

อัลกอริทึมสำหรับการประสานสัมพันธรีเลย์ป้องกันกระแสเกิน
แบบมีทิศทางในระบบไฟฟ้ากำลังขนาดใหญ่



นายสุภาวัชร นิยมพันธุ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2557

**ALGORITHM FOR DIRECTIONAL OVER-CURRENT
RELAY COORDINATION IN LARGE-SCALE
POWER SYSTEMS**

Supawatchara Niyomphant



**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Electrical Engineering
Suranaree University of Technology**

Academic Year 2014

อัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสนิน
แบบมีทิศทางในระบบไฟฟ้ากำลังขนาดใหญ่

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(รศ. ดร.กิตติ อัครกิจมงคล)

ประธานกรรมการ

(รศ. ดร.ชนัดชัช กุลวรรณิขพงษ์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(ผศ. ดร.บุญเรือง มะรังศรี)

กรรมการ

(ศ. ดร.ชูกิจ ลิ้มปิจำนงค์)

รองอธิการบดีฝ่ายวิชาการและนวัตกรรม

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

ศุภวัชร นิยมพันธุ์ : อัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมี
ทิศทางในระบบไฟฟ้ากำลังขนาดใหญ่ (ALGORITHM FOR DIRECTIONAL OVER-
CURRENT RELAY COORDINATION IN LARGE-SCALE POWER SYSTEMS)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.ชนัดชัย กุลวรวานิชพงษ์, 159 หน้า.

งานวิจัยนี้ได้นำเสนออัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง เพื่อหาค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (Time Dial Setting : TDS) และเวลาที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบโดยใช้วิธีกำหนดการเชิงเส้น (Linear Programming : LP) ซึ่งเป็นกล่องเครื่องมือของโปรแกรมแมทแลปและนำมาใช้ในการแก้ปัญหาการประสานสัมพันธ์ของรีเลย์ในระบบทดสอบ 3 บัส 6 บัส 7 บัส 8 บัส 9 บัส 30 บัส 4700 บัส 8166 บัสและ14454 บัส นอกจากนี้ได้แสดงผลลัพธ์ของการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ ในระบบทดสอบ โดยนำโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface : GUI) มาช่วยในการแสดงผลลัพธ์ เพื่อเพิ่มความสะดวกในการแสดงผลการทำงานของรีเลย์หลักและรีเลย์สำรอง ผลจากการดำเนินงานในข้างต้นพบว่า สามารถหาค่าเวลาที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์และค่าจำนวนเท่าของเวลาการทำงานของรีเลย์แต่ละตัวในระบบทดสอบจากการใช้วิธีกำหนดการเชิงเส้นได้และนอกจากนี้ยังได้นำวิธีพัฒนาการโดยใช้ผลต่างมาใช้ในการแก้ปัญหาการประสานสัมพันธ์รีเลย์ร่วมด้วย เพื่อสนับสนุนและยืนยันความถูกต้องถึงผลลัพธ์ที่ได้ ซึ่งผลลัพธ์ที่ได้จากทั้งสองวิธีการที่นำเสนอให้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ใกล้เคียงกัน จึงสรุปได้ว่าอัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยการใช้วิธีกำหนดการเชิงเส้นในงานวิจัยนี้ สามารถนำมาใช้แก้ปัญหาในการประสานสัมพันธ์รีเลย์ในระบบทดสอบได้

SUPAWATCHARA NIYOMPHANT : ALGORITHM FOR DIRECTIONAL
OVER-CURRENT RELAY COORDINATION IN LARGE-SCALE POWER
SYSTEMS. THESIS ADVISOR : ASSOC. PROF. THANATCHAI KULWORA
WANICHPONG, Ph.D., 159 PP.

COORDINATION OF DIRECTIONAL OVER-CURRENT RELAY IN LARGE
SCALE POWER SYSTEMS

This research presents an algorithm for directional over-current relay coordination for large-scale power systems by using linear programming (LP) to determine the time dial setting (TDS) and for minimizing relay coordination time. In this research, linear programming, which is exploited from MATLAB optimization toolbox, is used to solve the relay coordination problem and to be tested with 3-bus, 6-bus, 7-bus, 8-bus, 9-bus, 30-bus, 4700-bus, 8166-bus and 14454-bus power systems. Graphical user interface (GUI) to display the primary and backup relay position is also implemented. In addition, differential evolution (DE) algorithm is selected to confirm the validity of the results. The results of the two methods are similar. It that the proposed algorithm for coordination of directional over-current relay by using linear programming in this research can be used effectively to solve directional over-current relay coordination in large-scale power systems.

School of Electrical Engineering

Academic Year 2014

Student's Signature _____

Advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี เนื่องจากได้รับความช่วยเหลืออย่างยิ่ง ทั้งด้านวิชาการและด้านการดำเนินงานวิจัย จากบุคคลและกลุ่มบุคคลต่าง ๆ ได้แก่

รองศาสตราจารย์ ดร.ชนัดชัย กุลวรรณิขพงษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้คำปรึกษา แนะนำ และแนะแนวทางอันเป็นประโยชน์ยิ่งต่องานวิจัย รวมถึงได้ช่วยตรวจทานและแก้ไขรายงานวิทยานิพนธ์เล่มนี้จนทำให้มีความสมบูรณ์ยิ่งขึ้น อีกทั้งเป็นกำลังใจและเป็นแบบอย่างที่ดีในการดำเนินชีวิตหลาย ๆ ด้านให้กับผู้วิจัยเสมอมา

ขอขอบคุณอาจารย์ ดร.อุเทน ถิณ ซึ่งให้คำแนะนำในการทำงานวิจัยและให้คำปรึกษาในเรื่องการเขียนบทความทางวิชาการ รวมถึงพี่ ๆ เพื่อน ๆ น้อง ๆ บัณฑิตศึกษาทุกท่านในกลุ่มงานวิจัยและมิตรสหายทั้งในอดีตและปัจจุบันที่คอยให้คำแนะนำและให้กำลังใจในการทำวิจัยมาโดยตลอด

สุดท้ายนี้ ผู้วิจัยขอขอบคุณอาจารย์ผู้สอนทุกท่านที่ประสิทธิ์ประสาทความรู้ทางด้านต่าง ๆ ทั้งในอดีตและปัจจุบัน สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์เล่มนี้ ผู้วิจัยขอมอบให้กับบิดา มารดา รวมถึงญาติพี่น้องของผู้วิจัยทุกท่าน ที่ให้ความรัก กำลังใจ การอบรมเลี้ยงดู และให้การสนับสนุนทางการศึกษาอย่างดียิ่งมาโดยตลอด จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา

ศุภวัชร นิยมพันธุ์

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย)	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ญ
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของเบื้องต้น	2
1.4 ขอบเขตของการวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 การจัดรูปเล่มวิทยานิพนธ์	3
2 ปรัชญ์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง	4
2.1 บทนำ	4
2.2 ปรัชญ์วรรณกรรม	4
2.3 สรุปผลการศึกษาศิลปะปรัชญ์วรรณกรรม	9
2.4 ทฤษฎีที่เกี่ยวข้อง	10
2.4.1 บทนำ	10
2.5 ทฤษฎีเบื้องต้นของวิธีป้องกันกระแสน้ำ	10
2.5.1 วิธีป้องกันกระแสน้ำ	10
2.5.2 วิธีป้องกันกระแสน้ำแบบมีทิศทาง	15
2.5.3 การปรับตั้งวิธีป้องกันกระแสน้ำแบบมีทิศทาง	17
2.5.4 ช่วงความถี่ (grading margin)	17

สารบัญ (ต่อ)

หน้า

2.6	ทฤษฎีของการลัดวงจรในระบบไฟฟ้า	18
2.6.1	การลัดวงจรแบบ 3 เฟส (three phase fault)	19
2.6.2	การลัดวงจรแบบ เฟส-ดิน (single line to ground fault)	22
2.6.3	การลัดวงจรแบบ เฟส-เฟส (line to line fault)	24
2.6.4	การลัดวงจรแบบ เฟส-เฟส-ดิน (double line to ground fault)	26
2.6.5	สถิติการเกิดลัดวงจร	29
2.7	กำหนดการเชิงเส้น (Linear Programming : LP)	31
2.7.1	รูปแบบทั่วไปของปัญหาคำหนดการเชิงเส้น	31
2.7.2	โครงสร้างของกำหนดการเชิงเส้น	32
2.7.3	รูปแบบของกำหนดการเชิงเส้นในกล่องเครื่องมือ ของโปรแกรมเมทแลป	32
2.8	โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface : GUI)	33
2.8.1	คุณสมบัติของวัตถุส่วนติดต่อผู้ใช้ (user interface object)	33
2.9	สรุป	35
3	การประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยใช้กำหนดการเชิงเส้น	37
3.1	บทนำ	37
3.2	การเตรียมข้อมูลของระบบทดสอบ	37
3.2.1	ขั้นตอนการเตรียมข้อมูลของระบบทดสอบ	38
3.3	การเขียนโปรแกรมและรันข้อมูล	39
3.3.1	รูปแบบสมการของปัญหา	39
3.3.2	ฟังก์ชันวัตถุประสงค์ของปัญหาการประสานสัมพันธ์ของ รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง	40
3.3.3	เงื่อนไขบังคับของปัญหาการประสานสัมพันธ์ของ รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง	40

สารบัญ (ต่อ)

หน้า

3.4	การระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง	41
3.5	โปรแกรมส่วนต่อประสานกราฟิกสำหรับแสดงผล	43
3.5.1	หน้าต่างการแสดงผลด้วยโปรแกรมส่วนต่อประสานกราฟิก.....	44
3.6	ภาพรวมของการสร้างอัลกอริทึมการประสานสัมพันธ์รีเลย์ป้องกัน กระแสนแบบมีทิศทาง	45
3.6.1	ขั้นตอนในการสร้างอัลกอริทึมการประสานสัมพันธ์รีเลย์	47
3.6.2	ลักษณะเด่นในการสร้างอัลกอริทึมการประสานสัมพันธ์รีเลย์ ของงานวิจัย.....	50
3.7	สรุป	51
4	ผลการทดสอบ	52
4.1	บทนำ.....	52
4.2	ผลการทดสอบของระบบทดสอบ	52
4.2.1	ระบบทดสอบ 3 บัส	52
4.2.2	ระบบทดสอบ 6 บัส	57
4.2.3	ระบบทดสอบ 7 บัส	62
4.2.4	ระบบทดสอบ 8 บัส	68
4.2.5	ระบบทดสอบ 9 บัส	73
4.2.6	ระบบทดสอบ 30 บัส	78
4.2.7	ระบบทดสอบ 4700 บัส	86
4.2.8	ระบบทดสอบ 8166 บัส	89
4.2.9	ระบบทดสอบ 14454 บัส	91
4.3	สรุป	93
5	สรุปและข้อเสนอแนะ	96
5.1	สรุป	96
5.2	ข้อเสนอแนะเพื่อพัฒนางานวิจัยในอนาคต.....	97

สารบัญ (ต่อ)

หน้า

รายการอ้างอิง	98
ภาคผนวก	
ภาคผนวก ก. ข้อมูลของระบบทดสอบ	101
ภาคผนวก ข. โปรแกรมสำหรับการหาผลลัพธ์ของงานวิจัยในวิทยานิพนธ์	111
ภาคผนวก ค. บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา	149
ประวัติผู้เขียน	159



สารบัญตาราง

ตารางที่	หน้า
2.1	ปรีทัศน์วรรณกรรม..... 4
2.2	ข้อมูลการเกิดลัดวงจรในระบบไฟฟ้า แบ่งตามรายอุปกรณ์ในระบบไฟฟ้า..... 29
2.3	สถิติการเกิดลัดวงจร..... 30
4.1	การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 3 บัส 54
4.2	ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ ของระบบทดสอบ 3 บัส..... 55
4.3	การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 6 บัส 58
4.4	ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ ของระบบทดสอบ 6 บัส..... 60
4.5	การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 7 บัส 64
4.6	ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ ของระบบทดสอบ 7 บัส..... 66
4.7	การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 8 บัส 70
4.8	ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ ของระบบทดสอบ 8 บัส..... 72
4.9	การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 9 บัส 75
4.10	ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ ของระบบทดสอบ 9 บัส..... 77
4.11	ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ ของระบบทดสอบ 30 บัส..... 82
4.12	ค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 4700 บัส 88
4.13	ค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 8166 บัส 90
4.14	ค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 14454 บัส 92
4.15	ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ..... 93

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
ก.1 ข้อมูลบัสของระบบทดสอบ 3 บัส.....	103
ก.2 ข้อมูลสายส่งของระบบทดสอบ 3 บัส	103
ก.3 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 3 บัส	103
ก.4 ข้อมูลบัสของระบบทดสอบ 6 บัส.....	104
ก.5 ข้อมูลสายส่งของระบบทดสอบ 6 บัส	104
ก.6 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 6 บัส	104
ก.7 ข้อมูลบัสของระบบทดสอบ 7 บัส.....	105
ก.8 ข้อมูลสายส่งของระบบทดสอบ 7 บัส	105
ก.9 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 7 บัส	106
ก.10 ข้อมูลบัสของระบบทดสอบ 8 บัส.....	106
ก.11 ข้อมูลสายส่งของระบบทดสอบ 8 บัส	106
ก.12 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 8 บัส	107
ก.13 ข้อมูลหม้อแปลงไฟฟ้าของระบบทดสอบ 8 บัส	107
ก.14 ข้อมูลบัสของระบบทดสอบ 9 บัส.....	107
ก.15 ข้อมูลสายส่งของระบบทดสอบ 9 บัส	108
ก.16 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 9 บัส	108
ก.17 ข้อมูลหม้อแปลงไฟฟ้าของระบบทดสอบ 9 บัส	108
ก.18 ข้อมูลบัสของระบบทดสอบ 30 บัส.....	109
ก.19 ข้อมูลสายส่งของระบบทดสอบ 30 บัส	109
ก.20 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 30 บัส	111
ข.1 รูปแบบเมทริกซ์ฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 3 บัส	113
ข.2 รูปแบบเมทริกซ์เงื่อนไขบังคับของระบบทดสอบ 3 บัส	114

สารบัญรูป

รูปที่	หน้า
2.1	คุณลักษณะการทำงานของรีเลย์กระแสนิน..... 11
2.2	เส้นโค้งเวลาและกระแสของรีเลย์..... 12
2.3	ตำแหน่งการเรียกชื่อของรีเลย์เมื่อเกิดความผิดปกติ..... 13
2.4	เส้นโค้งเวลาและกระแสของรีเลย์ป้องกันแบบมาตรฐาน 14
2.5	มุมเฟสของกระแสและแรงดัน 16
2.6	ระบบไฟฟ้ากำลังแบบใช้สายส่ง 2 เส้นขนานกัน 17
2.7	แผนภาพสถานะผิดปกติในระบบไฟฟ้า..... 19
2.8	การเกิดลัดวงจรแบบ 3 เฟส 20
2.9	วงจรของการเกิดลัดวงจรแบบ 3 เฟส 21
2.10	การเกิดลัดวงจรแบบ เฟส-ดิน 22
2.11	วงจรของการเกิดลัดวงจรแบบ เฟส-ดิน 23
2.12	การเกิดลัดวงจรแบบ เฟส-เฟส 24
2.13	วงจรของการเกิดลัดวงจรแบบ เฟส-เฟส 25
2.14	การเกิดลัดวงจรแบบ เฟส-เฟส-ดิน..... 26
2.15	วงจรของการเกิดลัดวงจรแบบ เฟส-เฟส-ดิน..... 28
2.16	ส่วนประกอบของ GUI 35
3.1	ขั้นตอนการเตรียมข้อมูลของระบบทดสอบ 38
3.2	ขั้นตอนการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง 43
3.3	ออกแบบการแสดงผลด้วย GUI 45
3.4	ภาพรวมของอัลกอริทึมการประสานสัมพันธ์รีเลย์ป้องกันกระแสนินแบบมีทิศทาง 46
4.1	ระบบทดสอบ 3 บัส 53
4.2	แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 3 บัส 54
4.3	แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 3 บัส 55

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง ของระบบทดสอบ 3 บัต์.....	56
4.5 ระบบทดสอบ 6 บัต์	57
4.6 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 6 บัต์	58
4.7 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 6 บัต์	60
4.8 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง ของระบบทดสอบ 6 บัต์.....	62
4.9 ระบบทดสอบ 7 บัต์	63
4.10 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 7 บัต์	64
4.11 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 7 บัต์	66
4.12 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง ของระบบทดสอบ 7 บัต์.....	68
4.13 ระบบทดสอบ 8 บัต์	69
4.14 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 8 บัต์	70
4.15 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 8 บัต์	71
4.16 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง ของระบบทดสอบ 8 บัต์.....	73
4.17 ระบบทดสอบ 9 บัต์	74
4.18 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 9 บัต์	75
4.19 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 9 บัต์	76
4.20 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง ของระบบทดสอบ 9 บัต์.....	78
4.21 ระบบทดสอบ 30 บัต์	79
4.22 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 30 บัต์	80
4.23 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 30 บัต์	81

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.24 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่าง ของระบบทดสอบ 30 บัต์.....	86
4.25 ระบบทดสอบ 4700 บัต์	87
4.26 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 4700 บัต์	88
4.27 ระบบทดสอบ 8166 บัต์	89
4.28 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 8166 บัต์	90
4.29 ระบบทดสอบ 14454 บัต์	91
4.30 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 14454 บัต์	92



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบไฟฟ้ากำลังประกอบด้วย 3 ระบบคือ ระบบผลิต ระบบส่งจ่ายและระบบจำหน่าย ซึ่งทั้ง 3 ระบบนี้มีความสำคัญทุกระบบเนื่องจากระบบผลิตนั้นเป็นแหล่งกำเนิดกระแสไฟฟ้าซึ่งผลิตรมาจากทรัพยากรธรรมชาติเช่น ถ่านหิน ก๊าซธรรมชาติ หินน้ำมัน เป็นต้น จากนั้นจะส่งไปยังระบบส่งจ่ายผ่านสายส่งไฟฟ้าแรงสูงไปยังสถานีไฟฟ้าย่อยที่อยู่ในแต่ละภูมิภาคนั้น ๆ และส่งต่อผ่านระบบจำหน่ายไปยังกลุ่มผู้ใช้ไฟฟ้า การวิเคราะห์ระบบไฟฟ้ากำลังจึงเป็นสิ่งสำคัญ เช่น การศึกษาการไหลของกำลังไฟฟ้าในสภาวะการทำงานปกติ จะทำให้ทราบคุณลักษณะสมบัติของระบบไฟฟ้าภายใต้การทำงานในสภาวะปกติ ทำให้การจ่ายโหลดถูกจำกัดอยู่ที่ค่าพิกัดที่กำหนดได้ ถ้าโหลดเพิ่มขึ้นอย่างต่อเนื่องจนเกินค่าพิกัดแล้ว ระบบไฟฟ้ากำลังต้องมีอุปกรณ์ที่ทำหน้าที่ตรวจสอบกระแสไฟฟ้าหรือกำลังไฟฟ้าที่ไหลผ่านอุปกรณ์ดังกล่าวได้และอาจจะยอมให้เกิดการทำงานในสภาวะโหลดเกินได้ในช่วงเวลาสั้น ๆ ค่าหนึ่ง ถ้าเกินเวลาที่กำหนดแล้วโหลดเกินยังอยู่ อุปกรณ์ป้องกันต้องทำการปลดวงจรในส่วนนั้นออกเพื่อป้องกันการเกิดความเสียหายกับอุปกรณ์นั้น ๆ โดยจะเรียกการป้องกันในลักษณะนี้ว่า การป้องกันโหลดเกิน (overload protection) การวิเคราะห์การลัดวงจรหรือความผิดปกติ จะทำให้ทราบว่าเมื่อเกิดการลัดวงจรขึ้นจะมีกระแสไหลในสายส่งเส้นต่าง ๆ เป็นปริมาณเท่าใด อุปกรณ์ป้องกันต้องตรวจจับกระแสลัดวงจรนี้ได้และในขณะเดียวกันต้องทำหน้าที่ตัดส่วนของระบบที่เกิดการลัดวงจรนี้ออกไป โดยส่วนที่ไม่เกิดการลัดวงจรต้องยังคงสามารถทำงานได้ตามปกติ เรียกการป้องกันในลักษณะนี้ว่า การป้องกันการลัดวงจร (short-circuit protection) (ธนิตชัย กุลวรวานิชพงษ์, 2552)

ดังนั้น ในระบบไฟฟ้ากำลังจึงต้องมีระบบป้องกันเพื่อป้องกันความเสียหายที่จะเกิดขึ้น ในงานวิจัยนี้จึงได้ทำการศึกษาวิธีเลย์ป้องกันกระแสเกินแบบมีทิศทาง เนื่องจากวิธีชนิดนี้มีความสำคัญในการป้องกันสายส่งในระบบส่งจ่ายกำลังไฟฟ้า เพื่อที่จะแก้ปัญหาการประสานสัมพันธ์วิธีเลย์ป้องกันกระแสเกินแบบมีทิศทาง ได้ใช้วิธีการของกำหนดการเชิงเส้น (ธนิตชัย กุลวรวานิชพงษ์, 2550) เนื่องจากเป็นวิธีการที่มีรูปแบบเข้าใจง่าย ไม่ซับซ้อนและใช้เวลาในการหาผลลัพธ์ที่ค่อนข้างรวดเร็วกว่าเทคนิคขาคู่ลำดับอื่น ๆ แต่มีข้อจำกัดในเรื่องของเงื่อนไขบังคับที่อยู่

ในรูปแบบของฟังก์ชันเชิงเส้นเท่านั้น จึงได้เลือกใช้วิธีกำหนดการเชิงเส้นมาแก้ปัญหาในงานวิจัยนี้ เพื่อให้รีเลย์สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพในการป้องกันระบบไฟฟ้ากำลัง

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อนำวิธีกำหนดการเชิงเส้นมาใช้ในการแก้ปัญหาการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบทดสอบ

1.2.2 เพื่อระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ

1.2.3 เพื่อนำโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) มาใช้ในการแสดงผลลัพธ์ของการประสานสัมพันธ์รีเลย์ ผลลัพธ์ของการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ

1.3 ข้อตกลงเบื้องต้น

1.3.1 ระบบทดสอบที่ใช้เป็นระบบทดสอบของโปรแกรม Power World Simulator ประกอบด้วยระบบทดสอบ 3 บัส ระบบทดสอบ 6 บัส ระบบทดสอบ 7 บัส ระบบทดสอบ 8 บัส ระบบทดสอบ 9 บัส ระบบทดสอบ 30 บัส ระบบทดสอบ 4700 บัส ระบบทดสอบ 8166 บัสและระบบทดสอบ 14454 บัส

1.3.2 พิจารณาการเกิดลัดวงจรแบบสามเฟสสมมาตร

1.3.3 รีเลย์ที่ใช้เป็นรีเลย์ชนิดดิจิทัลรีเลย์

1.3.4 ใช้โปรแกรมเมทแลปในการหาผลลัพธ์ของการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์และค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุด

1.4 ขอบเขตของการวิจัย

1.4.1 ศึกษาและแก้ปัญหาการประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้นในระบบทดสอบ 3 บัส ระบบทดสอบ 6 บัส ระบบทดสอบ 7 บัส ระบบทดสอบ 8 บัส ระบบทดสอบ 9 บัส ระบบทดสอบ 30 บัส ระบบทดสอบ 4700 บัส ระบบทดสอบ 8166 บัสและระบบทดสอบ 14454 บัส

1.4.2 ศึกษาและแก้ปัญหาคำแนะนำรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ ได้แก่ ระบบทดสอบ 3 บัส ระบบทดสอบ 6 บัส ระบบทดสอบ 7 บัส ระบบทดสอบ 8 บัส ระบบทดสอบ 9 บัส ระบบทดสอบ 30 บัส ระบบทดสอบ 4700 บัส ระบบทดสอบ 8166 บัสและระบบทดสอบ 14454 บัส

1.4.3 ศึกษาและนำโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้(GUI) มาใช้ในการแสดงผลการทำงานของระบบคำแนะนำรีเลย์หลักและรีเลย์สำรอง ค่าจำนวนค่าของเวลาการทำงานของรีเลย์และค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้ความรู้ด้านการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้น

1.5.2 ได้ทราบตำแหน่งของรีเลย์หลักและสำรองในระบบทดสอบเมื่อเกิดการลัดวงจร ณ บัสใด ๆ ในระบบทดสอบ

1.5.3 ได้ความรู้ด้านการเขียนโปรแกรมแมทเพลและส่วนต่อประสานกราฟิกกับผู้ใช้

1.5.4 นำความรู้ที่ได้ไปประยุกต์ใช้กับระบบที่ซับซ้อนขึ้นหรือระบบที่ใช้งานจริงได้

1.5.5 ได้บทความวิจัยเผยแพร่ระดับชาติ

1.6 การจัดรูปเล่มวิทยานิพนธ์

วิทยานิพนธ์นี้ประกอบด้วย 5 บท ซึ่งในแต่ละบทได้นำเสนอดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ ข้อตกลงเบื้องต้น ขอบเขตของงานวิจัยวิทยานิพนธ์และประโยชน์ที่คาดว่าจะได้รับของงานวิจัย

บทที่ 2 กล่าวถึงปริทัศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้องกับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลัง

บทที่ 3 การประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยใช้วิธีกำหนดการเชิงเส้น

บทที่ 4 ผลการทดสอบ

บทที่ 5 บทสรุปและข้อเสนอแนะ

ภาคผนวก ก. ข้อมูลของระบบทดสอบต่าง ๆ

ภาคผนวก ข. โปรแกรมสำหรับการหาผลลัพธ์ของงานวิจัยในวิทยานิพนธ์

ภาคผนวก ค. บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

บทที่ 2

ปริทัศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

2.1 บทนำ

งานวิจัยนี้มีวัตถุประสงค์หลักคือ การประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลังขนาดใหญ่ โดยใช้วิธีกำหนดการเชิงเส้นในการแก้ปัญหา ซึ่งผลลัพธ์ที่ต้องการคือ ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (Time Dial Setting : TDS) และค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุด ส่วนวัตถุประสงค์รองคือการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบทดสอบโดยใช้โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) สำหรับใช้แสดงผลตำแหน่งของรีเลย์หลักและรีเลย์สำรองเมื่อเกิดความผิดปกติ ณ ตำแหน่งบัสใด ๆ ในระบบทดสอบ ในส่วนนี้จะนำเสนอการสำรวจปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง โดยได้ใช้ฐานข้อมูลที่เป็นแหล่งสะสมงานวิจัยและวรรณกรรมที่เกี่ยวข้องทางด้านวิศวกรรมศาสตร์และเทคโนโลยี เช่น ฐานข้อมูลของ IEEE และ Science direct เป็นต้น ซึ่งในแต่ละบทความวิจัย ผู้วิจัยวิทยานิพนธ์จะนำเสนอโดยเรียงตามลำดับปีที่ตีพิมพ์ รวมถึงอธิบายสาระสำคัญของแต่ละงานวิจัยไว้พอสังเขป

2.2 ปริทัศน์วรรณกรรม

จากการศึกษาปริทัศน์วรรณกรรมที่เกี่ยวข้อง พบว่าได้มีผู้วิจัยนำเสนองานวิจัยทางการประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง สามารถสรุปสาระสำคัญที่ได้จากงานวิจัยในแต่ละบทความตาม que ผู้วิจัยได้รวบรวมไว้ในตารางที่ 2.1 ดังนี้

ตารางที่ 2.1 ปริทัศน์วรรณกรรม

ปีที่ตีพิมพ์ (ค.ศ.)	คณะผู้วิจัย	สาระสำคัญของงานวิจัย
1988	Urdaneta, A.J. Nadira, R. and Perez Jimenes, L.G.	นำเสนอการใช้วิธีการหาค่าเหมาะสมที่สุดโดยตรง ซึ่งใช้การวนรอบเพื่อแก้ปัญหการประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบทดสอบขนาด 3 บัส โดยมีข้อจำกัดต่าง ๆ เพื่อหาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุด

ตารางที่ 2.1 ทัศนัวรรณกรรม (ต่อ)

ปีที่ตีพิมพ์ (ค.ศ.)	คณะผู้วิจัย	สาระสำคัญของงานวิจัย
1990	Ramaswami, R. Damborg, M.J. and Venkata, S.	นำเสนอการสร้างอัลกอริทึมการประสานสัมพันธ์ของรีเลย์ ป้องกันกระแสเกินแบบมีทิศทางในระบบส่งจ่ายกำลังไฟฟ้า ย่อยซึ่งตอบสนองต่อการเปลี่ยนแปลงโครงสร้างและโหลด ของระบบ โดยการหาค่าที่เหมาะสมสำหรับการปรับตั้งค่าการ ทำงานของรีเลย์ด้วยโปรแกรมทางคอมพิวเตอร์ ซึ่งอัลกอริทึม ที่สร้างขึ้นนี้ถูกพัฒนาให้มีความเหมาะสมกับระบบไฟฟ้า กำลังที่มีการเปลี่ยนแปลงของโครงสร้างหรือโหลดในระบบ เพื่อการป้องกันที่มีประสิทธิภาพมากยิ่งขึ้น
1993	Elrafie, H.B. and Irving, M.R.	นำเสนอวิธีการลดข้อจำกัดของวิธีกำหนดการเชิงเส้น โดย นำมาประยุกต์ใช้ในการแก้ปัญหาการประสานสัมพันธ์ของ รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบทดสอบขนาด 6 บัส ซึ่งหาค่าจำนวนเท่าของเวลาการทำงานของรีเลย์และ การเลือกค่ากระแสปรับตั้ง ผลลัพธ์ที่ได้เมื่อเปรียบเทียบกับ ผลลัพธ์ของวิธี break-point ซึ่งจะให้ผลลัพธ์ที่สอดคล้องตาม วัตถุประสงค์ที่ต้องการ สรุปได้ว่าวิธีการลดข้อจำกัดนี้เป็น วิธีการหนึ่งที่สามารถใช้ในการคำนวณหาค่าจำนวนเท่าของ เวลาการทำงานของรีเลย์ได้
1994	Perez L.G.	นำเสนอการใช้โปรแกรมทางคอมพิวเตอร์(CUPL) ซึ่งเป็น โปรแกรมภาษาระดับสูงที่ให้ผู้ใช้งานเข้าถึงฐานข้อมูลได้โดยตรง และสามารถใช้อนุบัติการทางคณิตศาสตร์เพื่อหาผลลัพธ์ที่ ต้องการได้ ผลจากการใช้โปรแกรมทางคอมพิวเตอร์นี้จะทำ ให้ทราบข้อมูลเชิงกราฟของรีเลย์ ซึ่งสามารถนำข้อมูลต่าง ๆ ที่ ได้มาวิเคราะห์เพื่อหาผลลัพธ์ที่เหมาะสมและนำผลลัพธ์ที่ได้ ไปปรับตั้งค่าการทำงานของรีเลย์ในระบบทดสอบและยังทำ ให้ทราบถึงหน้าที่การทำงานของรีเลย์ในระบบทดสอบด้วยว่า มีรีเลย์ตัวใดบ้างที่ทำหน้าที่เป็นรีเลย์หลักและรีเลย์สำรอง

ตารางที่ 2.1 ทัศนวิจารณ์กรรม (ต่อ)

ปีที่ตีพิมพ์ (ค.ศ.)	คณะผู้วิจัย	สาระสำคัญของงานวิจัย
1996	Urdaneta, A.J. Restrepo, H. Marquez, S. and Sanchez, J.	นำเสนอวิธีการกำหนดการเชิงเส้น ซึ่งนำมาใช้กับโครงสร้างของระบบทดสอบที่มีการเปลี่ยนแปลงได้ ซึ่งการแก้ปัญหาส่วนใหญ่เพื่อต้องการทราบค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางและค่ากระแสปรับตั้ง (pickup current : I_p) ของรีเลย์ในระบบทดสอบที่เหมาะสมในแต่ละสถานะการเปลี่ยนแปลงของระบบทดสอบ
1997	Kawahara, K. Sasaki, H. and Sugihara, H.	นำเสนอการแก้ปัญหาในเรื่องการปรับตั้งค่าการทำงานและการหน่วงเวลาของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง เริ่มจากการนำเสนอในเรื่องของวิธีการอย่างง่ายในการประสานสัมพันธ์ของรีเลย์โดยใช้เงื่อนไขบังคับในการแก้ปัญหาและตรวจสอบคู่การทำงานของรีเลย์หลักและรีเลย์สำรองในระบบหาค่าการปรับตั้งการทำงานของรีเลย์ โดยทำการเลือกค่าในการปรับตั้งการทำงานของรีเลย์คือ ค่า break point set (BPS) โดยใช้โปรแกรมชื่อว่า OPS83 เพื่อแก้ปัญหาและประยุกต์ใช้กับระบบทดสอบอย่างง่ายเพื่อทำการตรวจสอบประสิทธิภาพในการทำงานของรีเลย์หลังจากทำการปรับตั้งค่าการทำงาน of รีเลย์แล้ว
1998	Perez, L.G. Urdaneta, A.J. Sorrentino, E. Garayar, F. Urizar, A. Ledezma, J. C. Alcala, G.D.	นำเสนอการเปรียบเทียบเวลาในการทำงานของรีเลย์ระยะทาง ซึ่งเป็นรีเลย์หลักและรีเลย์ป้องกันกระแสเกินแบบมีทิศทางซึ่งเป็นรีเลย์สำรองซึ่งทดสอบกับระบบไฟฟ้ากำลังขนาด 69 kV แก้ไขปัญหาการประสานสัมพันธ์ของรีเลย์โดยใช้วิธีการกำหนดการเชิงเส้น ซึ่งแบ่งเป็น 3 กรณีสำหรับการพิจารณา คือ 1.การประสานสัมพันธ์แบบอิสระของรีเลย์ชนิดผกผันกับเวลาและรีเลย์ระยะทาง

ตารางที่ 2.1 ทัศนัวรรณกรรม (ต่อ)

ปีที่ตีพิมพ์ (ค.ศ.)	คณะผู้วิจัย	สาระสำคัญของงานวิจัย
	Carrion, N. Canache, C. Fernandez, J. Sanz, O. and Guevara, F.	2.การประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางชนิดผกผันกับเวลาและรีเลย์ระยะทาง 3.การประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางชนิดจำกัดเวลาและกระแส หลังจากทำการทดสอบสามารถสรุปผลทั้ง 3 กรณีได้ว่า ในแต่ละกรณีนั้นมีข้อดีและข้อเสียต่างกัน ทั้งนี้ขึ้นอยู่กับกรณีของการนำไปใช้งานในลักษณะเฉพาะของแต่ละพื้นที่และรูปแบบโครงสร้างของระบบไฟฟ้ากำลัง
2001	Urdaneta, A.J. Perez, L.G. Gomez, J.F. Feijoo, B. and Gonzalez, M.	นำเสนอวิธีการกำหนดการเชิงเส้นสำหรับการแก้ปัญหาการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลัง ระดับแรงดัน 69-115 kV ซึ่งปรับตั้งกราวด์รีเลย์และเฟสรีเลย์ โดยการนำเทคนิคการแก้ปัญหาแบบ pre-resolution มาประยุกต์ใช้ ซึ่งช่วยลดขนาดและความซับซ้อนของปัญหาจากการใช้วิธีการกำหนดการเชิงเส้นเพื่อแก้ปัญหาโดยใช้การวนรอบของวิธีการเพื่อหาผลลัพธ์ที่เหมาะสมที่สุด
2004	Jamali, S. and Pourtandorost, M.	นำเสนอวิธีการใหม่สำหรับการประสานสัมพันธ์ระหว่างรีเลย์ระยะทางและรีเลย์กระแสเกิน โดยใช้วิธีการกำหนดการเชิงเส้นซึ่งมีวัตถุประสงค์เพื่อปรับตั้งค่าจำนวนเท่าของเวลาการทำงานของรีเลย์หลักและรีเลย์สำรองในระบบ โดยมีข้อจำกัดที่เหมาะสมสำหรับการประสานสัมพันธ์รีเลย์ในระบบไฟฟ้า
2006	Giovanini, R. Hopkinson, K. Coury, D.V. and Thorp, J.S.	นำเสนอผลจากการศึกษาในฐานระบบที่มีพื้นที่ขนาดใหญ่เกี่ยวกับระบบการสื่อสารผ่านระบบอินทราเน็ต(intranet) สำหรับการประสานสัมพันธ์รีเลย์ของการป้องกันหลักและการป้องกันสำรองของรีเลย์ โดยใช้แบบจำลองเพื่อแสดงให้เห็นถึงวิธีการและผลในการใช้เครื่องมือที่เรียกว่า EPOCHS

ตารางที่ 2.1 ทัศนัวรรณกรรม (ต่อ)

ปีที่ตีพิมพ์ (ค.ศ.)	คณะผู้วิจัย	สาระสำคัญของงานวิจัย
		(Electric Power and Communication Synchronizing Simulation) ประกอบด้วยโปรแกรม EMTDC/PSCAD และระบบเครือข่ายของ NS2 สำหรับการจำลองผลของการสื่อสารแบบระยะไกลในระบบเครือข่าย โดยใช้ฐานข้อมูลในการตรวจสอบความผิดปกติและความล้มเหลวในการทำงานของเซอร์กิตเบรกเกอร์ที่เกิดขึ้น เพื่อการประสานสัมพันธ์ของการป้องกันหลักและการป้องกันสำรองของรีเลย์ ซึ่งวิธีการนี้เป็นวิธีที่ค่อนข้างซับซ้อน แต่มีประสิทธิภาพในการประสานสัมพันธ์ของรีเลย์มากขึ้น
2008	Ezzeddine, M. and Kaczmarek, R.	นำเสนอวิธีการลดข้อจำกัดของการประสานสัมพันธ์ของรีเลย์ โดยการใช้วิธีกำหนดการเชิงเส้นและการเลือกค่ากระแสปรับตั้ง ซึ่งทดสอบในระบบทดสอบขนาด 8 บัสและระบบทดสอบขนาด 14 บัสของมาตรฐาน IEEE ซึ่งแบ่งปัญหาออกเป็น 3 กรณี คือ 1.ใช้วิธีการของกำหนดการเชิงเส้นโดยไม่พิจารณาข้อจำกัดของเวลา 2.ใช้วิธีการแบบไม่เป็นเชิงเส้น 3.ใช้วิธีการของกำหนดการเชิงเส้นและลดข้อจำกัดของการประสานสัมพันธ์ของรีเลย์ จากผลการทดสอบทั้ง 3 กรณี และนำผลลัพธ์มาเปรียบเทียบกัน พบว่าผลลัพธ์ที่ได้จากการใช้วิธีกำหนดการเชิงเส้นในกรณีที่ 3 ให้ผลลัพธ์ตรงกับความต้องการและฟังก์ชันวัตถุประสงค์ที่กำหนดไว้
2009	Bedekar, P.P. Bhide, S.R. and Kale, V.S.	นำเสนอการใช้รีเลย์ป้องกันกระแสเกินเป็นอุปกรณ์ป้องกันในระบบจำหน่ายไฟฟ้ากำลังแบบวงแหวน การปรับตั้งค่าการทำงานของรีเลย์ในการประสานสัมพันธ์ของรีเลย์ที่เป็นรีเลย์หลักและรีเลย์สำรอง โดยการใช้วิธีการทางคณิตศาสตร์ที่เรียกว่า กำหนดการเชิงเส้น ซึ่งเป็นกล่องเครื่องมือ (toolbox)

ตารางที่ 2.1 ปฏิทัศน์วรรณกรรม (ต่อ)

ปีที่ตีพิมพ์ (ค.ศ.)	คณะผู้วิจัย	สาระสำคัญของงานวิจัย
		ในโปรแกรมแมทแลปมาใช้ในการแก้ปัญหาเนื่องจากเป็นโปรแกรมที่นิยมใช้กันอย่างกว้างขวางสำหรับการแก้ปัญหาและจำลองผลทางคอมพิวเตอร์ อีกทั้งมีการพัฒนาโปรแกรมให้มีความเหมาะสมในการใช้งานอย่างง่าย จึงได้ใช้โปรแกรมนี้เพื่อการแก้ปัญหการประสานสัมพันธ์รีเลย์

2.3 สรุปผลการศึกษาปฏิทัศน์วรรณกรรม

ปฏิทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องที่ได้นำเสนอในส่วนนี้ เป็นผลงานวิจัยที่เกี่ยวข้องในเรื่องของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยใช้วิธีกำหนดการเชิงเส้นในการแก้ปัญหาและการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบไฟฟ้ากำลัง มีการทดสอบกับระบบทดสอบขนาด 3 บัส 6 บัส 8 บัสและ 14 บัส กระทำภายใต้เงื่อนไขบังคับที่แตกต่างกันไปขึ้นอยู่กับวัตถุประสงค์ที่ต้องการและรูปแบบของระบบทดสอบ ซึ่งผลงานวิจัยที่เกี่ยวข้องต่าง ๆ ในข้างต้น ถือเป็นพื้นฐานที่สำคัญอย่างยิ่งต่อผู้วิจัยสำหรับการทำวิจัยวิทยานิพนธ์และเพื่อเป็นการทดสอบประสิทธิภาพของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางของวิธีกำหนดการเชิงเส้น งานวิจัยนี้จึงได้สร้างอัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้นซึ่งเป็นกล่องเครื่องมือของโปรแกรมแมทแลปมาใช้ในการแก้ปัญหาและทดสอบในระบบทดสอบขนาด 3 บัส 6 บัส 7 บัส 8 บัส 9 บัส 30 บัส 4700 บัส 8166 บัสและ 14454 บัส เพื่อหาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ด้วยเงื่อนไขข้อบังคับที่จะกล่าวต่อไปในบทที่ 3 และเพื่อเป็นการยืนยันถึงความถูกต้องของอัลกอริทึมที่สร้างขึ้น จึงนำวิธีการของวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution : DE) ที่พัฒนามาจากวิธีจินเนติกอัลกอริทึม (Genetic Algorithms : GA) ซึ่งเป็นเทคนิคชาญฉลาดที่ใช้งานกันอย่างแพร่หลาย มาใช้ในการแก้ปัญหาเพื่อเปรียบเทียบค่าของผลลัพธ์ที่ได้ ซึ่งรายละเอียดดังกล่าวข้างต้นจะกล่าวถึงในบทต่อไป ในงานวิทยานิพนธ์เล่มนี้

2.4 ทฤษฎีที่เกี่ยวข้อง

2.4.1 บทนำ

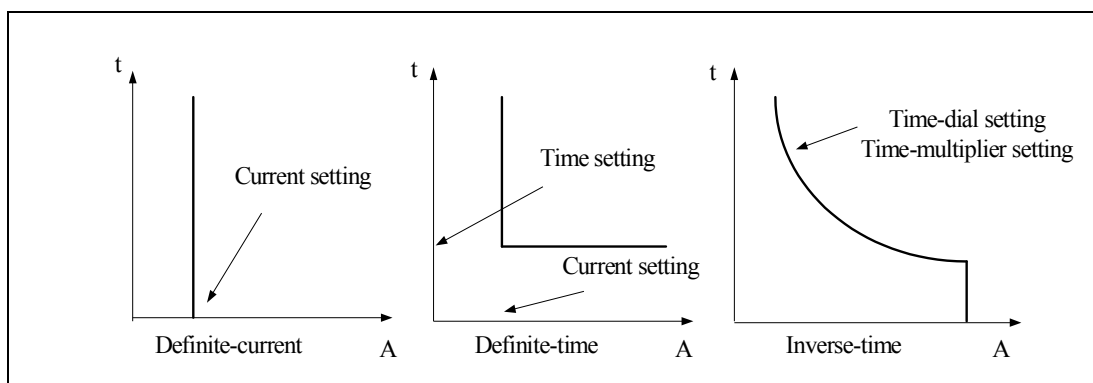
รีเลย์ป้องกันกระแสเกินแบบมีทิศทางเป็นอุปกรณ์ที่สำคัญในการป้องกันระบบไฟฟ้ากำลัง ทำหน้าที่ในการป้องกันความผิดปกติที่เกิดขึ้นในระบบไฟฟ้ากำลัง เช่น โหลดเกิน การลัดวงจร เป็นต้น ซึ่งการป้องกันความผิดปกติเหล่านี้ต้องอาศัยการทำงานร่วมกันของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางอย่างเหมาะสมและมีประสิทธิภาพ เพื่อป้องกันความเสียหายที่อาจเกิดขึ้นกับอุปกรณ์ไฟฟ้าและเพื่อให้เกิดความเชื่อถือได้ในระบบ โดยรีเลย์แต่ละตัวจะมีการปรับตั้งค่า TDS เพื่อให้รีเลย์สามารถทำงานได้อย่างเหมาะสมทั้งในสภาวะปกติและสภาวะการเกิดความผิดปกติต่าง ๆ ซึ่งการปรับตั้งค่า TDS ในปัจจุบันนี้มีหลายวิธีที่มีการพัฒนาและมีวิธีการมากมายเพื่อแก้ปัญหา ในงานวิจัยนี้ได้เลือกใช้วิธีกำหนดการเชิงเส้นซึ่งเป็นวิธีหนึ่งในการแก้ปัญหา วิธีกำหนดการเชิงเส้นเป็นวิธีพื้นฐานในการแก้ปัญหาเรื่องการประสานสัมพันธ์รีเลย์ อีกทั้งมีขั้นตอนที่เข้าใจง่าย ไม่ซับซ้อน และได้ผลลัพธ์ที่ค่อนข้างรวดเร็วกว่าเทคนิคขาคณิตศาสตร์อื่น ๆ จึงนำวิธีกำหนดการเชิงเส้นมาใช้แก้ปัญหาในการหาค่า TDS ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางแต่ละตัวที่อยู่ในระบบไฟฟ้ากำลัง พร้อมทั้งระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในการป้องกันการลัดวงจรที่เกิดขึ้นในระบบไฟฟ้ากำลังโดยใช้โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) ช่วยในการแสดงผลลัพธ์ดังที่กล่าวข้างต้น

ในส่วนนี้จะกล่าวถึงทฤษฎีเบื้องต้นของรีเลย์ป้องกันกระแสเกิน การลัดวงจรในรูปแบบต่าง ๆ วิธีกำหนดการเชิงเส้นและโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

2.5 ทฤษฎีเบื้องต้นของรีเลย์ป้องกันกระแสเกิน

2.5.1 รีเลย์ป้องกันกระแสเกิน

รีเลย์ป้องกันกระแสเกิน (over-current relay : o/c relay) แบ่งตามคุณลักษณะการทำงานได้ 3 ชนิดได้แก่ รีเลย์กระแสเกินชนิดจำกัดกระแส (definite-current o/c relay) รีเลย์กระแสเกินชนิดเวลาจำกัด (definite-time o/c relay) และรีเลย์กระแสเกินชนิดเวลาผกผัน (inverse-time o/c relay) รีเลย์กระแสเกินแต่ละชนิดมีกราฟคุณลักษณะของการทำงานที่ต่างกันดังรูปที่ 2.1 (ธนัชกุลวรรณิพงษ์, 2552)



รูปที่ 2.1 คุณลักษณะการทำงานของรีเลย์กระแสเกิน

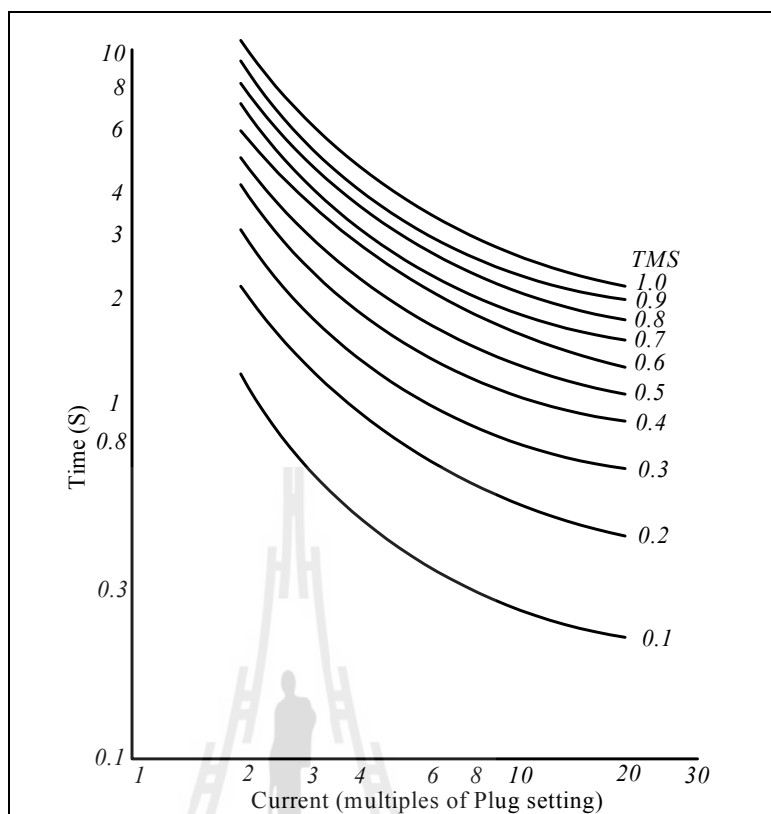
รีเลย์กระแสเกินชนิดจำกัดกระแสเป็นรีเลย์ที่ทำงานทันทีทันทีใดเมื่อมีกระแสไหลเกินค่าเริ่มต้นทำงานของรีเลย์ โดยปกติรีเลย์ชนิดนี้มักใช้งานร่วมกับรีเลย์กระแสเกินชนิดอื่น ๆ เช่น การทำงานร่วมกับรีเลย์กระแสเกินชนิดเวลาจำกัด เนื่องจากรีเลย์กระแสเกินชนิดเวลาจำกัดนี้จะมีช่วงเวลาทำงานของรีเลย์ที่แตกต่างกัน เพื่อลดปัญหาในเรื่องการทำงานพร้อมกันของรีเลย์ แต่การใช้งานรีเลย์ชนิดนี้มีข้อเสียคือ ไม่ว่าจะเป็นการเกิดโหลดเกินหรือเกิดการลัดวงจร รีเลย์จะทำงานด้วยความเร็วเท่ากัน ซึ่งในความเป็นจริงแล้วการเกิดลัดวงจรด้วยกระแสที่สูงกว่ารีเลย์ควรจะทำงานได้เร็วกว่า การเกิดโหลดเกินนั้นรีเลย์ต้องยอมให้มีการหน่วงเวลาที่ยาวนานกว่า เพื่อเหตุผลของการจัดลำดับความสำคัญการทำงานของรีเลย์ ทำให้รีเลย์ชนิดเวลาผกผันถูกนำมาใช้งานเนื่องจากรีเลย์ชนิดนี้มีช่วงเวลาการทำงานของรีเลย์เป็นสัดส่วนกลับกับค่าของกระแส กล่าวคือเมื่อเกิดความผิดปกติขึ้นในระบบที่ค่ากระแสที่เกินพิกัดน้อย รีเลย์จะมีการหน่วงเวลาในการทำงาน จะไม่ตัดในทันที ในขณะที่ถ้ากระแสมีค่าสูงเช่น เกิดความผิดปกติ รีเลย์ต้องทำงานอย่างรวดเร็วโดยมีการหน่วงเวลาน้อยหรือทำงานตัดวงจรทันที

การทำงานของรีเลย์ จะพิจารณาจากเส้นโค้งเวลาและกระแสของรีเลย์ดังรูปที่ 2.2 โดยที่ค่ากระแสต่างกัน รีเลย์กระแสเกินจะทำงานด้วยเวลาที่ต่างกันด้วย การตั้งค่าของรีเลย์มีดังนี้

1. Pickup setting เป็นการปรับตั้งค่าให้รีเลย์เริ่มต้นทำงานที่ค่ากระแสค่าหนึ่ง โดยที่รีเลย์จะทำงานเมื่อรีเลย์ตรวจพบกระแสที่มีค่าเท่ากับค่านี้หรือมากกว่า

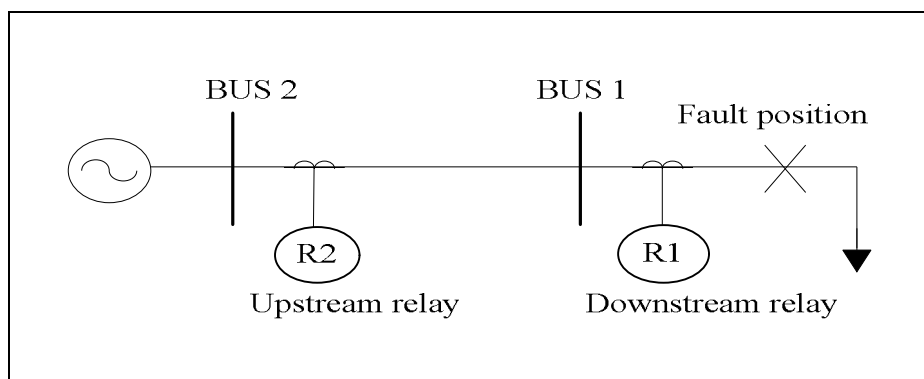
2. Current tap setting (CTS) เป็นค่าเทปของขดลวดด้านอินพุต เมื่อระบุค่ากระแสทำงานได้แล้ว ต้องทำการเลือก CTS ให้สอดคล้องเพื่อการป้องกันที่สมบูรณ์

3. Time-delay setting (TDS) เป็นการตั้งค่าการหน่วงเวลาในการทำงานของรีเลย์ หรือกล่าวอีกนัยหนึ่งได้ว่า TDS คือการเลือกเส้นโค้งเวลาการทำงานของรีเลย์นั่นเอง



รูปที่ 2.2 เส้นโค้งเวลาและกระแสของรีเลย์

การหน่วงเวลาของรีเลย์เป็นปัจจัยที่สำคัญอีกประการหนึ่งสำหรับการจัดลำดับความสัมพันธการทำงานของรีเลย์ป้องกันกระแสเกิน การหน่วงเวลาที่จะกล่าวถึงนี้จะพิจารณาสำหรับรีเลย์ตัวที่อยู่ติดกัน โดยจะเรียกรีเลย์ตัวที่อยู่ใกล้ความผิดปกติมากกว่าว่า downstream relay ส่วนรีเลย์ตัวที่อยู่ถัดไปจะเรียกว่า upstream relay ดังรูปที่ 2.3 โดยการจัดลำดับความสัมพันธ์ของรีเลย์ทั้ง 2 ตัวนี้ ให้ทำการบวกค่าเวลาส่วนเผื่อของการจัดลำดับความสัมพันธ์ (Coordination Time Interval : CTI) ประมาณ 0.3 – 0.5 วินาที



รูปที่ 2.3 ตำแหน่งการเรียกชื่อของรีเลย์เมื่อเกิดความผิดปกติ

ความไวในการทำงานของรีเลย์ยังถูกแบ่งออกได้เป็นหลายแบบตามรูปร่างของเส้นโค้งเวลาการทำงานของรีเลย์ ได้แก่

1. แบบเวลาผกผันมาตรฐาน (standard inverse time) พบได้ในระบบจำหน่ายและระบบส่งจ่ายทุกระดับแรงดัน ไม่ว่าจะเป็นรีเลย์หลักหรือรีเลย์สำรอง ปกติจะใช้ในกรณีที่ไม่มีการจัดลำดับความสัมพันธ์การทำงานของรีเลย์กับอุปกรณ์ป้องกันชนิดอื่น เช่น ฟิวส์ เป็นต้น มีสมการกำกับเส้นโค้งการทำงานดังสมการที่ (2-1)

$$t = \frac{0.14 \times TDS}{PSM^{0.02} - 1} \quad (2-1)$$

2. แบบเวลาผกผันยาวนาน (long time inverse) รีเลย์ชนิดนี้มักใช้เพื่อป้องกันตัวต้านทานสำหรับต่อลงดินของสายนิวทรัลหรือใช้สำหรับป้องกันโหลดเกินให้กับมอเตอร์หรือเครื่องกำเนิดไฟฟ้า มีสมการกำกับเส้นโค้งการทำงานดังสมการที่ (2-2)

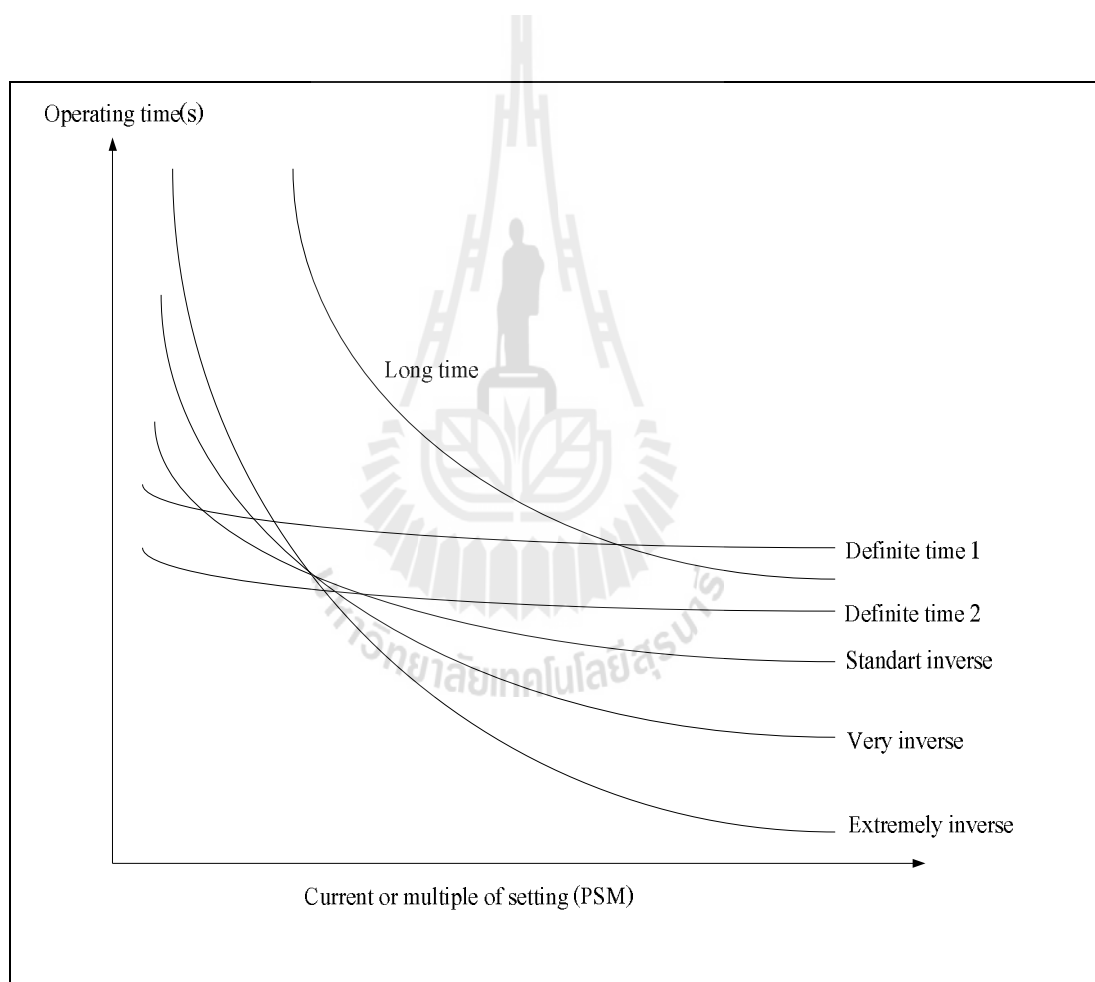
$$t = \frac{120 \times TDS}{PSM - 1} \quad (2-2)$$

3. แบบเวลาผกผันยาวนานมาก (very inverse time) การใช้รีเลย์กระแสเกินแบบเวลาผกผันมาตรฐานมีผลกระทบต่อการทำงานในกรณีที่เกิดการลัดวงจรใกล้ๆ กับแหล่งจ่าย จะทำให้เวลาในการกำจัดความผิดปกติโดยเฉลี่ยช้ากว่าเวลาในการกำจัดความผิดปกติ ณ ตำแหน่งอื่น ๆ การนำรีเลย์กระแสเกินแบบเวลาผกผันยาวนานมากมาใช้จะช่วยแก้ปัญหานี้ได้ มีสมการกำกับเส้นโค้งการทำงานดังสมการที่ (2-3)

$$t = \frac{13.5 \times TDS}{PSM - 1} \quad (2-3)$$

4. แบบเวลาผกผันยาวนานที่สุด (extremely inverse time) เหมาะสำหรับนำมาใช้จัดลำดับความสำคัญการทำงานของรีเลย์กับฟิวส์หรือใช้ป้องกันวงจรที่มีกระแสสูงขณะเปิดวงจร เช่น สายป้อน วงจรเครื่องทำความร้อน เป็นต้น มีสมการกำกับเส้นโค้งการทำงานดังสมการที่ (2-4)

$$t = \frac{80 \times TDS}{PSM^2 - 1} \quad (2-4)$$



รูปที่ 2.4 เส้นโค้งเวลาและกระแสของรีเลย์ป้องกันแบบมาตรฐาน

รีเลย์ที่ใช้ป้องกันระบบไฟฟ้ากำลัง แบ่งตามลำดับการทำงานได้ 2 ประเภทคือ

1. รีเลย์หลัก (primary relay) เป็นรีเลย์ที่ใช้ป้องกัน โดยปกติการป้องกันจะแบ่งเขตป้องกันไว้เฉพาะเมื่อเกิดการลัดวงจรขึ้นภายในเขตป้องกัน รีเลย์หลักจะสั่งให้เซอร์กิตเบรกเกอร์ทุกตัวในเขตป้องกันนั้นเปิดวงจร (trip) เขตป้องกันของรีเลย์หลักจะจัดแบ่งให้คาบเกี่ยวกัน (overlap) เพื่อป้องกันมิให้มีจุดบอดขึ้นในระบบป้องกัน รีเลย์หลักที่ทำงานถูกต้องจะสั่งให้เซอร์กิตเบรกเกอร์ทำงานน้อยที่สุดโดยจะสั่งให้เปิดวงจรเฉพาะส่วนที่เกิดการลัดวงจรขึ้นเท่านั้น

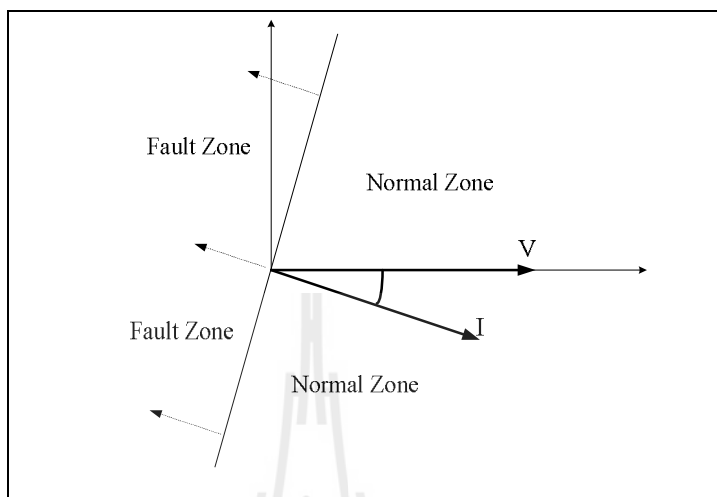
2. รีเลย์สำรอง (backup relay) ใช้ป้องกันแทนรีเลย์หลักกรณีที่รีเลย์หลักไม่ทำงานซึ่งอาจเกิดขึ้นได้เนื่องจากไม่ได้จ่ายแหล่งจ่ายไฟฟ้าให้กับวงจรสั่งเปิดวงจรหรือกลไกของเซอร์กิตเบรกเกอร์ขัดข้อง เป็นต้น เวลาการทำงานของรีเลย์สำรองจะช้ากว่ารีเลย์หลัก ส่วนตำแหน่งที่ตั้งติดตั้งของรีเลย์สำรองนี้จะอยู่ที่สถานีไฟฟ้าย่อยต่าง ๆ หรืออยู่ไกลจากรีเลย์หลัก ถ้าเป็นไปได้มักจะให้รีเลย์สำรองนี้อยู่คนละตำแหน่งกับรีเลย์หลัก คืออยู่ต่างสถานีไฟฟ้าย่อยกันและการสั่งเปิดวงจรเบรกเกอร์ด้วยรีเลย์สำรองนี้ อาจจะทำให้ส่วนที่ไม่เกี่ยวข้องกันกับจุดที่เกิดการลัดวงจรถูกสั่งให้เปิดวงจรเบรกเกอร์ไปด้วย นอกจากนี้หน้าที่ของรีเลย์สำรองอีกอย่างหนึ่งก็คือทำหน้าที่เป็นรีเลย์หลักในกรณีที่มีการซ่อมแซมหรือบำรุงรักษารีเลย์หลัก

2.5.2 รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง

รีเลย์ป้องกันกระแสเกินแบบมีทิศทางเป็นรีเลย์กระแสเกินที่มีตัวเปรียบเทียบเฟสหรือทิศทางการไหลของกำลังไฟฟ้า จึงเรียกว่ารีเลย์ป้องกันกระแสเกินแบบมีทิศทาง (directional over-current relay) นิยมใช้กับระบบไฟฟ้ากำลังที่มีแหล่งจ่ายมากกว่า 1 แหล่งจ่ายหรือในระบบไฟฟ้ากำลังที่มีวงจรลูปหรือโครงข่ายจำนวนมาก เมื่อเกิดความผิดปกติขึ้นในระบบไฟฟ้า กระแสลัดวงจรจะไหลผ่านรีเลย์ได้สองทิศทาง จึงจำเป็นต้องมีปริมาณอ้างอิงเพื่อใช้กำหนดทิศทางของกระแสลัดวงจรที่เกิดขึ้น โดยการตรวจสอบนี้ต้องใช้รีเลย์ที่วัดปริมาณ 2 ปริมาณ ได้แก่ กระแสที่ตรวจจับและสัญญาณอ้างอิง โดยปกติจะใช้แรงดันบัส ณ ตำแหน่งติดตั้งรีเลย์เป็นสัญญาณอ้างอิง เมื่อมุมเฟสของสัญญาณที่นำมาเปรียบเทียบกันซึ่งปกติได้แก่ กระแสและแรงดัน มีความต่างเฟสกันเกินกว่าที่กำหนด หมายความว่า การไหลของกำลังไฟฟ้าเกิดการกลับทิศทาง

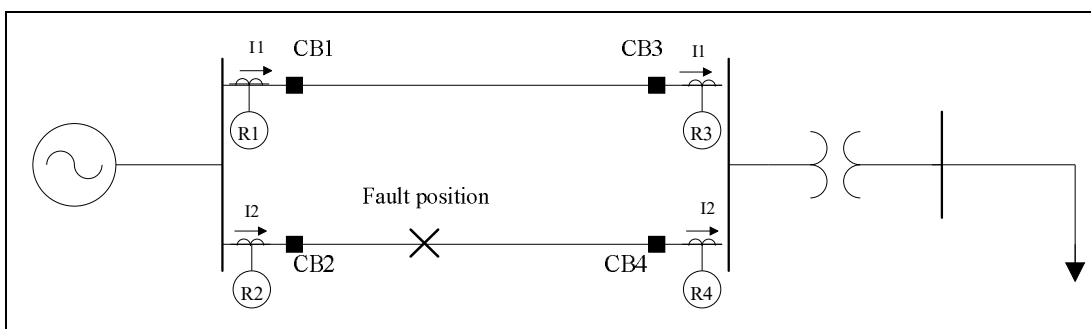
พิจารณารูปที่ 2.5 การตรวจสอบและพิจารณาทิศทางของกระแสในขณะใด ๆ เทียบกับเฟสอ้างอิง โดยในสภาวะการจ่ายโหลดปกตินั้น กระแสจะล้าหลังแรงดันเล็กน้อยไม่เกิน 90 องศา เมื่อเกิดการลัดวงจรจะส่งผลให้มุมเฟสของกระแสลัดวงจรมีค่าประมาณ 180 องศา ดังนั้น การตรวจสอบความต่างเฟสนี้ เมื่อเทียบกับเฟสอ้างอิงเดิมในสภาวะปกติ มุมเฟสของกระแสกับแรงดันจะต่างกันไม่เกิน 90 องศา แต่เมื่อเกิดความผิดปกติขึ้นในระบบ มุมเฟสจะต่างกันมากกว่า

90 องศา ทำให้นำมาใช้พิจารณาและวินิจฉัยเพื่อสั่งการไปยังเซอร์กิตเบรกเกอร์ให้ตัดวงจรได้นอกจากการใช้ขนาดกระแสมาพิจารณาเพียงอย่างเดียว



รูปที่ 2.5 มุมเฟสของกระแสและแรงดัน

จากระบบไฟฟ้ากำลังดังรูปที่ 2.6 จะพบว่า การใช้รีเลย์ป้องกันกระแสเกินดังรูป เมื่อเกิดการลัดวงจรขึ้นระหว่างสายส่งจะทำให้มีกระแสลัดวงจรไหลผ่าน R1 R2 R3 และ R4 นั่นคือ รีเลย์ทุกตัวเห็นการลัดวงจรและจะสั่งเซอร์กิตเบรกเกอร์ตัดวงจร ทำให้สายส่งทั้ง 2 ชุดถูกตัดขาดออก ส่งผลให้โหลดถูกขัดจังหวะการจ่ายไฟ แต่เมื่อมีการติดตั้งรีเลย์ป้องกันกระแสเกินแบบมีทิศทางจะทำให้รีเลย์ R2 ทำงานเนื่องจากมีค่ากระแสลัดวงจรสูงที่สุด ดังนั้น CB2 จะเปิดวงจร ในขณะเดียวกัน R4 จะตรวจพบความต่างเฟสที่เกิน 90 องศาและขนาดกระแสที่สูงเกินกำหนด ดังนั้น CB4 จะถูกเปิดวงจรออก ในขณะที่ R1 และ R3 มีการตรวจสอบทิศทางจะไม่พบความผิดปกติใด ๆ เพียงแต่มีกระแสค่าสูงชั่วขณะแต่จะยังไม่ทำงาน เมื่อ R2 และ R4 ทำงาน กระแสที่ผ่าน R1 และ R3 จะเข้าสู่สภาวะปกติ ทำให้สายส่งเส้นที่ไม่เกิดความผิดปกติยังคงทำงานจ่ายโหลดได้ตามปกติ



รูปที่ 2.6 ระบบไฟฟ้ากำลังแบบใช้สายส่ง 2 เส้นขนานกัน

2.5.3 การปรับตั้งรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง

ทำได้โดยการตั้งค่า pickup setting และค่า TDS ที่เหมาะสมทั้งสองทิศทาง

1. Pickup setting ให้พิจารณาค่าของโหลดสูงสุดที่เป็นไปได้จากการจ่ายโหลดทั้งสองทิศทางและให้ใช้ตัวปรับคูณที่เหมาะสม เช่น 1.25 1.3 1.4 หรือ 1.5 เป็นต้น
2. TDS ให้พิจารณาตามหลักการของรีเลย์กระแสเกินทุกประการ โดยให้บวกค่า time grading margin ประมาณ 0.3 – 0.5 วินาทีสำหรับ upstream relay
3. Instantaneous setting ให้ใช้ได้ตามความเหมาะสม

2.5.4 ช่วงความเผื่อ (grading margin)

การจัดลำดับเวลาการทำงานของรีเลย์ในระบบไฟฟ้าให้สัมพันธ์กันกล่าวคือรีเลย์ที่อยู่ใกล้ตำแหน่งการลัดวงจรที่สุดจะถูกจัดเป็นการป้องกันหลัก จะต้องทำงานก่อนรีเลย์ทุก ๆ ตัว โดยรีเลย์ตัวถัดไปจะจัดเป็นการป้องกันสำรองและจะต้องหน่วงเวลาไว้ปริมาณหนึ่ง การหน่วงเวลาของรีเลย์สำรองจะเรียกว่าช่วงความเผื่อ (grading margin) ที่จะต้องพิจารณาให้เหมาะสมเนื่องจากหากมีค่ามากเกินไป ความเสียหายต่อระบบและอุปกรณ์ไฟฟ้าจะมีมาก แต่ถ้ามีค่าน้อยเกินไปการทำงานของรีเลย์สำรองอาจทำงานก่อนรีเลย์ป้องกันหลักซึ่งจะลัดวงจรของระบบไฟฟ้าออกมากเกินไป ความจำเป็น ช่วงความเผื่อจะขึ้นอยู่กับปัจจัยต่าง ๆ ต่อไปนี้

- 1) เวลาในการตัดวงจรของเซอร์กิตเบรกเกอร์เมื่อมีไฟฟ้าเลี้ยงวงจรทริป จะเกิดแรงทำให้หน้าสัมผัสของเซอร์กิตเบรกเกอร์เคลื่อนที่ออกและเกิดอาร์กกระหว่างหน้าสัมผัสทั้งสอง เนื่องจากการเคลื่อนที่ทางกลจะต้องใช้เวลา โดยเวลาทั้งหมดในการตัดวงจรของเซอร์กิตเบรกเกอร์

จะขึ้นอยู่กับชนิดของเซอร์กิตเบรกเกอร์ด้วย โดยมีค่าประมาณ 5 วัฏจักร (cycle) หรือประมาณ 0.1 วินาที

2) เวลาที่รีเลย์ทำงานก่อน (overshoot time) หลังจากที่มีรีเลย์ถูกตัดไฟออกแล้ว รีเลย์ยังคงทำงานต่ออีกเล็กน้อยจนกระทั่งพลังงานที่เก็บไว้ของรีเลย์หมดไป ตัวอย่างเช่น induction disc element จะเก็บไว้ในรูปพลังงานจลน์ ส่วนสแตติกรีเลย์จะมีพลังงานที่เก็บไว้ในตัวเก็บประจุ โดยทั่วไปเวลาที่รีเลย์ทำงานก่อน จะมีค่าเท่ากับ 0.05 วินาทีสำหรับรีเลย์แบบดิจิทัลซึ่งเป็นรีเลย์ สมัยใหม่จะมีค่าน้อยมากจึงอาจตัดทิ้งได้

3) ค่าความผิดพลาดของอุปกรณ์ที่ใช้วัดเช่น รีเลย์ หม้อแปลงวัดกระแส จะมีความผิดพลาดและค่าลักษณะทางเวลาของรีเลย์มีค่าผิดพลาดทั้งทางด้านลบและทางด้านบวก โดยที่ค่าความผิดพลาดของหม้อแปลงวัดกระแสเกิดจากลักษณะทางแม่เหล็ก (magnetizing characteristic) โดยค่าความผิดพลาดของหม้อแปลงวัดกระแสจะไม่มีผลต่อรีเลย์แบบทำงานที่เวลาตายตัว (definite time overcurrent relays)

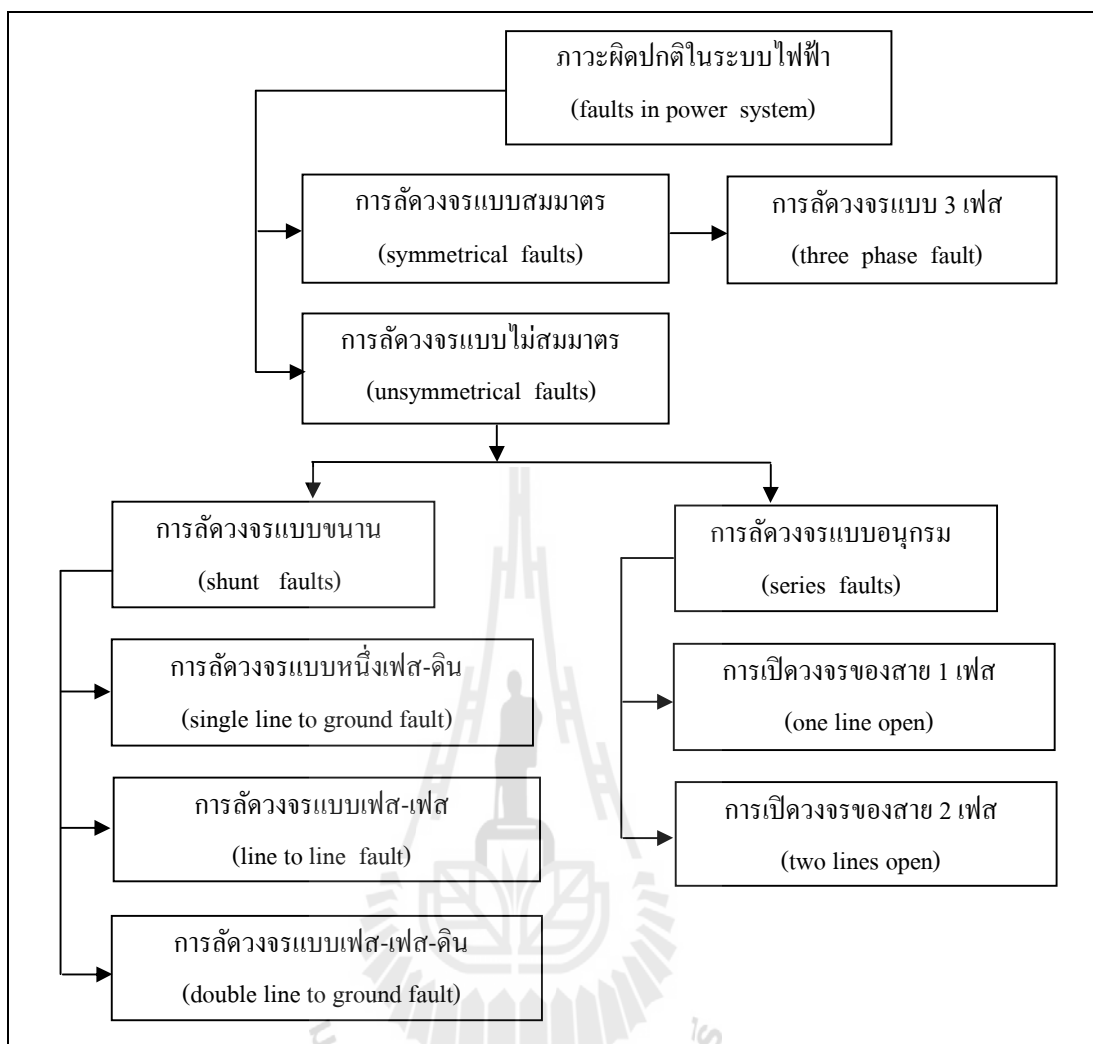
4) ค่าเวลาเผื่อไว้เพื่อความปลอดภัย (safety margin) ปกติแล้วเวลา 100 มิลลิวินาที จะถูกรวมเข้าไปในขั้นตอนสุดท้ายของการคำนวณเพื่อให้แน่ใจว่าการจัดลำดับเวลาการทำงานให้สัมพันธ์กันได้ถูกต้อง โดยค่าที่นิยมเลือกใช้กัน ดังนี้

- เวลาในการตัดวงจรของเซอร์กิตเบรกเกอร์ (circuit breaker)	0.10 วินาที
- เวลาที่รีเลย์ทำงานก่อน (overshoot time)	0.05 วินาที
- ค่าความผิดพลาดของอุปกรณ์ทั้งหมด	0.10 วินาที
- ค่าเวลาเผื่อไว้เพื่อความปลอดภัย (safety margin)	0.10 วินาที
ผลรวมทั้งหมด	0.35 วินาที

โดยทั่วไปค่าช่วงความเผื่อ (grading margin) ที่นิยมใช้กับคือ 0.3 - 0.5 วินาที ซึ่งเพียงพอสำหรับการตรวจจับได้อย่างถูกต้อง

2.6 ทฤษฎีของการลัดวงจรในระบบไฟฟ้า

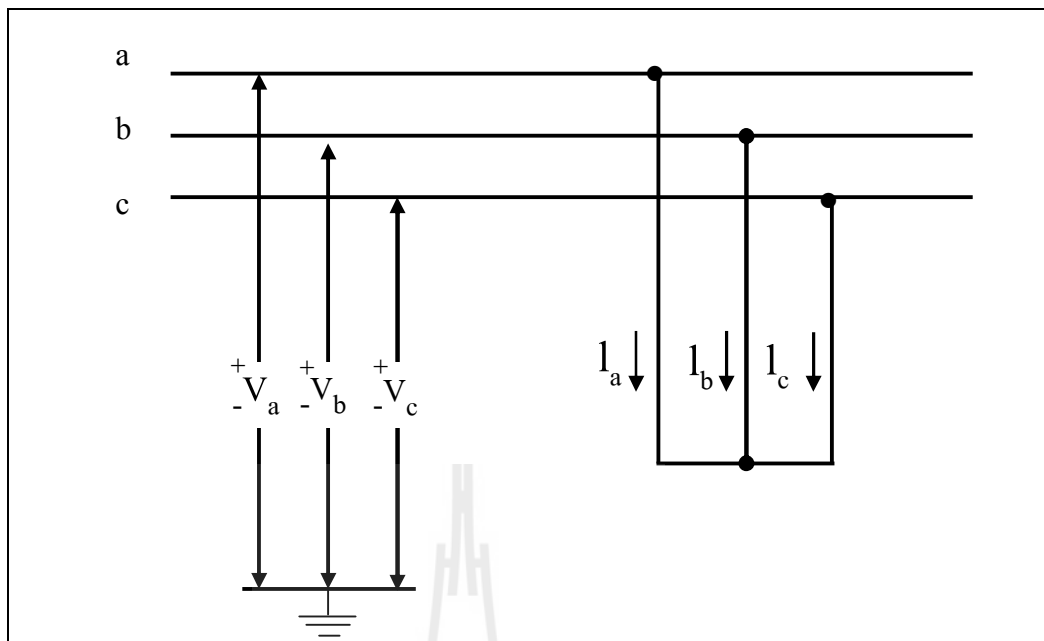
การคำนวณกระแสลัดวงจรในระบบไฟฟ้าจะใช้หลักการของวิธีการองค์ประกอบสมมาตร ซึ่งถูกคิดค้นขึ้นในปี ค.ศ.1918 โดย C.L.Fortescus ได้กล่าวไว้ว่า ในระบบไม่สมดุล n เฟส สามารถที่จะแก้ปัญหาก็ได้โดยใช้ระบบที่สมดุล n ระบบมีเฟสเซอร์ที่สมดุลอยู่โดยในระบบไฟฟ้า 3 เฟสจะประกอบไปด้วยองค์ประกอบลำดับศูนย์ องค์ประกอบลำดับบวกและองค์ประกอบลำดับลบ ซึ่งสามารถคำนวณหากระแสลัดวงจรในแต่ละประเภทได้ดังนี้



รูปที่ 2.7 แผนภาพสภาวะผิดปกติในระบบไฟฟ้า

2.6.1 การลัดวงจรแบบ 3 เฟส (three phase fault)

การลัดวงจรแบบ 3 เฟส ในระบบไฟฟ้ามีโอกาสเกิดขึ้นไม่บ่อยครั้งนัก แต่หากเกิดขึ้นแล้วมันจะมีขนาดกระแสลัดวงจรไหลรุนแรงมากกว่าการลัดวงจรประเภทอื่น ๆ แสดงได้ดังรูปที่ 2.8

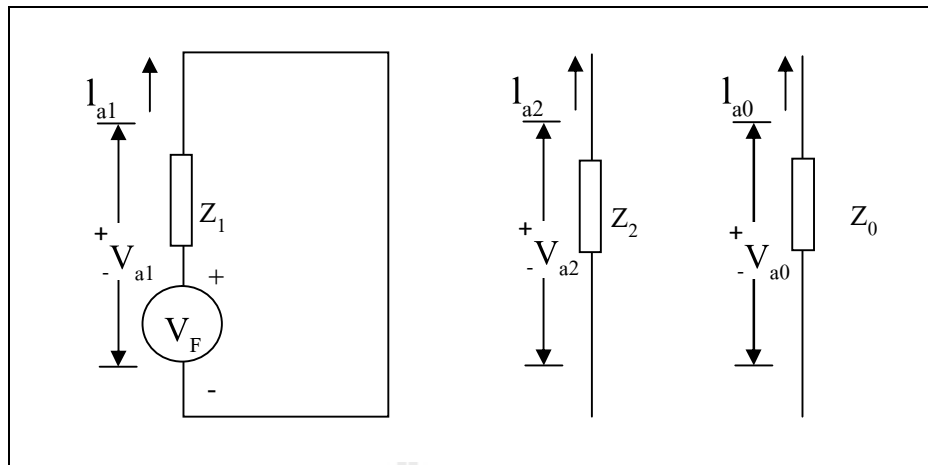


รูปที่ 2.8 การเกิดลัดวงจรแบบ 3 เฟส

จากรูปที่ 2.8 จะได้ค่ากระแส $I_a + I_b + I_c = 0$ และค่าของแรงดัน $V_a = V_b = V_c$
หรือแทนค่า $I_a + a^2 I_a + a I_a = 0$ ในสมการ จะได้

$$\begin{bmatrix} I_{a0} \\ I_{a1} \\ I_{a2} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} I_a \\ I_b = a^2 I_a \\ I_c = a I_a \end{bmatrix} \quad (2-5)$$

จะได้ $I_{a0} = I_{a1} = I_{a2}$ ซึ่งจะทำให้สามารถเขียนการเชื่อมโยงของวงจรลำดับศูนย์
ลำดับบวกและลำดับลบ ได้ดังรูปที่ 2.9



รูปที่ 2.9 วงจรของการเกิดลัดวงจรแบบ 3 เฟส

จากรูปที่ 2.9 จะได้

$$I_{a1} = \frac{V_F}{Z_1} = I_a \quad (2-6)$$

$$I_a = \frac{V_F}{Z_1} \quad (2-7)$$

หาแรงดันแต่ละเฟสได้จาก

แทนค่า $V_a = V_b = V_c$ ในสมการแรงดันลำดับต่าง ๆ

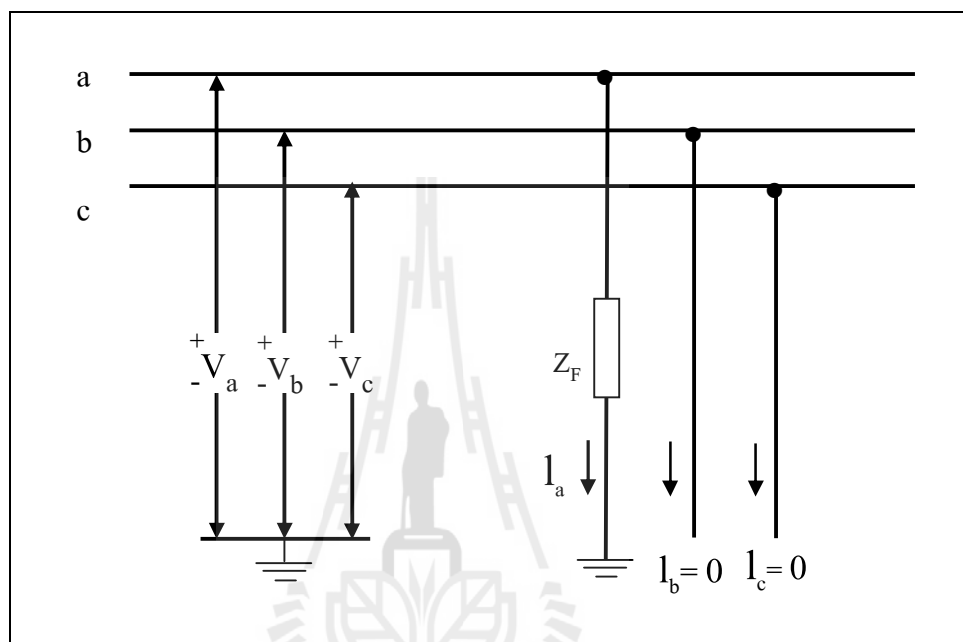
$$\begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} V_a \\ V_b = V_a \\ V_c = V_a \end{bmatrix} \quad (2-8)$$

จะได้ $V_{a0} = V_a$, $V_{a1} = V_{a2} = 0$ และสำหรับแรงดันเฟส V_b และ V_c หาได้จาก

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} \quad (2-9)$$

2.6.2 การลัดวงจรแบบ เฟส-ดิน (single line to ground fault)

การลัดวงจรแบบ เฟส-ดิน เป็นการลัดวงจรที่มีโอกาสเกิดขึ้นง่ายกว่าการลัดวงจรประเภทอื่น ๆ อาจเกิดจากการแฟลชข้ามลูกถ้วยเนื่องจากฟ้าผ่าหรือเกิดจากสายตัวนำสัมผัสกับโครงสร้างที่อยู่บนพื้นดิน ซึ่งแสดงได้ดังรูปที่ 2.10

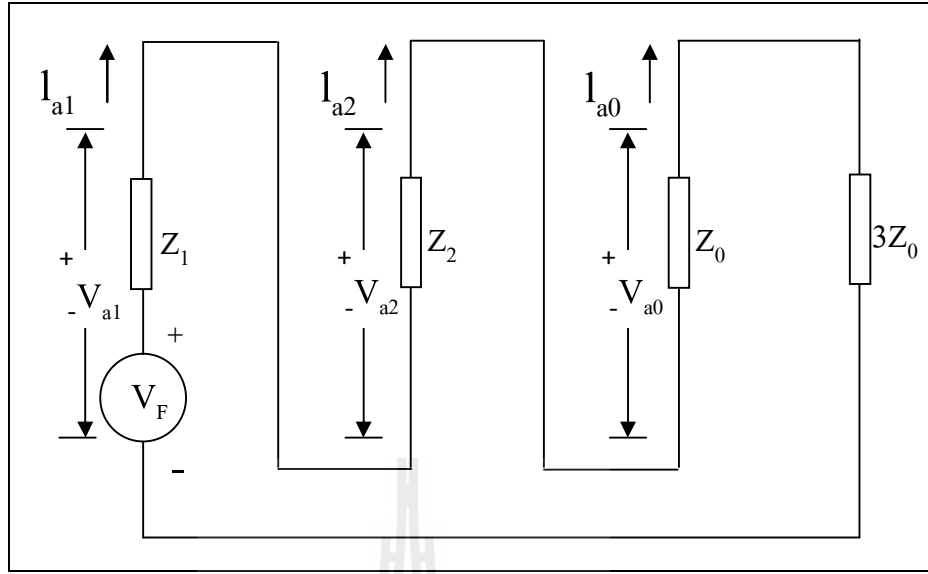


รูปที่ 2.10 การเกิดลัดวงจรแบบ เฟส-ดิน

จากรูปที่ 2.10 จะได้ค่าของกระแส $I_b = I_c = 0$ และค่าของแรงดัน $V_a = Z_F I_a$ แทนค่า $I_b = I_c = 0$ ในสมการกระแสลำดับต่าง ๆ

$$\begin{bmatrix} I_{a0} \\ I_{a1} \\ I_{a2} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} I_a \\ I_b = 0 \\ I_c = 0 \end{bmatrix} \quad (2-10)$$

จะได้ $I_{a0} = I_{a1} = I_{a2} = \frac{I_a}{3}$ ซึ่งทำให้สามารถเขียนการเชื่อมโยงของวงจรลำดับศูนย์ ลำดับบวกและลำดับลบ ได้ดังรูปที่ 2.11



รูปที่ 2.11 วงจรของการเกิดลัดวงจรแบบ เฟส-ดิน

จากรูปที่ 2.11 สามารถกำหนดค่าของกระแสและค่าของแรงดันต่าง ๆ ได้ดังนี้

$$I_{a1} = \frac{V_F}{Z_0 + Z_1 + Z_2 + 3Z_F} \quad (2-11)$$

$$I_a = 3I_{a0} = 3I_{a1} = 3I_{a2} \quad (2-12)$$

$$I_a = \frac{3V_F}{Z_0 + Z_1 + Z_2 + 3Z_F} \quad (2-13)$$

$$I_{a0} = -Z_0 I_{a0} \quad (2-14)$$

$$V_{a1} = V_F - Z_1 I_{a1} \quad (2-15)$$

$$V_{a2} = -Z_2 I_{a2} \quad (2-16)$$

$$V_a = V_{a0} + V_{a1} + V_{a2} \quad (2-17)$$

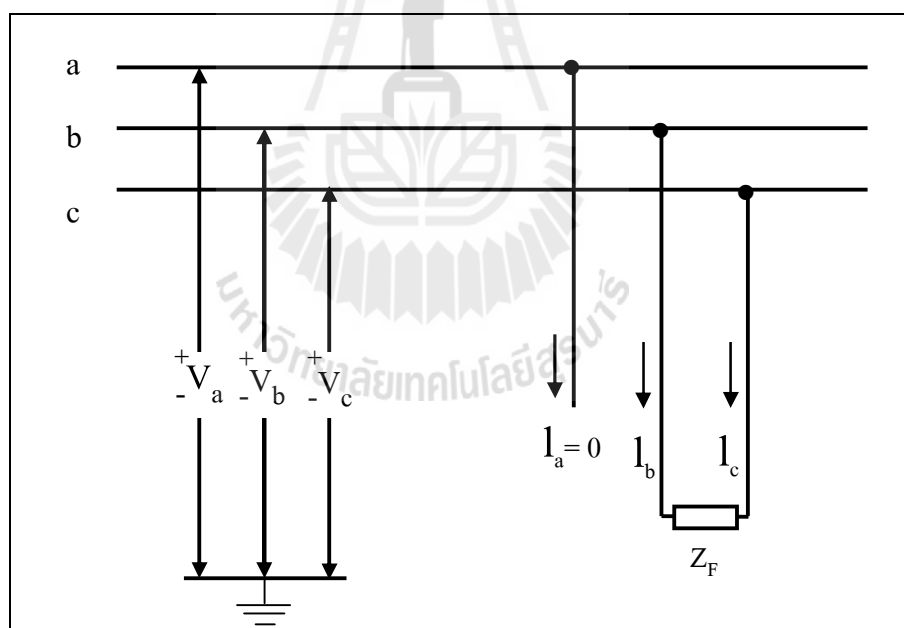
$$V_a = V_F - (Z_0 + Z_1 + Z_2)I_{a0} = 3Z_F I_{a0} \quad (2-18)$$

สำหรับค่าแรงดันเฟส V_b และ V_c หาได้จาก

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} \quad (2-19)$$

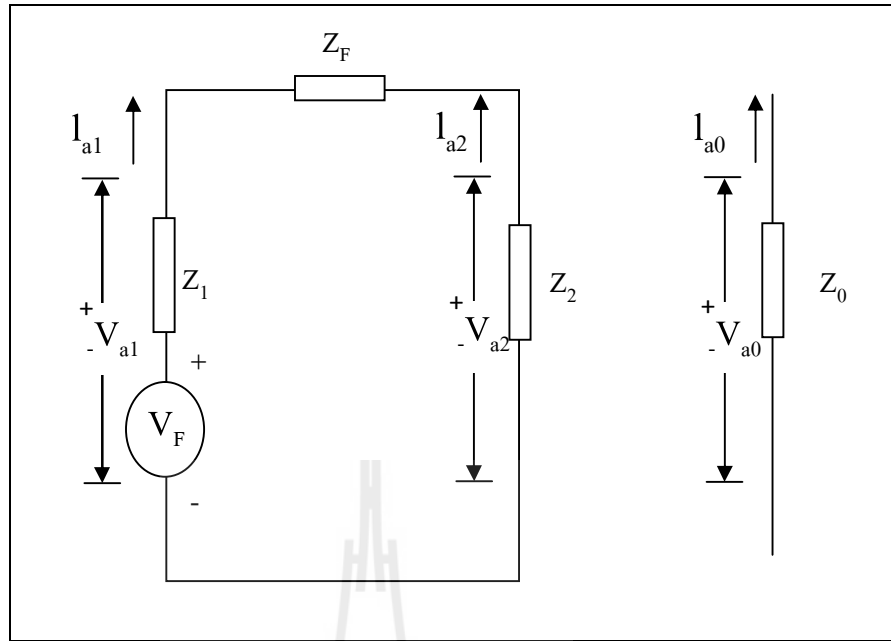
2.6.3 การลัดวงจรแบบ เฟส-เฟส (line to line fault)

การลัดวงจรแบบ เฟส-เฟส (line to line fault) ผ่านค่าอิมพีแดนซ์ที่เฟส b และ c ได้แสดงดังรูปที่ 2.12



รูปที่ 2.12 การเกิดลัดวงจรแบบ เฟส-เฟส

จากรูปที่ 2.12 ค่า $I_a = 0$ และค่า $I_b = -I_c$ ส่วนค่าของแรงดัน $V_b - V_c = I_b Z_F$ ซึ่งทำให้สามารถเขียนการเชื่อมโยงของวงจรลำดับศูนย์ ลำดับบวกและลำดับลบ ได้ดังรูปที่ 2.13



รูปที่ 2.13 วงจรของการเกิดลัดวงจรแบบ เฟส-เฟส

จากรูปที่ 2.13 จะได้ค่าของกระแสและค่าของแรงดัน ดังนี้

$$I_a = 0, I_b = -I_c, V_b - V_c = I_b Z_F \quad (2-20)$$

$$V_{a0} = 0 \quad (2-21)$$

$$V_{a1} = V_F - Z_1 I_{a1} = V_{a2} + Z_F I_{a1} \quad (2-22)$$

$$V_{a2} = -Z_2 I_{a2} \quad (2-23)$$

แทนค่า $I_{a0} = 0$ และ $I_{a1} = -I_{a2}$ ในสมการกระแสเฟส

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} I_{a0} = 0 \\ I_{a1} \\ I_{a2} = -I_{a1} \end{bmatrix} \quad (2-24)$$

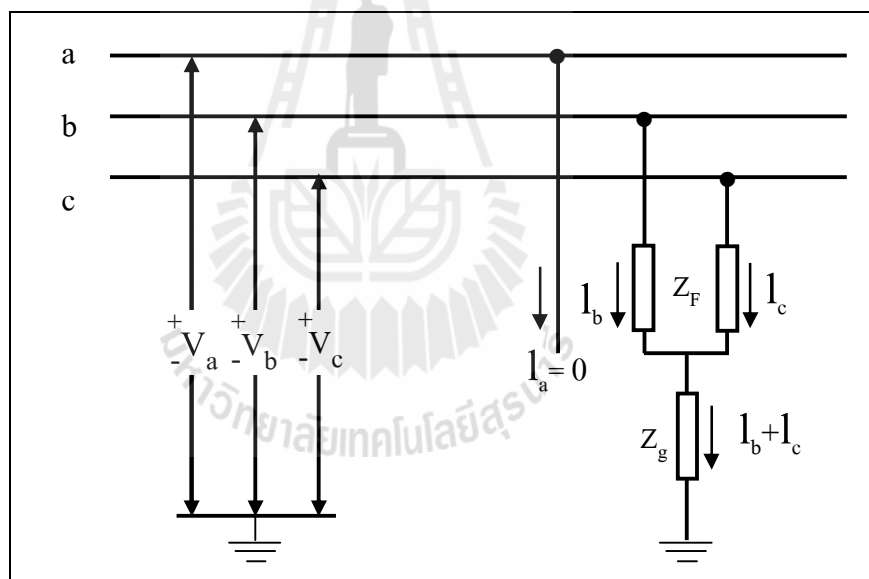
$$I_b = -I_c = a^2 I_{a1} - a I_{a1} \quad (2-25)$$

หาแรงดันแต่ละเฟสได้จาก

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix} \begin{bmatrix} V_{a0} = 0 \\ V_{a1} \\ V_{a2} \end{bmatrix} \quad (2-26)$$

2.6.4 การลัดวงจรแบบ เฟส-เฟส-ดิน (double line to ground fault)

การลัดวงจรแบบ เฟส-เฟส-ดิน ระหว่างเฟส b และเฟส c ผ่านอิมพีแดนซ์ลงดิน แสดงได้ดังรูปที่ 2.14



รูปที่ 2.14 การเกิดลัดวงจรแบบ เฟส-เฟส-ดิน

จากรูปที่ 2.14 สามารถกำหนดค่าของกระแสและค่าของแรงดันต่าง ๆ ได้ดังนี้

$$I_a = 0 = I_{a0} + I_{a1} + I_{a2} \quad (2-27)$$

$$V_b = (Z_F + Z_g)I_b + Z_g I_c \quad (2-28)$$

$$V_c = (Z_F + Z_g)I_b + Z_g I_b \quad (2-29)$$

$$V_b - V_c = Z_F(I_b - I_c) \quad (2-30)$$

$$V_b + V_c = Z_F[2I_{a0} - (I_{a1} + I_{a2})] + Z_g[4I_{a0} - 2(I_{a1} + I_{a2})] \quad (2-31)$$

และจากสมการของแรงดัน

$$V_b = V_{a0} + a^2 V_{a1} + a V_{a2} \quad (2-32)$$

$$V_c = V_{a0} + a V_{a1} + a^2 V_{a2} \quad (2-33)$$

$$V_b - V_c = -j\sqrt{3}(V_{a1} - V_{a2}) \quad (2-34)$$

$$V_b + V_c = [2V_{a0} - (V_{a1} - V_{a2})] \quad (2-35)$$

$$Z_F(I_b - I_c) = -j\sqrt{3}(V_{a1} - V_{a2}) \quad (2-36)$$

$$V_{a1} - V_{a2} = Z_F(I_{a1} + I_{a2}) \quad (2-37)$$

$$V_{a1} - Z_F I_{a1} = V_{a2} - Z_F I_{a2} \quad (2-38)$$

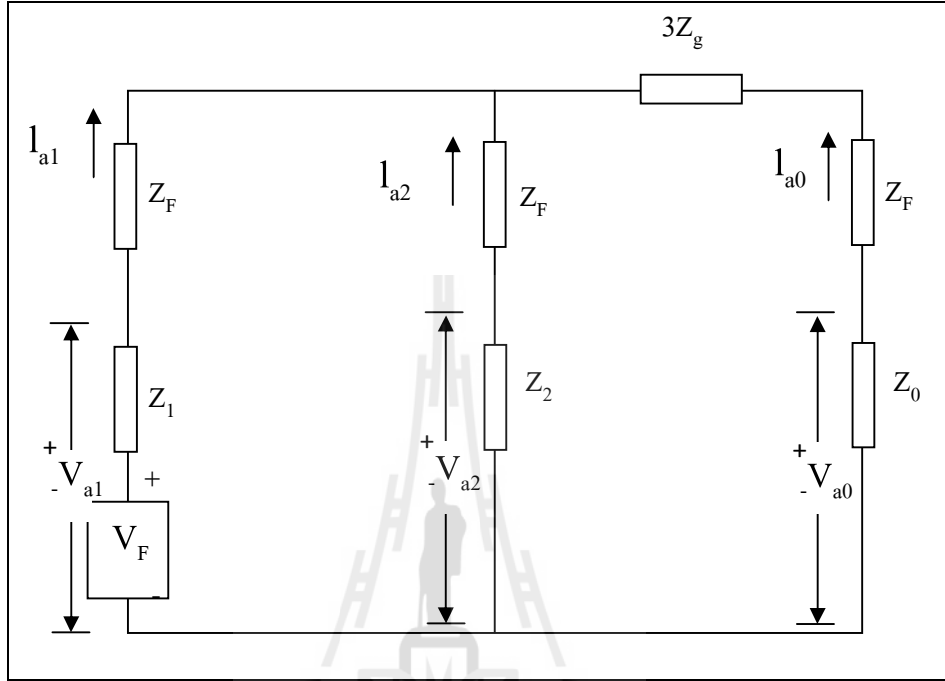
$$Z_F[2I_{a0} - (I_{a1} + I_{a2})] + Z_g[4I_{a0} - 2(I_{a1} + I_{a2})] = [2V_{a0} - (V_{a1} - V_{a2})] \quad (2-39)$$

$$2V_{a0} - 2Z_F I_{a0} - 4Z_g I_{a0} = V_{a1} + V_{a2} - Z_F(I_{a1} + I_{a2}) - 2Z_g(I_{a1} + I_{a2}) \quad (2-40)$$

$$I_{a1} + I_{a2} = -I_{a0} \quad (2-41)$$

$$\therefore V_{a1} = Z_F I_{a1} + V_{a0} - Z_F I_{a0} - 3Z_g I_{a0} \quad (2-42)$$

ซึ่งจะทำให้สามารถเขียนการเชื่อมโยงของวงจรลำดับศูนย์ ลำดับบวกและลำดับลบ แสดงได้ดังรูปที่ 2.15



รูปที่ 2.15 วงจรของการเกิดลัดวงจรแบบ เฟส-เฟส-ดิน

จากรูปที่ 2.15 สามารถกำหนดค่าของกระแสและค่าของแรงดันได้ดังนี้

$$I_{a1} = \frac{V_F}{Z_1 + Z_F \frac{(Z_2 + Z_F)(Z_0 + Z_F + 3Z_g)}{Z_2 + Z_0 + 2Z_F + 3Z_g}} \quad (2-43)$$

$$I_{a2} = \frac{Z_0 + Z_F + 3Z_g}{Z_2 + Z_0 + 2Z_F + 3Z_g} I_{a1} \quad (2-44)$$

$$I_{a2} = \frac{Z_2 + Z_F}{Z_2 + Z_0 + 2Z_F + 3Z_g} I_{a1} \quad (2-45)$$

$$V_{a0} = -Z_0 I_{a2} \quad (2-46)$$

$$V_{a1} = V_F - Z_1 I_{a1} \quad (2-47)$$

$$V_{a2} = -Z_2 I_{a2} \quad (2-48)$$

หาค่าของกระแสในแต่ละเฟสได้จาก

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} I_{a0} \\ I_{a1} \\ I_{a2} \end{bmatrix} \quad (2-49)$$

และหาค่าของแรงดันในแต่ละเฟสได้จาก

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} V_{a0} \\ V_{a1} \\ V_{a2} \end{bmatrix} \quad (2-50)$$

2.6.5 สถิติการเกิดลัดวงจร

การเกิดลัดวงจรในระบบไฟฟ้า ได้มีการรวบรวมโดยการไฟฟ้าที่เกี่ยวข้องและจากบริษัทผู้ผลิตและจำหน่ายอุปกรณ์ในระบบไฟฟ้า ดังแสดงในตารางที่ 2.2 (ชนิดชัย กุลวรวานิชพงษ์, 2552)

ตารางที่ 2.2 ข้อมูลการเกิดลัดวงจรในระบบไฟฟ้า แบ่งตามรายอุปกรณ์ในระบบไฟฟ้า

อุปกรณ์ไฟฟ้า	% การเกิดลัดวงจร
สายพาดอากาศ	50%
สายหุ้มฉนวน	10%
สวิตช์เกียร์	15%

ตารางที่ 2.2 ข้อมูลการเกิดลัดวงจรในระบบไฟฟ้า แบ่งตามรายอุปกรณ์ในระบบไฟฟ้า (ต่อ)

อุปกรณ์ไฟฟ้า	% การเกิดลัดวงจร
หม้อแปลงกำลัง	12%
หม้อแปลงเครื่องมือวัด	2%
อุปกรณ์ควบคุม	3%
อื่น ๆ	8%

จากตารางที่ 2.2 การลัดวงจรที่เกิดในระบบสายส่งเหนือศีรษะคิดเป็นปริมาณถึง 50% ของการลัดวงจรที่เกิดขึ้นทั้งหมด ดังนั้นการวิเคราะห์ถึงธรรมชาติของเหตุการณ์การลัดวงจรในระบบไฟฟ้าแบบ 3 เฟสแล้ว การเบรกคาว์ของฉนวนระหว่างเฟสใดเฟสหนึ่งลงดินจะเรียกว่า Single Line to Ground Fault (SLG) การเบรกคาว์ของฉนวนระหว่าง 2 เฟสคู่ใดคู่หนึ่ง จะเรียกว่า Line to Line Fault (L-L) การเบรกคาว์ระหว่างเฟส 2 เฟสลงดิน จะเรียกว่า Line to Line to Ground Fault (L-L-N) และการเบรกคาว์ของฉนวนระหว่างเฟส 3 เฟส จะเรียกว่า Three Phase Fault (L-L-L) และตารางที่ 2.3 ต่อไปนี้เป็นข้อมูลเบื้องต้นของการเกิดลัดวงจรในระบบไฟฟ้า

ตารางที่ 2.3 สถิติการเกิดลัดวงจร

ชนิดของการลัดวงจร	% การเกิดลัดวงจร
แบบ 3 เฟสสมมาตร (symmetrical three-phase fault)	2%
แบบ เฟส-ดิน (single line to ground fault)	85%
แบบ เฟส-เฟส (line to line fault)	8%
แบบเฟส-เฟส-ดิน (double line to ground fault)	5%

การเกิดลัดวงจรแบบสามเฟสสมมาตรหรือที่เรียกว่าฟอลต์แบบสามเฟสสมมาตร (symmetrical three phase fault) โดยทั่วไปเกิดจากความประมาทในการทำงาน เช่น กรณีจำเป็นต้องเข้าไปซ่อมบำรุงบัสบาร์ทั้ง 3 เฟสและมีการตัดไฟออก แล้วลัดวงจรทั้งสามเฟสเพื่อการทำประจุที่ยังตกค้างอยู่ในสายตัวนำอันเนื่องมาจากการเก็บและคายประจุของตัวเก็บประจุที่ค้าง (stray

capacitor) และเมื่อซ่อมบำรุงเสร็จเรียบร้อยแล้ว ผู้ปฏิบัติการอาจจะลืมปลดการลัดวงจรนี้ออก ดังนั้นเมื่อทำการสั่งให้เซอร์กิตเบรกเกอร์ทำงาน (on circuit breaker) เพื่อนำไฟเข้ามา จึงเกิดฟอลต์แบบสามเฟสสมมาตรขึ้นได้ ส่วนในกรณีการเกิดลัดวงจรเฟสเดียวลงดิน (single line ground fault) ที่เกิดขึ้นในวงจรสายส่งนั้นมักจะเป็นเหตุการณ์ที่เกิดขึ้นเพียงชั่วขณะ อันเนื่องมาจากมีตัวนำมาพาดผ่านสาย เช่น ฐ หรือกิ่งไม้ และเมื่อสิ่งเหล่านั้นหลุดล่องไปแล้ว เหตุการณ์การลัดวงจรก็จะยุติลงได้

2.7 กำหนดการเชิงเส้น (Linear Programming : LP)

ความเป็นเชิงเส้นคือคุณสมบัติที่นักวิจัยส่วนใหญ่ต้องการ เนื่องจากฟังก์ชันเชิงเส้นเข้าใจง่ายและถือเป็นเครื่องมือพื้นฐานในการแก้ปัญหาระบบไม่เชิงเส้นด้วย สมการของระบบไม่เชิงเส้นเมื่อพิจารณาในช่วงการเปลี่ยนแปลงค่าพารามิเตอร์ในช่วงแคบ ๆ สามารถประมาณได้ด้วยฟังก์ชันเชิงเส้น นอกจากนี้ปัญหามหาบางประเภทนิยามกำหนดให้อยู่ในรูปของฟังก์ชันเชิงเส้นเพื่อให้เข้าใจง่ายและไม่ซับซ้อนเช่น ปัญหาทางด้านเศรษฐศาสตร์ การประเมินต้นทุน ผลกำไรของสินค้าที่ผลิตหรือออกแบบ เป็นต้น ดังนั้นการศึกษากำหนดการเชิงเส้นมีความสำคัญและจำเป็นอย่างยิ่งในงานการออกแบบทางวิศวกรรม โดยปัญหาในรูปแบบของกำหนดการเชิงเส้นเป็นปัญหาเกี่ยวกับค่าเหมาะที่สุดที่มีฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับในรูปฟังก์ชันเชิงเส้น ดังรูปแบบตัวอย่างต่อไปนี้

2.7.1 รูปแบบทั่วไปของปัญหากำหนดการเชิงเส้น

$$\begin{aligned} & \text{Minimize} && f(x) = c^T x \\ & \text{Subject to} && Ax = b \\ & && x \geq 0 \end{aligned} \tag{2-51}$$

หรือกล่าวได้ว่ากำหนดการเชิงเส้นเป็นวิธีการทางคณิตศาสตร์ในการวิเคราะห์ปัญหาต่าง ๆ ภายใต้เงื่อนไขบังคับ ซึ่งเป็นการใช้สมการเชิงเส้นหรืออสมการเชิงเส้นสำหรับวางแผนเพื่อหาคำตอบของปัญหาหรือหาผลลัพธ์ที่ดีและเหมาะสมที่สุดได้แก่ การหาค่าสูงสุด (maximize) หรือการหาค่าต่ำสุด (minimize)

2.7.2 โครงสร้างของกำหนดการเชิงเส้น ประกอบด้วย

1. ตัวแปรที่ต้องตัดสินใจ (decision variable) คือ สิ่งที่ต้องการหาผลลัพธ์
2. ฟังก์ชันวัตถุประสงค์ (objective function) คือ จะต้องต้องมีเพียงวัตถุประสงค์เดียว ซึ่งอยู่ในรูปของเป้าหมายการหาค่าสูงสุดหรือการหาค่าต่ำสุด
3. เงื่อนไขบังคับ (constraints) คือ สมการหรืออสมการที่แสดงถึงขีดจำกัดในด้านความต้องการหรือเงื่อนไขต่าง ๆ ของปัญหา
4. ข้อจำกัด (restriction) คือ จะมีค่าตัวแปรที่ต้องตัดสินใจทุกตัวที่มีค่าไม่ติดลบ

2.7.3 รูปแบบของกำหนดการเชิงเส้นในกล่องเครื่องมือของโปรแกรมแมทแล็บ

มีฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับในรูปฟังก์ชัน ดังนี้

$$\begin{aligned}
 & \underset{x}{\text{Min}} && f^T x \\
 & \text{Subject to :} && A \cdot x \leq b \\
 & && Aeq \cdot x = beq \\
 & && lb \leq x \leq ub
 \end{aligned} \tag{2-52}$$

เมื่อค่า f b beq lb ub คือ เวกเตอร์ และค่า A Aeq คือ เมตริกซ์ โดยที่

ค่า f คือ เวกเตอร์สัมประสิทธิ์ของตัวแปรที่ต้องการหาผลลัพธ์

ค่า b คือ เวกเตอร์ของค่าคงที่ (ค่าคงที่ของสมการ)

ค่า beq คือ เวกเตอร์ของค่าคงที่ (ค่าคงที่ของสมการ)

ค่า lb คือ เวกเตอร์ของค่าขอบล่างของตัวแปรที่ต้องการหาผลลัพธ์

ค่า ub คือ เวกเตอร์ของค่าขอบบนของตัวแปรที่ต้องการหาผลลัพธ์

ค่า A คือ เมตริกซ์สัมประสิทธิ์ของตัวแปรในเงื่อนไขบังคับที่เป็นอสมการ

ค่า Aeq คือ เมตริกซ์สัมประสิทธิ์ของตัวแปรในเงื่อนไขบังคับที่เป็นสมการ

โครงสร้างของฟังก์ชันสำหรับหาค่าผลลัพธ์ในรูปแบบของกำหนดการเชิงเส้นในกล่องเครื่องมือของโปรแกรมแมทแล็บ เป็นดังสมการ (2-53)

$$[x] = \text{linprog}(f, A, b, Aeq, beq, lb, ub) \tag{2-53}$$

โดยที่ ค่า $A = []$ และค่า $b = []$ เมื่อไม่มีข้อจำกัดที่เป็นอสมการ

ค่า $Aeq = []$ และค่า $beq = []$ เมื่อไม่มีข้อจำกัดที่เป็นสมการ

2.8 โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface : GUI)

โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้เป็นส่วนติดต่อกับผู้ใช้ (user interface) ที่สร้างขึ้นด้วยวัตถุเชิงรูปภาพ (graphical object) แบบต่าง ๆ โดยทั่วไปผู้ใช้คอมพิวเตอร์ส่วนใหญ่เข้าใจถึงความหมายและวิธีใช้วัตถุเหล่านี้เป็นอย่างดีอยู่แล้ว (ปริญญ์ สงวนสัตย์, 2553)

วัตถุต่าง ๆ ที่ถูกออกแบบมาเป็นรูปแบบของ GUI จะพบว่าสามารถทำให้ผู้ใช้เข้าใจการใช้งานนั้นได้อย่างรวดเร็ว ช่วยเพิ่มความสมบูรณ์และความสะดวกให้แก่ผู้ใช้งานมากขึ้น ซึ่งในส่วนนี้จะใช้รูปแบบเครื่องมือช่วยในการเขียนภายในแมทเพล (Graphical User Interface Development Environment : GUIDE) เพื่อแสดงผลการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบทดสอบเมื่อเกิดการลัดวงจร ณ บัสใดบัสหนึ่งในระบบทดสอบ รวมถึงใช้แสดงผลลัพธ์ของการประสานสัมพันธ์รีเลย์ของระบบทดสอบต่าง ๆ ร่วมด้วย

2.8.1 คุณสมบัติของวัตถุส่วนติดต่อผู้ใช้ (user interface object)

วัตถุต่าง ๆ ที่สร้างขึ้นใน GUI เป็นวัตถุส่วนติดต่อผู้ใช้ มีหน้าที่แตกต่างกันไปตามคุณสมบัติของวัตถุ แสดงส่วนประกอบของ GUI ดังรูปที่ 2.16

1. check boxes จะมีการกำหนดค่าให้มีการทำงานเมื่อเราเลือกให้ค่าในกล่องอุปกรณ์นี้มีประโยชน์เพื่อให้ผู้ใช้เลือกหัวข้อการทำงานของโปรแกรมแบบต่าง ๆ ได้อย่างอิสระ การที่จะให้วัตถุ check box ทำงานเราจะใช้เมาส์กดไปที่บริเวณกล่องนั้น สถานะของการเลือกหรือไม่เลือกค่าตามที่กำหนดจะแสดงขึ้น

2. edit text boxes เป็นกล่องที่ผู้ใช้สามารถที่จะแก้ไขตัวอักษรที่บรรจุอยู่ภายในกล่องนั้นได้ เราจะใช้วัตถุ edit text เมื่อเราต้องการให้ผู้ใช้กำหนดค่าตัวอักษรเป็น input

3. axes เหมาะสำหรับการใช้ในการแสดงกราฟิก เช่น กราฟ ภาพ หรือวิดีโอ

4. list boxes เป็นการแสดงรายการที่ผู้เขียนโปรแกรมกำหนดขึ้น ซึ่งจะกำหนดโดยคุณสมบัติของ string และจะยอมให้ผู้ใช้สามารถที่จะเลือกรายการนั้นได้ที่ละหนึ่งรายการหรือมากกว่า

5. pop-up menus เป็นการเปิดรายการของตัวเลือกขึ้นหลังจากที่มีการกดเมาส์บริเวณเมนู เพื่อให้ผู้ใช้ได้เลือกรายการใดรายการหนึ่ง ซึ่งรายการของตัวเลือกจะกำหนดด้วยคุณสมบัติ String หากว่าเมนูนี้ไม่ได้เปิดขึ้น ค่าในเมนูจะแสดงค่าคุณสมบัติปัจจุบัน pop-up menus นี้มีประโยชน์ในการที่จะให้ผู้ใช้ได้เลือกตัวเลือกตามที่ต้องการ

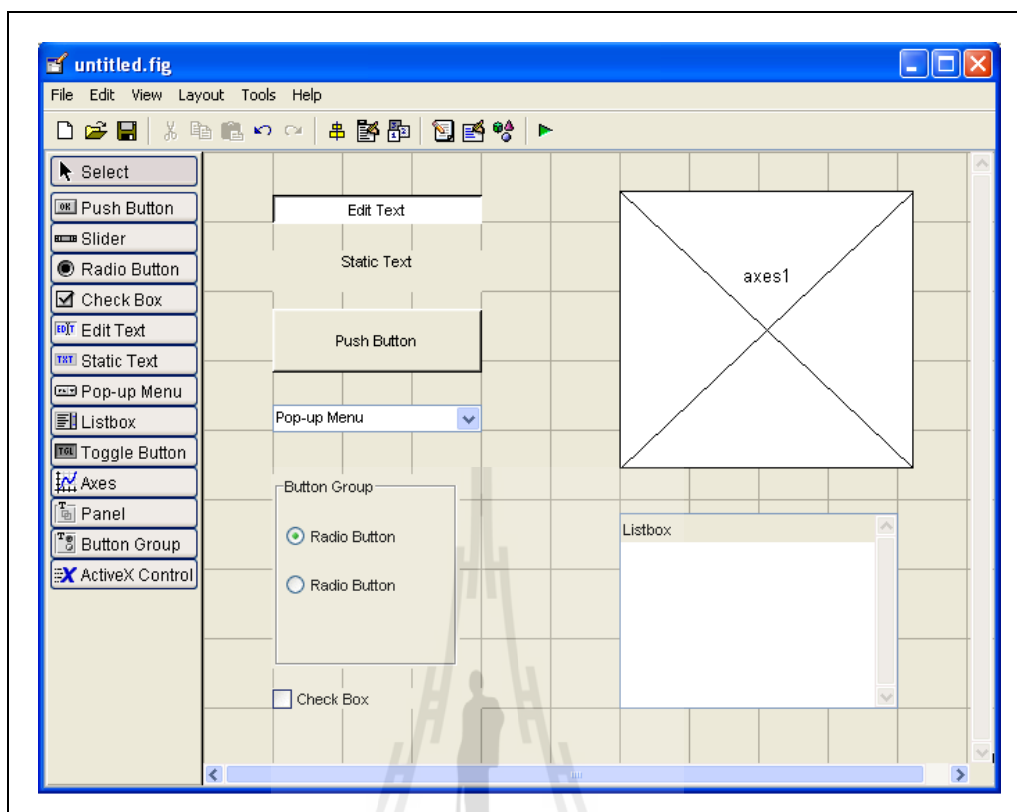
6. push buttons จะทำให้เกิดมีคำสั่งอื่น ๆ ตามมาหลังจากที่ผู้ใช้เลือกกดปุ่มนี้ โดยทั่วไปผู้ใช้จะใช้เมาส์ในการกดปุ่ม push buttons

7. radio buttons จะมีลักษณะการทำงานคล้ายกับ check boxes แต่ข้อแตกต่างสำคัญก็คือโดยทั่วไปแล้ว การใช้ radio button นี้เราจะจัดให้เป็นกลุ่มของตัวเลือกและในกลุ่มของ radio button นั้นจะมีตัวเลือกที่สามารถถูกเลือกได้เพียงครั้งละตัวเดียวในกลุ่มนั้น ในการที่จะให้ radio button ทำงานเราจะใช้เมาส์กดไปที่วัตถุนั้น จากนั้นสภาพของอุปกรณ์จะเปลี่ยนไปตามสภาพที่เราเลือก ผู้เขียน โปรแกรมจะต้องเป็นผู้กำหนดพฤติกรรมหลังจากที่มีการกดเมาส์ลงไป ณ วัตถุนั้น

8. sliders เป็นการป้อนค่าตัวเลขโดยอาศัยแถบเลื่อนนี้ ค่าที่ป้อนจะถูกกำหนดโดยผู้เขียนโปรแกรมและผู้ใช้งานทำการกำหนดค่าโดยการให้เมาส์กดแล้วเลื่อนแถบบน slider หรือใช้เมาส์กดบริเวณลูกศรเพื่อให้แถบค่อย ๆ เลื่อนไป ตำแหน่งของตัวเลื่อนบนแท่งจะเป็นค่าตัวเลขตามสัดส่วนของระยะบนแท่งเลื่อนนี้และค่าจะได้รับการคำนวณหลังจากที่เราปล่อยปุ่มเมาส์ เราสามารถที่จะตั้งค่าสูงสุด ค่าสุดและค่าปัจจุบันของ slider ได้

9. static text boxes หมายถึงตัวหนังสือที่ผู้ใช้ไม่สามารถที่จะแก้ไขได้ แต่เราซึ่งเป็นผู้เขียนโปรแกรมอาจแก้ไขได้ ดังนั้นโดยทั่วไปเราจึงใช้ในการเขียนตัวหนังสือเพื่อบอกถึงชื่อของส่วนต่าง ๆ หรืออาจเป็นการบอกค่าของอุปกรณ์ควบคุมบางอย่างหรือบอกถึงสถานะการทำงานของโปรแกรมรวมถึงการบอกวิธีการใช้ GUI ของเราอย่างคร่าว ๆ เป็นต้น และเมื่อผู้ใช้ไม่สามารถเปลี่ยนตัวอักษรใน static text ได้อย่าง interactive ดังนั้นเราจึงไม่สามารถที่จะเรียก callback ของ static text มาใช้ได้

10. toggle buttons เป็นปุ่มที่ทำหน้าที่เหมือนสวิตช์ไฟคือ จะมีค่าเป็น on หรือ off และเมื่อผู้ใช้เปลี่ยนค่ามันโดยการกดเมาส์ลงไป ในบริเวณของวัตถุนั้น จะเป็นการเรียก callback ของวัตถุให้ทำงานไปพร้อมกัน ซึ่ง toggle buttons มีประโยชน์ในการสร้าง toolbars



รูปที่ 2.16 ส่วนประกอบของ GUI

2.9 สรุป

ในส่วนของทฤษฎีที่เกี่ยวข้องนี้กล่าวถึงทฤษฎีความรู้ในเรื่องของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางซึ่งมีฟังก์ชันการทำงานเช่นเดียวกับรีเลย์ป้องกันกระแสเกิน แต่เพิ่มในส่วนของการเปรียบเทียบเฟสหรือทิศทางการไหลของกำลังไฟฟ้า จึงเรียกว่ารีเลย์ป้องกันกระแสเกินแบบมีทิศทาง สมการความไวของรีเลย์ ซึ่งในงานวิจัยนี้ได้ใช้สมการความไวแบบเวลาผกผันมาตรฐาน เนื่องจากนิยมใช้ในระบบจำหน่ายและระบบส่งจ่ายทุกระดับแรงดัน ไม่ว่าจะเป็นรีเลย์หลักหรือรีเลย์สำรอง การปรับตั้งค่าของรีเลย์ทั้งในส่วนของการปรับตั้ง (pickup setting) และค่าจำนวนเท่าของเวลาในการทำงานของรีเลย์ (TDS) ช่วงเวลาเพื่อของรีเลย์หลักและรีเลย์สำรองที่เหมาะสม รวมถึงทฤษฎีของการลัดวงจรในระบบไฟฟ้ารูปแบบต่าง ๆ เช่น การลัดวงจรแบบ 3 เฟส การลัดวงจรแบบ เฟส-ดิน การลัดวงจรแบบ เฟส-เฟส และการลัดวงจรแบบ เฟส-เฟส-ดิน ซึ่งในงานวิจัยนี้ได้เลือกใช้การลัดวงจรแบบ 3 เฟสสมมาตร มาเป็นข้อมูลในการพิจารณาหาผลลัพธ์เพื่อการปรับตั้งค่าการทำงานของรีเลย์ เนื่องจากการลัดวงจรชนิดนี้ทำให้เกิดขนาดของกระแสลัดวงจรไหลรุนแรงมากกว่าการลัดวงจรประเภทอื่น ๆ ซึ่งมีผลต่อฟิสิกส์ของอุปกรณ์ต่าง ๆ ในระบบส่งจ่าย

จึงจำเป็นต้องนำมาพิจารณาในการปรับตั้งรีเลย์ โดยการปรับตั้งรีเลย์ได้นำวิธีกำหนดการเชิงเส้นมาใช้ในการหาค่าจำนวนเท่าของเวลาการทำงานของรีเลย์และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุด และผู้วิจัยได้นำโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) มาใช้ในการแสดงผลพิกซ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์และค่าของฟังก์ชันวัตถุประสงค์ พร้อมกับแสดงผลพิกซ์ของรีเลย์หลักและรีเลย์สำรองในการป้องกันการลัดวงจรที่เกิดขึ้น เพื่อความสะดวกและเข้าใจง่ายในการใช้งานของทั้งผู้ใช้งานและผู้วิจัยเอง ในส่วนของรายละเอียดขั้นตอนต่าง ๆ ของการประสานสัมพันธ์รีเลย์ป้องกันการกระแสนเกินแบบมิติศทางโดยใช้กำหนดการเชิงเส้นจะกล่าวต่อไปในบทที่ 3



บทที่ 3

การประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยใช้กำหนดการเชิงเส้น

3.1 บทนำ

จากที่ได้กล่าวมาแล้วในบทที่ 2 เพื่อป้องกันความเสียหายต่ออุปกรณ์ไฟฟ้าในระบบและอันตรายที่อาจเกิดขึ้นกับสิ่งมีชีวิตเมื่อเกิดการลัดวงจรขึ้น การประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางอย่างมีประสิทธิภาพมีความจำเป็นอย่างยิ่งในการป้องกันการลัดวงจรและเพื่อให้รีเลย์สามารถทำงานได้อย่างเหมาะสมทั้งในสภาวะปกติและสภาวะที่เกิดการลัดวงจรต่าง ๆ จึงต้องปรับตั้งค่า TDS ซึ่งเป็นค่าสำหรับการทำงานของรีเลย์ โดยใช้วิธีกำหนดการเชิงเส้นมาแก้ปัญหาในการหาค่า TDS ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางแต่ละตัว ซึ่งผลการศึกษจะเป็นประโยชน์ในการปรับตั้งค่าการทำงานของรีเลย์ดังกล่าวได้อย่างเหมาะสมและมีประสิทธิภาพ รวมทั้งเข้าใจกระบวนการของการประสานสัมพันธ์รีเลย์ เพื่อนำมาพัฒนาและประยุกต์ใช้ในระบบไฟฟ้ากำลังจริงต่อไป

ในบทที่ 3 นี้ ได้ทำการอธิบายขั้นตอนและวิธีการประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง ซึ่งวิธีในการดำเนินงานวิจัยแบ่งออกเป็น 4 หัวข้อหลักประกอบด้วย 1.การเตรียมข้อมูลของระบบทดสอบ 2.การเขียนโปรแกรมและรันข้อมูล 3.การระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง 4.การเขียนโปรแกรมส่วนต่อประสานกราฟิกสำหรับแสดงผล จากหัวข้อดังกล่าวข้างต้นมีรายละเอียดดังต่อไปนี้

3.2 การเตรียมข้อมูลของระบบทดสอบ

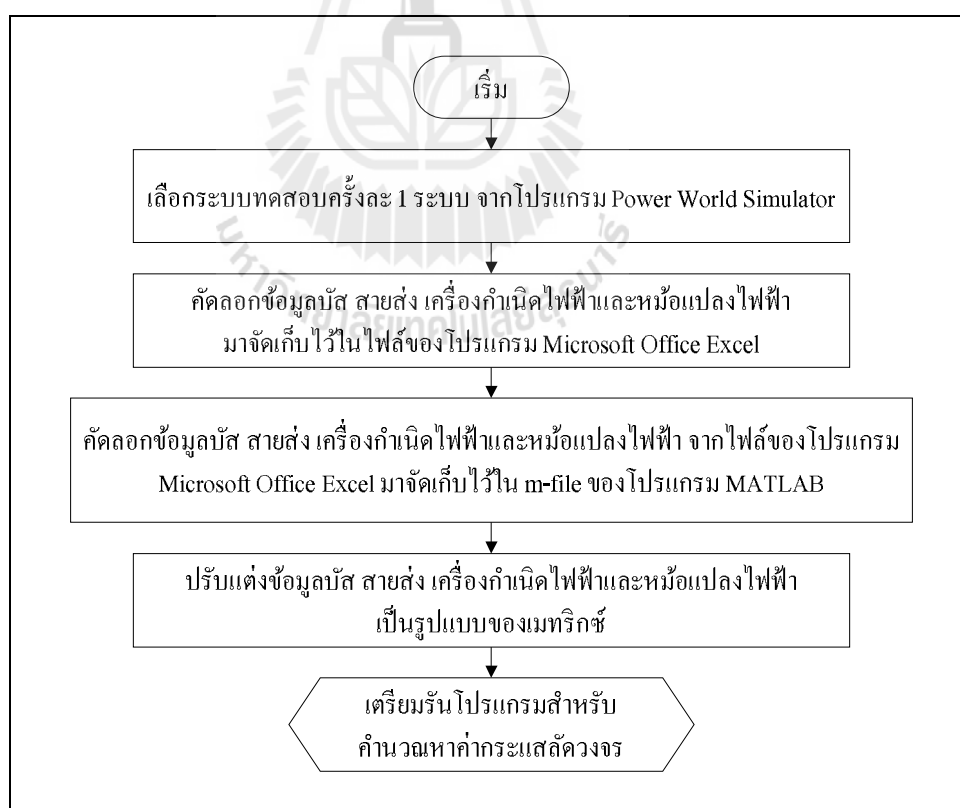
ระบบทดสอบที่นำมาทดสอบเป็นระบบที่มีอยู่ในโปรแกรม Power world simulator ซึ่งในงานวิจัยนี้ได้เลือกระบบสำหรับการทดสอบจำนวน 9 ระบบ คือ

- | | | |
|--------------------|---------------------|------------------------|
| 1. ระบบทดสอบ 3 บัส | 4. ระบบทดสอบ 8 บัส | 7. ระบบทดสอบ 4700 บัส |
| 2. ระบบทดสอบ 6 บัส | 5. ระบบทดสอบ 9 บัส | 8. ระบบทดสอบ 8166 บัส |
| 3. ระบบทดสอบ 7 บัส | 6. ระบบทดสอบ 30 บัส | 9. ระบบทดสอบ 14454 บัส |

3.2.1 ขั้นตอนการเตรียมข้อมูลของระบบทดสอบ

1. เลือกระบบทดสอบในโปรแกรม Power world simulator ครั้งละ 1 ระบบ
2. คัดลอกข้อมูลของระบบทดสอบได้แก่ ข้อมูลของสายส่ง ข้อมูลของเครื่องกำเนิดไฟฟ้า ข้อมูลของหม้อแปลงไฟฟ้าและข้อมูลของบัส โดยจัดเก็บข้อมูลไว้ในรูปแบบของไฟล์ excel ของโปรแกรม Microsoft Office Excel
3. คัดลอกข้อมูลของสายส่ง ข้อมูลเครื่องกำเนิดไฟฟ้า ข้อมูลหม้อแปลงไฟฟ้าและข้อมูลของบัสจากไฟล์ excel ในขั้นตอนที่ 2 มาจัดเก็บข้อมูลไว้ใน m-file ของโปรแกรมแมทแลปที่สร้างไว้สำหรับแต่ละระบบทดสอบและปรับแต่งข้อมูลเป็นรูปแบบของเมทริกซ์
4. เลือกระบบทดสอบที่เหลือแล้วทำตามขั้นตอนที่ 2 และ 3 จนครบทั้ง 9 ระบบ
5. เตรียมรันโปรแกรมเพื่อหาค่ากระแสลัดวงจรด้วยฟังก์ชันที่สร้างขึ้น

สรุปขั้นตอนการเตรียมข้อมูลของระบบทดสอบแสดงดังรูปที่ 3.1 และรายละเอียดของข้อมูลของสายส่ง ข้อมูลของเครื่องกำเนิดไฟฟ้า ข้อมูลของหม้อแปลงไฟฟ้าและข้อมูลบัสของระบบทดสอบที่ใช้สำหรับการรันโปรแกรมได้แสดงไว้ในภาคผนวก ก



รูปที่ 3.1 ขั้นตอนการเตรียมข้อมูลของระบบทดสอบ

3.3 การเขียนโปรแกรมและรันข้อมูล

การเขียนโปรแกรมแมทแลป (เฟล็กซ์ เฟอร์อะ, 2550) มีฟังก์ชันหลายฟังก์ชันที่สร้างขึ้นเพื่อใช้ในการรันโปรแกรมหาค่าผลลัพธ์ต่าง ๆ เช่น การสร้างเมทริกซ์แสดงตำแหน่งของรีเลย์ การสร้างเมทริกซ์ของฟังก์ชันวัตถุประสงค์ การสร้างเมทริกซ์ของฟังก์ชันเงื่อนไข การเลือกขนาดของหม้อแปลงกระแสและการรันโปรแกรมด้วยกำหนดการเชิงเส้น เป็นต้น รายละเอียดของฟังก์ชันต่าง ๆ ที่ใช้สำหรับการรันโปรแกรมได้แสดงไว้ในภาคผนวก ข ซึ่งก่อนการเขียนโปรแกรมหากล่าวนั้นจะต้องทราบพื้นฐานความรู้ของปัญหาและรูปแบบของข้อกำหนดต่าง ๆ เพื่อใช้ประกอบในการเขียนโปรแกรม ซึ่งมีรายละเอียดดังต่อไปนี้

3.3.1. รูปแบบสมการของปัญหา

สมการทางคณิตศาสตร์ของปัญหาในการประสานสัมพันธ์การทำงานของรีเลย์จะใช้รูปแบบสมการความไวของเส้นโค้งเวลาแบบเวลาผกผันมาตรฐาน (standard inverse time) โดยมีสมการดังสมการที่ (3.1) ซึ่งหาค่าของเวลาการทำงานของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง (i) ประกอบด้วยค่าจำนวนเท่าของเวลาการทำงานของรีเลย์หรือตัวคูณเวลา (Time Dial Setting : TDS) ค่าปรับคูณกระแสทำงาน (Plug Setting : PS) ค่ากระแสลัดวงจร (fault current : I_f) และค่ากระแสไหลอดปกติที่ไหลผ่านหม้อแปลงกระแสด้านทุติยภูมิ (CT_{pri_rating})

$$t = \frac{\alpha \times TDS}{\left(\frac{I_f}{PS \times CT_{pri_rating}} \right)^\beta - \gamma} \quad (3.1)$$

จากสมการจะมีเพียงค่า TDS เท่านั้นที่ไม่ทราบค่า ซึ่งจะใช้ในการปรับตั้งค่าการทำงานของรีเลย์ สัญลักษณ์ที่อยู่ในสมการได้แก่ α β และ γ คือค่าคงที่ของรีเลย์ที่ใช้เส้นโค้งเวลาแบบเวลาผกผันมาตรฐาน มีค่าดังนี้คือ $\alpha = 0.14$ $\beta = 0.02$ และ $\gamma = 1$ ส่วนค่า PS คือค่าปรับคูณของกระแสทำงาน ตัวปรับคูณที่นิยมใช้คือการยอมให้เกิดไหลอดเกินได้ไม่เกิน 25% ดังนั้นค่า PS จึงมีค่าเท่ากับ 1.25 และขนาดอัตราส่วนของหม้อแปลงกระแสที่ใช้พิจารณาในงานวิจัยนี้ กำหนดให้หม้อแปลงกระแสมีพิกัดด้านทุติยภูมิเท่ากับ 5 แอมแปร์

3.3.2 ฟังก์ชันวัตถุประสงค์ของปัญหาการประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง

ปัญหาการประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง จะพิจารณาในส่วนของรีเลย์หลักเท่านั้น เนื่องจากเป็นรีเลย์ที่มีหน้าที่ป้องกันการลัดวงจรที่เกิดขึ้นในระบบไฟฟ้าเป็นอันดับแรก โดยรีเลย์ที่อยู่ใกล้กับตำแหน่งที่เกิดการลัดวงจรจะเรียกว่า รีเลย์หลักใกล้จุดลัดวงจร ส่วนรีเลย์ที่อยู่ไกลจากตำแหน่งที่เกิดการลัดวงจรนั้นจะเรียกว่า รีเลย์หลักไกลจุดลัดวงจร จะได้ฟังก์ชันวัตถุประสงค์ดังสมการ (3.2) (Thangaraj, Pant, and Deep, 2010)

$$\text{Objective function} = \sum_{i=1}^{N_{cl}} T_{pri_cl_in}^i + \sum_{j=1}^{N_{far}} T_{pri_far_bus}^j \quad (3.2)$$

$$T_{pri_cl_in}^i = \frac{0.14 \times TDS^i}{\left(\frac{a^i}{PS^i \times b^i} \right)^{0.02} - 1} \quad (3.3)$$

$$T_{pri_far_bus}^j = \frac{0.14 \times TDS^j}{\left(\frac{c^j}{PS^j \times d^j} \right)^{0.02} - 1} \quad (3.4)$$

เมื่อ $T_{pri_cl_in}^i$ คือ เวลาทำงานของรีเลย์หลักที่อยู่ใกล้จุดลัดวงจร

$T_{pri_far_bus}^j$ คือ เวลาทำงานของรีเลย์หลักที่อยู่ไกลจุดลัดวงจร

N_{cl} คือ จำนวนของรีเลย์ที่อยู่ใกล้จุดลัดวงจร

N_{far} คือ จำนวนของรีเลย์ที่อยู่ไกลจุดลัดวงจร

a และ c คือ พารามิเตอร์ค่ากระแสลัดวงจรที่ผ่านหม้อแปลงกระแส

b และ d คือ พารามิเตอร์ค่ากระแสโหลดที่ผ่านหม้อแปลงกระแส

3.3.3 เงื่อนไขบังคับของปัญหาการประสานสัมพันธ์ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง

$$1. TDS_{min}^i \leq TDS^i \leq TDS_{max}^i \quad (3.5)$$

ค่าขอบล่าง $TDS_{min}^i = 0.05$ และค่าขอบบน $TDS_{max}^i = 1.1$

ค่า i คือหมายเลขของรีเลย์ เริ่มต้นจากหมายเลข 1 ถึงหมายเลขรีเลย์ตัวสุดท้าย

$$2. 1.25 \leq PS^i \leq 1.5 \quad (3.6)$$

ค่าขอบล่าง $PS_{\min}^i = 1.25$ และค่าขอบบน $PS_{\max}^i = 1.5$

(ในส่วนของค่า PS นี้ เมื่อใช้วิธีกำหนดการเชิงเส้น จะกำหนดให้ $PS = 1.25$)

$$3. T_{primary} \text{ มีค่าอยู่ระหว่าง } 0.05 \text{ ถึง } 1.00 \text{ วินาที}$$

(ค่า $T_{primary}$ อาจปรับเปลี่ยนค่าตามความเหมาะสมของระบบทดสอบ)

$$4. T_{backup} - T_{primary} - CTI \geq 0 \quad (3.7)$$

เมื่อสมการเวลาของ T_{backup} และสมการ $T_{primary}$ เป็นดังสมการ (3.8) - (3.9)

$$T_{backup}^i = \frac{0.14 \times TDS^p}{\left(\frac{e^i}{PS^i \times f^i} \right)^{0.02} - 1} \quad (3.8)$$

$$T_{primary}^i = \frac{0.14 \times TDS^q}{\left(\frac{g^i}{PS^i \times h^i} \right)^{0.02} - 1} \quad (3.9)$$

T_{backup} คือ เวลาการทำงานของรีเลย์สำรอง

$T_{primary}$ คือ เวลาการทำงานของรีเลย์หลัก

e และ g คือ พารามิเตอร์ของค่ากระแสลัดวงจรที่ผ่านหม้อแปลงกระแส

f และ h คือ พารามิเตอร์ของค่ากระแสโหลดปกติที่ผ่านหม้อแปลงกระแส

CTI คือ เวลาส่วนเผื่อของการประสานสัมพันธ์รีเลย์มีค่าเท่ากับ 0.3 วินาที

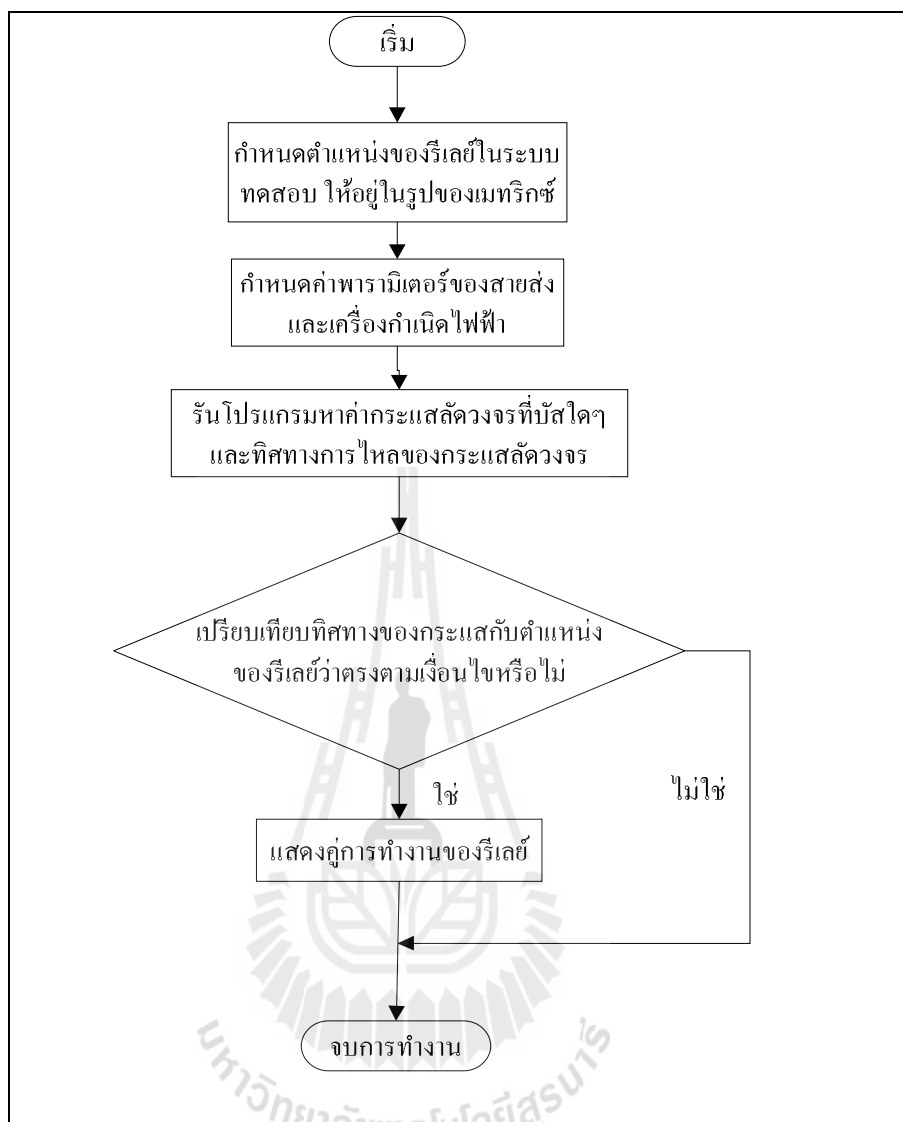
(CTI ย่อมาจาก Coordinating Time Interval)

3.4 การระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง

ในส่วนนี้ได้สร้างฟังก์ชันของโปรแกรมเมทแลปมาใช้ในการแก้ปัญหาเพื่อหาคู่การทำงานของรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ ในระบบทดสอบ โดยมีขั้นตอนการทำงานดังต่อไปนี้

1. กำหนดตำแหน่งของรีเลย์ที่อยู่ในระบบทดสอบ โดยเขียนฟังก์ชันของโปรแกรมเมทแลป เพื่อแสดงตำแหน่งของรีเลย์ทุกตัวและเก็บข้อมูลตำแหน่งของรีเลย์ทุกตัวให้อยู่ในรูปแบบของเมทริกซ์

2. กำหนดค่าพารามิเตอร์ของสายส่ง เช่น ความต้านทาน รีแอक्टแดนซ์ ค่าการอัดประจุครึ่งสาย (half-line charging : $B/2$) แท๊ปหม้อแปลงและพารามิเตอร์ของเครื่องกำเนิดไฟฟ้า เช่น ความต้านทานและรีแอक्टแดนซ์ โดยค่าพารามิเตอร์ต่าง ๆ นั้นคัดลอกมาจากระบบทดสอบของโปรแกรม Power World Simulator ซึ่งมีการกำหนดค่าพารามิเตอร์ในแต่ละระบบทดสอบไว้เรียบร้อยแล้วและเก็บข้อมูลของพารามิเตอร์ต่าง ๆ ให้อยู่ในรูปแบบของเมทริกซ์ เพื่อดำเนินการสำหรับการรันโปรแกรมหาค่ากระแสลัดวงจรของระบบทดสอบในขั้นตอนถัดไป
3. เขียนฟังก์ชันเมทแพลสำหรับหาค่าของกระแสลัดวงจรในระบบทดสอบ เมื่อเกิดการลัดวงจร ณ บัสต่าง ๆ เพื่อหาขนาดและทิศทางกระแสลัดวงจรในระบบทดสอบ
4. รันโปรแกรมสำหรับหาค่าของกระแสลัดวงจรในระบบทดสอบ จากนั้นเปรียบเทียบทิศทางกระแสลัดวงจรเมื่อเกิดการลัดวงจร ณ บัสใด ๆ กับตำแหน่งของรีเลย์ว่ารีเลย์ที่ตำแหน่งบัสที่เกิดการลัดวงจรนั้นซึ่งมีหน้าที่เป็นรีเลย์หลักจะมีรีเลย์ตัวใดบ้างที่ทำหน้าที่เป็นรีเลย์สำรอง โดยพิจารณาทิศทางกระแสลัดวงจรที่ผ่านรีเลย์ว่ามีทิศทางไหลของกระแสลัดวงจรมายังบัสที่เกิดการลัดวงจรหรือไม่ ถ้าใช่แสดงว่ารีเลย์ตัวนั้นทำหน้าที่เป็นรีเลย์สำรองให้กับรีเลย์หลัก ณ บัสนั้น ๆ โดยการเปรียบเทียบเงื่อนไขดังที่กล่าวข้างต้น ได้ใช้ฟังก์ชันของโปรแกรมเมทแพลที่สร้างขึ้นสำหรับการพิจารณาเงื่อนไขในส่วนนี้
5. แสดงผลลัพธ์การทำงานของรีเลย์หลักและรีเลย์สำรอง โดยขั้นตอนการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง แสดงดังรูปที่ 3.2



รูปที่ 3.2 ขั้นตอนการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง

3.5 โปรแกรมส่วนต่อประสานกราฟิกสำหรับแสดงผล

ในส่วนของการแสดงผลที่ได้แก่ การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรอง ค่า TDS ของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางและค่าของฟังก์ชันวัตถุประสงค์ในระบบทดสอบต่าง ๆ ได้ใช้โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) มาช่วยในการแสดงผลที่ต้องการ เพื่อเพิ่มความครบถ้วนในการแสดงผลและความสะดวกในการใช้งานบนหน้าต่างของ GUI เพียงหน้าต่างเดียว โดยได้ออกแบบหน้าต่างการทำงานของ GUI แสดงดังรูปที่ 3.3 ซึ่งรายละเอียดส่วนประกอบต่าง ๆ บนหน้าต่างโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ปรากฏอยู่ ได้อธิบายถึงการใช้งานในหัวข้อย่อยลำดับถัดไป

3.5.1 หน้าต่างการแสดงผลด้วยโปรแกรมส่วนต่อประสานกราฟิก

จากคุณสมบัติของวัตถุส่วนติดต่อผู้ใช้ที่ได้กล่าวมาแล้วเบื้องต้นในบทที่ 2 ในส่วนนี้ได้ออกแบบหน้าต่างโปรแกรมสำหรับแสดงผลลัพธ์ที่ต้องการของระบบทดสอบต่าง ๆ ได้แก่ คู่มือการทำงานของรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรที่บัสใด ๆ ในระบบทดสอบ ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์และค่าฟังก์ชันวัตถุประสงคที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ ผู้วิจัยได้ออกแบบหน้าต่างการแสดงผลลัพธ์เพื่อให้ผู้ใช้งานสามารถใช้งานได้อย่างสะดวกและเข้าใจได้ง่าย แสดงดังรูปที่ 3.3 และเพื่อให้มีความเข้าใจถึงกระบวนการในการแสดงผลลัพธ์ซึ่งมีส่วนสำคัญอยู่ 4 ส่วนบนหน้าต่างของโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ สามารถอธิบายได้ดังรายละเอียดต่อไปนี้

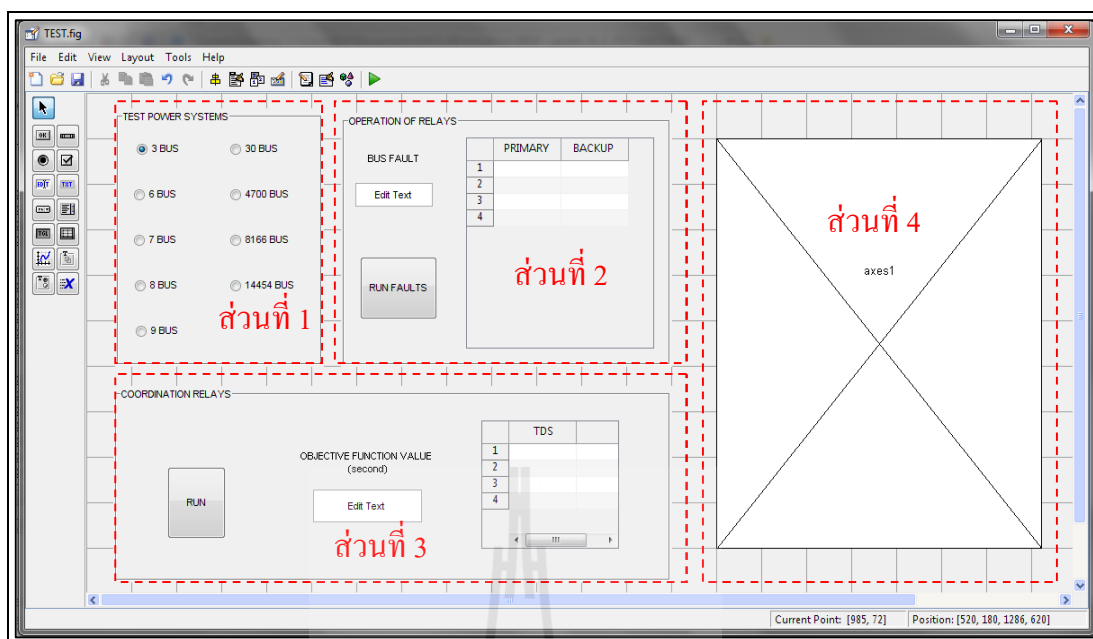
3.5.1.1 ส่วนที่ 1 เป็นส่วนสำหรับเลือกระบบทดสอบที่ต้องการทราบค่าผลลัพธ์ที่ต้องการ สามารถเลือกได้โดยใช้เมาส์คลิกเลือกที่วงกลมของระบบทดสอบแต่ละระบบทดสอบ (เลือกได้เพียงครั้งละ 1 ระบบทดสอบเท่านั้น)

3.5.1.2 ส่วนที่ 2 เป็นส่วนสำหรับแสดงผลลัพธ์ของการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ ของระบบทดสอบที่ได้เลือกไว้ในส่วนที่ 1 สามารถทดสอบได้โดยใส่หมายเลขบัสใดบัสหนึ่งในช่องว่างของ BUS FAULT (หมายเลขบัสที่ใส่ในช่องว่างจะมีค่าอยู่ในช่วงหมายเลข 1 จนถึงหมายเลขของระบบทดสอบที่เลือกไว้) หลังจากนั้นคลิกที่ปุ่ม RUN FAULTS ผลลัพธ์การทำงานของรีเลย์หลักและรีเลย์สำรองจะแสดงในตารางด้านขวามือ

3.5.1.3 ส่วนที่ 3 เป็นส่วนสำหรับแสดงผลลัพธ์ของค่าฟังก์ชันวัตถุประสงคที่น้อยที่สุดและค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ในระบบทดสอบที่เลือกไว้ในส่วนที่ 1 สามารถทดสอบได้โดยการคลิกที่ปุ่ม RUN ค่าฟังก์ชันวัตถุประสงคที่น้อยที่สุดจะปรากฏขึ้นในช่องว่างของ OBJECTIVE FUNCTION VALUE ส่วนค่าจำนวนเท่าของเวลาการทำงานของรีเลย์แต่ละตัวหรือค่า TDS สามารถดูได้จากตารางด้านขวามือ

3.5.1.4 ส่วนที่ 4 เป็นส่วนสำหรับแสดงรูปภาพของระบบทดสอบที่ได้เลือกไว้จากส่วนที่ 1

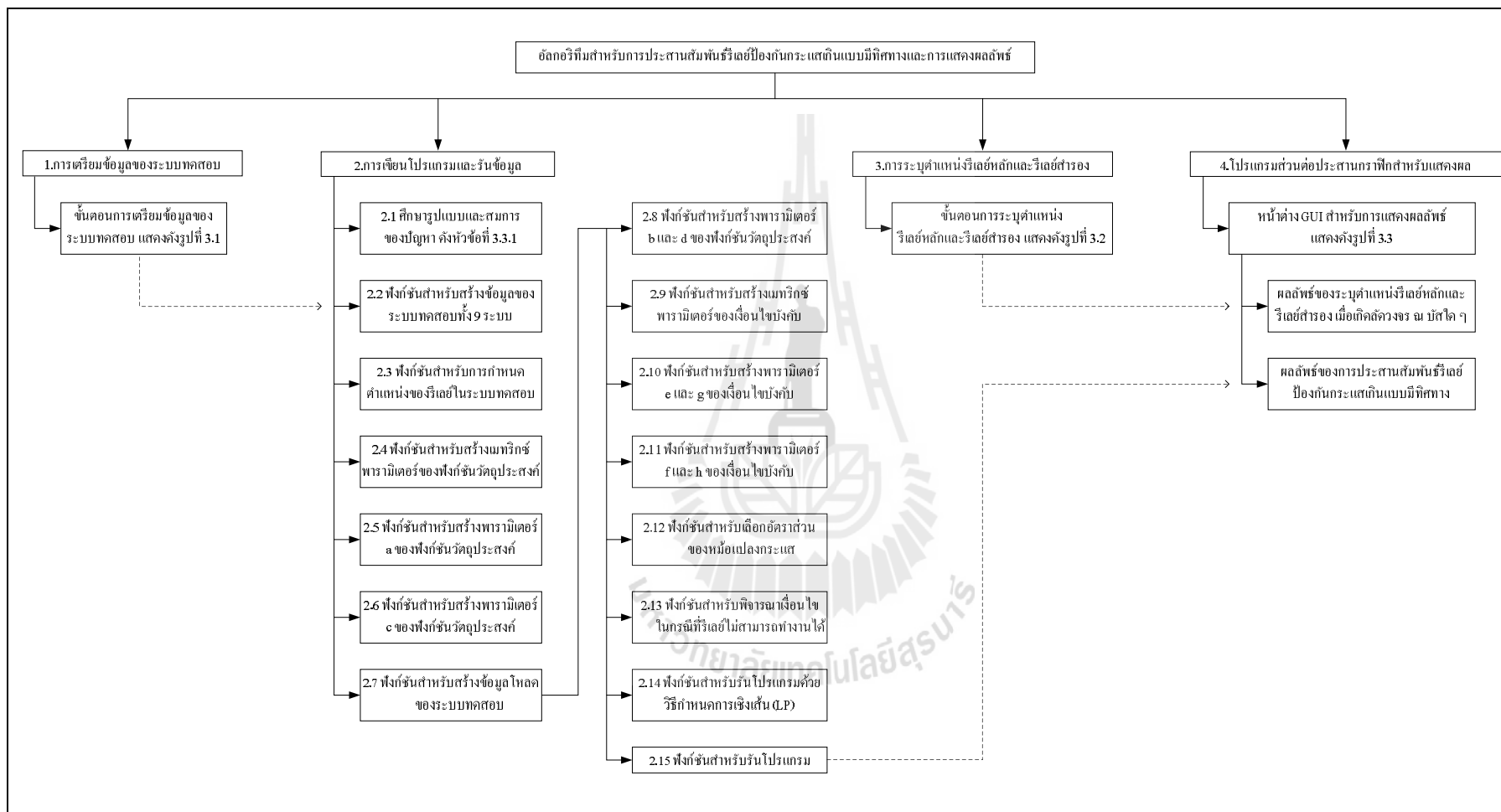
หน้าต่างการแสดงผลลัพธ์ของ GUI และส่วนสำคัญของกระบวนการแสดงผลลัพธ์ดังกล่าวข้างต้น เป็นเพียงการออกแบบหน้าต่างสำหรับแสดงผลลัพธ์ที่ต้องการเท่านั้น เพื่อให้หน้าต่างการแสดงผลสามารถแสดงผลลัพธ์ได้อย่างถูกต้องและสมบูรณ์นั้น ต้องดำเนินการเขียนโปรแกรมเพื่อกำหนดและควบคุมการทำงานของแต่ละวัตถุส่วนติดต่อผู้ใช้ที่ได้ออกแบบไว้ โดยรายละเอียดของโปรแกรมได้แสดงไว้ในภาคผนวก ข



รูปที่ 3.3 ออกแบบการแสดงผลด้วย GUI

3.6 ภาพรวมของการสร้างอัลกอริทึมการประสานสัมพัทธ์เลยป้องกันกระแสเกินแบบมีทิศทาง

จากวิธีการดำเนินงานวิจัยใน 4 หัวข้อหลักที่นำเสนอตามที่กล่าวมาข้างต้น เพื่อให้สามารถมองเห็นภาพรวมทั้งหมดของลำดับในการสร้างอัลกอริทึมการประสานสัมพัทธ์เลยป้องกันกระแสเกินแบบมีทิศทางในงานวิจัยนี้ จึงได้แสดงภาพรวมของงานวิจัย แสดงดังรูปที่ 3.4



รูปที่ 3.4 ภาพรวมของอัลกอริทึมการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง

3.6.1 ขั้นตอนในการสร้างอัลกอริทึมการประสานสัมพันธ์รีเลย์

จากรูปที่ 3.4 แสดงถึงภาพรวมของอัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ ป้องกันกระแสเกินแบบมีทิศทางในงานวิจัยนี้ มีขั้นตอนสำคัญอยู่ 4 ส่วน คือ

1. การเตรียมข้อมูลของระบบทดสอบ เป็นขั้นตอนสำหรับเตรียมข้อมูลของระบบทดสอบจากระบบทดสอบที่มีข้อมูลอยู่ใน โปรแกรม Power World Simulator และคัดลอกข้อมูลที่เป็นสำเนาไว้ในการรันโปรแกรมมาไว้ในไฟล์ของโปรแกรมแมทแลปในรูปแบบของเมทริกซ์ แสดงขั้นตอนการเตรียมข้อมูลของระบบทดสอบดังรูปที่ 3.1

2. การเขียนโปรแกรมและรันข้อมูล เป็นการสร้างฟังก์ชันต่าง ๆ ของโปรแกรมแมทแลปสำหรับการสร้างอัลกอริทึมการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง (รายละเอียดโปรแกรมของฟังก์ชันต่าง ๆ ได้แสดงไว้ในภาคผนวก ข) แสดงลำดับในการสร้างฟังก์ชันดังนี้

2.1 ศึกษารูปแบบและสมการของปัญหา ดังหัวข้อที่ 3.3.1 เพื่อทราบถึงฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับของปัญหา

2.2 ฟังก์ชันข้อมูลของระบบทดสอบ ได้ข้อมูลมาจากขั้นตอนการเตรียมข้อมูลของระบบทดสอบ มีฟังก์ชันข้อมูลของระบบทดสอบจำนวน 9 ฟังก์ชัน ประกอบด้วย

2.2.1 B3LP_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 3 บัส
2.2.2 bus6naps_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 6 บัส
2.2.3 B7flatlp_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 7 บัส
2.2.4 bus8_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 8 บัส
2.2.5 bus9_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 9 บัส
2.2.6 buslp30_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 30 บัส
2.2.7 P99sp_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 4700 บัส
2.2.8 LS1A98_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 8166 บัส
2.2.9 Wisconsin_pwb	คือ ฟังก์ชันข้อมูลของระบบทดสอบ 14454 บัส

2.3 ฟังก์ชันสำหรับการกำหนดตำแหน่งของรีเลย์ในระบบทดสอบ ฟังก์ชันที่ใช้คือฟังก์ชัน relaydata ซึ่งจะเก็บข้อมูลตำแหน่งของรีเลย์ในรูปแบบของเมทริกซ์

2.4 ฟังก์ชันสำหรับสร้างเมทริกซ์พารามิเตอร์ของฟังก์ชันวัตถุประสงค์ ซึ่งในเมทริกซ์ของฟังก์ชันวัตถุประสงค์ประกอบด้วยค่าหมายเลขของรีเลย์หลักตัวที่อยู่ใกล้จุดลัดวงจร

$(T_{pri_cl_in}^i)$ รีเลย์หลักตัวที่อยู่ใกล้จุดลัดวงจร $(T_{pri_far_bus}^i)$ พารามิเตอร์ a b c และ d ฟังก์ชันที่ใช้คือฟังก์ชัน parameter_OBJ ตัวอย่างเมทริกซ์ของฟังก์ชันวัตถุประสงค์ แสดงดังตารางที่ ข.1

2.5 ฟังก์ชันสำหรับสร้างพารามิเตอร์ a เมื่อพารามิเตอร์ a คือค่าของกระแสลัดวงจรที่พิจารณา ณ ตำแหน่งติดตั้งของรีเลย์หลักใกล้กับจุดลัดวงจร โดยค่าของ a เป็นค่ากระแสลัดวงจรที่ผ่านอัตราส่วนของหม้อแปลงกระแสด้านทุติยภูมิ ฟังก์ชันที่ใช้คือฟังก์ชัน `parameter_a`

2.6 ฟังก์ชันสำหรับสร้างพารามิเตอร์ c เมื่อพารามิเตอร์ c คือค่าของกระแสลัดวงจรที่พิจารณา ณ ตำแหน่งติดตั้งของรีเลย์หลักใกล้กับจุดลัดวงจร โดยค่าของ c เป็นค่ากระแสลัดวงจรที่ผ่านอัตราส่วนของหม้อแปลงกระแสด้านทุติยภูมิ ฟังก์ชันที่ใช้คือฟังก์ชัน `parameter_c`

2.7 ฟังก์ชันข้อมูลโหลดของระบบทดสอบ ใช้สำหรับเก็บข้อมูลของค่ากระแสโหลดปกติที่ไหลผ่านสายส่งในระบบทดสอบ ซึ่งเป็นข้อมูลกระแสโหลดปกติที่คัดลอกมาจากโปรแกรม Power World Simulator มีฟังก์ชันข้อมูลโหลดของระบบทดสอบจำนวน 9 ฟังก์ชันประกอบด้วย

2.7.1 <code>loadcurrent_B3LP_data</code>	คือ ข้อมูลโหลดระบบ 3 บัส
2.7.2 <code>loadcurrent_bus6naps_data</code>	คือ ข้อมูลโหลดระบบ 6 บัส
2.7.3 <code>loadcurrent_B7flatlp_data</code>	คือ ข้อมูลโหลดระบบ 7 บัส
2.7.4 <code>loadcurrent_bus8_data</code>	คือ ข้อมูลโหลดระบบ 8 บัส
2.7.5 <code>loadcurrent_bus9_data</code>	คือ ข้อมูลโหลดระบบ 9 บัส
2.7.6 <code>loadcurrent_buslp30_data</code>	คือ ข้อมูลโหลดระบบ 30 บัส
2.7.7 <code>loadcurrent_P99sp_data</code>	คือ ข้อมูลโหลดระบบ 4700 บัส
2.7.8 <code>loadcurrent_LS1A98_data</code>	คือ ข้อมูลโหลดระบบ 8166 บัส
2.7.9 <code>loadcurrent_Wisconsin_data</code>	คือ ข้อมูลโหลดระบบ 14454 บัส

2.8 ฟังก์ชันสำหรับสร้างพารามิเตอร์ b และ d ของฟังก์ชันวัตถุประสงค์ เมื่อพารามิเตอร์ b และ d คือค่าของกระแสโหลดปกติที่ไหลผ่านสายส่งในระบบทดสอบ ซึ่งเป็นกระแสโหลดปกติที่ไหลผ่านอัตราส่วนของหม้อแปลงกระแสด้านทุติยภูมิ ฟังก์ชันที่ใช้คือฟังก์ชัน `parameter_b_d`

2.9 ฟังก์ชันสำหรับสร้างเมทริกซ์พารามิเตอร์ของเงื่อนไขบังคับ ซึ่งเมทริกซ์พารามิเตอร์ของเงื่อนไขบังคับประกอบไปด้วยคู่การทำงานของรีเลย์หลัก ($T_{primary}$) รีเลย์สำรอง (T_{backup}) พารามิเตอร์ e f g และ h ฟังก์ชันที่ใช้คือฟังก์ชัน `parameter_constraint` ตัวอย่างเมทริกซ์ของเงื่อนไขบังคับ แสดงดังตารางที่ ข.2

2.10 ฟังก์ชันสำหรับสร้างพารามิเตอร์ e และ g ของเมทริกซ์เงื่อนไขบังคับ เมื่อพารามิเตอร์ e คือค่าของกระแสลัดวงจรที่ไหลผ่านรีเลย์สำรองและพารามิเตอร์ g คือค่าของกระแส

ลัดวงจรที่ไหลผ่านรีเลย์หลัก ซึ่งค่า e และ g เป็นค่ากระแสลัดวงจรที่ไหลผ่านอัตราส่วนของหม้อแปลงกระแสทางด้านทุติยภูมิ ฟังก์ชันที่ใช้คือฟังก์ชัน `parameter_e_g`

2.11 ฟังก์ชันสำหรับสร้างพารามิเตอร์ f และ h ของเมทริกซ์เงื่อนไขบังคับ เมื่อพารามิเตอร์ f และ h คือค่าของกระแสโหลดปกติที่ไหลผ่านสายส่งในระบบทดสอบ ซึ่งเป็นกระแสโหลดปกติที่ไหลผ่านอัตราส่วนของหม้อแปลงกระแสทางด้านทุติยภูมิ ฟังก์ชันที่ใช้คือฟังก์ชัน `parameter_f_h`

2.12 ฟังก์ชันสำหรับเลือกอัตราส่วนของหม้อแปลงกระแสที่ใช้ร่วมกับรีเลย์ในระบบทดสอบ ซึ่งพิจารณาการเลือกอัตราส่วนหม้อแปลงกระแสที่เหมาะสมจากค่ากระแสโหลดปกติที่คิดเพื่อโหลดเกินไว้ 125% เพื่อใช้ในการปรับสัดส่วนของค่ากระแสลัดวงจรและค่ากระแสโหลดปกติ

2.13 ฟังก์ชันสำหรับพิจารณาเงื่อนไขที่รีเลย์ไม่สามารถทำงานได้ ในกรณีที่เวลาในการทำงานของรีเลย์สำรองมีค่าน้อยกว่าเวลาในการทำงานของรีเลย์หลักหรือในกรณีที่เวลาในการทำงานของทั้งรีเลย์หลักและรีเลย์สำรองมีค่าน้อยกว่าศูนย์ หากเกิดกรณีดังกล่าวขึ้นฟังก์ชัน `cut_parameter` จะทำหน้าที่ในการตัดเงื่อนไขส่วนนี้ออกจากเมทริกซ์เงื่อนไขบังคับ

2.14 ฟังก์ชันสำหรับรันโปรแกรมด้วยวิธีกำหนดการเชิงเส้น เป็นฟังก์ชันที่ใช้สำหรับสร้างข้อมูลตัวแปรอินพุตที่จำเป็นในการแก้ปัญหาด้วยฟังก์ชันของกำหนดการเชิงเส้น ซึ่งเป็นกล่องเครื่องมือของโปรแกรมแมทแล็บ โดยอินพุตที่กล่าวถึงจะใช้เมทริกซ์ของฟังก์ชันวัตถุประสงค์และเมทริกซ์ของเงื่อนไขบังคับในการสร้างข้อมูลที่มีรูปแบบเฉพาะของอินพุตนั้น ๆ ขึ้นอยู่กับรูปแบบของปัญหา ฟังก์ชันที่ใช้ในการรันโปรแกรมด้วยกำหนดการเชิงเส้นมี 9 ฟังก์ชันประกอบด้วย

2.14.1 <code>run_LP_3BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 3 บัส
2.14.2 <code>run_LP_6BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 6 บัส
2.14.3 <code>run_LP_7BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 7 บัส
2.14.4 <code>run_LP_8BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 8 บัส
2.14.5 <code>run_LP_9BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 9 บัส
2.14.6 <code>run_LP_30BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 30 บัส
2.14.7 <code>run_LP_4700BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 4700 บัส
2.14.8 <code>run_LP_8166BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 8166 บัส
2.14.9 <code>run_LP_14454BUS</code>	คือ ฟังก์ชันรัน LP ระบบ 14454 บัส

2.15 ฟังก์ชันสำหรับรันโปรแกรม เป็นการรวมฟังก์ชันทั้งหมดดังกล่าวข้างต้น โดยเรียงลำดับฟังก์ชันต่าง ๆ ตั้งแต่ฟังก์ชันที่ 2.2 ถึงฟังก์ชันที่ 2.14 ดังรูปที่ 3.4 ฟังก์ชันที่ใช้ในการรันโปรแกรมมี 9 ฟังก์ชัน ประกอบด้วย

2.15.1 run_3bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 3 บัส
2.15.2 run_6bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 6 บัส
2.15.3 run_7bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 7 บัส
2.15.4 run_8bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 8 บัส
2.15.5 run_9bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 9 บัส
2.15.6 run_30bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 30 บัส
2.15.7 run_4700bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 4700 บัส
2.15.8 run_8166bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 8166 บัส
2.15.9 run_14454bus	คือ ฟังก์ชันรันโปรแกรมของระบบ 14454 บัส

3. การระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง เป็นการแสดงผลลัพธ์ของคูรีเลย์หลักและรีเลย์สำรองในการป้องกันการลัดวงจรที่เกิดขึ้น ณ บัสใด ๆ ในระบบทดสอบของแต่ละระบบทดสอบ โดยมีขั้นตอนการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองแสดงดังรูปที่ 3.2

4. โปรแกรมส่วนต่อประสานกราฟิกสำหรับแสดงผล เป็นหน้าต่างที่ใช้สำหรับการแสดงผลลัพธ์ของการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรอง ผลลัพธ์ของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยหน้าต่าง GUI สำหรับแสดงผลลัพธ์แสดงดังรูปที่ 3.3

3.6.2 ลักษณะเด่นในการสร้างอัลกอริทึมการประสานสัมพันธ์รีเลย์ของงานวิจัย

จากการสร้างอัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลังขนาดใหญ่ ซึ่งประกอบด้วยฟังก์ชันต่าง ๆ ที่ใช้ในการแก้ปัญหาตามที่กล่าวมาข้างต้น โดยฟังก์ชันทุกฟังก์ชันมีความสำคัญในการสร้างอัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์และมีหลายฟังก์ชันที่แสดงให้เห็นถึงลักษณะเด่นของงานวิจัยนี้ ซึ่งต่อไปนี้จะกล่าวถึงลักษณะเด่นของงานวิจัย กล่าวคือ

1. ในการแก้ปัญหการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยใช้วิธีกำหนดการเชิงเส้น ซึ่งเป็นกล่องเครื่องมือในโปรแกรมแมทแลบที่มีชื่อฟังก์ชันว่า linprog โดยฟังก์ชันนี้จะมีอินพุตหลายตัวแปรที่ใช้ในการแก้ปัญหา ค่าอินพุตของฟังก์ชัน linprog แสดงดังสมการที่ (2-53) ในบทที่ 2 ซึ่งค่าอินพุตต่าง ๆ ของฟังก์ชันนี้จะถูกสร้างขึ้นมาได้แบบอัตโนมัติ

2. ในการสร้างเมทริกซ์ของฟังก์ชันวัตถุประสงค์และเมทริกซ์ของเงื่อนไขบังคับ ตัวอย่างเมทริกซ์ของฟังก์ชันวัตถุประสงค์และเมทริกซ์ของเงื่อนไขบังคับ แสดงดังตารางที่ ข.1 และ ตารางที่ ข.2 ตามลำดับ โดยเมทริกซ์ทั้งสองเมทริกซ์นี้จะถูกสร้างขึ้นมาได้แบบอัตโนมัติ

3. การระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ สามารถที่จะระบุคู่การทำงานของรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรขึ้นทุกบัสได้ ช่วยในการจับคู่การทำงานของรีเลย์หลักและรีเลย์สำรองทั้งหมดได้แบบอัตโนมัติ ซึ่งข้อมูลคู่การทำงานของรีเลย์ดังกล่าวมีความสำคัญในการนำไปพิจารณาสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง มีประโยชน์อย่างยิ่งในกรณีที่ระบบทดสอบมีขนาดใหญ่ซึ่งยากต่อการพิจารณาคู่การทำงานของรีเลย์ด้วยสายตาและการคำนวณด้วยมือ

4. การใช้โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) สำหรับแสดงผลลัพธ์ของการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบทดสอบเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ รวมถึงผลลัพธ์ของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบทดสอบ ช่วยเพิ่มความสะดวกและความครบถ้วนในการแสดงผลลัพธ์ทั้งหมดบนหน้าต่างการทำงานของ GUI เพียงหน้าต่างเดียว

3.7 สรุป

ในบทที่ 3 ได้นำเสนอถึงวิธีการดำเนินงานวิจัยในการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยอธิบายถึงขั้นตอนหลัก 4 หัวข้อ ในส่วนหัวข้อแรกเป็นเรื่องของการเตรียมข้อมูลของระบบทดสอบ ซึ่งกล่าวถึงขั้นตอนการเตรียมข้อมูลของระบบทดสอบให้ถูกต้องเพื่อการรันโปรแกรมหาค่ากระแสลัดวงจรโดยใช้ฟังก์ชันของโปรแกรมแมทแลปที่สร้างขึ้น ในหัวข้อที่สองเป็นการเขียนโปรแกรมและรันข้อมูล โดยสร้างฟังก์ชันต่าง ๆ จากรูปแบบของฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับที่กำหนดเพื่อใช้ในการรันโปรแกรม ซึ่งรายละเอียดของฟังก์ชันต่าง ๆ แสดงไว้ในส่วนของภาคผนวก ข หัวข้อที่สามเป็นขั้นตอนในการสร้างฟังก์ชันสำหรับการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ ในระบบทดสอบ ซึ่งใช้โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับแสดงผลลัพธ์พร้อมกับค่า TDS ของรีเลย์และค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้น ซึ่งหน้าต่างโปรแกรมส่วนต่อประสานกราฟิกสำหรับการแสดงผลลัพธ์ดังกล่าวข้างต้น แสดงไว้ในหัวข้อที่ 3.5 และเพื่อความเข้าใจในลำดับของการสร้างอัลกอริทึมในงานวิจัยนี้ ได้แสดงภาพรวมของการสร้างอัลกอริทึมสำหรับการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลังขนาดใหญ่ ดังรูปที่ 3.4

บทที่ 4

ผลการทดสอบ

4.1 บทนำ

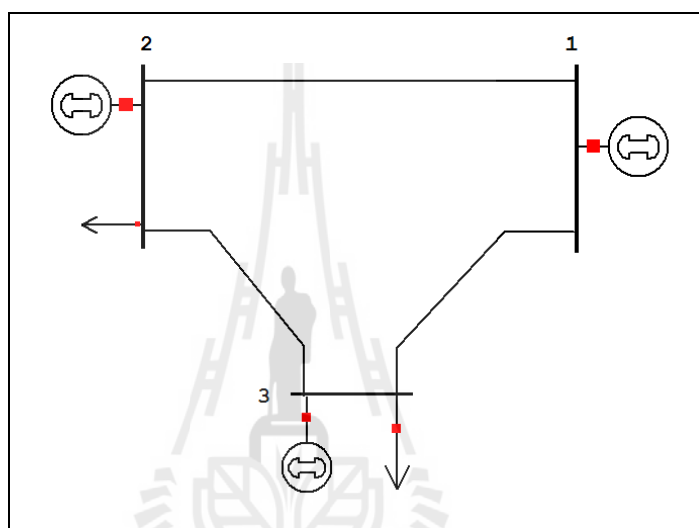
ในบทที่ 3 ได้นำเสนอขั้นตอนการดำเนินการในงานวิจัย ซึ่งกล่าวถึงขั้นตอนแรกในการแก้ปัญหาถึงขั้นตอนสุดท้ายซึ่งเป็นส่วนของการแสดงผลลัพธ์ ในบทนี้จึงนำเสนอผลลัพธ์ของการแก้ปัญหาการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางของระบบทดสอบแต่ละระบบทดสอบ เพื่อต้องการทราบค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดและค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) โดยการสร้างฟังก์ชันของโปรแกรมแมทแลปในการแก้ปัญหาดังวิธีการที่นำเสนอและเปรียบเทียบผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ที่ได้กับวิธีการวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution : DE) (Price, Storn, and Lampinen, 2005) ซึ่งทดสอบกับระบบทดสอบจำนวน 9 ระบบ ประกอบด้วยระบบทดสอบขนาด 3 บัส 6 บัส 7 บัส 8 บัส 9 บัส 30 บัส 4700 บัส 8166 บัสและ 14454 บัส นอกจากนี้ได้แสดงผลลัพธ์ของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ ในระบบทดสอบ โดยการแสดงผลลัพธ์ทั้งหมดดังกล่าวข้างต้นนี้ได้ใช้ GUI ช่วยในการแสดงผลลัพธ์ ซึ่งรายละเอียดของผลลัพธ์จากการทดสอบของระบบทดสอบแต่ละระบบ มีดังต่อไปนี้

4.2 ผลการทดสอบของระบบทดสอบ

ในบทที่ 2 ได้นำเสนอเกี่ยวกับความรู้พื้นฐานและรูปแบบของวิธีกำหนดการเชิงเส้นและบทที่ 3 ได้นำเสนอเกี่ยวกับข้อกำหนดต่าง ๆ ของปัญหาที่ใช้ในการแก้ปัญหาการประสานสัมพันธ์รีเลย์ ซึ่งจากผลการศึกษาในส่วนนี้และงานวิจัยอื่นๆ ที่เกี่ยวข้อง ทำให้สามารถนำความรู้ที่ได้ประกอบกับรูปแบบและหลักการของวิธีกำหนดการเชิงเส้น นำไปสู่การแก้ปัญหาเพื่อหาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์และค่า TDS ของรีเลย์ในระบบทดสอบ ซึ่งผลของการทดสอบในแต่ละระบบทดสอบ มีดังนี้

4.2.1 ระบบทดสอบ 3 บัส

การแก้ปัญหาการประสานสัมพันธรีเลย์โดยใช้วิธีกำหนดการเชิงเส้น ใช้หลักการตามที่ได้นำเสนอไว้ในหัวข้อ 3.3 โดยพิจารณาฟังก์ชันวัตถุประสงค์ดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบขนาด 3 บัสแสดงดังรูปที่ 4.1 โดยทดสอบในระดับแรงดันไฟฟ้าขนาด 345 kV ที่กำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 3 เครื่องและรายละเอียดข้อมูลของระบบทดสอบ 3 บัสแสดงไว้ในภาคผนวก ก

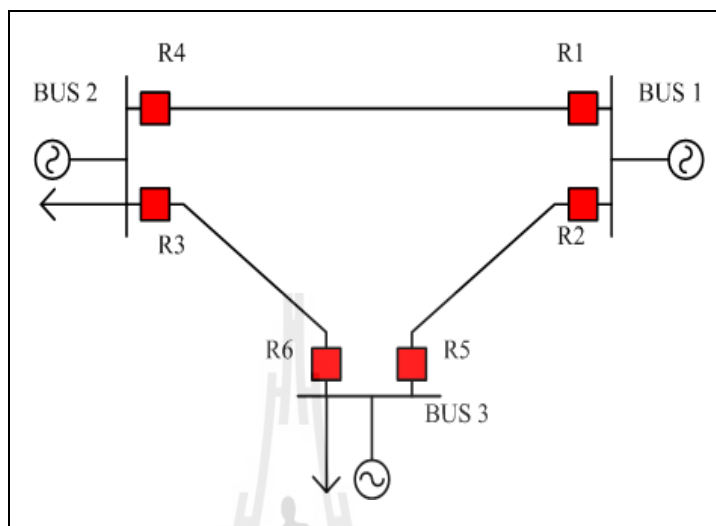


รูปที่ 4.1 ระบบทดสอบ 3 บัส

เพื่อกำหนดรีเลย์สำหรับติดตั้งในระบบทดสอบ ได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) สำหรับกำหนดตำแหน่งของรีเลย์รูปที่ 4.2 แสดงตำแหน่งของรีเลย์ที่อยู่ในระบบทดสอบ 3 บัส เพื่อใช้ในการประสานสัมพันธรีเลย์จากรูปมีรีเลย์ทั้งหมดจำนวน 6 ตัว ตั้งแต่รีเลย์หมายเลข 1 (R1) ถึงรีเลย์หมายเลข 6 (R6)

เมื่อกำหนดตำแหน่งของรีเลย์ในระบบทดสอบแล้ว สามารถระบุได้ว่ามีรีเลย์ตัวใดบ้างที่ทำหน้าที่เป็นรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจร ณ บัสใด ๆ ในระบบทดสอบ ซึ่งเมื่อระบบทดสอบเกิดการลัดวงจรขึ้นที่บัสใดบัสหนึ่งแล้ว รีเลย์ต้องพิจารณาได้ว่าทิศทางของกระแสลัดวงจรที่ไหลผ่านสายส่งในระบบทดสอบนั้น มีผลต่อการลัดวงจรในเหตุการณ์นั้น ๆ หรือไม่ ซึ่งกระบวนการทั้งหมดดังกล่าวนี้ ได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) ในการพิจารณาสำหรับการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรอง ซึ่งผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 3 บัส แสดง

ดังตารางที่ 4.1 จากตารางแสดงถึงบัสที่เกิดการลัดวงจรและมีการป้องกันปฐมภูมิซึ่งก็คือการทำงานของรีเลย์หลัก ส่วนการป้องกันทุติยภูมิคือการทำงานของรีเลย์สำรอง

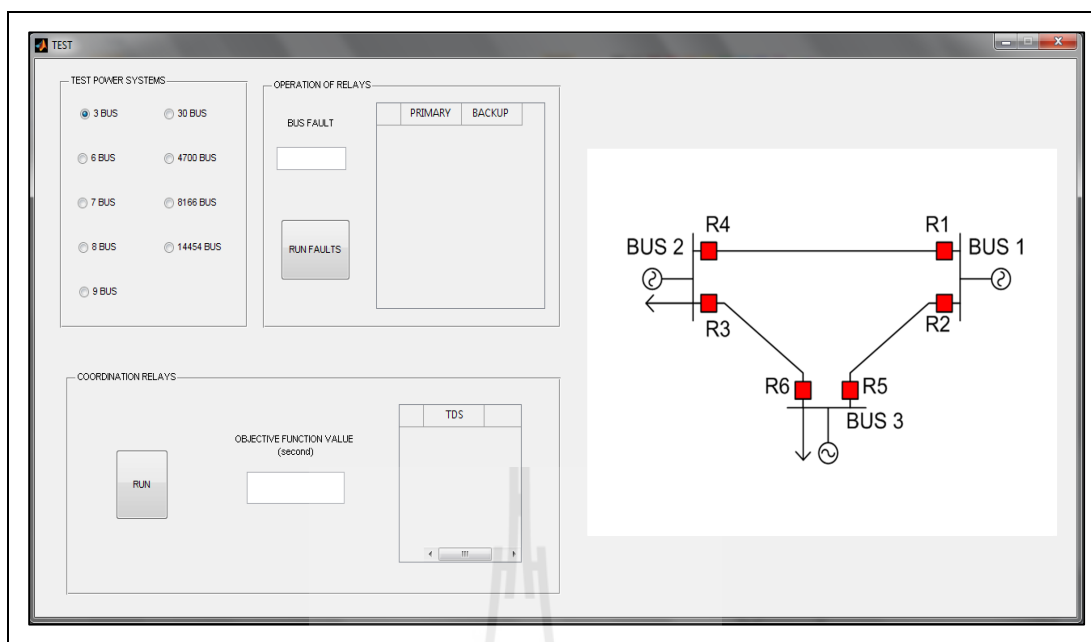


รูปที่ 4.2 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 3 บัส

ตารางที่ 4.1 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 3 บัส

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
1	1	4
	2	5
2	3	6
	4	1
3	5	2
	6	3

จากรูปที่ 4.3 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับแสดงผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 3 บัสเมื่อเกิดการลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



รูปที่ 4.3 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 3 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบ 3 บัส โดยใช้วิธีกำหนดการเชิงเส้นและเปรียบเทียบผลลัพธ์กับวิธีการวิวัฒนาการโดยใช้ผลต่าง แสดงผลลัพธ์ได้ดังตารางที่ 4.2 และกราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่างของระบบทดสอบ 3 บัส ดังรูปที่ 4.4

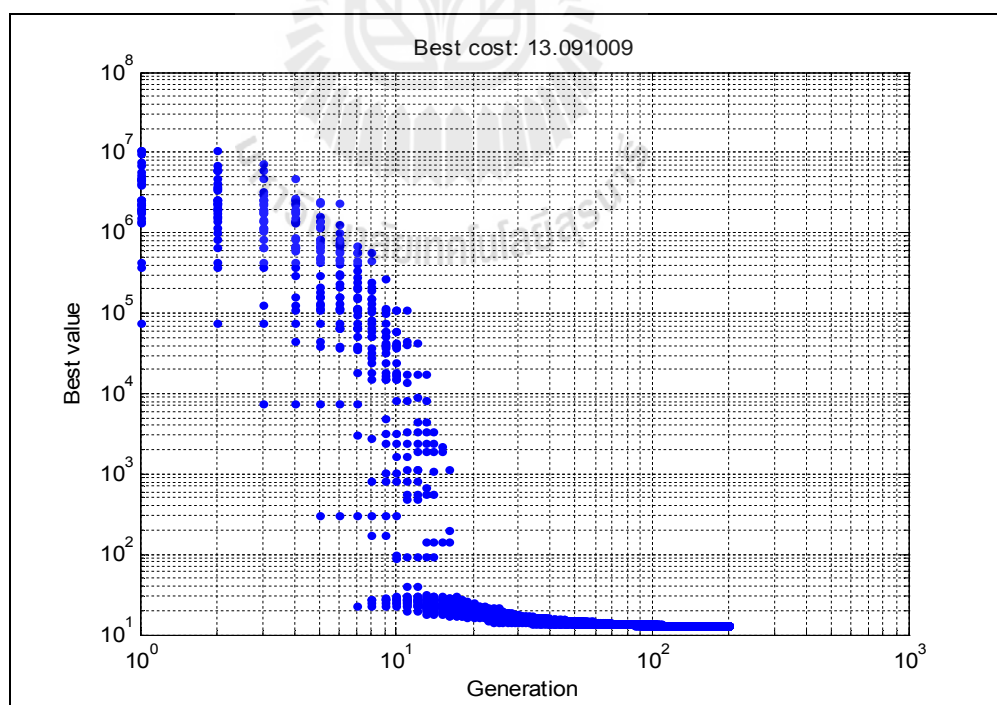
ตารางที่ 4.2 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 3 บัส

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการ โดยใช้ผลต่าง (DE)
	TDS	TDS
1	0.07	0.07
2	0.05	0.05
3	0.08	0.08
4	0.08	0.08

ตารางที่ 4.2 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 3 บัส (ต่อ)

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
5	0.05	0.05
6	0.07	0.07
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	13.09	13.09
จำนวนการวนรอบที่ใช้(รอบ)	9	200

ผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 3 บัสแสดงผลดังตารางข้างต้น พบว่าค่า TDS และค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการ โดยใช้ผลต่างให้ผลลัพธ์เท่ากับ 13.09 วินาที ซึ่งมีค่าเท่ากันแสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถนำมาใช้แก้ปัญหาในการประสานสัมพันธ์รีเลย์ในระบบทดสอบได้

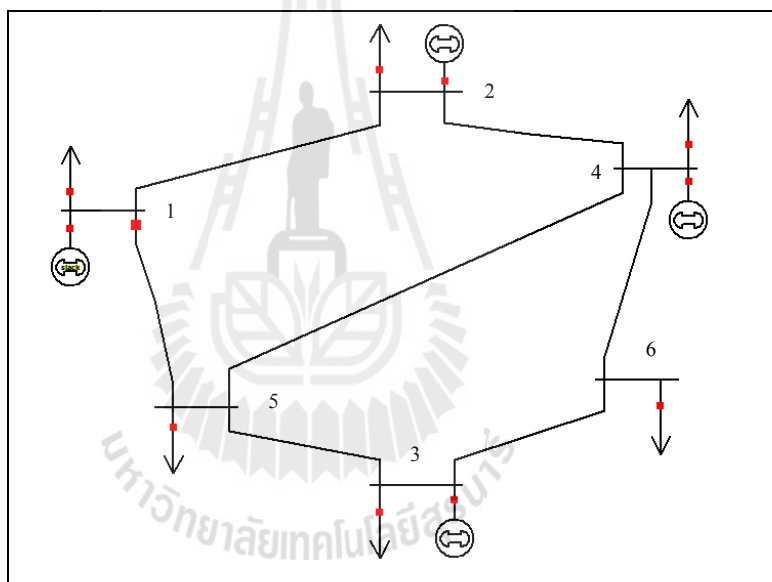


รูปที่ 4.4 กราฟแสดงการลู่เข้าของค่าตอบด้วยวิธีวิวัฒนาการ โดยใช้ผลต่างของระบบทดสอบ 3 บัส

จากกราฟแสดงให้เห็นถึงการลู่เข้าของค่าตอบของค่าฟังก์ชันวัตถุประสงค์ ซึ่งทำการวนรอบจำนวน 200 รอบ ได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์เท่ากับ 13.09 วินาที

4.2.2 ระบบทดสอบ 6 บัส

พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 6 บัส แสดงดังรูปที่ 4.5 ใช้กับระดับแรงดันไฟฟ้า 138 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 4 เครื่องและรายละเอียดข้อมูลของระบบทดสอบ 6 บัส ได้แสดงไว้ในภาคผนวก ก

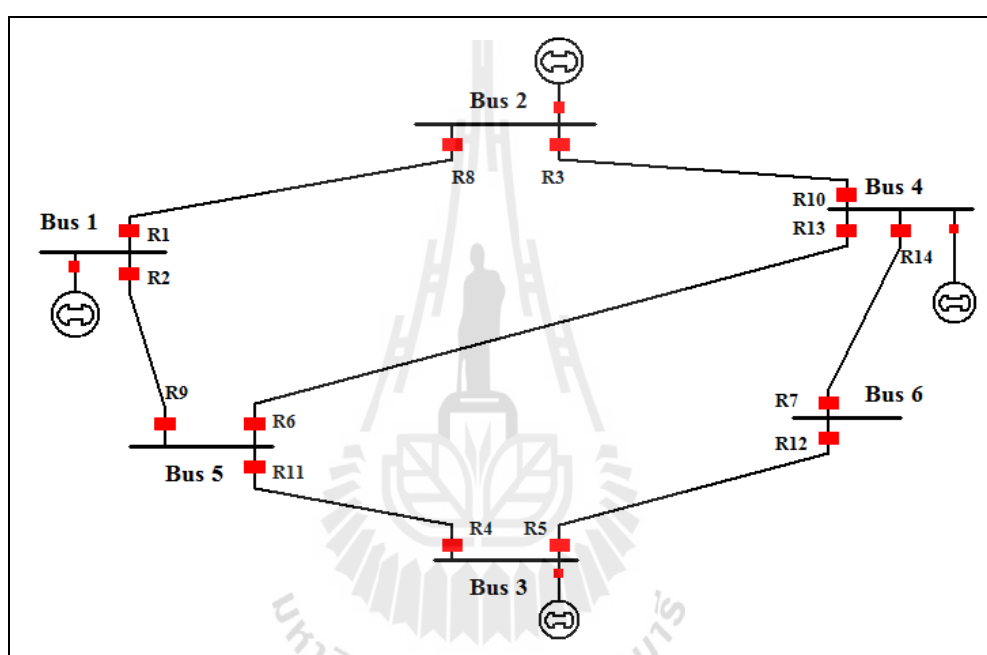


รูปที่ 4.5 ระบบทดสอบ 6 บัส

เพื่อกำหนดรีเลย์สำหรับติดตั้งในระบบทดสอบ ได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) สำหรับกำหนดตำแหน่งของรีเลย์ รูปที่ 4.6 แสดงตำแหน่งของรีเลย์ที่อยู่ในระบบทดสอบ 6 บัส เพื่อใช้ในการประสานสัมพันธ์รีเลย์ จากรูปมีรีเลย์ทั้งหมดจำนวน 14 ตัว ตั้งแต่รีเลย์หมายเลข 1 (R1) ถึงรีเลย์หมายเลข 14 (R14)

เมื่อกำหนดตำแหน่งของรีเลย์ในระบบทดสอบไว้แล้ว สามารถระบุได้ว่ามีรีเลย์ตัวใดบ้างที่ทำหน้าที่เป็นรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจร ณ บัสใด ๆ ในระบบทดสอบ ซึ่งเมื่อระบบทดสอบเกิดการลัดวงจรขึ้นที่บัสใดบัสหนึ่งแล้ว รีเลย์ต้องพิจารณาได้ว่าทิศทางของ

กระแสลัดวงจรที่ไหลผ่านสายส่งในระบบทดสอบนั้น มีผลต่อการลัดวงจรในเหตุการณ์นั้น ๆ หรือไม่ ซึ่งกระบวนการทั้งหมดดังกล่าวนี้จึงได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) ในการพิจารณาสำหรับการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรอง ซึ่งผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 6 บัส แสดงดังตารางที่ 4.3 จากตารางแสดงถึงบัสที่เกิดการลัดวงจรและมีการป้องกันปฐมภูมิซึ่งก็คือการทำงานของรีเลย์หลัก ส่วนการป้องกันทุติยภูมิคือการทำงานของรีเลย์สำรอง



รูปที่ 4.6 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 6 บัส

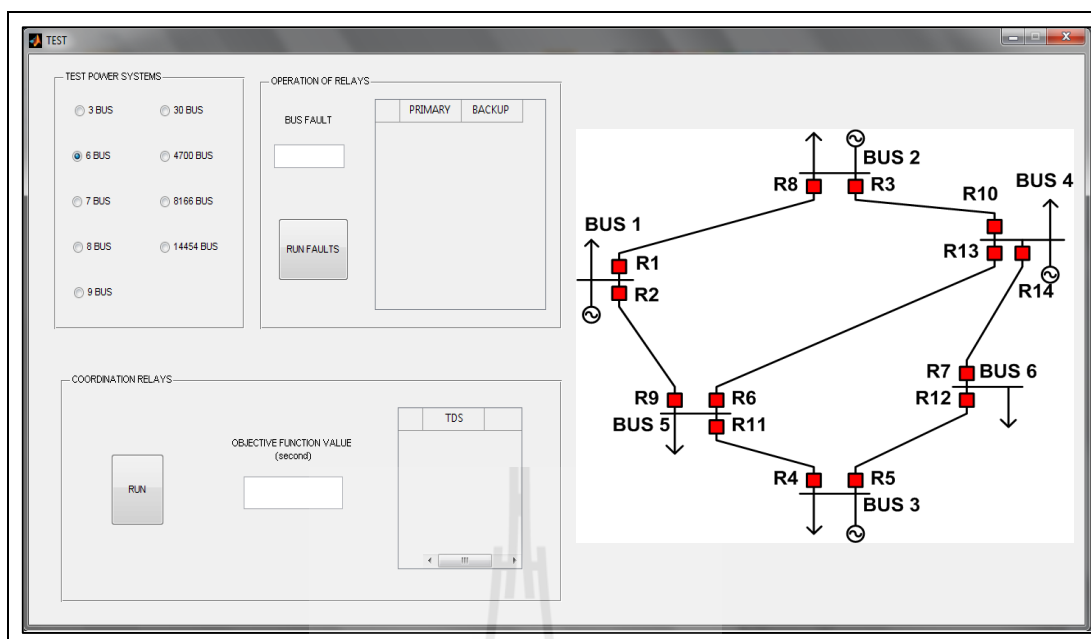
ตารางที่ 4.3 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 6 บัส

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
1	1	8
	2	9
2	3	10
	8	1

ตารางที่ 4.3 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 6 บัส (ต่อ)

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
3	4	11
	5	12
4	10	3
	13	6
	14	7
5	6	13
	9	2
	11	4
6	7	14
	12	5

จากรูปที่ 4.7 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับแสดงผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 6 บัสเมื่อเกิดการลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลของการประสานสัมพันธ์รีเลย์ป้องกันกระแสนเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



รูปที่ 4.7 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 6 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบ 6 บัส โดยใช้วิธีกำหนดการเชิงเส้นและเปรียบเทียบผลลัพธ์กับวิธีการวิวัฒนาการโดยใช้ผลต่าง แสดงผลลัพธ์ได้ดังตารางที่ 4.4 และกราฟแสดงการดูเข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่างของระบบทดสอบ 6 บัส ดังรูปที่ 4.8

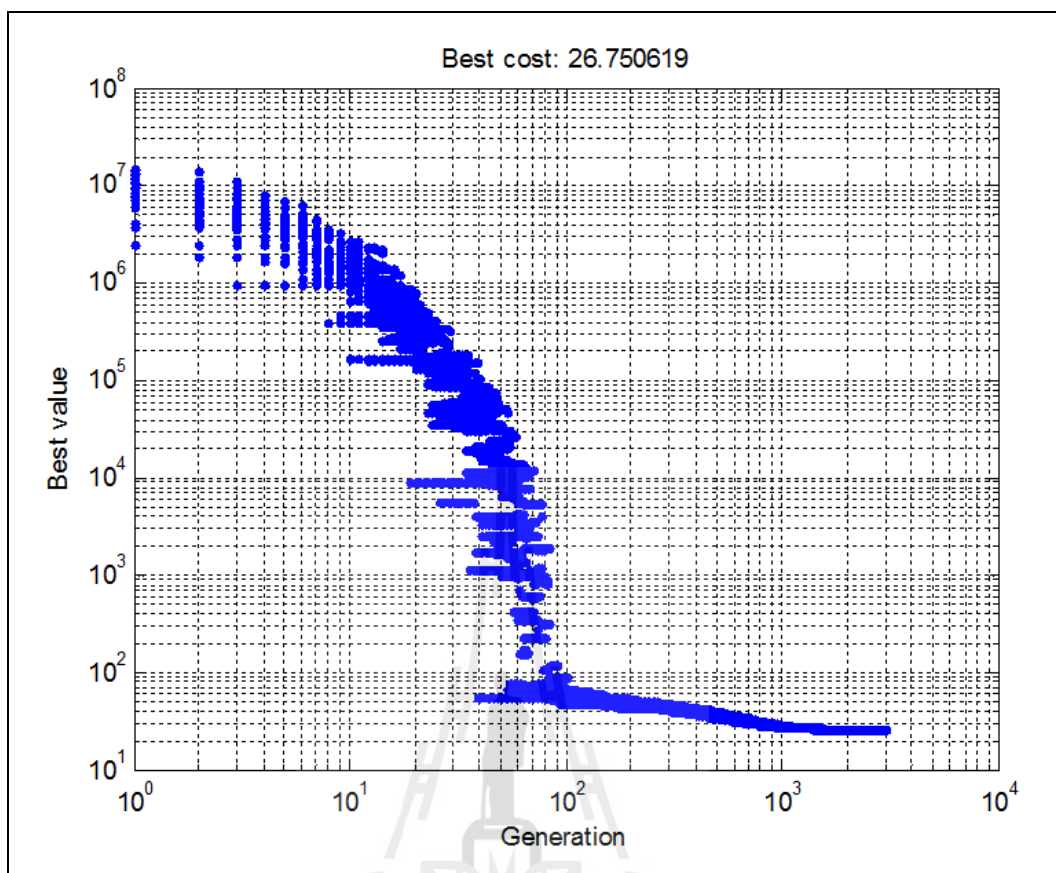
ตารางที่ 4.4 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 6 บัส

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
1	0.05	0.05
2	0.19	0.19
3	0.10	0.10
4	0.08	0.08

ตารางที่ 4.4 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 6 บัส (ต่อ)

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
5	0.75	0.75
6	0.05	0.05
7	0.05	0.05
8	0.10	0.10
9	0.06	0.06
10	0.14	0.14
11	0.37	0.37
12	0.14	0.14
13	0.08	0.08
14	0.05	0.05
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	26.65	26.75
จำนวนการวนรอบที่ใช้ (รอบ)	13	3000

ผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 6 บัสแสดงผลดังตารางข้างต้น พบว่าค่า TDS และค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการโดยใช้ผลต่างให้ผลลัพธ์เท่ากับ 26.65 วินาทีและ 26.75 วินาที ตามลำดับ ซึ่งมีค่าใกล้เคียงกันแสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถนำมาใช้แก้ปัญหาในการประสานสัมพันธรีเลย์ในระบบทดสอบได้

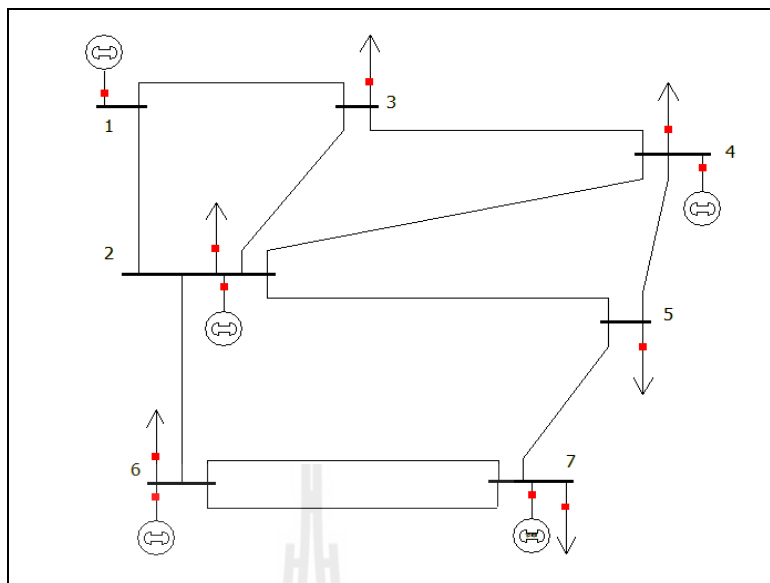


รูปที่ 4.8 กราฟแสดงการลู่เข้าของค่าตอบด้วยวิธีวิวัฒนาการ โดยใช้ผลต่างของระบบทดสอบ 6 บัส

จากกราฟแสดงให้เห็นถึงการลู่เข้าของค่าตอบของค่าฟังก์ชันวัตถุประสงค์ ซึ่งทำการวนรอบจำนวน 3000 รอบ ได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์เท่ากับ 26.75 วินาที

4.2.3 ระบบทดสอบ 7 บัส

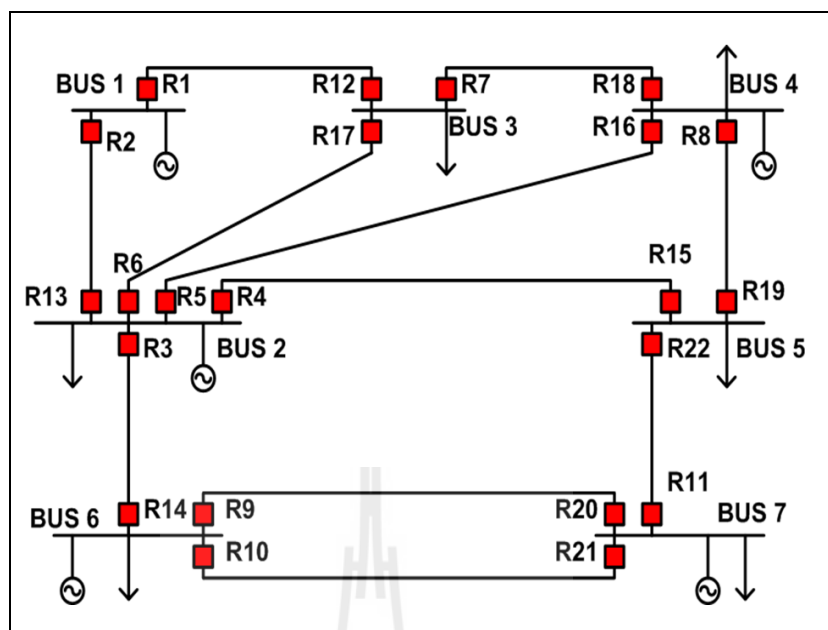
พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 7 บัส แสดงดังรูปที่ 4.9 ใช้กับระดับแรงดันไฟฟ้า 138 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 5 เครื่องและรายละเอียดข้อมูลของระบบทดสอบ 7 บัส ได้แสดงไว้ในภาคผนวก ก



รูปที่ 4.9 ระบบทดสอบ 7 บัส

เพื่อกำหนดรีเลย์สำหรับติดตั้งในระบบทดสอบ ได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) สำหรับกำหนดตำแหน่งของรีเลย์ รูปที่ 4.10 แสดงตำแหน่งของรีเลย์ที่อยู่ในระบบทดสอบ 7 บัส เพื่อใช้ในการประสานสัมพันธ์รีเลย์ จากรูปมีรีเลย์ทั้งหมดจำนวน 22 ตัว ตั้งแตรีเลย์หมายเลข 1 (R1) ถึงรีเลย์หมายเลข 22 (R22)

เมื่อกำหนดตำแหน่งของรีเลย์ในระบบทดสอบไว้แล้ว สามารถระบุได้ว่ามีรีเลย์ตัวใดบ้างที่ทำหน้าที่เป็นรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจร ณ บัสใด ๆ ในระบบทดสอบ ซึ่งเมื่อระบบทดสอบเกิดการลัดวงจรขึ้นที่บัสใดบัสหนึ่งแล้ว รีเลย์ต้องพิจารณาได้ว่าทิศทางของกระแสลัดวงจรที่ไหลผ่านสายส่งในระบบทดสอบนั้น มีผลต่อการลัดวงจรในเหตุการณ์นั้น ๆ หรือไม่ ซึ่งกระบวนการทั้งหมดดังกล่าวนี้จึงได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) ในการพิจารณาสำหรับการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรอง ซึ่งผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 7 บัส แสดงดังตารางที่ 4.5 จากตารางแสดงถึงบัสที่เกิดการลัดวงจรและมีการป้องกันปฐมภูมิซึ่งก็คือการทำงานของรีเลย์หลัก ส่วนการป้องกันทุติยภูมิคือการทำงานของรีเลย์สำรอง



รูปที่ 4.10 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 7 บัส

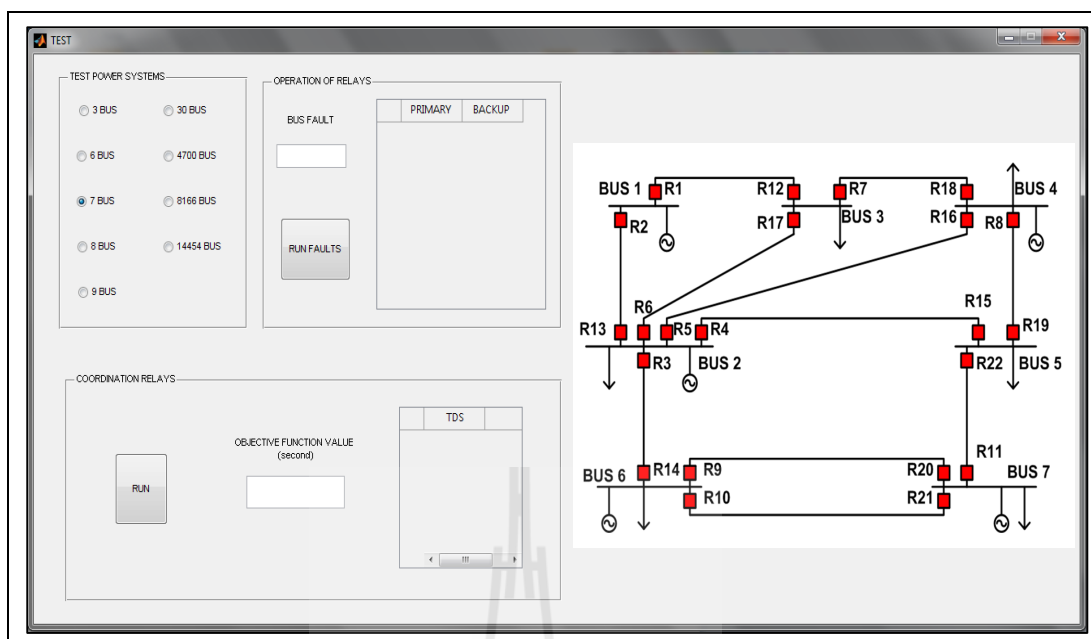
ตารางที่ 4.5 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 7 บัส

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
1	1	12
	2	13
2	3	14
	4	15
	5	16
	6	17
	13	2
	7	18
3	12	1
	17	6

ตารางที่ 4.5 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 7 บัส (ต่อ)

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
4	8	19
	16	5
	18	7
5	15	4
	19	8
	22	11
6	14	3
	9	20
	10	21
7	11	22
	20	9
	21	10

จากรูปที่ 4.11 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับแสดงผลการทำงานของ การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 7 บัสเมื่อเกิดลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธี กำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



รูปที่ 4.11 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 7 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบ 7 บัส โดยใช้วิธีกำหนดการเชิงเส้นและเปรียบเทียบผลลัพธ์กับวิธีการวิวัฒนาการโดยใช้ผลต่าง แสดงผลลัพธ์ได้ดังตารางที่ 4.6 และกราฟแสดงการถ่วงเข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่างของระบบทดสอบ 7 บัส ดังรูปที่ 4.12

ตารางที่ 4.6 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 7 บัส

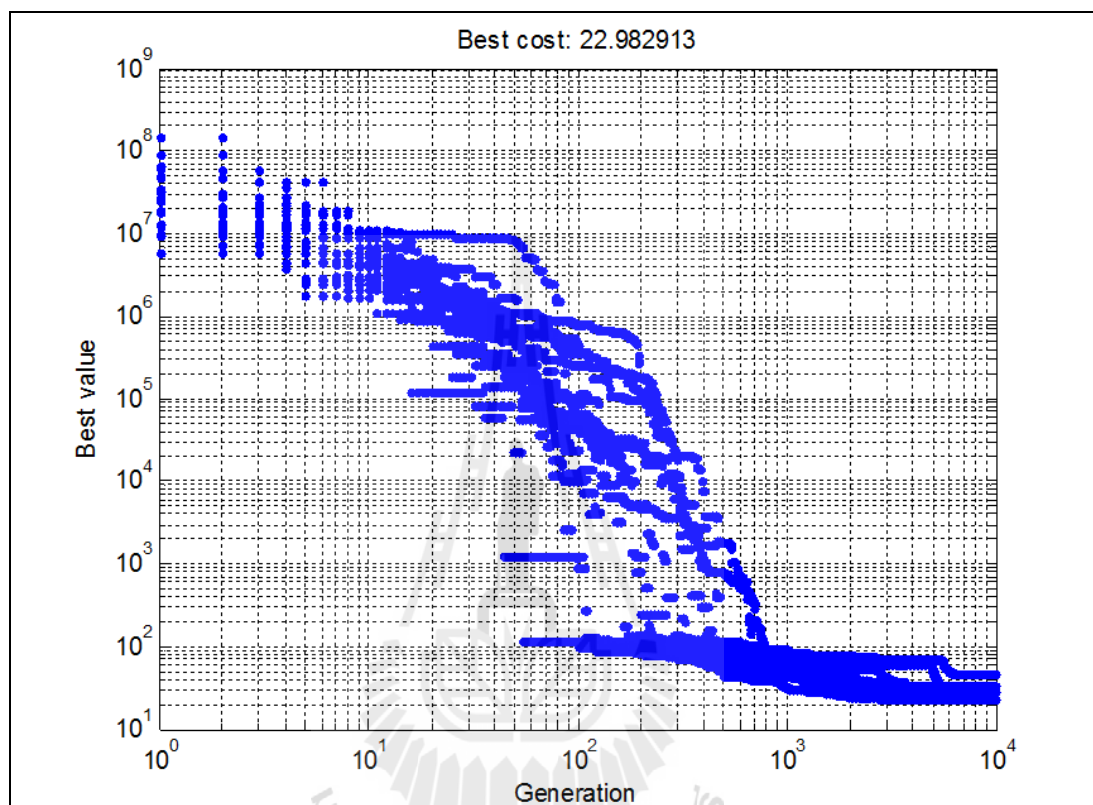
รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
1	0.10	0.10
2	0.19	0.19
3	0.26	0.26
4	0.06	0.06

ตารางที่ 4.6 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 7 บัส (ต่อ)

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
5	0.11	0.11
6	0.10	0.10
7	0.18	0.18
8	0.11	0.11
9	0.05	0.05
10	0.05	0.05
11	0.11	0.11
12	0.06	0.06
13	0.10	0.10
14	0.22	0.22
15	0.05	0.05
16	0.06	0.06
17	0.05	0.05
18	0.28	0.28
19	0.17	0.17
20	0.05	0.05
21	0.05	0.05
22	0.06	0.06
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	22.73	22.98
จำนวนการวนรอบที่ใช้ (รอบ)	12	10000

ผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 7 บัสแสดงผลดังตารางข้างต้น พบว่าค่า TDS และค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการ โดยใช้ผลต่างให้ผลลัพธ์เท่ากับ

22.73 วินาทีและ 22.98 วินาที ตามลำดับ ซึ่งผลลัพธ์มีค่าใกล้เคียงกันแสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถนำมาใช้แก้ปัญหาในการประสานสัมพันธ์ในระบบทดสอบได้

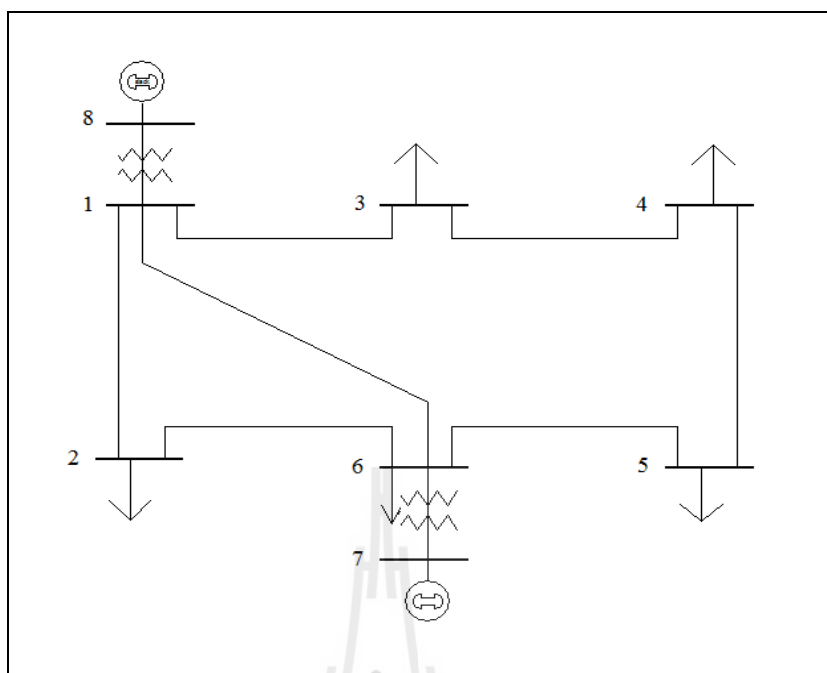


รูปที่ 4.12 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการ โดยใช้ผลต่างของระบบทดสอบ 7 บัส

จากกราฟแสดงให้เห็นถึงการลู่เข้าของคำตอบของค่าฟังก์ชันวัตถุประสงค์ ซึ่งทำการวนรอบจำนวน 10000 รอบ ได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์เท่ากับ 22.98 วินาที

4.2.4 ระบบทดสอบ 8 บัส

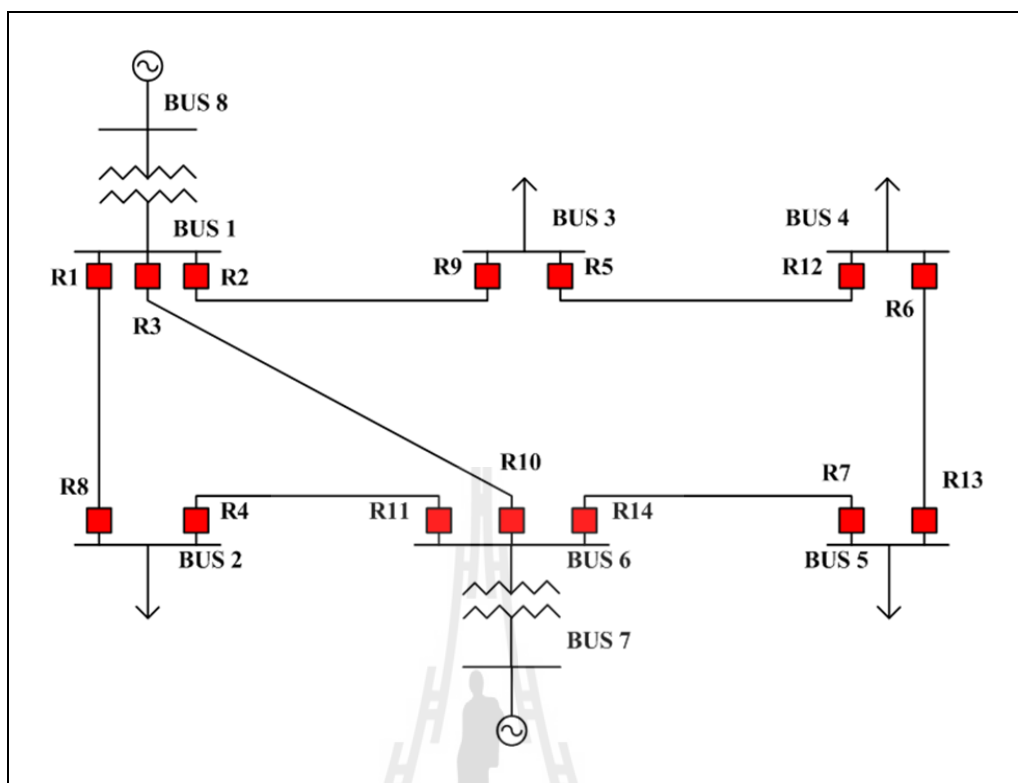
พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 8 บัส แสดงดังรูปที่ 4.13 ใช้กับระดับแรงดันไฟฟ้า 150 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 2 เครื่องและรายละเอียดข้อมูลของระบบทดสอบ 8 บัส ได้แสดงไว้ในภาคผนวก ก



รูปที่ 4.13 ระบบทดสอบ 8 บัส

เพื่อกำหนดรีเลย์สำหรับติดตั้งในระบบทดสอบ ได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) สำหรับกำหนดตำแหน่งของรีเลย์ รูปที่ 4.14 แสดงตำแหน่งของรีเลย์ที่อยู่ในระบบทดสอบ 8 บัส เพื่อใช้ในการประสานสัมพันธ์รีเลย์ จากรูปมีรีเลย์ทั้งหมดจำนวน 14 ตัว ตั้งแต่รีเลย์หมายเลข 1 (R1) ถึงรีเลย์หมายเลข 14 (R14)

เมื่อกำหนดตำแหน่งของรีเลย์ในระบบทดสอบไว้แล้ว สามารถระบุได้ว่ามีรีเลย์ตัวใดบ้างที่ทำหน้าที่เป็นรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจร ณ บัสใด ๆ ในระบบทดสอบ ซึ่งเมื่อระบบทดสอบเกิดการลัดวงจรขึ้นที่บัสใดบัสหนึ่งแล้ว รีเลย์ต้องพิจารณาได้ว่าทิศทางของกระแสลัดวงจรที่ไหลผ่านสายส่งในระบบทดสอบนั้น มีผลต่อการลัดวงจรในเหตุการณ์นั้น ๆ หรือไม่ ซึ่งกระบวนการทั้งหมดดังกล่าวนี้จึงได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) ในการพิจารณาสำหรับการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรอง ซึ่งผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 8 บัส แสดงดังตารางที่ 4.7 จากตารางแสดงถึงบัสที่เกิดการลัดวงจรและมีการป้องกันปฐมภูมิซึ่งก็คือการทำงานของรีเลย์หลัก ส่วนการป้องกันทุติยภูมิคือการทำงานของรีเลย์สำรอง



รูปที่ 4.14 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 8 บัส

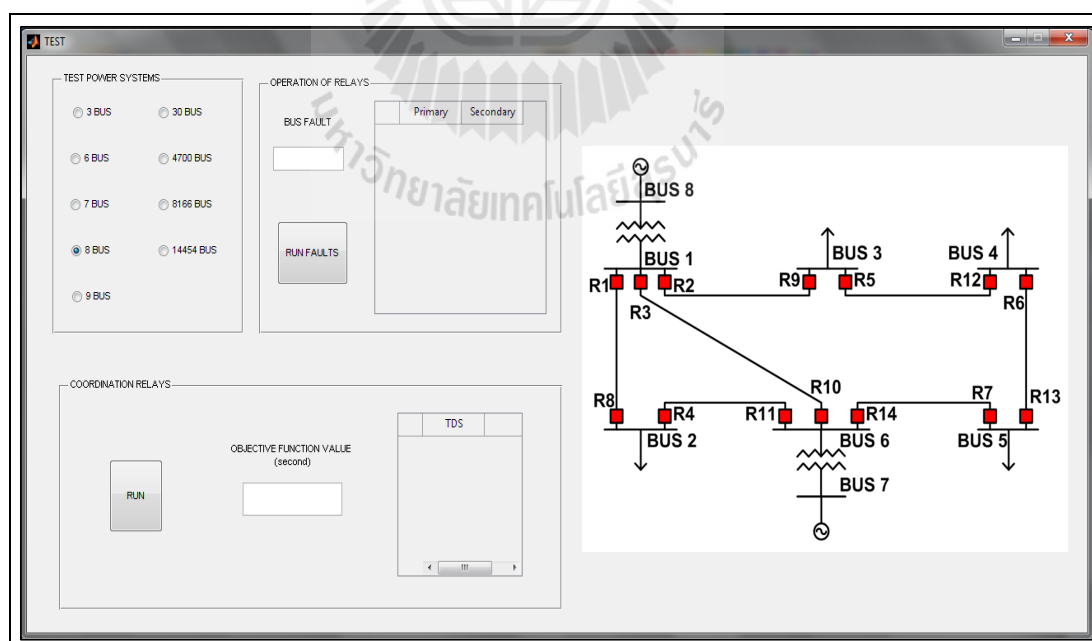
ตารางที่ 4.7 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 8 บัส

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
1	1	8
	2	9
	3	10
2	4	11
	8	1
3	9	2
	5	12
4	6	13

ตารางที่ 4.7 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 8 บัส (ต่อ)

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
4	12	5
5	7	14
	13	6
6	10	3
	11	4
	14	7

จากรูปที่ 4.15 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับแสดงผลการทำงานของ การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 8 บัสเมื่อเกิดลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธี กำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



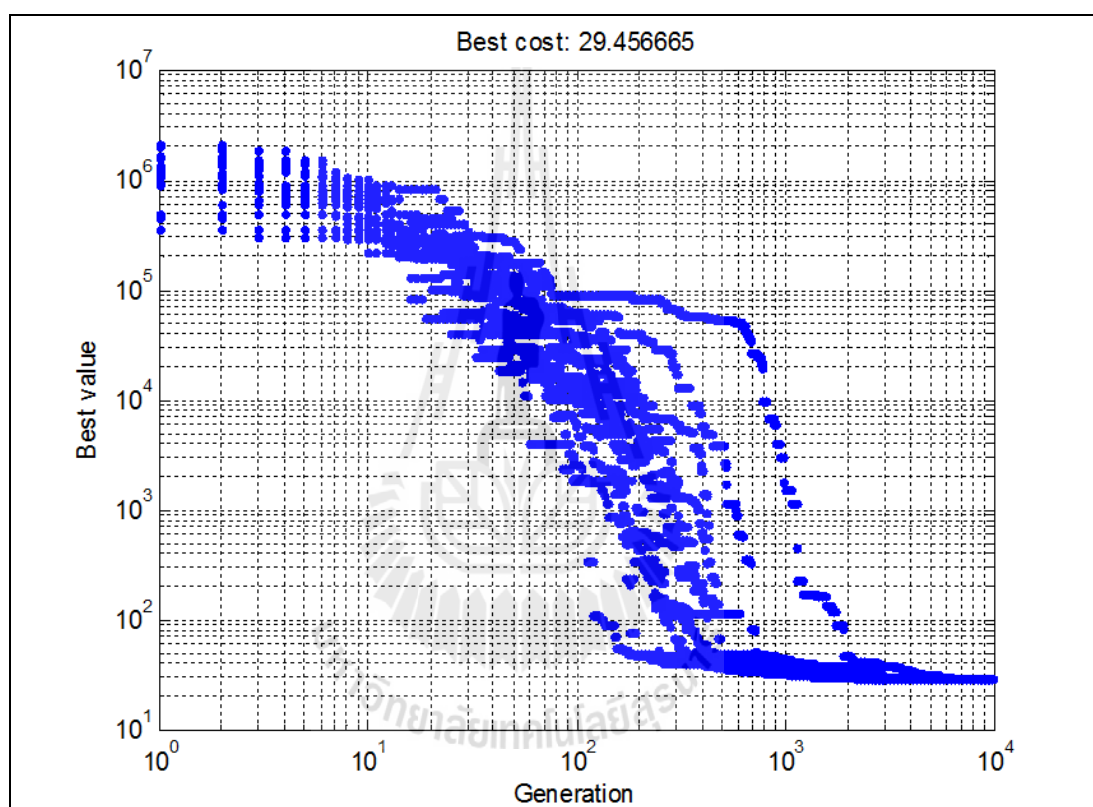
รูปที่ 4.15 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 8 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบ 8 บัส โดยใช้วิธีกำหนดการเชิงเส้นและเปรียบเทียบผลลัพธ์กับวิธีการวิวัฒนาการโดยใช้ผลต่าง แสดงผลลัพธ์ได้ดังตารางที่ 4.8 และกราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่างของระบบทดสอบ 8 บัส ดังรูปที่ 4.16

ตารางที่ 4.8 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 8 บัส

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
1	0.61	0.61
2	0.40	0.40
3	0.33	0.33
4	0.90	0.90
5	0.52	0.52
6	0.39	0.39
7	0.21	0.21
8	0.05	0.05
9	0.05	0.05
10	0.47	0.47
11	0.26	0.26
12	0.22	0.22
13	0.37	0.37
14	0.42	0.42
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	29.41	29.46
จำนวนการวนรอบที่ใช้ (รอบ)	9	10000

ผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 8 บัสดังแสดงผลลัพธ์ดังตารางข้างต้น พบว่าค่า TDS และค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการ โดยใช้ผลต่างให้ผลลัพธ์เท่ากับ 29.41 วินาทีและ 29.46 วินาที ตามลำดับ ซึ่งผลลัพธ์มีค่าใกล้เคียงกันแสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถนำมาใช้แก้ปัญหาในการประสานสัมพันธ์รีเลย์ในระบบทดสอบได้



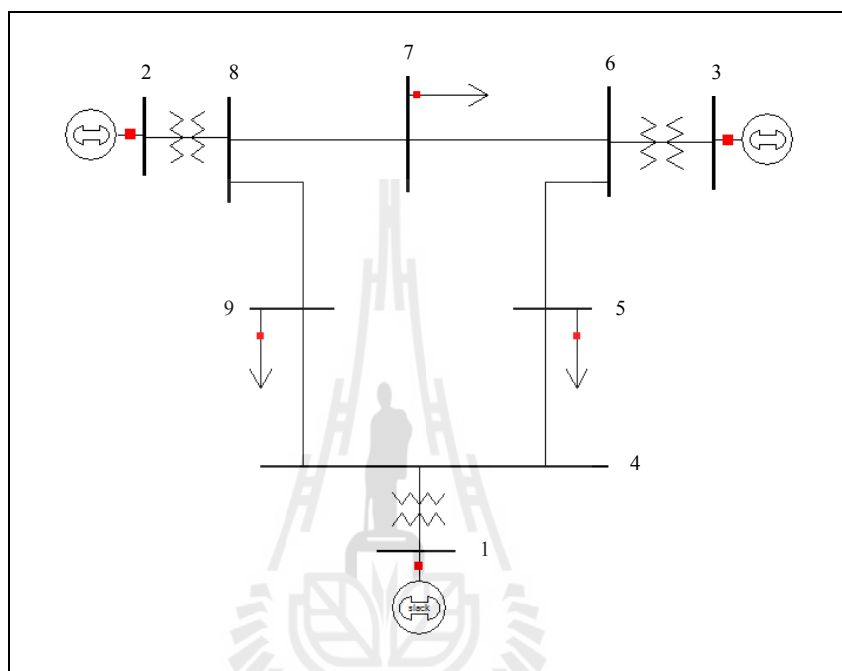
รูปที่ 4.16 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการ โดยใช้ผลต่างของระบบทดสอบ 8 บัสด

จากกราฟแสดงให้เห็นถึงการลู่เข้าของคำตอบของค่าฟังก์ชันวัตถุประสงค์ ซึ่งทำการวนรอบจำนวน 10000 รอบ ได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์เท่ากับ 29.46 วินาที

4.2.5 ระบบทดสอบ 9 บัสด

พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการ

ดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 9 บัส แสดงดังรูปที่ 4.17 (Minglan, Anurag, and Noel, 2007) (Yu, and Singh, 2006) ใช้กับระดับแรงดันไฟฟ้า 230 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 3 เครื่องและรายละเอียดข้อมูลของระบบทดสอบ 9 บัส ได้แสดงไว้ในภาคผนวก ก

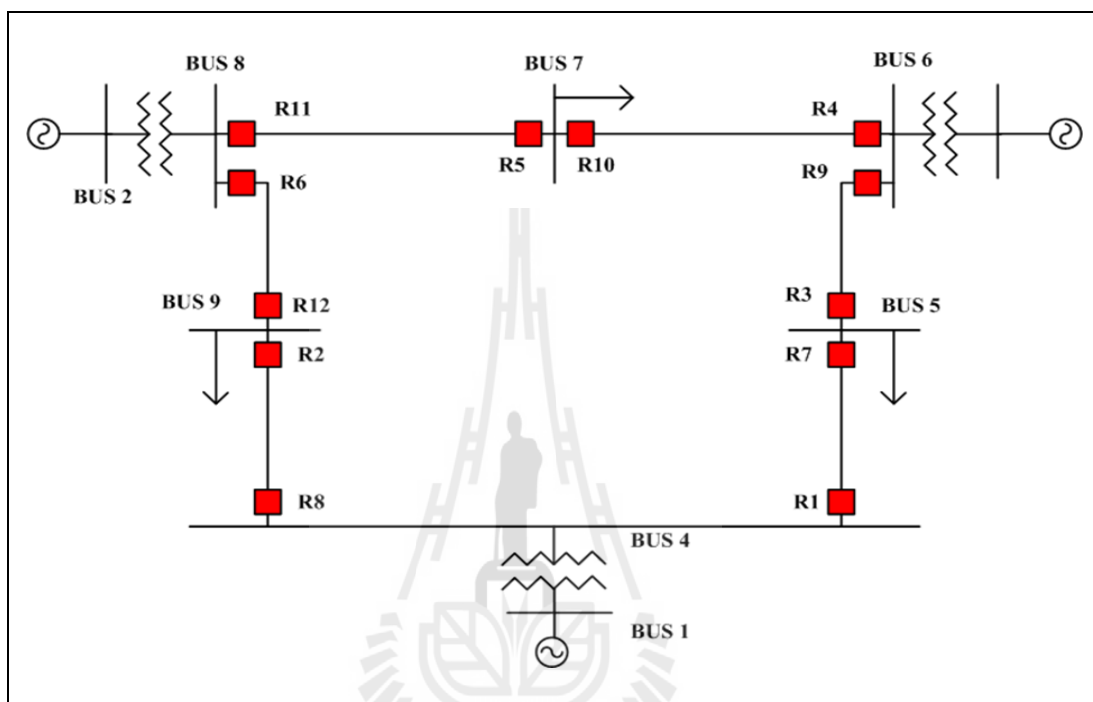


รูปที่ 4.17 ระบบทดสอบ 9 บัส

เพื่อกำหนดรีเลย์สำหรับติดตั้งในระบบทดสอบ ได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) สำหรับกำหนดตำแหน่งของรีเลย์ รูปที่ 4.18 แสดงตำแหน่งของรีเลย์ที่อยู่ในระบบทดสอบ 9 บัส เพื่อใช้ในการประสานสัมพันธ์รีเลย์ จากรูปมีรีเลย์ทั้งหมดจำนวน 12 ตัว ตั้งแต่รีเลย์หมายเลข 1 (R1) ถึงรีเลย์หมายเลข 12 (R12)

เมื่อกำหนดตำแหน่งของรีเลย์ในระบบทดสอบได้แล้ว สามารถระบุได้ว่ามีรีเลย์ตัวใดบ้างที่ทำหน้าที่เป็นรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจร ณ บัสใด ๆ ในระบบทดสอบ ซึ่งเมื่อระบบทดสอบเกิดการลัดวงจรขึ้นที่บัสใดบัสหนึ่งแล้ว รีเลย์ต้องพิจารณาได้ว่าทิศทางของกระแสลัดวงจรที่ไหลผ่านสายส่งในระบบทดสอบนั้น มีผลต่อการลัดวงจรในเหตุการณ์นั้น ๆ หรือไม่ ซึ่งกระบวนการทั้งหมดดังกล่าวนี้จึงได้เขียนฟังก์ชันของโปรแกรมแมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) ในการพิจารณาสำหรับการระบุตำแหน่งของรีเลย์หลักและ

รีเลย์สำรอง ซึ่งผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 9 บัส แสดงดังตารางที่ 4.9 จากตารางแสดงถึงบัสที่เกิดการลัดวงจรและมีการป้องกันปฐมภูมิซึ่งก็คือการทำงานของรีเลย์หลัก ส่วนการป้องกันทุติยภูมิคือการทำงานของรีเลย์สำรอง



รูปที่ 4.18 แสดงตำแหน่งของรีเลย์ในระบบทดสอบ 9 บัส

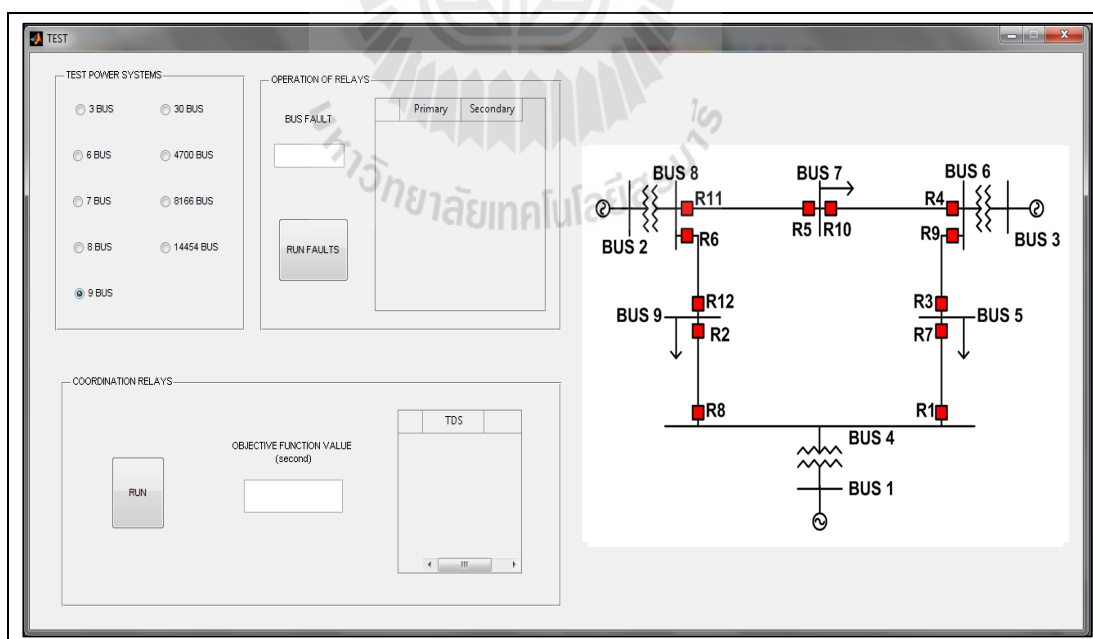
ตารางที่ 4.9 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 9 บัส

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
4	1	7
	8	2
5	3	9
	7	1
6	4	10
	9	3

ตารางที่ 4.9 การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 9 บัส (ต่อ)

บัสที่เกิดการลัดวงจร	การป้องกันปฐมภูมิ	การป้องกันทุติยภูมิ
7	5	11
	10	4
8	6	12
	11	5
9	2	8
	12	6

จากรูปที่ 4.19 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับแสดงผลการทำงานของ การระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 9 บัสเมื่อเกิดลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธี กำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



รูปที่ 4.19 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 9 บัส

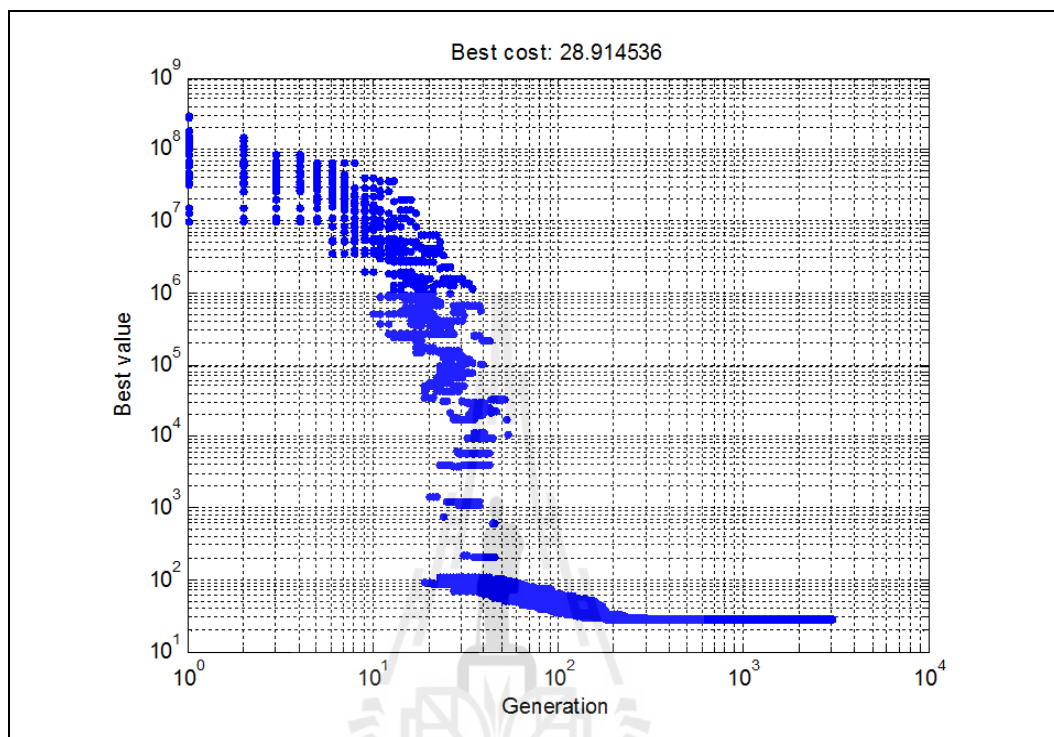
การทดสอบการแก้ปัญหาการประสานสัมพันธ์เพื่อหาผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์ในระบบทดสอบ 9 บัส โดยใช้วิธีกำหนดการเชิงเส้นและเปรียบเทียบผลลัพธ์กับวิธีการวิวัฒนาการ โดยใช้ผลต่าง แสดงผลลัพธ์ได้ดังตารางที่ 4.10 และกราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการ โดยใช้ผลต่างของระบบทดสอบ 9 บัส ดังรูปที่ 4.20

ตารางที่ 4.10 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 9 บัส

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
1	0.05	0.05
2	0.05	0.05
3	0.05	0.05
4	0.11	0.11
5	0.05	0.05
6	0.17	0.17
7	0.05	0.05
8	0.05	0.05
9	0.15	0.15
10	0.05	0.05
11	0.14	0.14
12	0.05	0.05
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	28.71	28.91
จำนวนการวนรอบที่ใช้ (รอบ)	10	3000

ผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 9 บัสแสดงผลดังตารางข้างต้น พบว่าค่า TDS และค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการ โดยใช้ผลต่างให้ผลลัพธ์เท่ากับ 28.71 วินาทีและ 28.91 วินาที ตามลำดับ ซึ่งผลลัพธ์มีค่าใกล้เคียงกันแสดงให้เห็นว่าอัลกอริทึมของ

วิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถนำมาใช้แก้ปัญหาในการประสานสัมพันธ์ในระบบทดสอบได้

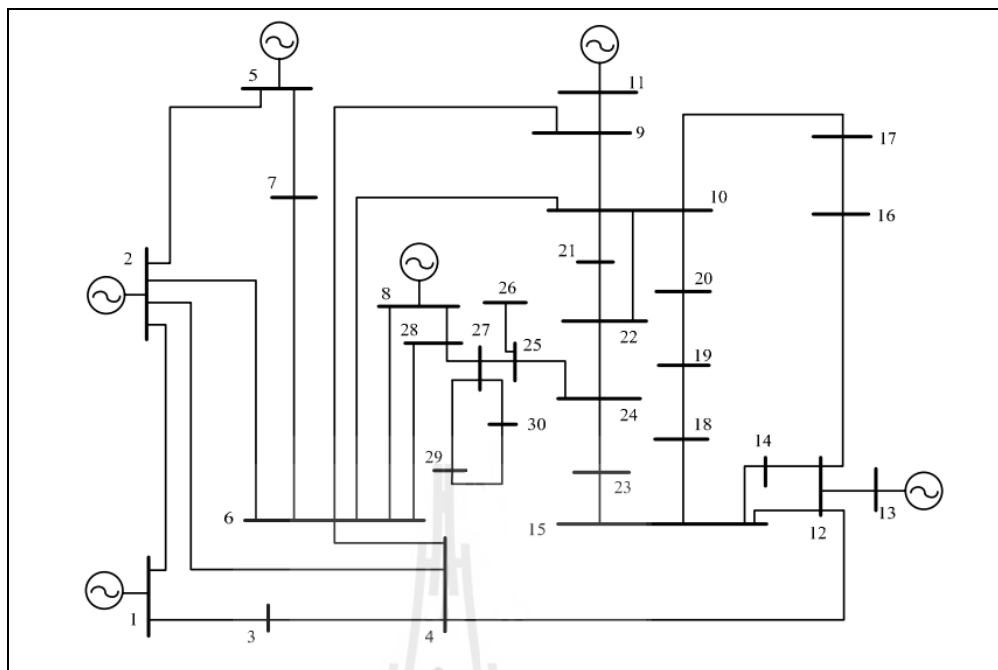


รูปที่ 4.20 กราฟแสดงการลู่เข้าของค่าตอบด้วยวิธีวิวัฒนาการ โดยใช้ผลต่างของระบบทดสอบ 9 บัส

จากกราฟแสดงให้เห็นถึงการลู่เข้าของค่าตอบของค่าฟังก์ชันวัตถุประสงค์ ซึ่งทำการวนรอบจำนวน 3000 รอบ ได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์เท่ากับ 28.91 วินาที

4.2.6 ระบบทดสอบ 30 บัส

พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 30 บัส แสดงดังรูปที่ 4.21 ใช้กับระดับแรงดันไฟฟ้า 138 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 6 เครื่องและรายละเอียดข้อมูลของระบบทดสอบ 30 บัส ได้แสดงไว้ในภาคผนวก ก

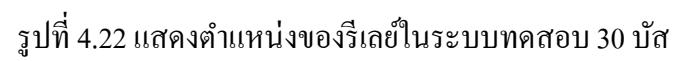


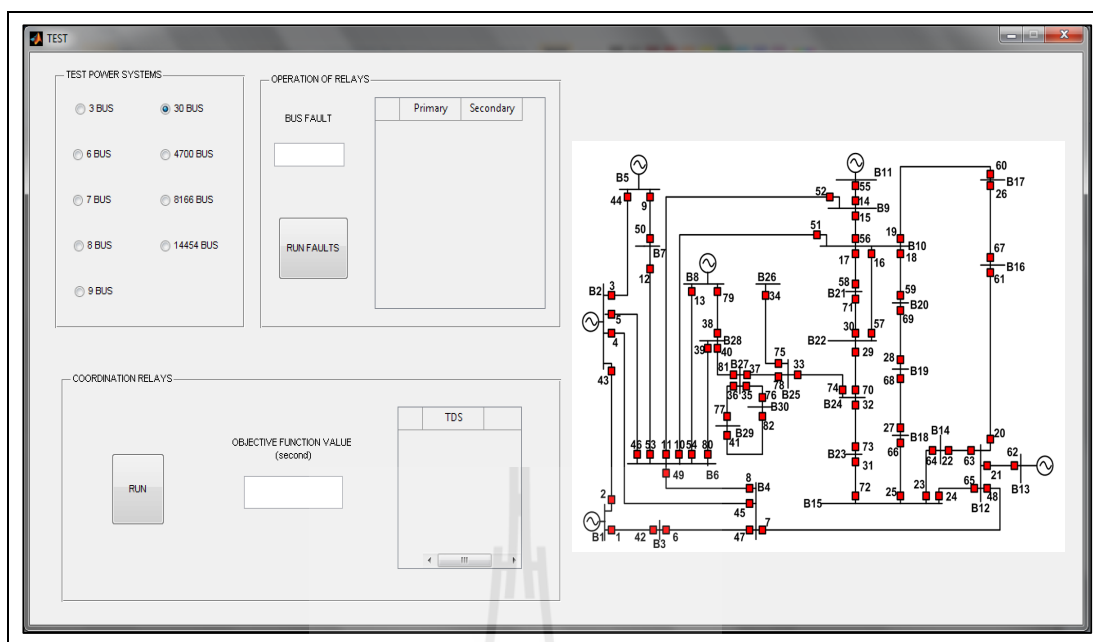
รูปที่ 4.21 ระบบทดสอบ 30 บัส

เพื่อกำหนดรีเลย์สำหรับติดตั้งในระบบทดสอบ ได้เขียนฟังก์ชันของโปรแกรม แมทแลป (รายละเอียดของฟังก์ชัน ได้แสดงไว้ในภาคผนวก ข) สำหรับกำหนดตำแหน่งของรีเลย์ รูปที่ 4.22 แสดงตำแหน่งของรีเลย์ที่อยู่ในระบบทดสอบ 30 บัส เพื่อใช้ในการประสานสัมพันธ์รีเลย์ จากรูปมีรีเลย์ทั้งหมดจำนวน 82 ตัว ตั้งแต่อรีเลย์หมายเลข 1 (R1) ถึงรีเลย์หมายเลข 82 (R82)

ในส่วนของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 30 บัส จะไม่แสดงผลในรูปแบบของตาราง เนื่องจากมีข้อมูลของรีเลย์หลักและรีเลย์สำรองเป็นจำนวนมาก จึงแสดงผลหลังจากการรันโปรแกรมในส่วนของหน้าต่างโปรแกรมส่วนต่อประสานกราฟิก กับผู้ใช้ที่ออกแบบไว้ดังรูปที่ 3.3 ในบทที่ 3

จากรูปที่ 4.23 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 30 บัสเมื่อเกิดการลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลลัพธ์ของการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงคที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์





รูปที่ 4.23 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 30 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบขนาด 30 บัสโดยใช้วิธีกำหนดการเชิงเส้นและเปรียบเทียบผลลัพธ์กับวิธีการวิวัฒนาการโดยใช้ผลต่างแสดงผลที่ได้ดังตารางที่ 4.11 และกราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่างของระบบทดสอบ 30 บัส ดังรูปที่ 4.24

ตารางที่ 4.11 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 30 ปี

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
1	0.11	0.70
2	0.14	0.25
3	0.24	0.16
4	0.11	0.08
5	0.07	0.07
6	0.08	0.05
7	0.34	0.09
8	0.30	0.05
9	0.16	0.08
10	0.05	0.19
11	0.29	1.09
12	0.20	0.49
13	0.05	0.15
14	0.05	1.08
15	0.24	0.21
16	0.60	0.14
17	0.82	0.05
18	0.45	0.05
19	0.84	0.81
20	0.83	0.05
21	0.05	0.25
22	0.33	0.06
23	1.09	1.09

ตารางที่ 4.11 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 30 บัส (ต่อ)

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
24	0.50	0.69
25	0.42	0.39
26	1.07	0.05
27	0.38	0.97
28	0.34	0.05
29	0.63	0.55
30	0.55	0.07
31	0.56	0.15
32	0.67	1.10
33	0.40	0.06
34	0.05	0.37
35	0.21	0.05
36	0.77	0.05
37	0.44	0.19
38	0.24	0.05
39	0.38	0.05
40	0.05	0.16
41	0.80	0.23
42	0.10	0.25
43	0.16	0.18
44	0.05	0.07
45	0.10	0.34
46	0.08	0.08

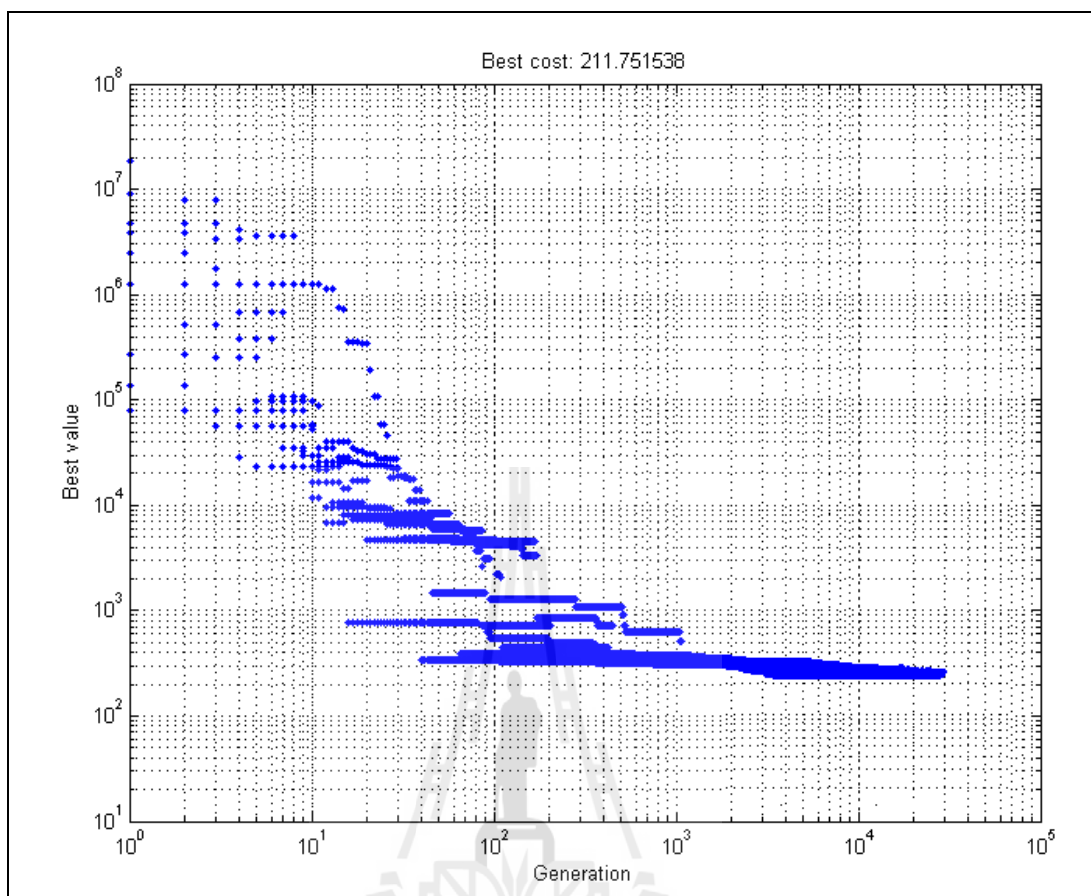
ตารางที่ 4.11 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 30 บัส (ต่อ)

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
47	0.20	0.24
48	0.25	0.19
49	0.32	1.02
50	0.13	0.40
51	0.20	1.03
52	0.20	0.93
53	0.31	0.48
54	0.38	0.81
55	0.05	0.05
56	0.28	0.13
57	0.61	0.18
58	0.84	0.91
59	0.11	0.06
60	0.37	0.75
61	0.73	0.34
62	0.05	0.12
63	0.60	0.36
64	0.76	0.07
65	0.47	0.77
66	0.31	0.05
67	0.87	0.32
68	0.46	0.71
69	0.64	0.66

ตารางที่ 4.11 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 30 บัส (ต่อ)

รีเลย์	กำหนดการเชิงเส้น (LP)	วิวัฒนาการโดยใช้ผลต่าง (DE)
	TDS	TDS
70	0.80	0.69
71	0.33	0.21
72	0.64	0.68
73	0.84	0.43
74	0.48	0.62
75	0.05	0.11
76	0.51	0.84
77	0.05	0.05
78	0.36	0.28
79	0.45	0.65
80	0.38	0.39
81	0.05	0.05
82	0.16	0.20
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	210.03	211.75
จำนวนการวนรอบที่ใช้ (รอบ)	15	30000

ผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 30 บัสแสดงผลดังตารางข้างต้น พบว่าค่า TDS และค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการ โดยใช้ผลต่างให้ผลลัพธ์เท่ากับ 210.03 วินาทีและ 211.75 วินาที ตามลำดับ ซึ่งผลลัพธ์มีค่าใกล้เคียงกันแสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถนำมาใช้แก้ปัญหาในการประสานสัมพันธ์รีเลย์ในระบบทดสอบได้

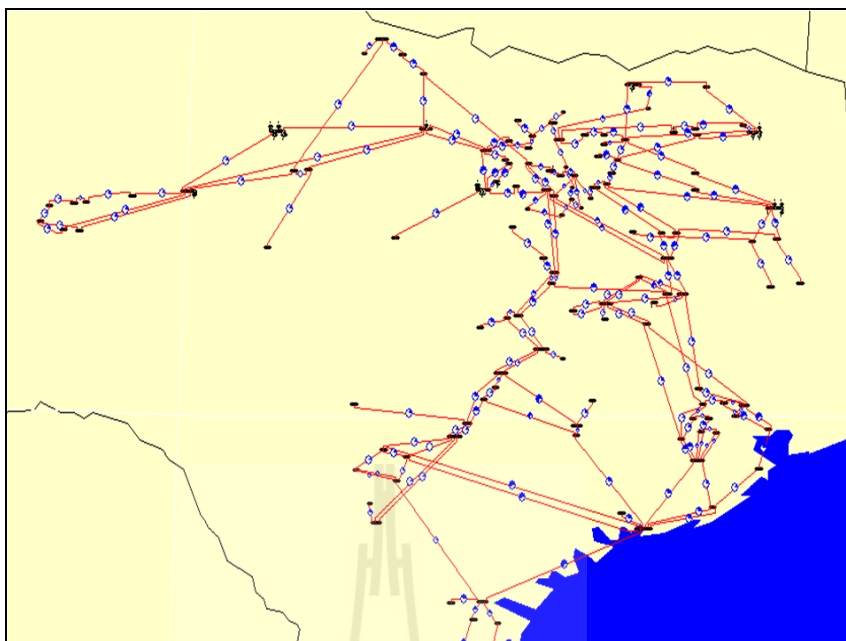


รูปที่ 4.24 กราฟแสดงการลู่เข้าของคำตอบด้วยวิธีวิวัฒนาการโดยใช้ผลต่างของระบบทดสอบ 30 บัส

จากกราฟแสดงให้เห็นถึงการลู่เข้าของคำตอบของค่าฟังก์ชันวัตถุประสงค์ ซึ่งทำการวนรอบจำนวน 30000 รอบ ได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์เท่ากับ 211.75 วินาที

4.2.7 ระบบทดสอบ 4700 บัส

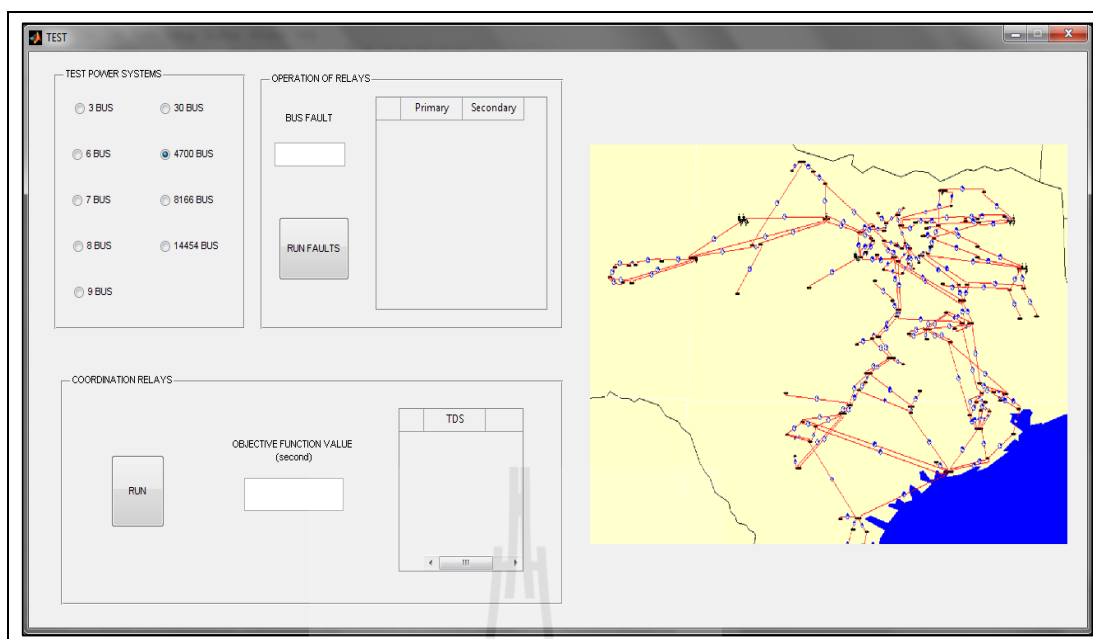
พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 4700 บัส แสดงดังรูปที่ 4.25 ใช้กับค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 408 เครื่อง และรายละเอียดข้อมูลของระบบทดสอบ 4700 บัส ได้แสดงไว้ในแผ่นซีดี



รูปที่ 4.25 ระบบทดสอบ 4700 บัส

จากระบบทดสอบข้างต้น มีรีเลย์ทั้งหมดจำนวน 10696 ตัว ซึ่งในส่วนของกระบวนการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 4700 บัส จะไม่แสดงผลในรูปแบบของตาราง เนื่องจากมีข้อมูลของรีเลย์หลักและรีเลย์สำรองเป็นจำนวนมาก จึงแสดงผลหลังจากการรันโปรแกรมในส่วนของหน้าต่างโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ออกแบบไว้ดังรูปที่ 3.3 ในบทที่ 3

จากรูปที่ 4.26 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับแสดงผลของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 4700 บัสเมื่อเกิดการลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลของการประสานสัมพันธ์รีเลย์ป้องกันกระแสน้ำแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



รูปที่ 4.26 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 4700 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบขนาด 4700 บัสโดยใช้วิธีกำหนดการเชิงเส้น แสดงผลลัพธ์ได้ดังตารางที่ 4.12 (เนื่องจากค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) มีข้อมูลเป็นจำนวนมาก จึงแสดงผลทั้งหมดไว้ในแผ่นซีดี)

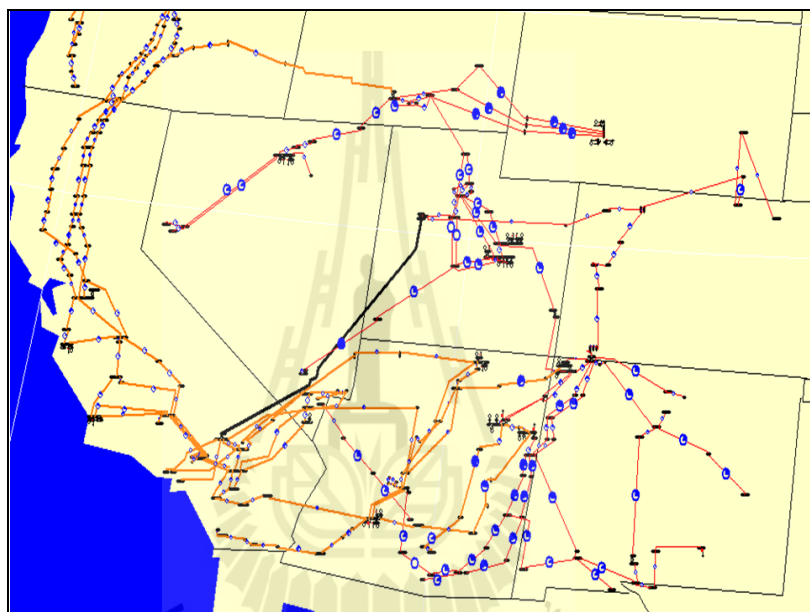
ตารางที่ 4.12 ค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 4700 บัส

ผลลัพธ์	วิธีกำหนดการเชิงเส้น (LP)
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	26440.82

ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 4700 บัสแสดงผลดังตารางข้างต้น พบว่าค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นให้ผลลัพธ์เท่ากับ 26440.82 วินาที แสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้นสามารถใช้หาค่าผลลัพธ์ที่ต้องการและแก้ปัญหาในการประสานสัมพันธ์รีเลย์ในระบบทดสอบได้

4.2.8 ระบบทดสอบ 8166 บัส

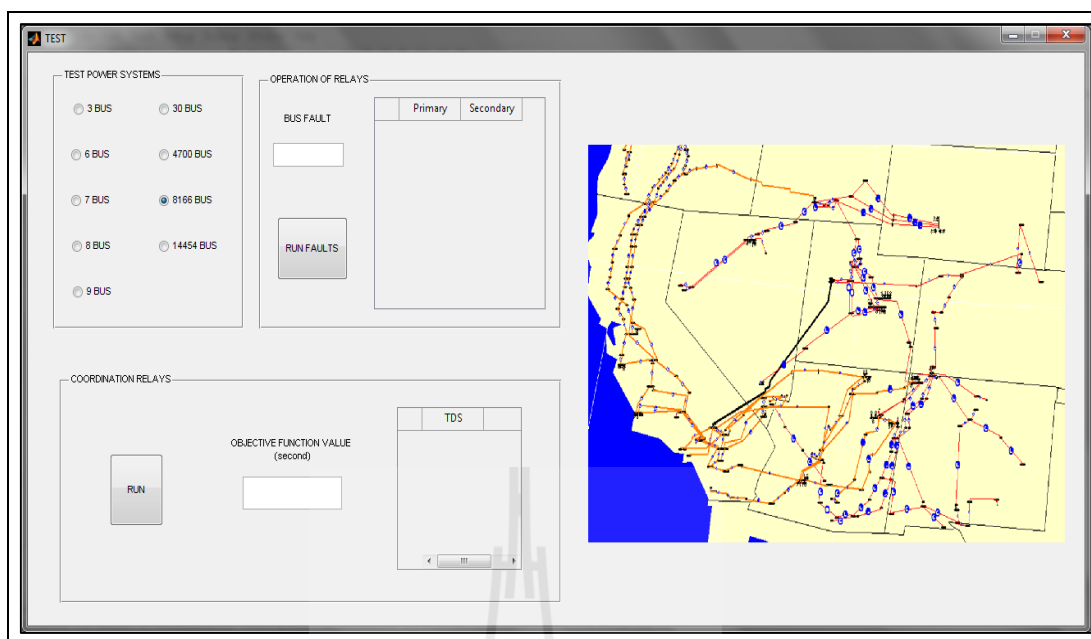
พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 8166 บัส แสดงดังรูปที่ 4.27 ใช้กับค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 1220 เครื่อง และรายละเอียดข้อมูลของระบบทดสอบ 8166 บัส ได้แสดงไว้ในแผ่นซีดี



รูปที่ 4.27 ระบบทดสอบ 8166 บัส

จากระบบทดสอบข้างต้น มีรีเลย์ทั้งหมดจำนวน 20642 ตัว ในส่วนของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 8166 บัส จะไม่แสดงผลในรูปแบบของตาราง เนื่องจากมีข้อมูลของรีเลย์หลักและรีเลย์สำรองเป็นจำนวนมาก จึงแสดงผลหลังจากการรันโปรแกรมในส่วนของหน้าต่างโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ออกแบบไว้ดังรูปที่ 3.3

จากรูปที่ 4.28 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับแสดงผลลัพธ์ของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 8166 บัสเมื่อเกิดการลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลลัพธ์ของการประสานสัมพันธ์รีเลย์ป้องกันกระแสนเกินแบบมีทิศทางโดยใช้วิธีกำหนดการเชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



รูปที่ 4.28 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 8166 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบขนาด 8166 บัสโดยใช้วิธีกำหนดการเชิงเส้น แสดงผลลัพธ์ได้ดังตารางที่ 4.13 (เนื่องจากค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) มีข้อมูลเป็นจำนวนมาก จึงแสดงผลทั้งหมดไว้ในแผ่นซีดี)

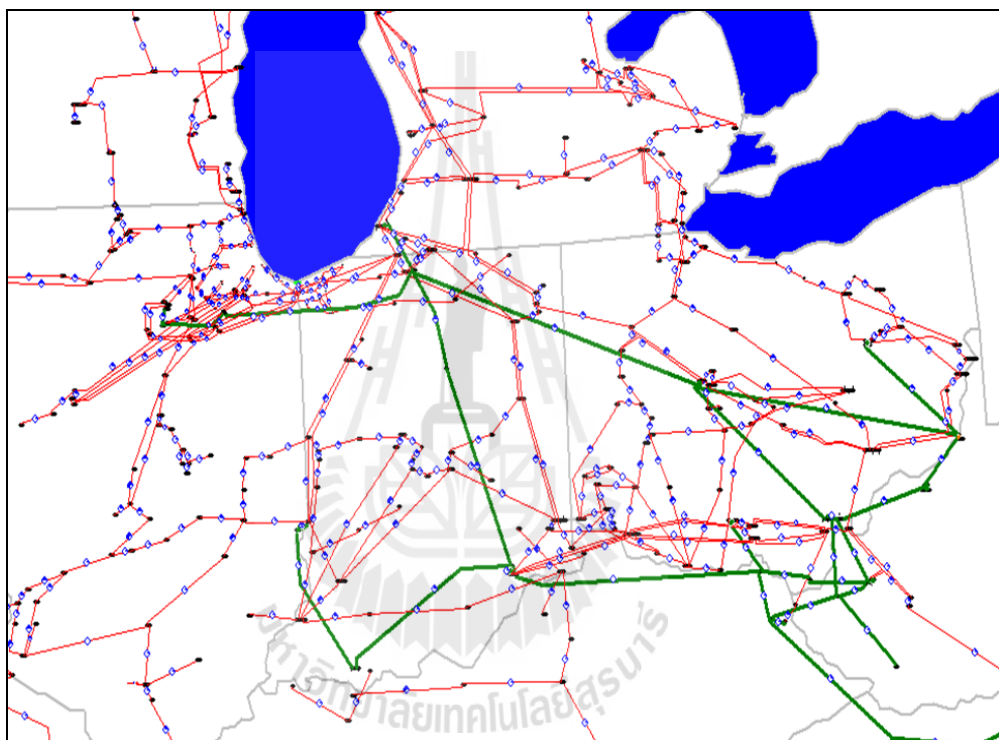
ตารางที่ 4.13 ค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 8166 บัส

ผลลัพธ์	วิธีกำหนดการเชิงเส้น (LP)
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	57211.25

ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 8166 บัสแสดงผลดังตารางข้างต้น พบว่าค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นให้ผลลัพธ์เท่ากับ 57211.25 วินาที แสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถใช้หาค่าผลลัพธ์ที่ต้องการและแก้ปัญหาในการประสานสัมพันธ์รีเลย์ในระบบทดสอบได้

4.2.9 ระบบทดสอบ 14454 บัส

พิจารณาฟังก์ชันวัตถุประสงค์ คือฟังก์ชันผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดที่เกิดการลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดที่เกิดการลัดวงจร รูปแบบสมการดังสมการที่ (3.2) และพิจารณาเงื่อนไขบังคับของปัญหาดังสมการที่ (3.5) – (3.9) ระบบทดสอบ 14454 บัส แสดงดังรูปที่ 4.29 ใช้กับค่ากำลังไฟฟ้าฐาน 100 MVA มีเครื่องกำเนิดไฟฟ้า 2085 เครื่อง และรายละเอียดข้อมูลของระบบทดสอบ 14454 บัส ได้แสดงไว้ในแผ่นซีดี

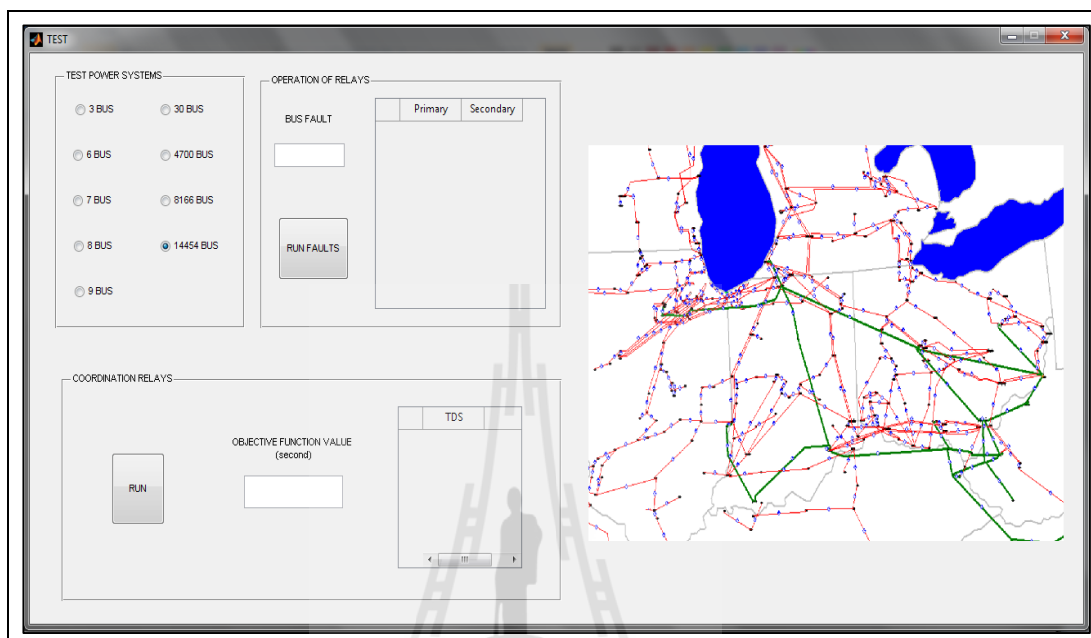


รูปที่ 4.29 ระบบทดสอบ 14454 บัส

จากระบบทดสอบข้างต้น มีรีเลย์ทั้งหมดจำนวน 37964 ตัว ในส่วนของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 14454 บัส จะไม่แสดงผลในรูปแบบของตาราง เนื่องจากมีข้อมูลของรีเลย์หลักและรีเลย์สำรองเป็นจำนวนมาก จึงแสดงผลหลังจากการรันโปรแกรมในส่วน of หน้าต่างโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ที่ออกแบบไว้ดังรูปที่ 3.3

จากรูปที่ 4.30 แสดงหน้าต่างการทำงานของ GUI ที่ใช้สำหรับการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบ 14454 บัสเมื่อเกิดการลัดวงจร ณ บัสใด ๆ รวมถึงแสดงผลลัพธ์ของการประสานสัมพันธ์รีเลย์ป้องกันกระแสนเกินแบบมีทิศทางโดยใช้วิธีกำหนดการ

เชิงเส้น ซึ่งมีส่วนสำหรับแสดงผลของค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) และค่าของฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์



รูปที่ 4.30 แสดงหน้าต่างการทำงานของ GUI ในระบบทดสอบ 14454 บัส

การทดสอบการแก้ปัญหาการประสานสัมพันธ์รีเลย์เพื่อหาผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในการประสานสัมพันธ์รีเลย์ในระบบทดสอบขนาด 14454 บัสโดยใช้วิธีกำหนดการเชิงเส้น แสดงผลลัพธ์ได้ดังตารางที่ 4.14 (เนื่องจากค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ (TDS) มีข้อมูลเป็นจำนวนมาก จึงแสดงผลทั้งหมดไว้ในแผ่นซีดี)

ตารางที่ 4.14 ค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 14454 บัส

ผลลัพธ์	วิธีกำหนดการเชิงเส้น (LP)
ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	99476.68

ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ในระบบทดสอบ 14454 บัสแสดงผลดังตารางข้างต้น พบว่าค่าฟังก์ชันวัตถุประสงค์ที่ได้จากวิธีกำหนดการเชิงเส้นให้ผลลัพธ์เท่ากับ

99476.68 วินาที แสดงให้เห็นว่าอัลกอริทึมของวิธีกำหนดการเชิงเส้นที่สร้างขึ้น สามารถใช้หาค่าผลลัพธ์ที่ต้องการและแก้ปัญหาในการประสานสัมพันธ์รีเลย์ในระบบทดสอบได้

4.3 สรุป

จากการแก้ปัญหการประสานสัมพันธ์รีเลย์ในระบบทดสอบทุกระบบ โดยการใช้วิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการ โดยใช้ผลต่าง ให้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ซึ่งเป็นเวลาที่เหมาะสมที่สุดในการประสานสัมพันธ์รีเลย์ แสดงผลลัพธ์ดังกล่าวไว้ในตารางที่ 4.15

ตารางที่ 4.15 ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ของระบบทดสอบ

ระบบทดสอบ	ค่าฟังก์ชันวัตถุประสงค์ (วินาที)	
	วิธีกำหนดการเชิงเส้น	วิธีวิวัฒนาการ โดยใช้ผลต่าง
ระบบทดสอบ 3 บัส	13.09	13.09
ระบบทดสอบ 6 บัส	26.65	26.75
ระบบทดสอบ 7 บัส	22.73	22.98
ระบบทดสอบ 8 บัส	29.41	29.46
ระบบทดสอบ 9 บัส	28.71	28.91
ระบบทดสอบ 30 บัส	210.03	211.75
ระบบทดสอบ 4700 บัส ใช้วิธีกำหนดการเชิงเส้นให้ผลลัพธ์เท่ากับ 26440.82 วินาที		
ระบบทดสอบ 8166 บัส ใช้วิธีกำหนดการเชิงเส้นให้ผลลัพธ์เท่ากับ 57211.25 วินาที		
ระบบทดสอบ 14454 บัส ใช้วิธีกำหนดการเชิงเส้นให้ผลลัพธ์เท่ากับ 99476.68 วินาที		

จากตารางที่ 4.15 สามารถสรุปได้ว่า

1. ระบบทดสอบขนาด 3 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 13.09 วินาทีและเมื่อใช้วิธีวิวัฒนาการโดยใช้ผลต่างจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 13.09 วินาที
2. ระบบทดสอบขนาด 6 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 26.65 วินาทีและเมื่อใช้วิธีวิวัฒนาการโดยใช้ผลต่างจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 26.75 วินาที

3. ระบบทดสอบขนาด 7 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 22.73 วินาทีและเมื่อใช้วิธีวิวัฒนาการโดยใช้ผลต่างจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 22.98 วินาที

4. ระบบทดสอบขนาด 8 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 29.41 วินาทีและเมื่อใช้วิธีวิวัฒนาการโดยใช้ผลต่างจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 29.46 วินาที

5. ระบบทดสอบขนาด 9 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 28.71 วินาทีและเมื่อใช้วิธีวิวัฒนาการโดยใช้ผลต่างจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 28.91 วินาที

6. ระบบทดสอบขนาด 30 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 210.03 วินาทีและเมื่อใช้วิธีวิวัฒนาการโดยใช้ผลต่างจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 211.75 วินาที

7. ระบบทดสอบขนาด 4700 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 26440.82 วินาที

8. ระบบทดสอบขนาด 8166 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 57211.25 วินาที

9. ระบบทดสอบขนาด 14454 บัส เมื่อใช้วิธีกำหนดการเชิงเส้นจะได้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 99476.68 วินาที

จากผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ข้างต้นพบว่า ผลลัพธ์จากการใช้อัลกอริทึมของวิธีกำหนดการเชิงเส้นให้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ใกล้เคียงกับวิธีวิวัฒนาการโดยใช้ผลต่างซึ่งค่าฟังก์ชันวัตถุประสงค์ที่ได้ในแต่ละระบบทดสอบ หมายถึงค่าผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ใกล้จุดลัดวงจรและผลรวมของเวลาในการทำงานของรีเลย์หลักตัวที่อยู่ไกลจุดลัดวงจรเมื่อพิจารณาการเกิดลัดวงจร ณ บัสต่าง ๆ ในระบบทดสอบทุกบัสหรือกล่าวได้ว่าค่าฟังก์ชันวัตถุประสงค์ของแต่ละระบบทดสอบคือผลรวมของเวลาที่น้อยที่สุดในการทำงานของรีเลย์หลักต้นสายและปลายสายซึ่งทำหน้าที่ป้องกันการลัดวงจรที่เกิดขึ้นในทุกกรณีของการเกิดลัดวงจรที่บัสในระบบทดสอบ

จากผลลัพธ์ของค่าจำนวนเท่าของเวลาในการทำงานของรีเลย์ (TDS) ในแต่ละระบบทดสอบ ผลลัพธ์ของค่า TDS ส่วนใหญ่ที่ได้จากการใช้อัลกอริทึมของวิธีกำหนดการเชิงเส้นให้ผลลัพธ์ของค่า TDS เท่ากับวิธีวิวัฒนาการโดยใช้ผลต่าง ซึ่งค่าของ TDS คือค่าที่ใช้ในการปรับตั้งค่าการหน่วงเวลาในการทำงานของรีเลย์แต่ละตัว โดยผลจากการเท่ากันของค่า TDS ทั้งสองวิธีการที่

นำเสนอเป็นผลมาจากการพิจารณาหาจุดค่าตอบหรือผลเฉลยของปัญหาค่าเหมาะที่สุดนี้ อยู่ภายใต้ขอบเขตหรือปริภูมิในการค้นหา โดยทั้งสองวิธีสามารถค้นหาผลเฉลยที่เป็นจุดต่ำสุดโดยรวมของปัญหา (global minimizer) ได้ ซึ่งค่าของผลเฉลยต่ำสุดนี้อยู่ในเงื่อนไขของขอบเขตการค้นหาหรือปริภูมิการค้นหา จึงได้ผลลัพธ์ของค่า TDS เท่ากันทั้งสองวิธี

ผลลัพธ์จากการใช้อัลกอริทึมของวิธีกำหนดการเชิงเส้นให้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ใกล้เคียงกับวิธีวิวัฒนาการโดยใช้ผลต่าง ซึ่งผลจากความใกล้เคียงกันของผลลัพธ์ทั้ง 2 วิธีนี้ ช่วยสนับสนุนถึงความถูกต้องของอัลกอริทึมวิธีกำหนดการเชิงเส้นที่สร้างขึ้นและผลลัพธ์ที่ได้จากการแก้ปัญหา เช่นเดียวกับผลลัพธ์ของค่าจำนวนเท่าของเวลาการทำงานรีเลย์และการแสดงผลลัพธ์ของการระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในระบบทดสอบต่าง ๆ สามารถแก้ปัญหาเพื่อหาผลลัพธ์ได้จากฟังก์ชันของโปรแกรมแมทแลปที่สร้างขึ้นได้และนำโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้มาช่วยในการแสดงผลของการประสานสัมพันธ์รีเลย์ทั้งในส่วน of ค่าฟังก์ชันวัตถุประสงค์ ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์และการแสดงผลลัพธ์ของการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรอง เพื่อเพิ่มความครบถ้วนในการแสดงผลที่บนหน้าต่าง GUI เพียงหน้าต่างเดียว ทำให้เกิดความสะดวกในการใช้งานของผู้ใช้มากยิ่งขึ้น



บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุป

งานวิจัยวิทยานิพนธ์นี้นำเสนอการแก้ปัญหาการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลังขนาดใหญ่ โดยงานวิจัยได้เริ่มจากการรวบรวมปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องทางด้านการประสานสัมพันธ์รีเลย์ในระบบทดสอบมาตรฐานต่าง ๆ ซึ่งจากการศึกษาค้นคว้าพบว่า การประสานสัมพันธ์รีเลย์มีการพัฒนามาอย่างต่อเนื่อง มีการทดสอบในระบบทดสอบมาตรฐานด้วยรูปแบบของฟังก์ชันวัตถุประสงค์และเงื่อนไขการทำงานของรีเลย์ที่แตกต่างกันไปในแต่ละรูปแบบของระบบทดสอบและวัตถุประสงค์ที่ต้องการทดสอบ มีหลายวิธีที่ใช้ในการแก้ปัญหาเพื่อวัตถุประสงค์ในการหาค่าที่เหมาะสมที่สุดในการป้องกันการลัดวงจรที่เกิดขึ้นในระบบทดสอบได้อย่างรวดเร็วและมีประสิทธิภาพ ซึ่งรายละเอียดการค้นคว้าต่าง ๆ ได้นำเสนอไว้ในบทที่ 2

การทำงานของรีเลย์ป้องกันกระแสเกิน รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง มีความสำคัญอย่างยิ่งในการป้องกันความผิดปกติในระบบไฟฟ้า เนื่องจากเป็นอุปกรณ์ป้องกันที่สำคัญในการประสานสัมพันธ์รีเลย์ที่มีผลกระทบโดยตรงต่อระบบทดสอบ วิธีการที่เลือกใช้สำหรับแก้ปัญหาในการประสานสัมพันธ์รีเลย์เพื่อปรับตั้งค่าการทำงานของรีเลย์ในระบบทดสอบคือวิธีกำหนดการเชิงเส้น โดยรายละเอียดของรีเลย์ป้องกันกระแสเกิน รีเลย์ป้องกันกระแสเกินแบบมีทิศทางและวิธีกำหนดการเชิงเส้นได้นำเสนอไว้ในบทที่ 2 ส่วนสมการของฟังก์ชันวัตถุประสงค์ที่ใช้หลักการทำงานของรีเลย์หลักในการป้องกันการลัดวงจรที่เกิดขึ้นเป็นอันดับแรกและสมการเงื่อนไขบังคับในการทำงานของรีเลย์หลักและรีเลย์สำรองได้นำเสนอไว้ในบทที่ 3

การดำเนินการในงานวิจัย แบ่งเป็น 2 ส่วน ส่วนแรกเป็นการแก้ปัญหาการประสานสัมพันธ์รีเลย์ ส่วนที่สองเป็นการระบุตำแหน่งของรีเลย์หลักและรีเลย์สำรองเมื่อเกิดการลัดวงจรขึ้น ณ บัสใด ๆ ในระบบทดสอบ ซึ่งทดสอบกับระบบทดสอบขนาดเล็กจนถึงระบบทดสอบขนาดใหญ่ ได้แก่ ระบบทดสอบ 3 บัส 6 บัส 7 บัส 8 บัส 9 บัส 30 บัส 4700 บัส 8166 บัสและ 14454 บัส รายละเอียดขั้นตอนการประสานสัมพันธ์รีเลย์ในระบบทดสอบดังกล่าวได้นำเสนอไว้ในบทที่ 3 และในการแก้ปัญหาการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางนั้น ใช้วิธีกำหนดการเชิงเส้น

เพื่อหาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดและค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ในระบบทดสอบและนำผลลัพธ์ที่ได้เปรียบเทียบกับวิธีการวิวัฒนาการ โดยใช้ผลต่าง ซึ่งรายละเอียดผลการทดสอบของระบบทดสอบต่าง ๆ ได้นำเสนอไว้ในบทที่ 4 และผลจากการดำเนินการในงานวิจัยสามารถสรุปได้ดังนี้

1. ผลการศึกษาการแก้ปัญหาการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางด้วยวิธีกำหนดการเชิงเส้นและวิธีวิวัฒนาการ โดยใช้ผลต่าง เมื่อนำผลลัพธ์มาเปรียบเทียบกันโดยพิจารณาฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับเดียวกันพบว่า จากการแก้ปัญหาในระบบทดสอบ 3 บัส 6 บัส 7 บัส 8 บัส 9 บัส 30 บัส 4700 บัส 8166 บัสและ 14454 บัส ทั้งสองวิธีการที่นำเสนอให้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ที่ใกล้เคียงกันมาก ซึ่งในปัจจุบันวิธีวิวัฒนาการโดยใช้ผลต่างได้ถูกพัฒนาขึ้นมาจากวิธีจินเนติกอัลกอริทึม (GA) โดยเป็นเทคนิคชาญฉลาดที่นิยมนำมาใช้งานกันอย่างแพร่หลาย ให้ผลลัพธ์อยู่ในเกณฑ์ที่ยอมรับได้และมีประสิทธิภาพในการแก้ปัญหาแบบไม่เชิงเส้นโดยทั่วไป เมื่อนำมาพิจารณาใช้แก้ปัญหาการประสานสัมพันธ์รีเลย์ด้วยฟังก์ชันวัตถุประสงค์และเงื่อนไขข้อบังคับเดียวกัน ผลลัพธ์ที่ได้จากวิธีวิวัฒนาการโดยใช้ผลต่างนี้ให้ผลลัพธ์ของค่าฟังก์ชันวัตถุประสงค์ใกล้เคียงกับผลลัพธ์ที่ได้จากวิธีกำหนดการเชิงเส้น จากผลของความใกล้เคียงกันของผลลัพธ์นี้ ส่งผลให้อัลกอริทึมของวิธีการกำหนดการเชิงเส้นที่สร้างขึ้นมีความถูกต้องและน่าเชื่อถือมากขึ้น ทำให้สามารถนำไปใช้แก้ปัญหาการประสานสัมพันธ์รีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบทดสอบต่าง ๆ ได้

2. สามารถใช้โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) ช่วยในการแสดงตำแหน่งของรีเลย์หลักและรีเลย์สำรองในระบบทดสอบเมื่อเกิดการลัดวงจรขึ้นในระบบทดสอบได้ รวมถึงแสดงค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดและค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ในระบบทดสอบต่าง ๆ ได้ ซึ่งประโยชน์จากการใช้โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ทำให้ช่วยเพิ่มความครบถ้วนในการแสดงผลที่ต้องการบนหน้าต่าง GUI เพียงหน้าต่างเดียว ทำให้เกิดความสะดวกในการใช้งานของผู้ใช้มากยิ่งขึ้น

5.2 ข้อเสนอแนะเพื่อพัฒนางานวิจัยในอนาคต

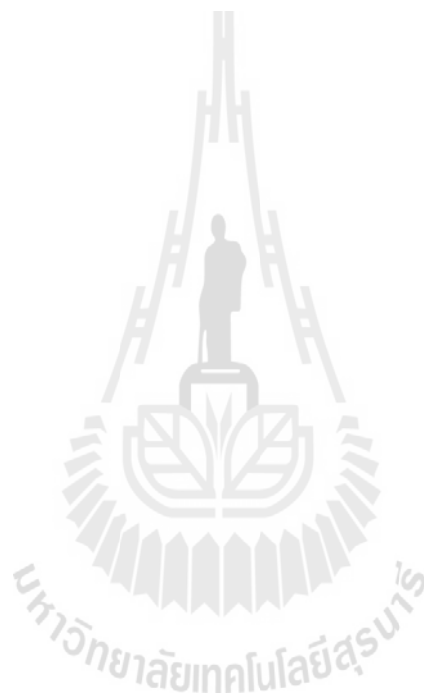
การประสานสัมพันธ์รีเลย์ในระบบทดสอบขนาดใหญ่โดยใช้อัลกอริทึมของวิธีกำหนดการเชิงเส้นรวมถึงเทคนิคชาญฉลาดอื่น ๆ จำเป็นต้องใช้คอมพิวเตอร์ที่มีสมรรถนะค่อนข้างสูงในการประมวลผล เนื่องจากอัลกอริทึมของแต่ละวิธีที่ค่อนข้างใช้เวลานานในการหาคำตอบและมีปริมาณของข้อมูลที่น่ามาพิจารณาเป็นจำนวนมาก

รายการอ้างอิง

- ธนัชชัย กุลวรรณิขพงษ์. (2550). เทคนิคการหาค่าเหมาะที่สุด. เอกสารประกอบการสอน สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.
- ธนัชชัย กุลวรรณิขพงษ์. (2551). การวิเคราะห์ระบบไฟฟ้ากำลัง. เอกสารประกอบการสอน สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.
- ธนัชชัย กุลวรรณิขพงษ์. (2552). การป้องกันและรีเลย์. เอกสารประกอบการสอน สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.
- ปริญญญา สงวนสัตย์. (2553). คู่มือ MATLAB ฉบับสมบูรณ์. นนทบุรี: ไอดีซี จำกัด.
- เผด็จ เผ่าละออ (2550). การใช้โปรแกรมเมทแลปสำหรับวิศวกรไฟฟ้า. เอกสารประกอบการสอน สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.
- Bedekar, P.P., Bhide, S.R., and Kale, V.S. (2009). Coordination of overcurrent relays in distribution system using linear programming technique. **IEEE Control, Automation Communication and Energy Conservation, 2009. INCACEC 2009.** Pages : 1-4.
- Elrafie, H.B., and Irving, M.R. (1993). Linear programming for directional overcurrent relay coordination in interconnected power systems with constraint relaxation. **Electric Power Systems Research.** Vol 27. Issue 3. Page : 209-216.
- Ezzeddine, M., and Kaczmarek, R. (2008). Reduction of effect of coordination constraints in the Linear optimization of operating times of overcurrent relays. **The 14th IEEE Mediterranean.** Pages : 707- 712.
- Giovanini, R., Hopkinson, K., Coury, D.V., and Thorp, J.S. (2006). A Primary and Backup Cooperative Protection System Based on Wide Area Agents. **Power Delivery IEEE Transactions on.** Volume:21. Issue 3. Pages : 1222–1230.
- Jamali, S., and Pourtandorost, M. (2004). New approach to coordination of distance relay zone-2 with overcurrent protection using linear programming methods. **IEEE UPEC2004 39th International.** Vol. 2. Page : 827-831.

- Kawahara, K., Sasaki, H., and Sugihara, H. (1997). An application of rule based system to the coordination of directional overcurrent relays. **Developments in Power System Protection, Sixth International Conference on** (Conf. Publ. No. 434). Pages : 58 – 61.
- Minglan, L., Anurag, K.S., and Noel, N.S. (2007). Modeling consideration in static and dynamic voltage stability studies of shipboard power system. **SCSC 2007**. Pages:53-60.
- Perez, L.G., Urdaneta, A.J., Sorrentino, E., Garayar, F., Urizar, A., Ledezma, J. C., Alcala G.D., Carrion, N., Canache, C., Fernandez, J., Sanz, O., and Guevara, F. (1998). Comparison of time coordination feasibility criteria for a sub-transmission system protection scheme. **IEEE International Caracas Conference on**. Pages : 314-319.
- Price, K., Storn, R., and Lampinen, J., *Differential Evolution - A Practical Approach to Global Optimization* Springer, 2005. Web site: <http://www.icsi.berkeley.edu/~storn/code.htm>
- Ramaswami, R., Damborg, M.J., and Venkata, S. (1990). Coordination of directional overcurrent relays in transmission systems. **IEEE Transaction on Power Delivery**. Vol.5. No.1.
- Rogers, C., and English, W. (1994). Automating relay coordination. **Computer Applications in Power**. Vol 7. No.3. Pages : 22-25.
- Thangaraj, R., Pant, M., and Deep, K. (2010). Optimal Coordination of Overcurrent Relays using Modified Differential Evolution Algorithms. **Engineering Applications of Artificial Intelligence, Elsevier Science**. Vol. 23 (5). Pages : 820-829.
- Urdaneta, A. J., Nadira, R., and Perez Jimenes, L. G. (1988). Optimal coordination of directional overcurrent relays in interconnected power system. **IEEE Transaction on Power Delivery**. Vol 3. No. 3 . Pages : 903-911.
- Urdaneta, A.J., Restrepo, H., Marquez, S., and Sanchez, J. (1996). Coordination of directional overcurrent relays timing using linear programming. **IEEE Transaction on Power Delivery**. Vol 11. No. 1. Pages : 122-129.
- Urdaneta, A.J., Perez, L.G., Gomez, J.F., Feijoo, B., and Gonzalez, M. (2001). Presolve analysis and interior point solutions of the linear programming coordination problem of directional overcurrent relays. **International Journal of Electric Power & Energy Systems**. Vol 23. Issue 8.

Yu, X., and Singh, C. (2006). Probabilistic Analysis of Load ability in Composite Power Systems Considering Security Constraints. **Power Systems Conference and Exposition, IEEE PES**. Pages : 832-838.



ภาคผนวก ก

ข้อมูลของระบบทดสอบ



ระบบทดสอบ

ระบบทดสอบที่ใช้ทดสอบการประสานสัมพันธ์ การระบุตำแหน่งรีเลย์หลักและรีเลย์สำรองในงานวิจัยวิทยานิพนธ์นี้ เป็นระบบทดสอบที่มีอยู่ในโปรแกรม Power World Simulator นำมาทดสอบจำนวน 9 ระบบ ประกอบด้วยระบบทดสอบขนาด 3 บัส 6 บัส 7 บัส 8 บัส 9 บัส 30 บัส 4700 บัส 8166 บัสและ 14454 บัส ซึ่งรายละเอียดข้อมูลของระบบทดสอบแสดงได้ดังนี้

ก.1 ระบบทดสอบ 3 บัส

ระบบทดสอบ 3 บัส ใช้ในระดับแรงดันไฟฟ้า 345 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีข้อมูลบัส ข้อมูลสายส่งและข้อมูลเครื่องกำเนิดไฟฟ้า ดังตารางที่ ก.1 ตารางที่ ก.2 และตารางที่ ก.3 ตามลำดับ

ตารางที่ ก.1 ข้อมูลบัสของระบบทดสอบ 3 บัส

Bus	Voltage (kV)	Magnitude voltage (p.u.)	Angle (degree)	Load	
				MW	Mvar
1	345.00	1.00	0.00	0.00	0.00
2	345.00	1.00	-1.91	0.00	0.00
3	345.00	1.00	-3.82	100.00	0.00

ตารางที่ ก.2 ข้อมูลสายส่งของระบบทดสอบ 3 บัส

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
1	1	2	0.00	0.10	0.00	0.00	0.00
2	1	3	0.00	0.10	0.00	0.00	0.00
3	2	3	0.00	0.10	0.00	0.00	0.00

ตารางที่ ก.3 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 3 บัส

Bus	R1 (p.u.)	jX1 (p.u.)	jX0 (p.u.)	Rg (p.u.)	jXg (p.u.)	Winding connection (Y=1)
1	0.00	1.00	0.00	0.00	0.00	1.00
2	0.00	1.00	0.00	0.00	0.00	1.00
3	0.00	1.00	0.00	0.00	0.00	1.00

ก.2 ระบบทดสอบ 6 บัส

ระบบทดสอบ 6 บัส ใช้ในระดับแรงดันไฟฟ้า 138 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีข้อมูลบัส ข้อมูลสายส่งและข้อมูลเครื่องกำเนิดไฟฟ้า ดังตารางที่ ก.4 ตารางที่ ก.5 และตารางที่ ก.6 ตามลำดับ

ตารางที่ ก.4 ข้อมูลบัสของระบบทดสอบ 6 บัส

Bus	Voltage (kV)	Magnitude voltage (p.u.)	Angle (degree)	Load	
				MW	Mvar
1	138.00	1.02	0.00	100.00	20.00
2	138.00	1.04	2.77	100.00	20.00
3	138.00	1.01	-3.68	100.00	20.00
4	138.00	1.03	1.03	100.00	20.00
5	138.00	0.99	-2.02	100.00	50.00
6	138.00	0.99	-3.47	100.00	10.00

ตารางที่ ก.5 ข้อมูลสายส่งของระบบทดสอบ 6 บัส

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
1	1	2	0.04	0.08	0.01	0.00	0.00
2	1	5	0.04	0.08	0.01	0.00	0.00
3	2	4	0.04	0.08	0.01	0.00	0.00
4	3	5	0.04	0.08	0.01	0.00	0.00
5	3	6	0.04	0.08	0.01	0.00	0.00
6	5	4	0.04	0.08	0.01	0.00	0.00
7	6	4	0.04	0.08	0.01	0.00	0.00

ตารางที่ ก.6 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 6 บัส

Bus	R1 (p.u.)	jX1 (p.u.)	jX0 (p.u.)	Rg (p.u.)	jXg (p.u.)	Winding connection (Y=1)
1	0.00	1.00	0.00	0.00	0.00	1.00
2	0.00	1.00	0.00	0.00	0.00	1.00
3	0.00	1.00	0.00	0.00	0.00	1.00
4	0.00	1.00	0.00	0.00	0.00	1.00

ก.3 ระบบทดสอบ 7 บัส

ระบบทดสอบ 7 บัส ใช้ในระดับแรงดันไฟฟ้า 138 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีข้อมูลบัส ข้อมูลสายส่งและข้อมูลเครื่องกำเนิดไฟฟ้า ดังตารางที่ ก.7 ตารางที่ ก.8 และตารางที่ ก.9 ตามลำดับ

ตารางที่ ก.7 ข้อมูลบัสของระบบทดสอบ 7 บัส

Bus	Voltage (kV)	Magnitude voltage (p.u.)	Angle (degree)	Load	
				MW	Mvar
1	138.00	1.05	4.43	0.00	0.00
2	138.00	1.04	3.17	40.00	20.00
3	138.00	0.99	-1.41	150.00	40.00
4	138.00	1.00	-0.43	80.00	30.00
5	138.00	1.02	-1.52	130.00	40.00
6	138.00	1.04	3.18	200.00	0.00
7	138.00	1.04	0.00	200.00	0.00

ตารางที่ ก.8 ข้อมูลสายส่งของระบบทดสอบ 7 บัส

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
1	1	3	0.02	0.24	0.03	0.00	0.00
2	1	2	0.01	0.05	0.25	0.00	0.00
3	2	6	0.01	0.06	0.03	0.00	0.00
4	2	5	0.01	0.12	0.02	0.00	0.00
5	2	4	0.02	0.18	0.02	0.00	0.00
6	2	3	0.02	0.18	0.02	0.00	0.00
7	3	4	0.002	0.03	0.01	0.00	0.00
8	4	5	0.02	0.24	0.03	0.00	0.00
9	6	7	0.02	0.24	0.03	0.00	0.00
10	6	7	0.02	0.24	0.03	0.00	0.00
11	7	5	0.005	0.06	0.02	0.00	0.00

ตารางที่ ก.9 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 7 บัส

Bus	R1 (p.u.)	jX1 (p.u.)	jX0 (p.u.)	Rg (p.u.)	jXg (p.u.)	Winding connection (Y=1)
1	0.00	1.00	0.00	0.00	0.00	1.00
2	0.00	1.00	0.00	0.00	0.00	1.00
4	0.00	1.00	0.00	0.00	0.00	1.00
6	0.00	1.00	0.00	0.00	0.00	1.00
7	0.00	1.00	0.00	0.00	0.00	1.00

ก.4 ระบบทดสอบ 8 บัส

ระบบทดสอบ 8 บัส ใช้ในระดับแรงดันไฟฟ้า 150 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีข้อมูลบัส ข้อมูลสายส่ง ข้อมูลเครื่องกำเนิดไฟฟ้าและข้อมูลหม้อแปลงไฟฟ้า ดังตารางที่ ก.10 ตารางที่ ก.11 ตารางที่ ก.12 และตารางที่ ก.13 ตามลำดับ

ตารางที่ ก.10 ข้อมูลบัสของระบบทดสอบ 8 บัส

Bus	Voltage (kV)	Magnitude voltage (p.u.)	Angle (degree)	Load	
				MW	Mvar
1	150.00	0.99	-1.92	0.00	0.00
2	150.00	0.99	-2.36	30.00	10.00
3	150.00	0.99	-2.60	40.00	15.00
4	150.00	0.99	-2.76	25.00	10.00
5	150.00	0.99	-2.64	10.00	8.00
6	150.00	0.99	-2.41	20.00	5.00
7	10.00	1.00	-2.41	0.00	0.00
8	10.00	1.00	0.00	0.00	0.00

ตารางที่ ก.11 ข้อมูลสายส่งของระบบทดสอบ 8 บัส

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
1	1	2	0.0018	0.0222	0.00	0.00	0.00
2	1	3	0.0018	0.0222	0.00	0.00	0.00
3	1	6	0.0022	0.0222	0.00	0.00	0.00
4	8	1	0.0000	0.02667	0.00	0.00	0.00
5	2	6	0.0018	0.0200	0.00	0.00	0.00
6	3	4	0.0018	0.0200	0.00	0.00	0.00
7	4	5	0.0022	0.0200	0.00	0.00	0.00
8	5	6	0.0022	0.0200	0.00	0.00	0.00

ตารางที่ ก.11 ข้อมูลสายส่งของระบบทดสอบ 8 บัส (ต่อ)

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
9	6	7	0.0000	0.02667	0.00	0.00	0.00

ตารางที่ ก.12 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 8 บัส

Bus	R1 (p.u.)	jX1 (p.u.)	jX0 (p.u.)	Rg (p.u.)	jXg (p.u.)	Winding connection (Y=1)
7	0.00	1.00	0.00	0.00	0.00	1.00
8	0.00	1.00	0.00	0.00	0.00	1.00

ตารางที่ ก.13 ข้อมูลหม้อแปลงไฟฟ้าของระบบทดสอบ 8 บัส

NO	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Tap	jX0 (p.u.)	Pri winding (Y=1)	Sec winding (Y=1)	Pri Rg	Pri jXg	Sec Rg	Sec jXg
1	8	1	0.00	0.00	1	0.00	1.00	1.00	0.00	0.00	0.00	0.00
2	6	7	0.00	0.00	1	0.00	1.00	1.00	0.00	0.00	0.00	0.00

ก.5 ระบบทดสอบ 9 บัส

ระบบทดสอบ 9 บัส ใช้ในระดับแรงดันไฟฟ้า 230 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีข้อมูลบัส ข้อมูลสายส่ง ข้อมูลเครื่องกำเนิดไฟฟ้าและข้อมูลหม้อแปลงไฟฟ้า ดังตารางที่ ก.14 ตารางที่ ก.15 ตารางที่ ก.16 และตารางที่ ก.17 ตามลำดับ

ตารางที่ ก.14 ข้อมูลบัสของระบบทดสอบ 9 บัส

Bus	Voltage (kV)	Magnitude voltage (p.u.)	Angle (degree)	Load	
				MW	Mvar
1	16.50	1.00	0.00	0.00	0.00
2	18.00	1.00	-7.62	0.00	0.00
3	13.80	1.00	-7.40	0.00	0.00
4	230.00	1.02	-2.95	0.00	0.00
5	230.00	1.02	-5.23	20.00	15.00
6	230.00	1.01	-7.40	0.00	0.00
7	230.00	1.01	-8.67	50.00	30.00
8	230.00	1.02	-7.62	0.00	0.00
9	230.00	1.03	-5.23	20.00	10.00

ตารางที่ ก.15 ข้อมูลสายส่งของระบบทดสอบ 9 บัส

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
1	1	4	0.00	0.06	0.00	0.00	0.00
2	2	8	0.00	0.06	0.00	0.00	0.00
3	3	6	0.00	0.06	0.00	0.00	0.00
4	4	5	0.02	0.09	0.00	0.00	0.00
5	9	4	0.01	0.09	0.00	0.00	0.00
6	5	6	0.04	0.17	0.00	0.00	0.00
7	6	7	0.01	0.10	0.00	0.00	0.00
8	7	8	0.01	0.07	0.00	0.00	0.00
9	8	9	0.03	0.16	0.00	0.00	0.00

ตารางที่ ก.16 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 9 บัส

Bus	R1 (p.u.)	jX1 (p.u.)	jX0 (p.u.)	Rg (p.u.)	jXg (p.u.)	Winding connection (Y=1)
1	0.00	1.00	0.00	0.00	0.00	1.00
2	0.00	1.00	0.00	0.00	0.00	1.00
3	0.00	1.00	0.00	0.00	0.00	1.00

ตารางที่ ก.17 ข้อมูลหม้อแปลงไฟฟ้าของระบบทดสอบ 9 บัส

NO	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Tap	jX0 (p.u.)	Pri winding (Y=1)	Sec winding (Y=1)	Pri Rg	Pri jXg	Sec Rg	Sec jXg
1	1	4	0.00	0.06	1.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00
2	3	6	0.00	0.06	1.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00
3	2	8	0.00	0.06	1.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00

ก.6 ระบบทดสอบ 30 บัส

ระบบทดสอบ 30 บัส ใช้ในระดับแรงดันไฟฟ้า 138 kV ที่ค่ากำลังไฟฟ้าฐาน 100 MVA มีข้อมูลบัส ข้อมูลสายส่งและข้อมูลเครื่องกำเนิดไฟฟ้า ดังตารางที่ ก.18 ตารางที่ ก.19 และตารางที่ ก.20 ตามลำดับ

ตารางที่ ก.18 ข้อมูลบัสของระบบทดสอบ 30 บัส

Bus	Voltage (kV)	Magnitude voltage (p.u.)	Angle (degree)	Load	
				MW	Mvar
1	138	1.00	0.00	0.00	0.00
2	138	1.00	-0.94	21.70	12.70
3	138	0.99	-2.95	2.40	1.20
4	138	0.98	-3.51	7.60	1.60
5	138	0.99	-3.01	0.00	0.00
6	138	0.98	-4.07	0.00	0.00
7	138	0.97	-4.19	23.00	11.00
8	138	0.98	-4.67	30.00	0.00
9	138	0.98	-6.39	0.00	0.00
10	138	0.98	-7.62	5.80	2.00
11	138	0.98	-6.39	0.00	0.00
12	138	0.99	-6.52	11.20	7.50
13	138	1.00	-4.41	0.00	0.00
14	138	0.98	-7.65	6.20	1.60
15	138	0.98	-8.00	15.00	2.50
16	138	0.98	-7.50	3.50	1.80
17	138	0.97	-8.14	18.00	5.80
18	138	0.96	-9.55	3.20	0.90
19	138	0.95	-10.27	15.00	3.40
20	138	0.96	-10.08	15.00	0.70
21	138	0.99	-7.59	17.50	11.20
22	138	1.00	-7.42	0.00	0.00
23	138	1.00	-7.49	12.00	1.60
24	138	0.99	-6.70	8.70	6.70
25	138	0.99	-3.47	0.00	0.00
26	138	0.97	-3.92	3.50	2.30
27	138	1.00	-1.21	0.00	0.00
28	138	0.99	-3.92	0.00	0.00
29	138	0.98	-2.50	2.40	0.90
30	138	0.97	-3.42	10.60	1.90

ตารางที่ ก.19 ข้อมูลสายส่งของระบบทดสอบ 30 บัส

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
1	1	3	0.05	0.19	0.01	0.00	0.00
2	1	2	0.02	0.06	0.01	0.00	0.00

ตารางที่ ก.19 ข้อมูลสายส่งของระบบทดสอบ 30 บัส (ต่อ)

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
3	2	5	0.05	0.20	0.01	0.00	0.00
4	2	4	0.06	0.17	0.01	0.00	0.00
5	2	6	0.06	0.18	0.01	0.00	0.00
6	3	4	0.01	0.04	0.00	0.00	0.00
7	4	12	0.00	0.26	0.00	0.00	0.00
8	4	6	0.01	0.04	0.00	0.00	0.00
9	5	7	0.05	0.12	0.01	0.00	0.00
10	6	10	0.00	0.56	0.00	0.00	0.00
11	6	9	0.00	0.21	0.00	0.00	0.00
12	7	6	0.03	0.08	0.00	0.00	0.00
13	8	6	0.01	0.04	0.00	0.00	0.00
14	9	11	0.00	0.21	0.00	0.00	0.00
15	9	10	0.00	0.11	0.00	0.00	0.00
16	10	22	0.07	0.15	0.00	0.00	0.00
17	10	21	0.04	0.08	0.00	0.00	0.00
18	10	20	0.09	0.21	0.00	0.00	0.00
19	10	17	0.03	0.09	0.00	0.00	0.00
20	12	16	0.10	0.20	0.00	0.00	0.00
21	12	13	0.00	0.14	0.00	0.00	0.00
22	14	12	0.12	0.26	0.00	0.00	0.00
23	15	14	0.22	0.20	0.00	0.00	0.00
24	15	12	0.07	0.13	0.00	0.00	0.00
25	15	18	0.11	0.22	0.00	0.00	0.00
26	17	16	0.08	0.19	0.00	0.00	0.00
27	18	19	0.06	0.13	0.00	0.00	0.00
28	19	20	0.03	0.07	0.00	0.00	0.00
29	22	24	0.12	0.18	0.00	0.00	0.00
30	22	21	0.01	0.02	0.00	0.00	0.00
31	23	15	0.10	0.20	0.00	0.00	0.00
32	24	23	0.13	0.27	0.00	0.00	0.00
33	25	24	0.19	0.33	0.00	0.00	0.00
34	26	25	0.25	0.38	0.00	0.00	0.00
35	27	30	0.32	0.60	0.00	0.00	0.00
36	27	29	0.22	0.42	0.00	0.00	0.00
37	27	25	0.11	0.21	0.00	0.00	0.00
38	28	8	0.06	0.20	0.01	0.00	0.00
39	28	6	0.02	0.06	0.00	0.00	0.00

ตารางที่ ก.19 ข้อมูลสายส่งของระบบทดสอบ 30 บัส (ต่อ)

Line number	From bus	To bus	R1 (p.u.)	jX1 (p.u.)	Half-line charging (p.u.)	R0 (p.u.)	jX0 (p.u.)
40	28	27	0.00	0.40	0.00	0.00	0.00
41	29	30	0.24	0.45	0.00	0.00	0.00

ตารางที่ ก.20 ข้อมูลเครื่องกำเนิดไฟฟ้าของระบบทดสอบ 30 บัส

Bus	R1 (p.u.)	jX1 (p.u.)	jX0 (p.u.)	Rg (p.u.)	jXg (p.u.)	Winding connection (Y=1)
1	0.00	1.00	0.00	0.00	0.00	1.00
2	0.00	1.00	0.00	0.00	0.00	1.00
13	0.00	1.00	0.00	0.00	0.00	1.00
22	0.00	1.00	0.00	0.00	0.00	1.00
23	0.00	1.00	0.00	0.00	0.00	1.00
27	0.00	1.00	0.00	0.00	0.00	1.00

ก.7 ระบบทดสอบ 4700 บัส ระบบทดสอบ 8166 บัส และระบบทดสอบ 14454 บัส

เนื่องจากทั้ง 3 ระบบ เป็นระบบทดสอบขนาดใหญ่ มีข้อมูลบัส ข้อมูลสายส่ง ข้อมูลเครื่องกำเนิดไฟฟ้าและข้อมูลหม้อแปลงไฟฟ้าเป็นจำนวนมาก จึงแสดงข้อมูลดังกล่าวข้างต้นไว้ในแผ่นซีดี

ภาคผนวก ข

โปรแกรมสำหรับการหาผลลัพธ์ของงานวิจัยในวิทยานิพนธ์



ข.1 รูปแบบเมทริกซ์ของฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับ

ข.1.1 รูปแบบเมทริกซ์ของฟังก์ชันวัตถุประสงค์

ตารางที่ ข.1 รูปแบบเมทริกซ์ฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 3 บัส

$T_{pri_cl_in}^i$			$T_{pri_far_bus}^i$		
TDS^i	a^i	b^i	TDS^j	c^i	d^i
1	15.97	2.79	4	7.61	2.79
2	10.65	3.72	5	5.07	3.72
3	15.97	2.79	6	7.61	2.79
4	15.97	2.79	1	7.61	2.79
5	10.65	3.72	2	5.07	3.72
6	15.97	2.79	3	7.61	2.79

จากตารางข้างต้น เป็นตัวอย่างรูปแบบเมทริกซ์ฟังก์ชันวัตถุประสงค์ของระบบทดสอบ 3 บัส ประกอบด้วยค่าพารามิเตอร์ต่าง ๆ ดังนี้

$T_{pri_cl_in}^i$ คือ เวลาทำงานของรีเลย์หลักที่อยู่ใกล้จุดลัดวงจร

$T_{pri_far_bus}^i$ คือ เวลาทำงานของรีเลย์หลักที่อยู่ไกลจุดลัดวงจร

a^i และ c^i คือ พารามิเตอร์ของค่ากระแสลัดวงจรที่ผ่านหม้อแปลงกระแส

b^i และ d^i คือ พารามิเตอร์ของค่ากระแสโหลดที่ผ่านหม้อแปลงกระแส

TDS^i และ TDS^j คือ คู่มหามเลขของรีเลย์หลักตัวที่อยู่ใกล้จุดลัดวงจรและรีเลย์หลักตัวที่อยู่ไกลจุดลัดวงจร เมื่อพิจารณาการเกิดลัดวงจร ณ จุดใดจุดหนึ่ง

ข.1.2 รูปแบบเมทริกซ์ของเงื่อนไขบังคับ

ตารางที่ ข.2 รูปแบบเมทริกซ์เงื่อนไขบังคับของระบบทดสอบ 3 บัส

T_{backup}^i			$T_{primary}^i$		
TDS^p	e^i	f^i	TDS^q	g^i	h^i
5	5.07	3.72	1	15.97	2.79
4	7.61	2.79	2	10.65	3.72
1	7.61	2.79	3	15.97	2.79
6	7.61	2.79	4	15.97	2.79
3	7.61	2.79	5	10.65	3.72
2	5.07	3.72	6	15.97	2.79

จากตารางข้างต้น เป็นตัวอย่างรูปแบบเมทริกซ์เงื่อนไขบังคับของระบบทดสอบ 3 บัส ประกอบด้วยค่าพารามิเตอร์ต่าง ๆ ดังนี้

$T_{primary}^i$	คือ เวลาทำงานของรีเลย์หลัก
T_{backup}^i	คือ เวลาทำงานของรีเลย์สำรอง
e^i และ g^i	คือ พารามิเตอร์ของค่ากระแสลัดวงจรที่ผ่านหม้อแปลงกระแส
f^i และ h^i	คือ พารามิเตอร์ของค่ากระแสโหลดปกติที่ผ่านหม้อแปลงกระแส
TDS^q และ TDS^p	คือ คู่มหามเลขของรีเลย์หลักและรีเลย์สำรอง ตามลำดับ

ข.2 ฟังก์ชันในโปรแกรม MATLAB ที่ใช้ในวิทยานิพนธ์

ฟังก์ชันต่อไปนี้เป็นฟังก์ชันที่สร้างขึ้นสำหรับการแก้ปัญหาการประสานสัมพันธรีเลย์ ป้องกันกระแสเกินแบบมีทิศทางในระบบทดสอบ ซึ่งมีรายละเอียดของชื่อฟังก์ชันและหน้าที่ดังนี้

ชื่อฟังก์ชัน	หน้าที่
B3LP_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 3 บัส
bus6naps_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 6 บัส
B7flatlp_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 7 บัส
bus8_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 8 บัส
bus9_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 9 บัส
buslp30_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 30 บัส
P99sp_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 4700 บัส
LS1A98_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 8166 บัส
Wisconsin_pwb	ฟังก์ชันข้อมูลของระบบทดสอบ 14454 บัส
relaydata(linedata,tfddata)	ฟังก์ชันสำหรับการกำหนดตำแหน่งของรีเลย์ในระบบทดสอบเมื่อ linedata คือข้อมูลสายส่งของระบบทดสอบ และ tfdata คือข้อมูลหม้อแปลงของระบบทดสอบ
parameter_OBJ(relay_data)	ฟังก์ชันสำหรับสร้างเมทริกซ์พารามิเตอร์ของฟังก์ชันวัตถุประสงค์ เมื่ออินพุต relay_data คือ เมทริกซ์ของตำแหน่งรีเลย์ในระบบทดสอบใด ๆ ที่สร้างขึ้นมาจากฟังก์ชัน relaydata
parameter_a(OBJ,relay_data,busvoltage)	ฟังก์ชันสำหรับสร้างพารามิเตอร์ a ของเมทริกซ์ฟังก์ชันวัตถุประสงค์ เมื่ออินพุต OBJ คือ เมทริกซ์ของฟังก์ชันวัตถุประสงค์ที่สร้างขึ้นมาจากฟังก์ชัน parameter_OBJ และ busvoltage คือ เมทริกซ์ของค่าแรงดันบัสแต่ละบัสของระบบทดสอบใด ๆ
parameter_c(OBJ,relay_data,busvoltage)	ฟังก์ชันสำหรับสร้างพารามิเตอร์ c ของเมทริกซ์ฟังก์ชันวัตถุประสงค์
loadcurrent_B3LP_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 3 บัส
loadcurrent_bus6naps_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 6 บัส

ชื่อฟังก์ชัน	หน้าที่
loadcurrent_B7flatlp_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 7 บัส
loadcurrent_bus8_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 8 บัส
loadcurrent_bus9_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 9 บัส
loadcurrent_buslp30_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 30 บัส
loadcurrent_P99sp_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 4700 บัส
loadcurrent_LS1A98_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 8166 บัส
loadcurrent_Wisconsin_data ()	ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 14454 บัส
parameter_b_d ()	ฟังก์ชันสำหรับสร้างพารามิเตอร์ b และ d ของฟังก์ชันวัตถุประสงค์
parameter_constraint(relay_data)	ฟังก์ชันสำหรับสร้างพารามิเตอร์ของเงื่อนไขบังคับ
parameter_e_g ()	ฟังก์ชันสำหรับสร้างพารามิเตอร์ e และ g ของเงื่อนไขบังคับ
parameter_f_h(OBJ,Constraint)	ฟังก์ชันสำหรับสร้างพารามิเตอร์ f และ h ของเงื่อนไขบังคับ โดยที่อินพุต Constraint คือค่าพารามิเตอร์ของเงื่อนไขบังคับที่ได้มาจากฟังก์ชัน parameter_e_g
CT_ratio(OBJ, Constraint)	ฟังก์ชันสำหรับการเลือกอัตราส่วนหม้อแปลงกระแสของรีเลย์ในระบบทดสอบใด ๆ เมื่ออินพุต Constraint ได้มาจากฟังก์ชัน parameter_f_h
cut_parameter(OBJ,Constraint)	ฟังก์ชันสำหรับพิจารณาเงื่อนไขและลดตัวแปรในกรณีที่รีเลย์ไม่สามารถทำงานได้
run_LP_3BUS (OBJ,Constraint)	ฟังก์ชันสำหรับรันกำหนดการเชิงเส้นของระบบ 3 บัส
run_LP_6BUS (OBJ,Constraint)	ฟังก์ชันสำหรับรันกำหนดการเชิงเส้นของระบบ 6 บัส
run_LP_7BUS (OBJ,Constraint)	ฟังก์ชันสำหรับรันกำหนดการเชิงเส้นของระบบ 7 บัส
run_LP_8BUS (OBJ,Constraint)	ฟังก์ชันสำหรับรันกำหนดการเชิงเส้นของระบบ 8 บัส
run_LP_9BUS (OBJ,Constraint)	ฟังก์ชันสำหรับรันกำหนดการเชิงเส้นของระบบ 9 บัส
run_LP_30BUS (OBJ,Constraint)	ฟังก์ชันสำหรับรันกำหนดการเชิงเส้นของระบบ 30 บัส
run_LP_4700BUS (OBJ,Constraint)	ฟังก์ชันรันกำหนดการเชิงเส้นของระบบ 4700 บัส
run_LP_8166BUS (OBJ,Constraint)	ฟังก์ชันรันกำหนดการเชิงเส้นของระบบ 8166 บัส
run_LP_14454BUS (OBJ,Constraint)	ฟังก์ชันรันกำหนดการเชิงเส้นของระบบ 14454 บัส

ชื่อฟังก์ชัน**หน้าที่**

run_3bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบทดสอบ 3 บัส

run_6bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบทดสอบ 6 บัส

run_7bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบทดสอบ 7 บัส

run_8bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบทดสอบ 8 บัส

run_9bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบทดสอบ 9 บัส

run_30bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบทดสอบ 30 บัส

run_4700bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบ 4700 บัส

run_8166bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบ 8166 บัส

run_14454bus

ฟังก์ชันสำหรับรันโปรแกรมของระบบ 14454 บัส



ข.3 รายละเอียดของฟังก์ชัน

ในส่วนนี้จะแสดงรายละเอียดของฟังก์ชันจากชื่อของฟังก์ชันในหัวข้อ ข.2 ซึ่งจะนำเสนอเฉพาะระบบทดสอบ 3 บัสเท่านั้น ส่วนระบบอื่น ๆ ฟังก์ชันที่ใช้จะเป็นฟังก์ชันเดียวกัน แตกต่างกันเพียง 3 ฟังก์ชันเท่านั้น เนื่องจากเป็นฟังก์ชันที่เป็นข้อมูลเฉพาะของระบบทดสอบนั้น ๆ ฟังก์ชันที่แตกต่างกัน คือ 1.ฟังก์ชันข้อมูลของระบบทดสอบ 2.ฟังก์ชันข้อมูลโหลดของระบบทดสอบ 3.ฟังก์ชันสำหรับรันโปรแกรมของระบบทดสอบ รายละเอียดของฟังก์ชันมีดังนี้

ข.3.1 ฟังก์ชัน B3LP_pwb.m

```
function B3LP_pwb()
global linedata gendata tfdata busvoltage
% Remark for linedata
% Column 1 : Line number
% Column 2 : From bus
% Column 3 : To bus
% Column 4 : Line resistance (p.u.)           "positive sequence"
% Column 5 : Line reactance (p.u.)           "positive sequence"
% Column 6 : Half-line charging (p.u.)
% Column 7 : Line resistance (p.u.)           "zero sequence"
% Column 8 : Line reactance (p.u.)           "zero sequence"
linedata = [ 1 1 2 0 0.1 0 0 0
             2 1 3 0 0.1 0 0 0
             3 2 3 0 0.1 0 0 0 ];

% Remark for gendata
% Column 1 : Gen number
% Column 2 : Bus number
% Column 3 : Total resistance (p.u.)
% Column 4 : Positive-sequence reactance (p.u.)
% Column 5 : Zero-sequence reactance (p.u.)
% Column 6 : Grounding resistance (p.u.) % inf for ungrounded case
% Column 7 : Grounding reactance (p.u.) % inf for ungrounded case
% Column 8 : Winding connection (Y = 1, D = 0)
gendata = [ 1 1 0 1 0 0 0 1
            2 2 0 1 0 0 0 1
            3 3 0 1 0 0 0 1 ];

% Remark for tfdata (transformer)
% Column 1 : TF number
% Column 2 : From bus
% Column 3 : To bus
% Column 4 : Resistance (p.u.)
% Column 5 : Positive-sequence reactance (p.u.)
% Column 6 : Tap setting (p.u.)
% Column 7 : Zero-sequence reactance (p.u.)
% Column 8 : Connection of primary winding (Y=1,D=0)
% Column 9 : Connection of secondary winding (y=1,d=0)
% Column 10 : Primary-side grounding resistance (p.u.)
% Column 11 : Primary-side grounding reactance (p.u.)
% Column 12 : Secondary-side grounding resistance (p.u.)
% Column 13 : Secondary-side grounding reactance (p.u.)
```

```

    tfdata = [ ];
    % Remark for busvoltage
    % Column 1 : bus number
    % Column 2 : voltage value(kV)
    busvoltage = [ 1  345
                   2  344.999
                   3  345.002 ];

    return
%=====
=====

๖.3.2 ฟังก์ชัน relaydata.m
=====
function [relay_data] = relaydata(linedata,tfdata)
    for i=1:size(linedata,1)
        for j=1:size(tfdata,1)
            if (linedata(i,2)==tfdata(j,2)) || (linedata(i,2)==
                tfdata(j,3)) && (linedata(i,3)==tfdata(j,2)) ||
                (linedata(i,3)==tfdata(j,3)))
                linedata(i,:)=0;
            else end
        end
    end
    % =====
    for i=size(linedata,1):-1:1
        if (linedata(i,:) == 0)
            linedata(i,:)=[ ];
        else end
    end
    for i=1:size(linedata,1)
        linedata(i,1)=i;
    end
    % =====
    x=size(linedata,1);
    rr=2*x;
    ss=rr-size(linedata,1);
    tt=zeros(ss,2);
    linedata=[linedata(:, [2,3]);tt];
    amount_relay=2*x;
    y=amount_relay;
    relay_data = zeros(y,5);
    % =====
    % ===== relay_data column 1 =====
    disp('=====')
    disp(' No. From To close Line ')
    disp('=====')
    for i=1:y
        relay_data(i,1)=i;
        i=i+1;
    end
    % =====

```

```

% ===== relay_data column 2 =====
for i=1:size(linedata,1)
    relay_data(i,2)=linedata(i,1);
end
for i=1:size(linedata,1)
    zz(i)=linedata(i,2);
end
z=y/2+1;
for i=z:y
    relay_data(i,2)=zz(i-y/2);
end
% =====
% ===== relay_data column 3 =====
for i=1:size(linedata,1)
    relay_data(i,3)=linedata(i,2);
end
for i=1:size(linedata,1)
    zzz(i)=linedata(i,1);
end
for i=z:y
    relay_data(i,3)=zzz(i-y/2);
end
% =====
% ===== relay_data column 4 =====
for i=1:size(linedata,1)
    relay_data(i,4)=relay_data(i,2);
end
% =====
% ===== relay_data column 5 =====
for i=1:size(linedata,1)
    relay_data(i,5)=i;
end
k=1;
for i=1:size(linedata,1)
    relay_data(i,5)=k;
    k=k+1;
end
k=1;
for i=z:y
    relay_data(i,5)=k;
    k=k+1;
end
%=====

```

ข.3.3 ฟังก์ชัน parameter_OBJ.m

```

=====
function [OBJ] = parameter_OBJ(relay_data)
    for i=1:size(relay_data,1)
        for j=1
            a(i)=relay_data(i,j);
        end
    end
    % =====

    for i=1:size(relay_data,1)
        for j=2
            b(i)=relay_data(i,j);
        end
    end
    % =====
    for i=1:size(relay_data,1)
        for j=3
            c(i)=relay_data(i,j);
        end
    end
    % =====
    for i=1:size(relay_data,1)
        for j=4
            d(i)=relay_data(i,j);
        end
    end
    % =====
    for i=1:size(relay_data,1)
        for j=5
            e(i)=relay_data(i,j);
        end
    end
    % =====
    k=size(relay_data,1);
    ABCD=zeros(k,6);
    for i=1:k
        for j=1:k
            if (e(i)==e(j)) && (i~=j)
                ABCD(i,:)=[a(i),0,0,a(j),0,0];
            else,end
        end
    end
    disp('=====')
    disp('=====')
    disp('      Pri_cl      Pri_far')
    disp('No.   a       b       No.   c       d ')
    disp('=====')
    disp('=====')
    OBJ=ABCD;
    %=====

```

๖.3.4 ฟังก์ชัน parameter_a.m

```

=====
function [OBJ] = parameter_a(OBJ,relay_data,busvoltage)
global linedata tfdata gendata busvoltage
[ybus,ybus1,ybus0] = mkf_ybus('B3LP_pwb');
OBJ1=OBJ;
for m=1:size(relay_data,1)
    b=relay_data(m,2)
    faultparam=struct('busfault',b,'ftype','TPF',
        'fault_imp',0.0+0.0i,'Vpre_fault',0);
    [If,Vf,Ibr,busfault]=faultcalc(ybus1,linedata,tfdata,
        gendata,faultparam);
    [A]=faultdisp(If,Vf,Ibr,busfault,'B3LP_pwb PowerWorld
        Example');
    aa=zeros(m,2)
    for k=1:size(relay_data,1)
        for l=1:size(A(:, :, b));
            if (k~=m) && (relay_data(k,2)==relay_data(m,2)) &&
                (relay_data(k,3)==A(l,2,b)) && (relay_data(k,2)
                    ==A(l,3,b)) && (A(l,5,b)>=0)
                aa(k,1)= A(l,4,b)
                aa(k,2)= A(l,3,b)
            else
                end
            end
        end
        for k=1:size(aa(:,1))
            for i=1:size(busvoltage(:,1))
                if aa(k,2)==busvoltage(i,1)
                    aa(k,1)= aa(k,1) * (100000000/(sqrt(3) * (busvoltage(i,2) * 1000)))
                    OBJ1(m,2)=sum(aa(:,1));
                else
                    end
                end
            end
        end
        m=m+1;

    end
% =====

```



```

OBJ2=OBJ;
for m=1:size(relay_data,1)
    for n=1:size(gendata,1)
        if (relay_data(m,2)==gendata(n,2))
            b=relay_data(m,2);
            faultparam = struct('busfault',b,'ftype','TPF',
                                'fault_imp',0.0+0.0i,'Vpre_fault',0);
            [If,Vf,Ibr,busfault]=faultcalc(ybus1,linedata,
                                            tfdata,gendata,faultparam);
            [A]=faultdisp(If,Vf,Ibr,busfault,'B3LP_pwb
                          PowerWorld Example');
            for l=1:size(A(:, :, b));
                if (gendata(n,2)==A(l,3,b)) && (A(l,2,b)==0)
                    OBJ2(m,2)= A(l,4,b);
                else,end
            end
            for i=1:size(busvoltage(:,1))
                if b==busvoltage(i,1)
                    OBJ2(m,2)=
OBJ2(m,2)*(100000000/(sqrt(3)*(busvoltage(i,2)*1000)));
                else,end
            end
            m=m+1;
        else end
    end
end
% =====
OBJ3=OBJ;
for m=1:size(relay_data,1)
    for p=1:size(tfdata,1)
        if (relay_data(m,2)==tfdata(p,2)) ||
            (relay_data(m,2)==tfdata(p,3))
            b=relay_data(m,2);
            faultparam = struct('busfault',b,'ftype','TPF',
                                'fault_imp',0.0+0.0i,'Vpre_fault',0);
            [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,tfdata,
                                            gendata,faultparam);
            [A]=faultdisp(If,Vf,Ibr,busfault,'B3LP_pwb PowerWorld
                          Example');
            for l=1:size(A(:, :, b));
                if (relay_data(m,2)==A(l,3,b)) && ((tfdata(p,2)
                    ==A(l,3,b)) || (tfdata(p,3)==A(l,3,b)))
                    OBJ3(m,2)= A(l,4,b);
                else,end
            end
            for i=1:size(busvoltage(:,1))
                if b==busvoltage(i,1)
                    OBJ3(m,2)=
OBJ3(m,2)*(100000000/(sqrt(3)*(busvoltage(i,2)*1000)));
                else,end
            end
            m=m+1;
        else end
    end
end
OBJ(:,2)= OBJ1(:,2)+OBJ2(:,2)+OBJ3(:,2);
Return

```

ข.3.5 ฟังก์ชัน parameter_c.m

```

=====
function [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
global linedata tfdata gendata busvoltage
[ybus,ybus1,ybus0] = mkf_ybus('B3LP_pwb');
for m=1:size(OBJ,1)
    for n=1:size(relay_data,1)
        if (OBJ(m,4))== (relay_data(n,1))
            b=relay_data(n,3);
            faultparam = truct('busfault',b,'ftype','TPF',
                               'fault_imp',0.0+0.0i,'Vpre_fault',0);
            [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                                              tfdata,gendata,faultparam);
            [A]=faultdisp(If,Vf,Ibr,busfault,'B3LP_pwb
                          PowerWorld Example');
            for l=1:size(A(:, :,b));
                if (relay_data(n,2)==A(l,2,b)) &&
                    (relay_data(n,3)==A(l,3,b))
                    OBJ(m,5)= A(l,4,b);
                else,end
            end
            for i=1:size(busvoltage(:,1))
                if b==busvoltage(i,1)
                    OBJ(m,5)=
                        OBJ(m,5) * (1000000000 / (sqrt(3) * (busvoltage(i,2) * 1000)));
                else,end
            end
        else end
    end
end
return
%=====

```

ข.3.6 ฟังก์ชัน loadcurrent_B3LP_data .m

```

=====
function [load_current]= loadcurrent_B3LP_data()
% ==== From_Number To_Number From_Amps To_Amps From_MW To_Mw ====
load_current_old = [ 1 2 55.81 55.81 33.3 -33.3
                     1 3 111.61 111.61 66.7 -66.7
                     2 3 55.81 55.81 33.3 -33.3 ];
load_current = load_current_old ;
for i=1:size(load_current_old ,1)
    if load_current(i,5)<0
        load_current(i,1)= load_current_old (i,2);
        load_current(i,2)= load_current_old (i,1);
        load_current(i,3)= load_current_old (i,4);
        load_current(i,4)= load_current_old (i,3);
        load_current(i,5)= load_current_old (i,6);
        load_current(i,6)= load_current_old (i,5);
    else end
end
return
%=====

```

ข.3.7 ฟังก์ชัน parameter_b_d.m

```

=====
function [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
% ===== parameter_b =====
for i=1:size(relay_data,1)
    for j=1:size(load_current,1)
        if (relay_data(i,2)==load_current(j,1))&&
            (relay_data(i,3)==load_current(j,2))||
            (relay_data(i,2) ==load_current(j,2))&&
            (relay_data(i,3)==load_current(j,1))
            OBJ(i,3)=load_current(j,3);
        else,end
    end
end
for i=1:size(relay_data,1)
    for j=1:size(load_current,1)
        if (relay_data(i,2)==load_current(j,2))&&
            (relay_data(i,3)==load_current(j,1))
            OBJ(i,3)=load_current(j,4);
        else,end
    end
end
end
% =====

```

```

% ===== parameter_d =====
for i=1:size(OBJ,1)
    for j=1:size(OBJ,1)
        if OBJ(i,4)==OBJ(j,1)
            OBJ(i,6)=OBJ(j,3);

            else,end
        end
    end
end
return
%=====

=====

บ.3.8 ฟังก์ชัน parameter_constraint.m

=====
function [Constraint1
Constraint2]=parameter_constraint(relay_data)
% ===== primary relay_close =====
k=1;
for i=1:size(relay_data,1)
    for j=1:size(relay_data,1)
        if (relay_data(i,2)==relay_data(j,3))&&
            (relay_data(i,5)~=relay_data(j,5))
            Constraint1(k,:)= [relay_data(j,1),0,0,relay_data(i,1),0,0];
            k=k+1;
        else,end
    end
end
% =====
% ===== primary relay far =====
Constraint2 = Constraint1;
% =====
disp('=====')
disp('=====')
disp('      Backup      Primary')
disp('No.    e        f      No.    g        h ')
disp('=====')
disp('=====')
%=====

```

ข.3.9 ฟังก์ชัน parameter_e_g.m

```

=====
function[Constraint]=parameter_e_g(relay_data,Constraint1,
    Constraint2,busvoltage,OBJ)
global linedata tfdata gendata busvoltage
[ybus,ybus1,ybus0]=mkf_ybus('B3LP_pwb');
% ===== parameter e of Tbackup relay =====
for m=1:size(Constraint1,1)
    for n=1:size(relay_data,1)
        if Constraint1(m,1)==relay_data(n,1)
            b=relay_data(n,3);
            faultparam = struct('busfault',b,'ftype','TPF',
                'fault_imp',0.0+0.0i,'Vpre_fault',0);
            [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,tfdata,
                gendata,faultparam);
            [A]=faultdisp(If,Vf,Ibr,busfault,'B3LP_pwb PowerWorld
                Example');
            for l=1:size(A(:, :, b), 1)
                if (relay_data(n,2)==A(l,2,b)) && (relay_data(n,3)==
                    A(l,3,b)) && (A(l,5,b)>=0)
                    Constraint1(m,2)=(A(l,4,b));
                elseif (relay_data(n,2)==A(l,2,b)) && (relay_data(n,3)==
                    A(l,3,b)) && (A(l,5,b)<0)
                    Constraint1(m,2)=0 ;
                else, end
            end
            for i=1:size(busvoltage(:,1))
                if (b==relay_data(n,3)) && (Constraint1(m,1)
                    ==relay_data(n,1))
                    AAA=relay_data(n,2);
                    if busvoltage(i,1)==AAA
                        BBB=busvoltage(i,2);
                        Constraint1(m,2)=
                            Constraint1(m,2)*(1000000000/(sqrt(3)*BBB*1000));
                    else,end
                else,end
            end
        else,end
    end
end
% =====
% ===== parameter g of Tprimary relay =====
for m=1:size(Constraint1,1)
    for n=1:size(OBJ,1)
        if Constraint1(m,4)==OBJ(n,1)
            Constraint1(m,5)=OBJ(n,2);
        else end
    end
end
% =====

```

```

% ===== parameter e_2 =====
for m=1:size(Constraint2,1)
    for n=1:size(relay_data,1)
        for l=1:size(relay_data,1)
            if (Constraint2(m,4)==relay_data(n,1)) && (Constraint2(m,1)
                ==relay_data(l,1)) && (relay_data(n,2)==relay_data(l,3))
                b=relay_data(n,3);
                faultparam = struct('busfault',b,'ftype','TPF',
                    'fault_imp',0.0+0.0i,'Vpre_fault',0);
                [If,Vf,Ibr,busfault] = faultcalc(ybusl,linedata,
                    tfdata,gendata,faultparam);
                [A]=faultdisp(If,Vf,Ibr,busfault,'B3LP_pwb PowerWorld
                    Example');
                for k=1:size(A(:, :,b),1)
                    if relay_data(l,2)==A(k,2,b) && (relay_data(l,3)
                        ==A(k,3,b)) && (A(l,5,b)>=0)
                        Constraint2(m,2)=(A(k,4,b));
                    elseif (relay_data(l,2)==A(k,2,b) && (relay_data(l,3)
                        ==A(k,3,b)) && (A(l,5,b)<0)
                        Constraint2(m,2)= 0 ;
                    end
                end
                for i=1:size(busvoltage(:,1))
                    ddd=relay_data(l,1);
                    eee=relay_data(l,2);
                    if eee==busvoltage(i,1)
                        jjj=busvoltage(i,2);
                        Constraint2(m,2)=Constraint2(m,2)*(1000000000/
                            (sqrt(3)*(busvoltage(i,2)*1000)));
                    else,end
                end
            else,end
        end
    end
end
% =====
% ===== parameter g_2 =====
for m=1:size(Constraint2,1)
    for n=1:size(OBJ,1)
        if Constraint2(m,4)==OBJ(n,4)
            Constraint2(m,5)=OBJ(n,5);
        else end
    end
end
Constraint=[Constraint1;Constraint2];
return
% =====

```

ข.3.10 ฟังก์ชัน parameter_f_h.m

```

=====
function [Constraint]=parameter_f_h(OBJ,Constraint)
% ===== parameter_f =====
for i=1:size(Constraint,1)
    for j=1:size(OBJ,1)
        if (Constraint(i,1)==OBJ(j,1))
            Constraint(i,3)=OBJ(j,3);
        else,end
    end
end
%=====
%===== parameter_h =====
for i=1:size(Constraint,1)
    for j=1:size(OBJ,1)
        if (Constraint(i,4)==OBJ(j,1))
            Constraint(i,6)=OBJ(j,3);
        else,end
    end
end
return
%=====
=====

```

ข.3.11 ฟังก์ชัน CT_ratio.m

```

=====
function [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
CT=[5/50    0    50
    5/100   51   100
    5/150   101  150
    5/200   151  200
    5/250   201  250
    5/300   251  300
    5/400   301  400
    5/450   401  450
    5/500   451  500
    5/600   501  600
    5/800   601  800
    5/900   801  900
    5/1000  901  1000
    5/1200 1001 1200 ];
% ===== FOR OBJ_column 3 =====
% =====
for i=1:size(OBJ,1)
    for j=1:size(CT,1)
        if (OBJ(i,3)*1.25)>CT(j,2) || (OBJ(i,3)*1.25)==CT(j,2) &&
            (OBJ(i,3)*1.25)<CT(j,3) || (OBJ(i,3)*1.25)==CT(j,3)
            CTR(i)=CT(j,1);
        elseif OBJ(i,3)>1200
            CTR(i)=CT(14,1);
        end
    end
end
end

```

```

% =====
y=[OBJ(:,1),CTR'];
A = y(:,2);
for i=1:length(y(:,1))
    switch A(i)
        case 5/50
            fprintf('CT ratio of relay number(%d) is 5/50\n',i)
        case 5/100
            fprintf('CT ratio of relay number(%d) is 5/100\n',i)
        case 5/150
            fprintf('CT ratio of relay number(%d) is 5/150\n',i)
        case 5/200
            fprintf('CT ratio of relay number(%d) is 5/200\n',i)
        case 5/250
            fprintf('CT ratio of relay number(%d) is 5/250\n',i)
        case 5/300
            fprintf('CT ratio of relay number(%d) is 5/300\n',i)
        case 5/400
            fprintf('CT ratio of relay number(%d) is 5/400\n',i)
        case 5/450
            fprintf('CT ratio of relay number(%d) is 5/450\n',i)
        case 5/500
            fprintf('CT ratio of relay number(%d) is 5/500\n',i)
        case 5/600
            fprintf('CT ratio of relay number(%d) is 5/600\n',i)
        case 5/800
            fprintf('CT ratio of relay number(%d) is 5/800\n',i)
        case 5/900
            fprintf('CT ratio of relay number(%d) is 5/900\n',i)
        case 5/1000
            fprintf('CT ratio of relay number(%d) is 5/1000\n',i)
        case 5/1200
            fprintf('CT ratio of relay number(%d) is 5/1200\n',i)
    end
end
% =====
OBJ;
for i=1:size(OBJ,1)
    OBJ(i,3)= OBJ(i,3)*y(i,2);
End
% =====
% ===== FOR OBJ_column 6 =====
for i=1:size(OBJ,1)
    for j=1:size(y,1)
        if OBJ(i,4)==y(j,1)
            y(j,2);
            switch y(j,2)
                case 5/50
                    fprintf('CT ratio of relay number(%d) is 5/50\n',OBJ(i,4))
                case 5/100
                    fprintf('CT ratio of relay number(%d) is 5/100\n',OBJ(i,4))
                case 5/150
                    fprintf('CT ratio of relay number(%d) is 5/150\n',OBJ(i,4))
                case 5/200
                    fprintf('CT ratio of relay number(%d) is 5/200\n',OBJ(i,4))
                case 5/250
                    fprintf('CT ratio of relay number(%d) is 5/250\n',OBJ(i,4))
            end
        end
    end
end

```



```

case 5/300
    fprintf('CT ratio of relay number(%d) is 5/300\n',OBJ(i,4))
case 5/400
    fprintf('CT ratio of relay number(%d) is 5/400\n',OBJ(i,4))
case 5/450
    fprintf('CT ratio of relay number(%d) is 5/450\n',OBJ(i,4))
case 5/500
    fprintf('CT ratio of relay number(%d) is 5/500\n',OBJ(i,4))
case 5/600
    fprintf('CT ratio of relay number(%d) is 5/600\n',OBJ(i,4))
case 5/800
    fprintf('CT ratio of relay number(%d) is 5/800\n',OBJ(i,4))
case 5/900
    fprintf('CT ratio of relay number(%d) is 5/900\n',OBJ(i,4))
case 5/1000
    fprintf('CT ratio of relay number(%d) is 5/1000\n',OBJ(i,4))
case 5/1200
    fprintf('CT ratio of relay number(%d) is 5/1200\n',OBJ(i,4))
end
else,end
end
end
% =====
for i=1:size(OBJ,1)
    for j=1:size(y,1)
        if OBJ(i,4)==y(j,1)
            OBJ(i,6)= OBJ(i,6)*y(j,2);
        else,end
    end
end
% =====
% ===== OBJ_column 2(fault current)=====
for i=1:size(OBJ,1)
    for j=1:size(y,1)
        if OBJ(i,1)==y(j,1)
            OBJ(i,2)= OBJ(i,2)*y(j,2);
        else,end
    end
end
% =====
% ===== OBJ_column 5(fault current)=====
for i=1:size(OBJ,1)
    for j=1:size(y,1)
        if OBJ(i,4)==y(j,1)
            OBJ(i,5)= OBJ(i,5)*y(j,2);
        else,end
    end
end
% =====
% ===== FOR Constraint column 2(fault current) =====
for i=1:size(Constraint,1)
    for j=1:size(y,1)
        if Constraint(i,1)==y(j,1)
            Constraint(i,2)= Constraint(i,2)*y(j,2);
        else,end
    end
end
end

```

```

% =====
% ==== FOR Constraint column 3(load current) =====
for i=1:size(Constraint,1)
    for j=1:size(y,1)
        if Constraint(i,1)==y(j,1)
            Constraint(i,3)= Constraint(i,3)*y(j,2);
        else,end
    end
end
% =====
% ==== FOR Constraint column 5(fault current)=====
for i=1:size(Constraint,1)
    for j=1:size(y,1)
        if Constraint(i,4)==y(j,1)
            Constraint(i,5)= Constraint(i,5)*y(j,2);
        else,end
    end
end
% =====
% ===== FOR Constraint_column 6 (load current) =====
for i=1:size(Constraint,1)
    for j=1:size(y,1)
        if Constraint(i,4)==y(j,1)
            Constraint(i,6)= Constraint(i,6)*y(j,2);
        else,end
    end
end
return
%=====
=====

ข.3.12 ฟังก์ชัน cut_parameter.m
=====
function [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    for i= size(Constraint,1):-1:1
        if (Constraint(i,2)< 1.25*Constraint(i,3)) || (Constraint(i,5)<
            1.25*Constraint(i,6)) || ((Constraint(i,2)==0) &&
            (Constraint(i,3)==0)) || ((Constraint(i,5)==0) &&
            (Constraint(i,6)==0)) || (Constraint(i,3)==0) ||
            (Constraint(i,6)==0) || (Constraint(i,2)==0) ||
            (Constraint(i,5)==0)
            Constraint(i,:)=[];
        else end
    end
end
%=====

```

ข.3.13 ฟังก์ชัน run_LP_3BUS.m

```

=====
Function [x,fval]=run_LP_3BUS(OBJ,Constraint)
data_OBJ=OBJ;
data_A=Constraint;
n=size(data_OBJ,1);
q=size(data_A,1);
disp('=====')
for p=1:n
    PS(p)=1.25;
end
for l=1:n
    for k=1:n
        a(k)= data_OBJ(k,2);
        b(k)= data_OBJ(k,3);
        PS_cl(k)= PS(1,k);
        if ((a(k)==0)&&(b(k)==0)) || (b(k)==0) || (a(k)==0)
            t_relay_close(k)=0;
        else
            t_relay_close(k)=(0.14/(((a(k)/(PS_cl(k)*b(k)))^0.02)- 1));
            if (t_relay_close(k)<0)
                t_relay_close(k)=0;
            else end
        end
    end
    t_relay_close(l)= t_relay_close(k);
end
%=====
for k=1:n
    m(k)=data_OBJ(k,4);
end
for l=1:n+1
    for k=1:n
        c(k)= data_OBJ(k,5);
        d(k)= data_OBJ(k,6);
        PS_far(k)= PS(1,m(k));
        if ((c(k)==0)&&(d(k)==0)) || (d(k)==0) || (c(k)==0)
            t_relay_far(m(k))=0;
        else
            t_relay_far(m(k))=(0.14/(((c(k)/(PS_far(k)*d(k)))^0.02)- 1));
            if (t_relay_far(m(k))<0)
                t_relay_far(m(k))=0;
            else end
        end
    end
    t_relay_far(l) = t_relay_far(m(k));
end
%=====
for l=1:n
    OBJ(l)= t_relay_close(l)+ t_relay_far(l);
    F(l)=OBJ(l);
end
%=====

```

```

A=zeros(n,n);
for k=1:q
    x(k)= data_A(k,4);
    y(k)= data_A(k,1);
end
for k=1:q
    g(k)= data_A(k,5);
    h(k)= data_A(k,6);
    PS_primary(k)= PS(1,x(k));
    t_primary_relay_number(k) =
        (0.14/(((g(k)/(PS_primary(k)*h(k)))^0.02)- 1));
    A(k,x(k))= t_primary_relay_number(k);
end
for k=1:q
    e(k)= data_A(k,2);
    f(k)= data_A(k,3);
    PS_backup(k)= PS(1,y(k));
    t_back_up_relay_number(k) =
        (0.14/(((e(k)/(PS_backup(k)*f(k)))^0.02)- 1));
    A(k,y(k))= -t_back_up_relay_number(k);
end
% =====
b=zeros(1,n);
for p=1:q
    b(p)= -0.3;
    p=p+1;
end
% =====
for p=1:n
    lb(p)= 0.05;
    p=p+1;
end
% =====
for p=1:n
    ub(p) = 1.1;
    p=p+1;
end
% =====
A_t_lb=zeros(n,n);
A_t_ub=zeros(n,n);
b_t_lb=zeros(n,1);
b_t_ub=zeros(n,1);
t_lb= 0.05; % lower operate time of primary relay = 0.05 second
t_ub= 1; % upper operate time of primary relay = 1 second
for i=1:n
    F(1,i)=F(i) ;
    A_t_lb(i,i)= -F(i);
    A_t_ub(i,i)= F(i);
    if F(i)==0
        b_t_lb(i,1)= t_lb;
    else
        b_t_lb(i,1)= -t_lb;
    end
    if F(i)==0
        b_t_ub(i,1)= t_ub;
    else
        b_t_ub(i,1)= t_ub;
    end
end

```

```

        end
    end
    A=[A;A_t_lb;A_t_ub];
    b=[b';b_t_lb;b_t_ub];
    [x,fval,exitflag,output,lambda] = linprog(F,A,b,[],[],lb,ub)
    return
%=====
=====

ข.3.14 ฟังก์ชัน run_3bus.m
=====

clear all;clc
global linedata tfdata busvoltage
format short g
B3LP_pwb();
[relay_data]=relaydata(linedata,tfdata);
[OBJ]=parameter_OBJ(relay_data);
[OBJ]=parameter_a(OBJ,relay_data,busvoltage);
[OBJ]=parameter_c(OBJ,relay_data,busvoltage);
[load_current]=loadcurrent_B3LP_data();
[OBJ]=parameter_b_d(OBJ,load_current,relay_data);
[Constraint1 Constraint2]=parameter_constraint(relay_data);
[Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
    busvoltage,OBJ);
[Constraint]=parameter_f_h(OBJ,Constraint);
[OBJ,Constraint]=CT_ratio(OBJ,Constraint);
[OBJ,Constraint]=cut_parameter(OBJ,Constraint);
[x,fval]=run_LP(OBJ,Constraint);
%=====

```

ข.4 รายละเอียดของโปรแกรมส่วนต่อประสานกราฟิก (GUI)

ข.4.1 ฟังก์ชันเริ่มต้นของ GUI

```

=====
                                ข.4.1 ฟังก์ชันเริ่มต้นของ GUI
=====
function varargout = TEST(varargin)
    gui_Singleton = 1;
    gui_State = struct('gui_Name',mfilename,...
                        'gui_Singleton',gui_Singleton,...
                        'gui_OpeningFcn',@TEST_OpeningFcn,...
                        'gui_OutputFcn',@TEST_OutputFcn,...
                        'gui_LayoutFcn',[],...
                        'gui_Callback',[]);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end
    if nargin
        [varargout{1:nargout}] = gui_mainfcn(gui_State,
        varargin{:});
    else
        gui_mainfcn(gui_State,varargin{:});
    end
    % End initialization code - DO NOT EDIT
    % =====
    % --- Executes just before TEST is made visible.
    function TEST_OpeningFcn(hObject,eventdata,handles,varargin)
    % This function has no output args, see OutputFcn.
    % hObject handle to figure
    % eventdata reserved - to be defined in a future version of
    MATLAB
    % handles structure with handles and user data (see GUIDATA)
    % varargin command line arguments to TEST (see VARARGIN)
    % =====
    % Choose default command line output for TEST
    handles.output = hObject; %Update handles structure
    guidata(hObject,handles);
    % =====
    % UIWAIT makes TEST wait for user response (see UIRESUME)
    % uiwait(handles.figure1);
    % =====
    % --- Outputs from this function are returned to the command
    line.
    function varargout = TEST_OutputFcn(hObject,eventdata,handles)
    % varargout cell array for returning output args (see
    VARARGOUT);
    % hObject handle to figure
    % eventdata reserved - to be defined in a future version of
    MATLAB
    % handles structure with handles and user data (see GUIDATA)
    % =====
    % Get default command line output from handles structure
    varargout{1} = handles.output;
    % =====
    function Turn_Off(off)
    set(off,'Value',0);
    %=====

```

ข.4.2 ฟังก์ชันของวัตถุ radiobutton1

```

=====
% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject,eventdata,handles)
% hObject handle to radiobutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of
radiobutton1
global linedata gendata tfdata busvoltage system bus_all
off = [handles.radiobutton2 handles.radiobutton3
handles.radiobutton4 handles.radiobutton5 handles.radiobutton6
handles.radiobutton7 handles.radiobutton8 handles.radiobutton9
];
Turn_Off(off);
set(handles.radiobutton1,'Value',1);axes(handles.axes1)
% Switches focus to axes object.
imshow('3BUS.png') % Display it with some command such as
bus_all=3;
B3LP_pwb()
system='B3LP';
a=[];
set(handles.edit1,'string',a)
set(handles.edit2,'string',a)
set(handles.uitable1,'data',a)
set(handles.uitable2,'data',a)
% =====

```

ข.4.3 ฟังก์ชันของวัตถุ radiobutton2

```

=====
% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject,eventdata,handles)
% hObject handle to radiobutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton2
global linedata gendata tfdata busvoltage system bus_all
off = [handles.radiobutton1 handles.radiobutton3
handles.radiobutton4 handles.radiobutton5 handles.radiobutton6
handles.radiobutton7 handles.radiobutton8 handles.radiobutton9 ];
Turn_Off(off);
set(handles.radiobutton2,'Value',1);
axes(handles.axes1) % Switches focus to this axes object.
imshow('6BUS.png') % Display it with some command such as
bus_all=6;
bus6naps_pwb()
system='bus6naps';
a=[] ; set(handles.edit1,'string',a)
set(handles.edit2,'string',a)set(handles.uitable1,'data',a)
set(handles.uitable2,'data',a)
% =====

```

ข.4.4 ฟังก์ชันของวัตถุ radiobutton3

```

=====
% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject,eventdata,handles)
% hObject handle to radiobutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of
  MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton3
global linedata gendata tfdata busvoltage system bus_all
off = [handles.radiobutton2 handles.radiobutton1
handles.radiobutton4 handles.radiobutton5 handles.radiobutton6
handles.radiobutton7 handles.radiobutton8 handles.radiobutton9 ];
Turn_Off(off);
set(handles.radiobutton3,'Value',1);
axes(handles.axes1) % Switches focus to this axes object.
imshow('7BUS.png') % Display it with some command such as
bus_all=7;
B7flatlp_pwb()
system='B7flatlp';
a=[];
set(handles.edit1,'string',a)
set(handles.edit2,'string',a)
set(handles.uitable1,'data',a)
set(handles.uitable2,'data',a)
% =====

```

ข.4.5 ฟังก์ชันของวัตถุ radiobutton4

```

=====
% --- Executes on button press in radiobutton4.
function radiobutton4_Callback(hObject,eventdata,handles)
% eventdata reserved - to be defined in a future version of
  MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton4
global linedata gendata tfdata busvoltage system bus_all
off = [handles.radiobutton2 handles.radiobutton3
handles.radiobutton1 handles.radiobutton5 handles.radiobutton6
handles.radiobutton7 handles.radiobutton8 handles.radiobutton9 ];
Turn_Off(off);
set(handles.radiobutton4,'Value',1);
axes(handles.axes1) % Switches focus to this axes object.
imshow('8BUS.png') % Display it with some command such as
bus_all=8;
bus8_pwb()
system='bus8';
a=[];
set(handles.edit1,'string',a)
set(handles.edit2,'string',a)
set(handles.uitable1,'data',a)
set(handles.uitable2,'data',a)
% =====

```


ข.4.6 ฟังก์ชันของวัตถุ radiobutton5

```

=====
% --- Executes on button press in radiobutton5.
function radiobutton5_Callback(hObject,eventdata,handles)
    % hObject handle to radiobutton5 (see GCBO)
    % eventdata reserved - to be defined in a future version of
    % MATLAB
    % handles structure with handles and user data (see GUIDATA)
    % Hint: get(hObject,'Value') returns toggle state of radiobutton5
    global linedata gendata tfdata busvoltage system bus_all
    off = [handles.radiobutton2 handles.radiobutton3
handles.radiobutton4 handles.radiobutton1 handles.radiobutton6
handles.radiobutton7 handles.radiobutton8 handles.radiobutton9 ];
    Turn_Off(off);
    set(handles.radiobutton5,'Value',1);
    axes(handles.axes1) % Switches focus to this axes object.
    imshow('9BUS.png') % Display it with some command such as
    bus_all=9;
    bus9_pwb()
    system='bus9';
    a=[];
    set(handles.edit1,'string',a)
    set(handles.edit2,'string',a)
    set(handles.uitable1,'data',a)
    set(handles.uitable2,'data',a)

```

```

% =====
=====

```

ข.4.7 ฟังก์ชันของวัตถุ radiobutton6

```

=====
% --- Executes on button press in radiobutton6.
function radiobutton6_Callback(hObject,eventdata,handles)
    % hObject handle to radiobutton6 (see GCBO)
    % eventdata reserved - to be defined in a future version of
    % MATLAB
    % handles structure with handles and user data (see GUIDATA)
    % Hint: get(hObject,'Value') returns toggle state of radiobutton6
    global linedata gendata tfdata busvoltage system bus_all
    off = [handles.radiobutton2 handles.radiobutton3
handles.radiobutton4 handles.radiobutton5 handles.radiobutton1
handles.radiobutton7 handles.radiobutton8 handles.radiobutton9 ];
    Turn_Off(off);
    set(handles.radiobutton6,'Value',1);
    axes(handles.axes1) % Switches focus to this axes object.
    imshow('30BUS.png') % Display it with some command such as
    bus_all=30;
    buslp30_pwb()
    system='buslp30';
    a=[];
    set(handles.edit1,'string',a)
    set(handles.edit2,'string',a)
    set(handles.uitable1,'data',a)
    set(handles.uitable2,'data',a)

```

ข.4.8 ฟังก์ชันของวัตถุ radiobutton7

```

=====
% --- Executes on button press in radiobutton7.
function radiobutton7_Callback(hObject,eventdata,handles)
% hObject handle to radiobutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of
  MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton7
global linedata gendata tfdata busvoltage system bus_all
off = [handles.radiobutton2 handles.radiobutton3
handles.radiobutton4 handles.radiobutton5 handles.radiobutton6
handles.radiobutton1 handles.radiobutton8 handles.radiobutton9 ];
Turn_Off(off);
set(handles.radiobutton7,'Value',1);
axes(handles.axes1) % Switches focus to this axes object.
imshow('4700BUS.png') % Display it with some command such as
bus_all=4700;
P99sp_pwb()
system='P99sp';
a=[];
set(handles.edit1,'string',a)
set(handles.edit2,'string',a)
set(handles.uitable1,'data',a)
set(handles.uitable2,'data',a)
% =====

```

ข.4.9 ฟังก์ชันของวัตถุ radiobutton8

```

=====
% --- Executes on button press in radiobutton8.
function radiobutton8_Callback(hObject,eventdata,handles)
% hObject handle to radiobutton8 (see GCBO)
% eventdata reserved - to be defined in a future version of
  MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton8
global linedata gendata tfdata busvoltage system bus_all
off = [handles.radiobutton2 handles.radiobutton3
handles.radiobutton4 handles.radiobutton5 handles.radiobutton6
handles.radiobutton7 handles.radiobutton1 handles.radiobutton9 ];
Turn_Off(off);
set(handles.radiobutton8,'Value',1);
axes(handles.axes1) % Switches focus to this axes object.
imshow('8166BUS.png') % Display it with some command such as
bus_all=8166;
LS1A98_pwb()
system='LS1A98';
a=[];
set(handles.edit1,'string',a)
set(handles.edit2,'string',a)
set(handles.uitable1,'data',a)
set(handles.uitable2,'data',a)
% =====

```

ข.4.10 ฟังก์ชันของวัตถุ radiobutton9

```

=====
% --- Executes on button press in radiobutton9.
function radiobutton9_Callback(hObject,eventdata,handles)
    % hObject handle to radiobutton9 (see GCBO)
    % eventdata reserved - to be defined in a future version of
    % MATLAB
    % handles structure with handles and user data (see GUIDATA)
    % Hint: get(hObject,'Value') returns toggle state of radiobutton9
    global linedata gendata tfdata busvoltage system bus_all
    off = [handles.radiobutton2 handles.radiobutton3
handles.radiobutton4 handles.radiobutton5 handles.radiobutton6
handles.radiobutton7 handles.radiobutton8 handles.radiobutton1 ];
    Turn_Off(off);
    set(handles.radiobutton9,'Value',1);
    axes(handles.axes1) % Switches focus to this axes object.
    imshow('14454BUS.png') % Display it with some command such as
    bus_all=14454;
    Wisconsin_pwb()
    system='Wisconsin';
    a=[];
    set(handles.edit1,'string',a)
    set(handles.edit2,'string',a)
    set(handles.uitable1,'data',a)
    set(handles.uitable2,'data',a)
%=====
=====

```

ข.4.11 ฟังก์ชันของวัตถุ edit1

```

=====
function edit1_Callback(hObject,eventdata,handles)
    % hObject handle to edit1 (see GCBO)
    % eventdata reserved - to be defined in a future version
    % handles structure with handles and user data (see GUIDATA)
    % Hints: get(hObject,'String') returns contents of edit1 as text
    % str2double(get(hObject,'String')) returns contents of edit1 as
    % a double
    global nf
    position=get(handles.edit1,'String');
    nf=str2num(position);
    % =====
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject,eventdata,handles)
    % hObject handle to edit1 (see GCBO)
    % eventdata reserved - to be defined in a future version
    % handles empty - handles not created until after all CreateFcns
    % called
    % Hint: edit controls usually have a white background on Windows.
    % See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

```

ข.4.12 ฟังก์ชันของวัตถุ pushbutton1

```

=====
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject,eventdata,handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of
  MATLAB
% handles structure with handles and user data (see GUIDATA)
global linedata tfdata gendata system nf relay_data A QQ
    data_current ybus ybus1 ybus0 If Vf Ibr busfault bus_all
format short g
switch system
    case 'B3LP'
        if nf <= 0 || nf > bus_all
            errordlg('Out of bus. Fault bus number must be
                between 1 - 3')
        else
            [relay_data]=relaydata(linedata,tfdata);
            [ybus,ybus1,ybus0]=mkf_ybus('B3LP_pwb');
            b=nf;
            faultparam = struct('busfault',b,'ftype','TPF',
                'fault_imp',0.0+0.0i,'Vpre_fault',0);
            [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                tfdata,gendata,faultparam);
            [A]=faultdisp(If,Vf,Ibr,busfault,'B3LP_pwb PowerWorld
                Example');
            data_current = [A(:, :,nf)];
            QQ = pairs_relay2(data_current,relay_data,nf);
            mydata = QQ(:, [2,3]) ;
            set(handles.uitable1,'data',mydata)
            columnHeader={'Primary','Secondary'};
            set(handles.uitable1,'ColumnName',columnHeader)
        end
    case 'bus6naps'
        if nf <= 0 || nf > bus_all
            errordlg('Out of bus. Fault bus number must be
                between 1 - 6')
        else
            [relay_data] = relaydata(linedata,tfdata);
            [ybus,ybus1,ybus0] = mkf_ybus('bus6naps_pwb');
            b=nf;
            faultparam = struct('busfault',b,'ftype','TPF',
                'fault_imp',0.0+0.0i,'Vpre_fault',0);
            [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                tfdata,gendata,faultparam);
            [A]=faultdisp(If,Vf,Ibr,busfault,'bus6naps_pwb PowerWorld
                Example');
            data_current = [A(:, :,nf)];
            QQ = pairs_relay2(data_current,relay_data,nf);
            mydata = QQ(:, [2,3]) ;
            set(handles.uitable1,'data',mydata)
            columnHeader={'Primary','Secondary'};
            set(handles.uitable1,'ColumnName',columnHeader)
        end
end

```

```

case 'B7flatlp'
    if nf <= 0 || nf > bus_all
        errordlg('Out of bus. Fault bus number must be
                between 1 - 7')
    else
        [relay_data] = relaydata(linedata,tfddata);
        [ybus,ybus1,ybus0] = mkf_ybus('B7flatlp_pwb');
        b=nf;
        faultparam = struct('busfault',b,'ftype','TPF',
                            'fault_imp',0.0+0.0i,'Vpre_fault',0);
        [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                                         tfdata,gendata,faultparam);
        [A]=faultdisp(If,Vf,Ibr,busfault,'B7flatlp_pwb PowerWorld
                        Example');
        data_current = [A(:, :,nf)];
        QQ = pairs_relay2(data_current,relay_data,nf);
        mydata = QQ(:, [2,3]) ;
        set(handles.uitable1,'data',mydata)
        columnHeader={'Primary','Secondary'};
        set(handles.uitable1,'ColumnName',columnHeaders)
    end
case 'bus8'
    if nf <= 0 || nf > bus_all
        errordlg('Out of bus. Fault bus number must be
                between 1 - 8')
    else
        [relay_data] = relaydata(linedata,tfddata);
        [ybus,ybus1,ybus0] = mkf_ybus('bus8_pwb');
        b=nf;
        faultparam = struct('busfault',b,'ftype','TPF',
                            'fault_imp',0.0+0.0i,'Vpre_fault',0);
        [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                                         tfdata,gendata,faultparam);
        [A]=faultdisp(If,Vf,Ibr,busfault,'bus8_pwb PowerWorld
                        Example');
        data_current = [A(:, :,nf)];
        QQ = pairs_relay2(data_current,relay_data,nf);
        mydata = QQ(:, [2,3]) ;
        set(handles.uitable1,'data',mydata)
        columnHeader={'Primary','Secondary'};
        set(handles.uitable1,'ColumnName',columnHeaders)
    end
case 'bus9'
    if nf <= 0 || nf > bus_all
        errordlg('Out of bus. Fault bus number must be
                between 1 - 9')
    else
        [relay_data] = relaydata(linedata,tfddata);
        [ybus,ybus1,ybus0] = mkf_ybus('bus9_pwb');
        b=nf;
        faultparam = struct('busfault',b,'ftype','TPF',
                            'fault_imp',0.0+0.0i,'Vpre_fault',0);
        [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                                         tfdata,gendata,faultparam);
        [A]=faultdisp(If,Vf,Ibr,busfault,'bus9_pwb PowerWorld
                        Example');
        data_current = [A(:, :,nf)];

```

```

QQ = pairs_relay2(data_current,relay_data,nf);
mydata = QQ(:,[2,3]) ;
set(handles.uitable1,'data',mydata)
columnHeaders={'Primary','Secondary'};
set(handles.uitable1,'ColumnName',columnHeaders)
end
case 'buslp30'
if nf <= 0 || nf > bus_all
    errordlg('Out of bus. Fault bus number must be
        between 1 - 30')
else
[relay_data] = relaydata(linedata,tfddata);
[ybus,ybus1,ybus0] = mkf_ybus('buslp30_pwb');
b=nf;
faultparam = struct('busfault',b,'ftype','TPF',
    'fault_imp',0.0+0.0i,'Vpre_fault',0);
[If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
    tfddata,gendata,faultparam);
[A]=faultdisp(If,Vf,Ibr,busfault,'buslp30_pwb PowerWorld
    Example');
data_current = [A(:, :,nf)];
QQ = pairs_relay2(data_current,relay_data,nf);
mydata = QQ(:,[2,3]) ;
set(handles.uitable1,'data',mydata)
columnHeaders={'Primary','Secondary'};
set(handles.uitable1,'ColumnName',columnHeaders)
end
case 'P99sp'
if nf <= 0 || nf > bus_all
    errordlg('Out of bus. Fault bus number must be
        between 1 - 4700')
else
[relay_data] = relaydata(linedata,tfddata);
[ybus,ybus1,ybus0] = mkf_ybus('P99sp_pwb');
b=nf;
faultparam = struct('busfault',b,'ftype','TPF',
    'fault_imp',0.0+0.0i,'Vpre_fault',0);
[If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
    tfddata,gendata,faultparam);
[A]=faultdisp(If,Vf,Ibr,busfault,'P99sp_pwb PowerWorld
    Example');
data_current = [A(:, :,nf)];
QQ = pairs_relay2(data_current,relay_data,nf);
mydata = QQ(:,[2,3]) ;
set(handles.uitable1,'data',mydata)
columnHeaders={'Primary','Secondary'};
set(handles.uitable1,'ColumnName',columnHeaders)
end
case 'LS1A98'
if nf <= 0 || nf > bus_all
    errordlg('Out of bus. Fault bus number must be
        between 1 - 8166')
else
[relay_data] = relaydata(linedata,tfddata);
[ybus,ybus1,ybus0] = mkf_ybus('LS1A98_pwb');
b=nf;

```

```

        faultparam = struct('busfault',b,'ftype','TPF',
                            'fault_imp',0.0+0.0i,'Vpre_fault',0);
        [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                                          tfdata,gendata,faultparam);
        [A]=faultdisp(If,Vf,Ibr,busfault,'LS1A98_pwb PowerWorld
                        Example');
        data_current = [A(:, :,nf)];
        QQ = pairs_relay2(data_current,relay_data,nf);
        mydata = QQ(:, [2,3]) ;
        set(handles.uitable1,'data',mydata)
        columnHeaders={'Primary','Secondary'};
        set(handles.uitable1,'ColumnName',columnHeaders)
    end
case 'Wisconsin'
    if nf <= 0 || nf > bus_all
        errordlg('Out of bus. Fault bus number must be
                  between 1 - 14454')
    else
        [relay_data] = relaydata(linedata,tfdata);
        [ybus,ybus1,ybus0] = mkf_ybus('Wisconsin_pwb');
        b=nf;
        faultparam = struct('busfault',b,'ftype','TPF',
                            'fault_imp',0.0+0.0i,'Vpre_fault',0);
        [If,Vf,Ibr,busfault] = faultcalc(ybus1,linedata,
                                          tfdata,gendata,faultparam);
        [A]=faultdisp(If,Vf,Ibr,busfault,'Wisconsin_pwb
                        PowerWorld Example');
        data_current = [A(:, :,nf)];
        QQ = pairs_relay2(data_current,relay_data,nf);
        mydata = QQ(:, [2,3]) ;
        set(handles.uitable1,'data',mydata)
        columnHeaders={'Primary','Secondary'};
        set(handles.uitable1,'ColumnName',columnHeaders)
    end
end
% =====
% =====

```

ข.4.13 ฟังก์ชันของวัตถุ pushbutton2

```

=====
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject,eventdata,handles)
    % hObject handle to pushbutton2 (see GCBO)
    % eventdata reserved - to be defined in a future version of
    % MATLAB
    % handles structure with handles and user data (see GUIDATA)
    format short g
    global linedata gendata tfdata busvoltage system
    system
    switch system
    case 'B3LP'
        [relay_data] = relaydata(linedata,tfdata);
        [OBJ]=parameter_OBJ(relay_data);
        [OBJ]=parameter_a(OBJ,relay_data,busvoltage);
        [OBJ]=parameter_c(OBJ,relay_data,busvoltage);
        [load_current]= loadcurrent_B3LP_data();
        [OBJ]=parameter_b_d(OBJ,load_current,relay_data);
        [Constraint1 Constraint2 ]= parameter_constraint(relay_data);
    end
end

```

```

[Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
    busvoltage,OBJ)
[Constraint]= parameter_f_h(OBJ,Constraint)
[OBJ,Constraint]=CT_ratio(OBJ,Constraint)
[OBJ,Constraint]= cut_parameter(OBJ,Constraint)
[x,fval]=run_LP_3BUS(OBJ,Constraint)
TDS=(x);
mydata = TDS(:,1) ;
set(handles.uitable2,'data',mydata)
columnHeaders={'TDS'};
set(handles.uitable2,'ColumnName',columnHeaders)
objval=(fval);
set(handles.edit2,'string',objval)
% =====
case 'bus6naps'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_bus6naps_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
        busvoltage,OBJ)
    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
    [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    [x,fval]=run_LP_6BUS(OBJ,Constraint)
    TDS=(x);
    mydata = TDS(:,1) ;
    set(handles.uitable2,'data',mydata)
    columnHeaders={'TDS'};
    set(handles.uitable2,'ColumnName',columnHeaders)
    objval=(fval);
    set(handles.edit2,'string',objval)
% =====
case 'B7flatlp'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_B7flatlp_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
        busvoltage,OBJ)
    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
    [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    [x,fval]=run_LP_7BUS(OBJ,Constraint)
    TDS=(x);
    mydata = TDS(:,1) ;
    set(handles.uitable2,'data',mydata)
    columnHeaders={'TDS'};
    set(handles.uitable2,'ColumnName',columnHeaders)
    objval=(fval);
    set(handles.edit2,'string',objval)

```



```

% =====
case 'bus8'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_bus8_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
                               busvoltage,OBJ)
    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
    [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    [x,fval]=run_LP_8BUS(OBJ,Constraint)
    TDS=(x);
    mydata = TDS(:,1) ;
    set(handles.uitable2,'data',mydata)
    columnHeader={'TDS'};
    set(handles.uitable2,'ColumnName',columnHeader)
    objval=(fval);
    set(handles.edit2,'string',objval)
% =====
case 'bus9'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_bus9_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
                               busvoltage,OBJ)
    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
    [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    [x,fval]=run_LP_9BUS(OBJ,Constraint)
    TDS=(x);
    mydata = TDS(:,1) ;
    set(handles.uitable2,'data',mydata)
    columnHeader={'TDS'};
    set(handles.uitable2,'ColumnName',columnHeader)
    objval=(fval);
    set(handles.edit2,'string',objval)
% =====
case 'buslp30'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_buslp30_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
                               busvoltage,OBJ)
    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)

```

```

[OBJ,Constraint]= cut_parameter(OBJ,Constraint)
[x,fval]=run_LP_30BUS(OBJ,Constraint)
TDS=(x);
mydata = TDS(:,1) ;
set(handles.uitable2,'data',mydata)
columnHeaders={'TDS'};
set(handles.uitable2,'ColumnName',columnHeaders)
objval=(fval);
set(handles.edit2,'string',objval)
% =====
case 'P99sp'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_ P99sp_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
        busvoltage,OBJ)
    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
    [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    [x,fval]=run_LP_4700BUS(OBJ,Constraint)
    TDS=(x);
    mydata = TDS(:,1) ;
    set(handles.uitable2,'data',mydata)
    columnHeaders={'TDS'};
    set(handles.uitable2,'ColumnName',columnHeaders)
    objval=(fval);
    set(handles.edit2,'string',objval)
% =====
case 'LS1A98'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_ LS1A98_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
        busvoltage,OBJ)
    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
    [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    [x,fval]=run_LP_8166BUS(OBJ,Constraint)
    TDS=(x);
    mydata = TDS(:,1) ;
    set(handles.uitable2,'data',mydata)
    columnHeaders={'TDS'};
    set(handles.uitable2,'ColumnName',columnHeaders)
    objval=(fval);
    set(handles.edit2,'string',objval)

```

```

% =====
case 'Wisconsin'
    [relay_data] = relaydata(linedata,tfddata)
    [OBJ]=parameter_OBJ(relay_data)
    [OBJ]=parameter_a(OBJ,relay_data,busvoltage)
    [OBJ]=parameter_c(OBJ,relay_data,busvoltage)
    [load_current]= loadcurrent_ Wisconsin_data()
    [OBJ]=parameter_b_d(OBJ,load_current,relay_data)
    [Constraint1 Constraint2 ]= parameter_constraint(relay_data)
    [Constraint]=parameter_e_g(relay_data,Constraint1,Constraint2,
                               busvoltage,OBJ)

    [Constraint]= parameter_f_h(OBJ,Constraint)
    [OBJ,Constraint]=CT_ratio(OBJ,Constraint)
    [OBJ,Constraint]= cut_parameter(OBJ,Constraint)
    [x,fval]=run_LP_14454BUS(OBJ,Constraint)
    TDS=(x);
    mydata = TDS(:,1) ;
    set(handles.uitable2,'data',mydata)
    columnHeader={'TDS'};
    set(handles.uitable2,'ColumnName',columnHeader)
    objval=(fval);
    set(handles.edit2,'string',objval)
end
% =====
=====

ข.4.14 ฟังก์ชันของวัตถุ edit2

=====
function edit2_Callback(hObject,eventdata,handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit2 as text
% str2double(get(hObject,'String')) returns contents of edit2 as
a double
% =====
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject,eventdata,handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all CreateFcns
called
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%=====

```

ภาคผนวก ค

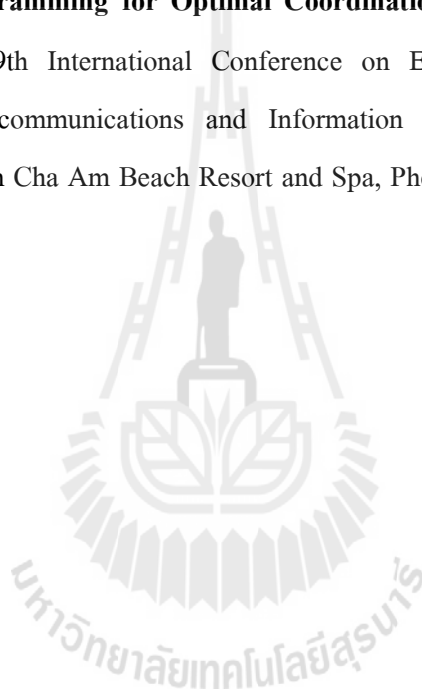
บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

ศุภวัชร นิยมพันธุ์ และ ธนัชชัย กุลวรวานิชพงษ์. (2554). การจัดลำดับความสัมพันธ์ที่เหมาะสมที่สุดของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางโดยใช้กำหนดการเชิงเส้น. การประชุมวิชาการเครือข่ายพลังงานแห่งประเทศไทยครั้งที่ 7 (E-NETT 7) ณ โรงแรมภูเก็ต ออร์คิด รีสอร์ท แอนด์สปา จังหวัดภูเก็ต วันที่ 3 - 5 พฤษภาคม 2554 : หน้า 329-332.

Niyomphant, S., Leeton, U., Chomnawang, N., and Kulworawanichpong, T., (2012). **Application of Linear Programming for Optimal Coordination of Directional Over-Current Relays**, IEEE 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2012). Novotel Hua Hin Cha Am Beach Resort and Spa, Phetchaburi, Thailand, pp. 1- 4, May 16 – 18, 2012.



การจัดลำดับความสัมพันธ์ที่เหมาะสมที่สุดของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง โดยใช้กำหนดการเชิงเส้น

Optimal Coordination of Directional Over-Current Relays Using Linear Programming

ศุภวัชร นิยมพันธุ์, ธนัตชัย กุลวรรณพงษ์ และ นิเมต ชมนาวัง
สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
ถนนมหาวิทยาลัย ด.สุรนารี อ.เมือง จ.นครราชสีมา 30000
E-mail: Supawatchara_@hotmail.com, Thanatchai@gmail.com, sut.mems@gmail.com

บทคัดย่อ

รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง เป็นอุปกรณ์ที่สำคัญในการป้องกันระบบไฟฟ้ากำลัง ทำหน้าที่ในการป้องกันความผิดปกติที่เกิดขึ้นในระบบไฟฟ้ากำลัง เช่น โหลดเกิน แรงดันเกิน การลัดวงจร เป็นต้น ซึ่งการป้องกันความผิดปกติเหล่านี้ ต้องอาศัยการจัดลำดับความสัมพันธ์ที่เหมาะสมของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางอย่างเหมาะสมและมีประสิทธิภาพ เพื่อป้องกันความเสียหายที่อาจเกิดขึ้นกับอุปกรณ์ไฟฟ้าและเพื่อให้เกิดความเชื่อถือได้ในระบบ โดยรีเลย์แต่ละตัวจะมีการปรับตั้งค่าจำนวนเท่าของเวลาการทำงาน (Time dial setting: TDS) เพื่อให้รีเลย์สามารถทำงานได้อย่างเหมาะสมทั้งในสภาวะปกติ และสภาวะที่เกิดความผิดปกติต่าง ๆ ซึ่งการปรับตั้งค่าจำนวนเท่าของเวลาการทำงาน ในที่นี้จะใช้กำหนดการเชิงเส้น ซึ่งเป็นเครื่องมือของโปรแกรมแมทแลปมาช่วยในการหาค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางแต่ละตัวที่อยู่ในระบบไฟฟ้ากำลัง ทำให้ทราบว่า รีเลย์แต่ละตัวนั้นมีหน้าที่เป็นรีเลย์หลักหรือรีเลย์สำรองในการป้องกันความผิดปกติที่อาจเกิดขึ้นในระบบไฟฟ้ากำลัง รวมถึงทราบค่าของเวลาในการทำงานร่วมกันของรีเลย์ในระบบได้ นอกจากนี้ยังเป็นการช่วยลดการสูญเสียพลังงานไฟฟ้าจากการเกิดความผิดปกติไปโดยเปล่าประโยชน์ ซึ่งเป็นการช่วยประหยัดพลังงานไฟฟ้าและสามารถที่จะนำไฟฟ้าไปใช้ให้เกิดประโยชน์สูงสุดในประเทศได้

คำสำคัญ: รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง, รีเลย์หลัก, รีเลย์สำรอง, กำหนดการเชิงเส้น

Abstract

Directional over-current relay is an important protective device in power system. It is used to protect electric power apparatus in power system when a fault or an overload occurs. The relays are configured by adjusting time dial setting (TDS) of function properly in normal and fault conditions. The TDS adjustment is performed by using linear programming. For test, a 6-bus test power system is situated. The results illustrated that linear programming is simple and can be used for solving optimal coordination of directional over-current relays.

Keywords: Directional over-current relay, primary relay, backup relay, linear programming

1. บทนำ

ในระบบไฟฟ้ากำลัง ประกอบด้วย 3 ระบบ คือ ระบบผลิต ระบบส่งจ่ายและระบบจำหน่าย ซึ่งทั้ง 3 ระบบนี้มีความสำคัญทุกระบบเนื่องจาก ระบบผลิตนั้นเป็นแหล่งกำเนิดกระแสไฟฟ้าซึ่งผลมาจากทรัพยากรธรรมชาติเช่น ถ่านหิน ก๊าซธรรมชาติ หินน้ำมัน เป็นต้น นอกจากนี้ยังมีการผลิตกระแสไฟฟ้าด้วยพลังงานทดแทน เช่น แสงอาทิตย์ น้ำ ลม เป็นต้น จากนั้นจะส่งไปยังระบบส่งจ่ายผ่านสายส่งไฟฟ้าแรงสูงไปยังสถานีไฟฟ้าย่อยที่อยู่ในแต่ละภูมิภาคนั้นๆ และส่งต่อผ่านระบบจำหน่ายไปยังกลุ่มผู้ใช้ไฟฟ้า เพื่อใช้ประโยชน์ในการดำเนินชีวิตและประกอบกิจการต่างๆ ปัญหาที่เกิดขึ้นในระบบไฟฟ้ากำลังส่วนใหญ่จะเกิดขึ้นในระบบจำหน่าย ซึ่งมีสาเหตุปัจจัยหลายอย่างเช่น มนุษย์ สัตว์และสิ่งมีชีวิตต่างๆ ปรากฏการณ์ธรรมชาติ เป็นต้น สิ่งเหล่านี้อาจส่งผลกระทบต่อเกิดความผิดปกติในระบบไฟฟ้ากำลังได้ ดังนั้นในระบบไฟฟ้ากำลังจึงต้องมีระบบป้องกันความผิดปกติเพื่อป้องกันความเสียหายที่จะเกิดขึ้น อีกทั้งเพื่อให้ระบบมีความเชื่อถือได้ในการทำงานของอุปกรณ์ป้องกันและป้องกันการดับของไฟฟ้าเป็นบริเวณกว้าง อันเป็นสิ่งที่ไม่พึงปรารถนา ด้วยเหตุนี้จึงต้องมีระบบป้องกันที่มีประสิทธิภาพ ซึ่งอาศัยการจัดลำดับความสัมพันธ์ที่เหมาะสมของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง[1] ทั้งหมดที่อยู่ในระบบ โดยการปรับตั้งค่าจำนวนเท่าของเวลาการทำงาน [2] ของรีเลย์แต่ละตัว เพื่อที่จะสามารถกำจัดความผิดพร่องนั้นออกจากระบบได้อย่างมีประสิทธิภาพ จึงจำเป็นต้องมีการจัดลำดับความสัมพันธ์ที่เหมาะสมของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางที่มีประสิทธิภาพเช่นเดียวกัน

2. พื้นฐานการทำงานร่วมกันของรีเลย์

รูปแบบสายบ่อนแบบรัศมีที่แสดงในรูปที่ 1 [3-5] เมื่อกำหนดตำแหน่งที่เกิดความผิดปกติที่จุด F รีเลย์ R_5 จะทำหน้าที่เป็นรีเลย์หลักในการป้องกันความผิดปกติที่เกิดขึ้น ถ้ารีเลย์ R_5 ไม่ทำงานหรือเกิดความผิดพลาดในการทำงาน รีเลย์ R_4 จะทำหน้าที่เป็นรีเลย์สำรองในการป้องกันความผิดปกติที่เกิดขึ้นในระบบแทนรีเลย์ R_5 หรือ

สามารถอธิบายการทำงานได้ว่า เมื่อมีความผิดพลาดเกิดขึ้น รีเลย์ทั้งสองตัวจะตรวจพบสัญญาณความผิดพลาดที่เกิดขึ้นพร้อมกัน แต่เนื่องจากรีเลย์ R_B อยู่ใกล้กับตำแหน่งของ F จึงทำหน้าที่เป็นรีเลย์หลัก ซึ่งจะใช้เวลาในการกำจัดความผิดพลาดที่รวดเร็วกว่า ส่วนรีเลย์ R_A อยู่ไกลจากตำแหน่ง F จึงทำหน้าที่เป็นรีเลย์สำรองโดยมีการตั้งเวลาแบบหน่วงเวลาในการทำงาน เพื่อให้รีเลย์มีลำดับการทำงานร่วมกันอย่างถูกต้องและเหมาะสมในการป้องกันความผิดพลาดในระบบ



รูปที่ 1 ระบบสายป้อนแบบรีเลย์

3. ปัญหาการหาค่าเหมาะที่สุด

3.1 ฟังก์ชันวัตถุประสงค์

ปัญหาการจัดลำดับความสัมพันธ์ที่เหมาะสมของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางนี้ จะพิจารณาเฉพาะในส่วนของรีเลย์หลักเท่านั้น เนื่องจากเป็นรีเลย์ที่มีหน้าที่ป้องกันความผิดพลาดที่เกิดขึ้นในระบบไฟฟ้าเป็นอันดับแรก โดยรีเลย์ที่อยู่ใกล้กับตำแหน่งที่เกิดความผิดพลาด จะเรียกว่า รีเลย์ใกล้ความผิดพลาด ส่วนรีเลย์ที่อยู่ไกลจากตำแหน่งที่เกิดความผิดพลาดนั้นจะเรียกว่า รีเลย์ไกลความผิดพลาด เมื่อพิจารณาเฉพาะรีเลย์หลัก จะได้สมการของฟังก์ชันวัตถุประสงค์[6] ดังสมการ (1)

$$\text{Minimize } \text{OBJ} = \sum_{i=1}^{N_{cl}} T_{pr_cl_in}^i + \sum_{j=1}^{N_{far}} T_{pr_far_bus}^j \quad (1)$$

เมื่อ N_{cl} คือ จำนวนของรีเลย์ที่อยู่ใกล้ตำแหน่งความผิดพลาด

N_{far} คือ จำนวนของรีเลย์ที่อยู่ไกลตำแหน่งความผิดพลาด

$T_{pr_cl_in}^i$ คือ เวลาการทำงานของรีเลย์ที่อยู่ใกล้ความผิดพลาด

$T_{pr_far_bus}^j$ คือ เวลาการทำงานของรีเลย์ที่อยู่ไกลความผิดพลาด

3.2 เงื่อนไขของปัญหา

กำหนดขอบเขตของปัญหา[6] ดังนี้

$$1. \text{TDS}_{min}^i \leq \text{TDS}^i \leq \text{TDS}_{max}^i$$

เมื่อ $\text{TDS}_{min}^i = 0.05, \text{TDS}_{max}^i = 1.1$ และ i เป็นจำนวนของ

รีเลย์ที่อยู่ใกล้ตำแหน่งความผิดพลาด ซึ่งเริ่มต้นจาก 1 จนถึง N_{cl}

$$2. \text{PS}_{min}^i \leq \text{PS}^i \leq \text{PS}_{max}^i$$

เมื่อ $\text{PS}_{min}^i = 1.25, \text{PS}_{max}^i = 1.5$

$$3. T_{primary}$$
 มีค่าอยู่ระหว่าง 0.05 ถึง 1.00

$$4. T_{backup} - T_{primary} - \text{CTI} \geq 0$$

เมื่อ T_{backup} คือ เวลาการทำงานของรีเลย์ที่ทำหน้าที่เป็นรีเลย์

สำรอง $T_{primary}$ คือ เวลาการทำงานของรีเลย์ที่ทำหน้าที่เป็นรีเลย์

หลัก CTI คือ เวลาการจัดความสัมพันธ์ของรีเลย์ มีค่าเท่ากับ 0.3

วินาที

3.3 กำหนดการเชิงเส้น

วิธีการทางคณิตศาสตร์ในการวิเคราะห์ปัญหาต่างๆ ซึ่งมีความสัมพันธ์กันระหว่างตัวแปรทุกตัวเป็นแบบเชิงเส้น หรืออาจไม่มีความสัมพันธ์แบบเชิงเส้น แต่ต้องทำการปรับค่าของตัวแปรโดยมีข้อจำกัดจำนวนมาก เพื่อทำให้เป็นแบบเชิงเส้น โดยกระทำการได้ข้อจำกัดต่างๆ เพื่อเลือกทางเลือกที่มีความเหมาะสมมากที่สุด ได้แก่ การหาค่าสูงสุด (maximize) หรือค่าต่ำสุด (minimize)

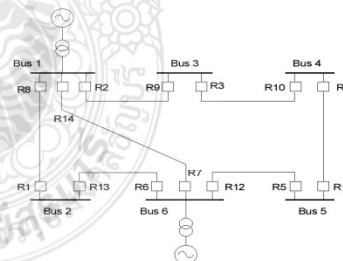
โครงสร้างของกำหนดการเชิงเส้นประกอบด้วย

1. ตัวแปรที่ต้องตัดสินใจ (Decision variable) คือ สิ่งที่ต้องการหาผลลัพธ์
2. ฟังก์ชันวัตถุประสงค์ (Objective function) คือ จะต้องมีความสัมพันธ์กันระหว่างตัวแปรทุกตัวเป็นแบบเชิงเส้น หรืออาจไม่มีความสัมพันธ์แบบเชิงเส้น แต่ต้องทำการปรับค่าของตัวแปรโดยมีข้อจำกัดจำนวนมาก เพื่อทำให้เป็นแบบเชิงเส้น โดยกระทำการได้ข้อจำกัดต่างๆ เพื่อเลือกทางเลือกที่มีความเหมาะสมมากที่สุด ได้แก่ การหาค่าสูงสุด (maximize) หรือค่าต่ำสุด (minimize)
3. เงื่อนไขบังคับ (Constraints) คือ สมการหรือสมการที่แสดงถึงขีดจำกัดในด้านความต้องการ หรือเงื่อนไขต่างๆ ของปัญหา
4. ข้อจำกัด (Restriction) คือ จะมีค่าตัวแปรที่ต้องตัดสินใจทุกตัวที่มีค่าไม่ติดลบ

4. การวิเคราะห์และระบบทดสอบ

4.1 กระแสไฟฟ้าลัดวงจรของรีเลย์หลักและรีเลย์สำรอง

การหาค่าของกระแสไฟฟ้าลัดวงจรของระบบ จะใช้โปรแกรม Power World Simulator โดยระบบที่พิจารณาเป็นระบบทดสอบขนาด 6 บัส ซึ่งประกอบด้วยสายส่งจำนวน 7 เส้น หม้อแปลง 2 ตัว รีเลย์ป้องกันกระแสเกินแบบมีทิศทาง 14 ตัว แสดงดังรูปที่ 2 [6] กำหนดข้อมูลของสายส่ง เครื่องกำเนิดไฟฟ้า หม้อแปลง กระแสไฟฟ้าลัดวงจรของรีเลย์หลักและรีเลย์สำรอง ดังตารางที่ 1 2 3 และ 4 ตามลำดับ [6]



รูปที่ 2 ระบบทดสอบขนาด 6 บัส

ตารางที่ 1 ข้อมูลของสายส่ง

สายส่ง	R (pu)	X (pu)	V (kv)
1	0.0018	0.0222	150
2	0.0018	0.0222	150
3	0.0018	0.02	150
4	0.0022	0.02	150
5	0.0022	0.02	150
6	0.0018	0.02	150
7	0.0022	0.0222	150

ตารางที่ 2 ข้อมูลของเครื่องกำเนิดไฟฟ้า

เครื่องกำเนิดไฟฟ้า	R (pu)	X (pu)	V (kv)
	0.000001	0.1	10

ตารางที่ 3 ข้อมูลของหม้อแปลง

หม้อแปลง	R (pu)	X (pu)
	0.000001	0.026666

ตารางที่ 4 ข้อมูลกระแสไฟฟ้าลัดวงจรของรีเลย์หลักและรีเลย์สำรอง

บัสที่ ลัดวงจร	รีเลย์ หลัก	รีเลย์ สำรอง	กระแสลัดวงจร ด้านปฐมภูมิ	กระแสลัดวงจร ด้านทุติยภูมิ
1	8	7	4,961.63	1,520.90
	8	9	4,961.63	410.83
	2	1	5,362.16	804.88
	2	7	5,362.16	1,528.07
14	1	4,232.59	794.10	
	9	4,232.59	407.23	
2	1	6	2,682.42	2,682.42
	13	8	2,490.68	2,490.68
3	3	2	3,334.45	3,334.45
	9	10	1,443.65	1,443.65
4	4	3	2,234.28	2,234.28
	10	11	2,334.62	2,334.62
5	5	4	1,352.84	1,352.84
	11	12	3,480.70	3,480.70
6	6	5	4,964.96	411.35
	6	14	4,964.96	1,522.84
	12	13	5,364.96	805.53
	12	14	5,364.96	1,529.3
	7	5	4,232.58	407.23
	7	13	4,232.58	794.09

4.2 กระแสไฟฟ้าในสภาวะปกติ

พิจารณาในสภาวะที่ระบบมีการจ่ายโหลด กำหนดโหลดที่บัส 2 มีขนาด 30 MW 10 MVAR โหลดที่บัส 3 มีขนาด 40 MW 15 MVAR, โหลดที่บัส 4 มีขนาด 25 MW 10 MVAR, โหลดที่บัส 5 มีขนาด 10 MW 8 MVAR และโหลดที่บัส 6 มีขนาด 20 MW 5 MVAR ซึ่งได้ค่ากระแสไฟฟ้าของระบบดังตารางที่ 5

ตารางที่ 5 กระแสไฟฟ้าในสภาวะจ่ายโหลด

รีเลย์	กระแสโหลด	รีเลย์	กระแสโหลด
1	132.69	8	132.69
2	220.51	9	220.51
3	54.62	10	54.62
4	52.1	11	52.1
5	101.79	12	101.79
6	26.1	13	26.1
7	145.08	14	145.08

จากนั้น เลือกอัตราส่วนของหม้อแปลงกระแส โดยพิจารณาจากการยอมให้เกิดสภาวะโหลดเกินได้ไม่เกิน 150% จะได้ค่าดังตารางที่ 6 เมื่อเลือกค่าอัตราส่วนหม้อแปลงกระแสของรีเลย์แต่ละตัวแล้ว สามารถหาค่าพารามิเตอร์ต่างๆของสมการฟังก์ชันวัดคู่ประสงค์ (1) และสมการเงื่อนไขได้ แสดงค่าพารามิเตอร์ในตารางที่ 7 และตารางที่ 8

ตารางที่ 6 อัตราส่วนของหม้อแปลงกระแส

รีเลย์	อัตราส่วน หม้อแปลง	รีเลย์	อัตราส่วน หม้อแปลง
1	200:5	8	200:5
2	400:5	9	400:5
3	100:5	10	100:5
4	100:5	11	100:5
5	150:5	12	150:5
6	50:5	13	50:5
7	250:5	14	250:5

ตารางที่ 7 ค่าคงที่ a^i , b^i , c^i และ d^i

ใกล้จุดลัดวงจร			ไกลจุดลัดวงจร		
รีเลย์	a^i	b^i	รีเลย์	c^i	d^i
1	67.06	3.32	8	63.19	3.32
2	67.02	2.76	9	5.35	2.76
3	166.72	2.73	10	73.08	2.73
4	111.71	2.61	11	117.70	2.61
5	45.05	3.39	12	116.94	3.39
6	496.48	2.61	13	82.95	2.61
7	84.65	2.90	14	31.31	2.90
8	124.04	3.32	1	20.79	3.32
9	18.05	2.76	2	42.03	2.76
10	116.72	2.73	3	112.66	2.73
11	174.03	2.61	4	68.50	2.61
12	178.81	3.39	5	14.23	3.39
13	249.06	2.61	6	272.05	2.61
14	84.65	2.90	7	31.31	2.90

ตารางที่ 8 ค่าคงที่ e^i , f^i , g^i และ h^i

รีเลย์สำรอง			รีเลย์หลัก		
รีเลย์	e^i	f^i	รีเลย์	g^i	h^i
1	20.12	3.32	2	67.03	2.76
1	19.85	3.32	14	84.65	2.90
1	7.83	3.32	2	42.03	2.76
2	41.68	2.76	3	166.72	2.73
2	28.16	2.76	3	112.66	2.73
3	111.71	2.73	4	111.71	2.61
3	68.50	2.73	4	68.50	2.61
4	67.64	2.61	5	45.09	3.39
4	21.36	2.61	5	14.24	3.39
5	13.57	3.39	7	84.65	2.90
5	13.71	3.39	6	496.50	2.61
6	268.24	2.61	1	67.06	3.32
6	83.16	2.61	1	20.79	3.32
7	30.42	2.90	8	124.04	3.32
7	30.56	2.90	2	67.03	2.76
7	11.89	2.90	2	42.03	2.76
8	62.27	3.32	13	249.07	2.61
8	20.74	3.32	13	82.96	2.61
9	5.14	2.76	8	124.04	3.32
9	5.09	2.76	14	84.65	2.90
10	72.18	2.73	9	18.05	2.76
10	21.42	2.73	9	5.35	2.76
11	116.73	2.61	10	116.73	2.73
11	73.09	2.61	10	73.09	2.73
12	116.02	3.39	11	174.04	2.61
12	78.47	3.39	11	117.70	2.61
13	80.55	2.61	12	178.83	3.39
13	79.41	2.61	7	84.65	2.90
13	35.17	2.61	12	116.96	3.39
14	30.46	2.90	6	496.50	2.61
14	30.59	2.90	12	178.83	3.39
14	13.36	2.90	12	116.96	2.90

4.3 ผลลัพธ์ของฟังก์ชันวัตถุประสงค์

หาค่าของเวลาในการจัดลำดับความสัมพัทธ์ที่เหมาะสมของรีเลย์ป้องกันกระแสเกินแบบมีทิศทาง และหาค่าจำนวนเท่าของเวลาการทำงานของรีเลย์ทุกตัว โดยใช้วิธีกำหนดการเชิงเส้น [7-8] จะได้ผลลัพธ์แสดงในตารางที่ 9

ตารางที่ 9 ค่าจำนวนเท่าของเวลาการทำงานของรีเลย์

รีเลย์	จำนวนเท่าของ เวลาการทำงาน	รีเลย์	จำนวนเท่าของ เวลาการทำงาน
1	0.3223	8	0.5320
2	0.4833	9	0.0777
3	0.5918	10	0.4102
4	0.4428	11	0.5746
5	0.2311	12	0.6216
6	0.8019	13	0.6716
7	0.4378	14	0.4419

จะได้ค่าของฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 30.0752 วินาที

5.สรุป

ปัญหาการจัดลำดับความสัมพัทธ์ที่เหมาะสมของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลัง เพื่อป้องกันความผิดพลาดที่เกิดขึ้นในระบบไฟฟ้านี้ ได้นำวิธีกำหนดการเชิงเส้นซึ่งเป็นเครื่องมือของโปรแกรมเมทแลปมาใช้ในการแก้ปัญหา โดยพิจารณาระบบทดสอบขนาด 6 บัส เพื่อหาค่าของฟังก์ชันวัตถุประสงค์ที่ต้องการโดยมีความต้องการหาค่าเวลาที่น้อยที่สุดในการทำงานร่วมกันของรีเลย์ ซึ่งจากการแก้ปัญหา จะได้ค่าของฟังก์ชันวัตถุประสงค์มีค่าเท่ากับ 30.0752 วินาทีและหาค่าจำนวนเท่าของเวลาการทำงาน สำหรับปรับปรุงของรีเลย์ป้องกันกระแสเกินแบบมีทิศทางทุกตัวในระบบไฟฟ้ากำลัง จากการนำวิธีกำหนดการเชิงเส้นมาแก้ปัญหาในเรื่องนี้ เนื่องจากวิธีกำหนดการเชิงเส้นเป็นวิธีการอย่างง่ายและไม่ซับซ้อนในการแก้ปัญหา เป็นวิธีที่นิยมใช้ในการแก้ปัญหาแบบเชิงเส้น จึงได้นำวิธีนี้มาใช้ในการแก้ปัญหา ซึ่งเป็นวิธีการที่ดีวิธีหนึ่งในการแก้ปัญหา

เอกสารอ้างอิง

- [1] Kawahara K., Sasaki H., and Sugihara H., "An application of rule based system to the coordination of directional overcurrent relays". Developments in power system protection, Sixth International Conference on(Conf. Publ.No 434),1997,Page:58-61.
- [2] Aggarwai C.,Mangalvedekar H.A., and Chaudhari H.B., "Effect of fault location on optimal coordination of directional Overcurrent relay". Tencon 2008-2009 IEEE Region 10 conference, pp.1-5.
- [3] Bedekar P.P., Bhide S.R., and Kale V.S., "Optimum Coordination of Overcurrent Relays in Distribution System Using Dual Simplex Method", Emerging Trends in Engineering and Technology ICETET), 2009 2nd International

Conference on. , pp.555-559.

- [4] Bedekar P.P., Bhide S.R., and Kale V.S., " Optimum Coordination of Overcurrent relays in distribution system using genetic algorithm", Powersystems,2009.ICPS'09. International conference on,Page.1-6
- [5] Bedekar P.P., Bhide S.R., and Kale V.S., " Optimum Time Coordination of Overcurrent Relays in Distribution System Using Big-M (Penalty) Method", WSEAS transactions on Power systems. Issue 11, Volume 4, November 2009.
- [6] Thangaraj R., Pant M., and Deep K., "Optimal Coordination of Overcurrent Relays using Modified Differential Evolution Algorithms", Engineering Applications of Artificial Intelligence, Elsevier Science, Vol. 23 (5), pp. 820 – 829, 2010.
- [7] The MathWorks Inc., Optimization Toolbox, CD-ROM Manual,2004.
- [8] Chattopadhyay B., Sachdev M.S., and Sidhu T.S., "An on-line relay coordination algorithm for adaptive protection using linear programming technique" ,Power Delivery,IEEE Transactions on Vol. 11, Issue 1, pp. 165 - 173, 1996.

Application of Linear Programming for Optimal Coordination of Directional Over-current Relays

Supawatchara Niyomphant

Power System Research Unit, School of Electrical
Engineering Institute of Engineering, Suranaree University
of Technology, Nakhon Ratchasima, THAILAND 30000
E-mail: supawatchara_@hotmail.com

Uthen Leeton

Power System Research Unit, School of Electrical
Engineering Institute of Engineering, Suranaree University
of Technology, Nakhon Ratchasima, THAILAND 30000
E-mail: sut.mems@gmail.com

Thanatchai Kulworawanichpong

Power System Research Unit, School of Electrical
Engineering Institute of Engineering, Suranaree University
of Technology, Nakhon Ratchasima, THAILAND 30000
E-mail: Thanatchai@gmail.com

Nimit Chomnawang

Microelectromechanical Systems Lab, School of Electrical
Engineering Institute of Engineering, Suranaree University
of Technology, Nakhon Ratchasima, THAILAND 30000
E-mail: supawatchara_@hotmail.com

Abstract—In this paper presents an algorithm for directional over-current relay coordination based on linear programming. The relays are configured by adjusting time dial setting (TDS) of function properly in normal and fault conditions. The TDS adjustment is performed by using linear programming. For test, 6-bus and 9-bus test power systems are situated. Furthermore, there are also identified primary relay and backup relay pairs when fault occurred. Performance above that can solve the problems; adjust TDS of each relay and determine the minimum time relay coordination. The results illustrated that linear programming is simple and can be used for solving optimal coordination of directional over-current relays.

Keywords—Directional over-current relay; primary relay; backup relay; linear programming

I. INTRODUCTION

Directional over-current relay is an important protective device in power system. It is used to protect electric power apparatus in power system when a fault or an overload occurs. Power system protection divided protected zone to protect, which can be divided into zone to protect the generator, sub-station, transmission line etc. Each zone is responsible for prevention and protections operate in separate zone of responsibility as quickly as possible from the system when fault occur in the system. This system is called primary protection system and if primary protection system fails or does not operate must have a backup protection system, which is responsible for the operations of the primary protection system to prevent damage to occur. Therefore should note that the relay position has served as primary relay or backup relay in case of a fault occurred. And to provide more effective protection, Relays must be coordinated in power systems.

This paper organizes a total of eight sections. Next section, Section II discusses basis of directional over-current relays coordination in power systems. Section III, describes the steps of pairs the operation of relay. Section

IV is the brief of linear programming technique. Section V described problem of relay coordination, which are objective function and constraints. Section VI discusses testing and analysis to power systems. The results of the solution show in section VII. Conclusion is in section VIII.

II. BASIS OF DIRECTIONAL OVER-CURRENT RELAY COORDINATION IN POWER SYSTEM

A radial feeder

A radial feeder is shown in figure 1 [1]-[2]. Define fault at point F. Relay R_B is first to operate, which called primary relay. If relay R_B is not operate or mal-operated. Relay R_A is second to operate, it is called backup relay. Describing the operation that when fault occurs, both relays are detected fault signal. Relay R_B is primary relay because the ones is close point F more than relay R_A . Relay R_A is backup relay which setting time delay to operate for proper protection and efficiency.



Figure 1. A radial feeder

III. THE PROBLEM OF PAIRS THE OPERATION OF RELAY

Methods of pairs the operation of the relay

The pairs operation of the relay when a fault occurs in the power systems to monitor and detect errors in the operation of the relay is easy. In this section of the application process MATLAB to solve a pair of relays, which will determine the direction of short-circuit currents, that the correct conditions of the relay or not. Then, show operation of the relay. The steps shown in Figure 2.

IV. LINEAR PROGRAMMING TECHNIQUE

A mathematical method to analyze problems, the relationship between variables is linear which performance under the constraints to select the appropriate alternative between minimum or maximum value. The structures of linear programming technique are consisting:

- Decision variable:* The desired results, is the time dial setting (TDS).
- Objective function:* Aims to find the minimum or maximum value.
- Constraints:* Equation or inequality that represents the extent of demand.
- Restriction:* Every decision variable is not negative.

The general linear programming problem can be expressed as a constrained optimization problem as follows.

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{Subject to} && A.x \leq b, \text{ inequality constraints} \\ &&& Aeq.x = beq, \text{ equality constraints} \\ &&& lb \leq x \leq ub, \text{ lower and upper boundary of } x \end{aligned}$$

Where, f , x , b , beq , lb and ub are vectors, A and Aeq are matrices.

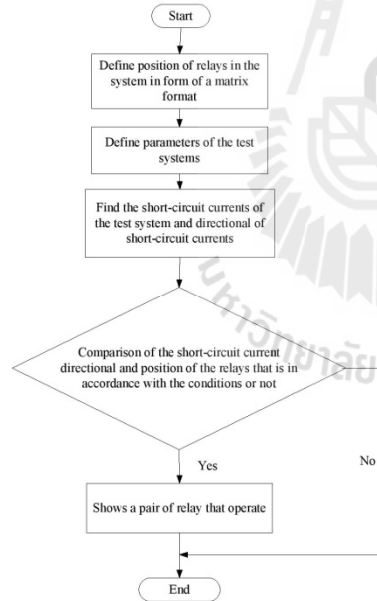


Figure 2. Flowchart of the pairs operation of the relay

V. RELAY COORDINATION PROBLEMS

A. Objective function

The optimal coordination problem of directional over-current relay will consider only the primary relay because primary relay is responsible for protecting the fault that occurs first. Relay near the fault position, it called a primary relay for close-in fault. And relay is far from the fault position, is called a backup relay for far-bus fault. The objective function [3]-[4] is defined as follows:

$$\text{Minimize } OBJ = \sum_{i=1}^{N_{cl}} T_{pr_cl_in}^i + \sum_{j=1}^{N_{far}} T_{pri_far_bus}^j \quad (1)$$

$$T = \frac{\left(\frac{K_i \times TDS}{\frac{I}{I_p}}\right)^{k_2}}{\left(\frac{I}{I_p}\right)^{k_2} - 1} = \frac{0.14 \times TDS}{\left(\frac{I}{PS \times CT_rating}\right)^{0.02} - 1} \quad (2)$$

$$T_{pr_cl_in}^i = \frac{0.14 \times TDS}{\left(\frac{d^i}{PS \times b^i}\right)^{0.02} - 1} \quad (3)$$

$$T_{pri_far_bus}^j = \frac{0.14 \times TDS}{\left(\frac{c^j}{PS \times d^j}\right)^{0.02} - 1} \quad (4)$$

Where, N_{cl} is number of relays responding for close-in fault, N_{far} is number of relays responding for far-bus fault, $T_{pr_cl_in}^i$ is an operating time of primary relay for close-in fault, $T_{pri_far_bus}^j$ is an operating time of primary relay for far-bus fault, PS is plug setting, CT_rating is load current passing through the current transformer, I is fault current passing through the relay.

B. Constraints

Define constraints [3]-[4] are as follows:

$$a) TDS_{min}^i \leq TDS^i \leq TDS_{max}^i$$

Where, $TDS_{min}^i = 0.05$, $TDS_{max}^i = 1.1$ and i is the number of relay near the fault which started from 1 up to N_{cl}

$$b) PS_{min}^i \leq PS^i \leq PS_{max}^i$$

Where, $PS_{min}^i = 1.25$, $PS_{max}^i = 1.5$

$$c) T_{primary}^i \text{ is between } 0.05 \text{ to } 1.00$$

$$d) T_{backup} - T_{primary} - CTI \geq 0$$

$$T_{backup}^j = \frac{0.14 \times TDS}{\left(\frac{e^j}{PS^j \times f^j}\right)^{0.02} - 1} \quad (5)$$

$$T_{primary}^j = \frac{0.14 \times TDS}{\left(\frac{g^j}{PS^j \times h^j}\right)^{0.02} - 1} \quad (6)$$

Where, T_{backup} is operating time of backup relay, $T_{primary}$ is operating time of primary relay. CTI is coordination time interval, is equal to 0.3 seconds.

VI. TESTING AND ANALYSIS

A. Short circuit current of the primary relay and backup relay

Determination of short-circuit current are use power world simulator program which was considered in 6 bus and 9 bus test systems, consist of generator, line, transformer and directional over-current relay. The test systems are shown in Figure 3 [5] and 4 [6], respectively.

B. Load current

Consider the 6 bus test system in a state where the system has to be loaded. The load on the bus 2 (30 MW, 10 MVAR), bus 3 (40 MW, 15 MVAR), bus 4 (25 MW, 10 MVAR), bus 5 (10 MW, 8 MVAR), bus 6 (20 MW, 5 MVAR). And 9 bus system, defined load on bus 5 (20 MW, 15 MVAR), bus 7 (50 MW, 30 MVAR), bus 9 (20 MW, 10 MVAR). Then run power world simulator program, the value of load current as shown in table 1 and 2. Short-circuit currents as shown in table 3 and 4 respectively. After that, select ratio of current transformer based on to allowing a maximum load conditions up to 150 %

VII RESULTS

Pairs of primary relay and backup relay, TDS and objective function value

From solution of a pairs of relays and coordination of directional over-current relays by using MATLAB program, the results of pairs of relays when fault condition on each bus. Consider fault occurring is symmetrical three-phase fault. Pairs of relays as shown in table 5 and 6. And time dial setting value as shown in table 7 and 8.

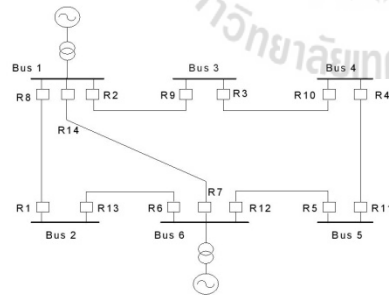


Figure 3. 6 bus test system

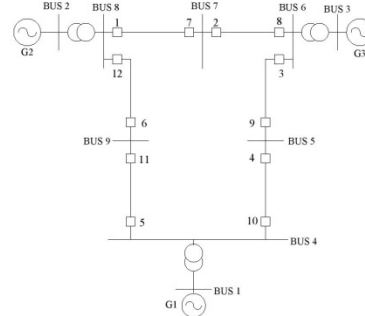


Figure 4. 9 bus test system

TABLE I. LOAD CURRENT OF 6 BUS TEST SYSTEM

Relay	Load current (A)	CT	Relay	Load current (A)	CT
1	132.69	200:5	8	132.69	200:5
2	220.51	400:5	9	220.51	400:5
3	54.62	100:5	10	54.62	100:5
4	52.1	100:5	11	52.1	100:5
5	101.79	150:5	12	101.79	150:5
6	26.1	50:5	13	26.1	50:5
7	145.08	250:5	14	145.08	250:5

TABLE II. LOAD CURRENT OF 9 BUS TEST SYSTEM

Relay	Load current (A)	CT	Relay	Load current (A)	CT
1	78.6	150:5	7	78.6	150:5
2	59.43	100:5	8	59.43	100:5
3	72.44	100:5	9	72.44	100:5
4	115.79	200:5	10	115.79	200:5
5	116.4	200:5	11	116.4	200:5
6	78.02	150:5	12	78.02	150:5

TABLE III. THE SHORT-CIRCUIT CURRENT OF THE 6 BUS TEST SYSTEM

Bus	Primary relay	Backup relay	Primary short-circuit current	Secondary short-circuit current
1	8	7	4961.63	1520.90
	8	9	4961.63	410.83
	2	1	5362.16	804.88
	2	7	5362.16	1528.07
	14	1	4232.59	794.10
2	14	9	4232.59	407.23
	1	6	2682.42	2682.42
	13	8	2490.68	2490.68
3	3	2	3334.45	3334.45
	9	10	1443.65	1443.65
4	4	3	2234.28	2234.28
	10	11	2334.62	2334.62
5	5	4	1352.84	1352.84
	11	12	3480.70	3480.70
6	6	5	4964.96	411.35
	6	14	4964.96	1522.84
	12	13	5364.96	805.53
	12	14	5364.96	1529.3
	7	5	4232.58	407.23
	7	13	4232.58	794.09

TABLE IV. THE SHORT-CIRCUIT CURRENT OF THE 9 BUS TEST SYSTEM

Bus	Primary relay	Backup relay	Primary short-circuit current	Secondary short-circuit current
4	10	11	2111.27	610.43
	5	4	2079.42	579.61
5	4	3	831.7	806.09
	9	10	1141.33	1132.38
6	3	2	2243.65	754.05
	8	9	2053.75	547.05
7	2	1	1244.47	1237.48
	7	8	1074.59	1063.05
8	1	6	2036.20	580.45
	12	7	2193.34	758.89
9	11	12	857.61	836.20
	6	5	1174.09	1164.97

TABLE V. RESULTS OF IDENTIFY THE RELAY OPERATES OF 6 BUS TEST SYSTEM

Bus fault	Primary protection	Secondary protection
	Relay number	Relay number
1	2	9
	8	1
	14	7
2	1	8
	13	6
3	3	10
	9	2
4	4	11
	10	3
5	5	12
	11	4
6	6	13
	7	14
	12	5

TABLE VI. RESULTS OF IDENTIFY THE RELAY OPERATES OF 9 BUS TEST SYSTEM

Bus fault	Primary protection	Secondary protection
	Relay number	Relay number
4	5	11
	10	4
5	4	10
	9	3
6	3	9
	8	2
7	2	8
	7	1
8	1	7
	12	6
9	6	12
	11	5

TABLE VII. TIME DIAL SETTING VALUE OF 6 BUS TEST SYSTEM

Relay	TDS	Relay	TDS
1	0.2693	8	0.4907
2	0.4218	9	0.0500
3	0.5218	10	0.3993
4	0.3806	11	0.5554
5	0.1848	12	0.5930
6	0.7204	13	0.6346
7	0.3781	14	0.4049
Objective function = 29.8687 seconds			

TABLE VIII. TIME DIAL SETTING VALUE OF 9 BUS TEST SYSTEM

Relay	TDS	Relay	TDS
1	0.3805	7	0.4474
2	0.3120	8	0.6121
3	0.3125	9	0.3811
4	0.1751	10	0.3845
5	0.2553	11	0.2444
6	0.2810	12	0.3891
Objective function = 26.4820 seconds			

VIII. CONCLUSIONS

A pairs of relays to monitor and detect errors in the operation of the relay. Power system protection allowing a clear division and coordination of directional overcurrent relay in power system problem to protect fault to occur. This has solved the problem using linear programming [7] based on 6 bus and 9 bus test systems to determine the minimum time relay coordination and time dial setting of all relay. The reason of the linear programming to solve the problem because this method is simple and uncomplicated way to solve the problem. It also reduces the complexity of many variables on solution of large power systems.

REFERENCES

- [1] P.P. Bedekar, S.R. Bhide, and V.S. Kale, "Optimum Coordination of Overcurrent Relays in Distribution System Using Dual Simplex Method", International Conference on Emerging Trends in Engineering and Technology 2nd (ICETET) 2009, pp.555-559.
- [2] P.P. Bedekar, S.R. Bhide, and V.S. Kale, "Optimum Coordination of overcurrent relays in distribution System using genetic algorithm", International conference on Power systems 2009 ICPS'09, pp.1-6.
- [3] M. Lin, A. K. Srivastava, and N. N. Schulz, "Modeling consideration in static and dynamic voltage stability studies of shipboard power systems". SCSC'2007, pp.53-60
- [4] R. Thangaraj, M. Pant, and K. Deep, "Optimal Coordination of overcurrent relay using Modified Differential Evolution Algorithms", Engineering Applications of Artificial Intelligence, Elsevier Science 2010, Vol. 23 (5), pp. 820 – 829.
- [5] F. Razavi, H. Askarian, M. Aldabbagh, and R. Mohammadi, "A new comprehensive genetic algorithm method for optimal overcurrent relays coordination", Electr. Power Syst. Res. doi:10.1016/j.epsr.2007.05.013, (2007).
- [6] X. Yu and C. Singh, "Probabilistic Analysis of Load ability in Composite Power Systems Considering Security Constraints" Power Systems Conference And Exposition, IEEE PES, pp. 832-838, Nov. 2006.
- [7] The MathWorks Inc., Optimization Toolbox, CD ROM, 2004.

ประวัติผู้เขียน

นายศุภวัชร นิยมพันธุ์ เกิดเมื่อวันที่ 10 ธันวาคม พ.ศ. 2530 สำเร็จการศึกษาระดับชั้นประถมศึกษาจากโรงเรียนอนุบาลสุรินทร์ ชั้นมัธยมศึกษาตอนต้นจากโรงเรียนสุรวิทยาคารและชั้นมัธยมศึกษาตอนปลายจากโรงเรียนสิรินธร อำเภอเมือง จังหวัดสุรินทร์ และสำเร็จการศึกษาระดับปริญญาตรี วิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) จากมหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปีการศึกษา พ.ศ. 2552 โดยหลังจากสำเร็จการศึกษาได้รับใบอนุญาตเป็นผู้ประกอบวิชาชีพวิศวกรรมควบคุม ระดับภาคีวิศวกร สาขาวิศวกรรมไฟฟ้ากำลัง

ปี พ.ศ.2553 เข้าศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี โดยขณะศึกษาได้ทำหน้าที่เป็นผู้สอนปฏิบัติการของสาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จำนวน 3 รายวิชา ได้แก่ (1) ปฏิบัติการเครื่องจักรกลไฟฟ้า 1 (2) ปฏิบัติการเครื่องจักรกลไฟฟ้า 2 (3) ปฏิบัติการวิศวกรรมไฟฟ้า 1 ในระหว่างการทำวิจัยวิทยานิพนธ์ผู้วิจัยมีความสนใจในงานด้านระบบไฟฟ้ากำลัง ซึ่งช่วยให้ผู้วิจัยได้นำประสบการณ์และความรู้ที่ได้จากการเป็นผู้ช่วยสอนและวิจัยมาประยุกต์ใช้กับงานวิจัยได้เป็นอย่างดี จากการทำวิจัยนี้ทำให้ผู้วิจัยมีความรู้และความเข้าใจทางด้านการประสานสัมพันธ์เลย์ป้องกันกระแสเกินแบบมีทิศทางในระบบไฟฟ้ากำลัง การเขียนโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) และมีผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างทำการวิจัยวิทยานิพนธ์โดยมีรายละเอียดปรากฏในภาคผนวก ค.