ระบบควบคุมที่ทนต่อความผิดพลาดบนขบวนการที่มีความสำคัญสูง

FAULT TOLERANCE CONTROL SYSTEM IN HIGH PRIORITY PROCESS

จตุพร รอดคำทุย

JATUPORN RODKAMTUI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2549

ISBN 974-15-2775-6

FAULT TOLERANCE CONTROL SYSTEM IN HIGH PRIORITY PROCESS

JATUPORN RODKAMTUI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INSTRUMENTATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2775-6

หัวข้อวิทยานิพนธ์	ระบบควบคุมที่ทนต่อความผิดพลาดบนขบวนการที่มีความสำคัญสูง
นักศึกษา	นายจตุพร รอดคำทุย
รหัสนักศึกษา	46061713
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.วิทยา ทิพย์สุวรรณพร

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ นำเสนอระบบควบคุมที่ทนต่อความผิดพร่องสำหรับกระบวนการที่มีความสำคัญสูงเพื่อใช้ควบคุมกระบวนการให้สามารถทำงานได้อย่างต่อเนื่อง โดยหากระบบควบคุมที่กำลังควบคุมกระบวนการอยู่เกิดความผิดพลาดไม่สามารถควบคุมกระบวนการได้ ระบบควบคุมสำรองที่ทำงานร่วมกันจะต้องสามารถทำงานทดแทนภายในเวลาที่จะไม่ทำให้กระบวนการหยุดการทำงาน ซึ่งเวลาที่จะไม่ทำให้กระบวนการหยุดการทำงานจะต่างกันไปตามกระบวนการ วิธีการที่นำเสนอวิทยานิพนธ์นี้คือ การนำหลักการของระบบควบคุมที่ทนต่อความผิดพร่องแบบต่างๆ มาประยุกต์ใช้ให้เข้ากับระบบควบคุมแบบคอมพิวเตอร์และระบบเครือข่ายอินทราเน็ต รวมทั้งการจัดระบบของฮาร์ดแวร์ให้เหมาะสม และนำระบบที่ออกแบบไปทดสอบเพื่อหาค่าเวลาที่ดีที่สุดที่จะไม่ทำให้กระบวนการหยุดทำงาน โดยระบบควบคุมที่ออกแบบสามารถทำงานทดแทนกันได้ภายในเวลาที่ดีที่สุด 1/1000 วินาที ซึ่งเพิ่มความน่าเชื่อถือระบบควบคุมให้สูงขึ้นเหมาะสำหรับควบคุมกระบวนการที่มีความสำคัญสูง

Thesis Title	Fault Tolerance Control System in High Priority Process
Student	Mr. Jatuporn Rodkamtui
Student ID.	46061713
Degree	Master of Engineering
Programme	Instrumentation Engineering
Year	2006
Thesis Advisor	Assoc. Prof. Vittaya Tipsuwannaporn

ABSTRACT

This thesis presents the Fault Tolerant Control System for the High Priority process that is able to control the process continuously. While the process is discontinued because of some control system errors, the redundant control system must be able to work instead in the time length without process shutdown. The time length depends on the systems. The presentation methods of this thesis is using various principles of the Fault Tolerant Control System apply with the computer control systems, Intranet network and hardware for designing the system that is able to work together appropriately. Then the designed system is tested for searching the best time length without process shutdown that is 1 millisecond. Therefore the designed system can increase more reliability of the High Priority Process.

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....IX

สารบัญรูป.....X

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 สมมติฐานของการศึกษา.....2

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....2

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....2

1.6 ขอบเขตการวิจัย.....3

1.7 ขั้นตอนการศึกษา.....3

บทที่ 2 ทฤษฎีพื้นฐานวิศวกรรมความน่าเชื่อถือระบบ.....5

2.1 ความน่าเชื่อถือของระบบ.....6

2.2 การเชื่อมต่อกันของอุปกรณ์แบบต่างๆ.....6

2.2.1 ระบบที่เชื่อมต่อกันแบบอนุกรมกัน.....6

2.2.2 ระบบที่เชื่อมต่อกันแบบขนานกัน.....8

2.2.3 ระบบที่เชื่อมต่อกันแบบผสมระหว่างอนุกรมและขนาน.....9

2.2.4 ระบบที่เชื่อมต่อกันแบบมีส่วนซ้ำสำรองสามโมดูล.....10

2.3 ระบบที่มีโมดูลซ้ำสำรอง.....13

2.3.1 การทำงานของโมดูลซ้ำสำรองแบบทำงานพร้อมกัน.....13

2.3.2 การทำงานของโมดูลที่ซ้ำสำรองแบบสำรองรอทำงาน.....14

2.4 การสับเปลี่ยนของระบบแบบโมดูลซ้ำสำรอง.....14

2.4.1 การสับเปลี่ยนของโมดูลที่สมบูรณ์แบบ.....14

2.4.1 การสับเปลี่ยนของโมดูลที่สมบูรณ์แบบ.....15

2.5 ระบบโหวตคะแนน.....15

สารบัญ(ต่อ)

หน้า

2.6 การหาความน่าเชื่อถือระบบด้วยวิธีมินิมัลคัทเซ็ท.....15
2.7 ความน่าเชื่อถือระบบซ้ำสำรองที่ทำงานแบบมีเงื่อนไข.....17
2.7.1 ความน่าเชื่อถือระบบที่มีโมดูลซ้ำสำรองเชื่อมต่อกันแบบขนาน 2 โมดูล.....17
2.7.2 ความน่าเชื่อถือระบบที่มีโมดูลซ้ำสำรองเชื่อมต่อกันแบบขนาน 3 โมดูล.....18
2.8 ความน่าเชื่อถือระบบซ้ำสำรองที่ทำงานโดยลำดับ.....20
2.9 ความน่าเชื่อถือในระบบคอมพิวเตอร์.....22
2.9.1 คำจำกัดความของความน่าเชื่อถือในระบบคอมพิวเตอร์.....22
2.9.2 สาเหตุความล้มเหลวในระบบคอมพิวเตอร์.....23
2.9.3 วิธีการปรับปรุงความน่าเชื่อถือของซอฟต์แวร์.....24
2.9.4 แบบจำลองความน่าเชื่อถือของซอฟต์แวร์.....26
2.9.5 การป้องกันความล้มเหลวในระบบคอมพิวเตอร์.....26

บทที่ 3 การจัดโครงสร้างระบบที่ทนต่อความผิดพร่องในการทดลอง
3.1 พื้นฐานระบบที่ทนต่อความผิดพร่อง.....29
3.2 ระบบที่ทนต่อความผิดพร่องแบบมีโมดูลซ้ำสำรอง 3 โมดูล.....31
3.3 การโหวตคะแนน.....32
3.4 วิธีคำนวณหาค่าอัตราการล้มเหลว.....34
3.5 การควบคุมความผิดพลาด.....35
3.6 โครงสร้างการทดลองระบบความคุมที่ทนความผิดพร่อง.....37
3.6.1 การจัดระบบของสถานีควบคุม.....38
3.6.2 การจัดระบบของสถานีเอชเอ็มไอ.....40
3.6.3 การจัดแบบจำลองกระบวนการควบคุมความดัน.....41

บทที่ 4 การหาค่าความน่าเชื่อถือของระบบและผลการทดลอง.....43
4.1 แบบจำลองที่ใช้หาความน่าเชื่อถือของระบบ.....43
4.2 การจัดรูปแบบการควบคุม.....44
4.3 การหาความน่าเชื่อถือของระบบในส่วนของฮาร์ดแวร์.....47
4.4 การหาความน่าเชื่อถือของระบบในส่วนของซอฟต์แวร์.....50
4.4.1 ความล้มเหลวและความผิดพร่อง.....51

สารบัญ(ต่อ)

หน้า

4.4.2 หลักการพื้นฐานสำหรับการวัดความน่าเชื่อถือของซอฟต์แวร์.....51
4.4.3 แบบจำลองความน่าเชื่อถือของซอฟต์แวร์.....52
4.4.4 วิธีทดสอบซอฟต์แวร์ด้วยวิธีกราฟ.....56
4.4.5 การหาค่าความน่าเชื่อถือซอฟต์แวร์ในการทดลอง.....56
4.5 การวิเคราะห์ความน่าเชื่อถือในโครงสร้างระบบที่ออกแบบ.....63
4.6 ผลการทดลองจากการสังเกตพฤติกรรมของสัญญาณ.....68
4.7 สรุปผลการทดลอง.....75

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....76

บรรณานุกรม.....78

ภาคผนวก.....80
ภาคผนวก ก. ข้อมูลทางเทคนิคของหน่วยอีเทอร์เน็ตไอ/โอ.....81
ภาคผนวก ข. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่.....85

ประวัติผู้เขียน.....91

สารบัญตาราง

ตารางที่ หน้า

2.1 การลงคะแนนของตัวลงคะแนนเอาท์พุต....10

3.1 ตัวอย่างการคำนวณอัตราการล้มเหลว ตามมาตรฐาน MIL-HDBK-217B....35

4.1 เงื่อนไขการลงคะแนนเพื่อเลือกข้อมูลส่งออกเอาท์พุต....44

4.2 ตารางความเป็นจริง....45

4.3 การใช้แผนที่คาร์นอฟเพื่อลดรูปสมการ....45

4.4 ตารางความเป็นจริงตามเงื่อนไขโหวตคะแนน....46

4.5 การใช้แผนที่คาร์นอฟเพื่อลดรูปสมการในงานวิจัย....46

4.6 อัตราการล้มเหลวที่ใช้ในการทดลอง....48

4.7 ข้อมูลการคำนวณหาค่า Failure Rate....49

4.8 การจำแนกกลุ่มของแบบจำลองความน่าเชื่อถือสำหรับซอฟต์แวร์....55

4.9 จำนวนความล้มเหลวสะสมที่เกิดขึ้นในช่วงเวลากระทำการเก็บข้อมูลทุกๆ 10000 รอบ การทำงานของซีพียูของสถานีเอชเอ็มไอ....59

4.10 จำนวนความล้มเหลวสะสมที่เกิดขึ้นในช่วงเวลากระทำการเก็บข้อมูลทุกๆ 10000 รอบ การทำงานของซีพียูของสถานีควบคุม....60

4.11 จำนวนความล้มเหลวสะสมที่เกิดขึ้นในช่วงเวลากระทำการเก็บข้อมูลทุกๆ 10000 รอบ การทำงานของซีพียูของโปรแกรมโหวตคะแนน....62

สารบัญรูป

รูปที่ หน้า

2.1 คุณลักษณะของอุปกรณ์แบบ Bathtub Curve....6

2.2 แบบจำลองระบบที่ต่ออนุกรมกันจำนวน n หน่วย....6

2.3 แบบจำลองของระบบที่ต่อขนานกันจำนวน n หน่วย....8

2.4 ระบบที่ต่อแบบอนุกรม-ขนาน และการลดรูปวงจร....9

2.5 การต่อระบบแบบซ้ำสำรองสามโมดูล....10

2.6 ระบบที่มีส่วนสำรอง....13

2.7 การหาความน่าเชื่อถือแบบมีเงื่อนไข....16

2.8 สาเหตุหลักที่ก่อให้เกิดความล้มเหลวในระบบคอมพิวเตอร์....17

2.9 ระบบแบบซ้ำซ้อนสามส่วนที่มีปริภูมิสถานะแบบซ่อมแซมได้....27

3.1 โมดูลซ้ำสำรอง 3 โมดูลพร้อมทั้งตัวโหวต 3 โมดูล....31

3.2 โมดูลซ้ำสำรองที่มีตัวโหวต 3 โมดูลที่มีความไวต่อการล้มเหลวของตัวโหวต....32

3.3 รูปแบบการโหวตลงคะแนน โดยเงื่อนไขต้องมีอินพุท 2 ใน3 ทำงานได้....32

3.4 ฮาร์ดแวร์ของตัวโหวตที่มีการประสานเวลาของอินพุตและเอาท์พุต....33

3.5 แผนกำหนดเวลาการประสานเวลาของตัวโหวต....33

3.6 ช่วงเวลายอมรับได้ของกระบวนการโดยทั่วไป....36

3.7 บล็อกไดอะแกรมของระบบควบคุมคอมพิวเตอร์....37

3.8 การจัดโครงสร้างระบบในการทดลอง....37

3.9 ระบบเครือข่ายอีเทอร์เน็ตแบบมีโมดูลซ้ำสำรอง....38

3.10 การกำหนดค่าเลขที่อยู่ไอพีให้กับโปรแกรมจัดการที่พัฒนาขึ้น....39

3.11 การกำหนดค่าพารามิเตอร์บนแผงต่อประสานระบบเครือข่ายอีเทอร์เน็ต....41

3.12 การกำหนดพารามิเตอร์ของหน่วยแบบจำลองกระบวนการ....42

4.1 ไดอะแกรมการเชื่อมต่อกันของอุปกรณ์ในการทดลอง. 43

4.2 ระบบควบคุมที่ทนทานต่อความผิดพร่อง....44

4.3 กราฟโครงสร้างของซอฟต์แวร์ที่ออกแบบโดยทั่วไป....56

4.4 กราฟการทำงานของสถานีเอชเอ็มไอ....58

4.5 การทำงานแบบแยกส่วนของเอชเอ็มไอ....58

4.6 กราฟการทำงานของสถานีควบคุม....60

4.7 การทำงานตามโครงสร้างของโปรแกรมสถานีควบคุม....61

4.8 การทำงานตามโครงสร้างของซอฟต์แวร์โหวตคะแนน....62

สารบัญรูป(ต่อ)

รูปที่ หน้า

รูปที่ 4.9 กราฟแสดงผลจากกระบวนการในสถานีควบคุมเอชเอ็มไอ.....68
รูปที่ 4.10 กราฟแสดงผลจากกระบวนการที่สถานีควบคุมทั้งสาม.....69
รูปที่ 4.11 สัญญาณการควบคุมเมื่อทั้งสามสถานีควบคุมทำงานตามปกติ.....70
รูปที่ 4.12 สัญญาณการควบคุมเมื่อสถานีควบคุมที่สองหยุดการทำงานส่วนสถานีอื่นทำงาน.....71
รูปที่ 4.13 สัญญาณการควบคุมเมื่อสถานีควบคุมที่สองหยุดการทำงานส่วนสถานีอื่นทำงาน.....71
รูปที่ 4.14 สัญญาณการควบคุมเมื่อสถานีควบคุมที่สามหยุดการทำงานส่วนสถานีอื่นทำงาน.....72
รูปที่ 4.15 สัญญาณการควบคุมเมื่อสถานีที่หนึ่งและสามหยุดการทำงาน.....73
รูปที่ 4.16 สัญญาณการควบคุมเมื่อสถานีที่หนึ่งถึงสามกลับทำงานได้ตามลำดับ.....73
รูปที่ 4.17 สัญญาณการควบคุมเมื่อทุกสถานีหยุดทำงานและกลับทำงานได้ใหม่.....74

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบควบคุมกระบวนการอัตโนมัติเป็นปัจจัยที่สำคัญสำหรับงานทางด้านระบบการผลิตของโรงงานอุตสาหกรรม รวมทั้งระบบที่ให้บริการสาธารณูปโภค เช่น ระบบจ่ายไฟฟ้าและน้ำสำหรับกระบวนการผลิต หากระบบควบคุมกระบวนการอัตโนมัติดังกล่าวหยุดการทำงานย่อมจะทำให้เกิดความเสียหายต่อกระบวนการและผลผลิตที่อยู่ในกระบวนการ เสียโอกาสในการผลิต อีกทั้งยังอาจทำให้มีผลกระทบต่อกระบวนการอื่นๆ ที่เกี่ยวข้องกันอีกด้วย เครื่องควบคุมตรรกแบบที่โปรแกรมได้หรือพีแอลซี (PLC) [1] เป็นเครื่องควบคุมที่นิยมนำมาใช้ในการควบคุมกระบวนการต่างๆ เนื่องจากมีการพัฒนาจนสามารถใช้ควบคุมกระบวนการได้เป็นอย่างดี แต่เป็นระบบที่ต้องใช้ฮาร์ดแวร์พิเศษและบุคลากรที่มีความเชี่ยวชาญเฉพาะ ทำให้ค่อนข้างยุ่งยากต่อการใช้งานหากเครื่องควบคุมตรรกแบบที่โปรแกรมได้เกิดความล้มเหลวในการทำงาน

การพัฒนาเทคโนโลยีของเครื่องคอมพิวเตอร์ส่วนบุคคลได้มีการพัฒนาไปอย่างรวดเร็ว สามารถใช้งานได้ง่ายเนื่องจากคนส่วนใหญ่มีความคุ้นเคยกับเครื่องคอมพิวเตอร์อยู่แล้ว หากนำเครื่องคอมพิวเตอร์มาประยุกต์ใช้กับระบบอัตโนมัติทางอุตสาหกรรม เพื่อเพิ่มความน่าเชื่อถือให้กับระบบควบคุมกระบวนการ จะทำให้มีความสะดวกและยุ่งยากน้อยกว่าการใช้เครื่องควบคุมตรรกแบบที่โปรแกรมได้เมื่อเครื่องควบคุมแบบคอมพิวเตอร์เกิดความล้มเหลวในการทำงาน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้ มุ่งหวังเพื่อศึกษาเทคนิคและวิธีการใช้เครื่องคอมพิวเตอร์ส่วนบุคคลมาเพิ่มความน่าเชื่อถือในการควบคุมกระบวนการ โดยใช้เครื่องคอมพิวเตอร์เป็นเครื่องควบคุมกระบวนการผ่านระบบเครือข่ายอินทราเน็ต[2] เพื่อลดปัญหาการหยุดการทำงานของกระบวนการอันเนื่องมาจากความล้มเหลวของระบบควบคุม ทำให้โอกาสที่ระบบควบคุมจะเกิดความล้มเหลวในการทำงานน้อยลงและสามารถเพิ่มความน่าเชื่อถือให้กับระบบควบคุมให้เพิ่มมากขึ้น ลดความเสียหายและผลกระทบต่อกระบวนการ ลดค่าใช้จ่ายในการซ่อมบำรุง เป็นต้น

วัตถุประสงค์ของการศึกษาวิจัยระบบควบคุมที่ทนต่อความผิดพร่องสำหรับกระบวนการที่มีความสำคัญสูงบนระบบเครือข่ายอินทราเน็ต สามารถแบ่งเป็นหัวข้อต่างๆ ได้ดังนี้

1.2.1 ศึกษาการติดต่อสื่อสารกันผ่านระบบเครือข่ายอินทราเน็ตของเครื่องควบคุมแบบคอมพิวเตอร์

ซึ่งมีความสามารถในการติดต่อสื่อสารกับอุปกรณ์ที่มาจากหลายแหล่งผู้ผลิต ตลอดจนเครือข่ายที่ระบบคอมพิวเตอร์มีการพัฒนาที่หลากหลายกว่า อีกทั้งการควบคุมกระบวนการในปัจจุบันจะเห็นว่า แม้ว่าจะใช้เครื่องพีแอลซีเป็นตัวควบคุมกระบวนการ แต่การแสดงผลของกระบวนการและการสั่งการจากผู้ปฏิบัติงานก็จะทำบนเครื่องคอมพิวเตอร์ เช่น ในระบบเอชเอ็มไอ (Human Machine Interface) [14] ข้อดีของระบบคอมพิวเตอร์และวิธีการควบคุมกระบวนการในปัจจุบันจึงทำให้ในงานวิจัยนี้ ต้องการเสนอแนวความคิดที่จะนำระบบคอมพิวเตอร์มาออกแบบเป็นระบบควบคุมที่ทนต่อความผิดพร่อง

1.6 ขอบเขตการวิจัย

ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการเพิ่มความน่าเชื่อถือได้ให้กับระบบควบคุม แบบคอมพิวเตอร์ โดยการนำเอาเครื่องคอมพิวเตอร์ส่วนบุคคลมาต่อกันเป็นระบบเครือข่ายให้ทำงานบนระบบเครือข่ายอินทราเน็ตและสามารถทำงานทดแทนกันได้ ทั้งในส่วนที่เป็นเอชเอ็มไอและส่วนที่เป็นตัวควบคุม เพื่อควบคุมให้กระบวนการนั้นให้ ทำงานอย่างต่อเนื่อง โดยผลที่คาดว่าจะได้รับคือ ระบบที่ทำการควบคุมไม่เกิดความล้มเหลวในการทำงานเมื่อได้จัดการให้ระบบควบคุมแบบคอมพิวเตอร์ในเครือข่ายอินทราเน็ตทำงานทดแทนกันได้แล้ว

1.7 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

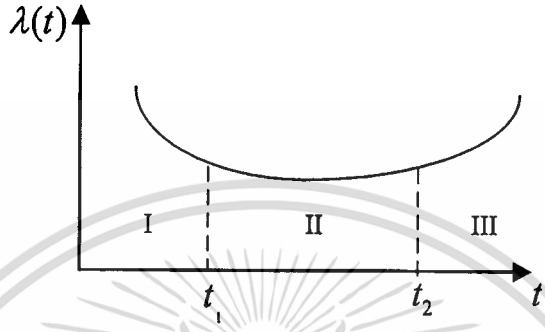
บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย และพื้นฐานของวิศวกรรมความน่าเชื่อถือ ซึ่งประกอบด้วย การต่อขององค์ประกอบในระบบในรูปแบบต่าง ได้แก่ การต่อแบบอนุกรม (Series Structure) แบบขนาน (Parallel Structure) แบบอนุกรมและขนาน (Series-Parallel Structure) แบบแยกความล้มเหลวร่วม (Triple Modular Redundancy) เป็นต้น การหาค่าความน่าเชื่อถือของระบบแบบต่างๆ การหาค่าความน่าเชื่อถือของระบบควบคุมแบบคอมพิวเตอร์ที่มีวงจรแบบซับซ้อน เช่น วิธีความน่าจะเป็นแบบมีเงื่อนไขและระบบมินิมัลคัทเซ็ท (Minimal Cut Set method) ตลอดจนระบบการทำงานทดแทนกันของระบบควบคุมแบบคอมพิวเตอร์ ซึ่งถูกใช้ในงานวิจัยนี้

บทที่ 3 กล่าวถึง พื้นฐานคุณสมบัติที่ควรมีของระบบที่ทนต่อความผิดพร่อง การประยุกต์ใช้งานระบบที่มีโมดูลซ้ำสำรอง 3 โมดูล วิธีการโหวต และวิธีการคำนวณหาค่าอัตราการล้มเหลวตามมาตรฐาน MIL-HDBK-217B [19] ตลอดจนการนำหลักการของระบบที่ทนต่อความผิดพร่องไปการออกแบบโครงสร้างการทดลอง

บทที่ 4 กล่าวถึง การหาค่าความน่าเชื่อถือของระบบที่ออกแบบในการทดลอง โดยแบ่งเป็น การวิเคราะห์หาค่าความน่าเชื่อถือของระบบในการทดลองโดยใช้วิธีแบบจำลองสมการทางคณิตศาสตร์ และความน่าเชื่อถือในแง่ของสัญญาณที่ได้จากการวัด

บทที่ 5 กล่าวถึง บทสรุปผลการวิจัยและข้อเสนอแนะ

หรือช่วงเสื่อมสภาพ (Wear-out, III) เป็นช่วงที่อุปกรณ์หรือระบบถูกใช้งานมาเป็นระยะเวลาที่ยาวนาน อาจจะมีปัจจัยอื่นมามีผลกระทบกับอุปกรณ์หรือระบบ ทำให้เกิดความล้มเหลวหรือความผิดพลาดในการทำงานมากขึ้น เช่น การกัดกร่อน หรือการเปลี่ยนอุปกรณ์บางส่วน ในช่วงนี้อัตราเสี่ยงต่อความล้มจะมีอัตราการเพิ่มขึ้นอย่างรวดเร็ว รูปร่าง Bathtub Curve ของแต่ละอุปกรณ์จะมีรูปร่างต่างกันไป



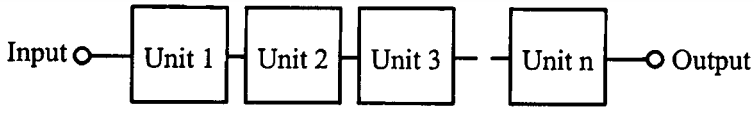
รูปที่ 2.1 คุณลักษณะของอุปกรณ์แบบ Bathtub Curve

2.2 การเชื่อมต่อกันของอุปกรณ์แบบต่างๆ

ระบบควบคุมที่ใช้ในการควบคุมกระบวนการจะมีความซับซ้อน เนื่องจากประกอบขึ้นจากระบบควบคุมหลายๆ ระบบที่มาทำงานให้สัมพันธ์กัน ซึ่งสามารถเขียนเป็นแบบจำลองการเชื่อมต่อของระบบนั้น ให้สามารถเข้าใจและศึกษาได้ง่ายขึ้นดังนี้คือ

2.2.1 ระบบหรืออุปกรณ์ที่เชื่อมต่อกันแบบอนุกรม

ระบบหรืออุปกรณ์พื้นฐานหากนำมาเชื่อมต่อกันแบบอนุกรมตามรูปที่2.2 เมื่อพิจารณาจะพบว่า อุปกรณ์หรือระบบที่เชื่อมต่อกันจะต้องสามารถทำงานได้ทุกหน่วย เพื่อที่จะทำให้ระบบโดยรวมอยู่ในสภาวะที่ทำงานได้ ถ้าอุปกรณ์หรือระบบพื้นฐานส่วนหนึ่งไม่สามารถทำงานได้ก็จะทำให้ระบบทั้งหมดหยุดการทำงานไปด้วย สามารถเขียนแบบจำลองความสัมพันธ์ได้ดังรูป



รูปที่ 2.2 แบบจำลองระบบที่ต่ออนุกรมกันจำนวน n หน่วย

ความน่าเชื่อถือของระบบระบบหรืออุปกรณ์ที่มีการเชื่อมต่อกันแบบอนุกรม สามารถหาความน่าเชื่อถือของระบบทั้งหมดได้จากสมการ (2.2)

$$R_s = P(E_1 \times E_2 \times E_3 \times \ldots \times E_n) \qquad (2.2)$$

กำหนดให้ R_s คือ ความน่าเชื่อถือของระบบ

E_n คือ เหตุการณ์ที่ระบบพื้นฐานที่ n จะไม่เกิดความล้มเหลว

$P(E_1 \times E_2 \times E_3 \times \ldots \times E_n)$ คือ ความน่าจะเป็นที่จะเกิดเหตุการณ์ E_1 ถึง E_n

หากระบบพื้นฐานที่ประกอบขึ้นเป็นอิสระต่อกัน คือ การที่ระบบพื้นฐานตัวใดตัวหนึ่งหยุดการทำงานไป จะไม่ส่งผลกระทบต่อความน่าเชื่อถือของระบบพื้นฐานตัวอื่นที่ต่อร่วมกัน ระบบแบบอนุกรมนี้ระบบพื้นฐานทั้งสองต้องทำงานได้ตามหน้าที่ของมัน จึงจะทำให้ระบบสามารถทำงานได้ตามปกติ หากกำหนดให้ $R_i = P(E_i)$ โดยที่ i จะมีค่าตั้งแต่ 1 ถึง n สามารถเขียนสมการได้ว่า

$$\begin{aligned} R_s &= R_1 \times R_2 \times R_3 \times \ldots \times R_n \\ &= \prod_{i=1}^{n} R_i \end{aligned} \tag{2.3}$$

จากสมการที่ (2.3) หากพิจารณาคุณลักษณ์ของอุปกรณ์สามารถอธิบายได้ด้วยสมการเลขชี้กำลัง (Exponential Equation) นั่นคือกำหนดให้ $R = e^{-\lambda t}$ ซึ่งสมการเลขชี้กำลังมักถูกนำมาใช้ในการแสดงพฤติกรรมของอุปกรณ์ โดยกำหนดให้ λ_i อัตราส่วนความล้มเหลวคงที่ และ t_i คือช่วงเวลาที่พิจารณาซึ่งเป็นช่วงเวลาที่ต้องการให้อุปกรณ์หรือระบบสามารถทำงานได้ไม่เกิดความล้มเหลว เมื่อแทนเลขชี้กำลังด้วย F_i จะทำให้ได้สมการคือ $R_i = e^{-Fi}$ จากสมการที่ (2.3) หากแทนค่า $R_1 = e^{-F1}, R_2 = e^{-F2}, \ldots, R_n = e^{-Fn}$ ลงในสมการที่ (2.3) สามารถเขียนใหม่ได้ว่า

$$\begin{aligned} R_s &= R_1 \times R_2 \times R_3 \times \ldots \times R_n \\ &= e^{-F1} \times e^{-F2} \times \ldots \times e^{-Fi} \times \ldots \times e^{-Fn} \\ &= e^{-(F1+F2+\ldots+Fi+\ldots+Fn)} \\ &= \exp\left(-\sum_{i=1}^{n} F_i\right) \end{aligned} \tag{2.4}$$

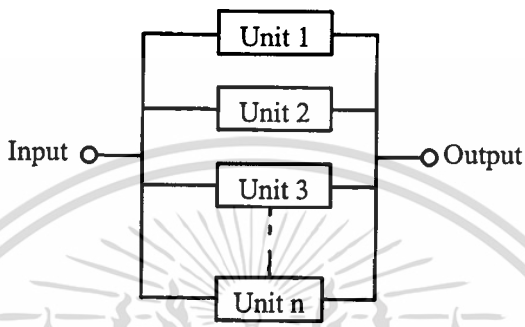
หาก $t_1 = t_2 = \ldots = t_n$ สามารถเขียนสมการที่ (2.4) ได้ใหม่ว่า

$$R_s = \exp\left(-t\sum_{i=1}^{n} \lambda_i\right) \tag{2.5}$$

ค่าความน่าเชื่อถือรวมของระบบจะมีค่าไม่มากไปกว่าค่าความน่าเชื่อถือที่น้อยที่สุดของระบบพื้นฐาน ดังนั้น ระบบที่มีรูปแบบการต่อแบบอนุกรมจึงต้องมีความน่าเชื่อถือของระบบพื้นฐานสูง โดยเฉพาะระบบที่มีจำนวนระบบหรืออุปกรณ์พื้นฐานต่อกันอยู่เป็นจำนวนมาก

2.2.2 ระบบหรืออุปกรณ์ที่เชื่อมต่อกันแบบขนาน

เมื่อนำระบบหรืออุปกรณ์พื้นฐานที่มีหน้าที่เหมือนกันมาเชื่อมต่อขนานกัน สามารถเรียกการต่อแบบนี้ได้อีกอย่างว่า ระบบทำงานทดแทนกัน นิยามความน่าเชื่อถือของระบบหรืออุปกรณ์ที่เชื่อมต่อกันแบบขนานกันคือ ระบบหรืออุปกรณ์ที่เชื่อมต่อกันแบบขนานจะทำงานได้ก็ต่อเมื่อมีระบบหรืออุปกรณ์พื้นฐานอย่างน้อยหนึ่งตัวสามารถทำงานได้แล้ว ระบบจะไม่เกิดความล้มเหลว สามารถเขียนแบบจำลองของระบบที่มีการเชื่อมต่อกันแบบขนานได้ดังรูปที่ 2.3



รูปที่ 2.3 แบบจำลองของระบบที่ต่อขนานกันจำนวน n หน่วย

การหาค่าความน่าเชื่อถือได้ของระบบที่มีการเชื่อมต่อกันของระบบหรืออุปกรณ์พื้นฐานแบบขนาน สามารถหาได้จากสมการ (2.6)

$$Q_p = P(\overline{E}_1 \times \overline{E}_2 \times \overline{E}_3 \times \ldots \times \overline{E}_n) \tag{2.6}$$

กำหนดให้ Q_p คือ ความน่าจะเป็นของระบบที่ต่ออุปกรณ์แบบขนานจะเกิดล้มเหลว

$P(\overline{E}_1 \times \overline{E}_2 \times \overline{E}_3 \times \ldots \times \overline{E}_n)$ คือ ความน่าจะเป็นที่จะเกิดความล้มเหลวของเหตุการณ์ $\overline{E}_1, \overline{E}_2, \overline{E}_3, \ldots, \overline{E}_n$

ค่าความน่าเชื่อถือของระบบที่มีระบบพื้นฐานจำนวน n ระบบต่อกันแบบขนานและระบบพื้นฐานทุกหน่วยเป็นอิสระต่อกันอาจเขียนสมการที่ (2.6) ใหม่ได้ว่า

$$Q_p = P(\overline{E}_1) \times P(\overline{E}_2) \times P(\overline{E}_3) \times \ldots \times P(\overline{E}_n) \tag{2.7}$$

กำหนดให้ $P(\overline{E}_1 \times \overline{E}_2 \times \overline{E}_3 \times \ldots \times \overline{E}_n)$ คือ ความน่าจะเป็นที่จะเกิดความล้มเหลวของเหตุการณ์ ตั้งแต่ถึง $\overline{E}_1$ ถึง $\overline{E}_n$

Q_p คือ ความน่าจะเป็นที่ระบบที่ต่อแบบขนานจะล้มเหลว

หากให้ $Q_i = P(\overline{E}_i)$ โดยที่ i = 1, 2, 3,..., n สามารถเขียนสมการที่ (2.7) ได้ใหม่ว่า

$$Q_p = Q_1 \times Q_2 \times Q_3 \times \ldots \times Q_n$$

$$= \prod_{i=1}^{n} Q_i \tag{2.8}$$

ค่าความน่าเชื่อถือของระบบแบบขนาน สามารถหาได้จากสมการ

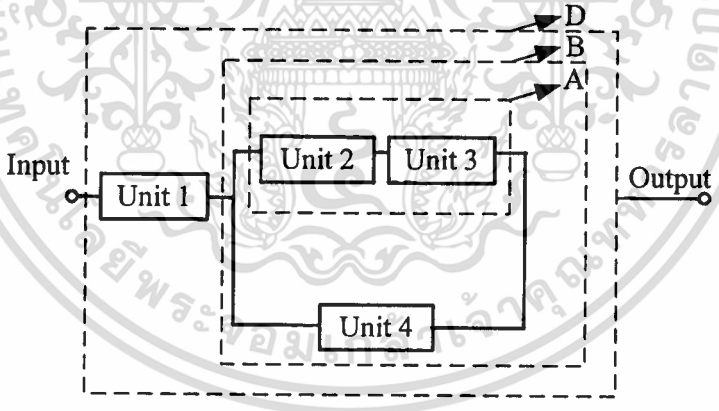
$$R_p = 1 - Q_p = 1 - \prod_{i=1}^{n} Q_i \tag{2.9}$$

กำหนดให้ R_p คือ ความน่าเชื่อถือของระบบที่ต่อกันแบบขนาน

จากสมการที่ (2.8) หากกำหนดให้ $R = e^{-\lambda t}$ โดยกำหนดให้ λ_i อัตราส่วนความล้มเหลวคงที่ และ t_i คือ ช่วงเวลาที่พิจารณาซึ่งเป็นช่วงเวลาที่ต้องการให้อุปกรณ์หรือระบบสามารถทำงานไม่เกิดความล้มเหลว เมื่อแทนเลขชี้กำลังด้วย F_i จะทำให้ได้สมการคือ $R_i = e^{-Fi}$ จากสมการที่ (2.8) หากแทนค่า $Q_1 = 1 - e^{-F1}, Q_2 = 1 - e^{-F2}, \ldots, Q_n = 1 - e^{-Fn}$ ลงในสมการที่ (2.8) สามารถเขียนใหม่ได้ว่า $Q_p = Q_1 \times Q_2 = \left(1 - e^{-F_1}\right)\left(1 - e^{-F_2}\right)$

2.2.3 ระบบหรืออุปกรณ์ที่เชื่อมต่อกันแบบผสมระหว่างอนุกรมและขนาน

ในระบบจริงนั้น อุปกรณ์ที่เชื่อมต่อกันเป็นระบบอาจจะประกอบไปด้วย อุปกรณ์ที่เชื่อมต่อกันแบบขนานและต่อกันแบบอนุกรมมารวมกันเป็นระบบ ซึ่งมีความซับซ้อนมากกว่าระบบที่มีการต่อเชื่อมของอุปกรณ์ที่มีเฉพาะแบบอนุกรมและขนาน โดยสามารถหาความน่าเชื่อของระบบได้ด้วยหลายวิธีการ เช่น การลดรูปของของระบบจนกระทั่งเหลือรูปวงจรแบบเดียว



รูปที่ 2.4 ระบบที่ต่อแบบอนุกรม-ขนาน และการลดรูปวงจร

จากรูปที่ 2.4 ให้ R_i คือ ความน่าจะเป็นที่อุปกรณ์หน่วยที่ i จะทำงานได้สำเร็จ

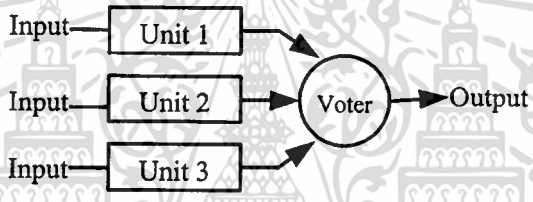
Q_i คือ ความน่าจะเป็นที่อุปกรณ์หน่วยที่ i จะทำงานล้มเหลว

การลดรูปวงจรทำโดยเริ่มลดรูปวงจรในส่วน A ซึ่งมีอุปกรณ์หน่วยที่ 2 และ 3 ต่ออนุกรมกันอยู่สามารถหาความน่าเชื่อถือส่วน A ได้จากสมการ $R_A = R_2 R_3$ จากนั้นจะพบว่าวงจรส่วน A ต่อเชื่อมขนานกันกับอุปกรณ์หน่วยที่ 4 กำหนดให้ส่วนนี้เป็นส่วน B ค่าความน่าเชื่อถือได้ส่วนนี้คือ $R_B = 1 - Q_4(1 - R_2 R_3)$ เมื่อลดรูปแล้วจะพบว่าเหลือส่วน C ซึ่งประกอบด้วยอุปกรณ์หน่วยที่ 1 และส่วน B สามารถหาความน่าเชื่อถือได้จากสมการ $R_T = R_D = R_1[1 - Q_4(1 - R_2 R_3)]$

ซึ่งเมื่อแทน $Q_4 = 1 - R_4$ จะทำให้ได้สมการ $R_T = R_D = R_1(R_2R_3 + R_4 - R_2R_3R_4)$ และสามารถความไม่น่าเชื่อถือรวมของระบบได้จากสมการ $Q_T = 1 - R_T - R_D$

2.2.4 ระบบหรืออุปกรณ์ที่เชื่อมต่อกันแบบโมดูลซ้ำสำรองสามโมดูล

ระบบหรืออุปกรณ์ที่มีการเชื่อมต่อแบบโมดูลซ้ำสำรองสามโมดูล (Triple Modular Redundancy, TMR) เป็นระบบที่นิยมใช้ในการออกแบบระบบคอมพิวเตอร์ โดย Von Neumann [8] เป็นผู้เสนอเป็นครั้งแรกในปี 1956 พื้นฐานระบบนี้จะประกอบไปด้วยโมดูลที่เป็นอิสระต่อกันจำนวนสามโมดูลผ่านไปยังโมดูลสำหรับโหวตคะแนน (Voter) เพื่อเลือกโมดูลเพียงโมดูลเดียวที่จะนำออกเอาท์พุตข้อดีในการใช้ระบบที่ต่อแบบซ้ำซ้อนสำรองสามโมดูล คือ หากกรณีที่มีเอาท์พุตของหนึ่งในสามโมดูลเกิดความผิดพลาด แต่เอาท์พุตจะยังคงทำงานถูกต้องอยู่เนื่องจากจะมีการทำงานโดยตัวโหวตคะแนน สามารถแสดงการต่อเชื่อมแบบโมดูลซ้ำสำรองสามโมดูลดังรูปที่ 2.5 และแสดงการทำงานโดยตัวโหวตคะแนนดังตารางที่ 2.1



รูปที่ 2.5 การเชื่อมต่อระบบหรืออุปกรณ์แบบโมดูลซ้ำสำรองสามส่วน

ตารางที่ 2.1 การโหวตคะแนนของตัวโหวตคะแนนเอาท์พุต

สถานะโมดูล			เอาท์พุตของตัวโหวต
โมดูลที่ 1	โมดูลที่ 2	โมดูลที่ 3	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

ในตารางที่ 2.1 เป็นเงื่อนไขที่ตัวโหวตคะแนนใช้ลงคะแนนเพื่อเลือกเอาท์พุตของโมดูลซ้ำสำรองเพียงหนึ่งโมดูลเป็นเอาท์พุตรวมของระบบเพื่อควบคุมกระบวนการ กำหนดให้ 1 คือ

สถานะที่โมดูลนั้นสามารถทำงานได้สำเร็จ และ 0 คือสถานะที่โมดูลนั้นไม่สามารถทำงานได้โดยกำหนดเงื่อนไขว่าจะต้องมีอย่างน้อย 2 ใน 3 โมดูลที่สามารถทำงานได้สำเร็จ ระบบรวมจึงจะสามารถทำงานได้สำเร็จ ซึ่งตรรกในการโหวตเขียนเป็นสมการพีชคณิตได้ คือ (A+B) C =AC+BC โดยกำหนดให้ A, B และ C คือสถานะการทำงานของโมดูลที่ 1, 2 และ 3 ตามลำดับ

จากรูปที่ 2.5 สามารถหาความน่าเชื่อรวมของระบบแบบโมดูลซ้ำสำรอง 3 โมดูล โดยใช้เงื่อนไขของตัวโหวตในตารางที่ 2.1 หาได้จากสมการ

$$R_{TMRV} = (3R_m^2 - 2R_m^3)R_v \tag{2.10}$$

กำหนดให้

R_{TMRV} คือ ความน่าเชื่อถือของระบบโดยรวมโมดูลของการโหวต

R_v คือ ความน่าเชื่อถือของโมดูลการโหวต

R_m คือ ความน่าเชื่อถือของโมดูลที่ m

หากกำหนดให้เป็นการโหวตที่สมบูรณ์ จะทำให้ $R_v = 1$ ดังนั้น

$$R_{TMR} = (3R_m^2 - 2R_m^3) \tag{2.11}$$

กำหนดให้

R_{TMR} คือ ความน่าเชื่อถือของระบบที่ต่อแบบโมดูลซ้ำสำรอง 3 โมดูล ที่มีการโหวตที่สมบูรณ์ (Perfect voter)

หากพิจารณาสมการที่ (2.11) จะพบว่าจำนวนองค์ประกอบที่มีในระบบ และเงื่อนไขของการโหวตมีผลกับสมการการหาค่าความน่าเชื่อถือของระบบ เลข 3 ที่พจน์แรกแสดงถึงจำนวนองค์ประกอบที่มีในระบบ เลข 2 คือ เงื่อนไขที่กำหนดให้มีองค์ประกอบอย่างน้อย 2 ตัวในระบบจะต้องสามารถทำงานได้ระบบรวมจึงจะทำงานได้สำเร็จ สามาถอธิบายที่มาด้วยทฤษฎีการจัดหมู่ (Combination Theorem) และทฤษฎีบททวินาม (Binomial Theorem) ดังนี้

เมื่อ $(a+b)^n = a^n + na^{n-1}b + \frac{n(n-1)a^{n-2}b^2}{2!} + \frac{n(n-1)(n-2)a^{n-3}b^3}{3!} + \ldots + b^n$

หากกำหนดให้ $a = p$ คือ ความน่าจะเป็นที่ระบบจะสามารถทำงานได้สำเร็จ $b = q$ คือ ความน่าจะเป็นที่ระบบจะไม่สามารถทำงานได้สำเร็จ ดังนั้น $(p+q) = (a+b) = 1$ จะทำให้ได้สมการที่ (2.12)

$$(p+q)^n = p^n + np^{n-1}q + \frac{n(n-1)p^{n-2}q^2}{2!} + \frac{n(n-1)(n-2)p^{n-3}q^3}{3!} + \ldots + q^n = 1 \tag{2.12}$$

สมการที่ (2.12) สามารถเขียนให้อยู่ในรูปเทอมของ rth คือ $rth = \frac{n!}{r!(n-r)!} p^{n-r} q^r$ โดยกำหนดให้ n คือ จำนวนเหตุการณ์ทั้งหมดที่เกิดขึ้น r คือจำนวนของเหตุการณ์ที่เกิดความล้มเหลวขึ้น

สมมุติให้ระบบมีองค์ประกอบทั้งหมดจำนวน n โมดูลและกำหนดให้ p คือ ความน่าจะเป็นที่อุปกรณ์จะสามารถทำงานได้สำเร็จ q คือ ความน่าจะเป็นที่อุปกรณ์นั้นจะไม่สามารถทำงานได้สำเร็จ ซึ่งหาได้จาก $q = 1 - p$ และกำหนดเหตุการณ์ที่จะเกิดขึ้นเพื่อพิจารณาโดยแทนค่าต่างๆ ลงในสมการในเทอมของ rth ดังนี้

กรณีที่สมมุติให้ระบบมีองค์ประกอบโมดูลเดียว นั่นคือ $n = 1$ ความน่าจะเป็นที่โมดูลจะสามารถทำงานได้สำเร็จคือ p

กรณีที่สมมุติให้ระบบมีองค์ประกอบจำนวน 2 โมดูล นั่นคือ $n = 2$ ความน่าจะเป็นที่โมดูลจะสามารถทำงานสำเร็จทั้งคู่คือ $p \times p = p^2$

กรณีที่สมมุติให้ระบบมีองค์ประกอบจำนวน 3 โมดูล นั่นคือ $n = 3$ ความน่าจะเป็นที่โมดูลจะสามารถทำงานได้สำเร็จทั้ง 3 โมดูลคือ $p \times p \times p = p^3$

ผลจากการแทนค่าจำนวนองค์ประกอบในระบบลงในสมการที่ 2.12 และเปรียบเทียบกับการกระจายตามทฤษฎีบททวินามพบว่า เมื่อระบบมีองค์ประกอบจำนวน n โมดูล ความน่าจะเป็นที่ทุกองค์ประกอบจะสามารถทำงานได้สำเร็จคือ p^n นั้นคือพจน์แรกที่ได้จากการกระจายตามทฤษฎีบททวินาม

กรณีที่สมมุติให้ระบบมีองค์ประกอบอยู่ 2 โมดูล นั่นคือ $n = 2$ โดยกำหนดให้โมดูลแรกคือ A และโมดูลที่ 2 คือ B และกำหนดเงื่อนไขของเหตุการณ์ที่สามารถเกิดขึ้นกับองค์ประกอบของระบบ 2 โมดูลคือ หากโมดูลหนึ่งทำงานได้สำเร็จแล้วอีกโมดูลหนึ่งจะต้องไม่ทำงานไม่สำเร็จ เช่น โมดูล A สามารถทำงานได้สำเร็จแต่โมดูล B ทำงานไม่สำเร็จ สามารถหาค่าความน่าเชื่อถือได้คือ $(p \times q)$ และเมื่อโมดูล A ทำงานไม่สำเร็จแล้วโมดูล B ทำงานได้สำเร็จ สามารถหาค่าความน่าเชื่อถือได้คือ $(q \times p)$ นั่นคือ $(p \times q) = (q \times p)$ จะเห็นว่าหากเกิดเหตุการณ์ตามเงื่อนไขดังกล่าวแล้ว จะเกิดเหตุการณ์ไม่เกิดร่วม (Mutually Exclusive) และเมื่อพิจารณาต่อไปอีกว่าระบบนี้มีองค์ประกอบอยู่ 2 โมดูลเมื่อเกิดเหตุการณ์ไม่เกิดร่วมแล้ว สามารถหาความจะน่าเป็นที่โมดูลหนึ่งจะทำงานสำเร็จ และอีกโมดูลหนึ่งจะทำงานไม่สำเร็จได้จาก $2pq$ เมื่อพิจารณาเปรียบเทียบกับการกระจายตามทฤษฎีบททวินามก็จะพบว่า ตรงกับพจน์กลางของการกระจาย $(p+q)^2 = p^2 + 2pq + q^2$

กรณีที่สมมุติให้ระบบมีองค์ประกอบอยู่ 3 โมดูล นั่นคือ $n = 3$ โดยกำหนดให้โมดูลแรกคือ A, B และ C ตามลำดับ และกำหนดเงื่อนไขของเหตุการณ์ที่สามารถเกิดขึ้นกับองค์ประกอบของระบบ 3 โมดูลคือ ต้องมีโมดูลอย่างน้อย 2 ใน 3 โมดูล ระบบรวมจึงจะสามารถทำงานได้สำเร็จ เช่น กรณีที่โมดูล A และ B สามารถทำงานได้สำเร็จและโมดูล C ไม่สามารถทำงานได้

สำเร็จ กรณีที่โมดูล A และ C สามารถทำงานได้สำเร็จและโมดูล B ไม่สามารถทำงานได้สำเร็จ และกรณีที่โมดูล B และ C ทำงานได้สำเร็จและโมดูล A ไม่สามารถทำงานได้สำเร็จ จะเห็นได้ว่าเงื่อนไขเช่นนี้ก็จะเกิดเหตุการณ์ไม่เกิดร่วมเช่นเดียวกัน โดยสามารถหาความน่าจะเป็นที่มีอย่างน้อย 2 ใน 3 โมดูลจะทำงานสำเร็จได้จาก $3p^2q$ เมื่อพิจารณาเปรียบเทียบกับการกระจายตามทฤษฎีบททวินามของ $(p+q)^3 = p^3 + 3p^2q + 3pq^2 + q^3$ ก็จะพบว่าตรงกับพจน์ที่สองของการกระจาย การความน่าเชื่อถือของระบบที่ประกอบด้วยองค์ประกอบ 3 โมดูลโดยมีเงื่อนไขว่า 2 ใน 3 โมดูลจะต้องสามารถทำงานได้สำเร็จจึงสามารถหาได้จากสมการ $p^3 + 3p^2q$ โดยที่ p^3 หมายถึงกรณีที่เกิดเหตุการณ์โมดูลทั้ง 3 หากแทนค่า $q = 1 - p$ ก็จะทำให้ได้สมการที่ (2.11)

2.3 ระบบที่มีโมดูลซ้ำสำรอง

ระบบที่มีโมดูลซ้ำสำรอง (Redundant Systems) ถูกนำมาใช้เพื่อเพิ่มความน่าเชื่อถือให้กับระบบ โดยในระบบจะประกอบด้วยโมดูลที่ซ้ำกันเพื่อให้สามารถทำงานทดแทนกัน เมื่อเกิดกรณีโมดูลที่กำลังทำงานอยู่ไม่สามารถทำงานได้ โมดูลซ้ำที่สำรองจะต้องสามารถทำงานทดแทนได้ทันที การเชื่อมต่อกันของอุปกรณ์ในระบบที่มีโมดูลซ้ำสำรองนั้น ตามหลักของวิศวกรรมความน่าเชื่อถือมีหลายรูปแบบด้วยกัน ซึ่งการเชื่อมต่อแต่ละแบบก็จะเหมาะสมกับการประยุกต์ใช้ในงานวิศวกรรมที่ต่างกันออกไป หากแบ่งลักษณะการทำงานของโมดูลซ้ำสำรอง อาจแบ่งได้โมดูลที่เชื่อมกันกันแบบขนานได้ 2 ลักษณะคือ แบบทำงานพร้อมกัน (Active Parallel) และแบบสำรองรอทำงาน (Stand-by) ดังที่แสดงในรูปที่ 2.6

(ก) Active Redundancy (ข) Stand-by Redundancy

รูปที่ 2.6 ระบบที่มีโมดูลซ้ำสำรองแบ่งตามลักษณะการทำงาน

2.3.1 การทำงานของโมดูลซ้ำสำรองแบบทำงานพร้อมกัน (Active Parallel)

โมดูลซ้ำสำรองที่ทำงานแบบทำงานพร้อมกัน บางครั้งเรียกว่า Hot Stand-by พบมาในการใช้ควบคุมกระบวนการทางอุตสาหกรรม มีลักษณะคล้ายกันกับโมดูลซ้ำสำรองที่ทำงานแบบสำรองรอทำงาน โดยแต่โมดูลที่ซ้ำสำรองแบบทำงานพร้อมกันนี้ จะทำงานควบคู่ไปกับโมดูลหลัก และคอยตรวจจับการหยุดการทำงานของโมดูลหลัก ถ้าหากโมดูลหลักหยุดทำงานลง โมดูลซ้ำสำรองแบบทำงานพร้อมกันจะเข้าไปทำงานแทนที่โมดูลหลักเองอย่างอัตโนมัติ ด้วยความรวดเร็ว เพราะโมดูลสำรองแบบทำงานพร้อมกันทำงานควบคู่ไปกับโมดูลหลักอยู่แล้ว ดังนั้นสภาวะการทำงานของระบบจะทำงานต่อเนื่องกันได้ทันที แต่เอาต์พุตของตัวควบคุมก็ยังต้องการช่วงเวลาหนึ่งในการสร้างสภาวะของเอาท์พุต ถ้าหากค่าเวลาก่อนที่กระบวนการจะหยุดทำงานมีค่าน้อยมากๆ น้อย

เกินกว่าช่วงเวลาที่ต้องการสร้างสภาวะของเอาท์พุตแล้ว กระบวนการก็จะหยุดทำงานได้

2.3.2 การทำงานของโมดูลที่ซ้ำสำรองแบบสำรองรอทำงาน (Stand-by Redundancy)

การทำงานของโมดูลที่ซ้ำสำรองแบบสำรองรอทำงานหรือบางครั้งเรียกว่า Cold Stand-by Redundancy ซึ่งเป็นระบบที่ง่ายที่สุด โดยจะมีโมดูลซ้ำสำรองที่สามารถต่อเข้าทำงานทดแทนโมดูลหลักได้อย่างรวมเร็ว เมื่อโมดูลหลักเกิดความล้มเหลวเนื่องจากเกิดความผิดพลาดบางอย่างขึ้น วิธีการนี้จะช่วยลดเวลาในการเปลี่ยน หรือซ่อมแซมโมดูลที่เกิดความล้มเหลวขึ้น เนื่องจากโมดูลสำรองสามารถทำงานทดแทนโมดูลหลักได้ทันที ไม่เสียเวลาในการตรวจวิเคราะห์หาสาเหตุการหยุดการทำงานของระบบ ซึ่งสามารถหาได้ในภายหลัง อีกทั้งไม่เสียเวลาในการติดตั้งชิ้นส่วนสำรองให้ระบบ แต่วิธีนี้จะเสียค่าใช้จ่ายมาก เพราะว่าต้องสร้างระบบสำรองขึ้นมาอีกระบบหนึ่งรอไว้ รวมด้วยค่าใช้จ่ายในการสร้างอุปกรณ์ที่ใช้ตัดต่อให้ระบบสำรองเข้ามาทำงานแทนระบบหลักด้วย ระบบนี้เหมาะกับกระบวนการที่มีค่าเวลาก่อนที่กระบวนการจะหยุดทำงานมากๆ

2.4 การสวิตซ์ของระบบแบบโมดูลซ้ำสำรอง

ในกรณีที่โมดูลซ้ำสำรองทำงานแบบพร้อมกัน ทุกโมดูลในระบบจะทำงานพร้อมๆ กัน ซึ่งต่างกับโมดูลซ้ำสำรองที่ทำงานแบบสำรองรอทำงาน โมดูลซ้ำสำรองจะทำงานก็ต่อเมื่อโมดูลหลักที่ทำงานอยู่เกิดความผิดพลาดบางอย่างจนไม่สามารถทำงานได้ จะเห็นการสับเปลี่ยนจากโมดูลหลักไปสู่โหมดการทำงานของโมดูลสำรอง จำเป็นต้องพิจารณาในส่วนของรูปแบบการสับเปลี่ยนโมดูล สามารถแบบแบ่งรูปแบบการสับเปลี่ยนได้ 2 ชนิด คือ การสับเปลี่ยนที่สมบูรณ์ (Perfect Switching) และการสับเปลี่ยนที่ไม่สมบูรณ์ (Imperfect Switching)

2.4.1 การสับเปลี่ยนของโมดูลที่สมบูรณ์แบบ

การสับเปลี่ยนของโมดูลที่สมบูรณ์แบบคือ เมื่อโมดูลหลักเกิดความล้มเหลงในการทำงานแล้ว การสับเปลี่ยนไปใช้โมดูลสำรองจะต้องไม่ติดขัด จากรูปที่ 2.6(ข) หากสมมุติมีการสับเปลี่ยนเป็นแบบสมบูรณ์แบบ และอัตราการล้มเหลวของการสับเปลี่ยนเป็นศูนย์ จะพบว่าระบบจะเกิดความล้มเหลวได้ก็ต่อเมื่อ โมดูล B กำลังอยู่ในโหมดของการทำงานแล้วเกิดความล้มเหลวขึ้น นั่นคือ เกิดเหตุการณ์ที่โมดูล A ซึ่งเป็นโมดูลหลักเกิดความล้มเหลวในการทำงาน แล้วมีการสับเปลี่ยนให้โมดูล B ทำงานทดแทน จากนั้นระบบจะล้มเหลวทันทีเมื่อโมดูล B สามารถหาความน่าจะเป็นที่จะเกิดความล้มเหลวขึ้นในระบบจากสมการที่ (2.13)

$$Q = Q(A)Q\left(B \mid \overline{A}\right) \tag{2.13}$$

หากโมดูล A และ B เป็นอิสระต่อกันสามารถเขียนสมการที่ (2.3) ใหม่ได้ว่า

$$Q = Q(A)Q(B) = Q_A Q_B \tag{2.14}$$

โดยที่ Q คือ ความน่าจะเป็นที่จะเกิดความล้มเหลวกับระบบ

Q_A คือ ความน่าจะเป็นที่จะเกิดความล้มเหลวในโมดูล A

Q_B คือ ความน่าจะเป็นที่จะเกิดความล้มเหลวในโมดูล B

2.4.2 การสับเปลี่ยนของโมดูลแบบไม่สมบูรณ์แบบ

ในกรณีของการสับเปลี่ยนของโมดูลแบบไม่สมบูรณ์แบบนั้น จะนำความน่าจะเป็นที่จะสับเปลี่ยนจากโมดูล A ไปสู่โมดูล B แล้วสามารถเกิดความล้มเหลวขึ้นมาพิจารณาด้วย จากรูปที่ 2.6(ข) หากกำหนดให้ P_s คือ ความน่าจะเป็นที่มีการสับเปลี่ยนจากโมดูล A ไปสู่โมดูล B ได้สำเร็จ สามารถหาความน่าจะเป็นที่ระบบจะเกิดความล้มเหลวได้คือ

$$\begin{aligned} Q &= Q_A Q_B P_s + Q_A(1 - P_s) \\ &= Q_A - Q_A P_s(1 - Q_B) \end{aligned} \tag{2.15}$$

2.5 ระบบโหวตคะแนน

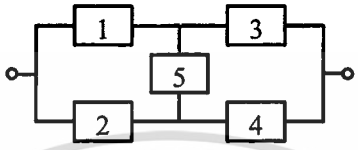
ในระบบควบคุมแบบคอมพิวเตอร์ (Computer Control System) ถูกนำมาใช้สำหรับการควบคุมกระบวนการให้มีความต่อเนื่องและปลอดภัยในการปฏิบัติงาน โดยในระบบที่ซ้ำสำรองจะมีโมดูลของการโหวตคะแนน (Voting System) เพื่อเลือกเอาท์พุตเพียงหนึ่งเอาท์พุตใช้ควบคุมกระบวนการ เงื่อนไขการโหวตที่เหมาะสมและนิยมใช้ในระบบคอมพิวเตอร์ควบคุมแบบโมดูลซ้ำสำรอง [16] คือ ระบบโหวต 2 ใน 3 โมดูล หมายถึง มีคอมพิวเตอร์ควบคุมกระบวนการจำนวน 3 โมดูลหรือเครื่องมือวัดคุมซึ่งต่อทำงานแบบขนานกัน จากนั้นจะมีการเปรียบเทียบเอาท์พุตของทั้ง 3 โมดูล หากมีเพียงหนึ่งเอาท์พุตที่ต่างไปจากเอาท์พุตอีกสองโมดูล โมดูลนั้นจะถูกเพิกเฉยไปจากระบบการโหวตคะแนน การหาความน่าเชื่อถือของระบบที่ใช้เงื่อนไข 2 ใน 3 ถูกแสดงไว้ในสมการที่ (2.10) และ (2.11) ระบบควบคุมที่ดีควรมีโมดูลการโหวตคะแนนที่มีความน่าจะเป็นที่จะเกิดความล้มเหลวน้อยมากๆ

2.6 การหาความน่าเชื่อถือระบบด้วยวิธีมินิมัลคัทเซ็ท

การเชื่อมต่อกันของอุปกรณ์ในระบบที่มีความซับซ้อน (Complex Systems) ซึ่งในระบบจะประกอบด้วยอุปกรณ์ที่เชื่อมต่อกันมากมายหลายโมดูล อาจทำให้ยากต่อการลดรูปการเชื่อมต่อของอุปกรณ์ต่างๆ ให้อยู่ในรูปพื้นฐานคือ การเชื่อมกันของอุปกรณ์แบบขนาน และการเชื่อมต่อแบบอนุกรม วิธีการมินิมัลคัทเซ็ท (Minimal Cut Set Methods) ได้ถูกนำมาใช้ในการแก้ปัญหานี้โดยใช้หลักการพิจารณา กลุ่มของโมดูลองค์ประกอบที่มีความสำคัญกับระบบซึ่งแต่ละโมดูลมี

ความสัมพันธ์กัน และมีผลต่อการทำงานของระบบว่าจะเกิดความล้มเหลวหรือไม่ หากเกิดโมดูลใดโมดูลหนึ่งในกลุ่มเกิดความล้มเหลวในการทำงาน

คัทเซ็ท เป็นเซ็ทของโมดูลองค์ประกอบในระบบที่มีเส้นทางเดินตั้งแต่ด้านอินพุทต์ของระบบไปสู่ด้านเอาท์พุทต์ของระบบ หากขาดโมดูลใดโมดูลหนึ่งในเส้นทางนั้นแล้วจะทำให้ระบบเกิดความล้มเหลวขึ้น หากพิจารณาในรูปที่ 2.7 สามารถเขียนเป็นเซ็ทของมินิมัลคัทเซ็ทได้ดังตารางที่ 2.2



รูปที่ 2.7 การเชื่อมต่อกันขององอุปกรณ์สำหรับการหาความน่าเชื่อถือแบบมินิมัลคัทเซ็ต

ตารางที่ 2.2 เซ็ตของโมดูลองค์ประกอบที่เป็นมินิมัลคัทเซ็ท

Minimal Cut Set	Component in the set
C_1	1,2
C_2	3,4
C_3	1,4,5
C_4	2,3,5

หากกำหนดให้ $\overline{C_i}$ คือความน่าจะเป็นที่จะทำให้ระบบเกิดความล้มเหลวในการทำงานโดยคิดด้วยวิธีคัทเซ็ทที่เส้นทางเดิน i สามารถเขียนเป็นสมการเพื่ออธิบายได้ดังนี้

$$P_f = P\left(\overline{C_1} \cup \overline{C_2} \cup \overline{C_3} \cup \ldots \cup \overline{C_m}\right) \tag{2.16}$$

โดยกำหนดให้ P_f คือ ความน่าจะเป็นที่ระบบจะเกิดความล้มเหลว

จากรูปที่ 2.7 สามารถเขียนเป็นสมการได้คือ

$$\begin{aligned} P_f &= P\left(\overline{C_1} \cup \overline{C_2} \cup \overline{C_3} \cup \overline{C_4}\right) \\ &= [P(\overline{C_1}) + P(\overline{C_2}) + P(\overline{C_3}) + P(\overline{C_4})] - [P(\overline{C_1} \cap \overline{C_2}) \\ &\quad + P(\overline{C_1} \cap \overline{C_3}) + P(\overline{C_1} \cap \overline{C_4}) + P(\overline{C_2} \cap \overline{C_3}) + P(\overline{C_3} \cap \overline{C_4}) \\ &\quad + P(\overline{C_2} \cap \overline{C_4})] + [P(\overline{C_1} \cap \overline{C_2} \cap \overline{C_3}) + P(\overline{C_1} \cap \overline{C_2} \cap \overline{C_4}) \\ &\quad + P(\overline{C_2} \cap \overline{C_3} \cap \overline{C_4}) + P(\overline{C_1} \cap \overline{C_3} \cap \overline{C_4})] \\ &\quad - [P(\overline{C_1} \cap \overline{C_2} \cap \overline{C_3} \cap \overline{C_4})] \end{aligned}$$

2.7 ความน่าเชื่อถือระบบซ้ำสำรองที่ทำงานแบบมีเงื่อนไข

อัตราการล้มเหลว (Failure Rate: λ) และค่าเวลาเฉลี่ยที่จะล้มเหลว (Mean Time to Failures: MTTF) เป็นอีกตัวแปรหนึ่งที่ใช้ในการพิจารณาหาความน่าเชื่อถือของระบบ โดยทั้งสองตัวแปรนั้นมีความสัมพันธ์กัน ในหัวข้อนี้เป็นการหาความน่าเชื่อถือระบบที่มีการทำงานแบบมีเงื่อนไขในเทอมของอัตราการล้มเหลว และค่าเฉลี่ยเวลาที่จะล้มเหลว

ความน่าเชื่อถือหรือความน่าจะเป็นที่จะไม่เกิดความล้มเหลวขึ้นในระบบ สามารถอธิบายคุณลักษณะของอุปกรณ์นั้นได้จากสมการ

$$R = e^{-\lambda t} = e^{-t/m} \tag{2.17}$$

$$m = MTTF = \frac{1}{\lambda} \tag{2.18}$$

โดยที่ R คือ ค่าความน่าเชื่อถือที่ระบบจะไม่เกิดความล้มเหลวในเวลา t

λ คือ ค่าอัตราการล้มเหลว

m คือ ค่าเวลาเฉลี่ยที่จะล้มเหลว

2.7.1 ความน่าเชื่อถือระบบที่มีโมดูลซ้ำสำรองเชื่อมต่อกันแบบขนาน 2 โมดูล

จากสมการที่ (2.8) ซึ่งเป็นสมการที่มีโมดูลซ้ำสำรองเชื่อมต่อกันแบบขนาน สมมุติให้ระบบที่กำลังพิจารณาประกอบด้วยองค์ประกอบ 2 โมดูลสามารถหาค่าความไม่น่าเชื่อถือของระบบได้จากสมการ $Q_1 \times Q_2$ หากแทนค่า $Q_1 = 1 - R_1 = 1 - e^{-F_1}$ และ $Q_2 = 1 - R_2 = 1 - e^{-F_2}$ จะทำให้ได้สมการ $Q_1 \times Q_2 = \left(1 - e^{-F_1}\right) \times \left(1 - e^{-F_2}\right) = 1 - e^{-F_1} - e^{-F_2} + \left(e^{-F_1} \times e^{-F_2}\right)$ นั้นคือสามารถหาความน่าเชื่อถือของระบบได้จากสมการดังนี้

$$R_T = e^{-F_1} - e^{-F_2} - e^{-(F_1 + F_2)} \tag{2.19}$$

สามารถเขียนสมการให้อยู่ในรูปความน่าเชื่อถือแต่ละโมดูลจะได้สมการ

$$R_T = R_1 + R_2 \times R_1 R_2 \tag{2.20}$$

หรือสามารถเขียนสมการให้อยู่ในรูปความไม่น่าเชื่อถือได้คือ

$$R_T = 1 - \left(Q_1 \times Q_2\right) \tag{2.21}$$

หากทั้ง 2 โมดูลมีอัตราการล้มเหลวเท่ากัน สามารเขียนสมการที่ (2.19) ถึง (2.21) ได้ใหม่ว่า $R_T = 2e^{-F} - e^{-2F}$, $R_T = 2R - R^2$, $R_T = 1 - Q^2$ ตามลำดับ ค่าเวลาเฉลี่ยที่จะล้มเหลว (Mean time to failure (MTTF) สามารถหาได้จากการอินทิเกรตฟังก์ชันความน่าจะเป็นในสมการที่ (2.19)

$$m = \int_0^\infty R_T dt = \int_0^\infty (e^{-\lambda_1 t} + e^{-\lambda_2 t} - e^{(-\lambda_1+\lambda_2)t}) dt$$

$$= \int_0^\infty e^{-\lambda_1 t} dt + \int_0^\infty e^{-\lambda_2 t} dt - \int_0^\infty e^{-(\lambda_1+\lambda_2)t} dt$$

$$= \int_0^\infty e^{-\lambda_1 t} \frac{-\lambda_1}{-\lambda_1} dt + \int_0^\infty e^{-\lambda_2 t} \frac{-\lambda_2}{-\lambda_2} dt - \int_0^\infty e^{-(\lambda_1+\lambda_2)t} \frac{-(\lambda_1+\lambda_2)}{-(\lambda_1+\lambda_2)} dt$$

$$= -\frac{1}{\lambda_1} \int_0^\infty e^{-\lambda_1 t} d(-\lambda_1 t) - \frac{1}{\lambda_2} \int_0^\infty e^{-\lambda_2 t} d(-\lambda_2 t) + \frac{1}{(\lambda_1+\lambda_2)} \int_0^\infty e^{-(\lambda_1+\lambda_2)t} d(\lambda_1+\lambda_2)t$$

$$= -\frac{1}{\lambda_1}[e^{-\lambda_1 t}]_0^\infty - \frac{1}{\lambda_2}[e^{-\lambda_2 t}]_0^\infty + \frac{1}{(\lambda_1+\lambda_2)}[e^{-(\lambda_1+\lambda_2)t}]_0^\infty$$

$$= -\frac{1}{\lambda_1}(0-1) - \frac{1}{\lambda_2}(0-1) + \frac{1}{\lambda_1+\lambda_2}(0-1)$$

$$= \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - \frac{1}{\lambda_1+\lambda_2}$$

หากทั้ง 2 โมดูลมีอัตราการล้มเหลวเท่ากันจะทำให้ได้สมการ

$$m = \frac{1}{\lambda} + \frac{1}{\lambda} - \frac{1}{2\lambda} = \frac{3}{2\lambda} \qquad (2.22)$$

2.7.2 ความน่าเชื่อถือระบบที่มีโมดูลซ้ำสำรองเชื่อมต่อกันแบบขนาน 3 โมดูล

การทำงานแบบมีเงื่อนไขของระบบที่มีโมดูลซ้ำสำรองจำนวน 3 โมดูล อาจแยกการทำงานแบบมีเงื่อนไขได้คือ กรณีที่ต้องมีอย่างน้อย 1 ใน 3 โมดูลทำงานสำเร็จระบบจึงจะไม่ล้มเหลวในการทำงาน และในกรณีที่ต้องมีอย่างน้อย 2 ใน 3 โมดูลทำงานสำเร็จระบบจึงจะไม่เกิดความล้มเหลวในการทำงาน อาจแยกการพิจารณาได้ดังนี้

กรณีที่ต้องมีอย่างน้อย 1 ใน 3 โมดูลต้องทำงานสำเร็จ

เมื่อพิจารณารูปที่ 2.3 และสมมุติให้มีโมดูลแบบซ้ำสำรองในระบบจำนวน 3 โมดูล ดังนั้นสามารถหาความน่าจะเป็นที่ระบบจะเกิดความล้มเหลวคือ $Q_1 \times Q_2 \times Q_3 = (1-e^{-F_1}) \times (1-e^{-F_2}) \times (1-e^{-F_3}) = 1 - e^{-F_1} - e^{-F_2} - e^{-F_3} + (e^{-F_1} \times e^{-F_2}) + (e^{-F_1} \times e^{-F_3}) + (e^{-F_2} \times e^{-F_3}) - (e^{-F_2} \times e^{-F_2} \times e^{-F_3})$ นั่นคือ สามารถหาความน่าเชื่อถือของระบบได้จากสมการ

$$R_T = e^{-F_1} + e^{-F_2} + e^{-F_3} - e^{-(F_1+F_2)} - e^{-(F_1+F_3)} - e^{-(F_2+F_3)} - e^{-(F_1+F_2+F_3)} \qquad (2.23)$$

$$= R_1 + R_2 + R_3 - (R_1 \times R_2) - (R_1 \times R_3) - (R_2 \times R_3) + (R_1 \times R_2 \times R_3) \qquad (2.24)$$

$$= 1 - (Q_1 \times Q_2 \times Q_3) \qquad (2.25)$$

หากโมดูลทั้ง 3 มีค่าอัตราการล้มเหลวเท่ากันสามารถเขียนสมการที่(2.23) ถึง (2.25) ใหม่ได้คือ $3e^{-F}-3e^{-2F}+e^{-3F}$, $3R-3R^2+R^3$ และ $1-Q^3$ ตามลำดับ ค่าเวลาเฉลี่ยที่จะล้มเหลว (Mean time to failure (MTTF) สามารถหาได้จากการอินทิเกรตฟังก์ชันความน่าจะเป็นในสมการที่ (2.23) ซึ่งจะทำให้ได้ค่าเวลาเฉลี่ยที่จะล้มเหลวคือ

$$m=\frac{1}{\lambda_1}+\frac{1}{\lambda_2}+\frac{1}{\lambda_3}-\frac{1}{\lambda_1+\lambda_2}-\frac{1}{\lambda_1+\lambda_3}-\frac{1}{\lambda_2+\lambda_3}+\frac{1}{\lambda_1+\lambda_2+\lambda_3} \tag{2.26}$$

หากทั้ง 3 โมดูลมีอัตราการล้มเหลวเท่ากัน สามารถเขียนสมการที่ (2.26) ใหม่ได้ว่า

$$m=\frac{3}{\lambda}-\frac{3}{2\lambda}+\frac{1}{3\lambda}=\frac{18}{6\lambda}-\frac{9}{6\lambda}+\frac{2}{6\lambda}=\frac{11}{6\lambda} \tag{2.27}$$

กรณีที่ต้องมีอย่างน้อย 2 ใน 3 โมดูลต้องทำงานสำเร็จ

เมื่อพิจารณารูปที่ 2.3 และสมมุติให้มีโมดูลแบบซ้ำสำรองในระบบจำนวน 3 โมดูล หากพิจารณาเงื่อนไขที่ 2 ใน 3 โมดูลจะต้องทำงานสำเร็จจะพบว่าหาก โมดูล A เกิดความล้มเหลวในการทำงานแล้วโมดูล B และ C จะต้องทำงานสำเร็จ หากโมดูล B เกิดความล้มเหลวแล้วโมดูล A และ C จะต้องทำงานสำเร็จ หากโมดูล C เกิดความล้มเหลวในการทำงานแล้ว โมดูล A และ B จะต้องทำงานสำเร็จ และจะพบว่าหากทั้ง 3 โมดูลสามารถทำงานได้สำเร็จระบบรวมก็จะทำงานได้สำเร็จเช่นกัน ความน่าเชื่อถือของระบบสามารถหาได้จากสมการดังนี้

$$\begin{aligned}R_T &= (R_1\times R_2\times R_3)+(Q_1\times R_2\times R_3)+(R_1\times Q_2\times R_3)+(R_1\times R_2\times Q_3)\\ &=(R_1\times R_2\times R_3)+\left(Q_1\times R_2\times R_3\times\frac{R_1}{R_1}\right)+\left(R_1\times Q_2\times R_3\times\frac{R_2}{R_2}\right)+\left(R_1\times R_2\times Q_3\times\frac{R_3}{R_3}\right)\\ &=(R_1\times R_2\times R_3)+\left(R_1\times R_2\times R_3\times\frac{Q_1}{R_1}\right)+\left(R_1\times R_2\times R_3\times\frac{Q_2}{R_2}\right)+\left(R_1\times R_2\times R_3\times\frac{Q_3}{R_3}\right)\\ &=(R_1\times R_2\times R_3)\times\left(1\times\frac{Q_1}{R_1}\times\frac{Q_2}{R_2}\times\frac{Q_3}{R_3}\right)\end{aligned} \tag{2.28}$$

หากแต่ละโมดูลมีค่าอัตราการล้มเหลวเท่ากันจะทำให้ได้สมการ

$$\begin{aligned}R_T &= R^3\left(1+3\frac{Q}{R}\right)=R^3+3R^2Q\\ &=R^3+3R^2(1-R)=R^3+3R^2-3R^3=3R^2-2R^3\end{aligned} \tag{2.29}$$

ค่าเวลาเฉลี่ยที่จะล้มเหลว (Mean time to failure (MTTF) สามารถหาได้จากการอินทิเกรตฟังก์ชันความน่าจะเป็นในสมการที่ (2.28) ซึ่งจะทำให้ได้ค่าเวลาเฉลี่ยที่จะล้มเหลวคือ

$$e^{-\lambda t} + \lambda t e^{-\lambda t} + \frac{(\lambda t)^2 e^{-\lambda t}}{2!} + \ldots = 1$$

$$e^{-\lambda t}\left(1 + \lambda t + \frac{(\lambda t)^2}{2!} + \ldots\right) = 1 \qquad (2.33)$$

กำหนดให้ λ คือ อัตราการล้มเหลว

t คือ ระยะเวลาที่ใช้ในการทดลอง

หากพิจารณาให้แต่ละโมดูลเหมือนกันโดยมีโอกาสที่จะเกิดความล้มเหลวเท่ากัน และตัวโหวตมีสามารถสับเปลี่ยนการทำงานของโมดูลอย่างสมบูรณ์นั่นคือ ค่าอัตราการล้มเหลวมีค่าเป็นศูนย์ สมมุติให้ระบบมีโมดูลจำนวน 2 โมดูลเชื่อมต่อกันแบบขนานโดยมีโมดูลหนึ่งเป็นโมดูลซ้ำสำรองและทำงานโดยลำดับ สามารถหาความน่าเชื่อถือของระบบได้จากสมการ

$$R_T = e^{-\lambda t}(1 + \lambda t) \qquad (2.34)$$

กรณีที่โมดูลในระบบมีจำนวน 3 โมดูล โดยกำหนดให้มี 2 โมดูลเป็นโมดูลซ้ำสำรองแบบทำงานโดยลำดับ สามารถหาความน่าเชื่อถือได้จากสมการ

$$R_T = e^{-\lambda t}\left[1 + \lambda t + \frac{(\lambda t)^2}{2}\right] \qquad (2.35)$$

กรณีที่โมดูลในระบบมีจำนวน n โมดูล โดยกำหนดให้มี $n-1$ โมดูลเป็นโมดูลซ้ำสำรองแบบทำงานโดยลำดับ สามารถหาความน่าเชื่อถือระบบได้จากสมการ

$$R_T = e^{-\lambda t}\left[1 + \lambda t + \frac{(\lambda t)^2}{2} + \frac{(\lambda t)^3}{6} + \ldots + \frac{(\lambda t)^{n-1}}{(n-1)!}\right] \qquad (2.36)$$

หากต้องการหาค่าเวลาเฉลี่ยที่จะล้มเหลว สามาถหาได้ด้วยการอินทิเกรตสมการที่ (2.34) ถึง (2.36) ซึ่งจะทำให้ได้ค่าเวลาเฉลี่ยที่จะล้มเหลวในแต่ละกรณีคือ กรณีที่มีระบบซ้ำสำรอง 2 โมดูล โดยมีโมดุลซ้ำสำรอง 1 โมดูล กรณีที่มีระบบซ้ำสำรอง 3 โมดูลโดยมีโมดูลซ้ำสำรอง 2 โมดูล และในกรณีที่ระบบซ้ำสำรองจำนวน n โมดูลโดยมีโมดูลซ้ำสำรองจำนวน $n-1$ โมดูล โดยหาค่าเวลาเฉลี่ยที่จะล้มเหลวได้ตามลำดับกรณีดังนี้

$$m = \frac{1}{\lambda} + \frac{1}{\lambda} = \frac{2}{\lambda} \qquad (2.37)$$

$$m = \frac{1}{\lambda} + \frac{1}{\lambda} + \frac{1}{\lambda} = \frac{3}{\lambda} \qquad (2.38)$$

$$m = \frac{1}{\lambda} + \frac{1}{\lambda} + .. + \frac{1}{\lambda} = \frac{n}{\lambda} \qquad (2.39)$$

2.9 ความน่าเชื่อถือในระบบคอมพิวเตอร์

ระบบคอมพิวเตอร์ได้ถูกพัฒนาให้มีศักยภาพทางด้านเทคโนโลยีให้สูงขึ้นอย่างรวดเร็วในเวลาอันสั้น ในงานวิจัยนี้ได้นำเอาระบบคอมพิวเตอร์มาใช้ในงานระบบควบคุม เพื่อที่จะให้ได้ระบบควบคุมแบบคอมพิวเตอร์ที่มีความน่าเชื่อถือสูง เหมาะสมกับการประยุกต์ใช้ในการควบคุมกระบวนการที่มีความสำคัญสูงเพื่อให้กระบวนการนั้นมีความต่อเนื่อง ระบบคอมพิวเตอร์ดังกล่าวจะประกอบไปด้วย 2 ส่วนด้วยกันนั้นคือ ฮาร์ดแวร์และซอฟต์แวร์ ฉะนั้นการพิจารณาความน่าเชื่อถือของระบบคอมพิวเตอร์นั้น ก็จำเป็นจะต้องพิจารณาทั้งด้านคือ ฮาร์ดแวร์และซอฟต์แวร์ ความน่าเชื่อถือของฮาร์ดแวร์และซอฟต์แวร์

2.9.1 คำจำกัดความของความน่าเชื่อถือในระบบคอมพิวเตอร์

ความผิดพลาดของชุดคำสั่ง (Software Error) ความผิดพลาดหรือความคลาดเคลื่อนในชุดคำสั่งอาจเกิดจากแนวคิดของผู้ออกแบบซอฟต์แวร์ ผู้ปฏิบัติงาน ที่ไม่ตรงกับจุดมุ่งหมายของการออกแบบหรือใช้งาน เช่น ความผิดพลาดที่เกิดจากความขัดแย้งของวากยสัมพันธ์ นั่นอาจเกิดจากผู้ออกแบบซอฟต์แวร์ไม่เข้าใจในไวยากรณ์ของภาษาคอมพิวเตอร์ที่ใช้ออกแบบ

ความผิดพร่อง (Fault) ความผิดพร่องตามนิยามแล้วกำหนดว่า ทุกอย่างที่ตรงกันข้ามที่ทำให้เกิดความไม่น่าเชื่อถือ ให้ถือว่าเป็นความผิดพร่อง

ความผิดพร่องที่ครอบคลุม (Covered fault) คือ ความผิดพร่องที่เกิดจากสิ่งซึ่งพิจารณาแล้วว่าสามารถกลับคืนสู่สภาวะที่สามารถใช้งานได้โดยอัตโนมัติ

ความผิดพร่องที่ไม่ครอบคลุม (Uncovered fault) คือ ความผิดพร่องที่เกิดจากสิ่งซึ่งพิจารณาแล้วว่าไม่สามารถกลับคืนสู่สภาวะที่สามารถใช้งานได้โดยอัตโนมัติ

แก้จุดบกพร่อง (Debugging) คือ กระบวนการซึ่งต้องการกำจัดความผิดพลาดอันจะเกิดขึ้นกับชุดคำสั่ง

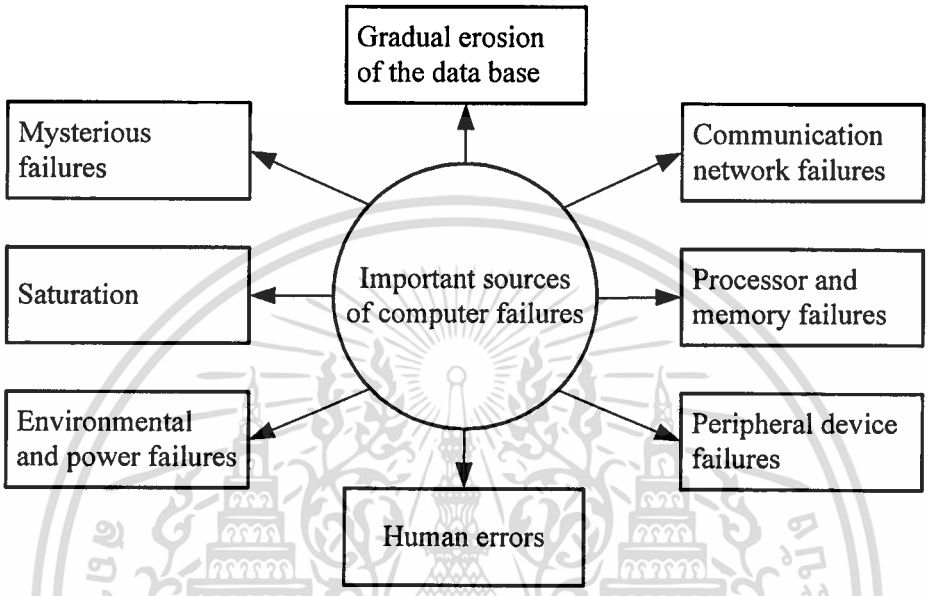
ความทนต่อความผิดพร่อง (Fault tolerant computing) คือ ความสามารถของระบบที่จะทำงานตามเป้าหมาย หรือจุดประสงค์ให้สำเร็จอย่างสมบูรณ์ ปราศจากความผิดพลาดที่เกิดทั้งที่เกิดกับฮาร์ดแวร์และซอฟต์แวร์

ความน่าเชื่อถือของซอฟต์แวร์ (Software reliability) คือ ความน่าจะเป็นที่ซอฟต์แวร์จะทำงานตามจุดประสงค์ภายในช่วงเวลาและเงื่อนไขที่กำหนด โดยปราศจากความความผิดพลาด

การทดสอบซอฟต์แวร์ (Software testing) คือ กระบวนการทดสอบสำหรับการทำงานของซอฟต์แวร์ เพื่อต้องการผลลัพธ์ว่าซอฟต์แวณ์ที่ได้ถูกต้องตามเงื่อนไขที่ต้องการหรือไม่

2.9.2 สาเหตุความล้มเหลวในระบบคอมพิวเตอร์ (Computer Failure Causes)

ระบบมีความไม่น่าเชื่อถือนั้นหรือ ระบบไม่สามารถทำงานได้อย่างต่อเนื่องอันเนื่องมากจากเกิดความล้มเหลว สาเหตุที่ก่อให้เกิดความล้มเหลวในระบบคอมพิวเตอร์ก็มีอยู่หลายสาเหตุไม่เพียงแต่ สาเหตุที่เกิดจากซอฟต์แวร์ หรือฮาร์ดแวร์ อาจจะเกิดความผิดพลาดที่เกิดจากผู้ใช้งานระบบ โดยสาเหตุหลักที่ทำให้ระบบคอมพิวเตอร์เกิดความล้มเหลวสามารถแสดงดังรูปที่ 2.8



รูปที่ 2.8 สาเหตุหลักที่ก่อให้เกิดความล้มเหลวในระบบคอมพิวเตอร์

จากสาเหตุที่ทำให้เกิดความล้มเหลวในระบบคอมพิวเตอร์จะเห็นได้ว่า มีหลายสาเหตุด้วยกันที่ทำให้เกิดความล้มเหลว เช่น ความผิดพลาดของมนุษย์ (Human Errors) ได้แก่ การทำงานที่ผิดวิธี ความเผลอ การขาดความดูแลเอาใจใส่ ยกตัวอย่างเช่น การเริ่มต้นระบบ การทำงานและการหยุดระบบที่ผิดวิธี ความผิดพลาดของหน่วยประมวลผลและหน่วยความจำ (Processor and memory failures) เป็นความล้มเหลวที่มีความรุนแรงมากถึงแม้ว่าจะไม่เกิดบ่อยๆ ก็ตาม สาเหตุของการเกิดมีได้หลายสาเหตุ ในปัจจุบันความล้มเหลวในการทำงานของหน่วยประมวลผลและหน่วยความจำ บริษัทผู้ผลิตได้ให้ความสำคัญเป็นอย่างมาก เพราะจะเป็นส่วนที่ช่วยเพิ่มความน่าเชื่อถือให้กับฮาร์ดแวร์ของตน ความล้มเหลวปราศจากร่องรอย (Mysterious Failures) เป็นความล้มเหลวหรือความผิดพลาดที่เกิดขึ้นโดยมิได้คาดไว้ ดังนั้นในระบบแบบเวลาจริงจึงไม่ได้มีการกำหนดเกี่ยวกับคุณสมบัติของความล้มเหลวชนิดนี้ เช่น ขณะทำงานตามปกติการทำงานของระบบก็หยุดกระทันหันโดยไม่มีสาเหตุว่าเกิดจากสาเหตุใด การเกิดเหตุการณ์ความล้มเหลวแบบนี้เรียกว่า ความผิดพลาดล้มเหลวโดยปราศจากร่องรอย ความล้มเหลวที่เกิดจากระบบเครือข่ายในการติดต่อสื่อสารล้มเหลว (Communication network failures) ในบางครั้งความล้มเหลวที่เกิดจากการติดต่อสื่อสารอาจเกิดขึ้นชั่วขณะซึ่งอาจเป็นคุณลักษณะของอุปกรณ์นั้น เช่น การหากมีการปิดและเปิดอุปกรณ์ใหม่อาจจะต้องใช้เวลาชั่วขณะหนึ่งในการปรับสัญญาณ ทำให้ส่วนๆ ไม่สามารถ

ติดต่อสื่อสารกับอุปกรณ์ในส่วนนี้ได้ ความล้มเหลวที่เกิดจากภาวะแวดล้อมและแหล่งจ่ายกำลังล้มเหลว (Environmental and power failures) อาจจะเกิดจากผลกระทบที่มาจากสภาวะแวดล้อม เช่น สนามแม่เหล็กไฟฟ้า หรือตัวประกอบอื่นๆ เช่น แรงดัน กระแส หรืออาจจะเป็นเรื่องของความถี่ในระบบ ความล้มเหลวที่เกิดจากอุปกรณ์ต่อพ่วง (Peripheral device failures) การล้มเหลวของระบบที่เกิดจากอุปกรณ์ต่อพ่วงแล้วทำให้ระบบล้มเหลว เป็นเหตุการณ์ที่เกิดไม่บ่อยครั้งแต่ก็มีความสำคัญ ความถี่ในการเกิดนั้นอาจเป็นการเกิดชั่วขณะหรือชั่วครู่ สาเหตุนั้นอาจจะมาจากในตัวอุปกรณ์เครื่องกลไฟฟ้าเอง ความล้มเหลวแบบค่อยเป็นค่อยไปของฐานข้อมูล (Gradual erosion of the data base) เช่น การเพิ่มความสัมพันธ์ของระบบฐานข้อมูลจนทำให้แนวความคิดที่ออกแบบความสัมพันธ์ของฐานข้อมูลในเริ่มแรกผิดเพี้ยนไปจากเดิม ทำให้ไม่สามารถจัดการฐานข้อมูลนั้นได้ทั้งหมด ความล้มเหลวที่เกิดจากการอิ่มตัวในระบบ (Saturation) เป็นความล้มเหลวที่เกิดจากการอิ่มตัวของระบบเช่น มีปริมาณข้อมูลจำนวนมากจนหน่วยประมวลผลประมวลผลเพื่อส่งผลลัพธ์ให้ส่วนที่ร้องขอไม่ทัน

2.9.3 วิธีการปรับปรุงความน่าเชื่อถือของซอฟต์แวร์

การจัดรูปแบบหรือแบ่งวิธีการปรับปรุงความน่าเชื่อถือซอฟต์แวร์ (Software Reliability Improvement Methods) มี 3 ขั้นตอนของวิธีการดังนี้คือ วิธีการออกแบบซอฟต์แวร์ที่ไว้วางใจได้ วิธีการออกแบบซอฟต์แวร์ที่ทนต่อความผิดพร่อง และวิธีการทดสอบความน่าเชื่อถือ

การออกแบบซอฟต์แวร์ที่ไว้ใจได้ (Reliable Software Design Methods)เป็นวิธีการและเทคนิคที่ช่วยนักเขียนโปรแกรมให้สามารถเขียนโปรแกรม ที่มีความวางใจได้ สามารถแบ่งวิธีการหรือเทคนิคนี้ออกเป็น 3 ส่วนได้แก่ 1.การออกแบบโปรแกรมเชิงโครงสร้าง (Structured Programming) มีวัตถุประสงค์ที่จะป้องกันความขัดแย้งหรือสับสนในแนวคิดของความต้องการของโปรแกรม เพื่อให้ง่ายต่อความเข้าใจในการออกแบบ และพัฒนาโปรแกรมว่าแต่ละส่วนมีความสัมพันธ์กันอย่างไรบ้าง มีตัวแปรอะไรบ้างที่จำเป็นต้องใช้ ตลอดรูปแบบของภาษาที่ใช้ในการเขียน เช่น GOTO ซึ่งจะต้องออกแบบว่าในโปรแกรมที่พัฒนาจะต้องใช้คำสั่งอะไรบ้าง มีการซับรูทีน และจะเรียกใช้ซับรูทีนนั้นเมื่อไร 2.การออกแบบโปรแกรมจากบนลงล่าง (Top-Down Programming) เพื่อแยกกระบวนการทำงานของโปรแกรมออกเป็นส่วนๆ เพื่อใช้ในการควบคุมโครงสร้างของโปรแกรมหรือควบคุมการไหลของโปรแกรม การออกแบบโปรแกรมจากบนลงล่างจะเริ่มจากจากแทนโมดูลของฟังก์ชันต่างๆ ในโปรแกรมโดยอาจจะแยกเป็นซับรูทีน และจะแยกว่าแต่ละซับรูทีนนั้นจะทำงานจนถึงสภาวะใดจึงจะออกจากซับรูทีนนั้น ซึ่งในภาคปฏิบัติอาจจะแทนฟังก์ชันต่างๆ เหล่านี้ด้วยแผนภาพวงจร หรือแผนภาพต้นไม้ก็เป็นได้ หรือแม้กระทั่งแผนภาพที่เป็นภาษาทางคอมพิวเตอร์ การออกแบบโปรแกรมจากบนลงล่างนี้มีประโยชน์ในการลดค่าใช้จ่ายในการทดสอบซอฟต์แวร์ และการเพิ่มความเชื่อมั่นให้กับซอฟต์แวร์ 3.การออกแบบโปรแกรมเชิงโครงสร้างข้อมูล (Data Structure Programming) มีลักษณะเป็นการกำหนด

ความสัมพันธ์และความเชื่อมโยงทางตรรกะ (Logical Linkage) เพื่อการนำมาประยุกต์ใช้งานในโปรแกรม ค่าแต่ละค่าของกลุ่มข้อมูลเหล่านี้ อาจจะเป็นข้อมูลเชิงเดี่ยว (Atomic or Scalar Data Type) เช่น จำนวนเต็ม จำนวนจริง หรือชนิดข้อมูลเชิงโครงสร้าง (Structured Data Type) เช่น สตริง อาร์เรย์ การออกแบบโปรแกรมเชิงโครงสร้างข้อมูลที่ดีและเหมาะสมจะช่วยให้โปรแกรมที่เขียนประมวลผลได้รวดเร็วและมีประสิทธิภาพ

วิธีออกแบบซอฟต์แวร์ที่ทนต่อความผิดพร่อง (Fault Tolerant Software Design Methods) เพื่อที่จะออกแบบซอฟต์แวร์ที่สามารถกู้กลับคืนสู่สภาวะที่สามารถใช้งานได้ เมื่อเกิดความผิดพลาดในการทำงาน ซึ่งสาเหตุที่จะต้องทำให้ออกแบบซอฟต์แวร์ที่ทนต่อความผิดพร่องนี้ก็เนื่องมาจากว่า ซอฟต์แวร์ที่ทำการพัฒนาขึ้นมานั้นเราจะพบว่า ไม่มีซอฟต์แวร์ใดเลยที่ไม่มีความผิดพลาด หรือในบางครั้งอาจจะไม่เจอความผิดพลาดในขณะที่ทำการทดสอบและตรวจหาความผิดพลาดแต่อาจจะพบในขณะที่นำไปใช้งาน ด้วยเหตุผลนี้จึงต้องใช้วิธีการของการออกแบบซอฟต์แวร์ที่ทนต่อความผิดพลาด เพื่อที่จะสามารถใช้งานซอฟร์แวร์นั้นได้อย่างต่อเนื่อง ปัญหาหลักของวิธีการออกแบบซอฟต์แวร์ที่ทนต่อความผิดพร่องนั่นคือ ราคาผลิตภัณฑ์ที่สูง ฉะนั้นวิธีการออกแบบดังกล่าวจึงถูกนำไปใช้ในกระบวนการ หรือใช้กับผลิตภัณฑ์ที่มีความสำคัญสูง มีความอันตรายหรือมีราคาสูง เช่น ซอฟต์แวร์ที่ใช้ควบคุมกระบวนการที่ต้องการความต่อเนื่อง ซอฟต์แวร์ที่ใช้ในการควบคุมดาวเทียม เราสามารถแบ่งวิธีออกแบบซอฟต์แวร์ที่ทนต่อความผิดพร่องได้ 3 วิธีด้วยกัน ได้แก่ 1. บล็อกสำหรับกู้ (Recovery-block) คือ วิธีการออกแบบในเรื่องที่พัฒนาชุดซอฟต์แวร์ภายใต้เงื่อนไขที่ว่า ซอฟต์แวร์แต่ละชุดนั้นมีความต้องการในการสร้างเหมือนกัน และมีออกแบบตามสมมติฐานที่ว่าแต่ละชุดของซอฟต์แวร์มีความเป็นอิสระต่อกัน ดังนั้น ความน่าจะเป็นที่จะเกิดความล้มเหลวของซอฟต์แวร์นั้นก็จะมีเพียงเล็กน้อย การปฏิบัติของวิธีบล็อกสำหรับกู้อาจแบ่งลำดับการทำงานได้สามขั้นตอนนั้นคือ การกู้คืนของสภาวะอินท์พุต การกระทำการของซอฟต์แวร์ที่เชื่อถือได้และการคืนค่าสภาวะอินท์พุต การยอมรับสภาวะของเอาท์พุต 2. ชุดโปรแกรมที่มีหลายชุด (N-version programming) เป็นการพัฒนาซอฟต์แวร์ที่มีหลายๆ ชุดและเป็นอิสระต่อกันซึ่งการทำงานจะทำงานแบบขนานกันไป ผลลัพธ์ของจะออกเอ้าท์พุตจะเกิดจากการเปรียบเทียบโดยโปรแกรมสุดท้าย เอาท์พุตที่ได้จะถูกต้องและกระบวนการที่ต่อเนื่อง 3. บล็อกกู้คืนที่มีความสอดคล้อง (Consensus recovery block) วิธีการนี้เป็นการรวมเอาวิธีการของบล็อกสำหรับกู้ และวิธีชุดโปรแกรมที่มีหลายชุดเข้าด้วยกัน โดยจะเพิ่มในส่วนของหน่วยก่อนที่จะออกเอาท์พุตนั้นคือ ส่วนของการโหวตโดยผลลัพธ์ที่ได้จะเกิดขึ้นเนื่องจากการโหวต วิธีการนี้อาจจะพูดได้ว่าเป็นวิธีการที่มีความน่าเชื่อถือกว่าแบบบล็อคกู้คืน และแบบชุดโปรแกรมที่มีหลายชุด

วิธีการทดสอบ (Testing) เพื่อทดสอบกระบวนการทำงานของซอฟต์แวร์ที่จะลดสิ่งที่ไม่ต้องการในการทำงานอันก่อให้เกิดความล้มเหลวในการทำงานนั้นมีหลายวิธีการ ซึ่งแต่ละวิธีการก็แตกต่างกันไป ซึ่งก็จำเป็นต้องเลือกใช้ให้เหมาะสมกับลักษณะของซอฟต์แวร์ที่ทำการออกแบบ

ได้แก่ 1.โมดูลทดสอบ (Module Testing) เป็นการทดสอบที่มุ่งพิจารณาทีละโมดูลโดยแยกออกจากโมดูลที่เหลืออยู่ในระบบ โดยปกติแล้วโมดูลที่ทำการทดสอบโปรแกรมจะถูกพัฒนาด้วยการเขียนโปรแกรมขึ้นมาใหม่เพื่อการทดสอบโดยเฉพาะแต่ละโมดูล เนื่องจากง่ายในการค้นพบข้อผิดพลาดได้ง่ายกว่าการทดสอบโมดูลที่โดยทั้งหมด 2. การทดสอบแบบบนลงล่าง (Top-Down Testing) เป็นการทดสอบโปรแกรมโดยรวมโปรแกรมทั้งหมดจากบนไปสู่ด้านล่างของโปรแกรมตามโครงสร้าง ข้อดีของการทดสอบแบบบนลงล่างคือ มีประสิทธิภาพในการกำหนดปัญหาของโปรแกรมด้านบนกว่าซึ่งง่ายในการทดสอบการแทนกรณีต่างๆ ที่จะเกิดขึ้นได้หลังจากเพิ่มอินต์พุตและเอาท์พุต 3.การทดสอบแบบล่างสู่บน (Bottom-Up Testing) เป็นการทดสอบโปรแกรมจากล่างสู่ด้านบนของโปรแกรมตามโครงสร้าง โดยแยกโมดูลที่ถูกทดสอบออกจากโมดูลอื่นในระบบ การทำสอบจะแทนค่าอินพุตท์ในโมดูลที่ทดสอบเพื่อดูฟังก์ชันการทำงานและความสัมพันธ์กับโมดูลด้านบน นอกจากโมดูลที่กำลังทดสอบและโมดูลอยู่ลำดับที่สูงกว่าแล้ว จะไม่มีการเรียกใช้งานโมดูลอื่นที่ไม่ใช้สองโมดูลนี้ ข้อดีของการทดสอบแบบนี้คือ มีประสิทธิภาพในการกำหนดปัญหาของโปรแกรมด้านล่างกว่า หรือโมดูลที่อยู่ด้านล่างกว่านั่นคือ เมื่อแทนกรณีต่างๆ ในโมดูลทดสอบซึ่งอยู่ต่ำกว่าจะพบการเปลี่ยนแปลงของฟังก์ชันการทำงานโมดูลที่สูงกว่าตามลำดับ ทำให้สามารถกำหนดขอบเขตของอินต์พุตในโมดูลทดสอบได้ดี เพื่อให้โมดูลที่สูงกว่าตามลำดับทำงานได้ถูกต้องและได้เอาท์พุตที่ต้องการ 4. การทดสอบแบบแซน-วิช (Sandwich Testing) เป็นการทดสอบที่รวมการทดสอบแบบบนลงล่าง และแบบล่างสู่บนเข้าด้วยกัน การทดสอบแบบนี้จะเห็นได้ชัดว่าเป็นความต้องการที่จะรวมข้อดีและต้องการกำจัดข้อเสีย ของการทดสอบแบบบนลงล่าง และแบบล่างสู่บนนั่นเอง

2.9.4 แบบจำลองความน่าเชื่อถือของซอฟต์แวร์ (Software Reliability Modes)

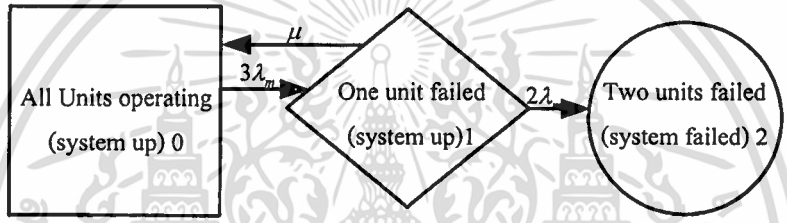
แบบจำลองความน่าเชื่อถือของซอฟต์แวร์นั้น สามารถแบ่งเป็นกลุ่มออกเป็น 4 กลุ่มได้แก่ กลุ่มที่เกี่ยวกับการก่อความผิดพร่อง (Fault seeding) กลุ่มที่เกี่ยวกับการนับจำนวนความผิดพร่อง (Failure count) กลุ่มที่เกี่ยวกับระยะเวลาระหว่างความล้มเหลว (Times between failures) กลุ่มที่เกี่ยวข้องกับโดเมนของตัวป้อน (Input domain based) โดยแต่ละกลุ่มก็มีแบบจำลองต่างๆ ให้เลือกใช้ที่เหมาะสมต่างกันไป เช่น แบบจำลองของมิลล์ (Mills Model) แบบจำลองของมูซา (Musa Model) แบบจำลองของซูแมน (Shooman Model) แบบจำลองกำลัง (Power Model) แบบจำลองของอากาศยาน (Air Force Model)

2.9.5 การป้องกันความล้มเหลวในระบบคอมพิวเตอร์ (Fault Masking)

การป้องกันความล้วเหลวที่มีในระบบคอมพิวเตอร์นั้น จะเป็นกล่าวถึงการทำให้ระบบคอมพิวเตอร์มีส่วนที่เรียกว่า โมดูลซ้ำสำรอง (Redundancy Modular) ซึ่งเป็นการนำหลักการที่ได้กล่าวไว้แล้วในเรื่องของแบบจำลองและการหาความน่าเชื่อถือของระบบ แบบจำลองที่มีโมดูลซ้ำ

สำรอง3โมดูล (Triple Modular Redundancy, TMR) ในถูกนำมาใช้ในงานวิจัยชิ้นนี้ โดยแบบจำลองดังกล่าวมีลักษณะคือ ประกอบด้วยองค์ประกอบจำนวน 3 องค์ประกอบที่เป็นอิสระต่อกันแต่มีเป้าหมายในการทำงานที่เป็นส่วนซ้ำซ้อนเหมือนกัน ส่วนออกขององค์ประกอบทั้งสามจะเข้าสู่หน่วยของการเปรียบเทียบด้วยการโหวต การทดลองแบบจำลองดังกล่าวถูกนำเสนอครั้งแรกโดย Von Neumann แบบจำลองการต่อมีลักษณะดังรูปที่ 2.5

เมื่อพิจารณาระบบที่มีความซ้ำซ้อนแบบสามส่วนว่า มีการโหวตแบบสมบูรณ์และแต่ละโมดูลเป็นอิสระต่อกันและสามารถทำการซ่อมแซมเมื่อเกิดความล้มเหลวได้ เมื่อมีโมดูลหนึ่งเกิดความล้มเหลวในระบบที่มีความซ้ำซ้อนสามส่วน โมดูลนั้นจะถูกซ่อมแซมทันที หากว่ามีมากกว่าหนึ่งโมดูลที่เกิดความล้มเหลวในระบบ ระบบแบบที่มีความซ้ำซ้อนสามส่วนจะไม่ถูกซ่อมแซม สามารถแสดงปริภูมิสถานะของระบบ (System state space) ดังรูปที่ 2.9



รูปที่ 2.9 ระบบแบบโมดูลซ้ำสำรอง 3 โมดูลที่มีปริภูมิสถานะแบบซ่อมแซมได้

จากรูปที่ 2.10 จะสามารถพิจารณาได้ว่า ระบบที่ถูกสร้างขึ้นมาจากโมดูลองค์ประกอบทั้ง 3 ส่วนนั้น เป็นอิสระต่อกันและมีคุณลักษณะที่เหมือนกันทุกประการ ระบบจะสามารถทำงานได้ไม่ล้มเหลวจนกระทั่งมีองค์ประกอบที่ล้มเหลวมากกว่าหนึ่งโมดูล อัตราส่วนความล้มเหลวและอัตราส่วนการซ่อมแซมมีค่าคงที่ เมื่อองค์ประกอบที่เกิดความล้มเหลวแล้ว เมื่อซ่อมแซมเสร็จนั้นเสมือนว่าองค์ประกอบนั้นเป็นองค์ประกอบใหม่ การพิจารณาแบบนี้เราใช้เทคนิคที่เรียกว่าเทคนิคของมาร์คอร์ฟ (Markov technique)

กำหนดให้ j คือ เป็นภาวะที่เกิดในระบบโดยที่

i = 0 (ระบบเริ่มต้นทำงาน)

i =1 (จำนวนหนึ่งโมดูลเกิดความล้มเหลว แต่ระบบยังคงทำงานอยู่)

i=2 (โมดูลมากว่าหนึ่งเกิดความล้มเหลว และระบบก็เกิดความล้มเหลว)

$P_i'(t)$ คือ ความน่าจะเป็นที่ระบบที่มีโมดูลซ้ำสำรอง 3 โมดูล จะอยู่ในภาวะ j ที่เวลา t โดยที่ j = 0,1,2

λ_m คือ อัตราส่วนการล้มเหลวของมอดูล (Failure rate)

μ คือ อัตราส่วนการซ่อมแซม (Repair rate)

สามารถเขียนสมการสมการเชิงอนุพันธ์ได้คือ

$$P_1^{'}(t) = \mu P_1^{'}(t) - 3\lambda_m P_0(t)$$
$$P_1^{'}(t) = 3\lambda_m P_0(t) - (2\lambda + \mu)P_1(t)$$
$$P_2^{'}(t) = 2\lambda_m P_1(t)$$

กำหนดให้เวลาที่ $t = 0, P_0(0) = 1$ และ $P_1(0) = P_1(0) = 0$ สามารถหาความน่าเชื่อถือของระบบที่มีการซ่อมแซมได้โดย

$$R_{TMRr} = P_0(t) + P_1(t)$$

บทที่ 3
การจัดโครงสร้างระบบที่ทนต่อความผิดพร่องในการทดลอง

ในหัวข้อนี้จะกล่าวถึง พื้นฐานคุณสมบัติที่ควรมีของระบบที่ทนต่อความผิดพร่อง การประยุกต์ใช้งานระบบที่มีโมดูลซ้ำสำรอง 3 โมดูล วิธีการโหวต และวิธีการคำนวณหาค่าอัตราการล้มเหลวตามมาตรฐาน MIL-HDBK-217B [19] ตลอดจนการนำหลักการของระบบที่ทนต่อความผิดพร่องไปการออกแบบโครงสร้างการทดลอง

3.1 พื้นฐานระบบที่ทนต่อความผิดพร่อง

การออกแบบและพัฒนาระบบให้มีคุณสมบัติทนต่อความผิดพร่อง ต้องพิจารณาประเด็นต่างๆ ตามเงื่อนไขความต้องการของระบบที่ทนต่อความผิดพร่อง เช่น การตรวจจับความผิดพร่อง (Fault Detection) สาเหตุที่ทำให้เกิดความผิดพร่อง (Fault Containment) ตำแหน่งความผิดพร่อง (Fault Location) การกู้คืนจากความผิดพร่อง (Fault Recovery) การปิดความผิดพร่อง (Fault Masking) เป็นต้น ในประเด็นต่างๆ เหล่านี้จะถูกนำมาใช้พิจารณาร่วมกับหลักการหลักระบบที่มีโมดูลซ้ำสำรอง (Concept of Redundancy) [19] เพื่อให้ระบบที่ออกแบบเป็นระบบที่ทนต่อความผิดพร่อง

ในปัจจุบันการออกแบบระบบที่ทนต่อความผิดพร่อง โดยใช้หลักการของการมีโมดูลซ้ำสำรองในระบบ และอาจแยกชนิดของโมดูลซ้ำสำรองตามลักษณะของโมดูลได้ 2 ชนิด ได้แก่ โมดูลซ้ำสำรองที่เป็นฮาร์ดแวร์ (Hardware Redundancy) และโมดูลซ้ำสำรองที่เป็นซอฟต์แวร์ นอกจากนี้ ยังมีโมดูลซ้ำสำรองส่วนที่เป็นสารสนเทศ (Information Redundancy) และโมดูลซ้ำสำรองส่วนของเวลา (Time Redundancy) ซึ่งอยู่ที่เงื่อนไขความต้องการของระบบและผู้ออกแบบ ในวิทยานิพนธ์ฉบับนี้ได้ใช้โมดูลซ้ำสำรองส่วนที่เป็นซอฟต์แวร์และฮาร์ดแวร์เป็นหลัก ในการเพิ่มความน่าเชื่อถือให้กับระบบ

โมดูลซ้ำสำรองที่เป็นฮาร์ดแวร์ มี 3 ชนิดที่เป็นพื้นฐาน [19] คือ แพสซิฟ (Passive) แอ็กทิฟ (Active) และแบบลูกผสม (Hybrid) โดยแบบแพสซิฟจะใช้เทคนิคของหลักการปิดความผิดพร่องเพื่อไม่ให้เกิดความผิดพร่องที่ขึ้น และป้องกันความผิดพร่องจากความผิดพลาดด้วย อาจกล่าวได้ว่า การมีโมดูลซ้ำสำรองที่เป็นฮาร์ดแวร์แบบแพสซิฟ เพื่อให้ระบบมีคุณสมบัติที่ทนต่อความผิดพร่องนั้น ไม่ต้องทำการอะไรเพิ่มเติมจากระบบเดิม เพราะเป็นใช้เป็นพื้นฐานในการป้องกันความผิดพร่องที่อาจเกิดขึ้นกับระบบในขณะที่ออกแบบอยู่แล้ว โมดูลฮาร์ดแวร์ที่ซ้ำสำรองแบบแอ็กทิฟ บางครั้งเรียกว่า วิธีการแบบพลวัต (Dynamic Method) [19] ใช้เทคนิคการตรวจหาความผิดพร่องในช่วงเวลาการทำงาน เมื่อพบความผิดพร่องจะมีการกระทำบางอย่าง เช่น การตัด

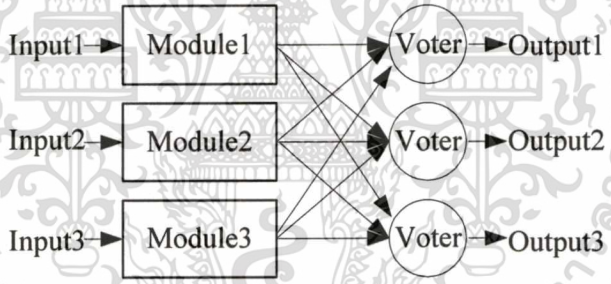
ส่วนที่เกิดความผิดพร่องนั้นออกจากระบบ แล้วมีการกู้คืนจากความผิดพร่อง โมดูลซ้ำสำรองที่เป็นฮาร์ดแวร์แบบสุดท้ายคือแบบลูกผสม ได้นำข้อดีของแบบแพสซิฟและแอ็กทิฟมาใช้งาน การปิดบังความผิดพร่องถูกนำมาใช้ป้องกันความผิดพลาดที่อาจเกิดขึ้น การตรวจจับความผิดพลาด การกู้คืนจากความผิดพร่อง คุณสมบัติเหล่านี้ล้วนถูกนำมาใช้ในระบบที่มีโมดูลซ้ำสำรองที่เป็นฮาร์ดแวร์แบบผสม วิธีการแบบลูกผสมนี้บ่อยครั้งถูกนำมาใช้ในกระบวนการที่มีความสำคัญ ที่ต้องการไม่ให้เกิดความล้มเหลวในการทำงานหรือต้องการความน่าเชื่อถือสูงๆ

โมดูลซ้ำสำรองที่เป็นซอฟต์แวร์ โดยถูกประยุกต์ใช้ในระบบคอมพิวเตอร์ซึ่งทำงานโดยผ่านทางซอฟต์แวร์ เช่น การตรวจจับความผิดพลาด หรือการทำงานให้ระบบทนต่อความผิดพร่อง เป็นต้น หลักการของโมดูลซ้ำสำรองที่เป็นแบบซอฟต์แวร์อาจแบ่งได้เป็น 3 ชนิด คือ การตรวจสอบความต้องกัน (Consistency Checks) การตรวจสอบสมรรถภาพ (Capability Checks) และวิธีการทำซ้ำซอฟต์แวร์ (Software Replication Methods) [19] การตรวจสอบความต้องกันใช้การพิสูจน์ความถูกต้องของสารสนเทศ โดยพิจารณาความรู้ด้านในคุณสมบัติของสารสนเทศนั้น การตรวจสอบแบบง่ายๆ ส่วนใหญ่จะใช้ฮาร์ดแวร์เป็นตัวตรวจสอบ แต่ก็มีจำนวนมากที่ใช้ซอฟต์แวร์เป็นตัวตรวจสอบ เช่น ในกระบวนการควบคุมที่มีการสุ่มตัวอย่างและเก็บข้อมูลจากตัวตรวจจับในกระบวนการจำนวนมาก แต่ละตัวตัวจับจะถูกตรวจสอบความถูกต้องให้อยู่ในช่วงที่ยอมรับค่าได้ ค่าความต้องกันนั้นอาจถูกใช้เปรียบเทียบเพื่อเป็นตัวชี้วัดการปฏิบัติงานของระบบได้อีกด้วย โมดูลซ้ำสำรองที่เป็นซอฟต์แวร์ชนิดต่อมาคือ ชนิดตรวจสอบสมรรถภาพ ถือเป็นการตรวจพิสูจน์ยืนยันว่าระบบมีความสามารถในการทำงานได้ตามที่กำหนดหรือไม่ เช่น หากระบบมีหน่วยประมวลผลหลายโมดูลเมื่อทำงานแล้ว ประสิทธิภาพของแต่ละหน่วยประมวลผลมีค่าเป็นเท่าไรและผลอย่างไรกับระบบเมื่อเทียบกับการมีหน่วยประมวลผลเดียว ส่วนนี้สามารถตรวจสอบผ่านทางซอฟต์แวร์ได้ โมดูลซ้ำสำรองที่เป็นแบบซอฟต์แวร์ชนิดสุดท้ายคือ การทำซ้ำซอฟต์แวร์หรือการทำซอฟต์แวร์หลายชุดนั่นเอง (N-Version Programming) [19] การทำโมดูลซอฟต์แวร์ซ้ำสำรองจำนวน N ชุดนั้นอาจไม่ได้ผลเนื่องจาก หากแต่ละชุดถูกพัฒนามาเหมือนกัน ฉะนั้นเมื่อเกิดความพร่องอย่างหนึ่งเกิดขึ้นจะทำให้เกิดความผิดพร่องทุกๆ โมดูลเช่นกัน ดังนั้นหากต้องการนำโมดูลซอฟต์แวร์ซ้ำสำรองไปใช้งานแล้ว จำเป็นต้องออกแบบให้มีการป้องกันความผิดพร่องเป็นอย่างดี หลักการของการใช้โปรแกรมจำนวน N ชุดได้ถูกนำเสนอโดย Auizieniz ในปี 1978 [19] โดยมีการออกแบบและเขียนโปรแกรมจำนวน N ชุด ซึ่งแต่ละชุดแบ่งกลุ่มการออกแบบไปโดยโปรแกรมเมอร์ แต่ละกลุ่มจะออกแบบโปรแกรมภายใต้ข้อกำหนด และความต้องการของระบบเดียวกัน

3.2 ระบบที่ทนต่อความผิดพร่องแบบมีโมดูลซ้ำสำรอง 3 โมดูล

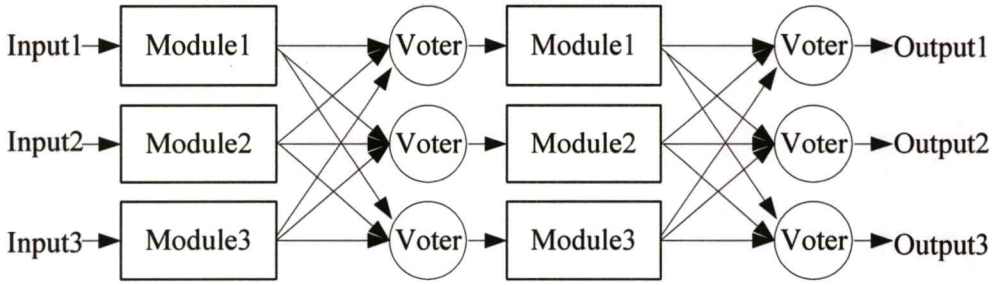
พื้นฐานการทำงานของระบบที่มีโมดูลซ้ำสำรอง 3 โมดูลจะประกอบด้วยฮาร์ดแวร์หรือซอฟต์แวร์จำนวน 3 โมดูลทำงานโดยมีตัวโหวตทำหน้าที่เลือกเอาท์พุตของระบบ ถ้ามีโมดูลหนึ่งใดเกิดความล้มเหลวอีก 2 เอาท์พุตของโมดูลที่เหลือจะถูกเลือกไปใช้งาน การทำงานและการต่อเชื่อมของระบบหรืออุปกรณ์มีลักษณะดังแสดงในรูปที่ 2.5 ในทางปฏิบัติโมดูลแต่ละโมดูลอาจแทนด้วยหน่วยประมวลผล หน่วยความจำ หรือหน่วยของซอฟต์แวร์ก็ได้ซึ่งโมดูลซอฟต์แวร์ทั้ง 3 โมดูลจะมีการพัฒนาคนละรุ่นกัน แต่มีข้อกำหนดในการทำงานเดียวกัน เมื่อพิจารณาในรูปที่ 2.5 จะพบว่า หากตัวโหวตเกิดความล้มเหลวในการทำงานแล้วก็อาจจะทำให้ระบบเกิดความล้มเหลวได้เหมือนกัน ดังนั้น ตัวโหวตคะแนนควรต้องมีความน่าเชื่อถือสูง

ระบบที่ทนต่อความผิดพร่องแบบมีโมดูลซ้ำสำรอง 3 โมดูลยังสามารถประยุกต์นำไปใช้งานดังรูปที่ 3.1 ซึ่งระบบประกอบด้วยโมดูลที่เป็นอิสระจำนวน 3 โมดูลโดยแต่ละโมดูลจะรับอินพุตแล้วนำไปประมวลผลแต่ละโมดูล ผลลัพธ์ที่ได้จากการโหวตคะแนนแต่ละโมดูลก็จะเหมือนกันจนกว่าจะมีอินพุตหนึ่ง เกิดความล้มเหลวขึ้นจึงจะทำให้เอาท์พุตต่างไป ทำให้สามารถเปรียบเทียบสภาวะที่อาจเกิดความล้มเหลวได้หลายสภาวะ



รูปที่ 3.1 โมดูลซ้ำสำรอง 3 โมดูลพร้อมทั้งตัวโหวต 3 โมดูล

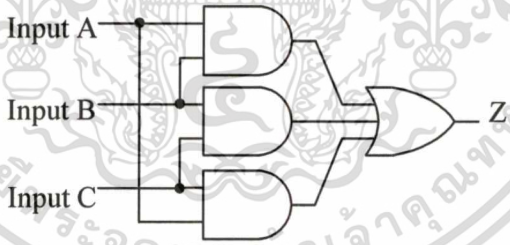
นอกจากนี้ ระบบแบบโมดูลซ้ำสำรอง 3 โมดูลยังสามารถนำสภาวะต่างๆ ที่มีความเชื่อมโยงถึงกันมาพิจารณาได้ ดังรูปที่ 3.2 ถ้าตัวโหวตเกิดความล้มเหลวในระบบช่วงสภาวะที่หนึ่งแล้ว สภาวะหลังต่อมาที่รับเอาท์พุตของสภาวะแรกมาเป็นอินพุตของสภาวะหลัง ก็จะรับสัญญาณที่ผิดพลาดเข้ามาเช่นกัน ระบบที่มีการทำซ้ำสำรอง 3 โมดูลที่ตัวโหวตโดยทั่วไปถูกเรียกว่า การคืนสภาวะขององค์ประกอบ (Restoring Organ) [19] เนื่องจากมีการตรวจแก้สัญญาณเอาท์พุตเมื่อเกิดเหตุการณ์ที่มีหนึ่งอย่างน้อยหนึ่งอินพุตเกิดความล้มเหลว นั้นก็คือสัญญาณที่ได้จะปลอดจากความผิดพร่องนั่นเอง



รูปที่ 3.2 โมดูลซ้ำสำรองที่มีตัวโหวต 3 โมดูลที่มีความไวต่อการล้มเหลวของตัวโหวต

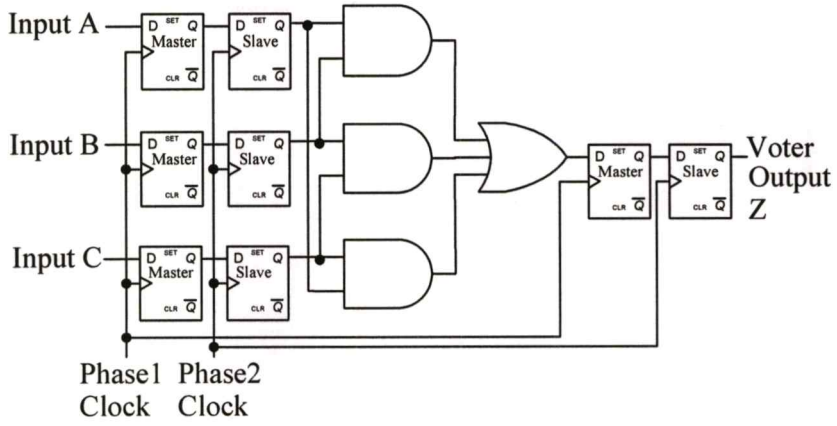
3.3 การโหวตคะแนน

การโหวตในระบบที่มีโมดูลซ้ำสำรอง N โมดูล (NMR System) [8] ถูกนำไปประยุกต์ใช้อย่างมากมายในงานอุตสาหกรรม เช่น การควบคุมอุณหภูมิในกระบวนการทางเคมี ซึ่งมีการรับสัญญาณจากอุปกรณ์วัดอุณหภูมิจำนวน 3 โมดูล การนำไปใช้งานตัวโหวตจะเลือกอินพุตจากสัญญาณของอุปกรณ์วัดทั้ง 3 ไปใช้งาน การโหวตนั้นสามารถกระทำกับสัญญาณที่เป็นแอนะล็อกและที่เป็นดิจิทัล จากตัวอย่างนอกจากจะใช้ตัวโหวตเป็นตัวถ่วงดุลสัญญาณทั้ง 3 จากอุปกรณ์วัดและสามารถใช้แก้ปัญหาในกระบวนการได้ด้วย โมดูลโหวตที่เป็นฮาร์ดแวร์ ในการออกแบบหรือก่อนนำไปใช้งานนั้น นิยมทำการออกแบบหรือตรวจสอบด้วยซอฟต์แวร์ก่อน หากเป็นข้อมูลชนิดดิจิทัล นิยมใช้โมดูลโหวตชนิดฮาร์ดแวร์ ซึ่งมีความง่ายในการออกแบบและใช้งาน เช่น รูปที่ 3.3

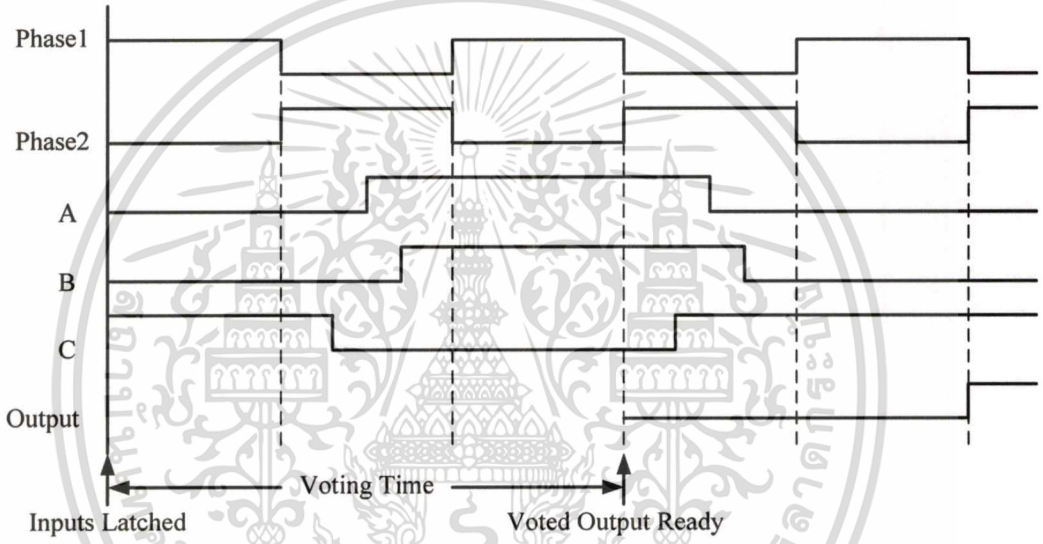


รูปที่ 3.3 รูปแบบการโหวตลงคะแนน โดยเงื่อนไขต้องมีอินพุต 2 ใน3 ทำงานได้

จากรูปที่ 3.3 เอาท์พุตของการโหวตจะมีสถานะเป็น 1 เมื่อสัญญาณอินพุตส่วนใหญ่มีสถานะเป็น 1 และเอาท์พุตของการโหวตจะเป็น 0 เมื่อสัญญาณอินพุตส่วนใหญ่เป็น 0 โครงสร้างของตัวโหวตในรูปที่ 3.3 สามารถเป็นแบบ 8 บิตและ 16 บิต ขึ้นอยู่กับการออกแบบและนำไปใช้งาน ซึ่งแต่ละวงจรจะทำงานเป็นอิสระแต่มีความสัมพันธ์กัน เวลาที่ต้องการใช้ในโมดูลโหวตชนิดฮาร์ดแวร์นี้ ในการทำงานจะขึ้นอยู่กับเวลาหน่วงในการแพร่กระจายของวงจรดิจิทัล



รูปที่ 3.4 ฮาร์ดแวร์ของตัวโหวตที่มีการประสานเวลาของอินพุตและเอาท์พุต



รูปที่ 3.5 แผนกำหนดเวลาการประสานเวลาของตัวโหวต

ในการประยุกต์ใช้งาน เวลาที่ใช้ในการโหวตเป็นตัวแปรสำคัญในกระบวนการ หากค่าของสัญญาณที่เข้ามามีความคลาดเคลื่อนหรือเหลื่อมล้ำเพียงนิดเดียว ก็จะทำให้การประมวลผลของตัวโหวตคะแนนคลาดเคลื่อนไป วงจรฟลิปฟลอป (Flip Flops Circuit) [19] [20] ถูกนำมาใช้เพื่อแก้ปัญหานี้ โดยใช้อินพุตของตัวโหวตทำการประสานเวลากับการโหวตในกระบวนการ ดังเช่นในรูปที่ 3.4 ใช้หลักการของดีฟลิปฟลอป (D-Flip Flops) มาใช้ในการประสานเวลาของตัวโหวต และหากพิจารณาไดอะแกรมของเวลาในรูปที่ 3.5 พบว่าการทำงานจะทำงาน 2 ช่วงสัญญาณเวลา โดยมีฟลิปฟลอปที่ทำงานแบบหลักและรอง (Master-Slave Flip Flops) [19] โดยอินพุตที่ส่งไปยังหน่วยโหวตจะถูกเก็บในฟลิปฟลอปตัวหลัก (Master Flip Flops) ขณะที่สัญญาณเวลาเป็นบวกขาขึ้น บนสัญญาณบวกขาขึ้นของช่วงเวลาทำงานช่วงที่ 2 ข้อมูลที่เข้ามาก็จะถูกเก็บในฟลิปฟลอปตัวรอง (Slave Flip Flop) แล้วข้อมูลทั้งสองจะถูกรวมกันในวงจรของการโหวตต่อไป การรวมกันของข้อมูลในวงจรจะถูกเก็บเป็นเอาท์พุตของฟลิบฟลอบตัวหลักบน

สัญญาณบวกขาขึ้นในช่วงการทำงานที่ 1 และเอาท์พุตของตัวโหวต (Z) ก็จะปรากฏหลังจากเกิดขอบสัญญาณขาขึ้นในช่วงที่ 2

นอกจากจะใช้โมดูลโหวตที่เป็นฮาร์ดแวร์แล้ว ยังสามารถใช้โมดูลที่เป็นซอฟต์แวร์ โดยกลไกการทำงานจะใช้รูทีนซอฟต์แวร์เพื่อจัดการการโหวต หากพิจารณาตัวอย่างในภาพที่ 3.6 ซึ่งเป็นระบบของหน่วยประมวลผลที่ใช้ซอฟต์แวร์ในการโหวต อุปกรณ์ตรวจวัดมีการสุ่มค่าและเก็บข้อมูลในหน่วยความจำสองช่องทาง (Two-port Memory) ทั้ง 3 โมดูล ในหน่วยความจำสองช่องทางจะประกอบด้วยวงจรสหสัญญาณ (Multiplexer) และทำการเข้าสหสัญญาณระหว่างสองข้อมูลที่มาจากสองผู้ใช้งานโดยมีความเร็วเพียงพอกับการใช้งาน หน่วยประมวลผลจะอ่านค่าจากเครื่องมือวัดได้ทางหน่วยความจำทั้งสาม การโหวตด้วยโปรแกรมสามารถดูค่าจากการสุ่มตัวอย่างเพื่อทำการเปรียบเทียบ หลังจากมีการประมวลผลของแต่ละหน่วยประมวลผลแล้ว ก็จะทำการเก็บค่าลงในหน่วยความจำที่อยู่ในระดับถัดมา หลักการของการทำงานเช่นนี้จะเหมือนกับการทำงานของระบบในรูปที่ 3.2 โดยแต่ละตัวโหวตจะทำงานด้วยซอฟต์แวร์และกระจายไปสู่หน่วยประมวลผลอื่นๆ ให้ทราบเมื่อมีความล้มเหลวเกิดขึ้น

3.4 วิธีคำนวณหาค่าอัตราการล้มเหลว

วิธีการประเมินความสามารถระบบที่ทนต่อความผิดพร่องมี 2 วิธีการหลัก [19] คือ การประเมินค่าเชิงปริมาณ (Quantitative) และการประเมินค่าเชิงคุณภาพ (Qualitative) การประเมินค่าเชิงคุณภาพ เป็นค่าที่ได้มาจากการวัดโดยใช้ความรู้สึก หรือจิตวิจัย (Subjective) เพื่อเปรียบเทียบเชิงพรรณนาระหว่างสิ่งหนึ่งกับอีกสิ่งหนึ่ง หรือกับสิ่งอื่นๆ เช่น การประเมินระบบที่ทนต่อความผิดพร่องว่าเหมาะสมและสะดวกกับการใช้งานเพียงใดจากผู้ใช้งานระบบ ซึ่งก็จะมีความเห็นที่เป็นความรู้สึกของผู้ใช้งานแตกต่างกันไป การประเมินค่าเชิงปริมาณเป็นเทคนิควิธีการ กำหนดลักษณะประจำของแต่ละสมาชิกในระบบเพื่อสามารถเปรียบเทียบกับระบบอื่นๆ เช่น ความน่าเชื่อถือของระบบหนึ่งที่อาจจะมีมากกว่าอีกระบบหนึ่ง จะต้องพิจารณาลักษณะประจำหลายอย่างด้วยกัน เช่น อัตราการล้มเหลว (Failure Rate) ค่าเฉลี่ยเวลาที่จะล้มเหลว (Mean Time to Failure: MTTF) ค่าเฉลี่ยเวลาระหว่างการล้มเหลว (Mean Time between failure: MTBF) เป็นต้น

ค่าอัตราการล้มเหลว (Failure Rate Calculation) เป็นค่าที่มีความสำคัญมากในการวิเคราะห์และประเมินความน่าเชื่อถือของระบบ เทคนิคที่พื้นฐานที่นิยมใช้คือ วิธีประมาณค่าอัตราความล้มเหลวตามข้อมูลและวิธีการของ United State Department of Defense (USDOD) MIL-HDBK-217 Standard [19] ซึ่งก็มีอยู่หลายเวอร์ชันด้วยกัน เช่น USDOD1965 [3] USDOD1974 [3] และ USDOD1979 [19] อย่างไรก็ตามทุกเวอร์ชันก็อยู่บนมาตรฐานเดียวกับ MIL-HDBK-217 ในทุกเวอร์ชันมีจุดประสงค์เพื่อพัฒนาแบบจำลองของอัตราการล้มเหลว ในอุปกรณ์อิเล็กทรอนิกส์โดยใช้วิธีการทดลองวิเคราะห์การล้มเหลวที่เกิดขึ้นจริงกับอุปกรณ์นั้นๆ ยกตัวอย่างเช่น ตาม

มาตรฐาน MIL-HDBK-217B ที่ถูกพัฒนาโดย Siewiorek และ Swarz ในปี 1982 [19] และ USDOD1974 [19] มีแบบจำลองในการทำนายค่าคงที่ของวงจรรวม (IC) ด้วยสมการดังนี้คือ $\lambda = \pi_L \pi_Q (C_1 \pi_T + C_2 \pi_E) \pi_P$ failures per million hours โดยที่ π_L คือ ค่าตัวประกอบในการเรียนรู้ (Learning Factor) π_T คือ ค่าองค์ประกอบอุณหภูมิ (Temperature Factor) π_E คือ ค่าองค์ประกอบของสิ่งแวดล้อม π_P คือ ค่าองค์ประกอบของพิน (Pin Factor) และ C_1 และ C_2 คือ องค์ประกอบที่ซับซ้อน (Complexity Factors) โดยสามารถแสดงข้อมูลบางส่วนที่ใช้มาตรฐาน MIL-HDBK-217B คำนวณหาค่าอัตราการล้มเหลวในอุปกรณ์อิเล็กทรอนิกส์ ดังแสดงในตารางที่ 3.1 [19]

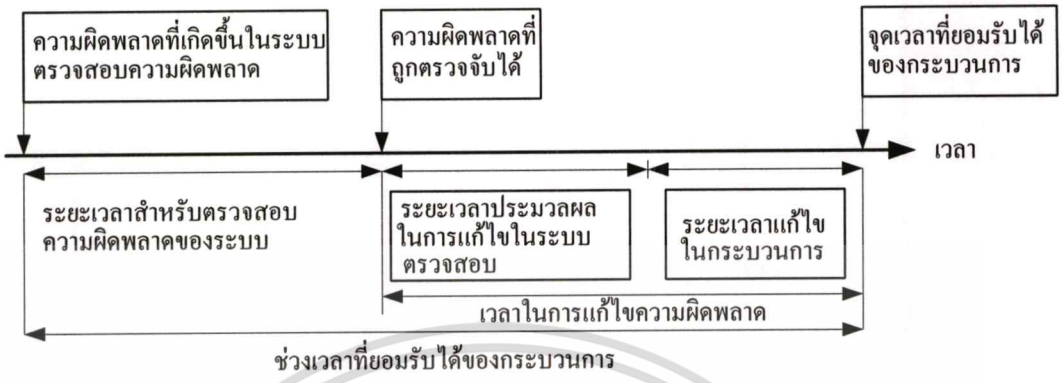
ตารางที่ 3.1 ตัวอย่างการคำนวณอัตราการล้มเหลว ตามมาตรฐาน MIL-HDBK-217B [19]

Typical failure rate calculate using MIL-HDBK-217B	
$(\pi_L = 1, \pi_Q = 16, \pi_T = 0.35, \pi_E = 0.2, \pi_P = 1)$	**Failure Rate (Failures per million hours)**
(a) Logic Circuits (Number of logic gates)	
50	0.1527
100	0.2312
200	0.3655
500	1.4483
1000	14.4880
(b) Memories (RAM) (Number of bits)	
1,024(1K)	0.8837
2,048(2K)	1.3491
8,192(8K)	3.1453
16,384(16K)	4.8033
32,768(32K)	7.3362

3.5 การควบคุมความผิดพลาด

การควบคุมความผิดพลาด (Failure Control Methodology) ที่อาจจะเกิดขึ้นกับระบบควบคุมกระบวนการให้อยู่ในช่วงเวลาที่ยอมรับได้ของกระบวนการ (Fault Tolerance Time of the Process) มีความจำเป็นอย่างยิ่งกับระบบหรือกระบวนการที่ต้องทำงานอย่างต่อเนื่อง ถ้ามีความผิดพลาดเกิดขึ้นกับระบบควบคุม โมดูลแบบซ้ำสำรองของระบบควบคุมที่ทนต่อความผิดพร่องต้องสามารถทำงานทดแทนโมดูลที่เกิดความผิดพลาดในการทำงาน หากใช้เวลาอยู่ในช่วงเวลานี้

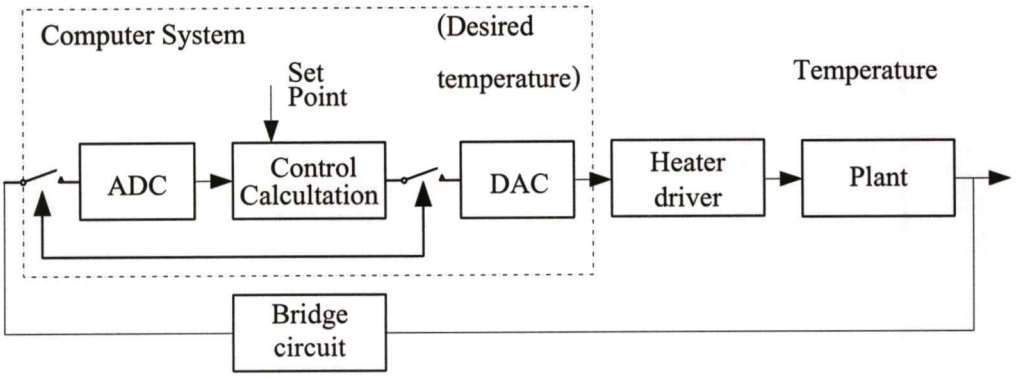
แล้วจึงจะทำให้ระบบควบคุมสามารถควบคุมกระบวนการต่อไปได้ โดยช่วงเวลานี้จะประมาณ 1 วินาที [3] ดังนั้นจะต้องใช้เวลาในการกำจัดหรือแก้ไขความผิดพลาดต่างๆ ที่อาจเกิดขึ้นให้อยู่ภายในช่วงระยะเวลานี้ ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 ช่วงเวลายอมรับได้ของกระบวนการโดยทั่วไป

หากพิจารณาในรูปที่ 3.7 เป็นระบบเวลาจริงแบบแข็งของลูปการควบคุมอุณหภูมิของระบบเป่าลมร้อนซึ่งในเทอมของการควบคุม ลูปของอุณหภูมิคือ ค่าของข้อมูลที่ทำการสุ่มเข้ามา (Sampled Data System) กลไกของการควบคุมสำหรับระบบนี้คือ T_s ซึ่งเป็นเวลาของการสุ่ม ถ้าสมมุติว่า เวลาของการสุ่มข้อมูลคือ 10 มิลลิวินาที ดังนั้นที่ 10 มิลลิวินาทีค่าอินพุตของอุณหภูมิจะต้องถูกอ่านค่าเข้ามาที่ในส่วนการประมวลผล หลังจากนั้นค่าของระบบการควบคุมจะถูกส่งออกไป และค่าเอาท์พุตของวาล์วก็จะคำนวณค่าแล้วส่งไปที่ตัวขับของระบบอุณหภูมิ

ข้อบังคับในเรื่องของเวลาที่มีความสัมพันธ์กับการทำงาน เช่น ถ้าสมมติว่าตัวเป่าลมร้อนถูกนำมาใช้เพื่อที่จะทำให้อุปกรณ์แห้ง อุปกรณ์อาจจะเสียหายได้ ถ้าการเป่านั้นทำให้อุณหภูมิที่เกิดขึ้นกับอุปกรณ์สูงกว่า 50 องศาเซลเซียส ที่ใช้เวลาน้อยกว่า 10 วินาที ซึ่งถ้าเกิดเหตุการณ์นี้ขึ้นมา คอมพิวเตอร์จะต้องทราบค่าและจะต้องสั่งให้ตัวทำความร้อนหยุดทำงาน ในกรณีของการเป่าลมร้อนซึ่งโอกาสของเหตุการณ์ที่เกิดขึ้นนี้มีสูง ค่าของข้อมูลที่สุ่มมานั้นจะไม่มีความสำคัญต่อระบบ ถ้า $9.95 \le T_s \le 10.05$ มิลลิวินาที ซึ่งอยู่ในค่ากลางของ T_s มิลลิวินาที อย่างไรก็ตามระบบจะไม่เป็นไปตามข้อกำหนดถ้าเวลาในการสุ่มอยู่ในช่วง $10 \le T_s \le 1000$ มิลลิวินาที กับค่ากลางของ $T_s = 10$ มิลลิวินาที ตลอดคาบเวลา 24 ชั่วโมง

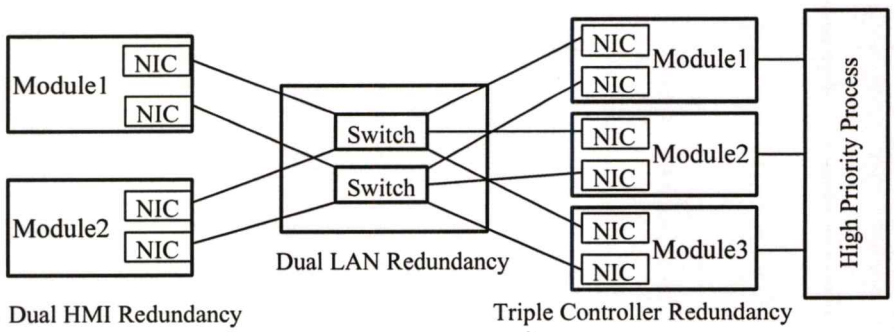


รูปที่ 3.7 บล็อกไดอะแกรมของระบบควบคุมคอมพิวเตอร์

รูปแบบการประมวลผลสำหรับระบบควบคุมที่ทนต่อความผิดพร่อง มีหลายรูปแบบด้วยกัน เช่น ระบบการประมวลผลด้วยฮาร์ดแวร์ซึ่งอาจแบ่งได้ 8 รูปแบบ [3] ได้แก่ 1oo1(One out of One Voting) 1oo2(One out of Two Voting) 2oo2(Two out of Two Voting) 1oo1D(One out of One Voting with Diagnostic System) 2oo3(Two out of Three Voting) 2oo2D(Two out of Two Voting with Diagnostic System) 1oo2D(One out of Two Voting with Diagnostic System) 1oo2D and Comparison หรือระบบประมวลผลแบบด้วยสนามแม่เหล็ก แต่สำหรับในวิทยานิพนธ์นี้ใช้เทคนิคของโปรแกรมในส่วนประมวลผล สำหรับระบบควบคุมที่ทนต่อความผิดพร่อง

3.6 โครงสร้างการทดลองระบบความคุมที่ทนความผิดพร่อง

การออกแบบโครงสร้างการทดลอง โดยการประยุกต์หลักการของระบบที่ทนต่อความผิดพร่องและคุณสมบัติในระบบควบคุมแบบคอมพิวเตอร์ มาเพิ่มความน่าเชื่อถือให้กับระบบควบคุมกระบวนการ สามารถแบ่งการทดลองออกเป็น 3 ส่วน คือ โมดูลซ้ำสำรองส่วนเอชเอ็มไอ โมดูลซ้ำสำรองส่วนเครื่องควบคุม และโมดูลซ้ำสำรองส่วนระบบเครือข่ายอินทราเน็ต โดยมีโปรแกรมจัดการที่ถูกพัฒนาขึ้นด้วยโปรแกรม Microsoft Visual C++ เวอร์ชัน 6.0 SP 5 เป็นตัวจัดการการทำงานของเครื่องคอมพิวเตอร์และระบบเครือข่าย สามารถแสดงโครงสร้างโดยรวมของการทดลองดังภาพที่ 3.8



รูปที่ 3.8 การจัดโครงสร้างระบบในการทดลอง

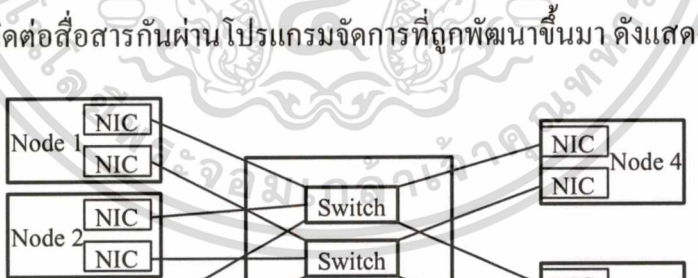
การทำงานของระบบซึ่งแบ่งตามหน้าที่การทำงาน และโครงสร้างระบบที่ออกแบบไว้สำหรับการทดลองมีดังนี้คือ

กระบวนการที่ใช้เป็นกระบวนการจำลองความดัน หากพิจารณารูปที่ 3.8 ตามโครงสร้างจะอยู่ในส่วนของ High Priority Process และทำงานอยู่บนเครื่องคอมพิวเตอร์จำลองกระบวนการ ซึ่งกระบวนการจำลองนี้ได้ถูกพัฒนาขึ้นด้วยโปรแกรม MATLAB 6.5 สาเหตุที่เลือกกระบวนการจำลองเนื่องมาจาก ต้องการศึกษาเหตุการณ์ต่างๆ ที่อาจเกิดขึ้นกับกระบวนการซึ่งอาจจะไม่เกิดขึ้นในทุกกระบวนการ อีกทั้งโปรแกรม MATLAB มีเครื่องมือมาตรฐานในการพัฒนาอย่างมากมายและมีประสิทธิภาพ สัญญาณจากแบบจำลองกระบวนการนี้จะมีการรับส่งค่าระหว่างแบบจำลองกระบวนการและเครื่องควบคุมผ่านทางแผงวงจรต่อประสานอินทราเน็ต โดยใช้การสื่อสารแบบดีดีอี (Data Dynamic Exchange)

สถานีควบคุมทั้ง 3 สถานีจะได้รับสัญญาณผ่านทางแผงต่อวงจรต่อประสานอินทราเน็ต ซึ่งถูกติดตั้งไว้แต่ละสถานี โดยจะได้รับสัญญาณพร้อมกันทั้ง 3 สถานี การรับส่งข้อมูลจะถูกจัดการด้วยโปรแกรมจัดการที่ถูกพัฒนาขึ้น นอกจากนี้แต่ละสถานีต้องลงโปรแกรม Sixnet I/O Tool Kit และ ISaGRAF เพื่อทำการติดต่อกับอินทราเน็ตไอ/โออีกด้วย

สถานีเอชเอ็มไอเป็นส่วนที่ติดต่อกับผู้ใช้งานซึ่งจะรับและส่งค่าจากกระบวนการ ผ่านทางระบบเครือข่ายอินทราเน็ต โดยใช้คุณสมบัติที่มีในโปรแกรมเอชเอ็มไอ ซึ่งจะต้องพารามิเตอร์ในโปรแกรมเอ็ชเอ็มไอว่า จะติดต่อกับสถานีควบคุมใดบ้างโดยมีเงื่อนไขอย่างไร

ระบบเครือข่ายที่ถูกออกแบบไว้สำหรับการทำระบบเครือข่ายที่ทนต่อความผิดพร่อง หากพิจารณาจะพบว่า ประกอบไปด้วยระบบเครือข่ายจำนวน 2 เครือข่ายโดยจะทำงานเป็นอิสระต่อกันและสามารถติดต่อสื่อสารกันผ่านโปรแกรมจัดการที่ถูกพัฒนาขึ้นมา ดังแสดงในรูปที่ 3.9



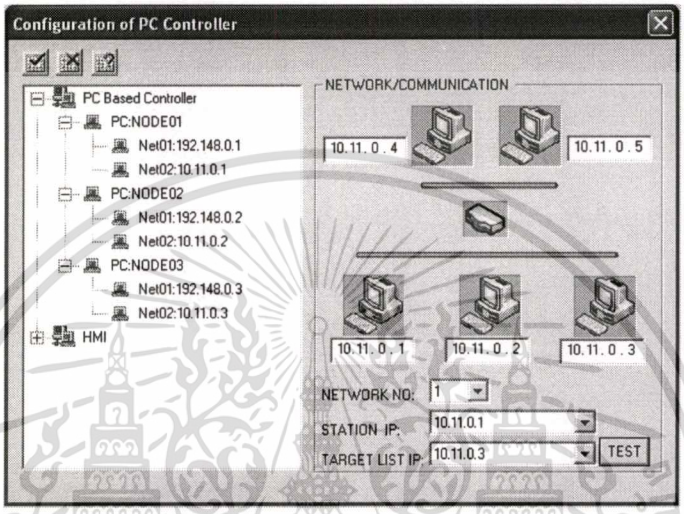
รูปที่ 3.9 ระบบเครือข่ายอินทราเน็ตแบบมีโมดูลซ้ำสำรอง

จากโครงสร้างของระบบที่ได้ออกแบบไว้ จำเป็นต้องมีการจัดระบบในส่วนต่างๆ และมีการกำหนดค่าพารามิเตอร์ในแต่ละส่วนดังนี้

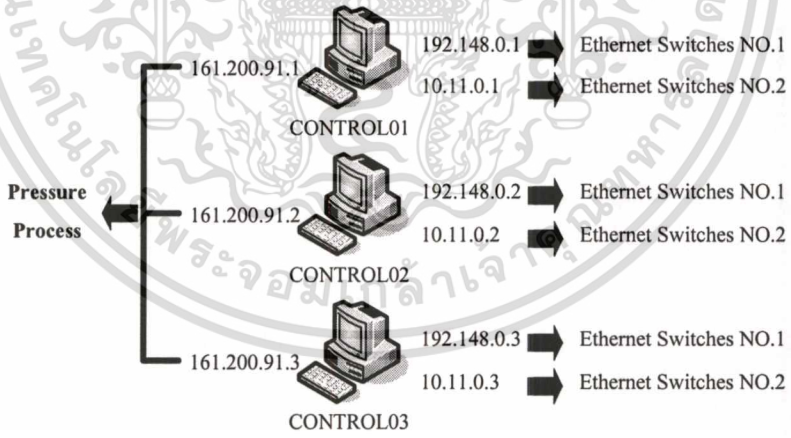
3.6.1 การจัดระบบของสถานีควบคุม

แบบจำลองของสถานีควบคุมที่ใช้ในงานวิจัยนี้คือ แบบจำลองทีเอ็มอาร์ ซึ่งนิยมในมาประยุกต์ใช้กับระบบคอมพิวเตอร์ [16] สถานีควบคุมที่ออกแบบจะประกอบไปด้วย 3 สถานีแต่ละ

สถานีจะถูกติดตั้งแผงวงจรต่อประสานสถานีละ 3 ชุดเพื่อใช้ติดต่อสื่อสารบนระบบเครือข่ายทั้ง 2 ลูปและการรับส่งค่าของกระบวนการจากชุดจำลองกระบวนการ นอกจากนี้ยังได้ลงโปรแกรม SIXNET I/O Tool Kit เพื่อให้โปรแกรมที่ถูกพัฒนาขึ้นมา สามารถติดต่อสื่อสารกับแบบจำลองกระบวนการผ่านทางหน่วยอินทราเน็ตไอ/โอ (SixTRAK) ได้ โดยมีข้อตกลงในการสื่อสารแบบดีดีอี/โอพีซี หลังจากนั้นจะต้องกำหนดค่าไอพีให้กับแผงวงจรต่อประสานและโปรแกรมจัดการอีกด้วย ดังแสดงในรูปที่ 3.10 ก. และรูปที่ 3.10 ข.



รูปที่ 3.10 ก. กำหนดค่าเลขที่อยู่ไอพีให้กับโปรแกรมจัดการที่พัฒนาขึ้น



รูปที่ 3.10 ข. กำหนดค่าเลขที่อยู่ไอพีให้กับแผงวงจรต่อประสานอินทราเน็ต

นอกจากการกำหนดค่าเลขที่อยู่ไอพีแล้วจะต้องกำหนดค่าพารามิเตอร์ต่างๆ กับโปรแกรมจัดการที่ถูกพัฒนาขึ้นอีกด้วยซึ่งจะได้กล่าวในหัวข้อต่อไป

ในงานวิจัยนี้ได้เลือกใช้โปรแกรม ISaGRAF เป็นตัวควบคุมกระบวนการ สาเหตุที่ทำให้ใช้โปรแกรม ISaGRAF เนื่องมาจากว่าเป็นโปรแกรมที่ถูกสร้างตามมาตรฐานของ IEC 1131-3 ซึ่งสามารถทำงานร่วมกับผลิตภัณฑ์ที่แหล่งผู้ผลิตมาตรฐาน เช่น Allen-Bradley Omron เป็นต้นและอีกประการหนึ่งที่เลือกใช้โปแกรม ISaGRAF นั่นก็คือ ต้องการควบคุมความคลาดเคลื่อนที่มาจาก

การพัฒนาโปรแกรมขึ้นมาเองถ้าหากไม่รัดกุมแล้วจะมีผลกระทบต่อผลการทดลองโดยตรง การสื่อสารระหว่างโปรแกรมที่ใช้ควบคุมกระบวนการคือ ISaGRAF และหน่วยอินทราเน็ตไอ/โอที่รับค่าจากแบบจำลองการควบคุมความดันจะติดต่อสื่อสารผ่านโอพีซี (OLE for Process Control) ซึ่งโปรแกรมที่พัฒนาขึ้นมาจัดการก็จะรับและส่งค่ากับอุปกรณ์ทั้ง 2 ด้านผ่าน ดีดีอี/โอพีซี เหมือนกัน

3.6.2 การจัดระบบของสถานีเอชเอ็มไอ

สถานีเอชเอ็มไอที่ใช้ในการทดลองประกอบด้วยคอมพิวเตอร์จำนวน 2 เครื่อง ทำหน้าที่เป็นส่วนที่ติดต่อสื่อสารกับผู้ปฏิบัติงานโดยทำงานบนระบบเครือข่ายอินทราเน็ตซึ่งทั้ง 2 เครื่องจะมีส่วนประกอบของฮาร์ดแวร์ที่เหมือนกันเพื่อสามารถทำงานทดแทนกันได้ ซึ่งแต่ละเครื่องจะประกอบไปด้วยแผงวงจรต่อประสานระบบเครือข่ายอินทราเน็ตจำนวน 2 ชุด

การทดลองในเบื้องต้น ผู้วิจัยได้พัฒนาโปรแกรมเอชเอ็มไอขึ้นเพื่อการทดลอง พบว่า การเขียนโปรแกรมเอชเอ็มไอเพื่อแสดงสภาวะจากกระบวนการขึ้นเองของผู้วิจัยนั้น มีกระทบต่อผลการทดลอง เนื่องจากการเขียนโปรแกรมหากไม่มีความรัดกุมแล้ว โปรแกรมต้องใช้เวลาช่วงหนึ่งในการประมวลผลในตัวโปรแกรมเอง ทำให้จะมีผลกระทบในส่วนของการจัดเวลาการทำงานของระบบ ทำให้การทดลองแต่ละครั้งหากมีการเปลี่ยนแปลงในส่วนของโปรแกรมเอชเอ็มไอที่พัฒนาขึ้น จะทำให้ผลการทดลองด้านเวลาเปลี่ยนไป ทำให้ผู้วิจัยตัดสินใจเลือกใช้โปรแกรมเอชเอ็มไอที่เป็นมาตรฐาน ด้วยเหตุผลที่ต้องการจะควบคุมตัวแปรเพื่อให้ผลการทดลองถูกต้องมากที่สุดโดยได้เลือกใช้โปรแกรมอินทัช (Intouch) ในงานวิจัยนี้

การกำหนดค่าพารามิเตอร์สำหรับสถานีเอชเอ็มไอสามารถแบ่งออกได้ 2 ส่วน คือ ส่วนที่จะต้องกำหนดเลขที่อยู่ไอพีบนแผงวงจรต่อประสาน ซึ่งถูกติดตั้งสถานีละ 2 เพื่อให้สามารถทำงานทดแทนกันบนระบบเครือข่าย 2 ลูปได้ โดยมีการกำหนดเลขที่อยู่ไอพีให้กับแผงวงจรต่อประสานอินทราเน็ตดังแสดงในรูปที่ 3.11

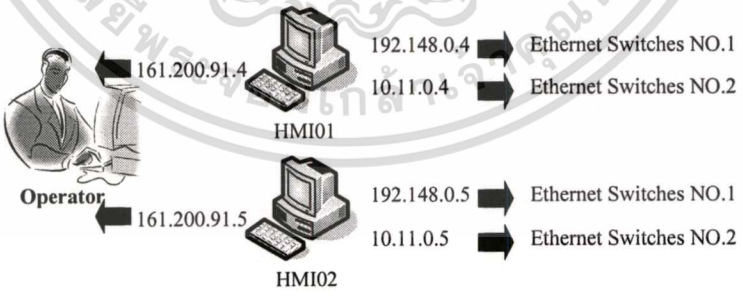
การกำหนดค่าพารามิเตอร์สำหรับสถานีเอชเอ็มไอส่วนที่ 2 เป็นการกำหนดพารามิเตอร์บนโปรแกรมเอชเอ็มไอเอง เพื่อให้สามารถติดต่อสื่อสารและทำงานร่วมกับสถานีอื่นๆได้ โดยมีรายละเอียดการกำหนดค่าพารามิเตอร์ดังนี้

ชื่อตัวแปร (Tag Name) เป็นการกำหนดค่าพารามิเตอร์หรือตัวแปรที่ต้องการแสดงผลหรือควบคุมในกระบวนการผ่านทางโปรแกรมเอชเอ็มไอ เช่น การแสดงค่าความดันที่เกิดขึ้นในกระบวนการที่ได้จากเครื่องวัด การควบคุมและแสดงสถานะเปิดปิดของวาวล์ เป็นต้น โดยจะต้องกำหนดชนิดตัวแปรให้ถูกต้อง ได้แก่ แอนะล็อก ดิจิทัล หรือแบบอักษร

ชื่อสำหรับเข้าถึง (Access Name) เป็นการกำหนดทางเข้าของข้อมูลซึ่งอาจจะมาจากกระบวนการหรือมาจากโปรแกรมประยุกต์อื่นๆ โปรแกรมเอชเอ็มไอสามารถกำหนดชื่อทางเข้าได้จำนวนไม่จำกัด โดยองค์ประกอบภายในแต่ละชื่อการเข้าถึงนั้นจะต้องกำหนด สถานีของ

แหล่งกำเนิดข้อมูล (Node Name) โดยอาจจะกำหนดเป็นชื่อสถานีหรือเลขที่อยู่ไอพีก็ได้ จากนั้นจะต้องกำหนดว่าโปรแกรมขับหรือโปรแกรมประยุกต์ (Application Name) ที่จะให้ข้อมูลนั้นชื่ออะไร เช่น การอ่านจากแผงวงจรต่อประสานไอ/โอแบบอินทราเน็ตที่มีชื่อโปรแกรมประยุกต์ว่า applicom หรือโปรแกรมเอ็กเซลล์มีชื่อโปรแกรมประยุกต์ว่า excel ซึ่งชื่อต่างๆ ดังที่กล่าวมานี้เป็นข้อกำหนดจากบริษัทผู้ผลิต จากนั้นจะต้องกำหนดหัวข้อที่จะสื่อสาร (Topic) การกำหนดหัวข้อในการสื่อสารนี้อาจจะกำหนดชื่ออะไรก็ได้ แต่ควรกำหนดให้สามารถเข้าใจได้ง่ายเช่น กำหนดว่า pressure_plant_zone1 ก็สามารถทราบได้ว่าน่าจะเกี่ยวข้องกับกระบวนการความดัน โดยนอกจากที่จะกำหนดหัวข้อที่จะสื่อสารบนโปรแกรมเอชเอ็มไอแล้ว ยังจะต้องกำหนดบนโปรแกรมขับโดยจะต้องกำหนดให้ตรงกันทั้งในโปรแกรมเอชเอ็มไอและโปรแกรมขับ เช่น หากต้องการติดต่อกับโปรแกรมที่พัฒนาขึ้นมาจัดการ หัวข้อที่จะสื่อสารจะต้องตรงกับที่กำหนดไว้ในโปรแกรมที่พัฒนาขึ้น หากไม่ไม่ตรงกันก็จะสามารถทำการติดต่อสื่อสารกันได้

ไอเท็มเนม (Item Name) เป็นการกำหนดเลขที่อยู่ของอุปกรณ์ในตัวแปรนั้น หลังจากที่ทำการกำหนดชื่อสำหรับเข้าถึง สถานีของแหล่งกำเนิดข้อมูล และหัวข้อในการสื่อสารแล้ว ยังจำเป็นต้องกำหนดว่าจะติดต่อสื่อสารกับอุปกรณ์ตัวใดด้วย ยกตัวอย่าง ในกระบวนการควบคุมความดันที่ตั้งหัวข้อการสื่อสารว่า pressure_plant_zone1 ภายในกระบวนการนี้อาจจะประกอบไปด้วยวาล์วควบคุมจำนวน 100 ตัว ทำให้ต้องกำหนดไปว่าจะไปทำการแสดงผลและควบคุมกับวาล์วตัวที่เท่าไร เช่น กำหนด Valve_001_Zone01 เป็นการแสดงผลและควบคุมวาวล์ควบคุมตัวที่หนึ่ง ซึ่งอยู่ในโซนที่หนึ่ง โดยการกำหนดในส่วนนี้จะต้องมีการอ้างอิงหน่วยของไอ/โอที่ต่อจริงกับกระบวนการด้วย อย่างไรก็ตามหากเปรียบเทียบกับเครื่องควบคุมแบบตรรกะแล้ว ไอเท็มเนมก็เปรียบเสมือนได้กับหน่วยของไอ/โอ ที่มีการกำหนดเลขที่อยู่ในหน่วยความจำนั้นเอง



รูปที่ 3.11 การกำหนดค่าพารามิเตอร์บนแผงต่อประสานระบบเครือข่ายอินทราเน็ต

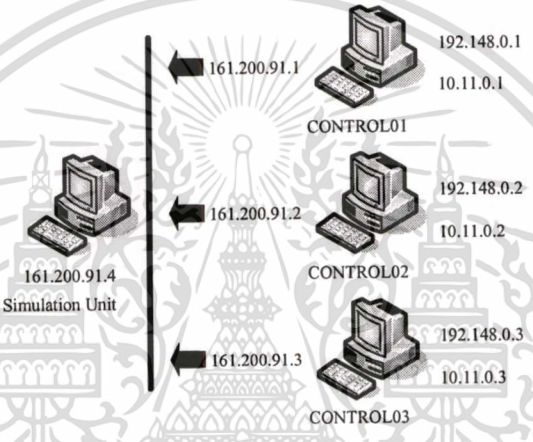
3.6.3 การจัดแบบจำลองกระบวนการควบคุมความดัน

การจัดแบบจำลองกระบวนการควบคุมความดันในงานวิจัยฉบับนี้ ได้ถูกพัฒนาขึ้นโดยใช้ศักยภาพความสามารถของเครื่องมือต่างๆ ที่มีอย่างมากมายในโปรแกรม MATLAB 6.5 เพื่อจำลองสถานการณ์ที่สามารถเกิดขึ้นได้กับกระบวนการ เนื่องจากโปรแกรม MATLAB มีความสามารถจำลองกระบวนการได้เสมือนกระบวนการจริง และยังสามารถจำลองเหตุการณ์ที่

อาจเกิดขึ้นได้จริงกับกระบวนการได้ครอบคลุม เช่น การวิเคราะห์เรื่องสัญญาณรบกวนที่อาจจะเกิดขึ้นในได้ในระบบควบคุมแบบคอมพิวเตอร์โปรแกรม MATLAB จึงถูกเลือกใช้ในงานวิจัยนี้

เครื่องคอมพิวเตอร์ 1 เครื่องถูกจัดเพื่อเป็นสถานีของแบบจำลองกระบวนการซึ่งจะทำการติดตั้งแผงวงจรต่อประสานระบบเครือข่ายอินทราเน็ตจำนวน 3 ชุด เพื่อให้สามารถสื่อสารส่งผ่านข้อมูลไปที่สถานีควบคุมทั้ง 3 ได้พร้อมๆ กันโดยโปรแกรมที่ถูกพัฒนาขึ้นมาจะเป็นตัวจัดการในการติดต่อสื่อสารกับสถานีควบคุม เช่น สภาวะการเชื่อมต่อ การส่งข้อมูลแบบแพร่กระจาย

การกำหนดพารามิเตอร์จะต้องกำหนดเลขที่อยู่ไอพีของให้อยู่ในวงเดียวกับเลขที่อยู่ไอพีที่กำหนดในแผนวงจรต่อประสานอินทราเน็ตไอ/โอ หากไม่อยู่ในลูปเดียวกันจะทำให้ไม่สามารถที่จะทำการติดต่อสื่อสารกันได้ ซึ่งสามารถแสดงในรูปที่ 3.12



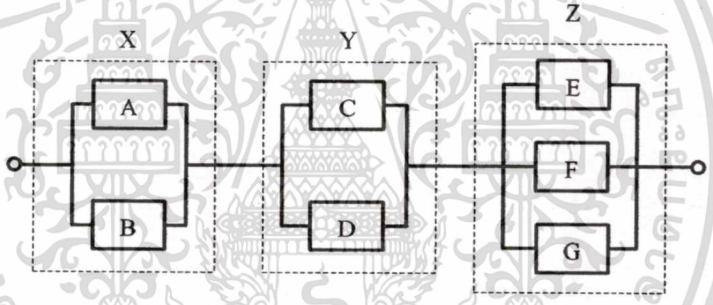
รูปที่ 3.12 การกำหนดพารามิเตอร์ของหน่วยแบบจำลองกระบวนการ

บทที่ 4
การหาค่าความน่าเชื่อถือของระบบและผลการทดลอง

ในบทนี้จะกล่าวถึง การหาค่าความน่าเชื่อถือของระบบที่ออกแบบในการทดลอง โดยใช้แบบจำลองความน่าเชื่อถือทางคณิตศาสตร์ และจากการสังเกตพฤติกรรมของสัญญาณในการทดลอง

4.1 แบบจำลองทางการเชื่อมต่อกันของอุปกรณ์ในการทดลอง

หากพิจารณาการจัดโครงสร้างของระบบในการทดลองในรูปที่ 3.8 จะพบว่าสามารถเขียนลักษณะของการเชื่อมต่อกันทางกายภาพของแต่ละโมดูลได้ใหม่ เพื่อให้ง่ายต่อการพิจารณา ดังรูปที่ 4.1



รูปที่ 4.1 ไดอะแกรมการเชื่อมต่อกันของอุปกรณ์ในการทดลอง

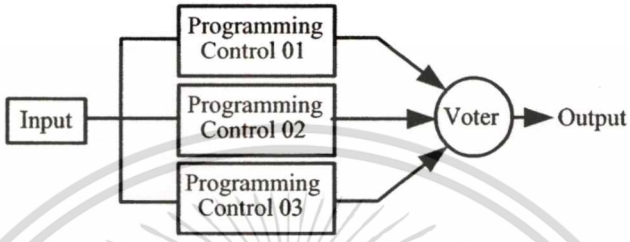
จากรูปที่ 4.1 กำหนดให้ A และ B คือ สถานีเอชเอ็มไอ C และ D คือ ระบบเครือข่ายอีเทอร์เน็ต 2 ลูป E, F และ G คือ สถานีควบคุมทั้ง 3 สถานีที่ใช้ควบคุมและรับค่าสัญญาณมาจากกระบวนการ

หากแบ่งตามหน้าที่การทำงาน (Functional) สามารถได้ 3 กลุ่มคือ กลุ่ม X ซึ่งประกอบไปด้วยโมดูลเอชเอ็มไอ A และ B กลุ่ม Y ซึ่งประกอบไปด้วยโมดูลของระบบเครือข่ายอีเทอร์เน็ต C และ D กลุ่ม Z ซึ่งประกอบไปด้วยโมดูลของเครื่องควบคุม E, F และ G ซึ่งทั้ง 3 กลุ่มเมื่อพิจารณาจากทางอินพุตไปสู่เอาท์พุต จะพบว่ามีการเชื่อมต่อกันแบบอนุกรมกันอยู่ สามารถหาความน่าเชื่อถือของระบบโดยการคิดความน่าเชื่อถือระบบแบบมีเงื่อนไขแต่ละกลุ่ม ซึ่งมีการเชื่อมต่อกันแบบขนาน แล้วความน่าเชื่อถือแต่ละกลุ่มมารวมกันแบบอนุกรม

การทำงานของระบบในแต่ละส่วน ประกอบไปด้วย การทำงานของส่วนที่เป็นฮาร์ดแวร์และส่วนที่เป็นซอฟต์แวร์ ซึ่งจะต้องทำงานร่วมกัน ดังนั้น ในการพิจารณาเรื่องความน่าเชื่อถือของระบบที่ออกแบบในการทดลอง จะต้องพิจารณาการทำงานของซอฟต์แวร์และฮาร์ดแวร์ซึ่งต้องทำงานร่วมกันในแต่ละส่วนด้วย เช่น การทำงานในส่วนที่เป็นสถานีควบคุม เป็นต้น

4.2 การจัดรูปแบบการควบคุม

การจัดรูปแบบการควบคุมเป็นการใช้โมดูลซ้ำสำรอง 3 โมดูลรับค่าจากกระบวนการหรืออุปกรณ์การวัดพร้อมกันทั้ง 3 โมดูลเพื่อนำมาประมวลผล หลังจากประมวลผลแล้วจะส่งผลลัพธ์ของการประมวลผลแต่ละโมดูลให้กับโมดูลที่ใช้สำหรับโหวตคะแนน เพื่อเลือกผลลัพธ์เพียงหนึ่งเดียวจากโมดูลประมวลผลออกสู่กระบวนการหรืออุปกรณ์ควบคุม ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 ระบบควบคุมที่ทนทานต่อความผิดพร่อง

ระบบควบคุมที่ทนต่อความผิดพร่องนี้ ได้ใช้ทฤษฎีด้านวิศวกรรมความน่าเชื่อถือเพื่อเพิ่มความน่าเชื่อถือให้กับระบบควบคุมโดยรวม อีกทั้งยังเป็นรูปแบบที่นิยมใช้ในระบบคอมพิวเตอร์ เนื่องจากโปรแกรมควบคุมทั้ง 3 จะทำงานพร้อมกันตลอดเวลา ดังนั้น หากระบบจะหยุดการทำงานก็ต่อเมื่อทั้ง 3 โปรแกรมหยุดการทำงานเสียก่อน

ในส่วนของการโหวต (Vote) เป็นการเลือกข้อมูลเพื่อส่งออกเอาท์พุต ตรงส่วนนี้จะถูกจัดการด้วยโปรแกรมจัดการที่ถูกพัฒนาขึ้น ซึ่งก็จะทำงานตามเงื่อนไขดังตารางที่ 4.1 [29] ในการทดลองนี้ได้กำหนดให้มีโมดูลอย่างน้อย 1 โมดูลสามารถงานได้สำเร็จ ระบบโดยรวมจึงจะไม่เกิดความล้มเหลว

ตารางที่ 4.1 เงื่อนไขการลงคะแนนเพื่อเลือกข้อมูลส่งออกเอาท์พุต

โปรแกรมควบคุม			หมายเลขที่ถูกเลือก (Vote)
สถานีที่1	สถานีที่2	สถานีที่3	
ทำงาน	ทำงาน	ทำงาน	สถานีที่ 3
ทำงาน	ไม่ทำงาน	ทำงาน	สถานีที่ 3
ไม่ทำงาน	ไม่ทำงาน	ทำงาน	สถานีที่ 3
ทำงาน	ทำงาน	ไม่ทำงาน	สถานีที่ 2
ไม่ทำงาน	ทำงาน	ไม่ทำงาน	สถานีที่ 2
ทำงาน	ไม่ทำงาน	ไม่ทำงาน	สถานีที่ 1

ข้อมูลในตารางที่ 4.1 เป็นเงื่อนไขการลงคะแนนเพื่อเลือกข้อมูลของตัวโหวต ซึ่งผู้วิจัยได้นำข้อมูลนี้มาจากผู้วิจัยอื่น [29] ซึ่งได้วิจัยไว้ก่อนมาใช้ โครงสร้างของการโหวตคะแนนนั้น Shiva

และ Wakerly [20] ได้เสนอพื้นฐานสำหรับการโหวตคะแนนโดยใช้การเขียนเงื่อนไขจากตารางความเป็นจริง (Truth Table) ในปี 1988 และ 1994 [20] ตามลำดับ ตัวอย่างเช่น หากเมื่อออกแบบเงื่อนไขในการโหวตแล้ว สามารถนำมาเขียนตารางความเป็นจริงได้ดังตารางที่ 4.2

ตารางที่ 4.2 ตารางความเป็นจริง

Input			**Output**
x_1	x_2	x_3	$f(x_1x_2x_3)$
0	0	0	Two
0	0	1	Or
0	1	0	Three
1	0	0	Zeroes
1	1	0	Two
1	0	1	Or
0	1	1	Three
1	1	1	ones

ในตารางที่ 4.2 สามารถเขียนเป็นสมการได้คือ

$$f_v(x_1x_2x_3) = x_1x_2\overline{x_3} + x_1\overline{x_2}x_3 + \overline{x_1}x_2x_3$$

และสามารถใช้แผนที่คาร์นอฟ (Karnaugh Map) [20] เพื่อช่วยในการลดรูปสมการได้คือ

ตารางที่ 4.3 การใช้แผนที่คาร์นอฟเพื่อลดรูปสมการ

x_1 \ x_2x_3	**00**	**01**	**11**	**10**
0	0	0	1	0
1	0	1	1	1

หลังจากใช้แผนคาร์นอฟเพื่อลดรูปสมการ จะทำให้ได้สมการเทอมที่ประหยัดคือ

$$f_v(x_1x_2x_3) = x_1x_2 + x_1x_3 + x_2x_3$$

จากเงื่อนไขของการโหวตคะแนนที่นำมาใช้ในงานวิจัยนี้ เมื่อนำมาเขียนบนตารางความเป็นจริงแล้วจะทำให้ได้ตารางความเป็นจริงดังตารางที่ 4.4 และเมื่อเขียนให้อยู่ในรูปของแผนที่คาร์นอฟ จะทำให้ได้แผนที่คาร์นอฟดังตารางที่ 4.5

ตารางที่ 4.4 ตารางความเป็นจริงตามเงื่อนไขโหวตคะแนน

Input			Output
x_1	x_2	x_3	$f(x_1x_2x_3)$
1	1	1	$1(x_3)$
1	0	1	$1(x_3)$
0	0	1	$1(x_3)$
1	1	0	$1(x_2)$
0	1	0	$1(x_2)$
1	0	1	$1(x_1)$
0	0	0	0
1	1	1	$1(x_3)$

ตารางที่ 4.5 การใช้แผนที่คาร์นอฟเพื่อลดรูปสมการในงานวิจัย

x_1 \ x_2x_3	00	01	11	10
0	0	1	1	1
1	1	1	1	1

จากตารางที่ 4.5 จะทำให้ได้สมการทั่วไปของการโหวตคือ

$$f_v(x_1x_2x_3) = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3} + x_1\overline{x_2}\,\overline{x_3} + x_1\overline{x_2}x_3 + x_1x_2x_3 + x_1x_2\overline{x_3}$$

ระบบควบคุมที่ทนต่อความผิดพร่องที่ออกแบบสำหรับการทดลองนั้น หากพิจารณาถึงสาเหตุที่จะทำให้เกิดความผิดพร่องขึ้นในระบบจะพบว่า มีสาเหตุหลัก 2 ประการ คือ ความผิดพร่องที่เกิดจากซอฟแวร์และความผิดพลาดที่เกิดจากฮาร์ดแวร์ ผู้วิจัยจึงแบ่งรูปแบบการทำให้ระบบหรืออุปกรณ์เกิดความล้มเหลว เพื่อต้องการศึกษาพฤติกรรมและวิเคราะห์หาความน่าเชื่อถือของระบบ โดยมีรูปแบบที่ทำให้ส่วนต่างๆ ของระบบเกิดความผิดพร่องดังนี้

เพื่อต้องการศึกษาพฤติกรรมของระบบในแง่ของสัญญาณการควบคุม ผู้วิจัยได้กำหนดให้แต่ละส่วนเกิดความล้มเหลวแบบสมบูรณ์เกิดขึ้น โดยความล้มเหลวแบบสมบูรณ์ที่เกิดขึ้นในแต่ละสถานี หมายถึง การไม่สามารถใช้งานได้ของทั้งสถานีนั้นโดยอาจจะเป็นสถานีเอชเอ็มไอ หรือสถานีควบคุม นั่นหมายความว่า อาจเกิดความล้มเหลวขึ้นในระบบในส่วนที่เป็นฮาร์ดแวร์ หรือซอฟต์แวร์ก็ได้ ซึ่งมีพฤติกรรมคล้ายกับการสถานีนั้นไม่มีแหล่งจ่ายไฟฟ้าจึงไม่สามารถใช้งานได้

เพื่อสังเกตพฤติกรรมในการทดลองจึงใช้วิธีการทำการหยุดจ่ายแหล่งกำลังไฟฟ้าให้กับสถานีทีละสถานี แล้วสังเกตพฤติกรรมและเก็บข้อมูลการเปลี่ยนแปลงในระบบทั้งหมด

การทดสอบการเกิดความล้มเหลวในส่วนที่สอง เพื่อยืนยันการพฤติกรรมที่ได้จากการสังเกตสัญญาณ ผู้วิจัยได้ใช้ทฤษฎีในการคำนวณหาค่าความน่าเชื่อถือ การทดสอบซอฟต์แวร์ มาคำนวณหาค่าความน่าเชื่อถือ เมื่อระบบเกิดความล้มเหลวในส่วนต่างๆ ตามแบบจำลองทางคณิตศาสตร์ที่ถูกนำมาใช้

4.3 การหาความน่าเชื่อถือของระบบในส่วนของฮาร์ดแวร์

จากโครงสร้างที่ได้ออกแบบสำหรับการทดลอง หากพิจารณาการทำงานของแต่ละโมดูลคอมพิวเตอร์ในส่วนขององค์ประกอบการทำงานจะพบว่า การทำงานแต่ละโมดูลจะประกอบด้วยการทำงานส่วนที่เป็นฮาร์ดแวร์และซอฟต์แวร์ซึ่งจำเป็นต้องทำงานร่วมกัน หากเกิดความล้มเหลวในส่วนใดส่วนหนึ่ง ทั้งในส่วนที่เป็นฮาร์ดแวร์หรือซอฟต์แวร์ ก็จะทำให้โมดูลคอมพิวเตอร์นั้นเกิดความล้มเหลวในการทำงาน ดังนั้น การหาความน่าเชื่อถือของแต่ละโมดูลคอมพิวเตอร์ เพื่อนำไปหาความน่าเชื่อถือรวมทั้งระบบ จำเป็นต้องพิจารณาในส่วนที่เป็นฮาร์ดแวร์และซอฟต์แวร์ การพิจารณาความน่าเชื่อถือของฮาร์ดแวร์ อัตราการล้มเหลว เป็นตัวแปรหนึ่งซึ่งจำเป็นต้องนำมาพิจารณา

อัตราการล้มเหลว เป็นคุณสมบัติเฉพาะของอุปกรณ์ที่มีความสำคัญ สามารถใช้พิจารณาสำหรับการออกแบบ พยากรณ์ และการประเมินความน่าเชื่อถือของระบบ สำหรับอัตราการล้มเหลวในฮาร์ดแวร์นั้น มีผู้เริ่มให้ความสนใจศึกษาตั้งแต่ปี 1950 [20] เป็นต้นมา ในองค์กรขนาดใหญ่ เช่น Radio Corporation of America, General Electric Company และ Motorola [20] เป็นต้น ต่างก็มีการตีพิมพ์ข้อมูลอัตราการล้มเหลวในผลิตภัณฑ์ของตน ซึ่งเป็นข้อมูลที่ได้จากการทดสอบผลิตภัณฑ์ ซึ่งในเวลาต่อมาได้ถูกรวบรวมเป็นหนังสือคู่มืออัตราการล้มเหลวในอุปกรณ์ (Part Failure Rate Handbooks) และปรากฏเป็นมาตรฐานต่างๆ ได้แก่ MIL-HDBK-217, 217A, 217B, 217C, 217D, 217E และ 217F [20] ซึ่งมาตรฐานเหล่านี้ถูกตีพิมพ์โดย Government Printing Office, Washington, DC [20] นอกจากนี้แล้วยังมีการรวบรวมและตีพิมพ์จากองค์กรอื่น เช่น Failure Rate Data Handbook (FARADA) ซึ่งถูกตีพิมพ์โดยโปรแกรม GIDEP (Government Industrial Data Exchange Program) [20] เป็นต้น

การหาค่าอัตราการล้มเหลวของฮาร์ดแวร์ในงานวิจัยนี้ ประกอบด้วยเครื่องคอมพิวเตอร์และอุปกรณ์ต่างๆ ที่เป็นองค์ประกอบในคอมพิวเตอร์ เช่น หน่วยประมวลผล หน่วยความจำ แผงวงจรต่อประสาน เป็นต้น และยังประกอบด้วยอุปกรณ์อื่นๆ เช่น อุปกรณ์ในระบบเครือข่าย เป็นต้น การหาค่าอัตราความล้มเหลวจะต้องนำค่าอัตราการล้มเหลวของแต่ละอุปกรณ์ย่อยเหล่านี้มาคิด หากเครื่องคอมพิวเตอร์นี้ไม่ได้ออกแบบและสร้างเป็นผลิตภัณฑ์ที่สำเร็จ และมีการทดสอบ

อุปกรณ์ที่เป็นองค์ประกอบร่วมกันทั้งหมดพร้อมกันเพื่อหาความอัตราการล้มเหลว ก็เป็นการยากที่จะระบุค่าอัตราความล้มเหลวของเครื่องคอมพิวเตอร์นั้นได้ ในการทดลองนี้ได้ใช้อุปกรณ์ที่มาจากหลายผู้ผลิต และบางผู้ผลิตไม่ต้องการเปิดเผยค่าอัตราความล้มเหลวและวิธีการทดสอบการหาค่าของผลิตภัณฑ์ ดังนั้น เพื่อเปรียบเทียบการเชื่อมต่อกันของอุปกรณ์ตามโครงสร้างที่ใช้ในการทดลองนี้กับการเชื่อมต่อของอุปกรณ์ตามโครงสร้างในวิธีพื้นฐานทั่วไปซึ่งมีผลต่อความน่าเชื่อถือของระบบ จึงได้ใช้ค่าตามคู่มือที่ได้กำหนดไว้ดังตารางที่ 4.6 [19]

ตารางที่ 4.6 อัตราการล้มเหลวที่ใช้ในการทดลอง

Item	Weight (pounds)	Power consumption (watts)	Failure rate (per 10^6 hours)
Majority Voter	0.425	18	40
Processor	0.875	21	85
Memory	2.500	35	42
Interface	0.425	11	75
Comparator	0.425	11	60
Analog to Digital	0.750	15	35
Digital to Analog	0.750	15	35
Power Supply	11.400	110	20
8 Card chassis	7.500	-	-
12 Card chassis	11.900	-	-
14 Card chassis	15.800	-	-

นอกจากค่าอัตราการล้มเหลวแล้ว ยังมีองค์ประกอบอื่นอีกที่จำเป็นต้องพิจารณา ในการคำนวณหาค่าความน่าเชื่อถือของอุปกรณ์ ได้แก่ อัตราการล้มเหลวประยุกต์ (Applied Failure Rates) ผลภาวะแวดล้อม (Environmental Effects) และ การลดระดับความสามารถ (Derating)

อัตราการล้มเหลวประยุกต์ กล่าวถึง ที่มาและองค์ประกอบการได้มาซึ่ง ค่าอัตราการล้มเหลว ค่าอัตราการล้มเหลวของอุปกรณ์ อาจได้มาจากการทดลอง หรือการใช้งานของอุปกรณ์นั้นจริงๆ ที่มีภาวะสิ่งแวดล้อมต่างๆ กันไป เช่น อุณหภูมิ ความสั่นสะเทือน เป็นต้น สามารถแบ่งชนิดของที่มาอัตราการล้มเหลวได้ 2 ชนิด [13] คือ ค่าอัตราการล้มเหลวทั่วไป (Data-Generic) [13] และค่าอัตราการล้มเหลวประยุกต์ (Data-Applied) [13] สำหรับค่าที่ใช้ในงานวิจัยนี้เป็นค่ามาตรฐานที่ได้จากการทดลอง

ผลสภาวะแวดล้อม เช่น การทำงานภายใต้สภาวะความร้อนสูง การทำงานภายใต้สภาวะแวดล้อมที่มีความสั่นสะเทือนสูง เป็นต้น สิ่งต่างๆ เหล่านี้มีผลต่อความน่าเชื่อถือของอุปกรณ์

ดังนั้น ผลจากสภาวะแวดล้อมจึงจำเป็นต้องนำมาคิดในการพยากรณ์ หรือคำนวณหาค่าความน่าเชื่อถือของอุปกรณ์ด้วย วิธีการเลือกตัวประกอบผลของสภาวะแวดล้อมซึ่งนำมาใช้คำนวณในการหาค่าความน่าเชื่อถือหรือการพยากรณ์นั้น สามารถหาได้จากคู่มือของอุปกรณ์ที่มาจากแหล่งผู้ผลิตนั้น ยกตัวอย่างเช่น Navy Failure Rate Data Handbook (FARADA) [13] เป็นต้น

การลดระดับความสามารถ เป็นตัวแปรที่ใช้คูณสำหรับหาค่าอัตราการล้มเหลวอีกตัวหนึ่ง ซึ่งจำเป็นต้องพิจารณา เนื่องจากรายละเอียดของอัตราการล้มเหลวที่ระบุไว้ในคุณสมบัติอุปกรณ์ เช่น ในอุปกรณ์อิเล็กทรอนิกส์ ที่มีการระบุรายละเอียดการทำงานขั้นพื้นฐาน ได้แก่ แรงดัน กำลังไฟฟ้า หรืออุณหภูมิเป็นต้น แต่ในทางปฏิบัติแล้วการใช้งาน รายละเอียดดังกล่าวอาจจะต่ำกว่าที่ระบุไว้ และนี่เองทำให้การลดระดับความสามารถมีผลต่ออัตราการล้มเหลวลดลง และการเพิ่มความน่าเชื่อถือ การหาค่าการลดระดับความสามารถที่เหมาะสมในการคำนวณ สามารถหาได้จากกราฟคุณสมบัติที่มาจากผู้ผลิต ตัวอย่างเช่น ในตารางของ FARADA [13] ได้เตรียมข้อมูลอัตราการล้มเหลวสำหรับการลดระดับความสามารถในการใช้งานอุปกรณ์ โดยส่วนใหญ่คือข้อมูลที่เป็นข้อกำหนดที่ออกโดยหน่วยงานที่รับผิดชอบ [13]

สำหรับการทดลองในวิทยานิพนธ์นี้ มีการคำนวณหาค่าความน่าเชื่อถือของอุปกรณ์ในการทดลอง โดยมีการกำหนดค่าตัวแปรต่างๆ ดังนี้

1. อัตราการล้มเหลว เป็นอัตราการล้มเหลวทั่วไปของอุปกรณ์ในตารางที่ 4.6
2. ค่าระดับความสามารถของระบบกำหนดให้เท่ากับ 40 % [13]
3. ค่าองค์ประกอบสิ่งแวดล้อม (k) ที่มีผลกับระบบ กำหนดให้เท่ากับ 50 [13]
4. ระยะเวลาการปฏิบัติการที่ต้องการ กำหนดให้เท่ากับ 75% ของระยะเวลาปฏิบัติการ 100 ชั่วโมง (75 Percent of mission) [13]

สามารถคำนวณหาอัตราการล้มเหลวในการข้อมูล ได้ดังตารางที่ 4.7

ตารางที่ 4.7 ข้อมูลการคำนวณหาค่า Failure Rate

Item	Failure Rate (per hours)	Derating Factor	K Factor Failure per hour	Failure Rate Per mission
Majority Voter	0.00004	0.000016	0.0008	0.06
Processor	0.000085	0.000034	0.0017	0.1275
Memory	0.000042	0.0000168	0.00084	0.063
Interface	0.000075	0.00003	0.0015	0.1125
Comparator	0.00005	0.000024	0.0012	0.09
Analog to Digital	0.000035	0.000014	0.0007	0.0525
Digital to Analog	0.000035	0.000014	0.0007	0.0525

การหาค่าความน่าเชื่อถือของโมดูลแต่ละโมดูล สามารถหาได้จากสมการ

$$R = e^{-\lambda t}$$

หากพิจารณาแต่ละโมดูลคอมพิวเตอร์ในส่วนที่เป็นฮาร์ดแวร์ ประกอบด้วยส่วนประกอบหลัก 3 ส่วนคือ หน่วยประมวลผล หน่วยความจำ และหน่วยของแผงวงจรต่อประสาน ส่วนประกอบอื่นที่อยู่ในระบบคอมพิวเตอร์ ผู้วิจัยไม่สามารถนำมาคำนวณหาความน่าเชื่อถือได้ เนื่องจากคอมพิวเตอร์ที่ใช้มีส่วนประกอบมาจากหลายผู้ผลิต และบางผู้ผลิตไม่ยินยอมเปิดเผยค่าอัตราการล้มเหลวและวิธีการทดสอบ ดังนั้นจึงพิจารณาเฉพาะส่วนประกอบหลัก 3 ส่วนที่ได้กล่าวไว้แล้วนั้น ดังนั้น หากส่วนประกอบหลักทั้ง 3 ส่วนใดส่วนหนึ่งเกิดความล้มเหลวแล้วจะทำให้โมดูลคอมพิวเตอร์นั้นเกิดความล้มเหลวด้วย อาจจะกล่าวได้ว่า ส่วนประกอบในโมดูลคอมพิวเตอร์นั้นมีการต่อเชื่อมกันของอุปกรณ์แบบอนุกรม ซึ่งสามารถหาค่าความน่าเชื่อถือของแต่ละองค์ประกอบได้ดังนี้

ความน่าเชื่อถือของหน่วยประมวลผล

$$R = e^{-\lambda t} = e^{-0.1275} = 0.8803$$

ความน่าเชื่อถือของหน่วยความจำ

$$R = e^{-\lambda t} = e^{-0.063} = 0.9389$$

ความน่าเชื่อถือของหน่วยวงจรต่อประสาน

$$R = e^{-\lambda t} = e^{-0.1125} = 0.8936$$

ความน่าเชื่อถือรวมของโมดูลคอมพิวเตอร์ส่วนที่เป็นฮาร์ดแวร์คือ

$$R = 0.8803 \times 0.9389 \times 0.8936 = 0.7383$$

จากค่าความน่าเชื่อถือที่คำนวณได้นั้นไม่สามารถนำมาเป็นค่าความน่าเชื่อถือทั้งหมดของโมดูลคอมพิวเตอร์ได้ เนื่องจากโมดูลคอมพิวเตอร์ประกอบด้วยส่วนที่เป็นซอฟต์แวร์ที่ต้องทำงานร่วมกัน ดังนั้น จึงจำเป็นต้องพิจารณาหาความน่าเชื่อถือในส่วนของซอฟต์แวร์ด้วย

4.4 การหาความน่าเชื่อถือของระบบในส่วนของซอฟต์แวร์

จากรูปแบบการทำงานในการทดลอง หากพิจารณาจะพบว่า สามารถแบ่งการทำงานของระบบออกเป็น 2 ชนิด คือ การทำงานของฮาร์ดแวร์และซอฟต์แวร์ หากขาดส่วนใดส่วนหนึ่งไปจะทำให้ระบบเกิดความล้มเหลวไม่สามารถทำงานตามวัตถุประสงค์ของระบบได้ ดังนั้น นอกจากจะพิจารณาความน่าเชื่อถือในส่วนที่เป็นฮาร์ดแวร์แล้ว ยังจำเป็นต้องพิจารณาความน่าเชื่อถือในส่วนของซอฟต์แวร์ด้วย นิยามของความน่าเชื่อถือในซอฟต์แวร์ (Software Reliability) [23] คือ ความน่าจะเป็นที่จะไม่เกิดความล้มเหลวขณะปฏิบัติการในช่วงเวลาและสภาวะแวดล้อมที่กำหนด สภาพพร้อมใช้งานของซอฟต์แวร์ (Software availability) คือ การที่ซอฟต์แวร์หรือระบบสามารถทำงานตามฟังก์ชันที่ยอมรับได้ ภายในเวลาปฏิบัติการที่กำหนด [23] จากนิยามพบว่า สาเหตุที่ทำให้ซอฟต์แวร์เกิดความล้มเหลว หรือไม่สามารถปฏิบัติการได้ตามวัตถุประสงค์ตามช่วงเวลาและสภาวะแวดล้อมที่กำหนด นั้นคือ การเกิดความผิดพร่องและความล้มเหลวขึ้นในระบบนั่นเอง

4.4.1 ความล้มเหลวและความผิดพร่อง

ความล้มเหลว (Failure) และความผิดพร่อง (Fault) เป็นนิยามที่จำเป็นต้องใช้สำหรับการพิจารณาความน่าเชื่อถือในซอฟต์แวร์ เมื่อกล่าวถึง ความล้มเหลวของซอฟต์แวร์จะหมายถึง การที่ระบบซอฟต์แวร์ไม่สามารถทำงานหรือปฏิบัติการตามความต้องการของระบบได้ [23] ดังเช่น ซอฟต์แวร์ที่เกิดความล้มเหลวในการทำงาน แล้วไม่สามารถแสดงรายงานผลจากกระบวนการได้อย่างถูกต้อง ซึ่งอาจจะต้องทำการเริ่มต้นระบบการทำงานของระบบซอฟต์แวร์นั้นใหม่ เป็นต้น ส่วนความผิดพร่องนั้น อาจกล่าวได้ว่า เป็นข้อบกพร่องที่มีในซอฟต์แวร์ เช่น การเขียนชนิดและเงื่อนไขของตัวแปรที่รับจากกระบวนการผิด เป็นต้น ความผิดพร่องและความล้มเหลวที่เกิดขึ้นกับซอฟต์แวร์นั้น สามารถป้องกันและกำจัดได้โดยใช้หลักการการประกันคุณภาพซอฟต์แวร์ และวิศวกรรมความน่าเชื่อถือซอฟต์แวร์มาประยุกต์ใช้ ซึ่งมีหลักการและทฤษฎีอยู่มากมาย เช่น หลักการวัดความน่าเชื่อถือของซอฟต์แวร์ การทำนายความน่าเชื่อถือและการทดสอบซอฟต์แวร์ เป็นต้น

4.4.2 หลักการพื้นฐานสำหรับการวัดความน่าเชื่อถือของซอฟต์แวร์

หลักการพื้นฐานสำหรับการวัดความน่าเชื่อถือของซอฟต์แวร์อาจแบ่งได้เป็น 2 ชนิด [23] ได้แก่ 1. การวัดอัตราการล้มเหลวของซอฟต์แวร์ในช่วงเวลากระทำการ (Execution time) 2. การวัดความถี่ที่เกิดการล้มเหลวในการทำงาน ตามข้อมูลซึ่งแสดงลักษณะเฉพาะในการปฏิบัติการ (Operational Profile)

อัตราการล้มเหลวของซอฟต์แวร์ในช่วงเวลากระทำการคือ อัตราการล้มเหลวของฟังก์ชันการทำงานของซอฟต์แวร์ในช่วงเวลากระทำงาน ช่วงเวลากระทำการคือ ช่วงเวลาประมวลผลของซอฟต์แวร์หรืออาจเป็นเวลาของหน่วยประมวลผล (CPU-Time) โดยปกติแล้วจะวัดช่วงเวลา

กระทำการอยู่ในรูปของวินาที (CPU-Seconds) หรือชั่วโมง (CPU-Hours) ความสัมพันธ์ระหว่างความล้มเหลวและช่วงเวลากระทำงาน หากพิจารณาสภาวะที่เหมาะสมของซอฟต์แวร์ เมื่อเครื่องคอมพิวเตอร์มีโหลดแต่ไม่มีการประมวลผลทำให้ไม่มีความผิดพลาด ช่วงเวลานี้เรียกว่า ช่วงเวลากระทำการเท่ากันศูนย์ ช่วงเวลาของซีพียูหรือช่วงเวลากระทำการนั้น ไม่สามารถทำการวัดได้โดยตรง ส่วนใหญ่แล้วจะใช้การประมาณค่า หรืออาจใช้แบบจำลองความน่าเชื่อถือในซอฟต์แวร์มาวิเคราะห์การทำงานในช่วงเวลากระทำการ

ข้อมูลแสดงลักษณะเฉพาะในการปฏิบัติการ คือ การพิจารณาที่ความถี่ของการเกิดความล้มเหลวของซอฟต์แวร์ในการทำงาน ข้อมูลเฉพาะในการปฏิบัติการคือ กลุ่มสมาชิกที่เป็นสภาวะของอินพุตที่มีสภาวะร่วมกันในการปฏิบัติงานตามปกติ

อย่างไรก็ตาม นอกจากการวัดความผิดพลาดอันเกิดจากการใช้งานจริง หรือการวัดเนื่องจากการทดสอบแล้ว วิธีการใช้แบบจำลองความน่าเชื่อถือซอฟต์แวร์ก็เป็นอีกวิธีหนึ่ง ที่สามารถอธิบายในส่วนของพฤติกรรมความเชื่อถือของซอฟต์แวร์ได้

4.4.3 แบบจำลองความน่าเชื่อถือของซอฟต์แวร์

แบบจำลองที่ใช้ในการประมาณค่าและวิเคราะห์ความน่าเชื่อถือของซอฟต์แวร์นั้นมีหลายชนิด แต่หากแบ่งตามองค์ประกอบแล้วสามารถแบ่งได้ 2 องค์ประกอบ [23] นั้นคือ องค์ประกอบที่ทำงานตามเวลาปฏิบัติการ (Execution Time Component) และองค์ประกอบที่ทำงานตามเวลาปฏิทิน (Calendar Time Component)

องค์ประกอบตามเวลาปฏิบัติการ เป็นองค์ประกอบที่มีความเกี่ยวข้องกับการเกิดความล้มเหลวในช่วงเวลาการปฏิบัติการ ความน่าเชื่อถือของซอฟต์แวร์จะขึ้นอยู่กับ คุณสมบัติที่ใช้ในการออกแบบและพัฒนา เช่น คุณสมบัติในการทำงานภายใต้สภาวะแวดล้อมที่ได้ออกแบบ โดยองค์ประกอบตามเวลาปฏิบัติการ สามารถแบ่งชนิดออกไปอีกได้ 2 ชนิด [23] นั้นคือ แบบจำลองความน่าเชื่อถือที่มีการเติบโต (Reliability Growth Models) และแบบจำลองความน่าเชื่อถือคงที่ (Constant Reliability Models) แบบจำลองความน่าเชื่อถือที่มีการเติบโต คือ รายละเอียดความสามารถในการทำงานตลอดระยะเวลาการทำงาน หรือการทดสอบและสามารถลดอัตราการล้มเหลวได้ ส่วนแบบจำลองความน่าเชื่อถือคงที่ คือ ภาวะที่เหมาะสมของหลังจากนำซอฟต์แวร์นั้นไปใช้งานจริง ทำให้ไม่สามารถลดอัตราการล้มเหลวได้

องค์ประกอบที่ทำงานตามเวลาปฏิทิน สิ่งซึ่งความน่าเชื่อถือมีความเกี่ยวข้องกับเวลาที่ล่วงไปตามปฏิทิน ทั้งเวลาปฏิบัติการและจำนวนครั้งของการเกิดความล้มเหลว การล่วงของเวลาตามปฏิทินจะขึ้นอยู่กับการใช้งานทรัพยากรจริง เช่น จำนวนรอบของการประมวลผลบนระบบคอมพิวเตอร์

แบบจำลองความน่าเชื่อถือของซอฟต์แวร์นั้นมีหลายแบบจำลอง ซึ่งจำเป็นต้องเลือกใช้ให้เหมาะสมกับซอฟต์แวร์ที่ได้พัฒนาขึ้น เช่น แบบจำลองของ Jelinski-Moranda [24],

Littlewood-Verrall [24] , Musa Basic Model [24] , Nonhomogeneous Poisson Process (NHPP) [24] เป็นต้น

แบบจำลองของ Jelinski-Moranda

แบบจำลองของ Jelinski-Moranda เป็นแบบจำลองความน่าเชื่อถือซอฟต์แวร์ที่ใช้เมื่อไม่ทราบค่าความล้มเหลวเริ่มต้น แต่กำหนดให้อัตราการล้มเหลวมีค่าคงที่ และเมื่อระบบตรวจพบความล้มเหลวแล้ว ระบบจะทำการแก้ไขข้อผิดพร่องทันทีจะไม่เกิดข้อผิดพลาดซ้ำขึ้นมาอีก เวลาในการล้มเหลวของแบบจำลองนั้น สามารถเกิดขึ้นได้โดยอิสระและมีการกระจายแบบเลขชี้กำลัง (Exponentially Distributed) สามารถอธิบายพฤติกรรมได้จากสมการที่ (4.1) [24]

$$Z(\tau) = K\left(\frac{E_T}{I_T}\varepsilon_c t\right) \tag{4.1}$$

โดยที่ $Z_{(\tau)}$ คือ อัตราการล้มเหลวที่เวลา τ

K คือ ส่วนคงที่

E_T คือ จำนวนความคลาดเคลื่อนขณะเริ่มต้นโปรแกรม

IT คือ ส่วนประกอบของโครงสร้างในโปรแกรม

ε_c คือ จำนวนความล้มเหลวสะสมในช่วงเวลา $[0,\tau]$

แบบจำลองของ Littlewood-Verrall

แบบจำลองของ Littlewood-Verrall เป็นแบบจำลองที่มีระยะเวลาของการเกิดความล้มเหลวเกิดขึ้นได้โดยอิสระ และมีการกระจายแบบเลขชี้กำลัง สามารถอธิบายพฤติกรรมได้จากสมการที่ (4.2) [24]

$$Z(t_i) = \frac{\alpha}{t_i + \beta_0 + \beta_1 i^2} \tag{4.2}$$

โดยที่ $Z_{(\tau)}$ คือ อัตราการล้มเหลวที่เวลา t_i

α, β_0, β_1 คือ เป็นพารามิเตอร์สำหรับโมดูล

i คือ จำนวนการล้มเหลวที่เกิดขึ้น

t_i คือ เวลาระหว่างการล้มเหลวครั้งที่ $i-1$ และ i

แบบจำลอง Musa Basic Model

แบบจำลองแบบ Musa Basic Model เป็นแบบจำลองที่มีเวลากระทำการระหว่างเกิดความล้มเหลว ซึ่งสามารถแบ่งเป็นช่วงๆ โดยแต่ละช่วงมีการกระจายแบบเลขชี้กำลัง ช่วงเวลาของการเกิดความล้มเหลวเกิดขึ้นโดยอิสระภายใต้การกระจายแบบปัวส์ซอง (Poisson distribution) และอัตราการล้มเหลวเป็นสัดส่วนโดยตรงกับความล้มเหลวที่ตรวจพบ สามารถอธิบายความสัมพันธ์ของตัวแปรในแบบจำลองของมูซาได้จากสมการที่ (4.3) [24]

$$\mu(t) = \upsilon_0\left(1 - \exp\left(-\frac{\lambda_0}{\upsilon_0}t\right)\right) \tag{4.3}$$

โดยที่ $\mu(t)$ คือ จำนวนค่าเฉลี่ยในช่วงเวลาที่พิจารณา t

λ_0 คือ ความหนาแน่นของการล้มเหลวในช่วงเวลาเริ่มต้น

υ_0 คือ ค่าการล้มเหลวทั้งหมดที่ได้จากการประมาณพฤติกรรมซอฟต์แวร์ในช่วงเวลาปฏิบัติการ

แบบจำลอง Nonhomogeneous Poisson Process (NHPP)

แบบจำลอง NHPP เป็นแบบจำลองที่มีความล้มเหลวสะสมมีการกระจายแบบปัวส์ซอง สามารถเกิดความล้มเหลวขึ้นในระบบอย่างอิสระและสามารถอธิบายความสัมพันธ์ได้ดังสมการที่ (4.4) [24]

$$\mu(t) = a(1 - \exp(-bt)) \tag{4.4}$$

โดยที่ $\mu(t)$ คือ ค่าเฉลี่ยจำนวนความล้มเหลวที่เกิดขึ้นที่เวลา t

a คือ ค่าที่ได้จากการประมาณจำนวนการล้มเหลวที่เกิดขึ้นอาจได้จากการสังเกตพฤติกรรมในช่วงเวลาปฏิบัติการ

b คือ ค่าความหนาแน่นของการลดระดับสำหรับพารามิเตอร์

แบบจำลองสำหรับหาความน่าเชื่อถือของซอฟต์แวร์ซึ่งมีหลายแบบจำลอง หากจำแนกกลุ่มของแบบจำลองความน่าเชื่อถือแล้ว สามารถออกได้ 4 กลุ่ม [11] ด้วยกัน ดังในตารางที่ 4.8

ตารางที่ 4.8 การจำแนกกลุ่มของแบบจำลองความน่าเชื่อถือสำหรับซอฟต์แวร์

No.	Classification	Description
I	Fault seeding	This incorporate those models that determine the number of faults in the program at zero time via seeding of extraneous fault.
II	Failure count	This includes models counting the number of failure/fault occurring in given time intervals.
III	Times between failure	This incorporates models providing the time between failure estimations.
IV	Input domain based	This incorporate mode that determine the program / software reliability under the circumstance the test cases are sampled randomly form a known operational distribution of inputs to the program/software

จากตารางที่ 4.8 การจำแนกกลุ่มของแบบจำลองความน่าเชื่อถือ มีรายละเอียดดังนี้

กลุ่มที่ 1 (Fault Seeding) เป็นกลุ่มพื้นฐานและการทำงานอินพุตให้เกิดความล้มเหลว เพื่อดูพฤติกรรมของระบบ การป้อนอินพุตดังกล่าว อาจเป็นการกระจายอิสระในโปรแกรมที่กำลังพิจารณาก็ได้ เช่น แบบจำลอง Mills Seeding Model [11]

กลุ่มที่ 2 (Failure Count) เป็นกลุ่มที่ใช้การนับจำนวนการเกิดความล้มเหลวขึ้น ภายในช่วงเวลาที่กำหนด ตัวอย่างเช่น Musa Model [11] และ Shooman Model [11]

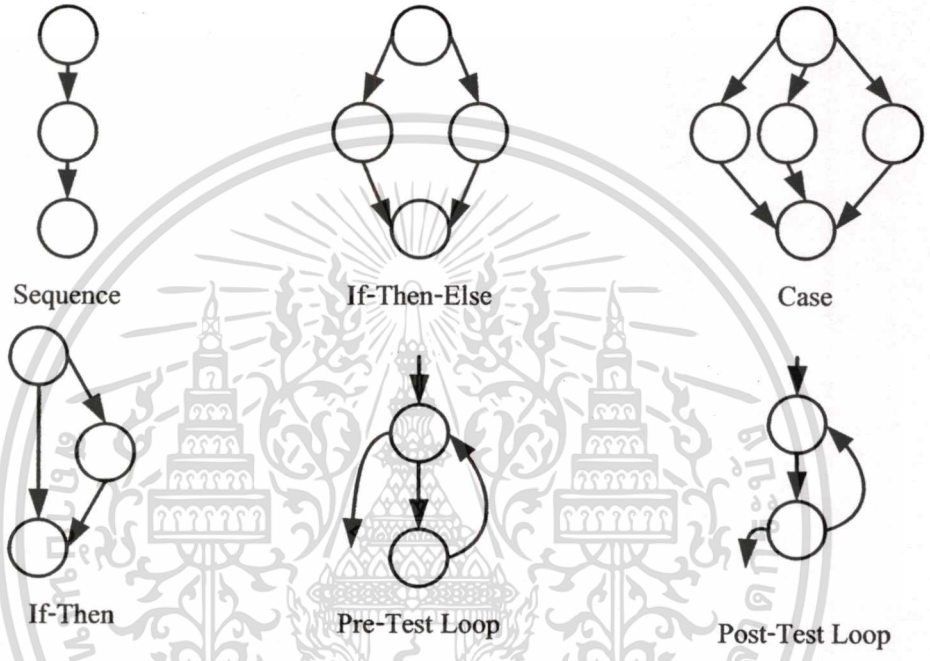
กลุ่มที่ 3 (Time between Failure) เป็นกลุ่มที่มีความผิดพร่องที่ฝังตัวอยู่ในซอฟต์แวร์ จากช่วงเวลาหนึ่งถึงอีกเวลาหนึ่ง ซึ่งได้จากการประมาณค่า ตัวอย่างเช่น Jelinski and Moronda Model [11] และ Schick and Wolverton model [11]

กลุ่มที่ 4 (Input Domain based) กลุ่มนี้มีสมมติฐาน 3 อย่างคือ 1. การเลือกอินพุตเป็นไปอย่างอิสระ 2. อินพุตโดเมนสามารถแบ่งเป็นกลุ่มๆ ในการทดลองได้ 3. ทราบรายละเอียดข้อมูลเกี่ยวกับอินพุต

นอกจากทราบแบบจำลองคณิตศาสตร์ที่ใช้หาค่าความน่าเชื่อถือของซอฟต์แวร์แล้ว ยังต้องทราบวิธีการทดสอบซอฟต์แวร์ ซึ่งมีอยู่หลายวิธี เช่น วิธีทดสอบโครงสร้างของซอฟต์แวร์ด้วยวิธีกราฟ เป็นต้น

4.4.4 วิธีทดสอบซอฟต์แวร์ด้วยวิธีกราฟ

การทดสอบทางโครงสร้างสำหรับซอฟต์แวร์ที่ถูกพัฒนาขึ้น สามารถทดสอบได้หลายรูปแบบ เช่น ทดสอบด้วยวิธีกราฟ (Graphs for Testing) [25] การทดสอบด้วยวิธีทัศนมิติ (Perspective Testing) เป็นต้น สามารถเขียนโครงสร้างของซอฟต์แวร์ตามเงื่อนไขโดยทั่วไปด้วยวิธีกราฟแสดงดังรูปที่ 4.3 โดยวิธีการทดสอบโครงสร้างซอฟต์แวร์ที่ออกแบบในงานวิจัย จะกล่าวในหัวข้อต่อไป



รูปที่ 4.3 การสร้างกราฟโครงสร้างของซอฟต์แวร์ที่ออกแบบโดยทั่วไป

4.4.5 การหาค่าความน่าเชื่อถือซอฟต์แวร์ในการทดลอง

สำหรับวิทยานิพนธ์ฉบับนี้ ได้เลือกใช้แบบจำลองของมูซา (Musa Model) เนื่องจากคุณสมบัติของแบบจำลองตรงตามคุณสมบัติและเงื่อนไขที่ทำการทดลอง โดยเป็นการหาค่าความน่าเชื่อถือของระบบในโดเมนของเวลาที่เกิดขึ้นจริงในเวลากระทำการ สมมุติฐานของการทดลองในงานวิจัยนี้มีดังนี้

1. เวลากระทำการระหว่างเกิดความล้มเหลวสามารถแบ่งเป็นช่วงๆ และมีการกระจายแบบเลขชี้กำลัง (Exponentially Distributed)
2. ช่วงของการล้มเหลวเกิดขึ้นโดยอิสระภายใต้การกระจายแบบปัวส์ซอง (Poisson Distribution)
3. อัตราการล้มเหลวเป็นสัดส่วนโดยตรงกับความล้มเหลวที่ตรวจพบ

จากความสัมพันธ์ที่แสดงในสมการที่ (4.3) สามารถทำการอธิบายแบบจำลองพื้นฐานความน่าเชื่อถือซอฟต์แวร์ของมูซา (Musa Basic Model) ซึ่งมูซาเป็นนำเสนอในปี 1975 [27] ดังนี้

แบบจำลองสมมุติให้เวลาที่จะเกิดการล้มเหลวจริงในระบบ มีการกระจายแบบเลขชี้กำลังดังในสมการที่ 4.5 [27]

$$f_a(\tau) = \phi \exp(-\phi\mu\tau) \tag{4.5}$$

จากสมการที่ (4.5) กำหนดให้ ϕ · คือ ค่าอัตราการล้มเหลวลงที่ $(z_a(\tau) = \phi)$ และมูซาได้อธิบายความสัมพันธ์ของอัตราการล้มเหลวดังสมการที่ (4.6)[27]

$$\phi = fK \tag{4.6}$$

โดยที่ f คือ ความถี่การปฏิบัติการของโปรแกรมเชิงเส้น

K คือ อัตราการเกิดความล้มเหลว

การคาดหมายค่าความล้มเหลวที่เวลา τ สามารถหาได้จากสมการที่ (4.7)[27]

$$\mu(\tau) = \nu_0[1 - \exp(-\phi B\upsilon)] \tag{4.7}$$

โดยที่ B คือ ตัวประกอบการลดการเกิดความล้มเหลว

สามารถหาค่าความหนาแน่นของการเกิดความล้มเหลวได้จากสมการที่ (4.8)[27]

$$\lambda(\tau) = \nu_0\phi B \exp(-\phi B\tau) \tag{4.8}$$

โดยที่ ν_0 คือ จำนวนความล้มเหลวที่เกิดขึ้นทั้งหมดมีค่าเท่ากับ $\frac{\omega_0}{B}$

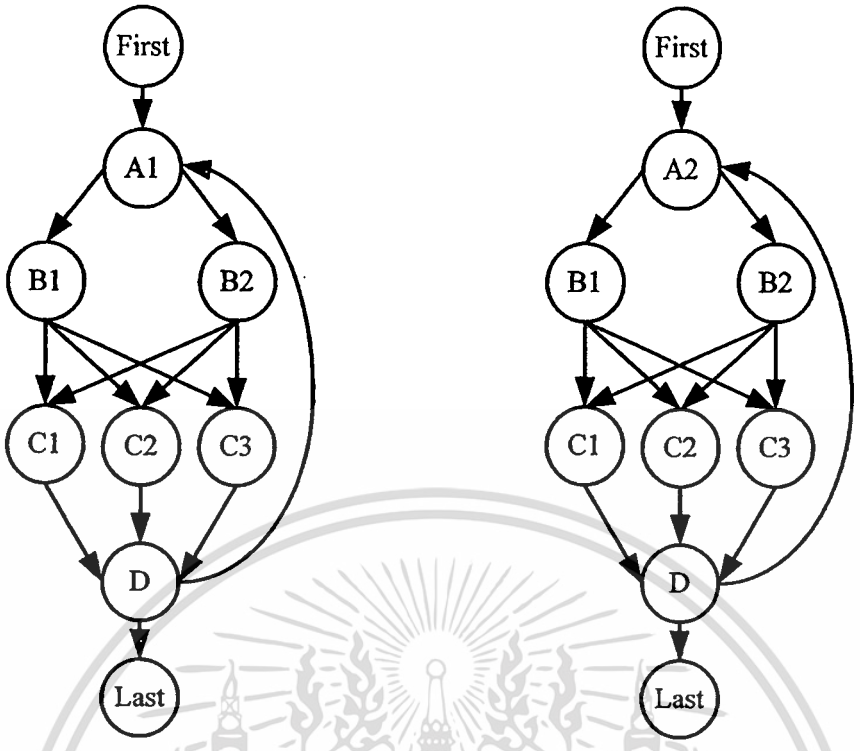
การทดลองในงานวิจัยนี้ สามารถเขียนเส้นทางการทำงานของส่วนต่างๆ ในระบบทั้งส่วนที่เป็นสถานีเอชเอ็มไอ และสถานีควบคุมได้ดังรูปที่ 4.4

กำหนดให้ A1 และ A2 คือ สถานีเอชเอ็มไอที่ 1 และ 2

B1 และ B2 คือ ระบบเครือข่ายที่ 1 และ 2

C1, C2 และ C3 คือ สถานีควบคุมที่ 1,2 และ 3

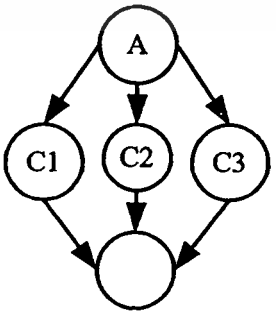
D คือ ชุดจำลองกระบวนการ



ก. กราฟการทำงานของสถานีเอชเอ็มไอที่ 1 ข. กราฟการทำงานของสถานีเอชเอ็มไอที่ 2

รูปที่ 4.4 กราฟการทำงานของสถานีเอชเอ็มไอ

การใช้ซอฟต์แวร์เอชเอ็มไอที่เป็นมาตรฐาน เพื่อควบคุมตัวแปรที่จะมีผลกระทบในการทดลอง ดังที่กล่าวไว้แล้ว ทำให้ผู้วิจัยไม่สามารถทราบถึงอัตราการล้มเหลวและวิธีการทดสอบของผู้ผลิตซอฟต์แวร์ จึงไม่สามารถนำอัตราการล้มเหลวของซอฟต์แวร์เอชเอ็มดังกล่าวมาใช้คำนวณได้ อย่างไรก็ตามในการใช้ซอฟต์แวร์มาตรฐานนี้ ผู้วิจัยต้องเขียนคำสั่งให้ซอฟต์แวร์สามารถทำงานได้ตามเงื่อนไข ผู้วิจัยจึงนำคำสั่งเงื่อนไขที่เขียนบนซอฟต์แวร์เอชเอ็มไอมาตรฐานมาวิเคราะห์หาค่าอัตราการล้มเหลว โดยการจำลองให้เกิดความผิดพลาดที่ช่วงเวลาการทำงานของซีพียู 1000000 ครั้ง ดังแสดงความล้มเหลวสะสมในตารางที่ 4.9 และสามารถแสดงการทำงานแบบแยกส่วนของสถานีเอชเอ็มไอดังรูป 4.5



ก. กราฟการทำงานของสถานีเอชเอ็มไอแบบแยกส่วน

```
if (สภาวะ C1 ที่จะเกิดการล้มเหลว)
   คำสั่ง X1;
else if (สภาวะ C2 ที่จะเกิดการล้มเหลว)
   คำสั่ง X2;
else
   คำสั่ง X3;
```

ข. ตัวอย่างเงื่อนไขการทำงานของสถานีที่เอชเอ็มไอ

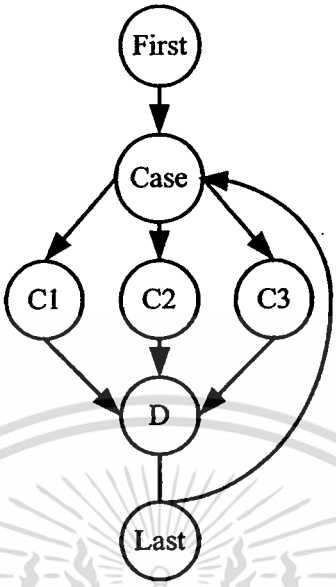
รูปที่ 4.5 การทำงานแบบแยกส่วนของเอชเอ็มไอ

ตารางที่ 4.9 จำนวนความล้มเหลวสะสมที่เกิดขึ้นในช่วงเวลากระทำการเก็บข้อมูลทุกๆ 10000 รอบการทำงานของซีพียูทั้งหมด 100 ครั้งของสถานีเอชเอ็มไอ

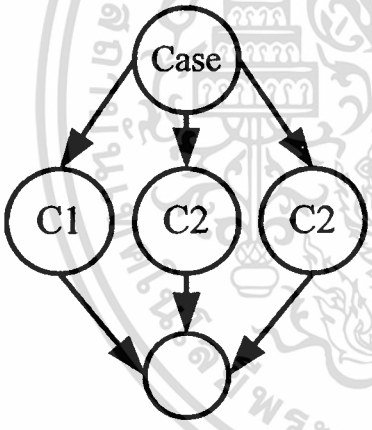
ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน
1	1143	26	28861	51	56758	76	84570
2	2250	27	29990	52	57858	77	85689
3	3413	28	31093	53	58953	78	86829
4	4551	29	32183	54	60031	79	87944
5	5680	30	33237	55	61158	80	89083
6	6798	31	34363	56	62298	81	90164
7	7886	32	35539	57	63427	82	91288
8	8991	33	36673	58	64592	83	92390
9	10125	34	37781	59	65735	84	93552
10	11253	35	38887	60	66889	85	94640
11	12400	36	39982	61	67978	86	95693
12	13522	37	41130	62	69141	87	96818
13	14592	38	42228	63	70224	88	97938
14	15758	39	43350	64	71301	89	99054
15	16840	40	44456	65	72402	90	100157
16	17967	41	45594	66	73521	91	101295
17	19047	42	46701	67	74600	92	102391
18	20141	43	47832	68	75714	93	103497
19	21249	44	48876	69	76787	94	104622
20	22361	45	50024	70	77937	95	105699
21	23465	46	51145	71	79026	96	106826
22	24595	47	52277	72	80157	97	107955
23	25681	48	53351	73	81259	98	109075
24	26729	49	54520	74	82389	99	110233
25	27812	50	55661	75	83493	100	111318

จากการทำงานแบบแยกส่วนในรูปที่ 4.5 และตารางที่ 4.9 สามารถพิจารณาหาค่าอัตราของการล้มเหลวได้คือ 0.11132 และมีความน่าเชื่อถือคือ 0.89465

การทำงานของสถานีควบคุมสามารถเขียนกราฟการทำงานได้ดังรูปที่ 4.6



รูปที่ 4.6 กราฟการทำงานของสถานีควบคุม



ก. กราฟการทำงานของสถานีควบคุม

```
switch (control)
{
   Case C3:คำสั่ง C3; break;
// คำสั่ง  C3 เป็นตัวหลัก
   Case C2:  คำสั่ง C2 ; break;
// คำสั่ง  C2 เป็นตัวหลัก
   Case C1:คำสั่ง C1;break; //คำสั่ง  C1 เป็นตัวหลัก
}
```

ข. ตัวอย่างเงื่อนไขการทำงานของสถานีควบคุม

รูปที่ 4.7 การทำงานตามโครงสร้างของโปรแกรมสถานีควบคุม

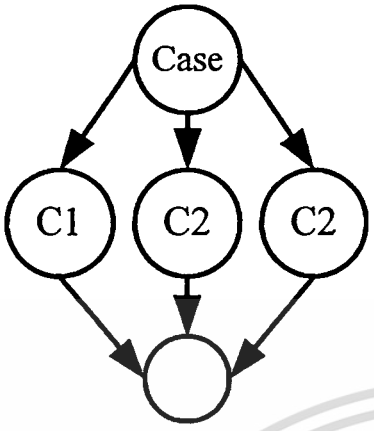
ตารางที่ 4.10 จำนวนความล้มเหลวสะสมที่เกิดขึ้นในช่วงเวลากระทำการเก็บข้อมูลทุกๆ 10000 รอบการทำงานของซีพียูทั้งหมด 100 ครั้งของสถานีควบคุม

ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน
1	1071	26	28829	51	56467	76	84103
2	2174	27	29968	52	57523	77	85206
3	3276	28	31091	53	58633	78	86326
4	4404	29	32136	54	59810	79	87414
5	5534	30	33269	55	60926	80	88579

ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน
6	6624	31	34368	56	62055	81	89692
7	7750	32	35453	57	63182	82	90786
8	8869	33	36591	58	64230	83	91885
9	9971	34	37683	59	65379	84	93019
10	11058	35	38755	60	66454	85	94109
11	12142	36	39883	61	67502	86	95287
12	13269	37	40985	62	68584	87	96427
13	14401	38	42047	63	69717	88	97541
14	15499	39	43126	64	70853	89	98654
15	16674	40	44245	65	72014	90	99775
16	17778	41	45395	66	73060	91	100876
17	18920	42	46477	67	74191	92	101979
18	20003	43	47634	68	75257	93	103071
19	21086	44	48788	69	76387	94	104197
20	22210	45	49924	70	77508	95	105324
21	23300	46	50979	71	78559	96	106415
22	24409	47	52080	72	79673	97	107429
23	25516	48	53189	73	80743	98	108556
24	26629	49	54295	74	81867	99	109651
25	27721	50	55391	75	82985	100	110760

จากตารางที่ 4.10 อัตราการล้มเหลวของสถานีควบคุมได้คือ 0.11076 และมีความน่าเชื่อถือคือ 0.89515

การทำงานของซอฟต์แวร์โหวตคะแนนสามารถเขียนกราฟการทำงานได้ดังรูปที่ 4.8



ก. กราฟการทำงานของซอฟต์แวร์โหวตคะแนน

```
switch (control)
{
    Case C3:คำสั่ง C3; break;
// คำสั่ง  C3 เป็นตัวหลัก
    Case C2:  คำสั่ง C2 ; break;
//  คำสั่ง  C2 เป็นตัวหลัก
  Case C1:คำสั่ง C1;break; //คำสั่ง  C1 เป็นตัวหลัก
}
```

ข. ตัวอย่างเงื่อนไขการโหวตทำงานโดยซอฟต์แวร์

รูปที่ 4.8 การทำงานตามโครงสร้างของซอฟต์แวร์โหวตคะแนน

ตารางที่ 4.11 จำนวนความล้มเหลวสะสมที่เกิดขึ้นในช่วงเวลากระทำการเก็บข้อมูลทุกๆ 10000 รอบการทำงานของซีพียูทั้งหมด 100 ครั้งของโปรแกรมโหวตคะแนน

ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน
1	1082	26	28953	51	56540	76	84144
2	2180	27	30019	52	57655	77	85346
3	3321	28	31130	53	58737	78	86430
4	4458	29	32280	54	59879	79	87565
5	5567	30	33343	55	60969	80	88665
6	6632	31	34431	56	62067	81	89745
7	7740	32	35580	57	63215	82	90857
8	8872	33	36688	58	64315	83	91961
9	9992	34	37806	59	65473	84	93077
10	11041	35	38828	60	66556	85	94230
11	12119	36	39943	61	67624	86	95346
12	13231	37	41008	62	68687	87	96497
13	14340	38	42151	63	69817	88	97597
14	15435	39	43258	64	70924	89	98705
15	16566	40	44368	65	72035	90	99806
16	17684	41	45434	66	73145	91	100924
17	18788	42	46547	67	74315	92	101983

ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน	ครั้งที่	จำนวน
18	19893	43	47648	68	75402	93	103070
19	21050	44	48747	69	76478	94	104188
20	22164	45	49780	70	77565	95	105272
21	23285	46	50904	71	78664	96	106420
22	24391	47	52044	72	79796	97	107516
23	25532	48	53166	73	80877	98	108594
24	26643	49	54298	74	81927	99	109632
25	27783	50	55377	75	83013	100	110701

จากตารางที่ 4.11 อัตราการล้มเหลวของซอฟต์แวร์การโหวตคะแนนได้คือ 0.11070 และมีความน่าเชื่อถือคือ 0.89521

เมื่อได้ความน่าเชื่อถือของโมดูลในแต่ละส่วนแล้ว สามารถวิเคราะห์หาค่าความน่าเชื่อถือในโครงสร้างของระบบที่ได้ออกแบบ โดยการนำความน่าเชื่อถือทั้งซอฟต์แวร์และฮาร์ดแวร์ไปคิดร่วมกัน

4.5 การวิเคราะห์ความน่าเชื่อถือในโครงสร้างระบบที่ออกแบบ

จากรูปที่ 4.1 กำหนดให้ A และ B คือ สถานีเอชเอ็มไอ C และ D คือ ระบบเครือข่ายอีเทอร์เน็ต 2 ลูป E, F และ G คือ สถานีควบคุมทั้ง 3 สถานีที่ใช้ควบคุมและรับค่าสัญญาณมาจากกระบวนการ

หากแบ่งตามหน้าที่การทำงาน (Functional) สามารถได้ 3 กลุ่มคือ กลุ่ม X ซึ่งประกอบไปด้วยโมดูลเอชเอ็มไอ A และ B กลุ่ม Y ซึ่งประกอบไปด้วยโมดูลของระบบเครือข่ายอีเทอร์เน็ต C และ D กลุ่ม Z ซึ่งประกอบไปด้วยโมดูลของเครื่องควบคุม E, F และ G ซึ่งทั้ง 3 กลุ่ม สามารถหาความวิเคราะห์หาความน่าเชื่อถือได้ดังนี้

4.5.1 การหาความน่าเชื่อถือแบบมีเงื่อนไขสำหรับสถานีเอชเอ็มไอ (กลุ่ม X)

เมื่อพิจารณากลุ่มของสถานีเอชเอ็มไอจะพบว่า ระบบเอชเอ็มไอจะเกิดความล้มเหลวได้เมื่อสถานีเอชเอ็มไอทั้งสถานี A และ B เกิดความล้มเหลวทั้งคู่ และมีเงื่อนไขเมื่อเกิดความล้มเหลวในระบบคือมีค่าอัตราความล้มเหลวแต่ละโมดูลคงที่และเท่ากัน จากการเชื่อมต่อกันแบบขนานของโมดูลเอชเอ็มไอ หากกำหนดให้ $R = e^{-\lambda t}$ โดยกำหนดให้ λ_i อัตราส่วนความล้มเหลวคงที่ และ t_i คือ ช่วงเวลาที่พิจารณาซึ่งเป็นช่วงเวลาที่ต้องการให้อุปกรณ์หรือระบบสามารถทำงานไม่เกิดความล้มเหลว เมื่อแทนเลขชี้กำลังด้วย F_i จะทำให้ได้สมการคือ $R_i = e^{-Fi}$ สามารถหาค่าความไม่น่าเชื่อถือของระบบได้จากสมการ

$$Q_X = Q_A \times Q_B \tag{4.9}$$

เมื่อ Q_X คือ ความไม่น่าเชื่อถือรวมของระบบเอชเอ็มไอ

Q_A คือ ความไม่น่าเชื่อถือของเอชเอ็มไอโมดูล A

Q_B คือ ความไม่น่าเชื่อถือของเอชเอ็มไอโมดูล B

หากแทนค่า $Q_A = 1 - R_A = 1 - e^{-F_A}$ และ $Q_B = 1 - R_B = 1 - e^{-F_B}$ จะทำให้ได้สมการความไม่น่าเชื่อถือของระบบ

$$Q_X = \left(1 - e^{-F_A}\right) \times \left(1 - e^{-F_B}\right) = 1 - e^{-F_A} - e^{-F_B} + \left(e^{-F_A} \times e^{-F_B}\right) \tag{4.10}$$

ความน่าเชื่อถือของระบบเอชเอ็มไอสามารถหาได้จากสมการ

$$R_X = e^{-F_A} + e^{-F_B} - e^{-(F_A + F_B)} \tag{4.11}$$

$$R_X = R_A + R_B \times R_A R_B \tag{4.12}$$

$$R_X = 1 - \left(Q_A \times Q_B\right) \tag{4.13}$$

เมื่อ R_X คือ ความไม่น่าเชื่อถือรวมของระบบเอชเอ็มไอ

R_A คือ ความไม่น่าเชื่อถือของเอชเอ็มไอโมดูล A

R_B คือ ความไม่น่าเชื่อถือของเอชเอ็มไอโมดูล B

4.5.2 การหาความน่าเชื่อถือแบบมีเงื่อนไขสำหรับระบบเครือข่ายอีเทอร์เน็ต (กลุ่ม Y)

เมื่อพิจารณากลุ่มของระบบเครือข่ายอีเทอร์เน็ตจะพบว่ามีรูปแบบการต่อเหมือนกับสถานีเอชเอ็มไอ จะพบว่าระบบเครือข่ายจะเกิดความล้มเหลวได้เมื่อระบบเครือข่ายทั้งสองลูปคือ C และ D เกิดความล้มเหลวทั้งคู่ และมีเงื่อนไขเมื่อเกิดความล้มเหลวในระบบคือมีค่าอัตราความล้มเหลวแต่ละโมดูลคงที่และเท่ากัน จากการเชื่อมต่อกันแบบขนานของของระบบเครือข่ายอีเทอร์เน็ต หากกำหนดให้ $R = e^{-\lambda t}$ โดยกำหนดให้ λ_i อัตราส่วนความล้มเหลวคงที่ และ t_i คือ ช่วงเวลาที่พิจารณาซึ่งเป็นช่วงเวลาที่ต้องการให้อุปกรณ์หรือระบบสามารถทำงานไม่เกิดความล้มเหลว เมื่อแทนเลขชี้กำลังด้วย F_i จะทำให้ได้สมการคือ $R_i = e^{-Fi}$ สามารถหาค่าความไม่น่าเชื่อถือของระบบได้จากสมการ

$$Q_Y = Q_C \times Q_D \tag{4.14}$$

เมื่อ Q_Y คือ ความไม่น่าเชื่อถือรวมของระบบเครือข่ายอีเทอร์เน็ต

Q_C คือ ความไม่น่าเชื่อถือของระบบเครือข่ายอีเทอร์เน็ตลูป C

Q_D คือ ความไม่น่าเชื่อถือของระบบเครือข่ายอีเทอร์เน็ตลูป D

หากแทนค่า $Q_C = 1 - R_C = 1 - e^{-F_C}$ และ $Q_D = 1 - R_D = 1 - e^{-F_D}$ จะทำให้ได้สมการความไม่น่าเชื่อถือของระบบ

$$Q_Y = \left(1 - e^{-F_C}\right) \times \left(1 - e^{-F_D}\right) = 1 - e^{-F_C} - e^{-F_D} + \left(e^{-F_C} \times e^{-F_D}\right) \quad (4.15)$$

ความน่าเชื่อถือของระบบเครือข่ายอีเทอร์เน็ตสามารถหาได้จากสมการ

$$R_Y = e^{-F_C} + e^{-F_D} - e^{-(F_C + F_D)} \quad (4.16)$$

$$R_Y = R_C + R_D \times R_C R_D \quad (4.17)$$

$$R_Y = 1 - \left(Q_C \times Q_D\right) \quad (4.18)$$

เมื่อ R_Y คือ ความน่าเชื่อถือรวมของระบบเครือข่ายอีเทอร์เน็ต

R_C คือ ความน่าเชื่อถือของระบบเครือข่ายอีเทอร์เน็ตลูป C

R_D คือ ความน่าเชื่อถือของระบบเครือข่ายอีเทอร์เน็ตลูป D

4.5.3 การหาความน่าเชื่อถือแบบมีเงื่อนไขสำหรับสถานีควบคุม (กลุ่ม Z)

เมื่อพิจารณากลุ่มของสถานีควบคุมหรือกลุ่ม Z ซึ่งประกอบด้วยโมดูลของสถานีควบคุมจำนวน 3 โมดูลเชื่อมต่อกันอยู่ การทำงานแบบมีเงื่อนไขของระบบที่มีโมดูลซ้ำสำรองจำนวน 3 โมดูล อาจแยกการทำงานแบบมีเงื่อนไขได้คือ กรณีที่ต้องมีอย่างน้อย 1 ใน 3 โมดูลทำงานสำเร็จระบบจึงจะไม่ล้มเหลวในการทำงาน และในกรณีที่ต้องมีอย่างน้อย 2 ใน 3 โมดูลทำงานสำเร็จระบบจึงจะไม่เกิดความล้มเหลวในการทำงาน อาจแยกการพิจารณาได้ดังนี้

กรณีที่ต้องมีอย่างน้อย 1 ใน 3 โมดูลต้องทำงานสำเร็จ

จากการเชื่อมต่อกันของโมดูลสถานีควบคุม หากกำหนดให้ $R = e^{-\lambda t}$ โดยกำหนดให้ λ_i อัตราส่วนความล้มเหลวคงที่ และ t_i คือ ช่วงเวลาที่พิจารณาซึ่งเป็นช่วงเวลาที่ต้องการให้อุปกรณ์หรือระบบสามารถทำงานไม่เกิดความล้มเหลว เมื่อแทนเลขชี้กำลังด้วย F_i จะทำให้ได้สมการคือ $R_i = e^{-Fi}$ สามารถหาค่าความไม่น่าเชื่อถือของระบบได้จากสมการ

$$Q_Z = \left(1 - e^{-F_E}\right) \times \left(1 - e^{-F_F}\right) \times \left(1 - e^{-F_G}\right) \quad (4.19)$$

และสามารถหาความน่าเชื่อถือของสถานีควบคุมได้จากสมการ

$$R_Z = e^{-F_E} + e^{-F_F} + e^{-F_G} - e^{-(F_E + F_F)} - e^{-(F_E + F_G)} - e^{-(F_F + F_G)} - e^{-(F_E + F_F + F_G)} \quad (4.20)$$

$$= R_E + R_F + R_G - \left(R_E \times R_F\right) - \left(R_E \times R_G\right) - \left(R_F \times R_G\right) + \left(R_E \times R_F \times R_G\right) \quad (4.21)$$

$$= 1 - \left(Q_E \times Q_F \times Q_G\right) \quad (4.22)$$

กรณีที่ต้องมีอย่างน้อย 2 ใน 3 โมดูลต้องทำงานสำเร็จ

โมดูลสถานีควบคุมแบบซ้ำสำรองในระบบจำนวน 3 โมดูล หากพิจารณาเงื่อนไขที่ 2 ใน 3 โมดูลจะต้องทำงานสำเร็จจะพบว่าเกิดเงื่อนไขการทำงานดังนี้ โมดูล E เกิดความล้มเหลวในการทำงานแล้วโมดูล F และ G จะต้องทำงานสำเร็จ หากโมดูล F เกิดความล้มเหลวแล้วโมดูล E และ G จะต้องทำงานสำเร็จ หากโมดูล G เกิดความล้มเหลวในการทำงานแล้ว โมดูล E และ F จะต้องทำงานสำเร็จ และจะพบว่าหากทั้ง 3 โมดูลสามารถทำงานได้สำเร็จระบบรวมก็จะทำงานได้สำเร็จเช่นกัน ความน่าเชื่อถือของระบบสามารถหาได้จากสมการดังนี้

$$
\begin{aligned}
R_Z &= (R_E \times R_F \times R_G) + (Q_E \times R_F \times R_G) + (R_E \times Q_F \times R_G) + (R_E \times R_F \times Q_G) \\
&= (R_E \times R_F \times R_G) + \left(Q_E \times R_F \times R_G \times \frac{R_E}{R_E}\right) + \left(R_E \times Q_F \times R_G \times \frac{R_F}{R_F}\right) + \left(R_E \times R_F \times Q_G \times \frac{R_G}{R_G}\right) \\
&= (R_E \times R_F \times R_G) + \left(R_E \times R_F \times R_G \times \frac{Q_E}{R_E}\right) + \left(R_E \times R_F \times R_G \times \frac{Q_F}{R_F}\right) + \left(R_E \times R_F \times R_G \times \frac{Q_G}{R_G}\right) \\
&= (R_E \times R_F \times R_G) \times \left(1 \times \frac{Q_E}{R_E} \times \frac{Q_F}{R_F} \times \frac{Q_G}{R_G}\right) \qquad (4.23)
\end{aligned}
$$

หรือ

$$
R_Z = e^{-(\lambda_E+\lambda_F+\lambda_G)t} + \left(1-e^{-\lambda_E t}\right)e^{-(\lambda_F+\lambda_G)t} + \left(1-e^{-\lambda_F t}\right)e^{-(\lambda_E+\lambda_G)t} + \left(1-e^{-\lambda_G t}\right)e^{-(\lambda_E+\lambda_F)t} \qquad (4.24)
$$

4.5.4 การหาความน่าเชื่อถือรวมแบบมีเงื่อนไขในการทดลอง

เมื่อพิจารณาไดอะแกรมการเชื่อมต่อของอุปกรณ์ในรูปที่ 4.1 จะพบว่ากลุ่มของ X, Y และ Z มีการเชื่อมต่อกันแบบอนุกรมซึ่งสามารถหาได้จากสมการที่ (2.4)-(2.5) และการทำงานของระบบเป็นการทำงานร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์ หากขาดส่วนใดส่วนหนึ่งจะทำให้ระบบไม่สามารถทำงานได้ ดังนั้น ในการหาความน่าเชื่อถือรวมแต่ละโมดูล สามารถใช้ข้อมูลที่คำนวณหาความน่าเชื่อถือในส่วนของฮาร์ดแวร์และซอฟต์แวร์ที่คำนวณไว้แล้วในหน้าที่ 50, 59 และ 61 และสามารถคำนวณหาความน่าเชื่อถือแต่ละโมดูลได้ดังนี้

ความน่าเชื่อถือโมดูลเอชเอ็มไอ เท่ากับ 0.7383 x 0.8946 = 0.66048

ความน่าเชื่อถือโมดูลเครื่องควบคุม เท่ากับ 0.7383 x 0.8952 = 0.66092

การหาความน่าเชื่อถือแบบมีเงื่อนไขสำหรับสถานีเอชเอ็มไอ (กลุ่ม X)

$$
\begin{aligned}
R_X &= R_1 + R_2 - R_1 R_2 \\
&= 0.66048 + 0.66048 - (0.66048 \times 0.66048) \\
&= 0.88473
\end{aligned}
$$

การหาความน่าเชื่อถือแบบมีเงื่อนไขสำหรับระบบเครือข่ายอีเทอร์เน็ต (กลุ่ม Y)

$$R_Y = e^{-F_C} - e^{-F_D} - e^{-(F_C+F_D)}$$

แทนค่า $F = \lambda t$ จะทำให้ได้

$$R_y = 2e^{-0.1125} - e^{-2(0.1125)}$$
$$= 1.7871\text{-}0.7985\text{=}0.9886$$

การหาความน่าเชื่อถือแบบมีเงื่อนไขสำหรับสถานีควบคุม (กลุ่ม Z) แบ่งตามกรณีดังนี้

กรณีที่ต้องมีอย่างน้อย 1 ใน 3 โมดูลต้องทำงานสำเร็จ

$$R_Z = 3R - 3R^2 + R^3$$
$$= (3 \times 0.66092) - (3 \times 0.66092^2) + 0.66092^3$$
$$= 0.96101$$

กรณีที่ต้องมีอย่างน้อย 2 ใน 3 โมดูลต้องทำงานสำเร็จ

$$R_Z = 3R^2 - 2R^3$$
$$= (3 \times 0.66092^2) - (2 \times 0.66092^3)$$
$$= 0.73305$$

การหาค่าความน่าเชื่อถือของตัวโมดูลโหวตคะแนน

$$R_v = 0.7383 \times 0.8952 = 0.66092$$

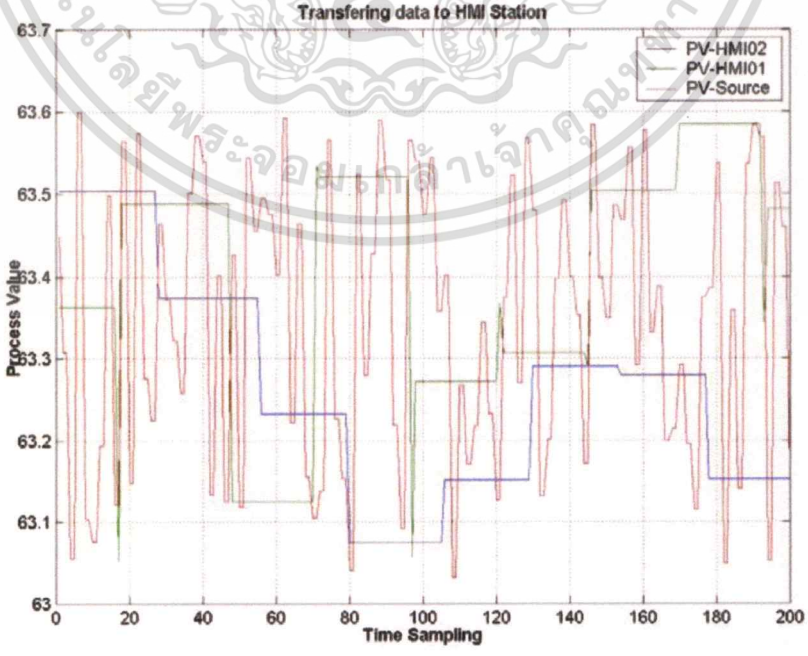
จากการคำนวณหาค่าความน่าเชื่อถือในระบบ โดยมีการต่อเชื่อมต่อกันของโมดูลแบบไม่มีโมดูลซ้ำสำรองทั้ง 3 กลุ่ม (X,Y,Z) โดยเชื่อมต่อกันแบบอนุกรมจะพบว่า ความน่าเชื่อถือรวมทั้งระบบจะเท่ากับ 0.4326 (ไม่มีตัวโหวตคะแนน) หากโมดูลกลุ่ม X และ Y เป็นระบบแบบมีโมดูลซ้ำสำรอง 2 โมดูลและโมดูลกลุ่ม Z มีโมดูลซ้ำสำรองแบบ 3 โมดูล โดยมีเงื่อนไขว่าแต่ละกลุ่มจะต้องมีอย่างน้อย 1 โมดูลที่สามารถทำงานได้ ระบบจึงจะไม่เกิดความล้มเหลวในการทำงาน ความน่าเชื่อถือรวมทั้งระบบตามเงื่อนไขนี้คือ 0.84054 หากรวมความน่าเชื่อถือของตัวโหวตระบบจะมีความน่าเชื่อถือเท่ากับ 0.75246 จะพบว่าหากมีการเชื่อมต่อกันของอุปกรณ์และมีเงื่อนไขในการทำงานแบบนี้แล้ว จะทำให้ความน่าเชื่อถือของระบบเพิ่มขึ้น

4.6 ผลการทดลองจากการสังเกตพฤติกรรมของสัญญาณ

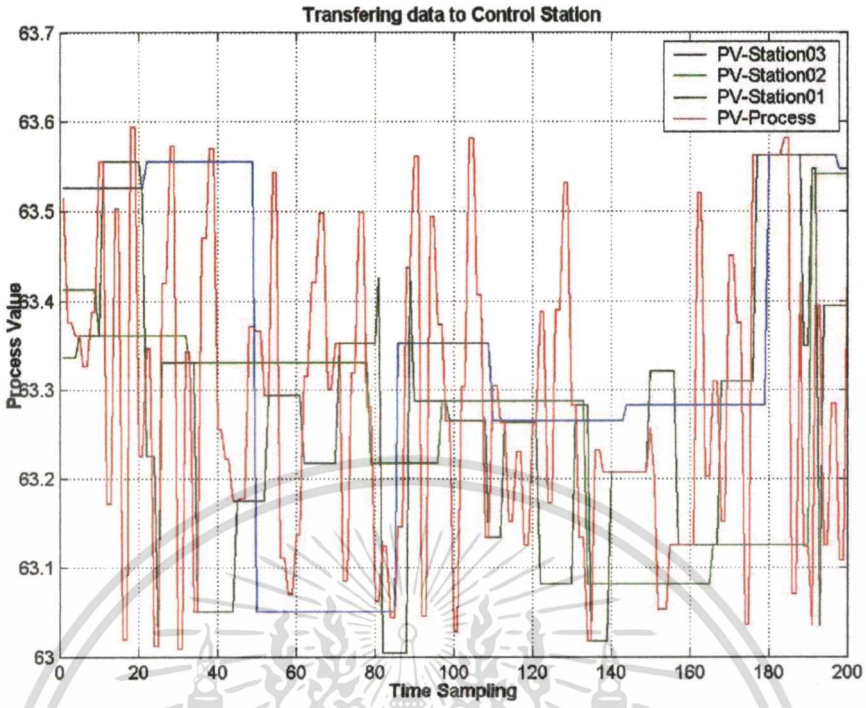
จากการทดลองในการควบคุมความดันกับแบบจำลองกระบวนการควบคุมความดัน โดยการกำหนดค่าพารามิเตอร์ต่างๆ ให้กับตัวควบคุมกระบวนการ เช่น ค่าเป้าหมายที่ต้องการควบคุม (Set Point) ค่าตัวแปรควบคุมพีไอดี (PID) จนค่าจากกระบวนการจำลองมีค่าได้ตรงตามเป้าหมาย จากนั้นจึงกำหนดเงื่อนไขของความล้มเหลวแบบต่างๆ ที่จะเกิดขึ้นเพื่อศึกษาและสังเกตค่าตัวแปรจากกระบวนการ (Process Variable, PV) โดยมีอีกหนึ่งสถานีที่ใช้สำหรับเฝ้าสังเกตและเก็บค่าจากกระบวนการและระบบควบคุม

ค่าพารามิเตอร์ที่ได้เป็นค่าที่วัดโดยใช้โปรแกรมที่พัฒนาขึ้นมาประมวลผล ซึ่งการทำงานของทุกสถานีจะถูกเก็บค่าต่างๆ ลงในฐานข้อมูลตามเหตุการณ์และช่วงเวลาที่กำหนด หมายความว่าจะเริ่มนับตั้งแต่การร้องขอข้อมูลจนกระทั่งรับข้อมูลแล้วเก็บข้อมูลที่ได้ในฐานข้อมูล เช่น หากต้องการอ่านค่าจากกระบวนการ ช่วงเวลาที่ใช้ทั้งหมดจะไม่ใช่ช่วงเวลาที่อ่านจากกระบวนการแล้วแสดงผล แต่จะเป็นช่วงเวลาที่อ่านค่าจากกระบวนการแล้วบันทึกในฐานข้อมูล (เก็บค่าที่อ่านได้ลงบนไฟล์ข้อความ (Text File) ตามช่วงเวลาของการสุ่มเก็บข้อมูล

ในการแสดงผลการทดลองได้นำค่าที่ต้องการสังเกตพฤติกรรม แสดงผลออกมาเป็นกราฟ โดยนำข้อมูลที่ถูกบันทึกลงในฐานข้อมูลออกมาเป็นกราฟด้วยโปรแกรม MATLAB โดยกำหนดสีสำหรับกราฟการควบคุมกระบวนการดังนี้คือ สีน้ำเงินเป็นค่าที่ส่งจากสถานีควบคุมที่หนึ่ง สีส้มเป็นค่าควบคุมที่ส่งจากสถานีที่สอง สีม่วงแดงเข้มเป็นค่าที่ส่งจากสถานีควบคุมที่สาม สีเขียวเป็นค่าควบคุมที่ได้จากการลงคะแนน

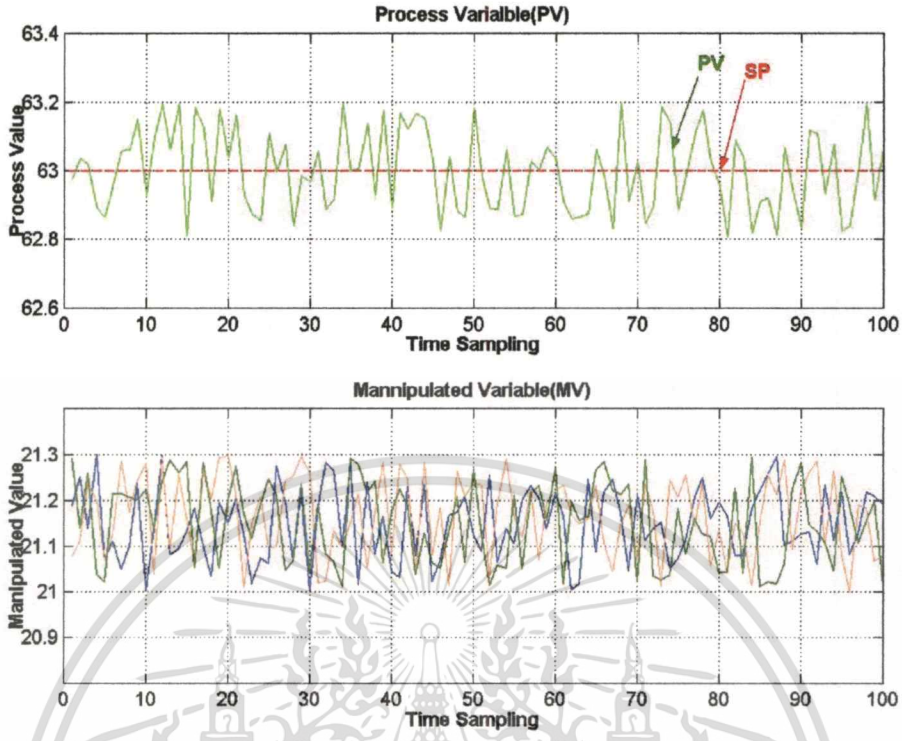


รูปที่ 4.9 กราฟแสดงผลจากกระบวนการในสถานีควบคุมเอชเอ็มไอ



รูปที่ 4.10 กราฟแสดงผลจากกระบวนการที่สถานีควบคุมทั้งสาม

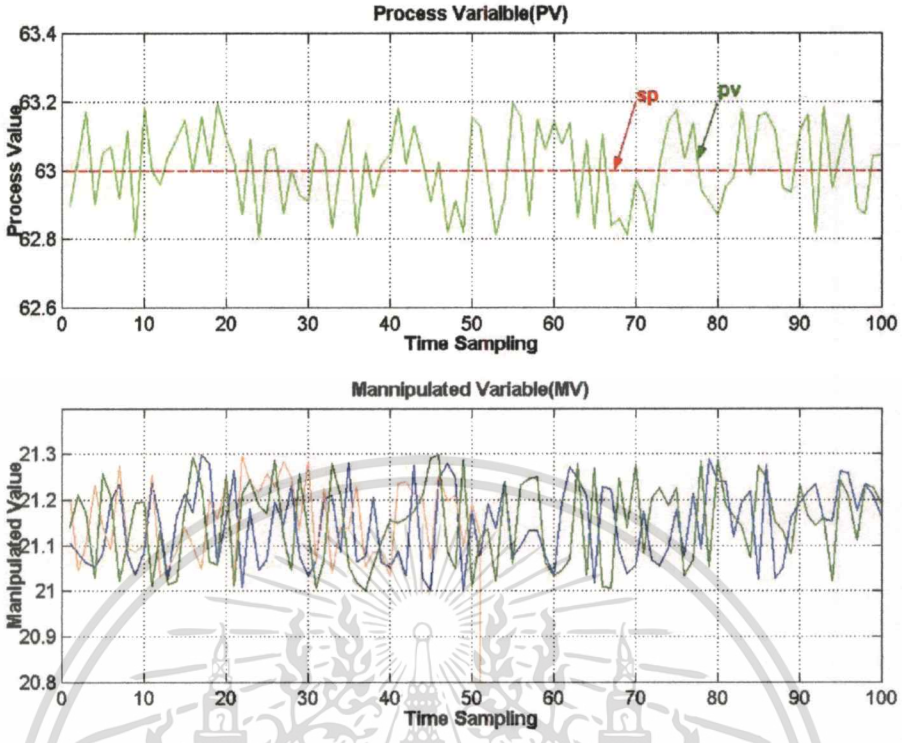
จากข้อมูลในรูปที่ 4.9 และ 4.10 แสดงถึงการส่งผ่านข้อมูลผ่านระบบเครือข่ายอีเทอร์เน็ต โดยเป็นการอ่านค่าจากกระบวนการเดียวกันที่มีการเปลี่ยนแปลงอยู่เสมอ ค่าที่แต่ละสถานีนั้นอ่านได้จะสังเกตได้ว่า ในช่วงเวลาเดียวกันจะอ่านค่าได้ต่างกันทั้งในส่วนที่เป็นสถานีเอชเอ็มไอและในส่วนที่เป็นสถานีควบคุม ทั้งนี้เกิดจากระบบเครือข่ายอีเทอร์เน็ตนั้นไม่รองรับทำงานแบบเวลาจริงได้ เนื่องจากระบบมีการควบคุมการส่งผ่านข้อมูลเพื่อป้องกันการชนกันของข้อมูล จึงทำให้ไม่สามารถกำหนดเวลาได้อย่างแน่นอน และยังประกอบด้วยตัวแปรอื่นอีกเช่น ปริมาณความหนาแน่นของข้อมูลในระบบเครือข่าย เป็นต้น อย่างไรก็ตามในระดับปฏิบัติการหากใช้วิธีการสังเกตทางหน้าจอประมวลผลจะพบความต่างของข้อมูลนั้นน้อยมาก



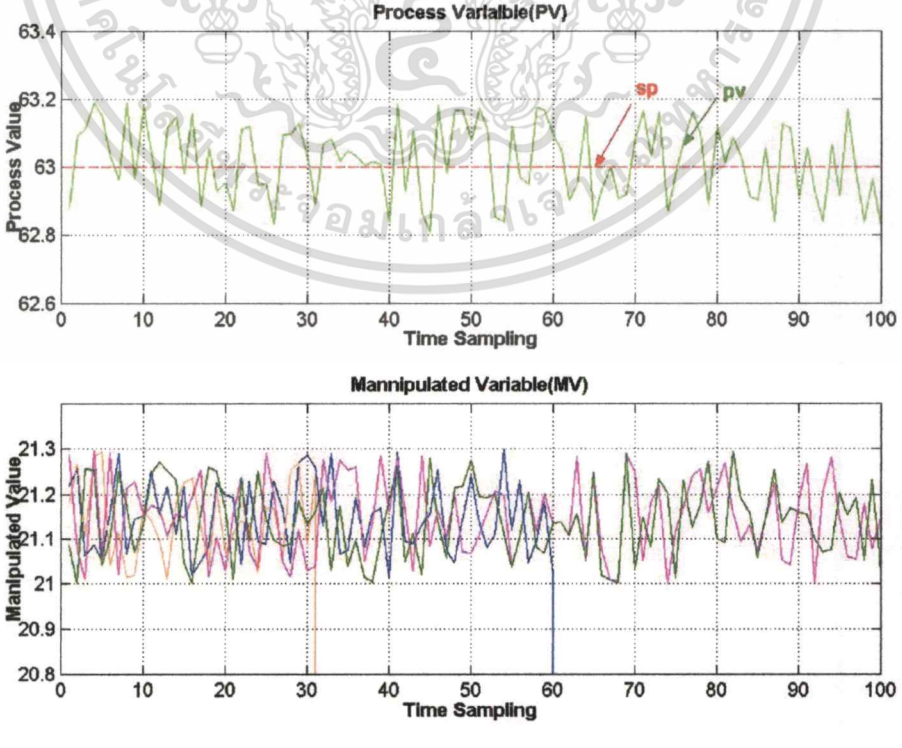
รูปที่ 4.11 สัญญาณการควบคุมเมื่อทั้งสามสถานีควบคุมทำงานตามปกติ

จากรูปที่ 4.11 เป็นภาวะทำงานปกติของระบบ ไม่มีสถานีใดเกิดความผิดพลาดในการทำงาน ส่วนด้านล่างเป็นสัญญาณควบคุมมีจำนวนสี่สัญญาณ โดยแยกเป็นสัญญาณที่มาจากสถานีควบคุมจำนวนสามสถานี และอีกหนึ่งสัญญาณเป็นสัญญาณที่เกิดจากการลงคะแนนเพื่อควบคุมกระบวนการ กราฟจำนวนหนึ่งเส้นจะทับกันพอดีเนื่องจากเป็นสัญญาณที่ได้จากการลงคะแนนเพื่อควบคุมกระบวนการ ค่าความต้องการของกระบวนการถูกต้องไว้ที่ 63

จากรูปที่ 4.12 เมื่อสถานีที่สองเกิดความล้มเหลวในการทำงาน ทำให้ไม่สามารถควบคุมกระบวนการได้โดยในภาพส่วนล่างจะพบสัญญาณควบคุมของสถานีที่สองซึ่งเป็นสีส้มตกลง แต่ระบบยังสามารถควบคุมกระบวนการได้ เนื่องจากตามเงื่อนไขแล้วสถานีที่สามจะเป็นตัวควบคุมกระบวนการ โดยสีเขียวด้านล่างเป็นสัญญาณที่ได้จากการโหวตคะแนนเพื่อควบคุมกระบวนการ

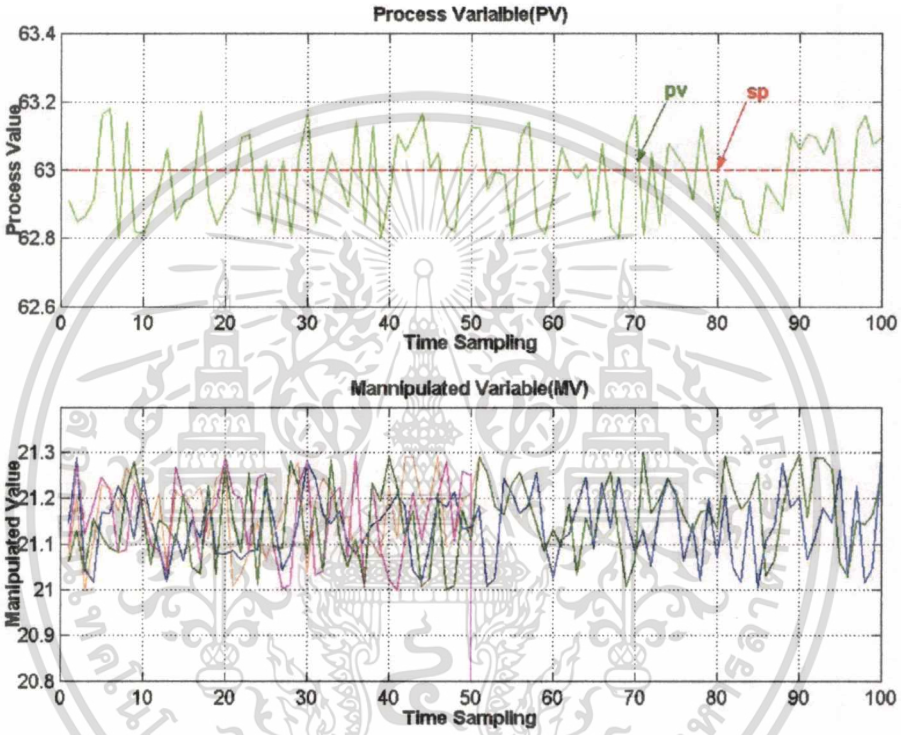


รูปที่ 4.12 สัญญาณการควบคุมเมื่อสถานีควบคุมที่สองหยุดการทำงานส่วนสถานีอื่นทำงาน



รูปที่ 4.13 สัญญาณการควบคุมเมื่อสถานีควบคุมที่สองหยุดการทำงานส่วนสถานีอื่นทำงาน

รูปที่ 4.13 แสดงสภาวะของระบบเมื่อสถานีที่สองและสถานีที่หนึ่งเกิดความล้มเหลวในการควบคุมตามลำดับ หากสังเกตในกราฟด้านล่างจะพบว่ากราฟสีส้มและสีน้ำเงินจะตกลงสู่ค่าศูนย์ตามลำดับ เนื่องจากสถานีที่สองและหนึ่งเกิดความล้มเหลวไม่สามารถส่งสัญญาณออกมาควบคุมกระบวนการได้ แต่เงื่อนไขถูกกำหนดว่าสถานีที่สามยังถูกลงคะแนนให้เป็นสถานีที่ควบคุมกระบวนการ ดังนั้นการควบคุมกระบวนการก็ยังคงดำเนินไปได้โดยที่ใช้สัญญาณสีเขียวซึ่งเป็นสัญญาณควบคุมจากกระบวนการที่สาม ไปควบคุมกระบวนการ

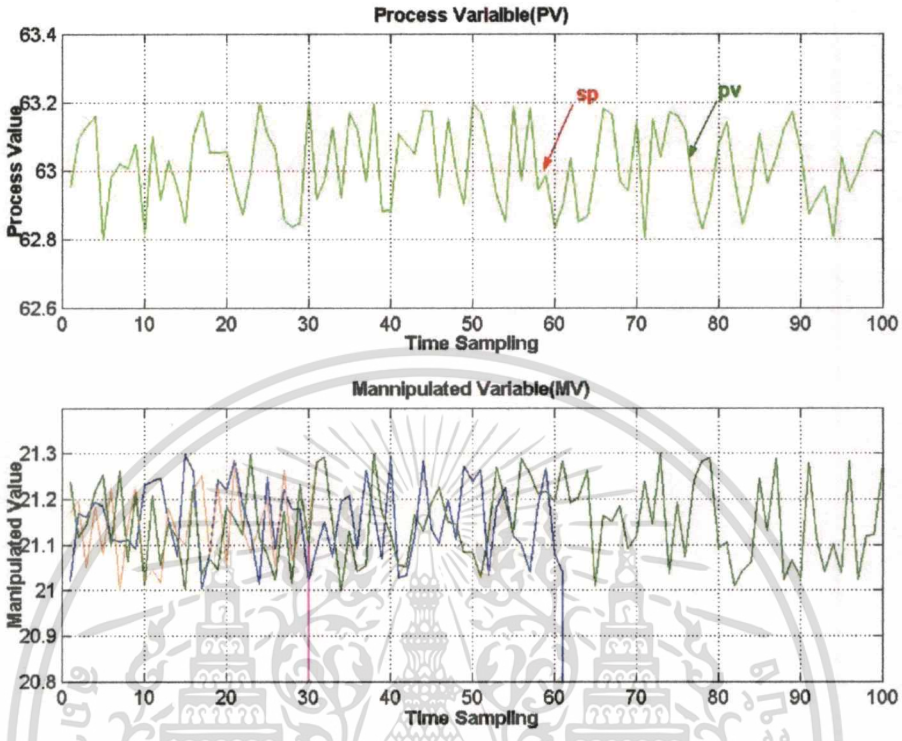


รูปที่ 4.14 สัญญาณการควบคุมเมื่อสถานีควบคุมที่สามหยุดการทำงานส่วนสถานีอื่นทำงาน

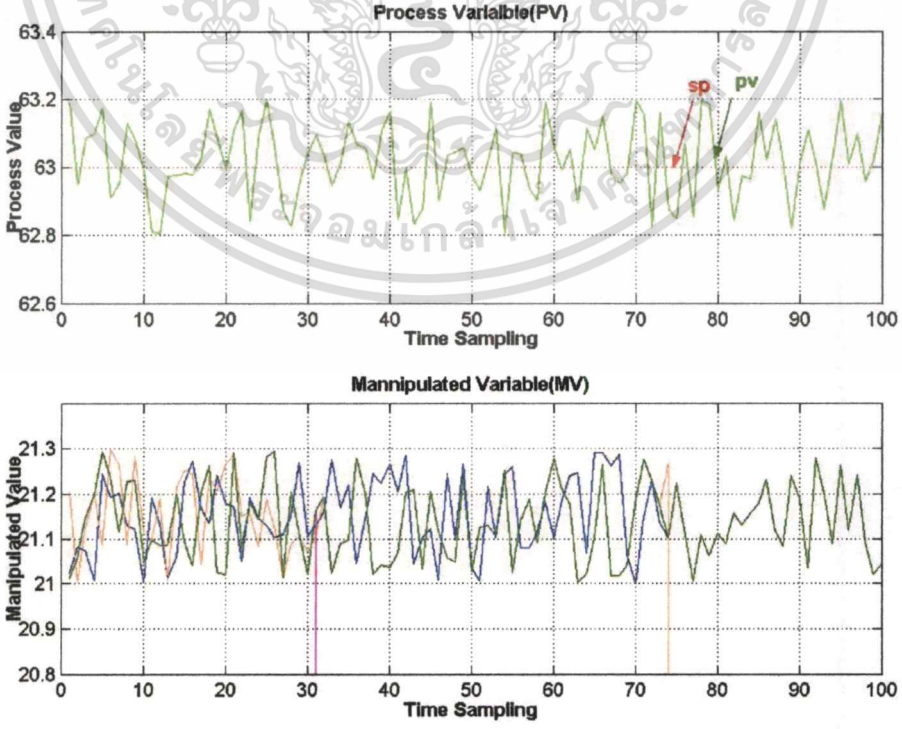
รูปที่ 4.14 แสดงการหยุดการทำงานของสถานีที่สามโดยสถานีนี้เป็นสถานีที่กำลังควบคุมกระบวนการ สังเกตได้จากกราฟสัญญาณควบคุมสีม่วงแดงเข้มในกราฟด้านล่างของรูปที่ 4.11 ตกลงสู่ศูนย์ โดยเหลือสัญญาณของสถานีที่สองและหนึ่งที่สามารถควบคุมกระบวนการได้ ซึ่งสีเขียวเป็นสัญญาณควบคุมที่ได้จากการลงคะแนนจะทับกับเส้นของสัญญาณควบคุมที่มาจากสถานีที่สองซึ่งเป็นสีส้ม

รูปที่ 4.15 เป็นการควบคุมกระบวนการเมื่อสถานีควบคุมที่สามและหนึ่ง เกิดความล้มเหลวในการควบคุมตามลำดับ โดยสังเกตจากกราฟด้านล่างในรูปที่ 4.15 จะพบว่ากราฟสีม่วงแดงเข้มจะตกลงสู่ศูนย์และกราฟสีน้ำเงินก็จะตกลงสู่ศูนย์เช่นกัน เมื่อสถานีที่สามและหนึ่งหยุดการทำงาน แต่การควบคุมกระบวนการยังสามารถดำเนินต่อไปได้ เนื่องจากตามเงื่อนไขดังกล่าว

สถานีที่สองถูกลงคะแนนให้เป็นผู้ควบคุมกระบวนการ โดยสัญญาณควบคุมจะทับกับสัญญาณลงคะแนนที่เป็นสีเขียว

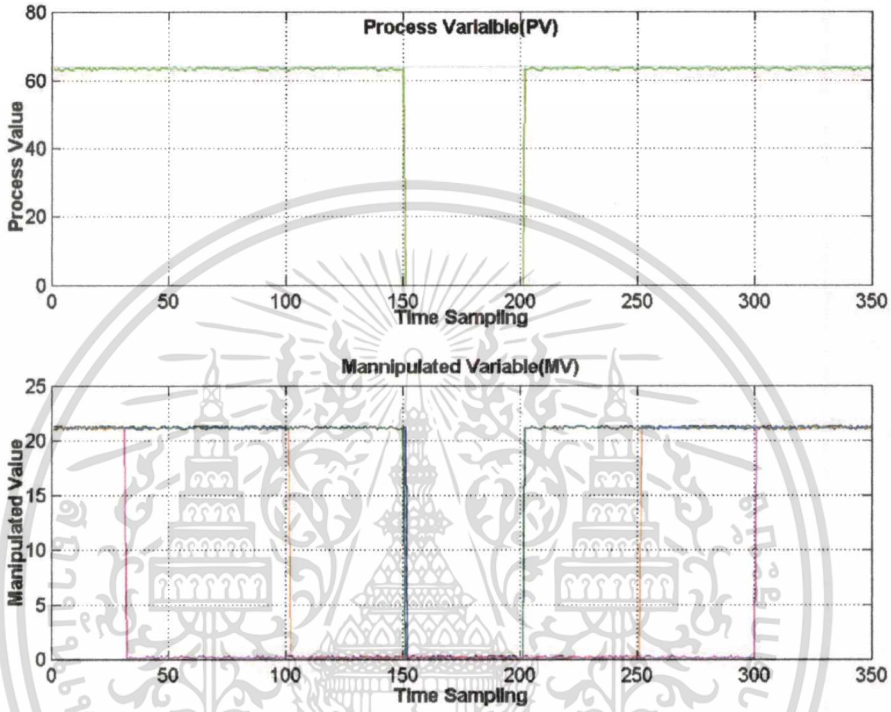


รูปที่ 4.15 สัญญาณการควบคุมเมื่อสถานีที่หนึ่งและสามหยุดการทำงาน



รูปที่ 4.16 สัญญาณการควบคุมเมื่อสถานีที่หนึ่งถึงสามกลับทำงานได้ตามลำดับ

รูปที่ 4.16 แสดงการทำงานเมื่อสถานีที่สามและสถานีที่สองเกิดความล้มเหลวในการทำงานตามลำดับ สังเกตจากกราฟควบคุมที่ส่งไปควบคุมกระบวนการจะพบว่า สัญญาณควบคุมของสถานีที่สามสีม่วงแดงเข้มและสัญญาณควบคุมของสถานีที่สองสีส้มจะตกลง และสัญญาณควบคุมจากสถานีที่หนึ่งได้ถูกลงคะแนนให้เป็นสัญญาณที่ใช้ในการควบคุมกระบวนการ



รูปที่ 4.17 สัญญาณการควบคุมเมื่อทุกสถานีหยุดทำงานและกลับทำงานได้ใหม่

รูปที่ 4.17 แสดงการทำงานที่เกิดการล้มเหลวโดยเริ่มจากสถานีควบคุมที่สามถึงหนึ่งตามลำดับ โดยเมื่อเกิดความล้มเหลวสถานีอื่นสามารถเข้าไปควบคุมกระบวนการแทน ซึ่งเป็นไปตามเงื่อนไขของการลงคะแนนที่ได้กำหนดไว้ จนกระทั่งทุกสถานีเกิดความล้มเหลวในการทำงานก็จะทำให้ระบบเกิดความล้มเหลว แต่เมื่อมีการกลับคืนสู่สภาวะเดิมก็จะทำให้เป็นไปตามเงื่อนไขของการลงคะแนนที่ได้กำหนดไว้

4.7 สรุปผลการทดลอง

จากผลการทดลอง ในการควบคุมความดันกับแบบจำลองกระบวนการควบคุมความดัน กระบวนการสามารถทำงานได้อย่างต่อเนื่อง แม้ว่าสถานีควบคุมใดสถานีหนึ่งจะหยุดการทำงาน แต่ลงคงเหลืออย่างน้อยหนึ่งสถานีที่ยังสามารถทำงานได้ การควบคุมกระบวนการก็ยังคงสามารถทำงานได้อย่างต่อเนื่องเช่นกัน โดยโปรแกรมการลงคะแนนที่ออกแบบไว้สามารถทำงานได้ตามที่กำหนดไว้ ช่วงเวลาในการในการสลับการทำงานนั้นขึ้นอยู่กับปัจจัยหลายอย่างเช่น ปริมาณข้อมูลที่มีในระบบเครือข่าย อย่างไรก็ตามในการทดลองนี้ช่วงเวลาของการสลับนั้นน้อยกว่า หนึ่งส่วนพันวินาที ซึ่งในระดับของโปรแกรมที่พัฒนาขึ้นไม่สามารถนับเวลาในระดับต่ำกว่าหนึ่งส่วนพันวินาทีได้ หากพิจารณาช่วงเวลาที่ยอมรับได้ที่อ้างอิงในหัวข้อที่ 3.4 เรื่องการควบคุมความผิดพลาด กล่าวว่า “ค่าเวลาที่ยอมรับได้โดยทั่วไปของกระบวนการใช้เวลาประมาณ 1 วินาที” ก็จะทำให้ระบบที่ออกแบบสามารถที่จะควบคุมกระบวนการต่อไปอย่างต่อเนื่อง เนื่องจากหากเกิดความล้มเหลวขึ้นกับโมดูลแล้ว โมดูลที่ซ้ำสำรองสามารถสับเปลี่ยนขึ้นมาทำงานทดแทนได้ภายในระยะเวลา 1 มิลลิวินาที

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

การควบคุมกระบวนการให้สามารถทำงานได้อย่างต่อเนื่องนั้น มีความสำคัญเป็นอย่างยิ่ง โดยเฉพาะกระบวนการที่มีความสำคัญสูง ซึ่งหากเกิดความผิดปกติขึ้นในกระบวนการดังกล่าว จะทำให้เกิดปัญหาตามมาอีกมากมาย จึงจำเป็นต้องหาวิธีการที่จะทำให้ระบบมีความต่อเนื่องในการทำการควบคุม เพื่อที่จะลดความเสียหายอันเกิดจากการที่ระบบควบคุมหยุดการทำงาน

ระบบควบคุมที่ทนต่อความผิดพร่อง เป็นวิธีหนึ่งที่ถูกใช้ในการเพิ่มความน่าเชื่อถือให้กับระบบควบคุม ผู้วิจัยได้ออกแบบการทดลองให้คล้ายคลึงกับการทำงานจริงของระบบควบคุมที่ใช้ในอุตสาหกรรมจริงๆ และได้เปลี่ยนตัวควบคุมจากเครื่องควบคุมตรรกแบบโปรแกรมได้เป็นเครื่องควบคุมที่เป็นแบบคอมพิวเตอร์ส่วนบุคคล ด้วยสาเหตุที่ต้องการจะให้เครื่องควบคุมสามารถทำงานตามสถาปัตยกรรมแบบเปิด ไม่จำกัดการติดต่อสื่อสารกับอุปกรณ์จากแหล่งผู้ผลิตเดียว ในการทดลองแบ่งการทดลองออกเป็นส่วนๆ ให้ทำงานเป็นอิสระต่อกัน คือ ส่วนที่เป็นสถานีเอชเอ็มไอซึ่งมีจำนวน 2 สถานี และส่วนที่เป็นสถานีตัวควบคุมซึ่งมีอยู่ 3 สถานีซึ่งสถานีควบคุมทั้ง 3 สถานีจะอ่านค่าจากกระบวนการเดียวกัน โดยพัฒนาโปรแกรมขึ้นมาจัดการระบบควบคุมที่ทนต่อความผิดพร่อง

จากผลการทดลองซึ่งแบ่งการทดลองออกเป็นส่วนๆ ทั้งในส่วนที่เป็นสถานีเอชเอ็มไอและสถานีตัวควบคุมซึ่งทำงานแยกเป็นอิสระต่อกัน พบว่าระบบแบบโมดูลซ้ำสำรองที่ทำการออกแบบสามารถเพิ่มความน่าเชื่อถือให้กับระบบ ทั้งในแง่ของแบบจำลองสมการทางคณิตศาสตร์และการสังเกตสัญญาณของระบบ โดยระบบที่ทำการพัฒนาขึ้นมาสามารถแสดงผลและทำการควบคุมได้อย่างต่อเนื่อง เมื่อเกิดความผิดพลาดขึ้นในการควบคุมกระบวนการ ไม่ว่าจะเกิดขึ้นในส่วนที่เป็นสถานีเอชเอ็มไอ ในส่วนที่เป็นตัวควบคุมหรือเกิดจากระบบเครือข่าย โปรแกรมที่พัฒนาขึ้นมาจัดการระบบควบคุมที่ทนต่อความผิดพร่องสามารถจัดการสลับการทำงานระหว่างสถานีที่เกิดความล้มเหลว และสถานีที่ยังคงสถานะใช้งานได้

วิธีการที่ถูกนำเสนอในวิทยานิพนธ์นี้ เป็นวิธีการหนึ่งที่สามารถเพิ่มความน่าเชื่อถือให้กับระบบควบคุมซึ่งมีสามารถรองรับการทำงานภายใต้สถาปัตยกรรมแบบเปิด การทดลองในงานวิจัยเป็นการทดลองทำระบบควบคุมที่ทนต่อความผิดพร่องเบื้องต้นเท่านั้น ยังไม่เป็นการสร้างระบบควบคุมที่ทนต่อความผิดพร่องอย่างสมบูรณ์ เนื่องจากในการทดลองนี้เป็นเพียงการทำระบบที่ทนต่อความผิดพร่องที่อาศัยซอฟต์แวร์ที่พัฒนาขึ้นมาจัดการเท่านั้น ข้อมูลในการทดลอง เช่น การตรวจจับความผิดพลาดที่เกิดขึ้นในระบบล้วนเป็นค่าที่ได้มาจากซอฟต์แวร์ทั้งสิ้น ในการพัฒนาซอฟต์แวร์ขึ้นมาไม่สามารถทำให้ซอฟต์แวร์ที่พัฒนาขึ้นมาสมบูรณ์ได้ทั้งหมด ซึ่งจะมีผลโดยตรงต่อการผลการทดลอง ดังนั้นการทำระบบที่ทนต่อความผิดพร่องให้สมบูรณ์ ยังจะต้องอาศัยการ

ออกแบบทางฮาร์ดแวร์เข้าช่วยด้วย อย่างไรก็ตามผลการทดลองก็สามารถรับรองได้ว่า หากใช้ระบบที่ออกแบบแล้วและเวลาที่กระบวนการยอมรับได้ไม่น้อยกว่า 1/1000 วินาที หากเกิดความผิดพร่องกับโมดูลใดก็ตาม โมดูลซ้ำสำรองสามารถสับเปลี่ยนขึ้นมาทำงานทดแทนได้ทันไม่ทำให้ระบบเกิดความล้มเหลว

ในการพัฒนาต่อไป เพื่อให้การควบคุมระบบดียิ่งขึ้น คือ ควรมีการศึกษาการเคลื่อนไหลของของข้อมูลในระบบเครือข่าย เนื่องจากระบบควบคุมแบบคอมพิวเตอร์ที่ทนต่อความผิดพร่องรองรับการทำงานตามสถาปัตยกรรมแบบเปิด หากมีการต่อขยายแผงวงจรต่อประสานจำนวนมากจะทำให้อัตราการส่งข้อมูลผ่านระบบเครือข่ายอินทราเน็ตลดลง ควรมีการศึกษาเกี่ยวกับข้อจำกัดอัตราการเคลื่อนไหลของข้อมูลในเครือข่ายอินทราเน็ต อีกทั้งการทดลองใช้ระบบอินทราเน็ตชนิดที่ถูกพัฒนาขึ้นมาใหม่ เช่น จิกะบิตอินทราเน็ต

บรรณานุกรม

[1] สุพรรณ กุลพานิชย์, Programmable Controller เทคนิคและการใช้งานเบื้องต้น (เล่ม 1), ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

[2] พิพัฒน์ หิรัณย์วณิชชากร, ระบบการสื่อสารและเครือข่ายคอมพิวเตอร์, ซีเอ็ดยูเคชั่น, 2544.

[3] ทวิช ชูเมือง, ระบบวัดคุมนิรภัยในอุตสาหกรรมกระบวนการผลิต, ซีเอ็ดยูเคชั่น, 2548.

[4] ตฤณ แสงสุวรรณ, การหาค่าความเชื่อถือได้ในระบบไฟฟ้ากำลังเบื้องต้น, สำนักพิมพ์มหาวิทยาลัยเกษตรศาสตร์, 2546.

[5] นิรุธ อำนวยศิลป์, Visual C++ and MFC Programming, บริษัท THAIDEV.COM, 2548.

[6] นิรุธ อำนวยศิลป์, Network and Protocos Programming Using C/C++, บริษัท THAIDEV.COM, 2548.

[7] มนัส สังวรศิลป์, คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์, สำนักพิมพ์อินโฟเพรส, 2543.

[8] Shu-Ho Dai, Ming-O Wang, Reliability Analysis in Engineering Applications, Van Nostrand Reinhold, New York, 1992.

[9] Life Cycle Costing for System Acquistions (interim), Guide No. LCC3, Department of Defense, Washington, D.C., January 1973.

[10] E.E.KEWIS, Introduction to Reliability Engineering, John Wiley & Sons Inc.,New York, 1996.

[11] B.S. Dhillon, DESIGN RELIABILITY Fundamentals and Applications, CRC Press LLC, New York, 1999.

[12] David J. Smith, Reliability Maintainability and Risk, Butterworth-Heinemann Ltd, 1993.

[13] BERTRAM L. AMSTADTER, RELIABILITY MATHEMATICS, McGRAW-HILL BOOK COMPANY, New York, 1971.

[14] John Park, Steve Mackay, Practical Data Acquisition for Instrumentation and Control System. Great Britain: Elsevier, 2003.

[15] Karl J. Astrom and Tore Hagglund, PID Controllers. United State of Amarica: Instrument Society of Amarica, 1988.

[16] R. Ramakumar, Engineering Reliability: Fundamentals and Applications, New Jersey, United States of America, Prentice-Hall Internationals, 1993.

[17] Agustin Rullan, "A Programmable Logic Controllers versus Personal Computers for Process Control" ,Computers ind. Engng, Vol. 33, pp. 412 –424, 1997.

[18] S. Vitturi, "PC-based automation system: an example of application for the real time control of blowing machines," Computer Standards & Interface 26, pp. 145 –146, 2004.

[19] Barry W. Johnson, "Design and Analysis of Fault-Tolerant Digital System", Addison-Wesley Publishing Company Inc., New York.1957.

[20] MARTIN L. SHOOMAN, "Reliability of Computer Systems And Networks", JHON WILEY & SONS,INC., New York. 2002.

[21] T. Suesut, Prayut Inban, A. Numsomran, V. Tipsuwanporn, "Redundant System based PLC Network for High Priority Process," ICCAS 2003 KOREA, October 22-25, 2003.

[22] J. Rodkamtui, A. Numsomram, V. Tipsuwanporn, "Fault Tolerant Control System in Critical Process Based on Ethernet Network", Proceeding of the Third AUS International Symposium on Mechatronics AUS-ISM06,UAE.April 18-23, 2006.

[23] WILLIAM W. EVERETT, "Software Reliability Measurement", IEEE JOURNAL ON SELECTED AREA IN COMMUNICATION, Vol 8, No. 2 February 1990.

[24] Allen P. Nikora, "An Experiment in Determining Software Reliability Model Applicability", IEEE TRANSACTIONS ON RELIABILITY. Vol. 19. No.1, MARCH 2000.

[25] JEFF TIAN, "Software Quality Engineering", A JOHN WILEY&SON,INC., PUBLICATION, Hoboken, New Jersey.2005.

[26] John D. Musa, "Software Reliability Measurement Prediction and Application", McGraw-Hill Book, Singapore, 1987.

[27] John D. Musa, "Software Reliability Engineering",McGraw-Hill, Singapore,1998.

[28] M. Xie, "Software Reliability Modeling",Wold Scientific Publishing Co.Pte.Ltd.,1991.

[29] อัษฎายุทธ แสงระยับ, "การเพิ่มความน่าเชื่อถือระบบพีแอลซีผ่านระบบเครือข่าย", วิทยานิพนธ์, บัณฑิตศึกษา, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2548.

ภาคผนวก

ภาคผนวก ก.

ข้อมูลทางเทคนิคของหน่วยอีเทอร์เน็ตไอโอ

SIXNET®
331 Ushers Road, P.O. Box 767, Clifton Park, NY 12065 USA
+1 (518) 877-5173, Fax +1 (518) 877-8346
email: sales@sixnetio.com www.sixnetio.com
Makes Your Job Easier

I/O Concentrator

Compact interface for Ethernet I/O, RS485 I/O, or Local I/O

- **Ethernet, RS232, RS485, & local I/O expansion**
  Four ports to connect thousands of I/O
  Choose from over 40 ***SIXNET*** I/O module types
  Interface to other vendors Modbus I/O
- **Flexible Communications**
  Supports Open Modbus/TCP, ASCII, & RTU
  Supports telephone, wireless, and other links
  Master, Slave, and Passthru functionality
- **Powerful configuration software**
  All features completely configurable
  Absolutely no programming required
- **Compact DIN-rail industrial package**
  Zone 2, UL, CSA, CE, and DNV rated
  -30 to +70°C operating temperature range

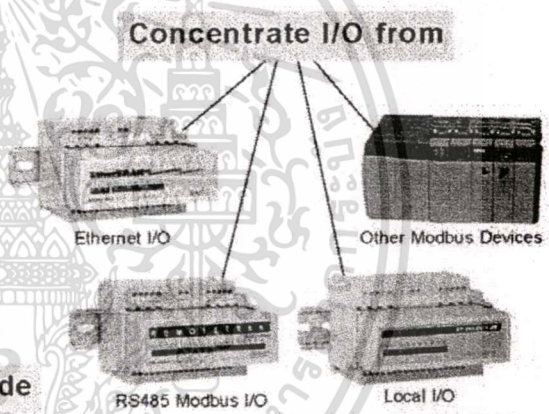
Universal I/O Interface for *SIXNET* and Modbus Systems

The EtherTRAK I/O Concentrator (ET-GT-ST-2) is an universal interface for your *SIXNET* and Modbus I/O. It provides transparent bridges between your Ethernet, RS232 or RS485 communications. It can poll your I/O modules or Modbus devices over one communication link and then make the data available over another.

This one I/O Concentrator replaces *SIXNET*'s legacy non-programmable SixTRAK Gateways (ST-GT-xxx-02N) and EtherTRAK Ethernet I/O Expander (ET-GT-ST-1).

Common applications

- Ethernet to SixTRAK or RemoteTRAK I/O interface
- Ethernet to third party Modbus I/O interface
- Ethernet to Legacy *SIXNET* hardware



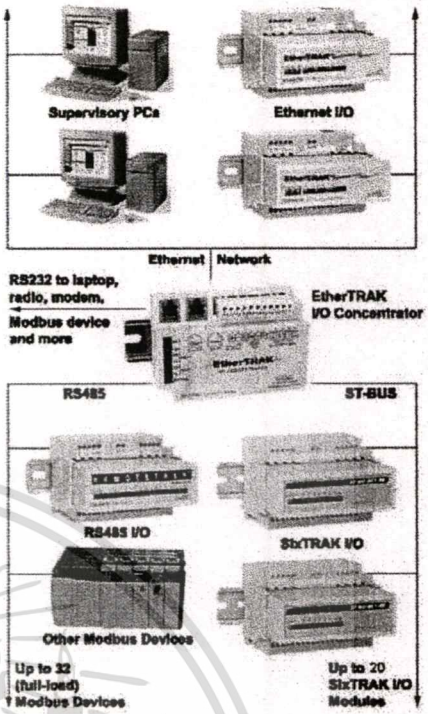
SIXNET I/O controller selection guide

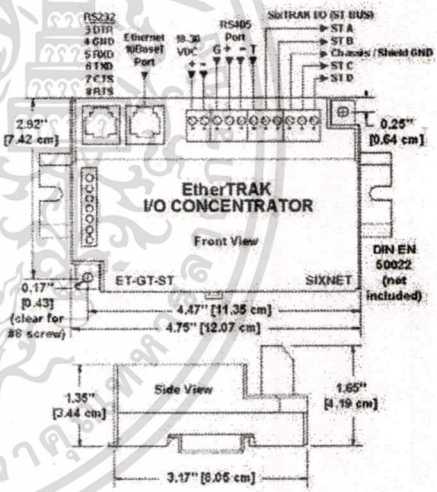
Product	Description	Maximum Memory	LINUX Enabled	ISaGRAF Programs	Data-logging	Total Ports	232 Ports	485 Ports	Ethernet Ports
ST-GT-xxx-02N	Legacy I/O Gateway *	64K	–	–	–	2	1	1	1
ET-GT-ST-1	Ethernet I/O Expander *	128K	–	–	–	2	1	–	1
ET-GT-ST-2	**I/O Concentrator**	**128K**	–	–	–	**3**	**1**	**1**	**1**
ST-GT-1210	I/O Controller	512K & 16M	OEM Only	1	Yes	3	1	1	1
ST-IPM-####	Open Controller	2M & 16M+	Yes	4	Yes	8	2	1	5

* Note: These legacy models have been replaced by the ET-GT-ST-2.

Performance Specifications

General	I/O Concentrator
Maximum I/O registers	2,048 each of analog, floats, & longs; 8,000 each of discrete inputs & outputs
Configuration & diagnostics	SIXNET I/O Tool Kit software
SixTRAK I/O (ST-BUS) port	Up to 20 modules
SixTRAK I/O channels	Up to 640
SixTRAK I/O specs.	See individual data sheets
ST-BUS specs.	See user manual for details
Ethernet port	RJ45, 10BaseT at 10Mbps
Isolation	1200 VRMS 1 minute
Protocols	Modbus and SIXNET over TCP or UDP
Serial ports	Up to 38,400 baud
RS232 port	RJ45 (TD, RD, CTS, RTS, DTR, GND)
RS485 port	Screws (485+, 485-, GND) 2-wire half-duplex
RS485 network	Up to 32 (full-load) stations
RS485 distance	Up to 0.5 miles (1 km)
Flow control	Hardware, software, half/full-duplex modem
Protocols	Modbus RTU, Modbus ASCII, and *SIXNET*
Modes	Master, slave, and passthru
Environmental	DIN rail or flat panel mount
Input voltage	18-30 VDC
Power (less modules)	1.6 watts
Temperature	-30 to 70°C (-40 to 85°C storage)
Humidity	5% to 95% RH (non-condensing)
Electrical Safety	UL 508, CSA C22.2/14; EN61010-1 (IEC1010); CE
EMI emissions	FCC part 15, ICES-003, Class A; EN55022, EN61326-1; CE
EMC immunity	EN61326-1 (EN61000-4-2,3,4,6); CE
Surge withstand	IEEE-472 (ANSI C37.90)
Vibration	IEC68-2-6
Hazardous locations (Class 1, Div 2, Groups A,B,C,D)	UL 1604, CSA C22.2/213, Cenelec EN50021 Zone 2
Marine & Offshore	DNV (Det Norske Veritas)





Ordering Information

ET-GT-ST-2	EtherTRAK I/O Concentrator
Note: Replaces legacy non-programmable SixTRAK Gateways (part numbers ST-GT-xxx-02N) and the EtherTRAK I/O Expander (ET-GT-ST-1).	
Accessories	
Local I/O modules	See SixTRAK ordering guide
Ethernet I/O modules	See EtherTRAK ordering guide
RS485 I/O modules	See RemoteTRAK ordering guide
VT-MODEM-1	Industrial telephone modem for remote access
RM-PS-024-01F	Universal AC/DC to 24 VDC power supply
SXTOOLS-# *	*SIXNET* I/O Tool Kit software for configuration and diagnostics (Level 1 is free)
PAK#####-#### *	Complete Packaged System – ready for installation
* Note: See separate *SIXNET* ordering guide for details.	

Specifications are subject to change.
Consult factory for latest information.

SIXNET ● PO Box 767 ● Clifton Park, NY 12065 USA ● +1 (518) 877-5173 ● Fax +1 (518) 877-8346 ● sales@sixnetio.com ● www.sixnetio.com

SIXNET®
331 Ushers Road, P.O. Box 767, Clifton Park, NY 12065 USA
+1 (518) 877-5173, Fax +1 (518) 877-8346
email: sales@sixnetio.com www.sixnetio.com

ET-MIX24880-D Performance Specifications

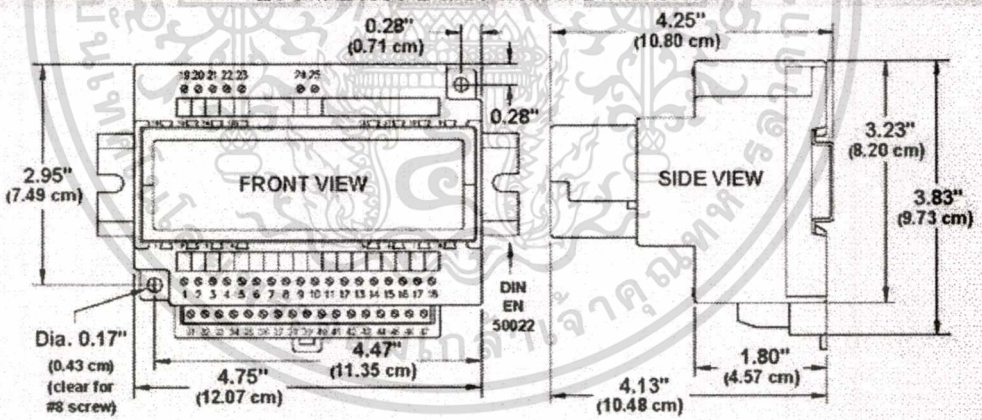
Ethernet Port	**10BaseT at 10 Mbps**
Isolation	1200 Volts RMS 1 minute
Protocols	SIXNET and Modbus over TCP/IP or UDP
RS485 Port	**Up to 38,400 baud**
Operation	Master (passthru) or slave
Supported modes (see the Usage Tips in the catalog for details)	Expansion I/O (as a master), distributed I/O or backup communications path (as a slave)
Protocols	SIXNET, Modbus RTU / ASCII
Maximum distance	Up to 0.5 mile (0.8 km)
Environmental	**DIN rail or panel mount**
Required user supply	10-30 VDC
Power (typ.) @ 24 VDC	1 Watt, 42 mA (excluding I/O)
Operating temp. range	-40 to +70°C
Storage temp. range	-40 to +85°C
Humidity (non-condens.)	5 to 95% RH
Vibration	IEC68-2-6
Electrical safety	UL508, CSA C22.2/14; EN61010
EMI emissions	FCC part 15, ICES-003; EN55022
EMC immunity	EN50082-1, EN61326-1
Surge withstand	IEEE-472
Hazardous locations (Class 1, Div 2 / Zone 2)	UL1604, CSA C22.2/213, EN50021, EEx nA II T4 X
Marine/offshore locations	Det Norske Veritas (DNV) No. 2.4 (Class A & B)

24 Discrete inputs *	**10-30 VDC**
Channels 1 through 8	Sinking or sourcing (as a group)
Channels 9 through 24	Sourcing only
Guaranteed ON voltage	9 VDC
Guaranteed OFF voltage	5.0 VDC
Guaranteed OFF current	1.5 mA DC
Input resistance & current	10K Ohms & 2.4 mA @ 24VDC
Filtered ON/OFF delay	25 mS (10 Hz max. counting)
Fast ON/OFF delay	4 mS (100 Hz max. counting)
Counters on 1st 8 channels	50 KHz on channel 1 and 2
Counter modes	Count up, run time and pulse rate (with selectable time bases)
8 Discrete Outputs *	**10-30 VDC**
Max. output current	1 A per channel, 8A per module
Max. OFF state leakage	0.05 mA
Min. load	1 mA
Inrush current	5 Amps (100 mS surge)
Typ. ON resistance / volt.	0.3 Ohms / 0.3 VDC (@1A)
8 Analog Inputs	**4-20 mA (other ranges available)**
A/D resolution	16 bits (0.003%)
Full scale accuracy	+/-0.1% (@20°C)
Span & offset temp. coeff.	+/-50 ppm per degree C
Input impedance	100 Ohm
Current protection	Self-resetting fuses
DMRR	66 dB at 50/60 Hz
Short circuit protection	Current limiting

* Note: Discrete I/O channels 16-24 can be configured individually to be discrete inputs or discrete outputs. There are a total of 24 discrete I/O channels.

Specifications are subject to change. Consult the factory for the latest information.

ET-MIX24880-D Mechanical Dimensions



ภาคผนวก ข.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. J. Rodkamtui, A. Numsomram, V.Tipsuwanporn, "**Fault Tolerant Control System In Critical Process Based on Ethernet Network**" International Symposium on Mechatronic (AUS-ISM06), AUS, Sharjah, UAE, April 18-23, 2006.

Proceedings of the 3rd AUS International Symposium on Mechatronics (AUS-ISM06)
American University of Sharjah, Sharjah, U.A.E.
April 18-20, 2006

American University of Sharjah

Third AUS-International Symposium on Mechatronics

April 18-20, 2006

Sharjah, UAE

Organized by:

Mechatronics Engineering Graduate Program

Sponsored By:

Dubai Aluminium

IEEE-UAE

Proceeding of the Third AUS International Symposium on Mechatronics AUS-ISM06, AUS, Sharjah, U.A.E. April 18-20, 2006

FAULT TOLERANT CONTROL SYSTEM IN CRITICAL PROCESS BASED ON ETHERNET NETWORK

J. Radkamtui, A. Numsomram , V. Tipsuwanporn

King Mongkut's Institute of Technology Ladkrabang
Faculty of Engineering
Department of Instrumentation Engineering
Bangkok 15520, Thailand.

ktvittay@kmitl.ac.th

P. Inban

Rajabhat Rajanagarindra University
Faculty of Industrial Technology
Department of Industrial Electrical Technology
Chachoengsao 24000, Thailand.
prof_prayuth@yahoo.com.sg

ABSTRACT

The purpose of this paper is to study and develop the redundant control system for the critical process. The critical process means a high priority process which is very essential to the global system. When a fault condition occurs on such process, it affects to other processes. In this paper, the gas pressure process is used a case study. This system consists of dual redundant supervisory control system (HMI Station), dual redundant Ethernet network, triple redundant controller system and triple redundant filed signal. The experimental result is able to indicate some fault tolerance features. Our system can increase the reliability as well as the risk of fault of a whole system and apply to any high priority process appropriately.

Keywords: Fault Tolerant Control System, Redundant, PC-Based, Ethernet.

1. INTRODUCTION

Generally, the critical process is meant to the high priority process, for instance a dangerous process, a high price production process and so on that should not be shut down such process. Whenever the fault appears on this part, it will be affected to a whole system. Therefore, the assumption in this paper proposes to prevent and solve this problem as well as increase the reliability of the system. Our scheme consists of dual redundant supervisory control system which works as HMI (Human Machine Interface) station by two personal computers that connected to the dual redundant Ethernet switch including to three-PC based control system that works as the partial redundancy control system. The gas pressure control process is used as a case study in this paper.

2. REDUNDANT SYSTEM

A redundant system has design for increasing reliability of control system in order to prevent failure in the system but it also makes higher cost. The redundant system has several types such as cold standby (imperfect switching), hot standby (perfect switching), CPU - redundancy (partial redundancy), full dual redundancy and fault tolerant system for more than two modules of controller. Each type has different its advantage and disadvantage which appropriate with different level of high priority process and mean time to shutdown of each process.

2.1. Equations

The system consists two independent system connected in parallel as shown in figure 1.

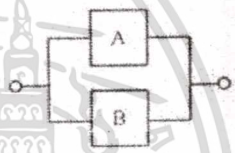


Figure 1. *Parallel system.*

For the parallel system, if system A or system B can be worked that means the system can be succeeded control. The system reliability can be obtain by

$$R_p = 1 - Q_A * Q_B \tag{1}$$

$$R_p = R_A + R_B - R_A * R_B \tag{2}$$

According to equation (1) and (2), if the numbers of parallel system increases that means reliability is increased but when the number of parallel system increased, it will be increasing cost and more maintenance required also.

2.2. Cold standby system (Imperfect Switching)

Cold standby system has one backup system that can switch from active system to standby system quickly when active system has some problem which make active system cannot control the process. This method can reduce time to change a spare part in order to maintenance the system. Because it has a spare part that

installed and waits to operation when active system has problems. The cost of this method is double and plus the cost of switching device. It is appropriate for longer.

2.3. Hot standby system (Perfect Switching)

Hot standby system is use widely in industry. It looks like cold standby system but the spare system is operated synchronization with active system and detects failure of active system. If active system failure, spare system will take over control process from active system automatically but the outputs of controllers need time to recover state. If recover time is more than permission down time then process must be shut down.

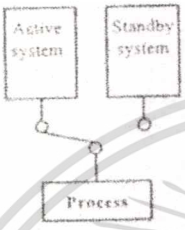


Figure 2. *Cold standby system*

2.4. Partial Redundancy system

In this paper, the Triple Modular Redundancy (TMR) is applied to control the gas pressure process. TMR uses for some processes that cannot have interrupt control because all components are active all time and the output is selected from output voter. The advantage of TMR fault tolerant system is uninterrupted control because all of controller are voting result of control data before send signal to control process therefore if one unit of controller fail, it has not affect to output signal. The reliability of TMR system can be obtained as following this equation.

$$R_{TMR} = \sum_{i=2}^{3} C_n^i R^i (1-R)^{n-i} \qquad (3)$$

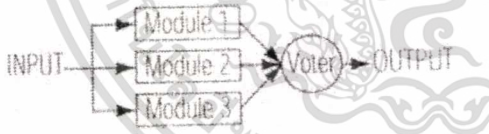


Figure 3. *Triple Modular Redundancy Configurations*

3. DISTRIBUTED REAL - TIME BASED ON ETHERNET

Historically, Ethernet [6] has been perceived as an unsuitable medium for real-time data distribution. The fear was that the fundamental access algorithm, CSMA/CD, and the popular transport protocol, TCP/IP, could not provide sufficiently consistent latency for deterministic applications. There are compelling reasons to utilize the Internet Protocol (IP) for real-time systems. IP is quickly becoming the lingua franca of the information age, reaching into every area of modern life. Designs using IP networking will be able to take advantage of this connectivity, for instance by monitoring and debugging systems remotely, collecting usage statistics, and even notifying service personnel of impending failures. However, many real-time application engineers are concerned by the fact that most IP systems currently utilize non-deterministic software and hardware. For instance, the most compelling implementations today use the Ethernet physical layer. Of course, IP can run over many media, and is in fact already becoming available on fast deterministic architectures such as FireWire (IEEE 1394). However, most hardware and software currently available for IP uses Ethernet hardware and Ethernet was not designed for fixed timing. Similarly, most networking software utilizes a transport protocol such as TCP. Like Ethernet, TCP was never designed for real-time operation. Instead, it provides reliable delivery-retrying dropped packets regardless of the delay incurred. Fortunately, these problems can be overcome.

3.1. Ethernet Access Algorithms

Most IP LANs use one of the flavors of the 802.3 Ethernet networking standard. Inexpensive, reliable hardware implementations, such as 10BaseT, are ubiquitous in offices and laboratories. Ethernet provides fast, efficient transport at either 10 or 100 Mbits/second. In real-time systems, Ethernet's utility is challenged because, in a busy environment, it does not provide a fixed time for nodes to access the network when multiple nodes try to access the network at the same time.

3.2. The Ethernet Network Redundancy

The weakness of star topology is the switching that will affect to a whole communication system when it fails. In this paper, we design the Dual Ethernet Network (DEN) in order to protect the communication system. The DEN connects to the HMI stations and the control stations which are PC-based control system. This part consists of two sets of Ethernet switch 10/100Mbs and each of computers has dual Ethernet Network Card that communicated over UTP media as illustrated in figure 4. The software to organize the data exchange is developed by Microsoft Visual C++ 6.0 based on windows 2000/XP operating system. Windows Socket API and CSocket Class including to MFC (Microsoft Foundation Classes) are employed in this system.

4. SYSTEM DESIGN STRUCTURE

In this article, the system structure is divided into four layers following as

- Supervisory Control System
- Dual Ethernet Network
- PC-based Control System
- Field devices input/output signal

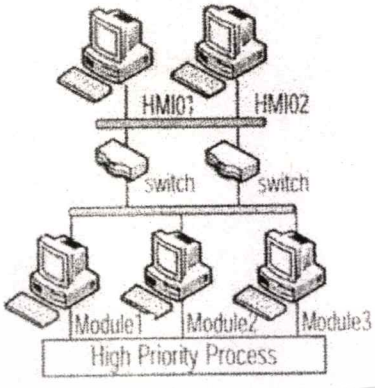


Figure 4. *Dual Ethernet Network system*

4.1. Supervisory Control System

This part is the HMI stations that each part consists of dual 10/100 Mbps Ethernet Card connecting to Ethernet network switch over UTP cable. For the part of software, there are the graphics user interface and the interfacing function that is developed by using Microsoft Visual C++ 6.0 The procedure of communication function follows as; the program will read parameters from the active PC-based controller through the Ethernet network as the same time it will check the status of PC-based controller for operating status is normally. If the program has found the failure on the controller it will change to get data from the other controllers. Such parameters are Access Name that is the address or name of the controller, and Topic that is the purpose of data exchange (read target / write target), and Item that is sub address or tag name of the component from the controller.

4.2. PC-Based Control System

The Programmable Logic Controller is widely used in industry because it is able to tolerate the industrial environment and there are many functions to support industrial applications such as PID, Fuzzy, and positioning control etc., however, the communication protocol is not opened and lacks to support the open system protocol. Presently, the PC-based control system for industrial is advance developed that can be applied to the industrial environment as well. Furthermore, it is designed base on the open architecture so that it is easy to connect to the other systems. For this reason, it brings about to design the control system based on PC. Each of PC-based PID controllers consists of analog input and analog output that sends and receives signal from the pressure transducer and control valve of the gas pressure process. The control parameters such as PID, set point and process monitoring can be done by PC-based controller. To design the redundant control system, there are similarly three control stations and each station can exchange data each other through the DEN, but only one station can be the active controller. When the active controller is failure with any reasons such as the communication failed, I/O failed or the controller failed then the other standby controllers will be appeared the active controller replacing the failed controller by voting method so that the highest performance controller will be awarded to the active controller. At the same time, the fault condition will be notified on the supervisory control system in order to repair such system.

5. THE EXPERIMENTAL RESULT

In this paper, the fault condition is defined for two cases, firstly the fault from hardware and second the fault from software. The fault condition can occur in all parts of a whole system so we separated it for each part for testing. Before testing the operation, the setting configuration is shown as figure 5.

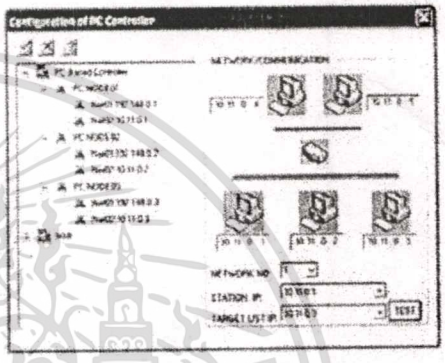


Figure 5a. *Configuration of PC-based controller.*

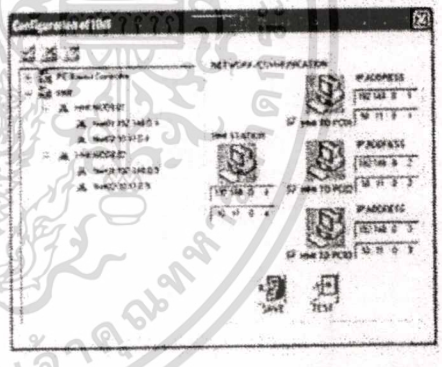


Figure 5b. *Configuration of HMI station.*

Firstly, the IP address has to set up for communicating among the computers. To check status of each station can be made by using Windows Socket API function which is a standard function in

window 32 bits operating system. Secondly, the other specifications are defined such as node name, application name and topic name. For example testing of operation, when the computer controller IP address 192.148.0.1 is working as the active controller has a fault condition then the dual HMI station will change reading data to the computer controller IP address 192.148.0.2. Similarly, when the active computer controller has a fault condition, the other standby controller will be operated as controller instead the failed controller.

Finally, to measure the maximum transfer time between controllers from active controller to standby controller by means turn off the power for 100 times. The result is illustrated on figure 7 that can be accepted for this control system. For the process with short delay time, it is necessary to analyze the processing time as well as the communication time because it is directly affected for controllability of that process.

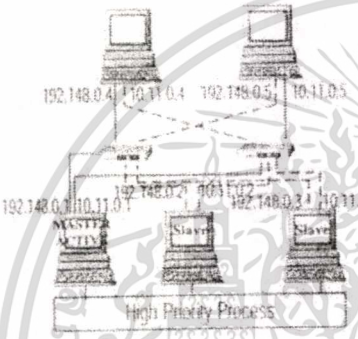


Figure 6a. *The normal operation.*

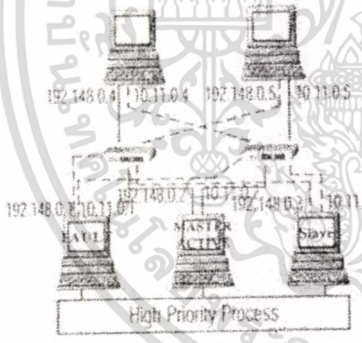


Figure 6b. *The fault condition.*

6. CONCLUSION

In this paper, the experiment of the gas pressure process control system via Dual Ethernet Network with Triple Module Redundancy control system can control the gas pressure process continuously although the PC-based controller has failed, other controllers can still instead the control task of this process as well as the HMI station can still monitor task even one of them have fault condition. The output signal can recover respond on range of permission down time of gas pressure process. If other processes have shorter permission down time than such process, it is necessary to consider the condition concerning the time response to improve the redundant system for control that process. Our system can increase the reliability as well as the risk of fault of a whole system and apply to any high priority process appropriately.

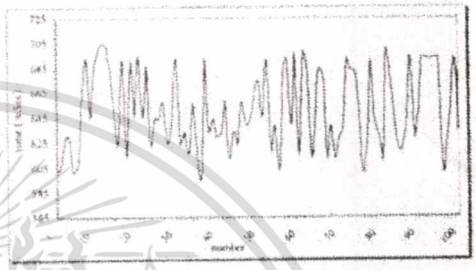


Figure 7. *Transferring time between changing condition*

7. REFERENCES

[1] R.Ramakumar., "Engineering Reliability: Fundamentals and Applications" Prentice-Hall International, Inc.1993.

[2] Shui-Ho Dai., and Ming - O Wang, "Reliability Analysis in Engineering Application" Van Nostrand Reinhold.1992.

[3] S.Vitturi., "PC-based automation system: an example of application for the real-time control of blowing machines" Computer Standards & Interfaces 26(2004).

[4] V Tipsuwanporn., A.Sangreyub., T.Sueaut., A.Numsomran and S.Gulphanich, "Development of PLC Fiber-optic Network for Redundant System", 2002 IEEE International Conference.

[5] Yixin Zhao, Feng Liu, "The implementation of a dual-redundant control system" Control Engineering Practice 12 (2004).

[6] http://www.rlr.com

ประวัติผู้เขียน

นายจตุพร รอดคำทุย เกิดเมื่อวันที่ 1 กรกฎาคม พ.ศ.2521 สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมการวัดคุม จากภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2544 เริ่มต้นทำงานที่บริษัทแซทจิเทท (ประเทศไทย) จำกัดในปีพ.ศ. 2545 ในตำแหน่ง วิศวกรระบบควบคุม และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมการวัดคุม ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2546