวนการก็เป็นสิ่งที่ควรศึกษาเช่นกัน เพื่อทำให้การปรับปรุงคุณภาพในการควบคุมระบบมีความสมบูรณ์มากยิ่งขึ้น

เอกสารอ้างอิง

[1] William R. Hodson. “Optimizing Fieldbus Link Schedules Makes a Difference!” **ISA EXPO**, 25-27 October 2005

[2] Amphawan Julsereewong, Prasit Julsereewong and Thipat Dejsiriphant. “Study on Field Control Locations for Foundation Fieldbus-Based Cascade Control Loop” **SICE Annual Conference**, 20-23 August 2012

[3] Phil Eastwood. “Specification and Implementation of a Fieldbus Project” **Jump Aboard**, 21-23 March 2007

[4] ประสิทธิ์ จุลเสรีวงศ์. **ระบบอัตโนมัติทางอุตสาหกรรม.** พิมพ์ครั้งที่ 2. กรุงเทพฯ : โรงพิมพ์ มีน เซอร์วิส ซัพพลาย. 2554

[5] Carlos A. Smith, Armando Corripio. **Principles and Practice of Automatic Process Control**. 3th ED. John Wiley & Sons (Asia) Pte. Ltd. 2006

[6] ทวิช ชูเมือง. **การออกแบบระบบเครื่องมือวัดและควบคุมทางอุตสาหกรรม เล่ม 1**. กรุงเทพ ฯ : โรงพิมพ์เอ็ช เอ็น กรุ๊ป. 2549

[7] Fieldbus Foundation. “Foundation Fieldbus Technical Overview”, Austin, USA. 2003

[8] Rosemount Inc. “Foundation Fieldbus Blocks”, Minnesota, USA. 2000

ภาคผนวก
บทความวิจัยที่ได้รับการตีพิมพ์

บทความวิจัยที่ได้รับการตีพิมพ์ในวารสารทางวิชาการระดับนานาชาติในวิทยานิพนธ์นี้มีรายละเอียดดังต่อไปนี้

[1] Amphawan Julsereewong, Prasit Julsereewong, Piyaporn Harntujit, “Foundation Fieldbus Segment Scheduling For Fast Loop Control”, **ICIC Express Letters**, vol.7, no. 4, pp. 1201-1207

Volume 7, Number 4, April 2013 ISSN 1881-803X

ICIC Express Letters

An International Journal of Research and Surveys

Editors-in-Chief
Yan Shi, Tokai University, Japan
Junzo Watada, Waseda University, Japan

Indexed by *EI Compendex (Elsevier)*
Scopus (Elsevier)
INSPEC (IET)

Published by ICIC International
http://www.ijicic.org/icicel.htm

ICIC Express Letters
Volume 7, Number 4, April 2013

ICIC International ©2013 ISSN 1881-803X
pp. 1201–1207

FOUNDATION FIELDBUS SEGMENT SCHEDULING FOR FAST LOOP CONTROLS

Amphawan Julsereewong, Prasit Julsereewong and Piyaporn Harntujit

Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang, Bangkok, 10520, Thailand
{ kamphaw; kjprasit }@kmitl.ac.th; harnutjt.p@hotmail.com

Received June 2012; accepted August 2012

Abstract. *In this article, a technique to obtain fast loop controls using Foundation Fieldbus-based control in the field strategy is presented. Macrocycle schedules for running a single control loop and running two control loops on a segment are examined. Performances of the proposed technique were studied through calculated macrocycle timing schedules produced by the DeltaV and CenumVP systems. Surpassing the traditional Distributed Control System (DCS) approach with control in the host, the proposed technique offers significant improvement in segment schedules with minimizing network scheduled communications.*

Keywords: Foundation Fieldbus, Segment scheduling, Macrocycle, Control in the field, Control in the host, DCS, Communications

1. Introduction. Original pneumatic control loops were implemented with Control in the Field (CIF), which involves distributing control functionality among instruments at the device level. In those days, all field devices used pneumatic 3-15 psig signal to communicate process variables and control signals, and every control loop had its own pneumatic controller [1]. By the mid-1990s, the new breed of smart field instrumentation was changing the outlook for end users of conventional Distributed Control System (DCS). Smart filed instruments not only perform a basic measurement or actuation but also have functions for control and asset management [2]. Foundation Fieldbus proposed by the Fieldbus Foundation is fully compliant with the international IEC 61158/61784/61804 standards, and thus, the only advanced digital bus technology in the market enables users to move control functionality to field devices using function blocks [3]. The Fieldbus Foundation has defined sets of standard function blocks and supports the standardization of these definitions; whereby each specification allows manufacturers to make their devices with competitive advantage and to add parameters that support features unique to their own products. The performances of these blocks are tested by the Foundation when devices are registered. Therefore, devices bearing the Foundation registration checkmark logo can be connected on the same network and exchanged information without loss of degree of integration. The most used function blocks for implementing basic control loops are analog input (AI) blocks, analog output (AO) blocks, and control algorithm PID blocks. During control strategy configuration, there is usually some freedom where PID block may be located either in the DCS controller (referred to as the host for Fieldbus installations) or in intelligent field devices. For basic PID and cascade loops on non-critical applications, the field-level control without participation of a host system except for initialization and supervision is preferred whenever possible [4]. In addition, CIF can be used to control individual loops or as part of a DCS-based advanced control scheme [5]. Implementing CIF also enables greater flexibility in plant automation strategies, and controllers are free to handle higher level control functions such as advanced control and optimization [6].

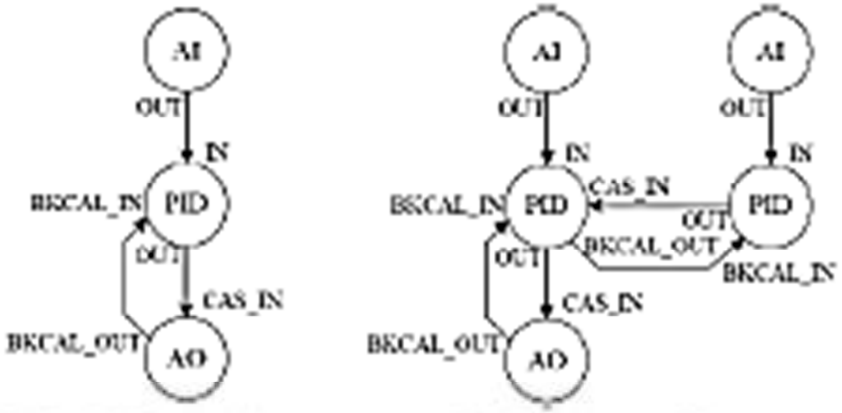
Applications where CIF strategy is especially effective include fast process applications such as fluid flow, liquid pressure loops, and some temperature loops. However, most end users today currently do not consider application of CIF because of the perception that it somehow introduces increased risk or reduced availability.

This paper deals with the technique to achieve fast loop controls using Foundation Fieldbus with CIF. Scheduling for running single control loop and two control loops on the segment during the software configuration is studied. The DeltaV and CentumVP systems, distributed by Emerson Process Management and Yokogawa, respectively, were used as the configuration software tools. Comparison results between produced macrocycle timing schedules based on field-level and host-level control strategies are also included.

2. Control Strategy Configuration Overview.

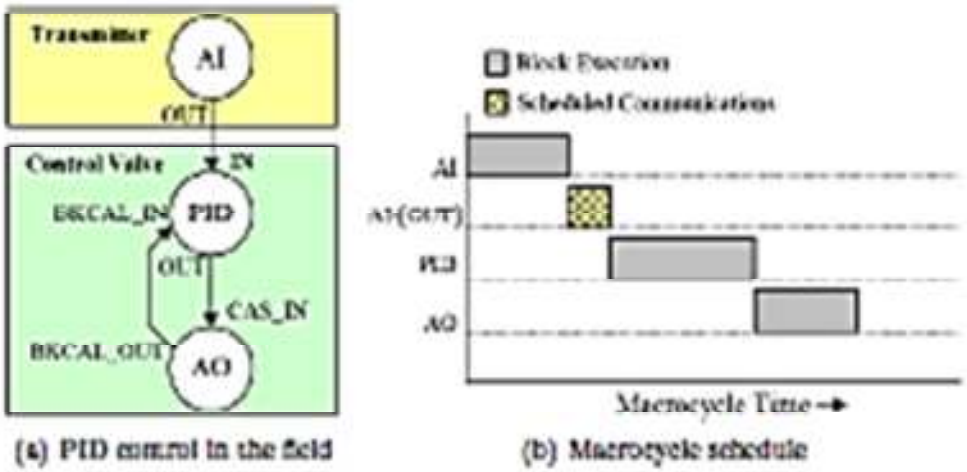
2.1. Fieldbus control loops. Foundation Fieldbus-based control loops are created by selecting function blocks and linking them together, so process and manipulated variables can be passed from one block to the next [3]. Figures 1(a) and 1(b) show the function block diagrams for basic PID control loop and cascade control loop, respectively.

2.2. Macrocycle timing schedule. By implementing PID control in the field as shown in Figure 2(a), Figure 2(b) displays a schedule for a transmitter with an AI block and a control valve with PID and AO blocks. Each function block in the control loop is executed at a specified rate within a Fieldbus device. Each time the block output is passed to the



(a) Basic PID control loop (b) Cascade control loop

FIGURE 1. PID and cascade loops represented as function block diagrams



(a) PID control in the field (b) Macrocycle schedule

FIGURE 2. Control in the field macrocycle with one PID loop

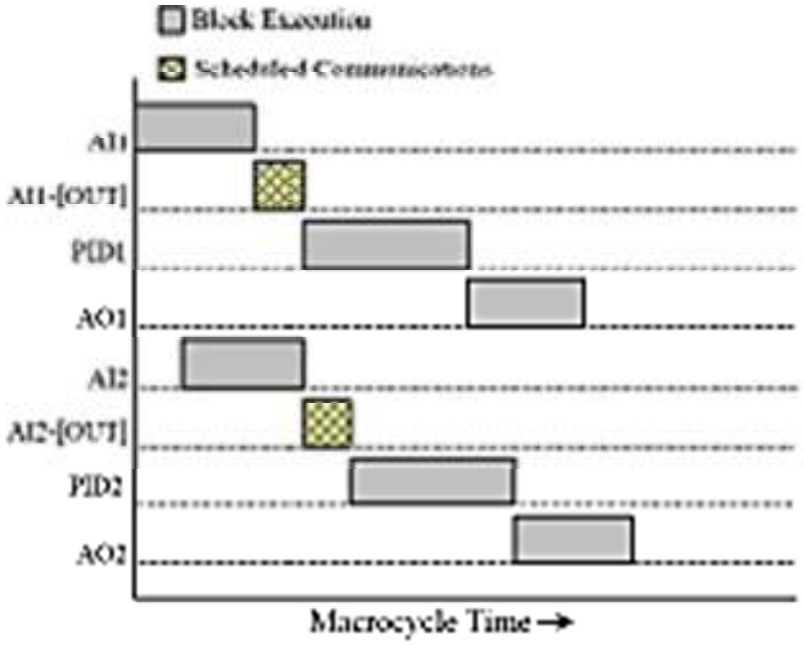


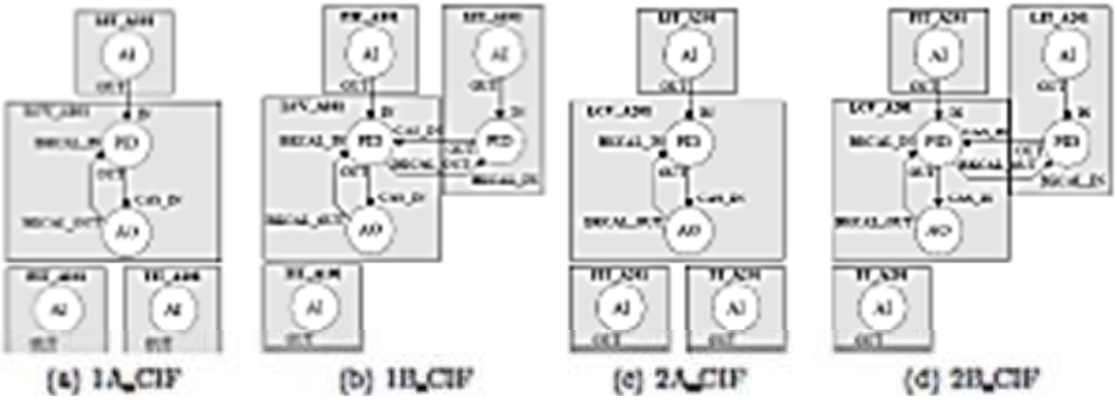
FIGURE 3. Control in the field macrocycle with two PID loops on the same segment

input of the next block. The period during which the function blocks are being executed is called a macrocycle. External function block links are communicated on the network, whereas internal links are passed to the next block within the device. Scheduling is the most common way to coordinate function block execution. Adding a second loop to the same segment may result in a longer macrocycle time, but not twice as long (see Figure 3) [7,8]. The macrocycle time can be estimated during definition of combining function blocks after the Fieldbus segment has been configured. Macrocycle calculation depends on several factors in system configuration such as number and type of Fieldbus devices, function block execution time, function block execution order, and the time required to transfer data between blocks (scheduled communications).

3. Studied Fieldbus Control Loops. Figures 4 and 5 show Fieldbus loops performing monitoring and control functionality for configuring field-level and host-level control strategies in one Fieldbus H1 segment, respectively. When creating control loops with CIF, the PID block locations are assigned to obtain the minimum number of scheduled communications over the network [9]. Table 1 summarizes the Fieldbus device details including tag, model, device revision, numbers of AI, AO, and PID blocks contained in the device, and block execution times.

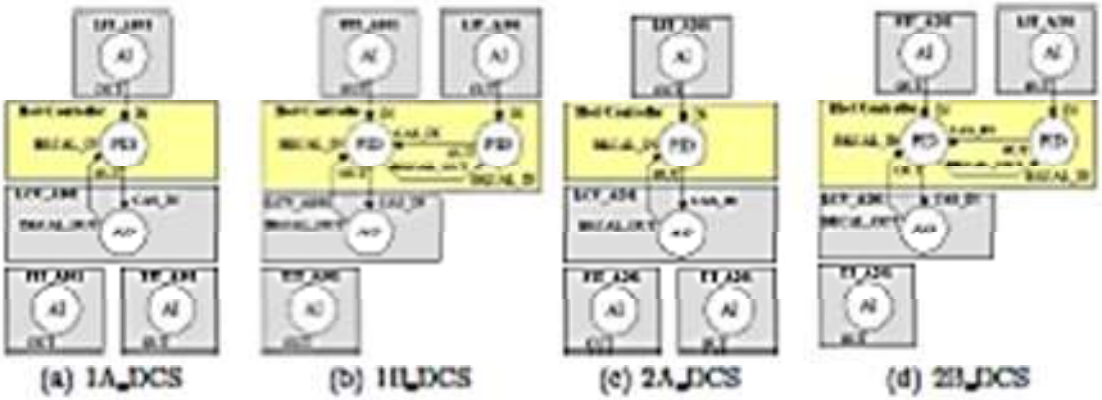
TABLE 1. Details of Foundation Fieldbus devices used for control strategy configuration

Tag	Model	Rev.	AI count	AI time	AO count	AO time	PID count	PID time
TIT_A101	3144P	2	3	60 ms	N/A	N/A	2	90 ms
LCV_A101	AVP323	2	N/A	N/A	1	75 ms	2	135 ms
FIT_A101	8732E	2	1	10 ms	N/A	N/A	1	10 ms
LIT_A101	EJX110A	3	3	30 ms	N/A	N/A	1	45 ms
TT_A201	STT35F	4	1	23 ms	N/A	N/A	1	90 ms
LCV_A201	DVC6003F	1	1	25 ms	1	25 ms	1	50 ms
FIT_A201	AXF	1	1	30 ms	N/A	N/A	1	50 ms
LIT_A201	DSTJ	1	2	75 ms	N/A	N/A	1	125 ms



(a) 1A_CIF (b) 1B_CIF (c) 2A_CIF (d) 2B_CIF

FIGURE 4. Fieldbus loops for field-level control strategy configuration



(a) 1A_DCS (b) 1B_DCS (c) 2A_DCS (d) 2B_DCS

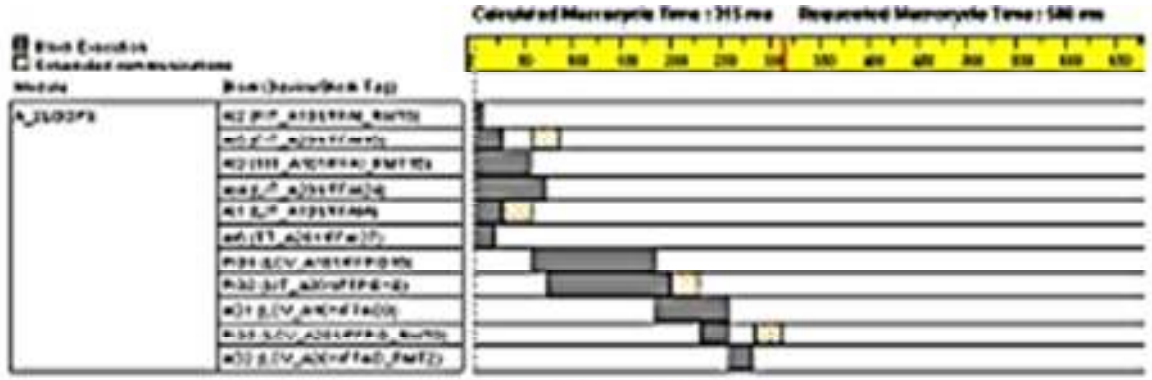
FIGURE 5. Fieldbus loops for host-level control strategy configuration

TABLE 2. Calculated macrocycle times for field-level control strategies

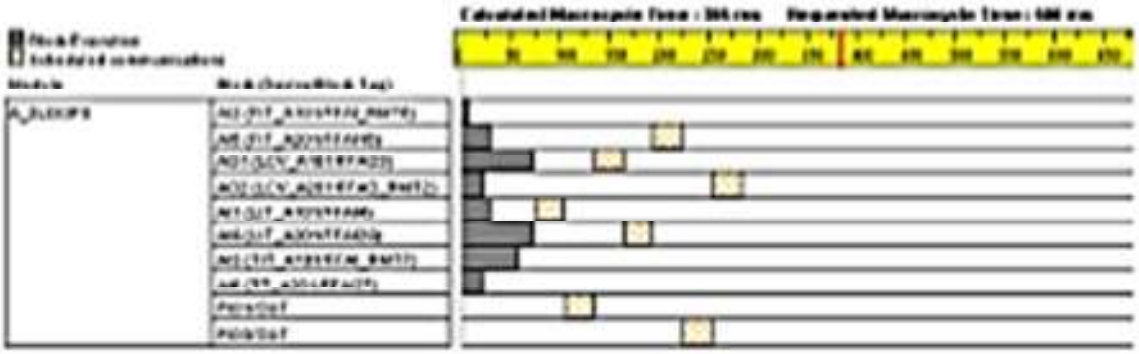
Case	Control loop(s) in the Fieldbus segment	Number of scheduled comm.	Calculated macrocycle time	
			DeltaV	CentumVP
1	1A_CIF	1	260 ms	263 ms
2	2A_CIF	1	160 ms	163 ms
3	1B_CIF	3	335 ms	308 ms
4	2B_CIF	3	315 ms	294 ms
5	1A_CIF + 2A_CIF	2	260 ms	263 ms
6	1B_CIF + 2B_CIF	6	460 ms	308 ms
7	1A_CIF + 2B_CIF	4	315 ms	294 ms
8	1B_CIF + 2A_CIF	4	335 ms	308 ms

TABLE 3. Calculated macrocycle times for host-level control strategies

Case	Control loop(s) in the Fieldbus segment	Number of scheduled comm.	Calculated macrocycle time	
			DeltaV	CentumVP
1	1A_DCS	3	215 ms	202 ms
2	2A_DCS	3	215 ms	122 ms
3	1B_DCS	4	245 ms	202 ms
4	2B_DCS	4	245 ms	154 ms
5	1A_DCS + 2A_DCS	6	335 ms	234 ms
6	1B_DCS + 2B_DCS	8	415 ms	323 ms
7	1A_DCS + 2B_DCS	7	385 ms	266 ms
8	1B_DCS + 2A_DCS	7	385 ms	266 ms



(a) Control in the field



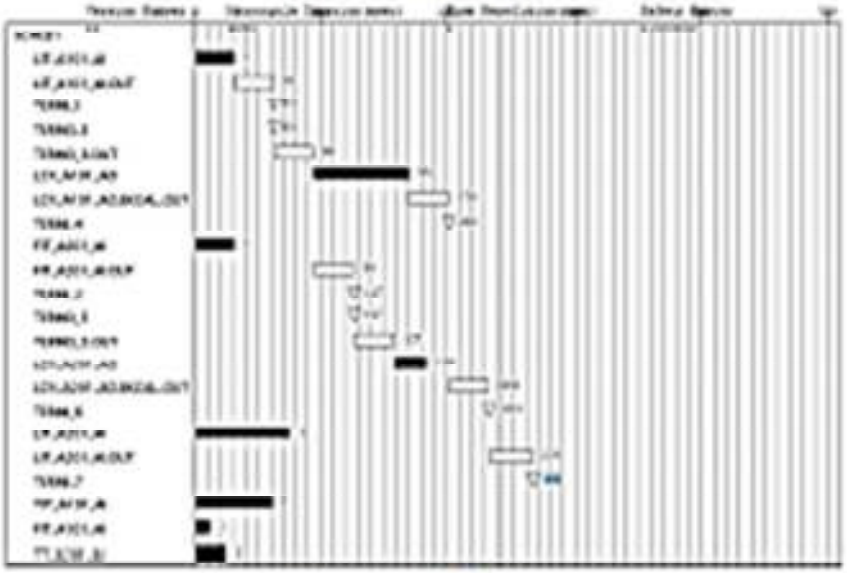
(b) Control in the host

FIGURE 6. Macrocycles produced by DeltaV for running loops of Case 7

4. Calculated Macrocycle Times. In order to estimate network loads for scheduled communications for field-level and host-level control strategies, the DelaV and CenturnVP systems were employed as the configuration tools. Tables 2 and 3 give the macrocycle times calculated by the DeltaV and CenturnVP systems for field-level and host-level control strategies, respectively. It should be noted that the calculated macrocycle times for traditional DCS control in Table 3 exclude the controller execution times. In addition, one must be aware that the execution of the PID algorithm in the host is typically not synchronized with the macrocycle execution on the Fieldbus segment. This means that a change in controller output to the control valve may not be sent until the following macrocycle. Segment macrocycle schedules produced by the DeltaV and CenturnVP for running PID and cascade control loops in the same segment of Case 7 are shown in Figures 6 and 7, respectively.

5. Comparison Results. Table 4 summarizes the estimated values of consumed bandwidths or communication loads for scheduled data transfers during one calculated macrocycle, where the default values of minimum schedule spacing in the DeltaV and CenturnVP are 30 ms and 31 ms, respectively. It is evident that for implementing control loops with traditional DCS approach, macrocycle schedules produced by the DeltaV and CenturnVP consume higher bandwidths for scheduled communications to publish function block outputs on the network. This will considerably reduce the control response period. CIF strategy can make the network communications more efficient because of fewer publish links per macrocycle. Leaving a portion of the macrocycle free of scheduled communications permits the system to perform other network functions such as updating graphics and faceplates as well as diagnostic information for maintenance more rapidly [7]. In addition, when using CIF strategy, one field device installed in the H1 segment may be defined as a backup LAS (Link Active Scheduler). This field device takes charge of all

(a) Control in the field



(b) Control in the host

FIGURE 7. Macrocycles produced by CenturnVP for running loops of Case 7

TABLE 4. Estimated values of consumed bandwidths for scheduled communications

Case	Number of scheduled communications		Consumed Bandwidth			
			DeltaV		CenturnVP	
	CIP	DCS	CIP	DCS	CIP	DCS
1	1	3	11.538%	41.860%	11.787%	46.040%
2	1	3	18.750%	41.860%	19.018%	76.230%
3	3	4	26.866%	48.980%	30.198%	61.390%
4	3	4	28.571%	48.980%	31.633%	67.391%
5	2	6	23.077%	50.704%	23.574%	79.487%
6	6	8	39.130%	57.831%	60.390%	76.780%
7	4	7	35.056%	54.545%	42.177%	81.579%
8	4	7	35.821%	54.545%	40.260%	81.579%

network communication in the segment in the event that the connection to the host system is lost. This is useful if, for example, the pressure or temperature in the system must be maintained, as in the case of important control loop which must remain in operation.

6. Conclusion. Foundation Fieldbus segment scheduling for field-level and host-level control strategies has been described in this paper. Major benefits provided by control in the field are reduced process controller loading, reduced network traffic enabling more loops per segment and increasing network availability, as well as improved control loop integrity and performance.

REFERENCES

[1] P. W. Murrill, *Fundamentals of Process Control Theory*, ISA, USA, 2000.

[2] H. Sasajima, Intelligent field devices and fieldbus solutions in PA industries, *SICE Annual Conference*, pp.1473-1478, 2004.

[3] Fieldbus Foundation, *Foundation Fieldbus Technical Overview*, USA, 2002.

[4] Fieldbus Foundation, *Foundation Fieldbus System Engineering Guidelines*, USA, 2010.

[5] Emerson Process Management, Answer to your questions about implementing Fieldbus in DeltaV systems, *White Paper*, pp.1-17, 2007.

[6] ARC Advisory Group, The business value proposition of control in the field, *ARC White Paper*, pp.1-20, 2009.

[7] W. R. Hudson, Optimizing Fieldbus link schedules makes a difference! *ISA EXPO*, pp.1-9, 2005.

[8] Emerson Process Management, Plan control response requirement, *White Paper*, pp.1-6, 2010.

[9] A. Juberaswong, P. Juberaswong and T. Dejpiriphan, Study on field control location for Foundation Fieldbus-based cascade control loop, *SICE Annual Conference*, pp.1706-1711, 2012.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวปิยพร หารธุจิต
วัน-เดือน-ปีเกิด	วันที่ 12 กรกฎาคม 2529
ที่อยู่	70/41 หมู่ 3 ต.สัตหีบ อ.สัตหีบ จ.ชลบุรี โทร. 083-122-2737
ประวัติการศึกษา	สำเร็จการศึกษาระดับปริญญาตรีหลักสูตรวิศวกรรมศาตรบัณฑิต (เกียรตินิยมอันดับ 2) สาขาวิศวกรรมระบบควบคุม จากคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2551 และในปีการศึกษา 2554 ได้เข้าศึกษาต่อระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมอัตโนมัติ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง