



การวิเคราะห์ระบบส่งจ่ายพลังงานไฟฟ้าแบบสามเฟส

THREE PHASE LOAD FLOW ANALYSIS



โดย

นายกิติชัย

เกียรติลักษณ์

นายประณิธิ

เชิธรวิชพันธ์

นายพลศักดิ์

ดวงศิริกุลวัฒนา

วัน เดือน ปี... ๑๐ ม.ค. ๒๕๖๑

เลขทะเบียน... ๐๖๕๐๑๙

เลขเรียกหนังสือ... T ๒๖๒๑๙ ๓๖

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๖๑

ปริญญาโทบริหารการศึกษา 2537


ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวิเคราะห์ระบบส่งจ่ายพลังงานไฟฟ้าแบบสามเฟส

ผู้จัดทำ

- 
1. นายกิตติชัย เกียรติลัษชานนท์
  2. นายประณิธิ เชียรวิชพันธุ์
  3. นายพลศักดิ์ ดวงศิริกุลวัฒนา

 อาจารย์ที่ปรึกษา

( ผศ.ศิริวัฒน์ โปธิเวชกุล )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ	I
ABSTRACT	II
สารบัญรูป	III
สารบัญตาราง	IV
บทที่ 1 บทนำ	1
บทที่ 2 การหาผลเฉลยของระบบสมการเชิงเส้น	3
บทที่ 3 การศึกษาโหลดโพลาร์	6
3.1 การแยกประเภทโพลาร์ในระบบไฟฟ้ากำลัง	6
3.2 สมการการไหลของกำลังไฟฟ้าในสายส่ง	7
3.3 การควบคุมกำลังไฟฟ้าในระบบโครงข่าย	8
3.4 การควบคุมกำลังไฟฟ้าในระบบโครงข่าย	11
3.5 วิธีการคำนวณแบบ Gauss	15
3.6 วิธีการคำนวณแบบ Gauss-Siedal	17
3.7 วิธีการคำนวณแบบ Newton-Raphson	18
3.8 ทฤษฎีและวิธีการคำนวณแบบ Fast decouple	29
บทที่ 4 การวิเคราะห์ระบบสามเฟส	36
4.1 การวิเคราะห์แกนอ้างอิง	36
4.2 แบบจำลองของสายส่งสามเฟส	46

<b>บทที่ 5 การศึกษาโหนดโฟลว์แบบสามเฟส</b>	<b>61</b>
5.1 บทนำ	61
5.2 สัญลักษณ์ต่างๆ	61
5.3 ปัญหาของสูตรคำนวณที่ใช้ในโหนดโฟลว์แบบสามเฟส	63
5.4 อัลกอริทึมของ Fast decouple แบบสามเฟส	68
<b>บทที่ 6 รูปและวิจารณ์</b>	<b>84</b>

ภาคผนวก

กิตติกรรมประกาศ

เอกสารอ้างอิง



รูป 3.1 การไหลของกำลังไฟฟ้า	7
รูป 3.2 การควบคุมระบบกำลังไฟฟ้า	9
รูป 3.3 เฟสเซอร์ไดอะแกรม สำหรับสายส่งกำลังผ่านอินพีแดนซ์ที่ต่ออนุกรม	11
รูป 3.4 หม้อแปลงเรกเตดิง สำหรับการควบคุมขนาดแรงดัน	13
รูป 3.5 หม้อแปลงเรกเตดิง สำหรับการควบคุมมุมเฟส	14
รูป 3.6 การแก้ปัญหาโหลดโฟลว์ โดยวิธีทำซ้ำแบบเกาส์ (Gauss method)	32
รูป 3.7 การแก้ปัญหาโหลดโฟลว์ โดยวิธีทำซ้ำแบบเกาส์-ไซเคิล (Gauss-Seidel method)	33
รูป 3.8 การแก้ปัญหาโหลดโฟลว์ โดยวิธีทำซ้ำแบบนิวตัน-ราฟสัน แบบพิกัดฉาก (Newton-Raphson method by Polar)	34
รูป 3.9 การแก้ปัญหาโหลดโฟลว์ โดยวิธีทำซ้ำแบบนิวตัน-ราฟสัน แบบพิกัดเชิงขั้ว (Newton-Raphson method by Rectangular)	35
รูป 4.1 แอตมิตแตนซ์ของระบบสามเฟสในแต่ละองค์ประกอบ	36
รูป 4.2 โครงข่ายเบื้องต้นของแอตมิตแตนซ์ที่คัปเปิลกัน 6 ชุด	40
รูป 4.3 แอตมิตแตนซ์ที่คัปเปิลกันของสองกลุ่ม	41
รูป 4.4 ตัวอย่างโครงข่ายที่แทนโดย จึงเกิดแอตมิตแตนซ์	43
รูป 4.5 ตัวอย่างโครงข่ายที่แทนโดย คอมพาวน์ แอตมิตแตนซ์	43
รูป 4.6 โครงข่ายเบื้องต้นและเมตริกซ์ของแอตมิตแตนซ์	44
รูป 4.7 วงจรสมมูลย์ของสายส่งสามเฟสแบบอนุกรม และแบบขนาน	47
รูป 4.8 แบบจำลอง Lumped- $\pi$ ของสายส่งสามเฟสระยะสั้นของอินพีแดนซ์แบบอนุกรม	50
รูป 4.9 คอมพาวน์ แอตมิตแตนซ์ ที่แทน การคัปเปิลของสายส่ง 3 เฟส 2 ชุด	51

	หน้า
รูป 4.10 คอมพาวนด์ แอตมิตแดนซ์ ที่แทน การคับเปิดของสายส่ง 3 เฟส 2 กลุ่ม	52
รูป 4.11 การเหนี่ยวนำระหว่างสายของสายส่งแบบขนาน	52
รูป 4.12 คาปาซิเตอร์ แบงค์ แบบขนาน	55
รูป 4.13 คาปาซิเตอร์ แบงค์ แบบอนุกรม ระหว่างโนด	56
รูป 4.14 ตัวอย่าง สายส่งที่แยกกันเป็นส่วนย่อย	57
รูป 4.15 พารามิเตอร์ที่ใช้ในการแปลงระบบสายส่ง	58
รูป 4.16 ตัวอย่างของระบบ เพื่อใช้อธิบายการแยกส่วนของส่วนส่ง	59
รูป 5:1 แบบจำลองของเครื่อง ริงโครนัส	65
รูป 5.2 เมตริกซ์ ของ คาปาซิเตอร์ แบบขนาน	76
รูป 5.3 ลำดับการทำซ้ำ สำหรับการศึกษาโหนดโฟลว์ของระบบไฟฟ้าสามเฟส	77
รูป 5.4 โครงสร้างโปรแกรม	79
รูป 5.5 การดูเข้าของกำลังไฟฟ้า สำหรับการแก้ปัญหาโหนดโฟลว์ทั้งแบบสามเฟส และแบบเฟสเดียว	81
รูป 5.6 การดูเข้าของศักดาไฟฟ้า สำหรับการแก้โหนดโฟลว์ทั้งแบบสามเฟสและ แบบเฟสเดียว	82

## สารบัญตาราง

หน้า

ตารางที่ 4.1 พารามิเตอร์ ABCD

59

ตารางที่ 5.1 ผลของการรู้เข้า

81



## การวิเคราะห์ระบบส่งจ่ายพลังงานไฟฟ้าแบบสามเฟส

นายกิตติชัย เกียรติลิขิตชานนท์  
นายประณิธิ เขียวฉวีพันธุ์  
นายพุลศักดิ์ ดวงศิริกุลวัฒนา  
ผศ.ศิริวัฒน์ โทธิเวชกุล อาจารย์ที่ปรึกษา  
อ.อานันท์วัฒน์ कुमार อาจารย์ที่ปรึกษา  
ปีการศึกษา 2537

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาและวิเคราะห์ระบบการส่งจ่ายพลังงานไฟฟ้า แบบสามเฟส ทั้งในสถานะที่ระบบรับภาระโหลดสมมูล และรับภาระโหลดไม่สมมูล โดยที่เมื่อระบบรับภาระโหลดสมมูล จะใช้การวิเคราะห์โหลดโพลาร์แบบเฟสเดียว(โหลดโพลาร์แบบสามเฟสสมมูล) และเมื่อระบบรับภาระโหลดไม่สมมูลจะทำการวิเคราะห์ โดยใช้โหลดโพลาร์แบบสามเฟส

จากการศึกษาและวิเคราะห์ระบบการส่งจ่ายพลังงานไฟฟ้าแบบสามเฟส สามารถนำไปช่วยในการวางแผนพัฒนาระบบในอนาคต เช่น การเชื่อมต่อระบบไฟฟ้ากำลังเพิ่มเติม การต่อโหลดเพิ่มเติม เป็นต้น รวมทั้งการศึกษาดังเสถียรภาพของระบบ

## THREE PHASE LOAD FLOW ANALYSIS

Kitichai Kietlatchanon

Pranithi Theanvanichpan

Poolsak Toungsirikulwattana

Assit.Prof Siriwat Photivejkul Advisor

Ananthawat Kunakorn Advisor

1994

### **Abstract**

The purpose of this study is to analyze the supply or absorbing power of three phase systems with balance loads and unbalance loads.

Analysis power system on steady state condition in three phase balance loads uses Single phase load flow for analysis is adequate. But in three phase unbalance loads, three phase load flow is necessary.

Studies and analysis of three phase distribution are essential for improved plans in future, such as connecting power systems with others and connecting load, etc.

## บทนำ

ระบบไฟฟ้ากำลัง อันประกอบด้วยเครื่องกำเนิดไฟฟ้า หม้อแปลงไฟฟ้า พาสซีฟโหลด แอกทีฟโหลด และระบบสายส่ง โดยที่ระบบไฟฟ้ากำลังจะต้องทำการจ่ายกำลังจริงและกำลังเสมือนให้เพียงพอต่อความต้องการเพื่อความมีเสถียรภาพของระบบไฟฟ้ากำลัง ดังนั้นจะต้องทำการศึกษาการส่งจ่ายพลังงานไฟฟ้า

โหลดโพลี เป็นการศึกษาภาวะการส่งจ่ายพลังงานไฟฟ้าที่ซับซ้อน ๆ ของระบบไฟฟ้ากำลัง การคำนวณโหลดโพลีเป็นการคำนวณเพื่อหาค่าแรงดัน มุมของแรงดัน กำลังจริง และกำลังเสมือนที่ซับซ้อน ๆ ของระบบไฟฟ้ากำลัง

จุดประสงค์ของการคำนวณโหลดโพลี นี้เพื่อจะทำการพิจารณาถึงคุณลักษณะที่เหมาะสมของการผลิตและการส่งจ่ายพลังงานไฟฟ้า เพื่อจ่ายโหลดให้เพียงพอแก่ความต้องการของบัสต่าง ๆ การผลิตกำลังไฟฟ้าดีสูงคำนึงถึง Economic load dispatch และค่าของแรงดันจะถูกควบคุมโดย AVR (Automatic Voltage Regulator)

โหลดโพลี มีประโยชน์ทั้งในแง่การสังเกตเสถียรภาพของระบบ และยังมีประโยชน์ในการวางแผนขยายระบบ อันได้แก่ การต่อระบบไฟฟ้ากำลังหนึ่งกับอีกระบบหนึ่ง การต่อโหลดเพิ่มขึ้น การต่อกับโรงจักรผลิตไฟฟ้าโรงใหม่ และการต่อสายส่งแรงสูงให้ยาวขึ้น เป็นต้น ซึ่งสามารถจำแนกได้ดังนี้คือ

1. ใช้ประกอบในการวางแผนของระบบ (system planning) ดังนี้คือ
  - การไหลของ MW และ Mvar ในบรานช์ของเน็ตเวิร์ก
  - บัสบาร์โวลเตจ
  - ผลในการจัดเตรียมวงจรและการต่อวงจรใหม่เข้ากับโหลดของระบบ (system loading)
  - ผลในการปรับมุมและปรับแรงดันให้สูงขึ้นให้เข้ากับโหลดของระบบ (system loading)
  - ใช้ในการปรับเปลี่ยนขนาดของตัวนำ และแรงดันของระบบ

2. ใช้ประกอบในการพิจารณาการปฏิบัติงานของระบบ (system operation) ดังนี้
  - ผลของการสูญเสียของเครื่องกำเนิดไฟฟ้าและสายส่งของระบบ
  - ช่วยในการเดินเครื่องและการจ่ายโหลดให้มีประสิทธิภาพ
  - ช่วยในการตรวจสอบและสูญเสียในระบบ
3. ใช้เป็นจุดเริ่มต้นในการศึกษาเรื่องอื่นได้ ดังนี้
  - ศึกษาถึงการป้องกันระบบทั้งกระแสและแรงดัน โดยแรงดันจะถูกตรวจสอบด้วย protection relays และกระแสจะถูกอินเตอร์รัพท์ โดยเซอร์กิตเบรกเกอร์
  - ศึกษาถึงเสถียรภาพของระบบ ในการหาเวลาตอบสนอง (time response) ของระบบไฟฟ้ากำลังที่มี

### มีต่อฟอลท์ (fault)

การแก้ปัญหาโหลดไฟลต์นี้สามารถคำนวณได้หลายวิธี เช่น

1. Gauss-Saidel method
2. Newton-Raphson method
3. Fast decouple method

โดยที่แต่ละวิธีจะมีขั้นตอน วิธีการคำนวณ และข้อกำหนดต่าง ๆ ที่แตกต่างกัน ส่งผลให้แต่ละวิธีจะมีอัตราการคำนวณช้าเร็วไม่เท่ากันและความซับซ้อนของสมการที่ใช้ในการคำนวณแตกต่างกันด้วย อย่างไรก็ตาม แต่ละวิธีก็จะมีข้อดีข้อเสียเช่นกัน

ระบบไฟฟ้ากำลังจะทำการส่งจ่ายพลังงานให้แก่โหลด ซึ่งโหลดอาจมีความสมดุลหรือไม่สมดุลก็ได้ โดยขณะที่โหลดมีความสมดุลจะสามารถทำการวิเคราะห์ระบบไฟฟ้ากำลังโดยใช้การคำนวณโหลดไฟลต์แบบเฟสเดียว ส่วนในกรณีที่โหลดมีความไม่สมดุลจะทำการวิเคราะห์ระบบไฟฟ้ากำลังโดยใช้การคำนวณโหลดไฟลต์แบบสามเฟส ซึ่งการคำนวณโหลดไฟลต์แบบเฟสเดียวจะใช้ในการวางแผนเกี่ยวกับระบบไฟฟ้ากำลัง ส่วนการคำนวณโหลดไฟลต์แบบสามเฟสจะใช้ประกอบการปฏิบัติงานของระบบไฟฟ้ากำลังเนื่องจากในการปฏิบัติงานระบบไฟฟ้ากำลังจะไม่สามารถควบคุมให้โหลดมีความสมดุลได้เพราะเป็นผลทางเศรษฐศาสตร์

## บทที่ 2

### การหาผลเฉลยของระบบสมการเชิงเส้น

ในการศึกษา Load flow จะพบว่าต้องมีการหาผลเฉลยของสมการเชิงเส้น เพื่อนำไปสู่การคำนวณหาสิ่งที่ต้องการต่อไป

การหาผลเฉลยของระบบสมการเชิงเส้น สามารถกระทำได้หลายวิธี ตัวอย่างเช่น

1. Direct Method
2. Guass Elimination Method
3. Guass Jordan Method

#### Guass Jordan Method

วิธีนี้เป็นวิธีที่พัฒนามาจากวิธีของ Guass Elimination Method ซึ่งจะแตกต่างกันตรงที่วิธี Guass Jordan ไม่ต้องมีการแทนค่ากลับ (Back Substitution) เหมือนเช่นใน Guass Elimination Method ทำให้มีความรวดเร็วในการหาคำตอบมากกว่า

จากระบบสมการเชิงเส้น

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = Y_1 \quad (2.1)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = Y_2 \quad (2.2)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = Y_3 \quad (2.3)$$

จากระบบสมการเชิงเส้น สามารถทำการหาผลเฉลยของสมการได้ตามวิธีของ Guass Jordan ได้ตามขั้นตอนดังนี้

#### Step 1

ทำการหารสมการที่ 2.1 ด้วยส.ป.ส. ของ  $x_1$  ก็คือ  $a_{11}$  จะได้

$$x_1 + a_{12}x_2 + a_{13}x_3 = Y_1 \quad (2.4)$$

$$a_{11} \quad a_{11} \quad a_{11}$$

### Step2

ทำการคูณสมการที่ (2.4) ด้วย ส.ป.ส. ของ  $x_1$  ในสมการโดยก่เว้นสมการที่ 2.1

$$a_{21}x_1 + a_{21}(a_{12})x_2 + a_{21}(a_{13})x_3 = a_{21}(Y_1) \quad (2.5)$$

$$a_{11} \quad a_{11} \quad a_{11}$$

$$a_{31}x_1 + a_{31}(a_{12})x_2 + a_{31}(a_{13})x_3 = a_{31}(Y_1) \quad (2.6)$$

$$a_{11} \quad a_{11} \quad a_{11}$$

### Step3

นำสมการที่ (2.5) และ (2.6) ไปลบกับสมการที่ (2.2) และ (2.3) เพื่อกำจัดส.ป.ส. ของ  $x_1$  ให้มีค่าเป็นศูนย์ จะได้ระบบสมการใหม่เป็น

$$x_1 + b_{12}x_2 + b_{13}x_3 = Z_1 \quad (2.7)$$

$$c_{22}x_2 + c_{23}x_3 = Z_2 \quad (2.8)$$

$$d_{32}x_2 + d_{33}x_3 = Z_3 \quad (2.9)$$

ซึ่งเป็นระบบสมการ  $m$  สมการและตัวไม่รู้ค่า  $n$  ตัว หรือ เขียนอยู่ในรูป

### Step4

ทำตาม Step 1 ถึง Step 3 เพื่อกำจัด ส.ป.ส. ของ  $x_2$  ในสมการที่ (2.7) และ (2.9) ให้เป็นศูนย์ ในกรณีของ  $x_2$  ก็เช่นเดียวกัน ทำตาม Step 1 ถึง Step 3 เพื่อกำจัด ส.ป.ส. ในสมการที่ (2.7) และ (2.8)

### Step5

จะได้ระบบสมการเชิงเส้นใหม่คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$x_1 + 0x_2 + 0x_3 = M_1 \quad (2.10)$$

$$0x_1 + x_2 + 0x_3 = M_2 \quad (2.11)$$

$$0x_1 + 0x_2 + x_3 = M_3 \quad (2.12)$$

ดังนั้นจะได้ผลเฉลยของสมการ คือ

$$x_1 = M_1$$

$$x_2 = M_2$$

$$x_3 = M_3$$



### บทที่ 3

#### การศึกษาโหลดโฟลว์

##### 3.1 การแยกประเภทบัสในระบบไฟฟ้ากำลัง

ในการคำนวณเกี่ยวกับ load flow เรานิยามที่จะแบ่งบัสในระบบไฟฟ้ากำลังออกเป็น 3 บัสหลัก ๆ ดังนี้

##### 1. Swing Bus หรือ Slack Bus

Swing bus หรือ slack bus เป็นบัสที่มีเครื่องกำเนิดไฟฟ้าต่ออยู่ เป็นบัสที่ควบคุมให้ voltage magnitude และ phase angle ( $V, \delta$ ) คงที่ตลอดเวลาโดยทั่วไปจะให้ ขนาด  $V = 1.0$  p.u. และ phase angle  $= 0^\circ$  เสมอ ในการส่งกำลังไฟฟ้า เราจะไม่ทราบการสูญเสียในสายส่งที่แน่นอนซึ่งจะขึ้นอยู่กับ  $V$  และ phase angle ของบัสนั้น ๆ ดังนั้น เราจะใช้ swing bus เป็นตัวเปรียบเทียบกับ bus อื่น ๆ ว่า ขนาด voltage และ phase angle ของบัสอื่น ๆ ว่าแตกต่างจาก swing bus อย่างไร การกำหนดให้  $V$  และ phase ของ swing bus มีค่าเท่ากับ 1.0 / เราเรียกว่า voltage flat profile

##### 2. Load Bus

load bus เป็นบัสที่มีโหลดต่ออยู่ ดังนั้นบัสนี้เป็นบัสที่กำหนดค่า real power และ reactive power ( $P, Q$ ) load flow equation จะคำนวณหาขนาด voltage ( $V$ ) และ phase angle ที่บัสนี้ออกมา

##### 3. Voltage Control Bus

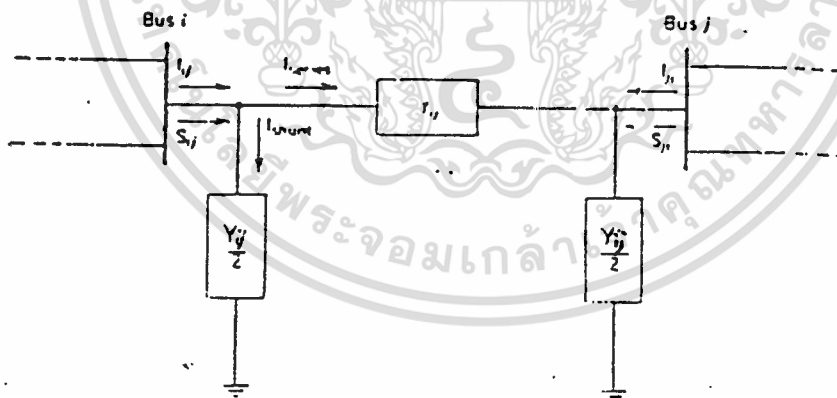
voltage control bus เป็นบัสที่มีเครื่องกำเนิดไฟฟ้าต่ออยู่ตัวเดียว หรืออาจมีโหลดต่ออยู่ด้วยก็ได้ เป็นบัสที่กำหนดค่า voltage และ real power ( $P, V$ ) ในการแก้ปัญหา load flow จะได้ว่า reactive power และ phase angle ออกมา

บัส	ค่าที่กำหนดมาให้	ค่าที่ต้องคำนวณ
slack bus	$V$ $\delta$	$P$ $Q$
load bus	$P$ $Q$	$V$ $\delta$
voltage control bus	$P$ $V$	$Q$ $\delta$

### 3.2 สมการการไหลของกำลังไฟฟ้าในสายส่ง (Line Flow Computation)

หลังจากจบการอิตเทอเรชันแล้ว ได้ค่าแรงดันที่บัสต่าง ๆ แล้ว จึงทำการคำนวณหา กำลังไฟฟ้าที่ไหลเข้า-ออกจากรบัส

จากรูป พิจารณาสายส่งที่เชื่อมต่อระหว่างบัส  $i$  และ  $j$



รูป 3.1 แสดง Power flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_{ij} = I_{series} + I_{shunt} \quad (3.2.1)$$

$$I_{series} = (V_i - V_j) Y_{ij} \quad (3.2.2)$$

$$I_{shunt} = V_i \frac{Y_{ij}}{2} \quad (3.2.3)$$

ดังนั้น

$$I_{ij} = (V_i - V_j) Y_{ij} + V_i \frac{Y_{ij}}{2} \quad (3.2.4)$$

เมื่อ  $Y_{ij}$  = admittance of line ij

$Y'_{ij}$  = total line-charging admittance

ถ้า real และ reactive power ที่ไหลจากบัส i ไป j สามารถแสดงได้ดังนี้

$$S_{ij} = P_{ij} + jQ_{ij} = V_i I_{ij}^* \quad (3.2.5)$$

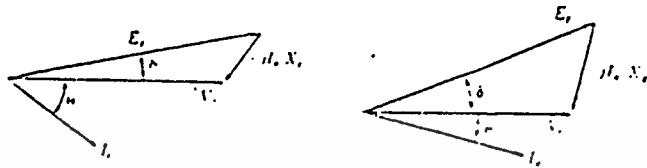
โดยการแทน (3.2.4) ใน (3.2.5)

$$S_{ij} = P_{ij} + jQ_{ij} = V_i (V_i^* - V_j^*) Y_{ij}^* + V_i V_i^* \left( \frac{Y_{ij}}{2} \right)^*$$

เมื่อ  $S_{ij}$  = Line Power Flow จาก bus i ไป bus j

### 3.3 การควบคุมกำลังไฟฟ้าในระบบโครงข่าย

จากรูปวงจรที่แสดงไว้ แสดงถึงการส่งจ่ายกำลังไฟฟ้าจาก synchronous generator ผ่านสายส่งไปยังระบบไฟฟ้ากำลังอย่างง่าย ๆ เราสามารถหาความสัมพันธ์ต่าง ๆ ได้ดังนี้



รูป 3.2 การควบคุมกำลังไฟฟ้า

$$\begin{aligned}
 I &= \frac{E_s / \delta - V / \theta}{Z / \theta} \\
 &= \frac{E_s / \delta - \theta - V / \theta}{Z} \\
 S_A &= E_s \cdot I \\
 &= E_s / \delta \left( \frac{E_s / \delta - \theta - V / \theta}{Z} \right) \\
 &= E_s / \delta \left( \frac{E_s / \delta \pm \theta - V / \theta}{Z} \right)
 \end{aligned}$$

$$= \frac{E_s^2}{Z} \cos \theta - \frac{E_s V_i}{Z} \cos(\theta + \delta)$$

ดังนั้น

$$P_A = \frac{E_s^2}{Z} \cos \theta - \frac{E_s V_i}{Z} \cos(\theta + \delta)$$

$$Q_A = \frac{E_s^2}{Z} \sin \theta - \frac{E_s V_i}{Z} \sin(\theta + \delta)$$

ในทำนองเดียวกัน

$$\begin{aligned} S_B &= \frac{V_i}{Z} \angle \theta ( \frac{V_i}{Z} \angle -\theta - \frac{E_s}{Z} \angle \delta - \theta ) \\ &= \frac{V_i^2}{Z} \cos \theta - \frac{E_s V_i}{Z} \cos(\theta - \delta) \end{aligned}$$

ดังนั้น

$$P_B = \frac{V_i^2}{Z} \cos \theta - \frac{V_i E_s}{Z} \cos(\theta - \delta)$$

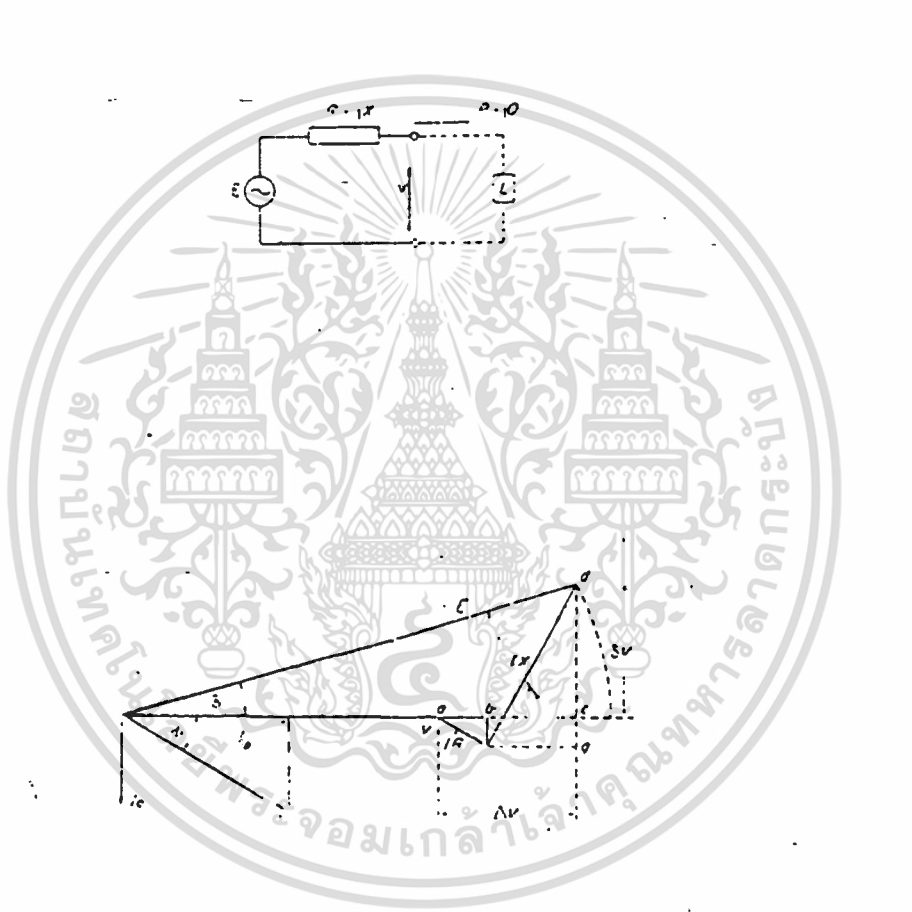
$$Q_B = \frac{V_i^2}{Z} \sin \theta - \frac{V_i E_s}{Z} \sin(\theta - \delta)$$

ถ้า power output ที่จ่ายไปยัง B จะสูงสุดเมื่อ  $\cos(\theta - \delta) = 1$  หรือ  $\theta = \delta$

### 3.4 การควบคุมระบบไฟฟ้ากำลัง

#### Control Voltage and Reactive Power

จากลักษณะ simple transmission line ดังในรูป เราสามารถคำนวณหาค่าความสัมพันธ์ระหว่าง  $E, V$ , ได้ดังนี้



รูป 3.3 เฟสเซอร์ไดอะแกรม สำหรับสายส่งกำลังผ่านอินพีแดนซ์ที่ต่ออนุกรม

*series impedance*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 3.3

$$\begin{aligned}
 E^2 &= (V + \Delta V)^2 + \delta V^2 \\
 &= (V + RI \cos\theta + XI \sin\theta)^2 + XI \cos\theta - RI \sin\theta)^2 \\
 E^2 &= (V + RP + XQ)^2 + (XP - RQ)^2 \\
 &\quad \quad \quad V \quad V \quad \quad V \quad \quad V
 \end{aligned}$$

ดังนั้น

$$\begin{aligned}
 \Delta V &= RP + XQ \\
 &\quad \quad \quad V \quad \quad V
 \end{aligned}$$

ถ้า generated bus จ่ายค่า real power คงที่ หรือค่า  $R = 0$  (ความต้านทานในสายส่งมีค่าเท่ากับศูนย์) จะได้ว่า การเปลี่ยนแปลงของ voltage จะขึ้นอยู่กับค่า reactive power และจาก phasor diagram และ สมการข้างต้น จะได้

$$\begin{aligned}
 \delta V &= XP - RQ \\
 &\quad \quad \quad V \quad \quad V
 \end{aligned}$$

ถ้า  $R$  มีค่าน้อยมาก

$$\delta V \propto Q$$

การเปลี่ยนแปลงของ rotor ใน synchronous generator จะขึ้นกับ real power และมีผลต่อความถี่ด้วย

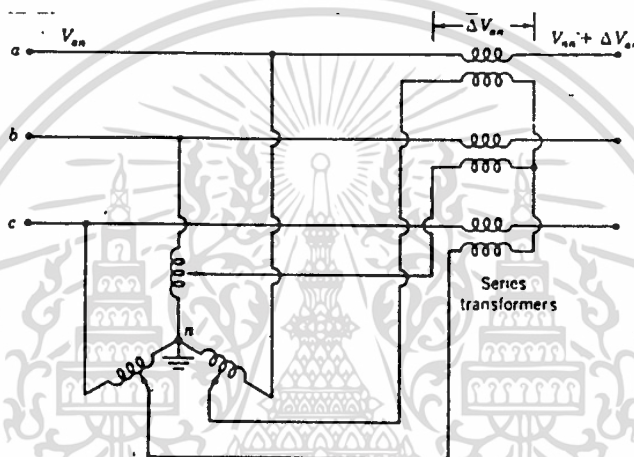
### Control By Transformer

ในการปฏิบัติงานในระบบไฟฟ้ากำลัง เมื่อแรงดันด้าน primary ของหม้อแปลงต่ำลงไปต่ำกว่าเกณฑ์เนื่องจากภาวะไฟฟ้าทางด้าน secondary ต้องการกระแสมาก จึงมีความ จำเป็นที่ต้องเปลี่ยนจำนวนรอบของขดลวดด้าน primary โดยใช้หม้อแปลงที่เปลี่ยนจำนวนรอบของขดลวดได้ (tap-changing transformer) ในการคำนวณโหลดโฟลว์ เมื่อคำนวณแล้วแรงดันที่บัสยังไม่เป็นที่ต้องการ หรือไม่มีโอกาสขงการการอิทธิพลอื่น โปรแกรมจะทำการเปลี่ยน จำนวนรอบของขดลวด แล้วคำนวณใหม่จนกว่าจะได้แรงดันตามที่ต้องการ

หม้อแปลงดังกล่าวนี้ สามารถใช้ควบคุมได้ทั้ง real power และ reactive power หม้อแปลงที่ใช้ในการปรับขนาด voltage ในช่วงแคบ ๆ จะมีย่านอยู่ในช่วง  $\pm 10\%$  และยังมีแบบที่ใช้ shift phase angle ของ line voltage และมีบางแบบที่ปรับได้ทั้งขนาด voltage และ phase angle

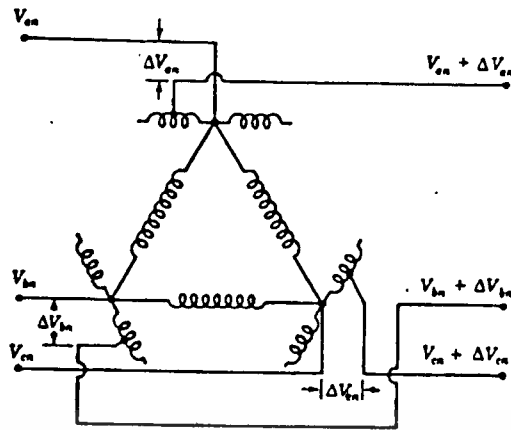
### Regulating Transformer

เป็นหม้อแปลงที่มีทั้งแบบควบคุม voltage magnitude และ phase-angle ตามรูป ส่วนมากจะใช้เพื่อปรับ voltage ในย่านแคบ ๆ มากกว่าจะใช้เปลี่ยนระดับ voltage



รูป 3.4 หม้อแปลงเรกูลเลตติ้ง สำหรับการควบคุมขนาดแรงดัน

จาก phasor diagram เราสามารถอธิบายการ shift phase ของหม้อแปลง ได้ดังนี้ แต่ละขดลวดในหม้อแปลงถูกพันอยู่แกนเหล็กเดี่ยว voltage ที่เหนี่ยวนำขึ้นจะเป็นมุม  $90^\circ$  out of phase กับจุดกึ่งกลางของแท่ง ดังนั้นเฟส voltage  $V_{an}$  จะมีค่าเพิ่มขึ้นด้วยค่า  $V_{an}$  ซึ่งอาจจะ inphase หรือ  $180^\circ$  out of phase กับ  $V_{an}$  จากรูปจะพบว่า line voltage ทั้งสามมี phase shift ออกไปแต่ขนาดมีการเปลี่ยนแปลงน้อยมาก



รูป 3.5 หม้อแปลงแรงดันสูง สำหรับการควบคุมมุมเฟส



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 วิธีการคำนวณแบบ Gauss

การหาค่าตอบของโหนดโวลต์ด้วยวิธีนี้ เริ่มต้นด้วยการสมมติว่า voltages ทุก ๆ บัส นอกจาก slack bus ซึ่งได้กำหนดค่า voltage และกำหนดให้คงที่ เราสามารถคำนวณค่ากระแสสำหรับบัสทุกบัสนอกจาก slack บัส จากสมการเหล่านี้

$$\begin{aligned} I_p &= P_p - jQ_p, \quad p = 1, 2, 3, \dots, n \\ V_p & \quad p = s \end{aligned} \quad (3.5.1)$$

เมื่อ  $n$  เป็นจำนวนบัสของ Network และจาก

$$I_p = \frac{S_p^*}{V_p} \quad (S_p = V_p I_p^*) \quad (3.5.2)$$

และ

$$I_{bus} = Y_{bus} V_{bus}$$

กำหนดให้ ground เป็น reference bus จะมีสมการอยู่  $n-1$  สมการ ที่เขียนได้ในรูปแบบดังนี้

$$V_p = \frac{1}{Y_{pp}} \left( I_p - \sum_{q=1}^{q \neq p} Y_{pq} V_q \right) \quad (3.5.3)$$

ถ้า bus currents คำนวณจากสมการ (3.5.1) แทนค่า slack bus voltage และ bus voltage ที่กำหนดค่าประมาณลงในสมการ (3.5.3) เพื่อคำนวณค่า bus voltage ชุดใหม่และค่า bus voltage ชุดใหม่ นี้ จะนำไปแทนในสมการ (3.5.1) เพื่อคำนวณหาค่ากระแสชุดใหม่ เพื่อนำไปคำนวณในสมการ (3.5.3) และจะทำซ้ำไปเรื่อย ๆ จนกว่า ค่าความเปลี่ยนแปลงของ bus voltages ทุกแห่ง อยู่ในย่านที่ยอมรับได้ ภายหลังการคำนวณหาค่า voltage แล้วจะทำให้ คำนวณค่า power ของ slack bus และ line flow ได้

สมการของ Network (3.5.3) และ bus loading equation (3.5.1) สามารถ นำมารวมกันได้ เป็น

n

$$V_p = \frac{1}{Y_{pp}} (P_p - jQ_p - \sum_{q=1, q \neq p}^n Y_{pq} V_q) \quad p = 1, 2, \dots, n$$

(3.5.4)

สมการของ load flow เหล่านี้เป็นสมการแบบ nonlinear ซึ่งสามารถแก้โดยใช้วิธี iteration ในการคำนวณ เราจะลดเวลาที่ใช้ให้มากที่สุด โดยลดขั้นตอนทางคณิตศาสตร์ โดยให้

$$\frac{1}{Y_{pp}} = L_p$$

$$Y_{pq}$$

เขียนสมการ (3.5.4) ใหม่ได้ดังนี้

$$V_p = (P_p - jQ_p) L_p - \sum_{q=1, q \neq p}^n Y_{pq} L_p V_q \quad p = 1, 2, \dots, n \quad (3.5.5)$$

$$\text{ให้ } (P_p - jQ_p) L_p = K L_p$$

$$\text{และ } Y_{pq} L_p = Y L_{pq}$$

และ bus voltage equation (3.5.5) จะกลายเป็น

$$V_p = K L_p - \sum_{q=1, q \neq p}^n Y L_{pq} V_q \quad p = 1, 2, \dots, n \quad (3.5.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนของ การศึกษา load flow โดยสมมุติว่าเป็นระบบสมดุล และใช้วงจรสมมูลแบบเฟส เดียวแบบ positive sequence network เพราะว่าไม่มี mutual coupling

สำหรับวิธี Gauss Iteration สามารถพิจารณาได้จากโฟลว์ชาร์ท ดังรูป 3.6

### 3.6 Gauss-Seidel iterative method using $Y_{bus}$

สมการ bus voltage (3.5.6) สามารถหาคำตอบโดย Gauss-Seidel iterative method โดยวิธีนี้ค่า voltage  $V_p^{k+1}$  ที่คำนวณได้ใหม่ จากถูกแทนด้วย  $V_p^k$  และใช้ในสมการลำดับถัดมา สำหรับวิธี Gauss-Seidel Iteration สามารถพิจารณาจากโฟลว์ชาร์ทดังรูปที่ 3.7



### 3.7 วิธี Newton-Raphson (Newton-Raphson method)

วิธีนิวตัน-ราฟสัน เป็นวิธีที่เร็วและมีประสิทธิภาพมากในการคำนวณโหนดโพลาร์ หลักการของวิธีนิวตัน-ราฟสัน ได้จากการทออนุกรมของเทย์เลอร์ (Taylor's series expansion) สำหรับฟังก์ชันที่มีตัวแปรตั้งแต่สองตัวขึ้นไป สมมติมีฟังก์ชันซึ่งมีตัวแปร  $x_1$  กับ  $x_2$

$$k_1 = f_1(x_1, x_2) \quad (3.7.1)$$

$$k_2 = f_2(x_1, x_2) \quad (3.7.2)$$

โดย  $k_1$  และ  $k_2$  เป็นค่าคงที่ ถ้าคาดการณ์ว่า  $x_{1(0)}$  และ  $x_{2(0)}$  มีค่าใกล้เคียงกับคำตอบของทั้งสองสมการและกำหนดว่า  $x_{1(0)} + \Delta x_{1(0)}$  กับ  $x_{2(0)} + \Delta x_{2(0)}$  เป็นคำตอบของทั้งสองสมการ นั้นหมายความว่า ถ้าหา  $x_{1(0)}$  และ  $x_{2(0)}$  ได้ก็จะได้คำตอบของสมการทั้งสอง ดังนั้นเขียนสมการทั้งสองได้ดังนี้

$$k_1 = f_1(x_1, x_2) = f_1(x_{1(0)} + \Delta x_{1(0)}, x_{2(0)} + \Delta x_{2(0)})$$

$$k_2 = f_2(x_1, x_2) = f_2(x_{1(0)} + \Delta x_{1(0)}, x_{2(0)} + \Delta x_{2(0)})$$

เมื่อกระจายอนุกรมของเทย์เลอร์

$$k_1 = f_1(x_{1(0)}, x_{2(0)}) + x_{1(0)} \frac{\partial f_1}{\partial x_1} + x_{2(0)} \frac{\partial f_1}{\partial x_2} + \dots \quad (3.7.3)$$

$$k_2 = f_2(x_{1(0)}, x_{2(0)}) + x_{1(0)} \frac{\partial f_2}{\partial x_1} + x_{2(0)} \frac{\partial f_2}{\partial x_2} + \dots \quad (3.7.4)$$

พหุคูณเชิงอนุกรม (partial derivative) ที่มีอันดับมากกว่า 1 ไม่ได้เขียนไว้ในสมการที่ (3.7.3) และ (3.7.4) ซึ่งถ้าตัดพหุคูณเชิงอนุกรมที่มีอันดับมากกว่า 1 ออกจะเขียนสมการที่ (3.7.3) และ (3.7.4) ในรูปแบบเมตริกซ์ได้

$$\begin{vmatrix} k_1 - f_1(x_{1(0)}, x_{2(0)}) \\ k_2 - f_2(x_{1(0)}, x_{2(0)}) \end{vmatrix} = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{vmatrix} \begin{vmatrix} \Delta x_{1(0)} \\ \Delta x_{2(0)} \end{vmatrix} \quad (3.7.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมตริกซ์จตุรัสของพหุเชิงเส้นเดอริเวทีฟ มีชื่อเรียกว่า "จาโคเบียนเมตริกซ์" (Jacobian matrix)  $[J]$  ค่าสมาชิกใน  $[J]$  หาได้จากการแทนค่า  $x_{1(0)}$  และ  $x_{2(0)}$  ในพหุเชิงเส้นเดอริเวทีฟ ถ้า  $x_{1(0)}$  และ  $x_{2(0)}$  ไม่ใช่คำตอบของสมการที่ (1) และ (2) ดังนี้

$$k_{1(0)} = k_1 - f_1(x_{1(0)}, x_{2(0)})$$

$$k_{2(0)} = k_2 - f_2(x_{1(0)}, x_{2(0)})$$

นำไปแทนค่าในสมการที่ (3.7.5) ได้

$$\begin{bmatrix} k_{1(0)} \\ k_{2(0)} \end{bmatrix} = [J] \begin{bmatrix} x_{1(0)} \\ x_{2(0)} \end{bmatrix}$$

โดยการหาอินเวิร์สของ  $[J]$  จะทำให้หาค่า  $x_{1(0)}$  และ  $x_{2(0)}$  ได้ เนื่องจากมีการตัดบางส่วนของการกระจายอนุกรมของเทย์เลอร์ ค่า  $x_{1(0)}$  และ  $x_{2(0)}$  ที่นำไปบวกกับ  $x_{1(0)}$  ตามลำดับย่อมไม่ใช่คำตอบจึงต้องทำการคำนวณซ้ำอีก โดยใช้ค่า  $x_1$  และ  $x_2$  ใหม่ดังนี้

$$x_{1(1)} = x_{1(0)} + \Delta x_{1(0)}$$

$$x_{2(1)} = x_{2(0)} + \Delta x_{2(0)}$$

จะทำการคำนวณซ้ำจนกว่า  $x_k$  และ  $x_{(k-1)}$  เข้าใกล้กันมาก คล้ายๆ กับการอิตเทอเรชันนั่นเอง

**การประยุกต์วิธีนิวตัน-ราฟสัน ในการแก้ปัญหาโหลดฟลิว**

สมการสำหรับคำนวณกำลังที่บัส  $p$  หาได้จาก

$$\begin{aligned} S_p &= V_p I_p^* \\ &= P_p + jQ_p \end{aligned}$$

ดังนั้น

$$P_p - jQ_p = V_p I_p^* \quad (3.7.6)$$

แต่เรามีสมการของกระแสคือ

$$I_p = \sum_{q=1}^n Y_{pq} V_q$$

แทนค่าในสมการ (3.7.6) จะได้

$$P_p - jQ_p = V_p^* \sum_{q=1}^n Y_{pq} V_q \quad (3.7.7)$$

กำหนดให้  $V_p = e_p + jf_p$  และ  $Y_{pq} = G_{pq} - jB_{pq}$  สมการ (3.7.7) จะกลายเป็น

$$P_p - jQ_p = (e_p - jf_p) \sum_{q=1}^n (G_{pq} - jB_{pq}) (e_q + jf_q) \quad (3.7.8)$$

แยกส่วนจริงและส่วนจินตภาพออกจากกัน

$$P_p = \sum_{q=1}^n e_p (e_q G_{pq} + f_q B_{pq}) + f_p (f_q G_{pq} - e_q B_{pq})$$

$$Q_p = \sum_{q=1}^n f_p (e_q G_{pq} + f_q B_{pq}) - e_p (f_q G_{pq} - e_q B_{pq}) \quad (3.7.9)$$

วิธีนิวตัน-ราฟสัน สามารถใช้จัดชุดของสมการไม่เชิงเส้นข้างต้น ให้เป็นสมการเชิงเส้นได้ ซึ่งมีการเปลี่ยนแปลงของกำลังจริงและกำลังเสมือนและส่วนประกอบของบัสโวลเตจ ดังนี้

$$\begin{array}{c}
 \left| \begin{array}{c} \Delta P_1 \\ \vdots \\ \Delta P_{n-1} \end{array} \right| = \left| \begin{array}{cccc} \partial P_1 & \dots & \partial P_1 & \partial P_1 & \dots & \partial P_1 \\ \partial^{\circ} 1 & & \partial^{\circ} n-1 & \partial^{\prime} 1 & & \partial^{\prime} n-1 \\ \dots & & \dots & \dots & & \dots \\ \partial P_{n-1} & \dots & \partial P_{n-1} & \partial P_{n-1} & \dots & \partial P_{n-1} \\ \partial^{\circ} 1 & & \partial^{\circ} n-1 & \partial^{\prime} 1 & & \partial^{\prime} n-1 \end{array} \right| \left| \begin{array}{c} \Delta e_1 \\ \vdots \\ \Delta e_{n-1} \end{array} \right| \\
 \\
 \left| \begin{array}{c} \Delta Q_1 \\ \vdots \\ \Delta Q_{n-1} \end{array} \right| = \left| \begin{array}{cccc} \partial Q_1 & \dots & \partial Q_1 & \partial Q_1 & \dots & \partial Q_1 \\ \partial^{\circ} 1 & & \partial^{\circ} n-1 & \partial^{\prime} 1 & & \partial^{\prime} n-1 \\ \dots & & \dots & \dots & & \dots \\ \partial Q_{n-1} & \dots & \partial Q_{n-1} & \partial Q_{n-1} & \dots & \partial Q_{n-1} \\ \partial^{\circ} 1 & & \partial^{\circ} n-1 & \partial^{\prime} 1 & & \partial^{\prime} n-1 \end{array} \right| \left| \begin{array}{c} \Delta f_1 \\ \vdots \\ \Delta f_{n-1} \end{array} \right|
 \end{array} \tag{3.7.10}$$

เมื่อเมตริกซ์สัมประสิทธิ์คือ Jacobian และบัสที่ n เป็น swing bus ในรูปของเมตริกซ์สมการที่ (3.7.10) เขียนได้เป็น

$$\left| \begin{array}{c} \Delta P \\ \vdots \\ \Delta Q \end{array} \right| = \left| \begin{array}{cc} J_1 & J_2 \\ \vdots & \vdots \\ J_3 & J_4 \end{array} \right| \left| \begin{array}{c} \Delta e \\ \vdots \\ \Delta f \end{array} \right|$$

สมการที่ใช้ในการหาสมาชิกของจาโคเบียน สามารถพิสูจน์จากสมการกำลังของบัส ค่ากำลังจริงของสมการที่ (3.7.9) คือ

n

$$P_p = e(e G_{p p} + f B_{p p}) + f(f B_{p p} - e B_{p p}) + \sum_{q=1}^n e(e G_{p q} + f B_{p q}) + f(f G_{p q} - e B_{p q})$$

q = 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

q ≠ p

(3.7.11)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คิฟเฟอเรนเชียล สมาชิก off-diagonal ของ  $J_1$  จะได้

$$\frac{\partial P_p}{\partial c_p} = c_p G_{pq} - f_p B_{pq} \quad q \neq p$$

$$\frac{\partial P_p}{\partial c_{pp}}$$

และ สมาชิก diagonal ของ  $J_1$  คือ

$$\frac{\partial P_p}{\partial c_p} = 2c_p G_{pp} + f_p B_{pp} - f_{pp} B_{pp} + \sum_{q=1, q \neq p}^n (c_q G_{pq} + f_q B_{pq}) \quad (3.7.12)$$

$$\frac{\partial P_p}{\partial c_p}$$

$q=1$

$q \neq p$

อย่างไรก็ตาม สมการกระแสที่ บัส  $p$  คือ

$$I_p = c_p + j d_p = (G_{pp} - j B_{pp})(c_p + j f_p) + \sum_{q=1, q \neq p}^n (G_{pq} - j B_{pq})(c_q + j f_q)$$

$q=1$

$q \neq p$

ที่ซึ่งสามารถแยกเป็นส่วนจริงและส่วนจินตภาพ ได้ดังนี้

$$c_p = c_p G_{pp} + f_p B_{pp} + \sum_{q=1, q \neq p}^n (c_q G_{pq} + f_q B_{pq})$$

$q=1$

$q \neq p$

(3.7.13)

$n$

$$d_p = f_p G_{pp} - c_p B_{pp} + \sum_{q=1, q \neq p}^n (f_q G_{pq} - c_q B_{pq})$$

$q=1$

$q \neq p$

$p = 1, 2, \dots, n-1$

จากสมการ (3.7.11) สมาชิก off-diagonal ของ  $J_2$  คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\partial P_p}{\partial f_q} = e_{p, pq} B_{p, pq} + f_{p, pq} G_{p, pq} \quad , q \neq p$$

$$\frac{\partial f_p}{\partial f_q}$$

และสมาชิก diagonal ของ  $J_2$  คือ

$$\frac{\partial P_p}{\partial f_p} = e_{p, pp} B_{p, pp} + 2f_{p, pp} G_{p, pp} - \sum_{q=1}^n e_{p, qp} B_{p, qp} + (f_{p, pq} G_{p, pq} - e_{p, pq} B_{p, pq}) \quad (3.7.14)$$

$$\frac{\partial f_p}{\partial f_p}$$

$q \neq p$

นำส่วนจินตภาพของกระแสดำเนินการที่ (3.7.13) แทนในสมการที่ (3.7.14) จะได้

$$\frac{\partial P_p}{\partial f_p} = e_{p, pp} B_{p, pp} + f_{p, pp} G_{p, pp} + d_p$$

$$\frac{\partial f_p}{\partial f_p}$$

ค่ากำลังเสมือนจากสมการ (3.7.9) จะเป็น

$$Q_p = f_{p, p} (e_{p, p} G_{p, p} + f_{p, p} B_{p, p}) - e_{p, p} (f_{p, p} G_{p, p} - e_{p, p} B_{p, p}) + \sum_{q=1}^n f_{p, q} (e_{p, q} G_{p, q} + f_{p, q} B_{p, q}) - e_{p, q} (f_{p, q} G_{p, q} - e_{p, q} B_{p, q})$$

$q \neq p$

$p = 1, 2, \dots, n-1$

(3.7.15)

ดิฟเฟอเรนเชียล สมาชิก off-diagonal ของ  $J_3$  คือ

$$\frac{\partial Q_p}{\partial e_p} = e_{p, pq} B_{p, pq} + f_{p, pq} G_{p, pq} \quad p \neq q$$

$$\frac{\partial e_p}{\partial e_p}$$

และสมาชิก diagonal ของ  $J_3$  คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\partial Q_p}{\partial e_p} = f_p G_{pp} - f_p G_{pp} + 2e_p B_{pp} - \sum_{q=1}^n (f_q G_{pq} - e_q B_{pq}) \quad (3.7.16)$$

$$\frac{\partial e_p}{\partial f_p}$$

q=1

q≠p

นำส่วนจินตภาพของกระแสจากสมการ (3.7.12) แทนลงในสมการ (3.7.16) จะได้

$$\frac{\partial Q_p}{\partial e_p} = e_p B_{pp} + f_p G_{pp} - d_p$$

$$\frac{\partial e_p}{\partial f_p}$$

จากสมการที่ (3.7.15) สมาชิกของ off-diagonal ของ  $J_4$  คือ

$$\frac{\partial Q_p}{\partial f_q} = e_p G_{pq} + f_p B_{pq} \quad q \neq p$$

$$\frac{\partial f_q}{\partial e_p}$$

และสมาชิก diagonal ของ  $J_4$  คือ

$$\frac{\partial Q_p}{\partial f_p} = e_p G_{pp} + 2f_p B_{pp} - \sum_{q=1}^n e_q G_{pq} + (e_q G_{pq} + f_q B_{pq}) \quad (3.7.17)$$

$$\frac{\partial f_p}{\partial e_p}$$

นำส่วนจริงของกระแสจากสมการ (3.7.13) แทนในสมการ (3.7.17) จะได้

$$\frac{\partial Q_p}{\partial f_p} = -e_p G_{pp} + f_p B_{pp} + C_p$$

$$\frac{\partial f_p}{\partial e_p}$$

กำหนดค่าเริ่มต้นของบัสโวลเตจ ค่ากำลังจริงและกำลังปรากฏจะคำนวณได้จากสมการ (3.7.9) การเปลี่ยนแปลงของกำลังที่บัส ได้จากผลต่างค่าที่กำหนด (scheduled) กับค่าที่คำนวณได้

$$\Delta P_p^k = P_{p(\text{scheduled})}^k - P_p^k$$

$$\Delta Q_p^k = Q_{p(\text{scheduled})}^k - Q_p^k \quad p = 1, 2, \dots, n-1$$

ค่าประมาณของโวลเตจที่บัสและกำลังจะใช้ในการคำนวณกระแสบัส เพื่อใช้ในการหาค่าสมาชิกของจาโคเบียน สมการที่ (3.7.10) สามารถให้ค่า  $\Delta e_p$  และ  $\Delta f_p$ ,  $p = 1, 2, \dots, n-1$  โดยทางตรงหรือวิธีทำซ้ำ ดังนั้นค่าประมาณอันใหม่ของบัสโวลเตจจะเป็น

$$e_p^{k+1} = e_p^k + \Delta e_p^k$$

$$f_p^{k+1} = f_p^k + \Delta f_p^k$$

ทำซ้ำจนกว่า  $\Delta P_p^k$  และ  $\Delta Q_p^k$  ของทุกบัสอยู่ในช่วงที่ยอมรับได้

วิธีนิวตัน-ราฟสัน ยังสามารถประยุกต์เพื่อใช้หาค่าตอบของโพลดิโพลว์เมื่อสมการอยู่ในรูปฟังก์ชันเชิงซ้อนในระบบพิกัดเชิงขั้ว

$$V_p = V_p e^{j\theta_p} \text{ และ } Y_{pq} = Y_{pq} e^{-j\theta_{pq}}$$

แทนค่าลงในสมการ(3.7.7) ดังนั้นกำลังที่บัส  $p$  คือ

$$P_p - jQ_p = \sum_{q=1}^n |V_p V_q Y_{pq}| e^{-j(\theta_{pq} + \delta_p - \delta_q)}$$

เพราะว่า

$$e^{-j(\theta_{pq} + \delta_p - \delta_q)} = \cos(\theta_{pq} + \delta_p - \delta_q) - j\sin(\theta_{pq} + \delta_p - \delta_q)$$

ดังนั้นส่วนจริงและส่วนจินตภาพของกำลังจะเป็น

$$P_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \cos(\theta_{pq} + \delta_p - \delta_q)$$

$$Q_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \sin(\theta_{pq} + \delta_p - \delta_q)$$

$$P = 1, 2, \dots, n-1$$

$$(3.7.18)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ  $J_1$ :

$$\frac{\partial P_p}{\partial \delta_q} = |V_p V_q Y_{pq}| \sin(\theta_{pq} + \delta_p - \delta_q) \quad q \neq p$$

$$\frac{\partial P_p}{\partial \delta_q} = - \sum_{q=1}^{q \neq p} |V_p V_q Y_{pq}| \sin(\theta_{pq} + \delta_p - \delta_q)$$

สำหรับ  $J_2$ :

$$\frac{\partial P_p}{\partial V_q} = |V_p Y_{pq}| \cos(\theta_{pq} + \delta_p - \delta_q) \quad q \neq p$$

$$\frac{\partial P_p}{\partial V_p} = 2 |V_p Y_{pp}| \cos(\delta_p + \delta_p - \delta_p) + \sum_{q=1}^{q \neq p} |V_p Y_{pq}| \cos(\delta_p + \delta_p - \delta_q)$$

สำหรับ  $J_3$ :

$$\frac{\partial Q_p}{\partial \delta_q} = - |V_p V_q V_{pq}| \cos(\theta_{pq} + \delta_p - \delta_q) \quad q \neq p$$

$$\frac{\partial Q_p}{\partial \delta_q} = - \sum_{q=1}^{q \neq p} |V_p V_q V_{pq}| \cos(\theta_{pq} + \delta_p - \delta_q)$$

สำหรับ  $J_q$ :

$$\frac{\partial Q_p}{\partial V_q} = |V_p Y_{pq}| \sin(\theta_{pq} + \delta_p - \delta_q) \quad q \neq p$$

$$\frac{\partial V_q}{\partial V_p}$$

$$\frac{\partial Q_p}{\partial V_p} = 2|V_p Y_{pp}| \sin \theta_{pp} - \sum_{q=1}^{q \neq p} |V_q Y_{pq}| \sin(\theta_{pq} + \delta_p - \delta_q)$$

$$\frac{\partial V_p}{\partial V_p}$$

$q \neq p$

ดังนั้น สมการที่แสดงความสัมพันธ์ของกำลังที่เปลี่ยนไป กับขนาดของโวลเตจและมุมเฟส สำหรับวิธีนิวตัน-ราฟสันจะได้

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta E \end{bmatrix}$$

การประมาณค่าของวิธีนิวตัน-ราฟสัน

โดยทั่วไป สำหรับการเปลี่ยนแปลงขนาดของบัสโวลเตจที่น้อย ๆ ค่ากำลังจริงที่บัสจะเปลี่ยนแปลงน้อยมาก ทำนองเดียวกันสำหรับการเปลี่ยนแปลงของมุมเฟสที่น้อย ๆ ของบัสโวลเตจค่ากำลังเสมือน จะเปลี่ยนแปลงน้อยมาก ดังนั้น โดยใช้ระบบพิกัดเชิงขั้ว คำตอบของปัญหาโพลดิฟเฟอเรนเชียลสามารถหาโดยสมมติว่าสมาชิกของแมทริกซ์ย่อย  $J_2$  และ  $J_3$  เป็นศูนย์ ดังนั้นจะได้สมการแมทริกซ์ ดังนี้

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & 0 \\ 0 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta E \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำตอบสุดท้ายที่สมบูรณ์ จะหาได้โดยวิธีทำซ้ำเพียงหนึ่งถึงสองครั้งเท่านั้น ในการหาค่าจาโคเบียนเมื่อใช้ระบบพิกัดแกนมุมฉาก คำตอบของโหนดโพล์สามารถหาได้โดยละทิ้งค่าสมาชิก off-diagonal ของเมทริกซ์ย่อย  $J_1, J_2, J_3$  และ  $J_n$  ของจาโคเบียน ซึ่งจะได้สมการแสดงการเปลี่ยนแปลงของกำลังจริงและกำลังเสมือนที่บัส  $P$  ดังนี้

$$\begin{aligned} \Delta P_p &= -\frac{\partial P_p}{\partial e_p} \Delta e_p + \frac{\partial P_p}{\partial f_p} \Delta f_p \\ &= \Delta e_p (e_p G_{pp} - f_p B_{pp} + C_p) + \Delta f_p (e_p B_{pp} + f_p G_{pp} + d_p) \\ \Delta Q_p &= \frac{\partial Q_p}{\partial e_p} \Delta e_p + \frac{\partial Q_p}{\partial f_p} \Delta f_p \\ &= \Delta e_p (e_p B_{pp} + f_p G_{pp} - d_p) + \Delta f_p (-e_p G_{pp} + f_p B_{pp} + C_p) \\ & \quad p = 1, 2, \dots, n-1 \end{aligned}$$

สมการเหล่านี้สามารถหาคำตอบได้โดยใช้วิธี Gauss-Seidel

### 3.8 ทฤษฎีและวิธี Fast Decoupled Load Flow

#### 3.8.1 สมการของ Fast Decoupled Load Flow

การเปลี่ยนแปลงเพียงเล็กน้อยของขนาดแรงดัน ( $v$ ) มีผลกระทบต่อการเปลี่ยนแปลงของกำลังจริง ( $P$ ) น้อยมาก ในทำนองเดียวกัน การเปลี่ยนแปลงเพียงเล็กน้อยของมุมเฟสของแรงดัน ( $\delta$ ) มีผลกระทบต่อการเปลี่ยนแปลงของกำลังเสมือน ( $Q$ ) น้อยมาก จากคุณสมบัติดังกล่าวข้างต้นจึงเกิดหลักการ Decoupled ขึ้นมา จากหลักการดังกล่าวจะสามารถเขียนสมการการคำนวณโหลดโพล์โดยให้  $P, Q, V$  decoupled ซึ่งกันและกันได้ดังนี้

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} \quad (3.8.1)$$

เมื่อเขียนสมการ (3.8.1) แยกกันออกมาจะได้

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H \\ L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} \quad (3.8.2a)$$

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H \\ L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} \quad (3.8.2b)$$

โดยที่บัส  $k$  มี mismatch

$$\Delta P_k = P_k^{sp} - V_k \sum_m V_m (G_{km} \cos \theta_{km} + B_{km} \sin \theta_{km}) \quad (3.8.3a)$$

$$\Delta Q_k = Q_k^{sp} - V_k \sum_m V_m (G_{km} \sin \theta_{km} - B_{km} \cos \theta_{km}) \quad (3.8.3b)$$

เมื่อหารสมการ (3.8.2) ด้วยขนาดแรงดัน

$$\Delta P/V = A \Delta \delta \quad (3.8.4a)$$

$$\Delta Q/V = C \Delta V \quad (3.8.4b)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. ตัดองค์ประกอบซึ่งมีผลต่อ reactive power มากกว่า real power ออกจาก [B] ในสมการ (3.8.5a) ได้แก่ shunt reactor, off-normal in phase transformer tap
2. ตัดองค์ประกอบซึ่งมีผลต่อการ shift ของมุมเฟสของแรงดันออกจาก [B] ในสมการ (3.8.5b) ได้แก่ phase shifter
3. จัดสมการ (3.8.5) ใหม่โดยย้ายค่า V ที่อยู่ซ้ายไปยังด้าน power mismatch และกำหนดค่า V ซึ่งอยู่ทางขวาเป็น 1.0 p.u. เพราะฉะนั้น จะได้สมการของ Fast Decoupled flow คือ

$$\Delta P/V = B' \Delta \delta \quad \text{order}(N-1) \quad (3.8.6a)$$

$$\Delta Q/V = B'' \Delta V \quad \text{order}(N-M) \quad (3.8.6b)$$

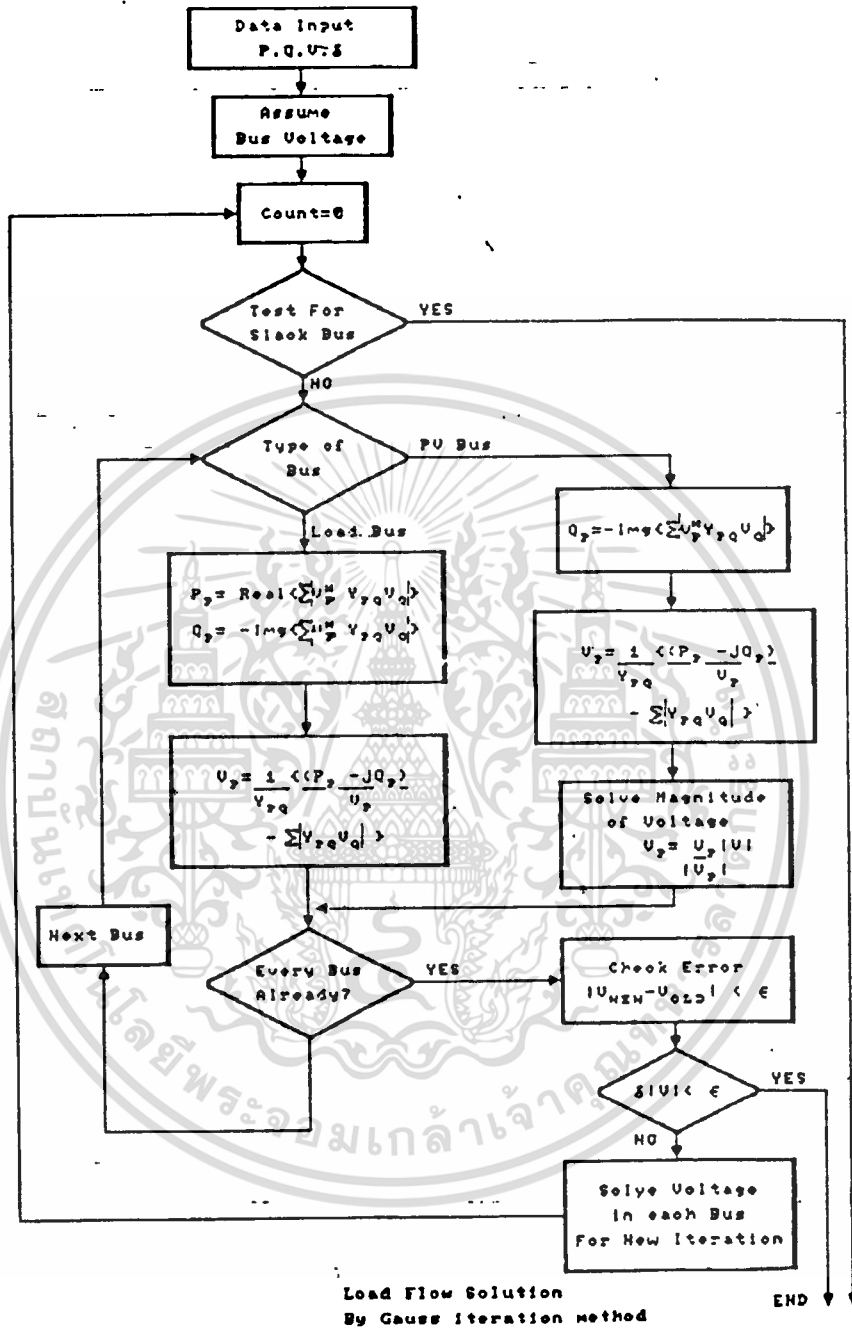
โดยที่

$$B_{kk}' = -1/X_{km}, \quad m \neq k$$

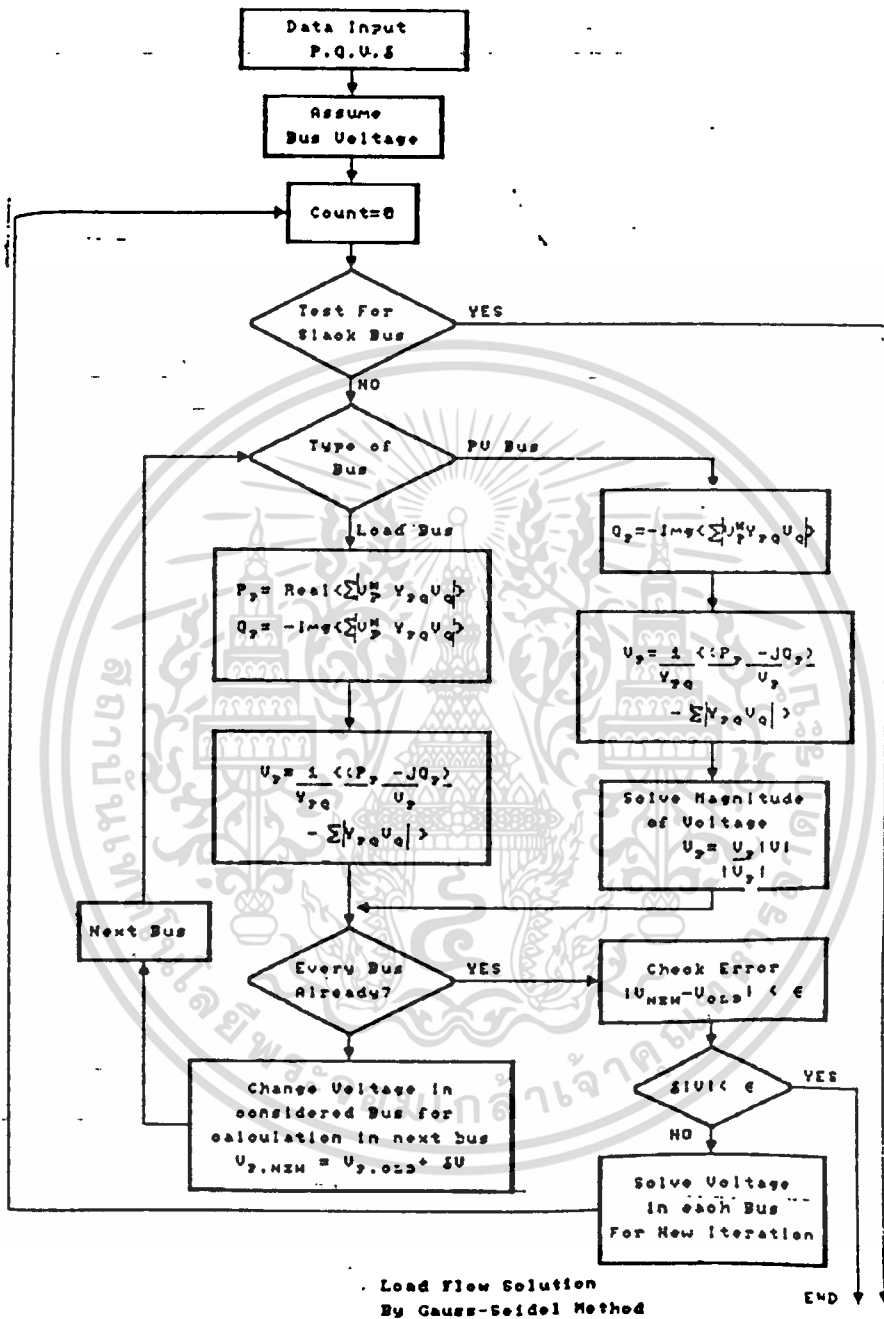
$$B_{km}'' = -B_{km}, \quad m \neq k$$

$$B_{kk}'' = \sum B_{km}$$

Fast Decoupled loadflow เป็นวิธีคำนวณซึ่งมีประสิทธิภาพสูงมาก



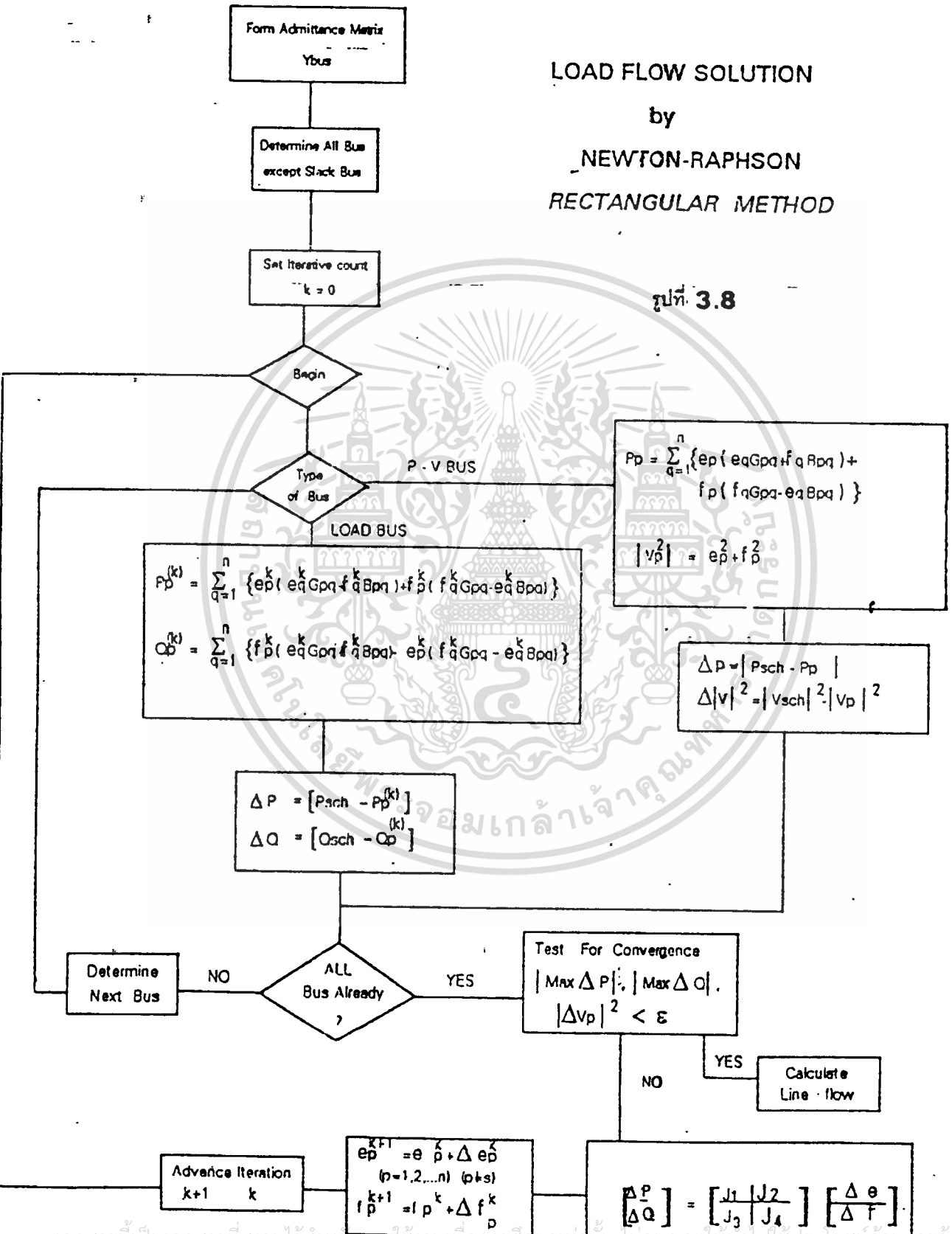
รูปที่ 3.6



รูปที่ 3.7

LOAD FLOW SOLUTION  
by  
NEWTON-RAPHSON  
RECTANGULAR METHOD

รูปที่ 3.8



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะภายในเท่านั้น ไม่ควรเผยแพร่หรือใช้เพื่อการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# LOAD FLOW SOLUTION

by  
NEWTON-RAPHSON  
POLAR METHOD

รูปที่ 3.9

Form Admittance Matrix  
 $Y_{bus}$

Determine All Bus  
except Slack Bus

Set Iterative count  
 $k = 0$

Begin

Type  
of Bus

P - V BUS

$$P_D = \sum_{q=1}^n [V_p V_q Y_{pq}] \cos(\theta_{pq} + \delta_p - \delta_q)$$

LOAD BUS

$$P_D^{(k)} = \sum_{q=1}^n [V_p V_q Y_{pq}] \cos(\theta_{pq} + \delta_p - \delta_q)$$

$$Q_D^{(k)} = \sum_{q=1}^n [V_p V_q Y_{pq}] \sin(\theta_{pq} + \delta_p - \delta_q)$$

$$\Delta P = [P_{sch} - P_p]$$

$$\Delta P = [P_{sch} - P_p^{(k)}]$$

$$\Delta Q = [Q_{sch} - Q_p^{(k)}]$$

Determine  
Next Bus

NO

ALL  
Bus Already  
?

YES

Test For Convergence  
 $|\text{Max } \Delta P|, |\text{Max } \Delta Q| < \epsilon$

NO

YES

Calculate  
Line flow

Advance Iteration  
 $k+1$        $k$

$$\delta_p^{k+1} = \delta_p^k + \Delta \delta_p^k$$

$(p=1,2,\dots,n) (p \neq s)$

$$V_p^{k+1} = V_p^k + \Delta V_p^k$$

Solve For Correction Magnitude and angle

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}$$

วิเคราะห์ระบบ 3 เฟส ( Three phase system analysis )

## การวิเคราะห์แกนอ้างอิง ( Frame of reference analysis )

โดยปกติแล้วเรามักจะใช้ Symmetrical component transformation ซึ่งเป็นการวิเคราะห์ ทางคณิตศาสตร์ซึ่งนำเสนอโดย Fortescue ซึ่งกล่าวไว้ว่า "ระบบที่มี n เวกเตอร์หรือปริมาณใดๆ ที่สามารถที่จะแยกเป็นอิสระจากกันได้ เมื่อระบบ n เวกเตอร์แยกเป็นระบบที่สมมาตร n เฟส" ดังนั้นทั้งระบบ 3 เฟสของกระแส หรือโวลเตจ สามารถแปลงให้อยู่ในระบบที่สมมาตรกันของแต่ละเวกเตอร์ได้ ซึ่งในการพิจารณาถึงข้อสมมุติฐานและวิธีการนี้ จะต้องระมัดระวังโดยการตรวจสอบให้ดีเสียก่อน

พิจารณาตัวอย่างง่ายๆ คือ series admittance ของระบบสายส่ง 3 เฟส ซึ่งแสดงในรูป 4.1 ซึ่งเป็นแบบที่มีการ couple กันในแต่ละ coils สามารถเขียน admittance matrix ซึ่งแสดงความสัมพันธ์ระหว่าง กระแสและโวลเตจได้ดังนี้

$$[I]_{abc} = [Y]_{abc} [V]_{abc} \quad (4.1.1)$$

เมื่อ

$$[I]_{abc} = [I_a \ I_b \ I_c]^T$$

$$[V]_{abc} = [V_a \ V_b \ V_c]^T$$

และ

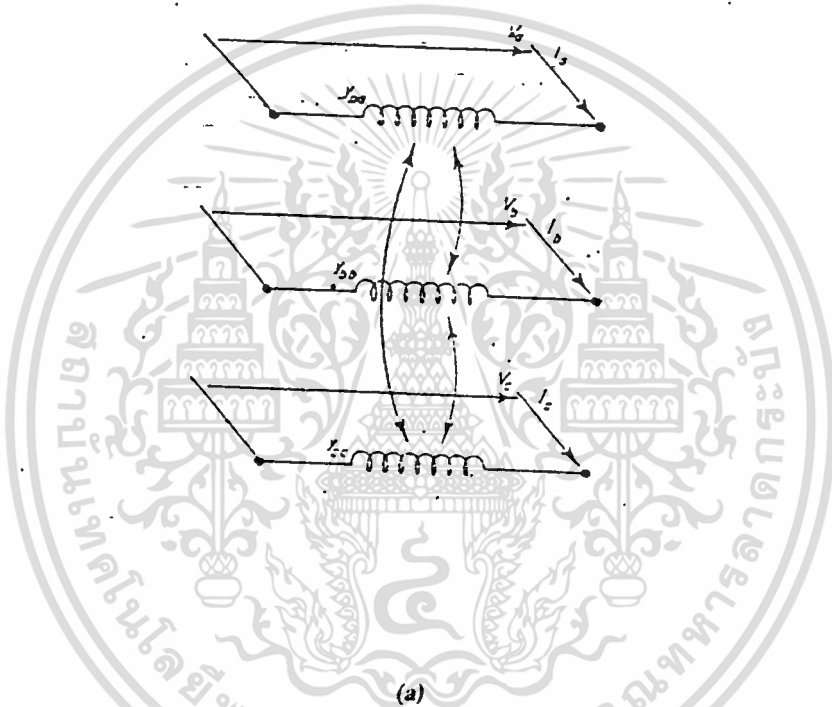
$$[Y]_{abc} =$$

$Y_{aa}$	$Y_{ab}$	$Y_{ac}$
$Y_{ba}$	$Y_{bb}$	$Y_{bc}$
$Y_{ca}$	$Y_{cb}$	$Y_{cc}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป (4.1.2)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้ symmetrical component transformation คอยล์ 3 ตัวในรูปที่ 1 สามารถจะแทนด้วย คอยล์ 3 ตัวที่ไม่ coupling กัน ด้วยวิธีนี้จึงจะสามารถทำให้แต่ละคอยล์ แยกเป็นอิสระจากกันได้โดยใช้การวิเคราะห์ทางคณิตศาสตร์



$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} y_{aa} & y_{ab} & y_{ac} \\ y_{ba} & y_{bb} & y_{bc} \\ y_{ca} & y_{cb} & y_{cc} \end{bmatrix} * \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

(b)

รูป 4.1 แสดงการแทน admittance ของระบบ 3 เฟสในแต่ละ element

(a) Series admittance element

(b) Admittance matrix representation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริมาณที่ถูกแปลงแล้ว ( โดยใช้ตัวท้าย 012 เพื่อแสดงถึง Zero, positive, negative sequence ตามลำดับ ) ซึ่งมีความสัมพันธ์กับปริมาณที่เป็นเฟสดังนี้

$$[V_{012}] = [T_1]^{-1} [V_{abc}] \quad (4.1.3)$$

$$[I_{012}] = [T_1]^{-1} [I_{abc}] \quad (4.1.4)$$

$$= [T_1]^{-1} [Y_{abc}] [T_1] [V_{012}] \quad (4.1.5)$$

— [T<sub>1</sub>] คือ transformation matrix —

โวลเตจและกระแสที่ถูกแปลงแล้วจะมีความสัมพันธ์กับ admittance matrix ที่ถูกแปลงแล้วคือ

$$[Y_{012}] = [T_1]^{-1} [Y_{abc}] [T_1] \quad (4.1.6)$$

สมมติว่าแต่ละ element มีความสมดุล เราจะได้

$$\begin{aligned} Y_{aa} &= Y_{bb} = Y_{cc} \\ Y_{ba} &= Y_{bc} = Y_{ca} \\ Y_{ca} &= Y_{cb} = Y_{ac} \end{aligned} \quad (4.1.7)$$

และถ้าชุดของ Invariant matrixes [T<sub>1</sub>] เป็นจริง การแปลงใน (4.1.6) จะเป็น Diagonal matrix [Y]<sub>012</sub>

จากที่กล่าวมาว่าระบบ 3 เฟสที่มีการ mutual couple จะถูกแทนด้วยระบบที่ uncouple ที่สมมาตรกัน 3 ชุด และถ้าในกรณีที่มีการส่งจ่ายกำลัง ( generation ) และโหลดมีความสมดุลหรือสมมติว่าสมดุล ระบบ 3 เฟสจะกลายเป็นระบบ single phase ซึ่งเป็นระบบ positive sequence ซึ่งสามารถละทิ้งอีก 2 เฟสได้ ซึ่งเป็นกรณีของ single phase load flow

แต่ถ้าในกรณีที่ admittance matrix [y<sub>abc</sub>] เป็นระบบไม่สมดุลเราไม่สามารถที่จะวิเคราะห์ด้วยแกนอ้างอิงของ Symmetrical component ซึ่งแกนอ้างอิงของ Symmetrical component จะใช้ได้เพียงเมื่อ network element อยู่ในภาวะสมดุล เช่นในเครื่องกำเนิดไฟฟ้าแบบเชิงโรตอร์

อย่างไรก็ตาม ข้อสมมติฐานบางอย่างก็อาจไม่เป็นจริงเสมอไป unsymmetrical inter-phase coupling ก็อาจจะมีในสายส่งและในเขตที่ใกล้กับหม้อแปลง และผลของการ coupling ระหว่าง sequence network ยิ่งกว่านั้นยังมีเฟสชิฟ phase shift ซึ่งเกิดจากการต่อหม้อแปลงแบบต่างๆ ซึ่งยากต่อการแทนใน sequence component model อีกด้วย

### ประโยชน์ของการใช้ Phase coordinates ใน symmetrical component

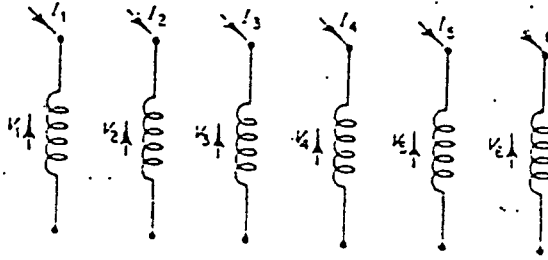
1. element ในระบบยังคงเหมือนเดิม
2. ลักษณะต่างๆ ของ asymmetric impedance, mutual coupling และระหว่าง element อื่นๆ และการสลับสายสามารถทำได้ง่ายขึ้น
3. ไม่มีปัญหาในเรื่อง phase shift ของหม้อแปลง

### การวิเคราะห์ Compound admittance

เมื่อทำการวิเคราะห์ three phase networks โหนดทั้ง 3 ที่บัสบาร์ จะมีความเกี่ยวข้องกันในการเชื่อมต่อกัน รูปแสดง network ในความหมายของ compound admittance ซึ่งหลักสำคัญคือการใช้ปริมาณของ matrix แทน admittance ของ network

กฎเกณฑ์และสมการของ network ทั่วๆ ไปจะยังใช้ได้ ใน compound network โดยแทนปริมาณเฟสเดียวด้วยแมทริกซ์ที่เหมาะสม

พิจารณา single admittance ที่มี mutually couple กัน 6 ตัว primitive network ซึ่งแสดงไว้ในรูปที่ 4.2



รูป 4.2 โครงข่ายเบื้องต้นของแอดมิตแตนซ์ที่คับเปิดกัน 6 ชุด

$$[Y_{aa}] = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix}$$

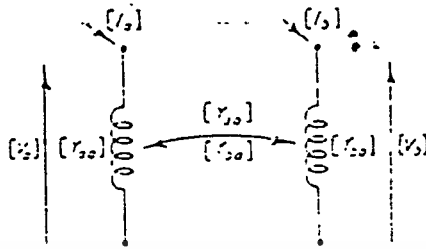
$$[Y_{bb}] = \begin{bmatrix} y_{44} & y_{45} & y_{46} \\ y_{54} & y_{55} & y_{56} \\ y_{64} & y_{65} & y_{66} \end{bmatrix}$$

$$[Y_{ab}] = \begin{bmatrix} y_{14} & y_{15} & y_{16} \\ y_{24} & y_{25} & y_{26} \\ y_{34} & y_{35} & y_{36} \end{bmatrix}$$

$$[Y_{ba}] = \begin{bmatrix} y_{41} & y_{42} & y_{43} \\ y_{51} & y_{52} & y_{53} \\ y_{61} & y_{62} & y_{63} \end{bmatrix}$$

จากรูปเราได้ทำการแบ่งกลุ่ม คอยล์ 6 ตัว เป็น 2 กลุ่ม คือ compound coil (a) และ (b) แต่ละอันประกอบด้วย admittance ข้อย ๆ 3 อัน ดังแสดงในรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.3 แอดมิทแตนซ์ที่คัปเปิลกันของสองกnoten

ความสัมพันธ์ระหว่าง primitive admittance matrix กับกระแสที่ไหลในแต่ละโหนดและโวลเตจในแต่ละ branch แสดงได้ดังนี้

$I_1$	$y_{11}$	$y_{12}$	$y_{13}$	$y_{14}$	$y_{15}$	$y_{16}$	$v_1$
$I_2$	$y_{21}$	$y_{22}$	$y_{23}$	$y_{24}$	$y_{25}$	$y_{26}$	$v_2$
$I_3$	$y_{31}$	$y_{32}$	$y_{33}$	$y_{34}$	$y_{35}$	$y_{36}$	$v_3$
$I_4$	$y_{41}$	$y_{42}$	$y_{43}$	$y_{44}$	$y_{45}$	$y_{46}$	$v_4$
$I_5$	$y_{51}$	$y_{52}$	$y_{53}$	$y_{54}$	$y_{55}$	$y_{56}$	$v_5$
$I_6$	$y_{61}$	$y_{62}$	$y_{63}$	$y_{64}$	$y_{65}$	$y_{66}$	$v_6$

\*

(4.1.8)

แยกสมการ (4.1.8) เป็น 3\*3 เมตริกซ์ และ 3\*1 เวกเตอร์ สามารถกลายเป็น

$[I_a]$	$=$	$[Y_{aa}]$	$[Y_{ab}]$	$[V_a]$
$[I_b]$		$[Y_{ba}]$	$[Y_{bb}]$	$[V_b]$

(4.1.9)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ

$$\begin{aligned} [I_i] &= [I_1 I_2 I_3]^T \\ [I_j] &= [I_1 I_2 I_3]^T \end{aligned} \quad (4.1.10)$$

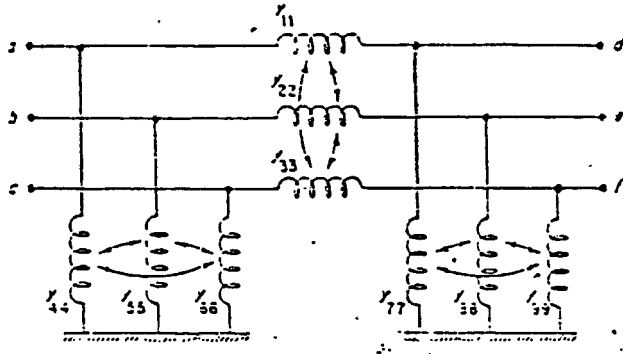
จากการตรวจสอบจะพบว่า  $[Y_{ab}]$  และ  $[Y_{ba}]$  มีความสัมพันธ์กันคือ  $[Y_{ab}] = [Y_{ba}]^T$  และถ้า  $y_{ik} = y_{ki}$  เมื่อ  $i = 1$  ถึง  $3$  และ  $k = 4$  ถึง  $6$  และถ้าการ coupling กันระหว่าง admittance ทั้ง 2 กลุ่มเป็น bilateral กันในกรณีนี้สมการ (4.1.9) จะเขียนได้เป็น

$$\begin{bmatrix} [I_i] \\ [I_j] \end{bmatrix} = \begin{bmatrix} [Y_{aa}] & [Y_{ab}] \\ [Y_{ba}] & [Y_{bb}] \end{bmatrix} * \begin{bmatrix} [V_i] \\ [V_j] \end{bmatrix}$$

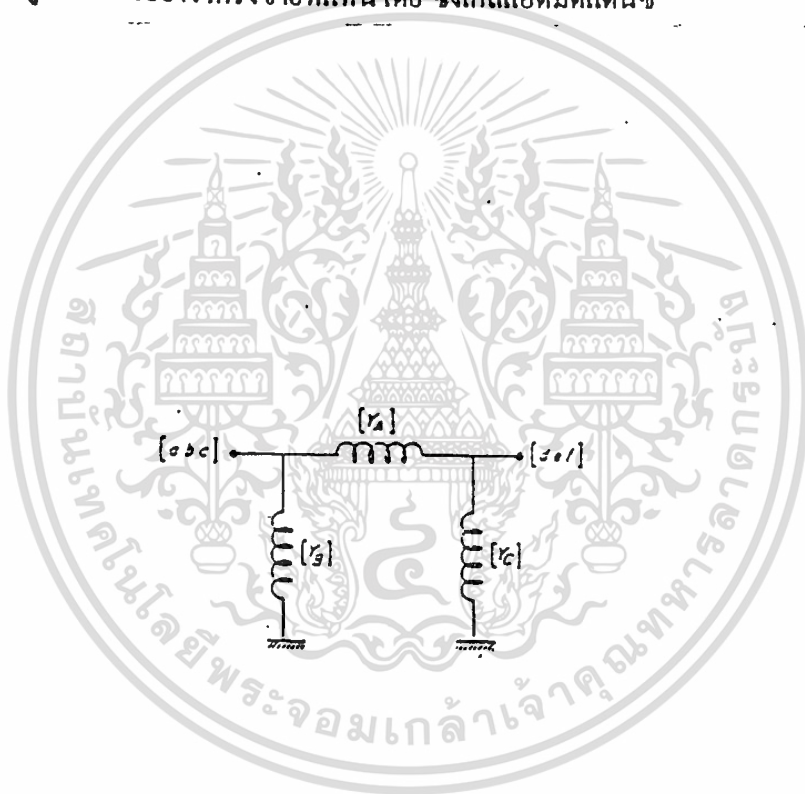
(4.1.11)

ดังนั้น Primitive network สำหรับ compound admittance จะถูกสร้างขึ้นโดยตรงจาก admittance แต่ละตัว และปริมาณต่างๆ ยังเป็นเมตริกซ์ที่อยู่ในตำแหน่งเดียวกับของ compound admittance admittance matrix ของ network ใดๆ จะประกอบด้วย compound admittance ซึ่งถูกสร้างโดยวิธี linear transformation อิลิเมนต์ที่มาต่อกันจะเป็นเมตริกซ์ขนาด  $n \times n$  เมื่อ  $n$  เป็นมิติของ compound admittance และถ้าสามารถแบ่งแยกเมตริกซ์ของ network ใดๆ ออกเป็น identity elements ซึ่งมีขนาดเท่ากันได้มากกว่าหนึ่ง การใช้ compound admittance จะยังมีประโยชน์มากขึ้น

ตัวอย่างในการพิจารณาแสดงในรูป 4.4 และ 4.5 ซึ่งเป็นรูปแสดงส่วนหนึ่งของสายส่ง admittance matrix สร้างขึ้นโดย single และ compound admittance primitive networks และ admittance อื่นๆ แสดงในรูปที่ 4.6 matrix ที่แสดงถึง single และ compound network แสดงโดยสมการที่ (4.1.12) และ (4.1.13) ตามลำดับ

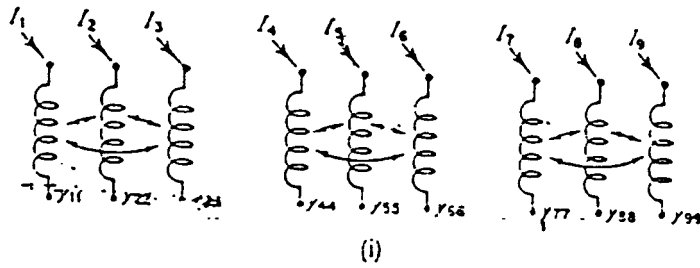


รูป 4.4 ตัวอย่างโครงข่ายที่แทนโดย จิงเกิลแอตมิตแดนซ์

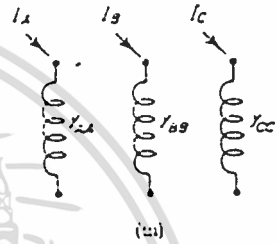


รูป 4.5 ตัวอย่างโครงข่ายที่แทนโดย คอมพาร์น แอตมิตแดนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า-  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$Y_{11}$	$Y_{12}$	$Y_{13}$							
$Y_{21}$	$Y_{22}$	$Y_{23}$							
$Y_{31}$	$Y_{32}$	$Y_{33}$							
		$Y_{44}$	$Y_{45}$	$Y_{46}$					
		$Y_{54}$	$Y_{55}$	$Y_{56}$					
		$Y_{64}$	$Y_{65}$	$Y_{66}$					
					$Y_{77}$	$Y_{78}$	$Y_{79}$		
					$Y_{87}$	$Y_{88}$	$Y_{89}$		
					$Y_{97}$	$Y_{98}$	$Y_{99}$		



$Y_A$		
$Y_B$		
$Y_C$		

(ii)

(iv)

รูป 4.6 โครงข่ายเบื้องต้นและเมตริกซ์ของแอดมิตแตนซ์

(i) Primitive network โดยใช้ single admittance

(ii) Primitive admittance matrix

(iii) Primitive network โดยใช้ compound admittance

(iv) Primitive admittance matrix

$v_1$
$v_2$
$v_3$
$v_4$
$v_5$
$v_6$
$v_7$
$v_8$
$v_9$

$$=$$

-1			1		
	-1			1	
		-1			1
1					
	1				
		1			
			1		
				1	
					1

$$*$$

$v_a$
$v_b$
$v_c$
$v_d$
$v_e$
$v_f$

(4.1.12)

$[v_A]$
$[v_B]$
$[v_C]$

$$=$$

-1	1
1	
	1

$$*$$

$[v_{ABC}]$
$[v_{DEF}]$

(4.1.13)

เนื้ตเวอ์รคของ admittance matrix จะถูกกำหนดโดยสมการ Linear Transformation ดังนี้

$$[Y_{node}] = [C]^T [Y_{prim}] [C]$$

การคูณเมตริกซ์นี้สามารถใช้ทั้งรูปเต็มของเมตริกซ์หรืออยู่ในรูปแบ่งกันแล้วก็ได้ ผลของการแบ่งกัน จะได้

$$[Y_{node}] = \begin{bmatrix} [Y_A] + [Y_B] & -[Y_A] \\ -[Y_A] & [Y_A] + [Y_C] \end{bmatrix}$$

## กฎในการสร้าง admittance matrix ในเน็ตเวิร์ก

วิธี Linear Transformation อาจจะใช้หา admittance matrix ของเน็ตเวิร์กทั่วไปได้ แต่ในกรณีพิเศษคือเน็ตเวิร์กไม่มีการ mutual coupling ระหว่างกัน กฎเกณฑ์ง่าย ๆ ในการสร้าง admittance matrix โดยการตรวจสอบ โดยให้ compound network ที่ไม่มีการ mutual coupling ระหว่าง compound admittance โดยทำตามดังนี้

1.เทอม diagonal หากจากผลรวมของแต่ละ branch admittance ซึ่งต่ออยู่กับโหนดต่าง ๆ ที่เราพิจารณา

2.เทอม off-diagonal หากจากค่าลบของผลรวมของ branch admittance ซึ่งต่ออยู่ระหว่างโหนดที่พิจารณา

### 4.2 แบบจำลองของสายส่ง 3 เฟส (Tree phase model of transmission lines)

ค่าต่าง ๆ ของสายส่งมักจะคำนวณจากลักษณะทางเรขาคณิตของสาย ค่าพารามิเตอร์ที่คำนวณได้จะอยู่ในรูปของ series impedance และ shunt admittance ต่อหน่วยความยาวของสาย ผลของ ground current และ earth wires จะรวมอยู่ในการคำนวณของพารามิเตอร์เหล่านี้

Series impedance

สายส่งแบบ 3 เฟสที่มี ground wire แสดงในรูปที่ 4.7 (a)

สำหรับเฟส a สามารถเขียนสมการได้ดังนี้

$$V_a - V'_a = I_a(R_a + j\omega L_a) + I_b(j\omega L_{ab}) + I_c(j\omega L_{ac}) + j\omega L_{ag} - j\omega L_{an} I + V_n$$

$$V_n = I_n(R_n + j\omega L_n) - I_a(j\omega L_{na}) - I_b(j\omega L_{nb}) - I_c j\omega L_{nc} - I_g j\omega L_{ng}$$

แทนค่าด้วย

$$I_n = I_a + I_b + I_c + I_g$$

$$V_a - V'_a = I_a(R_a + j\omega L_a) + I_b j\omega L_{ab} + I_c j\omega L_{ac} + j\omega L_{ag} - j\omega L_{an}(I_a + I_b + I_c + I_g) + V_n$$

จัดกลุ่มและแทนค่า  $V_n$

$$\begin{aligned} \Delta V_a &= V_a - V'_a \\ &= I_a(R_a + j\omega L_a + j\omega L_{an} + R_n + j\omega L_n - j\omega L_{nn}) \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$+I_b(j\omega L_{ab}-j\omega L_{an}+R_n+j\omega L_n-j\omega L_{nb})$$

$$+I_c(j\omega L_{ac}-j\omega L_{an}+R_n+j\omega L_n-j\omega L_{nc})$$

$$+I_g(j\omega L_{ag}-j\omega L_{an}+R_n+j\omega L_n-j\omega L_{ng})$$

$$\Delta V_a = I_a(R_a+j\omega L_a-2j\omega L_{an}+R_n+j\omega L_n)$$

$$+I_b(j\omega L_{ab}-j\omega L_{an}+R_n+j\omega L_n-j\omega L_{nb})$$

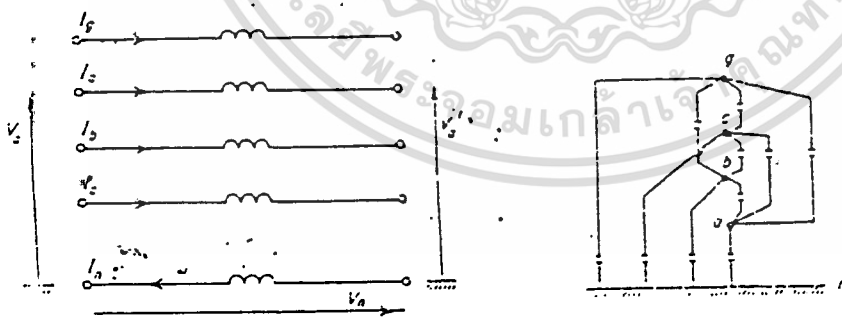
$$+I_c(j\omega L_{ac}-j\omega L_{an}+R_n+j\omega L_n-j\omega L_{nc})$$

$$+I_g(j\omega L_{ag}-j\omega L_{gn}+j\omega L_{an}+R_n+j\omega L_n)$$

$$\Delta V_n = Z_{aa-n}I_a + Z_{ab-n}I_b + Z_{ac-n}I_c + Z_{ag-n}I_g \tag{4.2.1}$$

$V_n$	$Z_{aa-n}$	$Z_{ab-n}$	$Z_{ac-n}$	$Z_{ag-n}$	$I$
$\Delta V$	$Z_{ba-n}$	$Z_{bb-n}$	$Z_{bc-n}$	$Z_{bg-n}$	$I$
$\Delta V$	$Z_{ca-n}$	$Z_{cb-n}$	$Z_{cc-n}$	$Z_{cg-n}$	$I$
$\Delta V$	$Z_{ga-n}$	$Z_{gb-n}$	$Z_{gc-n}$	$Z_{gg-n}$	$I$

$$\tag{4.2.2}$$



(a)

(b)

รูป 4.7 วงจรสมมูลย์ของสายส่งสามเฟสแบบอนุกรม และแบบขนาน

(b) Three phase transmission shunt impedance equivalent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราสนใจเพียงคุณลักษณะของ phase conductor เราอาจใช้ three phase conductor แทนสายส่งได้ ซึ่งจะเขียนสมการ (4.2.2) ในรูปของ partitioned matrix ได้ดังนี้

$$\begin{bmatrix} \Delta v_{abc} \\ \Delta v_g \end{bmatrix} = \begin{bmatrix} Z_A & Z_B \\ Z_C & Z_D \end{bmatrix} \begin{bmatrix} I_{abc} \\ I_g \end{bmatrix} \quad (4.2.3)$$

จาก (4.2.3)

$$\Delta v_{abc} = Z_A I_{abc} + Z_B I_g \quad (4.2.4)$$

$$\Delta v_g = Z_C I_{abc} + Z_D I_g \quad (4.2.5)$$

จากสมการ (4.2.3) และ (4.2.4) และสมมติค่า ground wire เป็น zero potential

$$\Delta v_{abc} = Z_{abc} I_{abc} \quad (4.2.6)$$

เมื่อ

$$Z_{abc} = Z_a - Z_b Z_c^{-1} Z_c =$$

Z'aa-n	Z'ab-n	Z'ac-n
Z'ba-n	Z'bb-n	Z'bc-n
Z'ca-n	Z'cb-n	Z'cc-n

Shunt admittance

จากรูป 7 (b) potential ของ line conductor จะมีความสัมพันธ์กับ conductor charges โดยสมการของเมตริกซ์

ดังนี้

$$\begin{bmatrix} V_a \\ V_b \\ V_c \\ V_g \end{bmatrix} = \begin{bmatrix} P_{aa} & P_{ab} & P_{ac} & P_{ag} \\ P_{ba} & P_{bb} & P_{bc} & P_{bg} \\ P_{ca} & P_{cb} & P_{cc} & P_{cg} \\ P_{ga} & P_{gb} & P_{gc} & P_{gg} \end{bmatrix} \begin{bmatrix} Q_a \\ Q_b \\ Q_c \\ Q_g \end{bmatrix} \quad (4.2.7)$$

พิจารณาพร้อมกับ series impedance matrix จะได้

$$V_{abc} = P'_{abc} Q_{abc} \quad (4.2.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $P'_{abc}$  คือ  $3 \times 3$  เมตริกซ์ ซึ่งผลรวมของ ground wire และ capacitance matrix ของ transmission line ในรูปที่ 4.7 เขียนได้เป็น

$$C_{abc} = P'^{-1}_{abc} =$$

Caa	-Cab	-Cac
-Cba	Cbb	-Cbc
-Cca	-Ccb	Ccc.

Series impedance และ Shunt admittance แทนสายส่ง 3 เฟส ด้วยแบบจำลอง  $\pi$  ดังแสดงในรูปที่ 4.8 (a) และแทนสายส่งด้วย เมตริกซ์ในรูป 4.8 (b) ซึ่งเมตริกซ์ทั้งสองสามารถแทนด้วย compound admittances รูป 4.8 (c) ดังที่ได้อธิบายไว้แล้ว

โดยการใช้การแปลง admittance matrix ซึ่งจะแสดงกระแสที่ไหลเข้าโหนดในรูป 4.8 (c) ซึ่งมีความสัมพันธ์กับ โวลเตจ ดังสมการ (4.2.9)

$$\begin{bmatrix} [I_i] \\ [I_k] \end{bmatrix} = \begin{bmatrix} [Z]^{-1} + [Y]/2 & -[Z]^{-1} \\ -[Z]^{-1} & [Z]^{-1} + [Y]/2 \end{bmatrix} \begin{bmatrix} [V_i] \\ [V_k] \end{bmatrix}$$

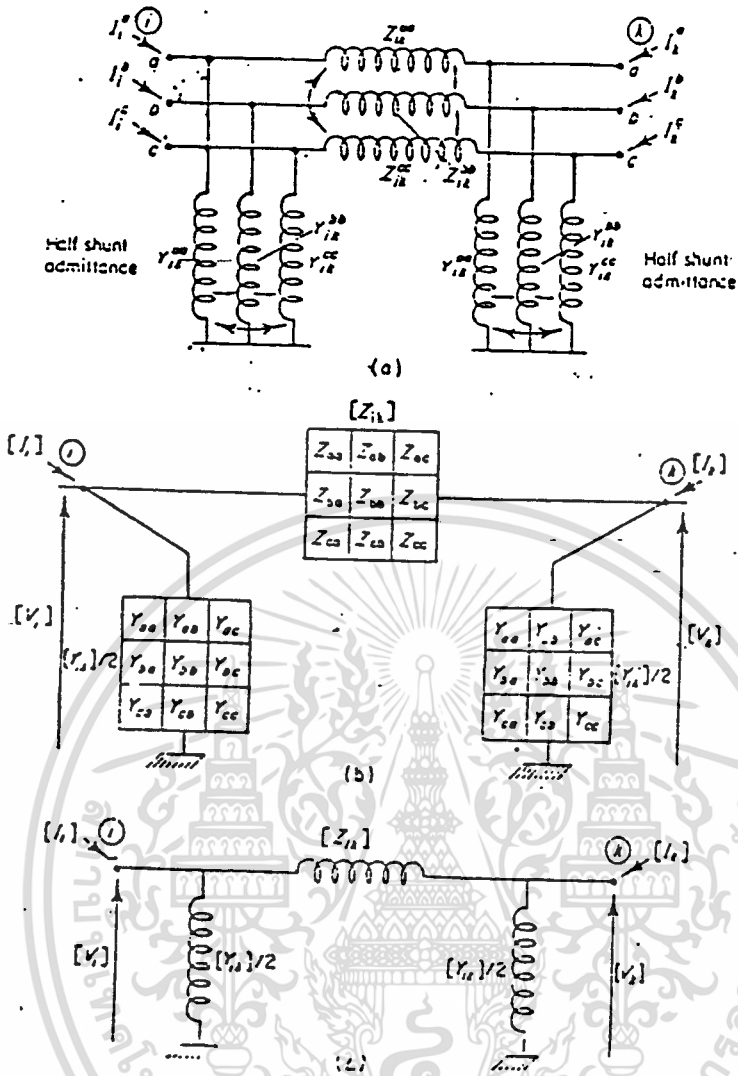
$6 \times 1$      $6 \times 6$      $6 \times 1$

โดยสมการข้างต้นเป็นการสร้าง admittance matrix แทนสายส่งระยะสั้นระหว่างบัส i และ k ในเทอมของ  $3 \times 3$  เมตริกซ์

โดยแบบจำลองนี้อาจจะมีความแม่นยำน้อย ในกรณีของสายส่งระยะไกล ซึ่งเป็นผลจากความยาวคลื่น ถ้าพิจารณาถึงความถี่ฮาร์โมนิกด้วยแล้ว ผลอันนี้จะน้อยมาก โดยใช้สายส่งและทฤษฎีของ wave propagation เราจะหาแบบจำลองได้ละเอียดยิ่งขึ้น อย่างไรก็ตามถ้าพิจารณาในแง่ความถี่ปกติ เราสามารถใช้แบบจำลองของ  $\pi$  network ในสายส่งระยะยาว

**Mutually coupled three phase lines**

เมื่อสายส่ง 2 เส้นหรือมากกว่าวางอยู่ในแนวเดียวกัน จะต้องคำนึงถึงผลของ สนามไฟฟ้าสถิตย์ (electrostatics) และสนามแม่เหล็กไฟฟ้า (electromagnetic) พิจารณากรณีง่ายที่สุดคือ มีสายส่ง 3 เฟส 2 ชุด ซึ่งมี mutually coupled ระหว่างกัน การ coupled กันของสาย 2 ชุด ถูกแยกเป็นระบบย่อย (subsystem) ซึ่งประกอบด้วย บัสบาร์ของระบบ สายที่มีการ coupled กันแสดงในรูปที่ 4.9 เมื่อแต่ละอีลิเมนต์เป็น  $3 \times 3$  compound admittance และทั้งกระแสและโวลเตจเป็นเวกเตอร์แบบ  $3 \times 1$



รูป 4.8 แบบจำลอง Lumped- $\pi$  ของสายส่งสามเฟสระยะสั้นของอินพีแดนซ์แบบอนุกรม

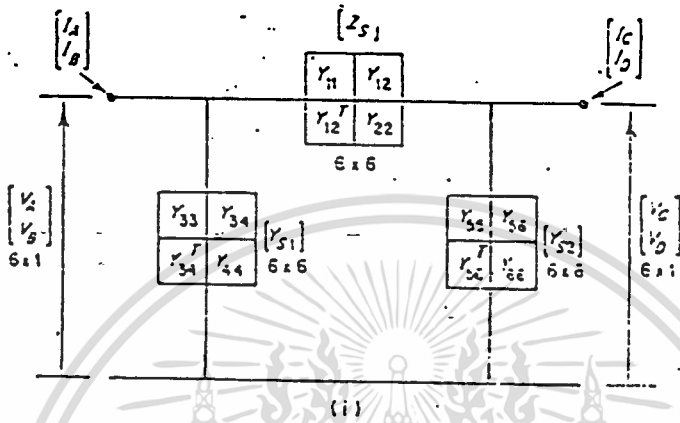
- (a) Full circuit representation
- (b) Matrix equivalent
- (c) โดยใช้ Three-phase compound admittance

coupled series elements ซึ่งแทน electromagnetic coupling ในขณะที่ coupled shunt elements ซึ่งแทน capacitive หรือ electrostatic coupling ซึ่ง coupling parameters เหล่านี้สามารถแทนด้วยรูปมาตรฐานของ line parameters ของสายส่งได้

โดย admittance ซึ่งแสดงในรูป 4.9 และใช้ linear transformation สำหรับ compound network ซึ่ง admittance matrix สำหรับระบบย่อยถูกกำหนดโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบย่อยอาจจะเขียนใหม่ได้ดังรูป 4.10 แต่ละคู่ของ couple 3\*3 compound admittance จะถูกแทนด้วย 3\*3 compound admittance และแสดงการแทนเมตริกซ์ไว้ในรูปด้วย โดยใช้แบบจำลองนี้และสัญลักษณ์ที่แสดงใน admittance blocks ในรูป ค่า admittance matrix อาจเขียนในเทอมของ 6\*6 compound coils



รูปที่ 4.9

รูป 4.9 คอมพาว์น แอดมิทแตนซ์ ที่แทน การกลับเบิ้ลของสายส่ง 3 เฟส 2 ชุด

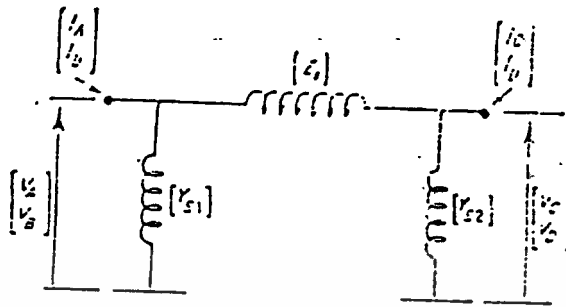
tion)

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_D \end{bmatrix} = \begin{bmatrix} [Z_S]^{-1} + [Y_{S1}] & -[Z_S]^{-1} \\ -[Z_S]^{-1} & [Z_S]^{-1} + [Y_{S2}] \end{bmatrix} \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_D \end{bmatrix}$$

(4.2.11)

ซึ่งจะเห็นได้ชัดว่ามีรูปแบบเหมือนสมการ (4.2.10) โดยการแบ่งเมตริกซ์

การแทนในรูป 4.9 มีความซับซ้อนและละเอียดแสดงโดยสมการ (4.2.11) ซึ่งคล้ายกับผลที่ใช้ 3\*3 compound admittance ใน single three-phase line



รูป 4.10 คอมพาวน์ แอดมิตแตนซ์ ที่แทน การคัปเปิลของสายส่ง 3 เฟส 2 กล่ม

*6\*6 Compound admittance representation of two coupled three phase lines*

*(6\*6 Compound admittance representation)*



รูป 4.11 การเหนี่ยวนำระหว่างสายของสายส่งแบบขนาน

(4.2.10)

$I_A$	=	$Y_{11}+Y_{23}$	$Y_{12}+Y_{34}$	$-Y_{11}$	$-Y_{12}$	$V_a$
$I_B$		$Y_{12}^T+Y_{34}^T$	$Y_{22}+Y_{44}$	$-Y_{12}^T$	$-Y_{22}$	$V_b$
$I_C$		$-Y_{11}$	$-Y_{12}$	$Y_{11}+Y_{55}$	$Y_{12}+Y_{56}$	$V_c$
$I_D$		$-Y_{12}^T$	$-Y_{22}$	$Y_{12}^T+Y_{56}^T$	$Y_{22}+Y_{66}$	$V_d$
$12 \times 1$		$12 \times 12$				$12 \times 1$

ถ้าสมมติว่า mutual coupling เป็น bilateral ดังนั้น  $Y_{12}=Y_{21}^T$  เป็นต้น

ระบบย่อยอาจเขียนใหม่ได้ดังรูป 4.10 แต่ละคู่ของ couple 3\*3 compound admittance จะถูกแทนด้วย 3\*3 compound admittance และแสดงการแทนเมตริกซ์ไว้ในรูปด้วย โดยใช้แบบจำลองนี้และสัญลักษณ์ที่แสดงใน admittance blocks ในรูป ค่า admittance matrix อาจเขียนในเทอมของ 6\*6 compound coils

(4.2.11)

$\begin{bmatrix} I_A \\ I_B \end{bmatrix}$	=	$[Z_S]^{-1}+[Y_{S1}]$	$-[Z_S]^{-1}$	$\begin{bmatrix} V_A \\ V_B \end{bmatrix}$
$\begin{bmatrix} I_C \\ I_D \end{bmatrix}$		$-[Z_S]^{-1}$	$[Z_S]^{-1}+[Y_{S2}]$	$\begin{bmatrix} V_C \\ V_D \end{bmatrix}$

ซึ่งจะเห็นได้ชัดว่ามีรูปแบบเหมือนสมการ (4.2.10) โดยการแบ่งเมตริกซ์

การแทนในรูป 4.10 มีความซับซ้อนและละเอียดแสดงโดยสมการ (4.2.11) ซึ่งคล้ายกับผลที่ใช้ 3\*3 compound admittance ใน single three-phase line

ข้อมูลที่จะเป็นเพื่อสร้างเมตริกซ์ขึ้นนั้นจะคล้ายกับแบบ single line คือ จะเป็น series impedance และ shunt admittance และจะพบว่าเป็นแบบ 3\*3 สำหรับ single line 6\*6 สำหรับ Two couple lines, 9\*9 สำหรับ three couple lines และ 12\*12 สำหรับ four couple lines

การหาเมตริกซ์  $[Z_S]$  และ  $[Y_S]$  admittance matrix สำหรับระบบย่อยถูกสร้างโดยใช้สมการ (4.2.11)

เมื่อ巴士าร์ทั้งหมดของ couple line มีความแตกต่างกัน ระบบย่อยอาจรวมกันโดยตรงเป็น system admittance matrix อย่างไรก็ตาม ถ้า巴士าร์ไม่มีความแตกต่างกันแล้ว admittance matrix อาจหาได้จากสมการ (4.2.11) ซึ่งจะพิจารณาต่อไป

การพิจารณา terminal connection

admittance matrix ที่กล่าวมาแล้วสามารถลดรูปลงได้ถ้ามี element ที่ต่างกันอยู่ ติดอยู่ที่巴士าร์เดียวกัน พิจารณาสายส่งที่ขนานกัน 2 เส้น ในรูปที่ 4.11

ถ้าใช้ admittance matrix ที่ได้กล่าวมาแล้วแสดงความสัมพันธ์ของกระแสและโวลเตจที่บัสบาร์ทั้ง 4 คือ A1,A2,B1,B2 แสดงความสัมพันธ์ได้ดังนี้

$$\begin{bmatrix} I_{A1} \\ I_{A2} \\ I_{A3} \\ I_{A4} \end{bmatrix} = \begin{bmatrix} Y_{A1 A2 B1 B2} \end{bmatrix} \begin{bmatrix} V_{A1} \\ V_{A2} \\ V_{A3} \\ V_{A4} \end{bmatrix}$$

(4.2.12)

nodal injected current ที่บัสบาร์ A ( $I_A$ ) เขียนได้เป็น

$$I_A = I_{A1} + I_{A2}$$

ทำนองเดียวกัน

$$I_B = I_{B1} + I_{B2}$$

จากการสังเกตในรูปที่ 4.11

$$V_A = V_{A1} = V_{A2}$$

$$V_B = V_{B1} = V_{B2}$$

เมตริกซ์ที่ต้องใช้ในการแสดงความสัมพันธ์ระหว่างกระแสและโวลเตจของบัสบาร์ หาได้

จากสมการ (4.2.12) และเงื่อนไขที่ได้กล่าวไว้แล้วคอนตันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} I_A \\ I_B \end{bmatrix} = \begin{bmatrix} Y_{AB} \end{bmatrix} \begin{bmatrix} V_{A1} \\ V_{A2} \end{bmatrix}$$

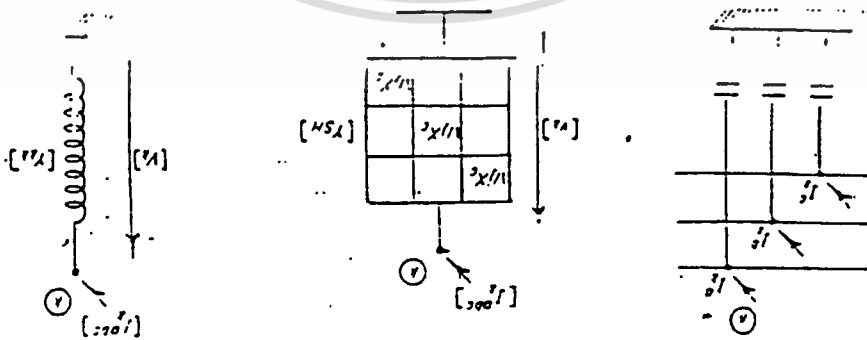
(4.2.13)

เมตริกซ์  $[Y_{AB}]$  เป็น nodal admittance matrix ของ subsystem และสังเกตว่าเมตริกซ์ในสมการ (4.2.12) จะต้องนำไปใช้คำนวณหาค่า power line flows ในส่วนต่างๆ

#### Shunt elements

shunt reactors และ capacitors จะใช้ในระบบไฟฟ้ากำลังเพื่อควบคุม reactive power ข้อมูลสำหรับ element เหล่านี้โดยทั่วไปจะอยู่ในรูปพิกัด MVA และ kV ของตัวมันเองค่า equivalent phase admittance ในรูป p.u. ซึ่งคำนวณได้จากข้อมูลเหล่านี้

พิจารณา three phase capacitor bank ในรูปที่ 4.12 โดยแทนในรูปเมตริกซ์ ในทำนองเดียวกับที่ได้กล่าวมาแล้ว admittance matrix สำหรับ shunt elements โดยทั่วไปจะเป็น diagonal ซึ่งไม่มีการ coupling ระหว่างองค์ประกอบของแต่ละเฟส ซึ่งเมตริกซ์นี้จะถูกต่อเข้ากับระบบ admittance matrix โดยใช้แต่เพียงค่า self admittance ของแต่ละบัส



รูป 4.12 คาปาซิเตอร์ แบงก์ แบบขนาน

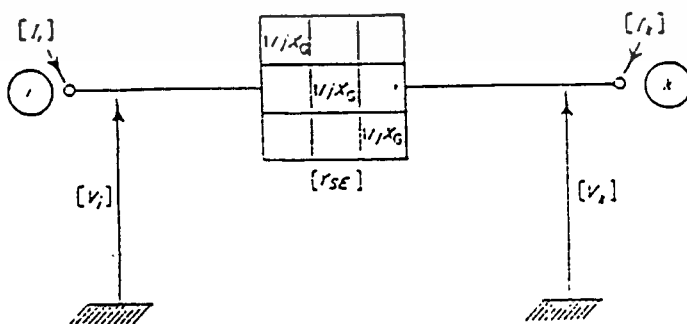
## Series element

element ที่ต่อโดยตรงระหว่างบัส 2 บัส จะพิจารณาเป็น series elements ตัวอย่างเช่น series capacitor bank ซึ่งไม่มีการ coupling กัน ซึ่งจะเขียนเป็น admittance matrix ที่มีแต่ diagonal ซึ่งแทนได้ในรูปที่ 4.13

Admittance matrix สำหรับ subsystem สามารถเขียนโดยการตรวจสอบ

$$[Y] = \begin{bmatrix} [Y_{SE}] & -[Y_{SE}] \\ -[Y_{SE}] & [Y_{SE}] \end{bmatrix}$$

(4.2.14)



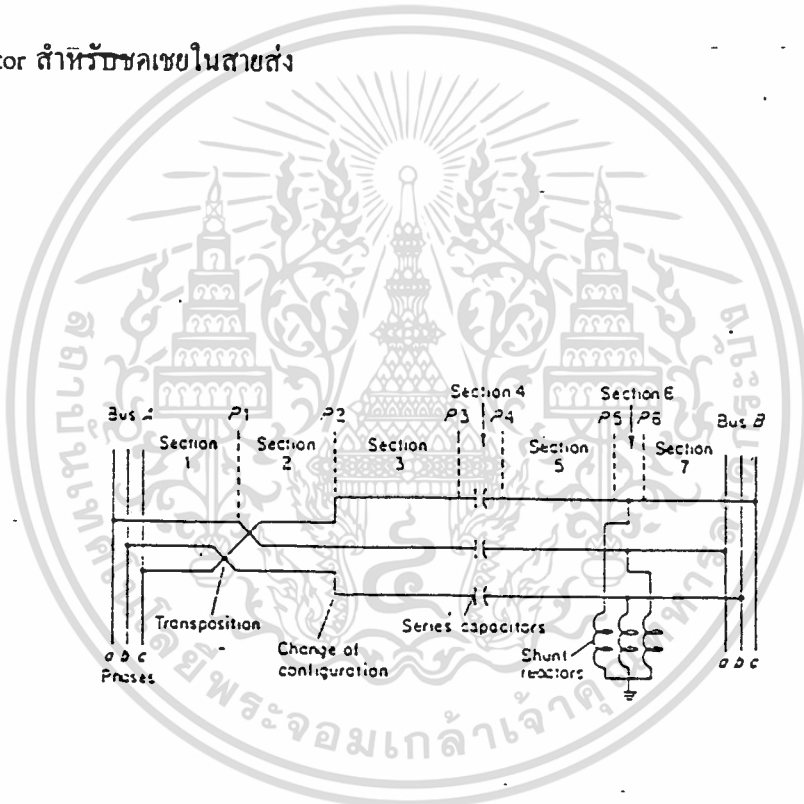
รูป 4.13 คาปาซิเตอร์ แบงก์ แบบอนุกรม ระหว่างโหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การแยกส่วนของสายส่ง ( Line sectionalization )

สายส่งสามารถแยกเป็นส่วนย่อยได้ ตามเงื่อนไขต่อไปนี้

- การสลับสายของตัวนำชนิดสาย ( line conductor )
- การเปลี่ยนชนิดของฐานรองรับเสาของสายส่ง
- การเปลี่ยนแปลงความชันของดิน
- การปรับปรุงแบบจำลองของสายส่ง
- Series capacitor สำหรับชดเชยในสายส่ง



รูป 4.14 ตัวอย่าง สายส่งที่แยกกันเป็นส่วนย่อย

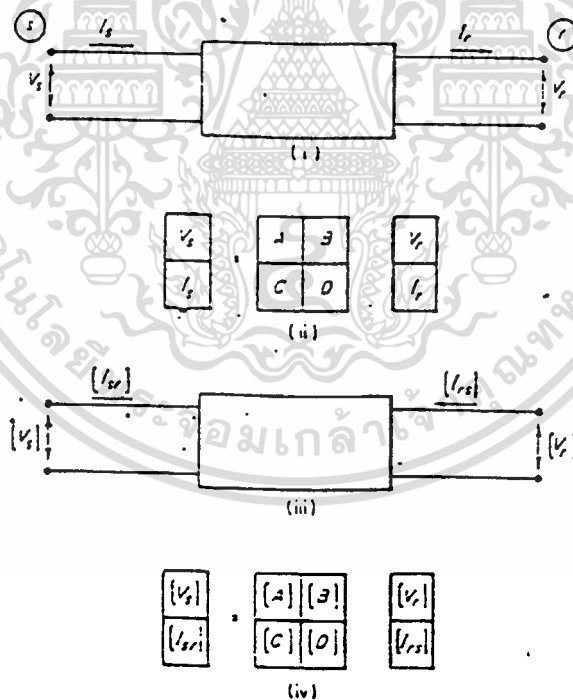
Network ในรูป 4.14 จะถูกพิจารณาเป็นส่วนย่อย ค่า admittance matrix ระหว่างบัส A และ B หาได้โดยพิจารณาเป็นส่วนๆ ไป โดยหาค่า ABCD หรือ พารามิเตอร์ของสายส่ง หลังจากนั้นคูณค่าเหล่านี้จะได้ admittance matrix ที่เป็นผลลัพธ์ได้ในที่สุด

วิธีนี้จะใช้ทฤษฎีของ two port network รวบรวมกันหลายๆ ตัวเป็น multi-two-port networks ค่ากระแสและโวลเตจของ matrix อันใหม่ ได้แสดงในรูปที่ 4.25 พร้อมกับค่า matrix parameter A B C D

ขนาดของ parameter matrix ขึ้นอยู่กับจำนวนส่วนที่แยกออก เช่น 3,6,9 หรือ 12 สำหรับ 1,2,3 หรือ 4 mutually coupled three phase elements เป็นต้น

### ลักษณะที่น่าสนใจ

- (a) เพื่อความสะดวกของโปรแกรม จะให้หม้อแปลงเป็นแบบอุดมคติ
- (b) เครื่องหมาย dot แสดงการ coupling ซึ่งกำหนดให้เป็นศูนย์ เพื่อแสดงถึงขนาดของ matrix ได้อย่างถูกต้อง
- (c) ในระบบ per unit mutual coupling ระหว่างสาย 220 kV และ 66kV ถูกแสดงในรูป voltage base ซึ่งกำหนดโดยค่าเฉลี่ยเรขาคณิต ของ base line neutral voltages ของวงจรขนาน 2 วงจร



รูป 4.15 พารามิเตอร์ที่ใช้ในการแปลงระบบสายส่ง

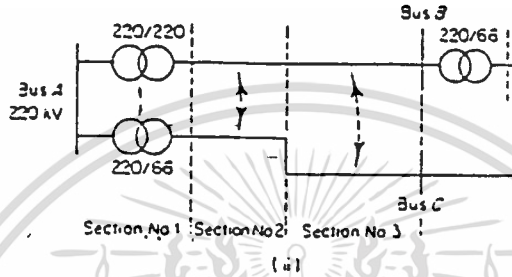
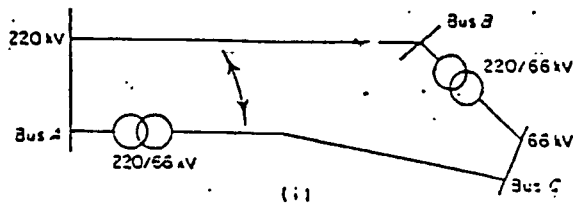
(i) Normal two port network

(ii) transmission parameters

(iii) Multi-two-port network

(iv) Matrix transmission parameters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.16 ตัวอย่างของระบบ เพื่อใช้อธิบายการแยกส่วนของส่วนส่ง

(i) System single line diagram

(ii) System redrawn to illustrate line sectionalization

ตารางที่ 4.1 พารามิเตอร์ ABCD

Transmission line	$[u] + [Z][Y]/2$	$-[Z]$
	$[Y]([u] + [Z][Y]/4)$	$-([u] + [Y][Z]/2)$

Transformer	$-[Y_{sr}]^{-1}[Y_{ss}]$	$[Y_{sr}]^{-1}$
	$[Y_{rs}] - [Y_{rr}][Y_{sr}]^{-1}[Y_{ss}]$	$[Y_{rr}][Y_{sr}]^{-1}$

Shunt element	$[u]$	$[0]$
	$[Y_{sh}]$	$-[u]$

Series element	$[u]$	$-[Y_{se}]^{-1}$
	$[0]$	$-[u]$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตาราง 4.1 [u] เป็น unit matrix, [o] เป็น เมตริกซ์อื่นๆ เป็นไปตามที่ได้กล่าวมาแล้วตามหัวข้อก่อน เมตริกซ์เหล่านี้จะมีมิติขนาดเท่าใด ขึ้นอยู่กับจำนวน couple three phase element ในแต่ละส่วน

ผลลัพธ์ของพารามิเตอร์ ABCD จะนำไปใช้ในการหา equivalent nodal admittance matrix สำหรับ subsystem ตามสมการข้างล่างนี้

$$[Y] = \begin{bmatrix} [D][B]^{-1} & [C]-D[B]^{-1}[A] \\ [B]^{-1} & -[B]^{-1}[A] \end{bmatrix}$$

(4.2.15)

### Formation of the system admittance matrix

Subsystem admittance matrix สามารถรวมเข้ากันเพื่อ system admittance matrix ได้ดังนี้

1. ค่า self admittance ของบัสบาร์ใดคือผลรวมของ self admittance matrix ย่อยๆทั้งหมดที่บัสบาร์นั้น
2. ค่า mutual admittance ระหว่าง บัสบาร์ 2 จุดใดๆ จะเป็นผลรวมของ mutual admittance matrix ย่อยๆทั้งหมดของ subsystem ของโหนด 2 โหนดนั้น

## บทที่ 5

การศึกษาโหลดโฟลว์แบบสามเฟส

## 5.1 บทนำ

ในกรณีที่ศึกษาระบบไฟฟ้ากำลังที่อยู่ในภาวะ steady state ถ้าเราไม่คำนึงถึงความไม่สมดุลของระบบ ( System unbalance ) แล้วเราสามารถจะใช้ single phase load flow คำนวณได้ก็เพียงพอ อย่างไรก็ตามในทางปฏิบัติ ก็อาจจะเป็นการเปลี่ยนแปลง ในการทำให้โหลดสมดุล หรืออิมพีแดนซ์ของสายส่งสมดุล อันเนื่องจากสายส่งไฟฟ้าแรงสูงที่ไม่มีการไขว้สาย

ผลของระบบไฟฟ้ากำลังที่ไม่สมดุล คือ จะมีกระแสลำดับเฟสลบ (negative sequenc current) ซึ่งทำให้โรเตอร์ของเครื่องจักรเกิด overheating ขึ้นได้ และ กระแสลำดับเฟสศูนย์ (Zero sequence current) ยังทำให้เกิดการทำงานผิดปกติ และยังเพิ่มความสูญเสียในสายส่งขนานที่ไม่ได้ไขว้สายไว้

แบบจำลองระบบ 3 เฟสที่ใช้หลักการรวม network admittance ทั้งหมด ดังได้กล่าวมาแล้ว เพื่อใช้สำหรับ three phase load flow ต่อไป และเวลาที่ใช้ในการคำนวณและ storage program ที่ใช้จะมีค่ามากกว่าในกรณีของ single phase มาก ดังนั้น algorithms ที่ใช้ในการคำนวณ จะต้องมีประสิทธิภาพที่ดีกว่าใน single phase load flow โดยทั่วไปการศึกษเกี่ยวกับ three-phase load flow จะใช้ในการวางแผนและเตรียมการมากกว่าที่จะใช้ขณะปฏิบัติการ

## 5.2 สัญลักษณ์ต่างๆ ( Notation )

เพื่อความสะดวก จะกำหนดสัญลักษณ์ต่างๆ ต่อไปนี้แทน three phase vector และ matrix elements โดยใช้ตัวห้อยท้าย และตัวยกขึ้นดังนี้

เมื่อพิจารณา a.c.system ที่มีบัสบาร์ทั้งหมด  $n$  จะได้

$$n = n_b + n_g$$

$n_b$  คือจำนวนบัสบาร์ ที่มีอยู่ในระบบจริงๆ

$n_g$  คือจำนวนเครื่องจักรไฟฟ้าแบบ ซิงโครนัส ( Synchronous machine )

ตัวห้อยท้าย ( subscript )  $i, j$  แทนบัสบาร์ในระบบดังนี้

$i = 1, nb$  แทนบัสบาร์ที่มีอยู่ในระบบจริงๆ คือ load busbars รวมกับ generator terminal busbars ทั้งหมด

$i = nb + 1, nb+ng-1$  แทน generator terminal busbars ซึ่งไม่รวม slack machine

$i = nb+ng$  แทน internal busbar ที่ slack machine

และใช้ *script* ต่อไปนี้แทนความหมายต่อไปนี้

reg แทน voltage regulator

int แทน internal busbar ของ generator

gen. แทน generator

Superscripts  $p, m$  แทนความหมายว่า 3 เฟสของบัสบาร์ใดๆ

สัญลักษณ์ที่สำคัญ

$[I]$  = เวกเตอร์ของ nodal injected current

$[V]$  = เวกเตอร์ของ nodal voltage  $[|V| e^j]$

$[Z]$  =  $[R] + j[X]$  = impedance matrix

$[Y]$  =  $[G] + j[B]$  = admittance matrix

$\Delta P_i^p$  = ค่าความแตกต่างระหว่างกำลังจริงที่กำหนดขึ้นกับกำลังจริงที่สามารถคำนวณได้ที่บัสบาร์  $i$  เฟส  $p$

$\Delta Q_i^p$  = ค่าความแตกต่างระหว่างกำลังเสมือนที่กำหนดขึ้นกับกำลังเสมือนที่สามารถคำนวณได้ที่บัสบาร์  $i$  เฟส  $p$

$(P_i^p)^p$  = ค่ากำลังจริงที่กำหนดขึ้น (specified real power) ที่บัส  $i$  เฟส  $p$

$(Q_i^p)^p$  = ค่ากำลังเสมือนที่กำหนดขึ้น (specified reactive power) ที่บัส  $i$  เฟส  $p$

$|V|_i^p$  = ขนาดของโวลเตจที่บัส  $i$  เฟส  $p$

$\theta_i^p - \theta_k^m$  = มุมระหว่างบัสบาร์  $i$  เฟส  $p$  และบัสบาร์  $k$  เฟส  $m$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $|V|_{int, j}^k$  = มุมของโวลเตจที่ internal busbar ของ generator ตัวที่  $j$
- $P_{gen, j}^s$  = specified total power output ของ generator ตัวที่  $j$
- $V_{reg, j}^p$  = voltage regulator specification สำหรับโวลเตจที่ขั้วทั้ง 3 เฟสของ generator ตัวที่  $j$

### 5.3 ปัญหาของสูตรคำนวณที่ใช้ใน three phase load flow ( Formulation of the three phase load flow problem )

#### แบบจำลองของ synchronous machine

synchronous machine ถูกออกแบบให้มีความสมมาตรกันของขดลวด และสามารถแทนด้วย sequence impedance ซึ่งจำเป็นสำหรับการวิเคราะห์ภาวะ steady state unbalance

การแทน generator ด้วย phase component สามารถหาได้จาก sequence impedance matrix  $(Z_g)_{012}$  ตามลำดับดังนี้

$$[Z_g]_{abc} = [T_s][Z_g]_{012}[T_s]^{-1} \quad (5.3.1)$$

$$= [T_s][Z_g]_{012}[T_s] \quad (5.3.2)$$

เมื่อ

$$[T_s]_{abc} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix}$$

(5.3.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ 'a' คือ complex operation  $e^{j2\pi/3}$  ดังนั้น phase component matrix จะเป็นดังนี้

$$[Z_g]_{abc} = \begin{bmatrix} Z_0+Z_1+Z_2 & Z_0+aZ_1+a^2Z_2 & Z_0+a^2Z_1+aZ_2 \\ Z_0+a^2Z_1+aZ_2 & Z_0+Z_1+Z_2 & Z_0+aZ_1+a^2Z_2 \\ Z_0+aZ_1+a^2Z_2 & Z_0+a^2Z_1+aZ_2 & Z_0+Z_1+Z_2 \end{bmatrix}$$

(5.3.4)

Phase component model ของ generator ได้แสดงไว้ในรูปที่ 5.1 (a) และ machine excitation จะมีความสมมาตรกันทั้ง 3 เฟส และ โวลเตจ internal หรือ excitation busbar จะเป็นแบบ 3 เฟส สมดุลย์ ดังนี้

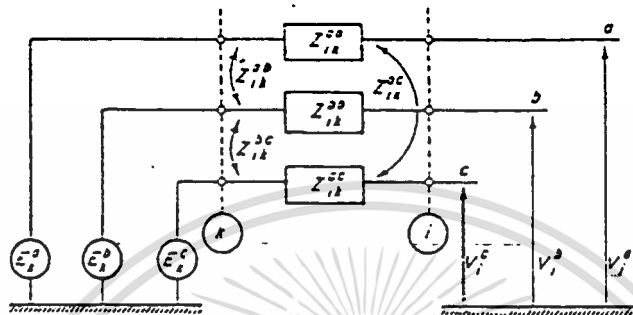
$$E_k^a = E_k^b = E_k^c \quad (5.3.5)$$

และ

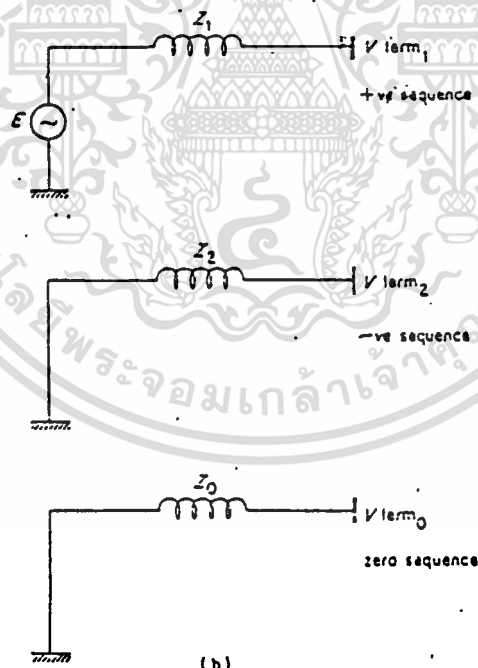
$$\theta_k^a = \theta_k^b + 2\pi/3 = \theta_k^c - 2\pi/3 \quad (5.3.6)$$

ในกรณีของ three phase load flow ค่า voltage regulator จะต้องคำนวณให้ละเอียด เพราะมีผลอย่างมากต่อการปฏิบัติงานในภาวะไม่สมดุลย์ เมื่อพิจารณารูป 5.1 (b) ค่า machine excitation จะเป็นแบบ 3 เฟส สมดุลย์ ซึ่งจะมีเพียง positive sequence voltage เท่านั้นที่มี internal busbar ในเทอมของ sequence voltage ค่า positive sequence voltage จะหาได้จากค่า excitation และค่า positive sequence voltage drop อันเนื่องมาจาก positive sequence current ที่ไหลผ่าน positive sequence reactance ส่วนค่า negative และค่า Zero sequence voltage จะมีผลน้อยมากต่อ excitation และ positive sequence impedance

บางครั้งก็มีข้อกำหนดในการรวม excitation ของ machine และ positive sequence reactance ของ machine เมื่อไรก็ตามที่ต้องการรู้ค่า excitation ของ machine (เช่นในการ fault) จึงจำเป็นต้องใช้ค่า positive sequence impedance



(a)



(b)

รูป 5.1 แบบจำลองของเครื่อง ซิงโครนิส

a) Phase component representation

b) Symmetrical component representation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติ three-phase load flow จะมีการกำหนดให้ positive sequence reactance มีค่าเล็กน้อย เพื่อที่จะลด excitation voltage ให้มีขนาดใกล้เคียงกับ system voltage โดยการลดมุมระหว่าง internal busbar และ terminal busbar

ดังนั้นในการสร้างแบบจำลอง generator โดยใช้สมการ (5.3.4) ค่าที่กำหนดขึ้นมักใช้เป็น  $Z_1$  แต่ในขณะที่ค่าจริงๆ จะเป็น  $Z_0$  และ  $Z_2$  และค่าความสูญเสียอันเนื่องมาจากการไม่สมดุลจะถือว่าน้อยมาก

Nodal admittance matrix มีความสัมพันธ์กับ inject current ของ generator บั๊สบาร์กับ nodal voltage หาได้โดยการอินเวอร์สเมตริกซ์ของ series impedance matrix ที่หาจากสมการ (5.3.4)

### การกำหนดค่าตัวแปร ( specified variable )

ตัวแปรต่อไปนี้จะต้องกำหนดให้มีค่าต่ำสุดและเพียงพอสำหรับใช้ในระบบ 3 เฟส ภายใต้ภาวะ steady state

-ขนาดโวลเตจของ slack generator internal busbar  $V_{int,j}$  เมื่อ  $j = nb + ng$   
( กำหนดให้มุม  $\theta_{int,j}$  เป็นมุมอ้างอิง )

-ขนาดของโวลเตจ internal busbar  $V_{int,j}$  และมุม  $\theta_{int,j}$  ที่ generator ตัวอื่น  
คือ  $j = nb + 1, nb + ng - 1$

-ขนาดของโวลเตจทั้ง 3 เฟส ( $V^p$ ) และมุม ( $\theta^p$ ) ของ generator terminal busbar  
และ Load busbar ทุกตัวในระบบคือ  $i = 1, nb$  และ  $p = 1, 3$

มีเพียง 2 ตัวแปรเท่านั้น ที่สัมพันธ์กับ generator internal busbar และถ้า 3 เฟสสมดุล แล้วไม่มีความจำเป็นที่ต้องรักษาโวลเตจและมุมเป็นตัวแปรอีก สมการที่ใช้สำหรับหาคำตอบต่างๆ จะหาได้จากเงื่อนไขต่อไปนี้

-รวมค่ากำลังจริงและกำลังเสมือนที่โหลดทั้งหมดในแต่ละ System busbar

-กำหนดค่า Voltage regulator สำหรับ synchronous ทุกๆ ตัว

-รวมค่ากำลังจริงที่จ่ายออกจาก Synchronous machine แต่ละตัว โดยไม่รวม slack machine ทำโดยกำหนดให้โวลเตจแต่ละเฟสมีขนาดคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Derivation of equation

ระบบ 3 เฟส โดยทั่วๆ ไปเขียนได้อยู่ในรูปสมการดังนี้

$$[I] - [Y][V] = 0$$

และ system admittance matrix  $[Y]$  ดังได้กล่าวมาแล้ว ซึ่งแต่ละเฟสจะมีทั้ง inductive และ capacitive mutual coupling ระหว่างเฟส และตัววงจรเอง โดยใช้วิธีทางคณิตศาสตร์

จะหาสมการของ system admittance matrix ได้คือ

$$[Y] = [G] + j[B] \quad (5.3.7)$$

โดยที่

1. สำหรับแต่ละเฟส (p) ที่มีโหลดต่ออยู่ และที่ generator terminal busbar (i)

$$\begin{aligned} \Delta P_i^p &= (P_i^p)^{sp} - P_i^p \\ &= (P_i^p)^{sp} - V_i^p \sum_{k=1}^m \sum_{m=1}^3 V_k^m [G_{ik}^{pm} \cos \theta_{ik}^{pm} + B_{ik}^{pm} \sin \theta_{ik}^{pm}] \end{aligned} \quad (5.3.8)$$

$$\begin{aligned} \Delta Q_i^p &= (Q_i^p)^{sp} - Q_i^p \\ &= (Q_i^p)^{sp} - V_i^p \sum_{k=1}^m \sum_{m=1}^3 V_k^m [G_{ik}^{pm} \sin \theta_{ik}^{pm} - B_{ik}^{pm} \cos \theta_{ik}^{pm}] \end{aligned} \quad (5.3.9)$$

2. สำหรับ generator j แต่ละตัว

$$(V_{reg})_j = f(V_k^1, V_k^2, V_k^3) \quad (5.3.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
เมื่อ k เป็นบัสบาร์ของ generator ตัวที่ j  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สำหรับ generator  $j$  ทุกตัว ซึ่งไม่รวม Slack machine คือ  $j$  ไม่เท่ากับ  $nb + ng$

$$(\Delta P_{gen})_j = (P_{gen}^{sp})_j - (P_{gen})_j \quad (5.3.11)$$

$$= (P_{gen}^{sp})_j - \sum_{m=1}^n V_{int, j} \sum_{k=1}^n V_k^m [G_{jk}^{pm} \cos \theta_{jk}^{pm} + B_{jk}^{pm} \sin \theta_{jk}^{pm}]$$

แม้ว่า summation ของ  $k$  จะครอบคลุม system busbar ทั้งหมดแต่ mutual term  $G_{jk}$  และ  $B_{jk}$  จะไม่เป็นศูนย์เมื่อ  $k$  เป็น terminal busbar ของ generator ตัวที่  $j$  สังเกตว่าค่า real power specified สำหรับ generator จะเป็นผลรวมของ real power ที่ terminal หรือ excitation ขณะที่ในทางปฏิบัติปริมาณเหล่านี้คือ กำลังที่ไหลออกจาก terminal busbar ด้วยผลอันนี้จะทำให้ไม่คิดค่าความสูญเสียใน generator

ค่าความสูญเสียของ generator จะมีผลน้อยมากต่อการทำงานของระบบ และสามารถ คำนวณได้จาก sequence admittance ในตอนท้ายของการคำนวณ ผลรวมของกำลังที่ไหลออกจาก generator อาจคำนวณได้เนื่องจาก generator internal busbar ไม่ถูกต้องกับ อิลิเมนต์อื่นๆ ในระบบ การหาค่าตอบของ โหลดโฟลว์ คือการพยายามทำให้ด้านซ้ายของสมการ (5.3.8) ถึง (5.3.11) มีค่าเป็นศูนย์

#### 5.4 อัลกอริทึมของ Fast decouple แบบสามเฟส

รูปมาตรฐานของ Newton-Raphson algorithm สามารถที่จะนำไปใช้ในการแก้สมการ (5.3.8) ถึง (5.3.11) คือการใช้ iteration ในการหาค่าตอบของสมการ matrix ดังนี้

$$\begin{array}{rcccl} \Delta P & A & E & I & M & \Delta \theta \\ \Delta P_{gen} & = & B & F & J & N & \theta_{int} \\ \Delta Q & & C & G & K & P & \Delta V/V \\ \Delta V_{reg} & & D & H & L & R & \Delta V_{int}/V_{int} \end{array} \quad (5.4.1)$$

เมตริกซ์ทางด้านขวามือของสมการ (5.4.1) คือ Jacobian matrix ของ first order partial derivatives

จากกรณีของ decouple single-phase load flow ผลของ  $\Delta\theta$  ที่มีต่อ reactive power และ ผลของ  $\Delta V$  ที่มีต่อ real power สามารถละทิ้งได้ สมการ (5.4.1) อาจจะถูกดัดแปลงโดย

$$[I] = [M] = [J] = [N] = 0$$

และ

$$[C] = [G] = 0$$

โดยที่ค่า voltage regulator specified จะถูกสมมติให้อยู่ในรูปของ terminal voltage magnitude ดังนั้น

$$[D] = [H] = 0$$

สมการ (5.4.1) อาจเขียนใหม่ในรูป decouple ได้ดังนี้

$$\begin{aligned} \Delta P_i^p &= A E \Delta\theta_k^m \\ \Delta P_{gen,j} &= B F \Delta\theta_{ml}^m \end{aligned} \quad (5.4.2)$$

สำหรับ  $i, k = 1, nb$  และ  $j, l = 1, ng - 1$  ( ไม่รวม slack generator )

$$\Delta Q_i^p = K P \Delta V_k^m / V_k^m \quad (5.4.3)$$

$$\Delta V_{reg,j} = L R \Delta V_{int l} / V_{int l}$$

สำหรับ  $i, k = 1, nb$  และ  $j, l = 1, ng - 1$  ( รวม slack generator ) พิจารณา Jacobian submatrix ในการคำนวณตั้งแต่สมการ (3.8) ถึง (3.11) จะต้องจำไว้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ  $V_1^1 = V_1^2 = V_1^3 = V_{int}^1$  และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\theta_1^1 = \theta_1^2 - 2\pi/3 = \theta_1^3 + 2\pi/3 = \theta_{int}$$

เมื่อ 1 คือ ตัวที่บอกถึง generator internal busbar สัมประสิทธิ์ของสมการเมตริกซ์ (5.3.2) คือ

$$[A^{pm}_{ik}] = [\partial \Delta P^p / \partial \theta^m_k]$$

$$A^{pm}_{ik} = V_i^p V_k^m [G^{pm}_{ik} \sin \theta^{pm}_{ik} - B^{pm}_{ik} \cos \theta^{pm}_{ik}]$$

นอกจาก

$$A^{nm}_{kk} = -B^{nm}_{kk} (V_k^m)^2 - Q_k^m$$

$$[B^{m}_{jk}] = [\partial \Delta P_{gen,j} / \partial \theta^m_k]$$

$$= \sum_{p=1} V_{int,p} V_k^m [G^{pm}_{jk} \sin \theta^{pm}_{jk} - B^{pm}_{jk} \cos \theta^{pm}_{jk}]$$

$$[E^p_{ii}] = [\Delta P^p / \theta_{int,i}]$$

$$= \sum_{p=1} V_{int,p} V_i^p [G^{pm}_{ii} \sin \theta^{pm}_{ii} - B^{pm}_{ii} \cos \theta^{pm}_{ii}]$$

$$[F_{ji}] = [\partial P_{gen,j} / \partial \theta_{int,i}]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $[F_{jj}] = 0$  สำหรับทุกที่ ๆ  $j \neq 1$  เพราะว่า generator ตัวที่  $j$  ไม่ถูกต้องเข้ากับ internal busbar ของ generator ตัวที่ 1

$$[F_{jj}] = \sum_{p=1}^n (-B_{jj}^{pp} (V_{int,1})^2 - Q_j^p)$$

$$+ \sum_{m=1, m \neq p}^n (V_{int,1})^2 [G_{jj}^{pm} \sin \theta_{jj}^{pm} - B_{jj}^{pm} \cos \theta_{jj}^{pm}]$$

สัมประสิทธิ์ ของสมการเมตริกซ์ (5.4.3) คือ

$$-[k_{kk}^{pm}] = V_k^m [ \partial \Delta Q^p / \partial V_k^m ]$$

เมื่อ

$$k_{kk}^{pm} = V_k^m V_k^p [ G_{kk}^{pm} \sin \theta_{kk}^{pm} - B_{kk}^{pm} \cos \theta_{kk}^{pm} ]$$

นอกจาก

$$k_{kk}^{mm} = -B_{kk}^{mm} (V_k^m)^2 + Q_k^m$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$-[L_{jk}^m] = V_k^m [\partial \Delta V_{reg,j} / \partial V_k^m]$$

ให้  $[L_{jk}^m] = V_k^m [L_{jk}^m]'$  เมื่อ  $k$  คือ terminal busbar ของ generator ตัวที่  $j$

$$\text{และ } L_{jk}^m = 0$$

$$-[P_{jl}^p] = V_{int,l} [\partial \Delta Q_l^p / \partial V_{int,l}]$$

$$= V_{int,l} \sum_{m=1} V_l^p [G_{lm}^{pm} \sin \theta_{lm}^{pm} - B_{lm}^{pm} \cos \theta_{lm}^{pm}]$$

$$-[R_{jl}^p] = [\partial \Delta V_{reg,j} / \partial V_{int,l}]$$

$$= 0$$

สำหรับ  $j, l$  ซึ่งค่า voltage regulator specification ไม่ถูกรวมเข้ากับตัวแปร  $V_{int}$  แม้ว่าสมการข้างต้นจะซับซ้อน แต่ความหมายจะคล้ายคลึงกับ single phase Jacobian elements โดยทั่วไป

### การประมาณค่า Jacobian

การประมาณค่านี้จะคล้ายกับในกรณีของ single phase load flow โดยสามารถทำได้ดังนี้

1. ที่โหนดทั้งหมด ( เฟสทั้งหมดของบัสทุกบัส ) จะได้

$$Q_k^m \ll B_{kk}^{mm} (V_k^m)^2$$

2. ระหว่างโหนดที่ต่อกันและมีเฟสเดียวกันจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น  $\cos \theta_{ik}^{mm} \approx 1$  คือ  $\theta_{ik}^{mm}$  มีค่าน้อยมาก และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ

$$G_{ik}^{mm} \sin \theta_{ik}^{mm} \ll B_{ik}^{mm}$$

3. ยิ่งกว่านั้นถ้า ค่า phase-angle unbalance ที่บัสบาร์ใดๆ มีค่าน้อยจะสามารถประมาณค่า สำหรับระบบ 3 เฟส ได้ดังนี้

$$\theta_{kk}^{pm} \approx \pm 120^\circ \text{ สำหรับ } p \neq m$$

4. จากผลของข้อ 2 และ 3 มุมระหว่างเฟสที่ต่างๆ กัน ที่ต่อระหว่างบัสบาร์ จะมีค่าประมาณ  $120^\circ$  คือ

$$\theta_{jk}^{pm} \approx \pm 120^\circ \text{ สำหรับ } p \neq m$$

หรือ

$$\cos \theta_{jk}^{pm} \approx -0.5$$

และ

$$\sin \theta_{jk}^{pm} \approx \pm 0.866$$

ค่าเหล่านี้จะเปลี่ยนแปลงเป็นค่า phase shift  $30^\circ$  สำหรับการต่อหม้อแปลง 3 เฟสแบบ star-delta เปลี่ยนแปลงเร็วมาก และต่างไปจากมุม  $120^\circ$  เล็กน้อย ซึ่งทำให้การคอนเวอร์จ ของ phase unbalance ที่บัสบาร์ มีความช้ากว่า เมื่อเทียบกับ voltage magnitude และค่านุม

จากวิธีประมาณค่า Jacobian ใน ข้อ 1 ถึง 4 แทนค่า เหล่านี้ลงในสมการ (5.4.2) และ (5.4.3) จะได้

$$\Delta P_i^p = [V_i^p M_{ik}^{pm} V_k^m] [\sum_{m=1} V_i^p M_{jl}^{pm} V_{int,l}] \Delta \theta_k^m$$

$$\Delta P_{gen,j} = [\sum_{p=1} V_{int,j} M_{jk}^{pm} V_k^m] [\sum_{m=1} \sum_{p=1} V_{int,j} M_{jl}^{pm} V_{int,l}] \Delta \theta_{int,l}$$

(5.4.4)

$$\Delta Q_i^p = [V_i^p M_{ik}^{pm} V_k^m] [\sum_{m=1} V_i^p M_{il}^{pm} V_{int,l}] \Delta V_k^m / V_k^m$$

$$\Delta V_{reg,j} = V_k^m [L'] [0] \Delta V_{int,l} / V_{int,l}$$

(5.4.5)

เมื่อ

$$M_{kk}^{pm} = G_{kk}^{pm} \sin \theta_{kk}^{pm} - B_{kk}^{pm} \cos \theta_{kk}^{pm}$$

เมื่อ

$$\theta_{kk}^{mm} = 0$$

เอกสารนี้เป็นเอกสารที่  $\theta_{kk}^{mm} = 0$  รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\theta_{\mu}^{pm} \approx \pm 120^\circ \text{ สำหรับ } p \neq m$$

ทุกเทอมในเมตริกซ์ [M] เป็นค่าคงที่ ซึ่งหาได้จาก system admittance matrix เมตริกซ์ [M] เหมือนกับเมตริกซ์ [-B] นอกจากเทอม off-diagonal ซึ่งต่อโหนดที่เฟสต่างกันซึ่งเทอมเหล่านี้จะถูกตัดแปลงโดยมุม  $120^\circ$  ที่เพิ่มเข้าไปยังเทอม  $G_{\mu}^{pm} \sin \theta_{\mu}^{pm}$

การลดโครงสร้างของ Jacobian submatrix จะช่วยลดลดความซับซ้อนของโปรแกรมลงได้มาก จากที่ได้กล่าวมา Jacobian elements ในสมการ (5.4.4) และ (5.4.5) จะเป็นดังนี้

- ด้านซ้ายมือของสมการจะแปลงใหม่เป็น  $[\Delta P^p/V^p]_i, [\Delta P_{gen,j}^p/V_{int,j}^p], [\Delta Q^p/V^p]_i$
- ในสมการ (5.4.4) ทางด้านขวามือจะเป็นค่าใน V เทอมให้เป็น 1 p.u.
- ในสมการ (5.4.5) ทางด้านขวามือ จะยกเลิกเทอม V ทางด้านขวา แต่เหลือค่าเวกเตอร์ ไว้

เท่านั้น

จากการตัดแปลงจะได้สมการใหม่ดังนี้

$$\begin{matrix} \Delta P^p_i/V^p_i \\ \vdots \\ \Delta P^p_{gen,j}/V_{int,j}^p \end{matrix} \begin{matrix} M_{\mu}^{pm} \\ \\ \sum_{p=1} M_{jk}^{pm} \end{matrix} \begin{matrix} \sum_{n=1} M_{nl}^{pm} \\ \sum_{p=1} \sum_{m=1} M_{jl}^{pm} \end{matrix} \begin{matrix} \Delta \theta_{\mu}^p \\ \\ \Delta \theta_{int,l} \end{matrix} \quad (5.4.6)$$

$$\begin{matrix} \Delta Q^p_i/V^p_i \\ \\ \end{matrix} \begin{matrix} M_{\mu}^{pm} \\ \\ \sum_{m=1} M_{nl}^{pm} \end{matrix} \begin{matrix} \sum_{n=1} M_{nl}^{pm} \\ \\ \end{matrix} \begin{matrix} \Delta V_{\mu}^m \\ \\ \end{matrix}$$

$$= \begin{matrix} \Delta V_{reg}^p \\ \\ \end{matrix} \begin{matrix} [L_{\mu}^m]^T \\ \\ 0 \end{matrix} \begin{matrix} \Delta V_{int,l} \end{matrix} \quad (5.4.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

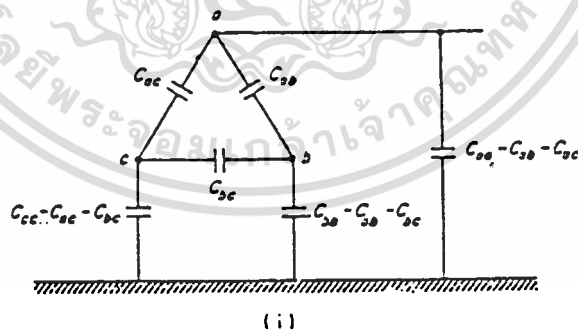
และ  $[L_{\mu}^m] = [\partial \Delta V_{\mu} / \partial V^m]$  ขณะที่  $V_{\mu}$  จะมีลักษณะเชิงเส้น ตาม terminal voltage ดังนั้น  $[L]$  จะเป็นเมตริกซ์ที่คงที่

เนื่องจาก Jacobian matrix  $[B']$  และ  $[B'']$  ในสมการ (5.4.6) และ (5.4.7) จะประมาณว่าคงที่ได้ Zero diagonal element ในสมการ (5.4.7) เป็นผลมาจากการจัดเรียงสมการและตัวแปร

ค่า capacitance matrix และลักษณะของมันได้แสดงไว้ในรูปที่ 5.2 ซึ่งเป็นกรณีของ single three phase line ถ้ามี capacitance  $n$  ตัว coupling กับสายส่งที่ขนานกัน จะเขียนเป็นเมตริกซ์ขนาด  $3n \times 3n$

ในกรณีของ single-phase load flow ค่า shunt capacitance เป็นค่า positive sequence capacitance ซึ่งหาได้ทั้งแบบ phase-to-phase และ phase-to-earth ของสาย ค่า shunt capacitance matrix จะมีผลต่อ MVAR ที่ไหล เมื่อพิจารณาเหมือน single phase fast decoupling ซึ่งจะไม่คิดค่า shunt capacitance matrix ในสูตรของ  $[B']$  ซึ่งจะเป็นการเพิ่มอัตราकरणคอนเวอร์จของกำลังจริง

ค่า capacitance coupling three phase lines มีผลทำให้ interline capacitance จะมีค่าน้อยมากเมื่อเปรียบเทียบกับ self capacitance ของเฟส ผลจาก tap คือค่า 3 ซึ่งเกิดจากการต่อหม้อแปลงแบบ Y-Δ ซึ่งถูกรวมเข้าในขณะหาค่าเมตริกซ์  $[B']$

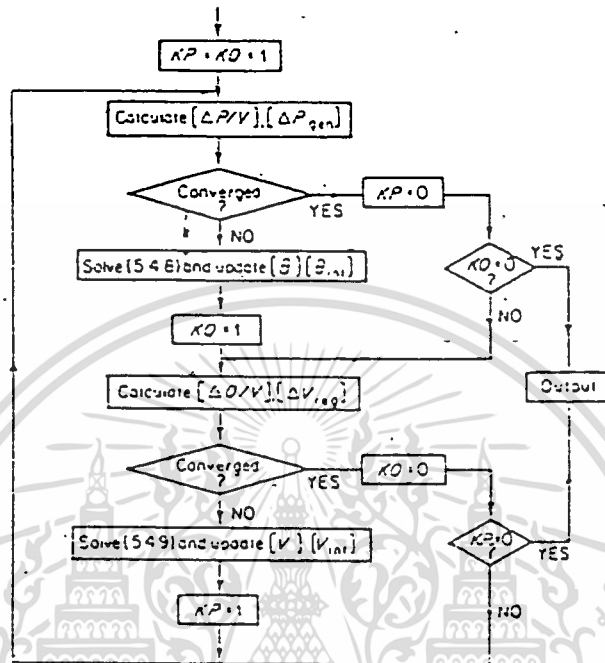


(ii)

$C_{aa}$	$-C_{ab}$	$-C_{ac}$
$-C_{ba}$	$C_{bb}$	$-C_{bc}$
$-C_{ca}$	$-C_{cb}$	$C_{cc}$

รูป 5.2 เมตริกซ์ ของ คาปาซิเตอร์ แบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 5.3 ลำดับการทำซ้ำ สำหรับการศึกษาโหลดไหลว่งของระบบไฟฟ้าสามเฟส

โดยการดัดแปลงสมการดังที่กล่าวมาแล้ว จะได้ algorithmic 2 สมการสุดท้ายเป็น

$$\Delta P/V = B'_m \Delta \theta \quad (5.4.8)$$

$$\Delta P_{gen}/V_{int} \quad \Delta \theta_{int}$$

$$\Delta Q/V = B''_m \Delta V \quad (5.4.9)$$

$$\Delta V_{reg} \quad \Delta V_{int}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่ Jacobian  $[B'_{ij}]$  และ  $[B''_{ij}]$  จะมีค่าคงที่ ความชันของพื้นที่ผิวขนาดหลายมิติ โดยใช้ฟังก์ชันทางด้านซ้ายมือของสมการ

สมการ (5.4.8) และ (5.4.9) หากคำตอบได้โดยการ iteration ตามลำดับ ดังในรูป 5.3

### Generator model และ fast decoupled algorithm

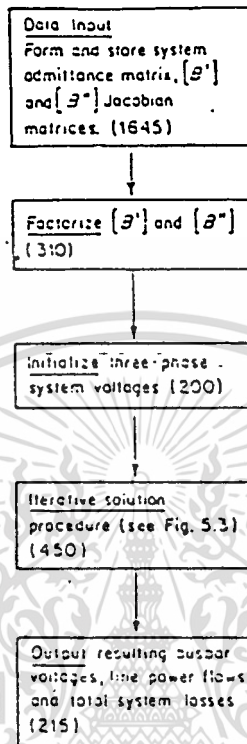
การหา fast decoupled algorithm โดยใช้ข้อสมมุติต่างๆ ที่ทำให้ Jacobian matrix มีค่าคงที่ และข้อสมมุติฐานนี้ยังใช้พิจารณาถึง excitation ของ generator ใน system busbar โดยไม่คำนึงถึง Voltage magnitudes และมุมระหว่างบัสบาร์ ซึ่งมีผลกับโหลดของเครื่องจักร และ positive sequence reactance

อัตราคอนเวอร์จที่แย่งและข้อจำกัดของโหลดที่ติดต่อกับ generator อาจจะมีผลเสียโดยตั้งค่า positive sequence reactance ของ generator ให้มีค่าต่ำ ( ประมาณ 0.01 p.u. บนพิกัดของเครื่อง ) โดยที่จะไม่มีผลของความสูญเสียมาเกี่ยวข้องด้วย

### โครงสร้างของโปรแกรมคอมพิวเตอร์ ( Structure of the computer program )

#### ข้อมูลอินพุต ( Input data )

ข้อมูลอินพุต จะใช้ข้อมูลดิบซึ่งได้แสดงในตัวอย่าง และจะนำข้อมูลดิบเหล่านี้ไปใช้เพื่อหา system admittance matrix ซึ่งก็คือ  $B'$  และ  $B''$  นั้นเอง โดยทั้ง system admittance matrix และ Jacobian matrix จะถูกเก็บโดยใช้เทคนิคแบบ sparsity ซึ่งทำให้เป็นโครงสร้างแบบ 3\*3 matrix block ซึ่งมีประโยชน์มากใน three phase system matrix



รูป 5.4 โครงสร้างโปรแกรม

### การกำหนดค่าเริ่มต้น ( Starting values )

ค่าเริ่มต้นจะถูกกำหนดได้ดังนี้

- ค่า nonvoltage controlled busbar จะถูกกำหนดให้เป็น 1 p.u. ที่ทุกๆ เฟส
- ที่ generator terminal busbar ค่าโวลเตจทั้งหมด จะถูกกำหนดตามค่า voltage regulator specification
- มุมของ system busbar ทั้งหมดจะถูกกำหนดเป็น  $0, -120^{\circ}, 120^{\circ}$  สำหรับทั้ง 3 เฟสตามลำดับ
- ค่า generator internal voltage และมุมของมัน จะถูกคำนวณ specified real power โดยปราศจากการประมาณค่าที่คิดว่าเป็น โดยการสมมุติให้ค่า reactive power = 0 สำหรับ slack machine ค่า real power จะถูกประมาณให้เป็นผลต่าง ระหว่างค่าของโวลเตจทั้งหมด และค่า generator ทั้งหมด รวมกับค่าเปอร์เซ็นต์ของความสูญเสียที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับบางกรณี การคอนเวอร์จอาจจะช้าหรือยาก เราจะแก้ไขปัญหานี้โดยใช้ Single phase load flow เพื่อหาค่าเริ่มต้น ซึ่งค่าเหล่านี้ภายใต้ภาวะ steady state unbalance จะเป็นการประมาณที่ดีของทั้งโวลเตจ และ มุมของ generator internal และจะถูกคำนวณโดย single phase real และ reactive power และโดยปกติการ iteration ของกรณี 3 เฟสจะมีความซับซ้อนกว่าของ 1 เฟสเสมอ

### Iteration solution

{ กระบวนการ iteration รูป (5.3) จะเป็นส่วนสำคัญในการหาค่าต่างๆ ทั้ง system voltage, load และ generation ได้ทุกค่า

### ผลที่ได้ ( Output results )

ค่า three phase busbar voltage, line power flow และ total system loss ที่คำนวณได้จะแสดงผลออกมาโดยการพิมพ์

### ลักษณะของอัลกอริทึม ( Performance of algorithm )

ในหัวข้อนี้จะกล่าวถึงลักษณะต่างๆ ที่มีผลต่อการ convergence กล่าวคือลักษณะของ three phase algorithm จะถูกทดสอบทั้งในภาวะ balanced และ unbalanced และจะมีการเปรียบเทียบกับกรณี single phase algorithm ด้วย

### ลักษณะภายใต้ภาวะสมดุลย์ ( Performance under balanced nonditions )

ระบบ 3 เฟสสมดุลย์ที่ปฏิบัติงานภายใต้ ภาวะ balanced load สามารถจะใช้แบบจำลองที่มีแต่ positive sequence และใช้ได้ทั้ง three phase และ single phase load flow ในการวิเคราะห์ระบบ ภายใต้เงื่อนไขนี้ เราสามารถเปรียบเทียบลักษณะของ three phase และ single phase fast decoupled algorithm ได้

โดยที่ three phase system transmission line จะถูกแทนด้วยเมตริกซ์ขนาด  $3 \times 3$  และหม้อแปลงจะถูกแทนด้วย แบบจำลองที่มีพารามิเตอร์สมดุลกันทุกๆ เฟส และ generator จะถูกแทนด้วยแบบจำลองที่มีพารามิเตอร์ตาม sequence impedance

โดยทั่วไป จำนวน iteration ที่ใช้ในการคอนเวอร์จ ทั้งแบบ single phase และ three phase algorithm ได้แสดงไว้ในตารางที่ 1 ลักษณะต่างๆ เช่น การต่อหม้อแปลง และ negative และ Zero sequence impedance จะไม่มีผลต่อการ converge ของระบบ 3 เฟสภายใต้ ภาวะ มีแต่ค่า positive sequence network เท่านั้น ที่มีกำลังไหลผ่านและไม่มีการ coupling ระหว่าง sequence network ทั้ง negative และ Zero sequence ในระบบ 3 เฟส ในภาวะสมดุล จะไม่มีผลต่อการปฏิบัติงานและลักษณะของ algorithm

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

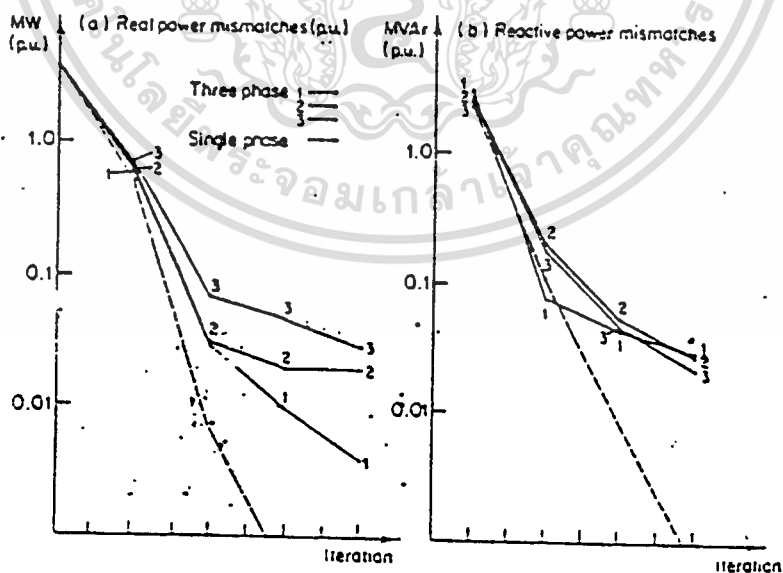
Case	Number of busbars	Single-phase load flow	Balanced three phase load flow		Typical three-phase unbalance
			$\mu$	$\Delta$	
1	5	4,3	4,3	4,3	6,6
2	6	3,3	3,3	3,3	8,8
3	14	3,3	3,3	3,3	6,5
4	17	3,3	3,3	3,3	8,7
5	30	3,3	3,3	3,3	6,6

Convergence tolerance is 0.1 MW/MVAR. The numerical results, (i,j), should be interpreted as follows:

i—refers to the number of real power-angle update iterations.

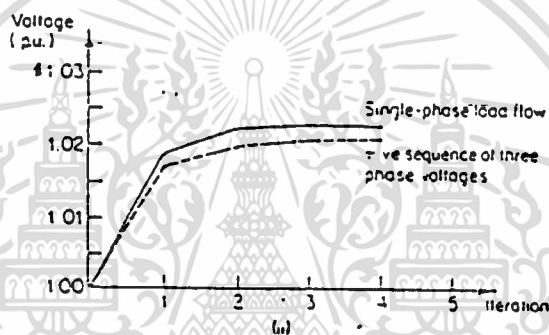
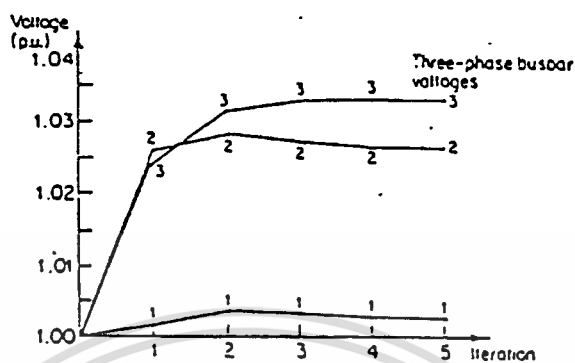
j—refers to the number of reactive power-voltage update iterations.

### ตารางที่ 5.1 ผลของการลู่เข้า



รูป 5.5 การลู่เข้าของกำลังไฟฟ้า สำหรับการแก้ปัญหาโหลดโพลาร์ทั้งแบบสามเฟส

เอกสารนี้เป็นเอกสารและแบบแปลนเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 5.6 การลู่เข้าของศักดาไฟฟ้า สำหรับการแก้ไหลดโวลท์ทั้งแบบสามเฟสและแบบเฟสเดียว

(i) Three phase voltages

(ii) Single-phase and three-phase positive sequence voltages

### ลักษณะเมื่อเป็นระบบไม่สมดุลย์

จำนวน iteration เพื่อให้ convergence สำหรับระบบอันเดียวกัน แต่อยู่ในภาวะ steady state unbalance ที่ได้แสดงไว้ในตาราง 5.1 ซึ่งการคอนเวอร์จ แบบนี้ จะแย้งว่าแบบ balanced และจะต้องใช้การ iteration มากขึ้นอีก 2-3 ครั้ง

กราฟในรูป 5.5 แสดงถึงการเริ่มต้นการคอนเวอร์จ ของระบบ 3 เฟส ซึ่งมีค่าใกล้เคียงกับกรณี single phase load flow แต่อย่างไรก็ตาม ขณะที่เข้าใกล้ค่าตอบการคอนเวอร์จของระบบ 3 เฟส จะเริ่มช้ากว่า แม้ว่าโวลเตจและมุมที่ unbalance จะเริ่มตั้งแต่การ iteration ครั้งแรก ซึ่งมันจะมีผลต่อการ

คอนเวอร์จบ้าง จนกว่า positive sequence power flow จะเข้าใกล้การคอนเวอร์จ ไปใช้ประโยชน์ด้านการคำนวณ ไม่ว่ารณิติใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของการคอนเวอร์จ three phase voltage ได้แสดงไว้ในรูปที่ 5.6 (i) ลักษณะการคอนเวอร์จ ของ positive sequence component ของ unbalanced voltage แสดงในรูป 5.6 (ii) เมื่อเทียบกับการคอนเวอร์จของ single phase load flow และจากรูปจะพบว่า positive sequence voltage ของ three phase unbalanced load flow จะมีลักษณะการคอนเวอร์จใกล้เคียงกับกรณีของ single-phase fast-decoupled load flow

การคอนเวอร์จของระบบ unbalance บางครั้งจะช้า แต่มีความยืดหยุ่นดี ลักษณะที่เกี่ยวข้องกับ three phase load flow และอิทธิพลของมันที่มีต่อการคอนเวอร์จที่น่าสนใจมีดังนี้

- ความไม่สมมาตร ( asymmetry ) ของ system parameters
- unbalance ของ system loading
- ผลของการต่อหม้อแปลง
- Mutual coupling ระหว่าง สายส่งที่ขนานกัน



## บทที่ 6

## บทสรุปและวิจารณ์

การคำนวณโหลดโพลีสามารถจำแนกได้ 2 วิธี คือ

1. โหลดโพลีแบบเฟสเดียว ซึ่งเป็นการคำนวณโหลดโพลีในสถานะที่ระบบไฟฟ้ากำลังแบบสามเฟสมีลักษณะที่สมดุล โดยที่โหลดของระบบจะต้องมีความสมดุลด้วย ซึ่งการคำนวณแบบนี้ประกอบด้วยหลายวิธี ได้แก่

1.1 Gauss-Saidel method

1.2 Newton-Raphson method

1.3 Fast decouple method

การคำนวณในวิธีต่าง ๆ ข้างต้น จะมีลักษณะที่แตกต่างกันทั้งในด้านขั้นตอน สมการคำนวณ และข้อกำหนดเพิ่มเติมต่างๆ ส่งผลให้แต่ละวิธีมีข้อแตกต่างกันคือ

1. ความเร็วในการประมวลผล

สมการของวิธี Fast decouple จะมีรูปแบบที่ง่ายและ ไม่มีความยุ่งยากสลับซับซ้อน ส่งผลทำให้การคำนวณเป็นไปอย่างรวดเร็วและใช้เวลาในการคำนวณน้อยในแต่ละ iteration

2. การใช้หน่วยความจำ

ค่า Jacobian ในสมการของวิธี Fast decouple จะมีการจัดเก็บค่าสมาชิกเฉพาะส่วนที่เป็น upper triangular ดังนั้นจะทำให้ ปริมาณหน่วยความจำที่ถูกใช้จะมีขนาดลดลงอย่างมากเมื่อเทียบกับวิธีอื่นๆ

3. ความสามารถในการวิเคราะห์

วิธีของ Fast decouple เป็นวิธีที่มีสมการในการคำนวณที่ง่าย และ ไม่มีความสลับซับซ้อนมากนัก ทำให้การศึกษาและวิเคราะห์กระทำได้ง่าย และนำไปสู่การเขียนโปรแกรมได้ง่ายและสะดวกยิ่งขึ้น ก่อให้เกิดประสิทธิภาพในการทำงานสูงขึ้นด้วย

ดังนั้นในการคำนวณโหลดไฟลว์จะใช้วิธีของ Fast decouple เป็นพื้นฐานในการคำนวณทั้งในกรณีโหลดไฟลว์แบบเฟสเดียวและแบบสามเฟส

**2. โหลดไฟลว์แบบสามเฟส** ซึ่งเป็นการคำนวณโหลดไฟลว์ในสถานะของระบบไฟฟ้ากำลังแบบสามเฟสที่ระบบมีลักษณะไม่สมดุล โดยในที่นี้จะทำการพิจารณาเฉพาะเมื่อระบบรับภาระโหลดไม่สมดุลและองค์ประกอบอื่น ๆ ของระบบมีลักษณะที่สมดุลด้วย โดยจะไม่พิจารณาในส่วนของหม้อแปลงซึ่งใช้วิธีของ Fast decouple ที่มีอัลกอริทึมแบบสามเฟส

ในการคำนวณ

ข้อแตกต่างของวิธีการคำนวณโหลดไฟลว์แบบเฟสเดียวกับแบบสามเฟส คือ

1. การใช้หน่วยความจำ

การคำนวณโหลดไฟลว์แบบเฟสเดียว จะมีการใช้หน่วยความจำที่น้อยกว่าการคำนวณโหลดไฟลว์แบบสามเฟสเนื่องจากมีใช้ข้อมูลในการคำนวณที่น้อยกว่า

2. ความเร็วในการประมวลผล การคำนวณโหลดไฟลว์แบบเฟสเดียว จะมีความเร็วในการประมวลผลเร็วกว่าการคำนวณโหลดไฟลว์แบบสามเฟส เนื่องจากในกรณีของแบบเฟสเดียวมีข้อมูลน้อยกว่าและสมการที่ใช้ในการคำนวณยุ่งยากและสลับซับซ้อนน้อยกว่า

3. ความสามารถในการวิเคราะห์ การคำนวณโหลดไฟลว์แบบเฟสเดียว จะสามารถทำความเข้าใจได้ง่ายกว่า เนื่องจากมีความสลับซับซ้อนน้อยกว่าแบบสามเฟส

4. ลักษณะการลู่เข้า การคำนวณโหลดไฟลว์แบบสามเฟส จะมีการลู่เข้าที่ช้ากว่าแบบเฟสเดียว เนื่องจากผลของความไม่สมมาตรของโหลด

การคำนวณโหลดไฟลว์ทั้งสองแบบสามารถนำไปใช้ในการวางแผนและพัฒนาระบบไฟฟ้ากำลัง โดยที่การคำนวณโหลดไฟลว์แบบเฟสเดียวหรือแบบสามเฟสสมดุล จะนำไปใช้ในการวางแผนเพื่อพัฒนาระบบ ส่วนในกรณีของโหลดไฟลว์แบบสามเฟสไม่สมดุล จะนำไปใช้พิจารณาและวิเคราะห์ในขณะระบบกำลังปฏิบัติงาน ซึ่งทั้งสองวิธีจะสามารถช่วยให้ระบบไฟฟ้ากำลังทำงานได้อย่างมีประสิทธิภาพและมีประสิทธิภาพสูงสุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คู่มือการใช้โปรแกรมแบบต่อเฟส

### ลำดับการทำงานของโปรแกรม

1. จาก DOS PROMPT เลือก DIRECTORY ที่โปรแกรม THREE PHASE LOAD FLOW ANALYSIS อยู่ แล้วพิมพ์ IPHASE จะเป็นการเข้าโปรแกรม

2. ในรูปที่ 1. หน้าจอจะแสดงเมนูให้เลือกวิธีการคำนวณ โดยแบ่งเป็น

1. วิธีการคำนวณแบบเกาส์ไซเดล
2. วิธีการคำนวณแบบนิวตันราฟสัน
3. วิธีการคำนวณแบบคิป์เปิล
4. วิธีการคำนวณแบบฟาส-คิป์เปิล
5. เลิกการทำงาน

3. ในรูปที่ 2. จากหน้าจอให้เลือก คือ

1. Read old file
2. Write new file
3. Exit

จะทำการเลือกว่าจะทำรายการอะไร โดยที่

-เมื่อเลือกเลข 1 จะทำการอ่านข้อมูลเดิม

-เมื่อเลือกเลข 2 จะทำการเขียนข้อมูลใหม่

-เมื่อเลือกเลข 3 จะเลิกการทำงาน

4. ป้อนชื่อ File ที่ต้องการอ่านข้อมูลหรือเขียนข้อมูลใหม่

5. ป้อนจำนวนบัสที่มีอยู่ในระบบที่ต้องการศึกษา

6. ป้อนข้อมูลอิมพีแดนซ์ของสายส่งระหว่างบัสต่าง ๆ ในระบบที่ต้องการศึกษา โดยที่จะทำการป้อนข้อมูลในรูปของ  $R + jX$  ระหว่างบัสนั้นๆ

7. ป้อนข้อมูลเบื้องต้นเกี่ยวกับค่าโวลต์เดจของบัส, มุมของบัส, ค่ากำลังจริงและค่ากำลังเสมือน การคำนวณของบัสนั้นๆ โดยที่จะแยกทำการป้อนค่าประเภทของบัส คือ

เอกสารนี้เป็นการนำเอาเอกสารทุกครั้งที่มีการนำไปใช้

-เมื่อบัสที่เลือกคือ Slack bus จะทำการป้อนข้อมูลเบื้องต้น คือ ค่าโวลต์เดจของบัส และมุมของบัส

-เมื่อบัสที่เลือกคือ PV bus จะทำการป้อนข้อมูลเบื้องต้น คือ ค่าโวลต์เดจของบัส และค่ากำลังจริง

-เมื่อบัสที่เลือกคือ PQ bus จะทำการป้อนข้อมูลเบื้องต้น คือ ค่ากำลังจริงและค่ากำลังเสมือน

8. ป้อนค่าพื้นฐานของ MVA และขนาดโวลต์เดจพื้นฐาน

9. โปรแกรมจะทำการคำนวณหาค่าของโวลต์เดจของบัส, มุมของบัสและกำลังไฟฟ้าที่ส่งจ่ายในระบบ

10. หน้าจอจะแสดงผลลัพธ์ของโวลต์เดจของบัส, มุมของบัส, ค่ากำลังจริง, ค่ากำลังเสมือนของบัส และค่ากำลังที่ส่งจ่ายระหว่างบัสต่าง ๆ

หมายเหตุ

-โปรแกรม New\_fast เป็นโปรแกรมที่มีรูปแสดงความสัมพันธ์ของค่าแอดมิตแตนซ์ระหว่างบัสต่างๆ ภายในระบบที่ต้องการศึกษา ดังแสดงในรูปที่ 3.

## คู่มือการใช้โปรแกรมแบบสามเฟส

### ลำดับการทำงานของโปรแกรม

1. จาก DOS PROMPT เลือก DIRECTORY ที่โปรแกรม THREE PHASE LOAD FLOW ANALYSIS อยู่ แล้วพิมพ์ 3PHASE จะเป็นการเข้าโปรแกรม

2. ในรูปที่ 4. จากหน้าจอให้เลือก คือ

1. Read old file

2. Write new file ---

3. Exit

จะทำการเลือกจะทำรายการอะไร โดยที่

-เมื่อเลือกเลข 1 จะทำการอ่านข้อมูลเดิม

-เมื่อเลือกเลข 2 จะทำการเขียนข้อมูลใหม่

-เมื่อเลือกเลข 3 จะเลิกการทำงาน

ในกรณีที่เลือกเลข 1. หน้าจอจะแสดงข้อมูลที่อ่านจากแฟ้มข้อมูลเดิมทั้งหมด

ในกรณีที่เลือกเลข 2. ผู้ใช้จะต้องทำตามลำดับตั้งแต่ข้อ 4. จนถึงข้อ 7.

3. จากรูปที่ 5. ให้ทำการป้อนชื่อ File ที่ต้องการอ่านข้อมูลหรือเขียนข้อมูลใหม่

4. จากรูปที่ 5. ให้ทำการป้อนจำนวนบัสที่มีอยู่ในระบบที่ต้องการศึกษา

5. จากรูปที่ 5. ให้ทำการป้อนข้อมูลอิมพีแดนซ์ของสายส่งระหว่างบัสต่าง ๆ ในระบบที่ต้องการศึกษา โดยที่จะทำการป้อนข้อมูลในรูปของ  $G + jB$  ในแต่ละเฟสของบัสที่มีความสัมพันธ์กัน

6. จากรูปที่ 6. ให้ทำการเลือกประเภทของบัสว่าเป็น Slack bus, PV bus หรือ PQ bus

7. จากรูปที่ 7. ให้ทำการป้อนข้อมูลเบื้องต้นเกี่ยวกับค่าโวลต์เดจของบัส, มุมของบัส, ค่ากำลังจริงและค่ากำลังเสมือนของบัสนั้นๆ โดยที่จะแยกทำการป้อนตามประเภทของบัส คือ

-เมื่อบัสที่เลือกคือ Slack bus จะทำการป้อนข้อมูลเบื้องต้น คือ ค่าโวลต์เดจของบัส

และมุมของบัส

-เมื่อ巴士ที่เลือกคือ PV bus จะทำการป้อนข้อมูลเบื้องต้น คือ ค่าโวลต์เดจของ巴士 และค่ากำลังจริง

-เมื่อ巴士ที่เลือกคือ PQ bus จะทำการป้อนข้อมูลเบื้องต้น คือ ค่ากำลังจริงและค่ากำลังเสมือน

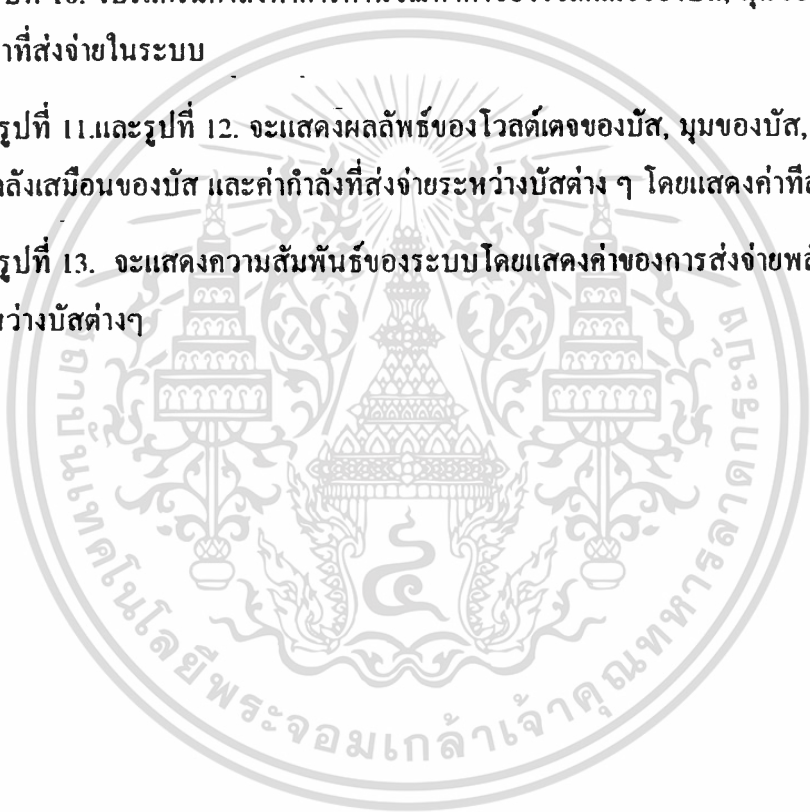
8. จากรูปที่ 8. ให้ทำการป้อนค่าพื้นฐานของ MVA และขนาดโวลต์เดจพื้นฐาน

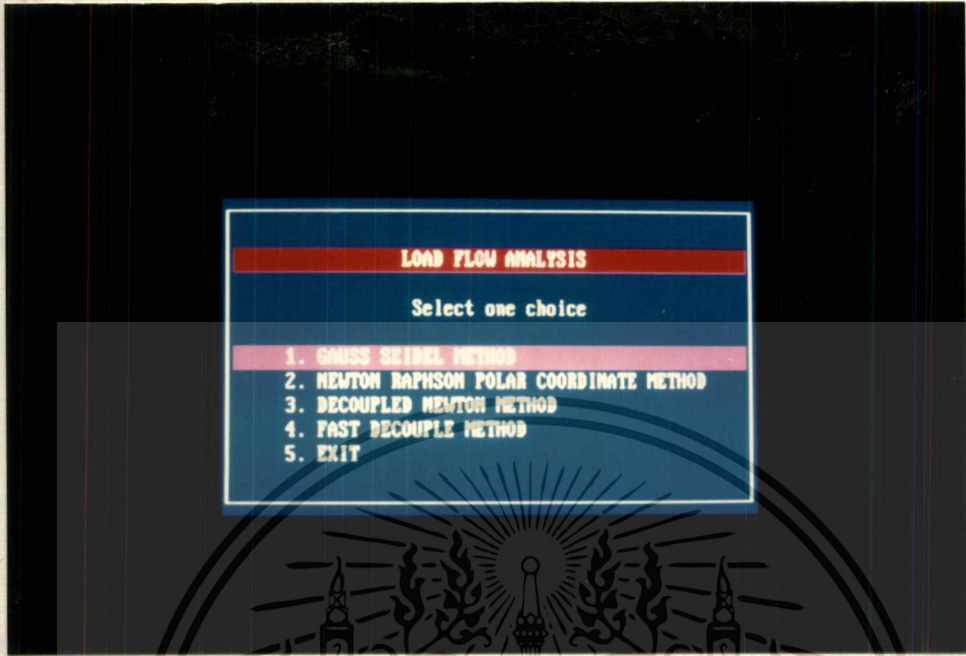
9. จากรูปที่ 9 จะแสดงความสัมพันธ์ของระบบโดยแสดงค่าของแอดมิตแดนซ์ระหว่าง巴士ต่างๆ

10. จากรูปที่ 10. โปรแกรมกำลังทำการคำนวณหาค่าของโวลต์เดจของ巴士, มุมของ巴士และกำลังไฟฟ้าที่ส่งจ่ายในระบบ

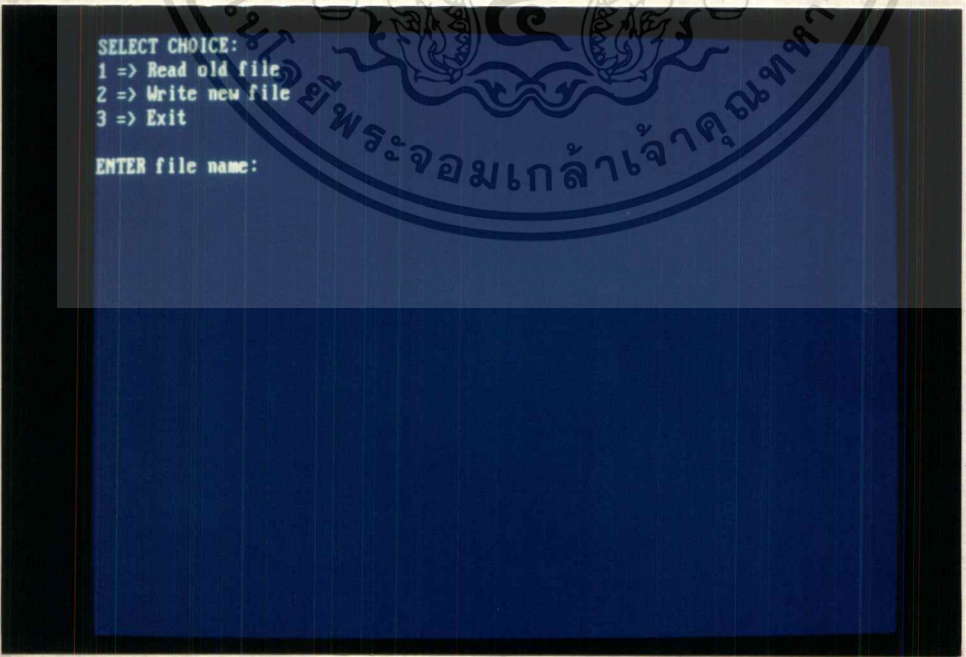
11. จากรูปที่ 11. และรูปที่ 12. จะแสดงผลลัพธ์ของโวลต์เดจของ巴士, มุมของ巴士, ค่ากำลังจริง, ค่ากำลังเสมือนของ巴士 และค่ากำลังที่ส่งจ่ายระหว่าง巴士ต่าง ๆ โดยแสดงค่าที่ละเฟส

12. จากรูปที่ 13. จะแสดงความสัมพันธ์ของระบบโดยแสดงค่าของการส่งจ่ายพลังงานไฟฟ้าระหว่าง巴士ต่างๆ





รูปที่ 1.

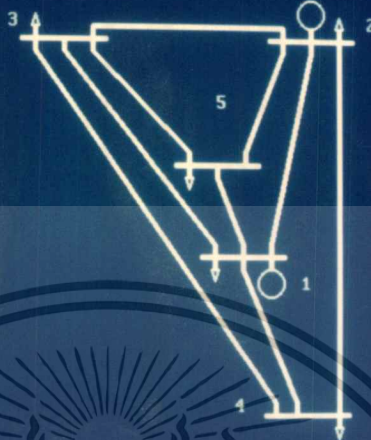


รูปที่ 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

•• THREE PHASE LOAD FLOW ••

$Z(3)(2) = 0.10 + j0.20$  P.U  
 $Z(3)(1) = 0.10 + j0.20$  P.U  
 $Z(2)(1) = 0.10 + j0.20$  P.U  
 $Z(3)(5) = 0.10 + j0.20$  P.U  
 $Z(2)(5) = 0.10 + j0.20$  P.U  
 $Z(1)(5) = 0.10 + j0.20$  P.U  
 $Z(3)(4) = 0.10 + j0.20$  P.U  
 $Z(2)(4) = 0.10 + j0.20$  P.U  
 $Z(1)(4) = 0.10 + j0.20$  P.U



รูปที่ 3.

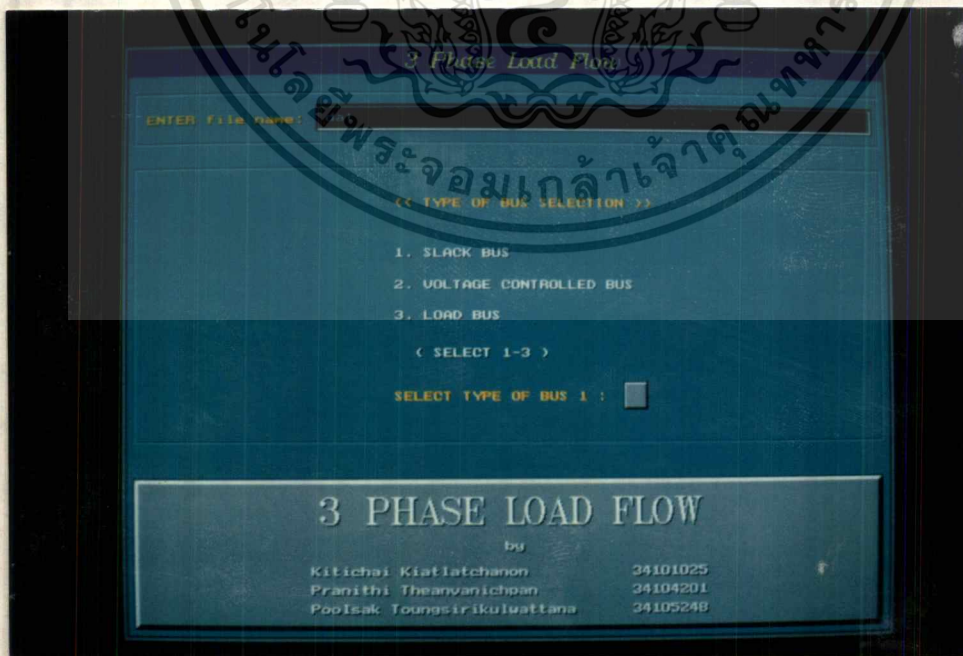
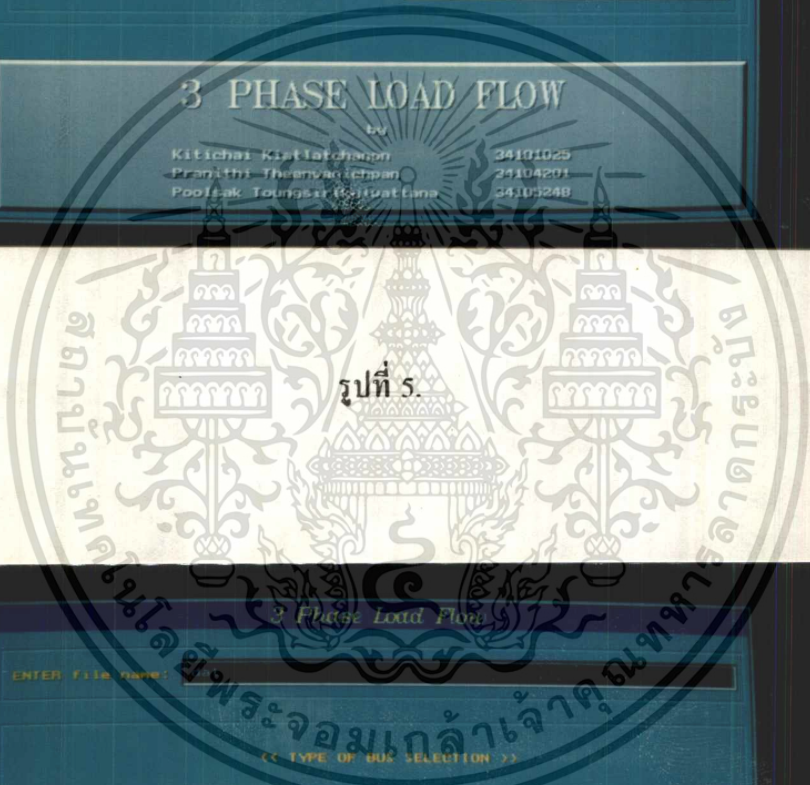
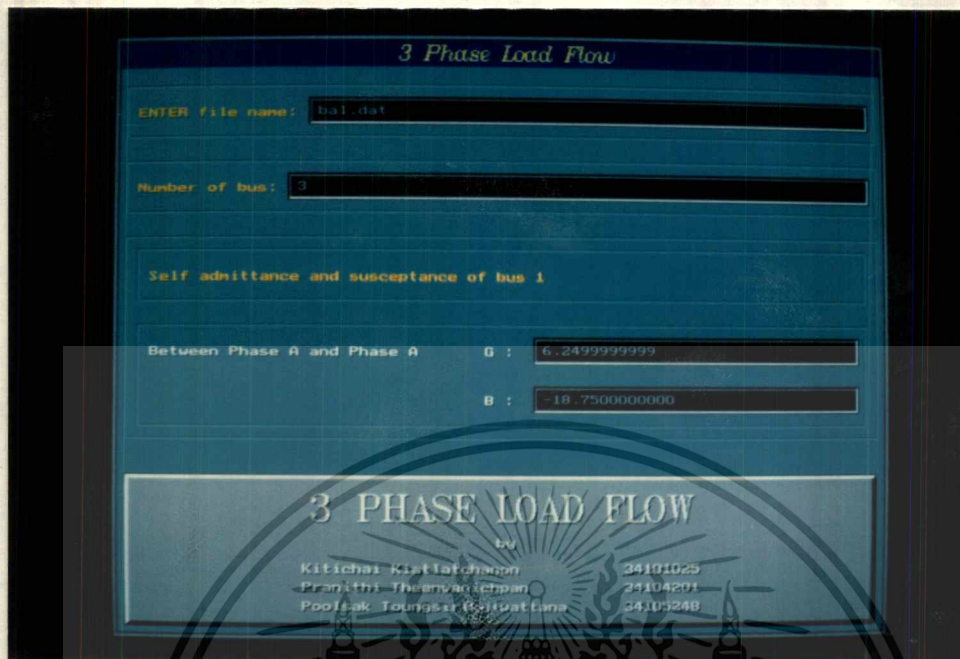
3 Phase Load Flow

Please select one :

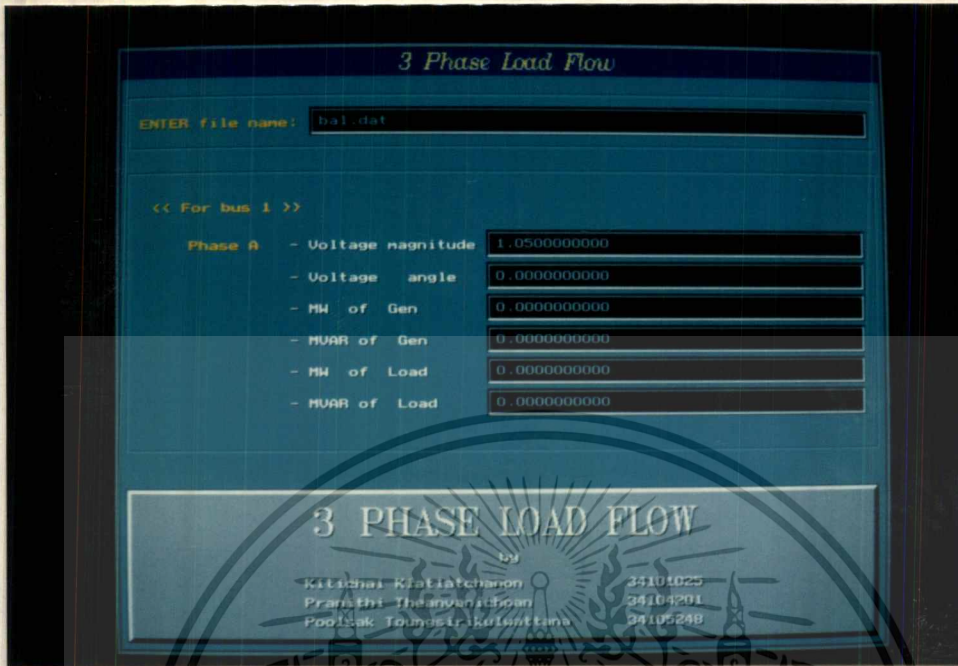
- 1: Read old data files
- 2: Write new data files
- 3: Exit

SELECT

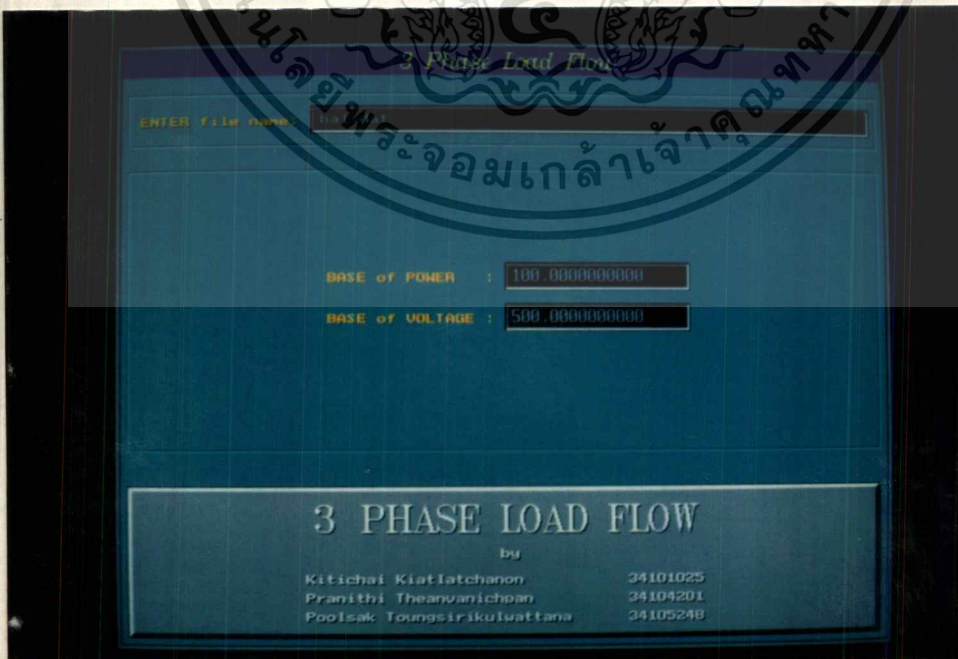
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงรูปที่ 4. และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



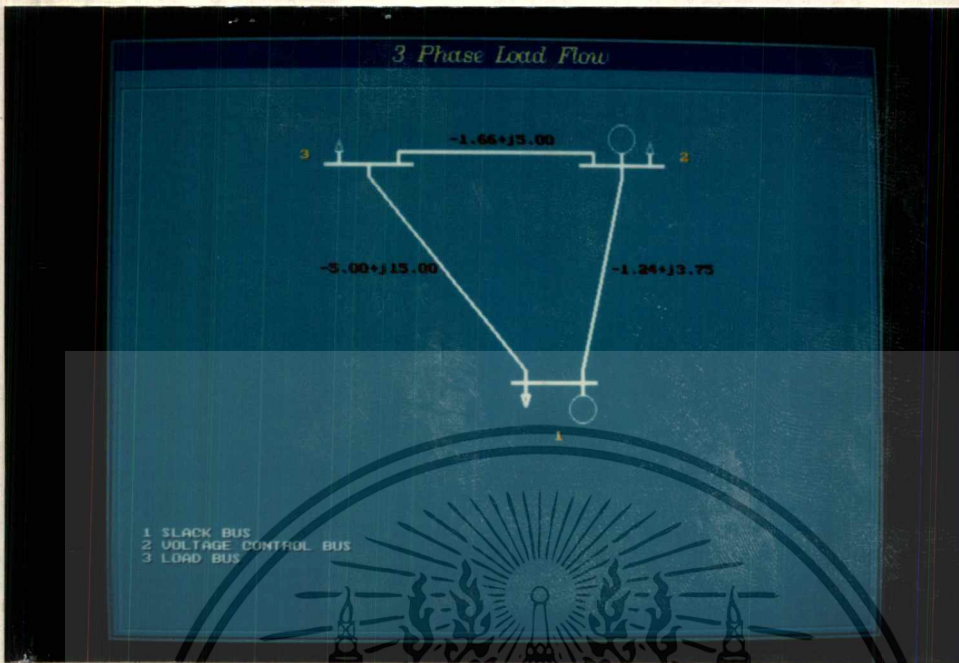
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่รูปที่ 6. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



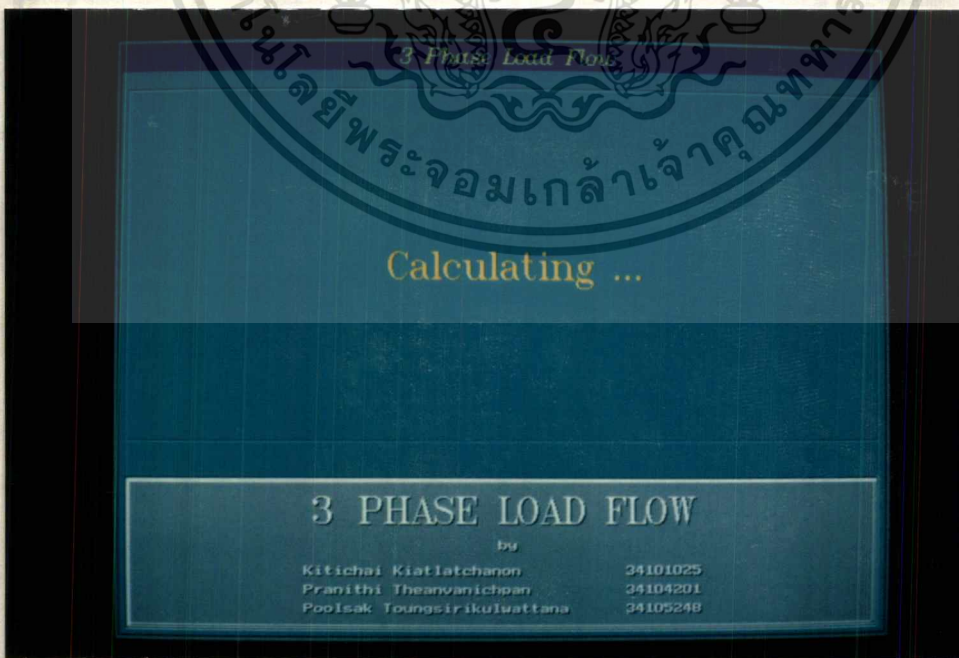
รูปที่ 7.



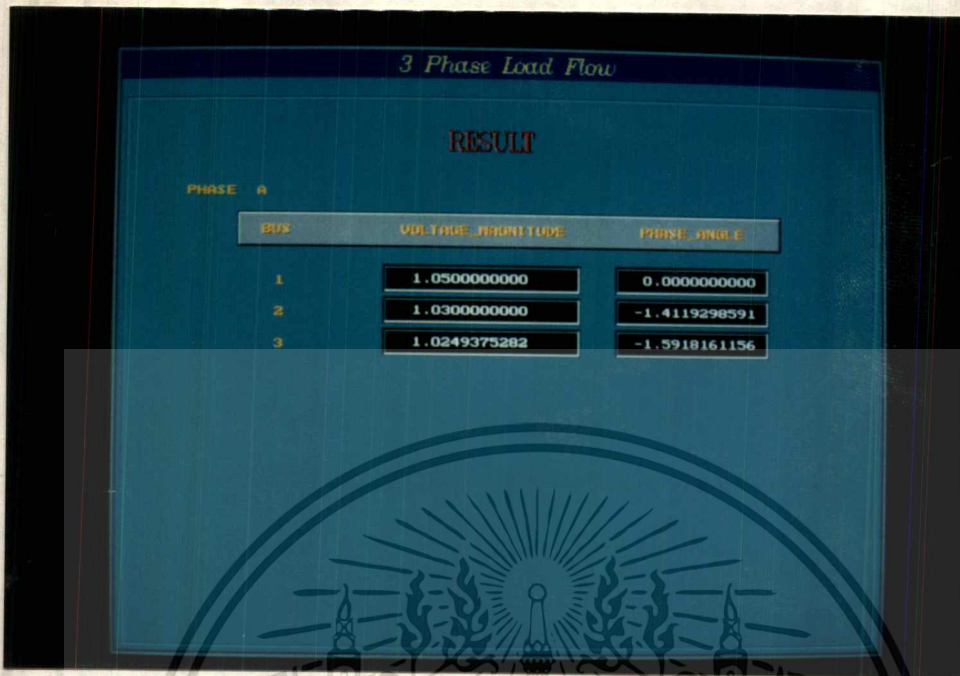
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงรูปที่ 8. และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



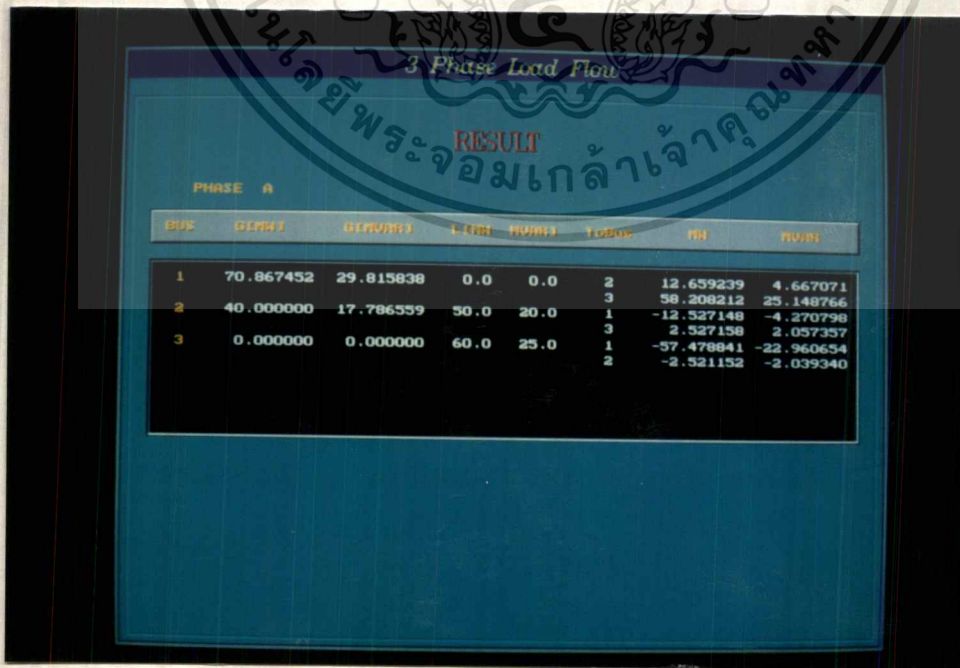
รูปที่ 9.



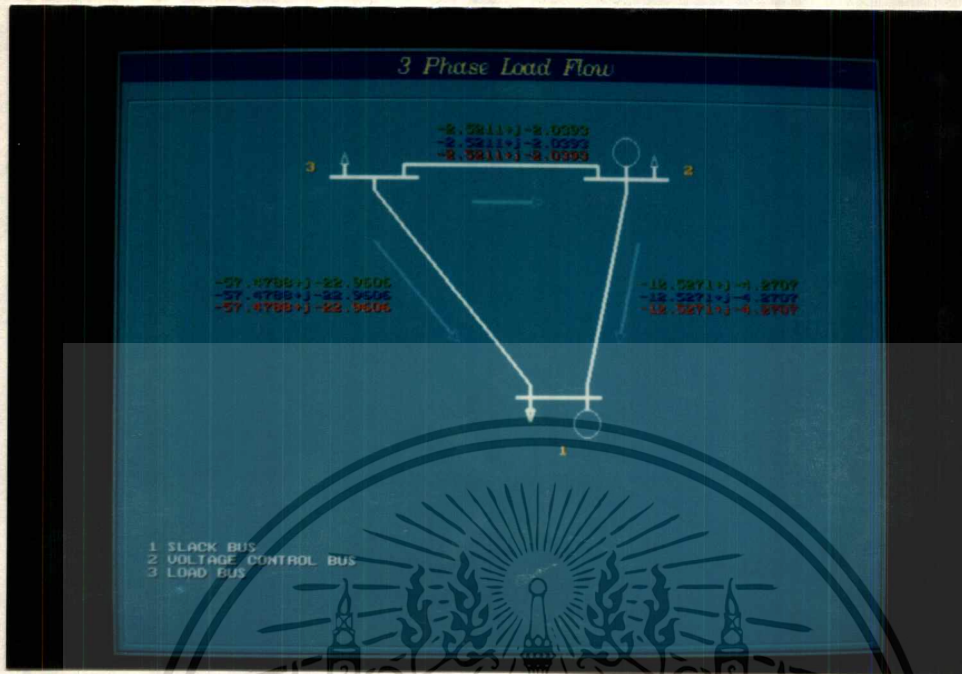
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 10.  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 12.  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## FILE MENU

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <process.h>
#include <dos.h>
```

```
#define X1 15
#define Y1 8
#define X2 64
#define Y2 17
```

```
void fill_window(int x1,int y1,int x2,int y2,int color);
void select(int new_choice,int old_choice);
void highlight(int method,int is_new);
void single_line_frame(int x1,int y1,int x2,int y2,int forecolor,int backcolor);
void cursor_off(void);
void cursor_on(void);
void running_topic(void);
void choose_sound(void);
```

```
void main()
{
```

```
    int max_choice = 4;
    int old_choice ,new_choice;
    int i;
    char ch='0';
```

```
    start:
    old_choice = new_choice = max_choice;
    cursor_off();
    textbackground(0);
    clrscr();
    fill_window(X1,Y1,X2,Y2,9);
    single_line_frame(X1-1,Y1-1,X2+1,Y2+1,14,9);
    running_topic();
    for(i=1;i<=max_choice;i++)
```

```
    {
        select(new_choice,old_choice);
        old_choice = new_choice;
        new_choice--;
    }
```

```
    gotoxy(X1,Y1+3);
    textcolor(14);
    textbackground(1);
    printf("        Select one choice");
```

```
    old_choice = -1;
    new_choice = 1;
    while (1)
    {
        ch = getch();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(ch!=13)
{
    sound(2000);
    delay(15);
    nosound();
}
if(ch==72)
{
    new_choice--;
    if(new_choice==0) new_choice = max_choice;
    select(new_choice,old_choice);
    old_choice = new_choice;
}
if(ch==80)
{
    new_choice++;
    if(new_choice==max_choice+1) new_choice = 1;
    select(new_choice,old_choice);
    old_choice = new_choice;
}
if((ch==13)&&(new_choice==1))
{
    choose_sound();
    system("c:\\dos\\mode co80");
    system("c:\\bc\\output\\p_succ11.exe");
    goto start;
}
if((ch==13)&&(new_choice==2))
{
    choose_sound();
    system("c:\\dos\\mode co80");
    system("");
    goto start;
}
if((ch==13)&&(new_choice==3))
{
    choose_sound();
    system("c:\\dos\\mode co80");
    system("");
    goto start;
}
if((ch==13)&&(new_choice==max_choice))
{
    choose_sound();
    system("c:\\dos\\mode co80");
    exit(1);
}
}
}

```

```

void select(int new_choice,int old_choice)

```

```

{
    highlight(old_choice,0);
    highlight(new_choice,1);
}

```

```

void highlight(int method,int is_new)

```

{ เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char method_name[54];
switch(method) {
case 1: strcpy(method_name, " 1. GAUSS SEIDEL METHOD ");
break;
case 2: strcpy(method_name, " 2. NEWTON RAPHSON POLAR COORDINATE METHOD ");
break;
case 3: strcpy(method_name, " 3. FAST DECOUPLE METHOD ");
break;
case 4: strcpy(method_name, " 4. EXIT ");
break;
}
if(is_new==0)
{
textbackground(1);
textcolor(15);
gotoxy(X1,Y1+4+method);
cprintf("%s",method_name);
}
else
{
textbackground(5);
textcolor(11);
gotoxy(X1,Y1+4+method);
cprintf("%s",method_name);
}
}

void fill_window(int x1,int y1,int x2,int y2,int color)
{
int i,j;
textbackground(color);
for(i=x1;i<=x2;i++)
{
for(j=y1;j<=y2;j++)
{
gotoxy(i,j);
cprintf(" ");
}
}
}

void single_line_frame(int x1,int y1,int x2,int y2,int forecolor,int backcolor)
{
int i,j;
textcolor(forecolor);
textbackground(backcolor);
gotoxy(x1,y1);
cprintf("U");
for(i=x1+1;i<=x2-1;i++)
{
gotoxy(i,y1);
cprintf("A");
}
cprintf(";");
for(j=y1+1;j<=y2-1;j++)
{

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cprintf("");
    gotoxy(x2,j);
    cprintf("");
}
gotoxy(x1,y2);
cprintf("A");
for(i=x1+1;i<=x2-1;i++)
{
    gotoxy(i,y2);
    cprintf("A");
}
cprintf("U");
}

```

```

void cursor_off(void)
{
    union REGS reg;
    reg.h.ch = 0x20;
    reg.h.cl = 0;
    reg.h.ah = 1;
    int86(0x10,&reg,&reg);
}

```

```

void cursor_on(void)
{
    union REGS reg;
    reg.h.ch = 0x00;
    reg.h.cl = 13;
    reg.h.ah = 1;
    int86(0x10,&reg,&reg);
}

```

```

void running_topic(void)
{
    int i;
    for(i=Y2-1;i>Y1;i--)
    {
        textbackground(1);
        gotoxy(X1,i+1);
        cprintf("
        ");
        if(i!=Y2-1)
        {
            gotoxy(X1,i);
            textbackground(12);
            textcolor(15);
            cprintf("
            LOAD FLOW ANALYSIS
            ");
        }
        sound(i*100);
        delay(70);
        nosound();
    }
}

```

```

void choose_sound(void)
{

```

sound(950);  
 delay(70);  
 sound(1280);

เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ใดๆ การผิดใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(70);  
sound(1750);  
delay(70);  
nosound();  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## FILE GUASS-SIEDEL METHOD

```
/* input data from keyboard */
/* many PV bus */
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <ctype.h>
#include <process.h>

#define SP_MAX_BUS 6
#define ERR 0.00000001

// declare function
int input_new_filedata_RX(void);
int input_new_filedata_VPQ(void);
int get_old_filedata(void);
void calculate(void);
void getstring(void);
void Make_RX_to_GB(void);
void Make_KL(void);
void Make_YL(void);
void Make_I(void);
void Cal_V(void);
void Cal_Error(void);
void Cal_PQ(void);
void Make_Table(void);
void graph(void);

// declare global variable
int slack_bus, v_control[SP_MAX_BUS], pv;
char filename[80];
FILE *fp;

/*----- External variable for Make_Ybus */
struct Data{
    int Bus_num;
    int t[SP_MAX_BUS];
    double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
    double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Yp[SP_MAX_BUS];
    double DIVIDE, DATA_1, DATA_2, DATA_3;
    double G_P[SP_MAX_BUS], G_Q[SP_MAX_BUS], BASE;
    double L_P[SP_MAX_BUS], L_Q[SP_MAX_BUS];
    double BASE_V, MAGNI_V[SP_MAX_BUS];
    double ANGLE_V[SP_MAX_BUS];
} Dat;

int MAX_BUS;
double MAGNI_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double ANGLE_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
/*----- Make_BVPQ */
double BASE, BASE_V;
int CHECK_LOAD_BUS[SP_MAX_BUS];
double Psch[SP_MAX_BUS], Qsch[SP_MAX_BUS];
double MAGNI_V[SP_MAX_BUS];
double ANGLE_V[SP_MAX_BUS];
double G_P[SP_MAX_BUS], G_Q[SP_MAX_BUS];
double L_P[SP_MAX_BUS], L_Q[SP_MAX_BUS];
/*----- Make_Delta_PQ */
double DATA_C[SP_MAX_BUS];
double DATA_D[SP_MAX_BUS];
double M[SP_MAX_BUS];
double A[SP_MAX_BUS];

```

```

/*----- Make_Curl_PQ */

```

```

double MAG_KL[SP_MAX_BUS];
double ANG_KL[SP_MAX_BUS];
double MAG_YL[SP_MAX_BUS][SP_MAX_BUS];
double ANG_YL[SP_MAX_BUS][SP_MAX_BUS];
/*----- Make_Jacobian */

```

```

double DELTA_VOL[SP_MAX_BUS];
double DELTA_ANG[SP_MAX_BUS];
/*----- Cal_Error */

```

```

int ITERATION;
double ERROR;

```

```

/*----- Cal_PQ */
double FLOW_P[SP_MAX_BUS][SP_MAX_BUS];
double FLOW_Q[SP_MAX_BUS][SP_MAX_BUS];
/*----- */

```

```

void main(void)
{

```

```

int ITERATION, PHASE_ABC;
char c=0;

```

```

while (1) // always loop
{ // start while loop

```

```

clrscr();
printf("SELECT CHOICE:\n");
printf("1 => Read old file\n");
printf("2 => Write new file\n");
printf("3 => Exit\n");

```

```

c=getch();

```

```

switch(c) {

```

```

case '1':if (get_old_filedata()==1)

```

```

{

```

```

Make_RX_to_GB();
calculate();
getch();

```

```

}

```

```

break;

```

```

case '2':if (input_new_filedata_RXO==1)

```

ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        Make_RX_to_GB();
        if (input_new_filedata_VPQO==1)
        {
            calculate();
            getch();
        }
    }
    break;
case '3':printf("\nBye\n");
    exit(1);
}
} // end while loop
}
//----- End of Main -----

```

```

void calculate(void)
{
    struct dostime _t t;

    clrscr();
    printf("GAUSS SEIDEL METHOD");
    t.hour = 0;
    t.minute = 0;
    t.second = 0;
    t.hsecond = 0;
    _dos_settime(&t);
    _dos_gettime(&t);
    for(ITERATION = 1 ; ITERATION < 1000 ; ITERATION++)
    {
        //gotoxy(1,3);
        //printf("ITERATION = [%d] = TOTAL ",ITERATION);

        Make_KL();
        Make_YL();
        Make_IO();
        Cal_V0();
        Cal_Error();
        if( ERROR == 0 )
        { gotoxy(1,3);
            printf("ITERATION = [%d] = TOTAL ",ITERATION);
            printf(" \t ERROR = [%10.8f] \n",ERR);
            Cal_PQ();
            ITERATION = 1100;
        }
    }
    _dos_gettime(&t);
    Make_Table();
    gotoxy(56,40);
    printf("Total time : %d:%d:%d.%d ",t.hour,t.minute,t.second,t.hsecond);
}

```

```

void getstring(void)

```

```

{
    int ch,X,Y;
    int i=0;
    while(ch!=13)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

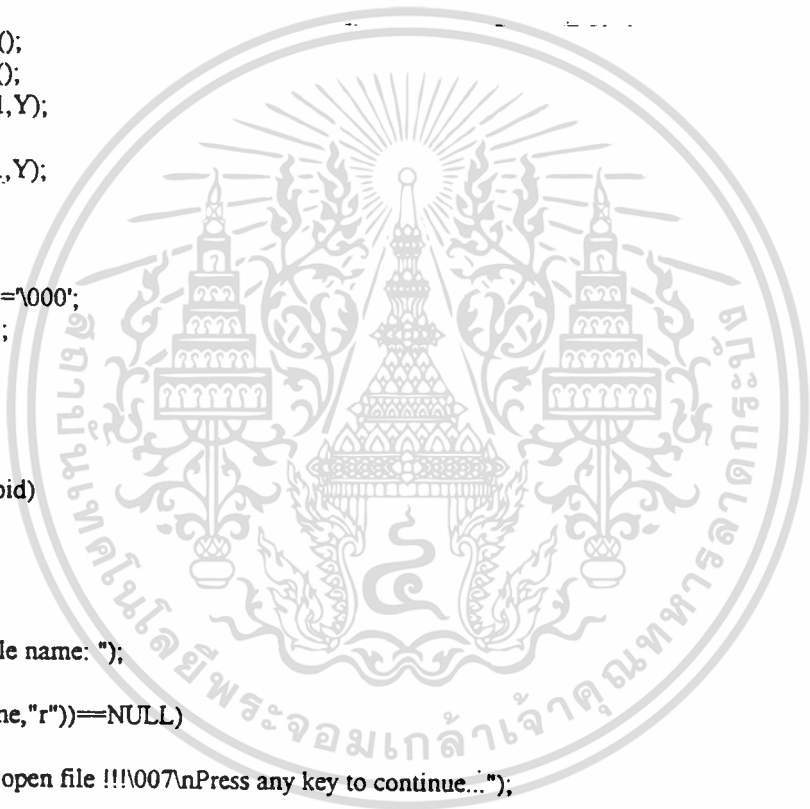
```

{
    ch = getch();
    ch = tolower(ch);
    //if (((ch>=97)&&(ch<=122))||(ch=='.')||(ch=='\')||(ch>=30)&&(ch>=39))
    if ((ch>=21)&&(ch<=176))
    {
        filename[i] = ch;
        printf("%c",ch);
        i++;
    }
    if (ch==47)
    {
        filename[i] = '\\';
        i++;
    }
    if (ch==8)
    {
        i--;
        X=wherex();
        Y=wherey();
        gotoxy(X-1,Y);
        printf(" ");
        gotoxy(X-1,Y);
    }
    if (ch==13)
    {
        filename[i]='\000';
        printf("\n");
    }
}

int get_old_filedata(void)
{
    int d=0,P,Q,n,m;
    int success = 1;

    printf("\nENTER file name: ");
    getstring();
    if((fp=fopen(filename,"r")==NULL)
    {
        printf("\nCan not open file !!!\007\nPress any key to continue...");
        success = 0;
        getch();
    }
    else
    {
        // Clear array
        for(P = 1 ; P < MAX_BUS ; P++)
        { for(Q = 1 ; Q < MAX_BUS ; Q++)
            { Img_Ypq2[P][Q] = 0;
              Real_Zpq[P][Q] = 0;
              Img_Zpq[P][Q] = 0;
            }
        }
        // read data
        fread(&Dat,sizeof(struct Data),1,fp);
        // display data

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clrscr();
MAX_BUS = Dat.Bus_num+1;
printf("Number of bus : %d\n",MAX_BUS-1);
for(P = 1; P < MAX_BUS ;P++)
{ // start for1 loop
  for(Q =1 ;Q < MAX_BUS ;Q++)
  { // start for2 loop
    Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q];
    Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q];
    Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];
    if(P!=Q)
    {
      printf("Impedance and line charging between bus %d and %d\n",P,Q );
      printf(" R   : %f\n",Real_Zpq[P][Q]);
      printf(" X   : %f\n",Img_Zpq[P][Q]);
      printf(" L chrg : %f\n",Img_Ypq2[P][Q]);
      getch();
    }
  } // end for2 loop
} // end for1 loop
pv= 0;
m = 0;
n = 1;
for(d=1;d<MAX_BUS;d++)
{
  switch(Dat.t[d]) {
  case 1 : n--;
            slack_bus = d;
            break;
  case 2 : m++;
            v_control[m] = d;
            pv = m ;
            break;
  case 3 : break;
  }
  MAGNI_V[d] = Dat.MAGNI_V[d];
  ANGLE_V[d] = Dat.ANGLE_V[d];
  G_P[d] = Dat.G_P[d];
  G_Q[d] = Dat.G_Q[d];
  L_P[d] = Dat.L_P[d];
  L_Q[d] = Dat.L_Q[d];
  printf("\nMagnitude and angle of Bus no. %d \n",d);
  printf(" Magnitude : %f\n",MAGNI_V[d]);
  printf(" Angle   : %f\n",ANGLE_V[d]);
  printf(" Mw of Gen. of Bus no. %d : %f\n",d,G_P[d]);
  printf(" Mvar of Gen. of Bus no. %d : %f\n",d,G_Q[d]);
  printf(" Mw of Load of Bus no. %d : %f\n",d,L_P[d]);
  printf(" Mvar of Load of Bus no. %d : %f\n",d,L_Q[d]);
  getch();
}

BASE = Dat.BASE;
BASE_V = Dat.BASE_V;
printf("\nBase of Power   : %f\n",BASE);
printf("Base of Voltage : %f\n",BASE_V);
getch();
fclose(fp);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(success);
}

int input_new_filedata_RX(void)
{
char ch;
int P,Q;
int success=1;

printf("\nEnter file name: ");
getstring();
if((fp=fopen(filename,"r"))!=NULL)
{
printf("\nWARNING !!\n");
printf(" Overwrite exist file ? (Y/N) ");
ch=getch();
if ((ch=='N')||(ch=='n'))
{
success = 0;
goto out_RX;
}
}
fclose(fp);
if((fp=fopen(filename,"w"))==NULL)
{ // start if3
success = 0;
printf("\nError !!!\007\nPress any to continue...");
getch();
} // end if3
else
{ // start else3
printf("\nEnter number of Bus :");
scanf("%d",&MAX_BUS);
Dat.Bus_num = MAX_BUS;
MAX_BUS = MAX_BUS+1;
for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
{ Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0 ; /* Clear all array */
Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q] = 0 ; /* for next loop */
Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0 ;
}
}
}

for(P = 1; P < MAX_BUS ;P++)
{ // start for1 loop
for(Q =1 ;Q < MAX_BUS ;Q++)
{ // start for2 loop
if(P<Q)
{ // start if4
printf("Is there Impedance and line charging between bus %d and %d(Y or N) : ",
P,Q);

ch = getch();
ch = toupper(ch);
if(ch == 'Y')
{
printf("\n");
printf(" R : ");
scanf("%le",&Dat.Real_Zpq[P][Q]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q];
    printf(" X : ");
    scanf("%le",&Dat.Img_Zpq[P][Q]);
    Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q];
    printf(" L chrg : ");
    scanf("%le",&Dat.Img_Ypq2[P][Q]);
    Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];
}
else
{
    Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q] = 0;
    Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0;
    Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0;
}
Dat.Real_Zpq[Q][P] = Dat.Real_Zpq[P][Q];
Dat.Img_Zpq[Q][P] = Dat.Img_Zpq[P][Q];
Dat.Img_Ypq2[Q][P] = Dat.Img_Ypq2[P][Q];
Real_Zpq[Q][P] = Real_Zpq[P][Q];
Img_Zpq[Q][P] = Img_Zpq[P][Q];
Img_Ypq2[Q][P] = Img_Ypq2[P][Q];
} // end if4
else
{ // start else4
    if(P==Q)
    { Real_Zpq[P][P] = Dat.Real_Zpq[P][P] = 0;
      Img_Zpq[P][P] = Dat.Img_Zpq[P][P] = 0;
      Img_Ypq2[P][P] = Dat.Img_Ypq2[P][P] = 0;
    }
} //end else4
} // end for2 loop
} // end for1 loop
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} // end else3
out_RX : return(success);
}

void Make_RX_to_GB(void)
{
    double DIVIDE , DATA_1 , DATA_2 , DATA_3;
    double Img_Yp[SP_MAX_BUS];
    double G[SP_MAX_BUS][SP_MAX_BUS],B[SP_MAX_BUS][SP_MAX_BUS];
    int P, Q, COUNT;
    char ch ;

    for(P = 1 ; P < MAX_BUS ; P++)
    { for(Q = 1 ; Q < MAX_BUS ; Q++)
      { DIVIDE = Real_Zpq[P][Q]*Real_Zpq[P][Q] ;
        DIVIDE = DIVIDE + Img_Zpq[P][Q]*Img_Zpq[P][Q] ;

        if ( DIVIDE != 0 )
        { Real_Ypq[P][Q] = Real_Zpq[P][Q] / DIVIDE ;
          Img_Ypq[P][Q] = -1*(Img_Zpq[P][Q]/DIVIDE) ;
        }
        /* -1 Because conjugate */
        else
        { Real_Ypq[P][Q] = 0 ;
          Img_Ypq[P][Q] = 0 ;
        }
      }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
/*----- Cal Yp */
for(P = 1 ; P < MAX_BUS ; P++) Img_Yp[P]=0 ;/*Protecthang*/

for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
  {
    Img_Yp[P] = Img_Yp[P] + Img_Ypq2[P][Q] ;
  }
}
/*----- Cal G_B */
for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
  {
    G[P][Q] = 0 ;
    B[P][Q] = 0 ;
  } /* Protect hang on next loop */
}

for(P = 1 ; P < MAX_BUS ; P++)
  B[P][P] = B[P][P] + Img_Yp[P] ;
/* Self admittances must include Yp */

for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
  {
    if( P == Q )
    {
      for(COUNT = 1 ; COUNT < MAX_BUS ; COUNT++)
      {
        G[P][P] = G[P][P] + Real_Ypq[P][COUNT] ;
        B[P][P] = B[P][P] + Img_Ypq[P][COUNT] ;
      }
    }
    else
    {
      G[P][Q] = -1*(Real_Ypq[P][Q]) ;
      B[P][Q] = -1*(Img_Ypq[P][Q]) ;
    } /* Mutual admittances value are negative */
  }
}

/*----- Cal Polar */
for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
  {
    DATA_1 = B[P][Q]*B[P][Q] + G[P][Q]*G[P][Q] ;
    MAGN_Ybus[P][Q] = sqrt(DATA_1) ;

    if( G[P][Q] != 0 ) DATA_1 = B[P][Q] / G[P][Q] ;
    else DATA_1 = 0 ;
    DATA_1 = atan(DATA_1) ; /* Value are in radian */
    DATA_1 = ( DATA_1 * 180 ) / 3.141592654 ;
    if( G[P][Q] < 0 ) DATA_1 = DATA_1 + 180 ;
    ANGLE_Ybus[P][Q] = DATA_1 ;
  }
}
}
//----- End of Make_RX_to_GB -----

```

```
int input_new_filedata_VPQ(void)
```

```
{
  int k,d,m,n,t[SP_MAX_BUS];
  double TEMP_DATA;
  int success=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clrscr();
pv = 0;
m = 0;
n = 1;
if((fp=fopen(filename,"w"))==NULL)
{ // start if1
    success = 0;
    printf("\nError !!!\007\nPress any to continue...");
    getch();
} // end if1
else
{ // start else1
    printf("%52s\n","<< TYPE OF BUS SELECTION >>");
    printf("%44s\n","1. SLACK BUS");
    printf("%57s\n","2. VOLTAGE CONTROLLED BUS");
    printf("%43s\n","3. LOAD BUS");
    printf("%45s\n","( SELECT 1-3 )");
    for (d=1; d < MAX_BUS ;d++)
    {
        printf(" TYPE OF BUS [%d] SELECT ==> ",d);
        scanf("%d",&t[d]);
        Dat.t[d] = t[d];
    }
    clrscr();
    for (d=1; d < MAX_BUS ;d++ )
    {
        switch(t[d]) {
            case 1 : n--;
                slack_bus = d;
                printf(" BUS [%d] SLACK BUS >> ",d);
                break;
            case 2 : m++;
                v_control[m] = d;
                pv = m ;
                printf(" BUS [%d] PV BUS >> ",d);
                break;
            case 3 : printf(" BUS [%d] LOAD BUS >> ",d);
                break;
            default: printf("%35s","<<< DATA ERROR ! >>>");
                success = 0;
                goto out_VPQ;
        }
    }
    if(n<0)
    {
        printf(" <<< DATA ERROR ! >>>\007\n");
        printf(" DUPLICATE SLACK BUS !!\n");
        printf(" Press any key to continue...");
        success = 0;
        getch();
        goto out_VPQ;
    }
    printf("%18s","MAGNITUDE : ");
    scanf("%le",&MAGNI_V[d]);
    Dat.MAGNI_V[d] = MAGNI_V[d];
    printf("%40s","ANGLE : ");
    scanf("%le",&ANGLE_V[d]);
    Dat.ANGLE_V[d] = ANGLE_V[d];
}

```

```

if( d != slack_bus )
{ /*
    if( d == v_control[m] )
    { /*
        printf("%+0s", "MW OF GEN : ");
        scanf("%le",&G_P[d]);
        Dat.G_P[d] = G_P[d];
        printf("%+0s", "MVAR OF GEN : ");
        scanf("%le",&G_Q[d]);
        Dat.G_Q[d] = G_Q[d];
        /* }
    else
    {
        G_P[d] = 0 ;
        G_Q[d] = 0 ;
        /*/
        printf("%+0s", "MW OF LOAD : ");
        scanf("%le",&L_P[d]);
        Dat.L_P[d] = L_P[d];
        printf("%+0s", "MVAR OF LOAD : ");
        scanf("%le",&L_Q[d]);
        Dat.L_Q[d] = L_Q[d];
    }
else
{
    Dat.G_P[slack_bus] = G_P[slack_bus] = 0;
    Dat.G_Q[slack_bus] = G_Q[slack_bus] = 0;
    Dat.L_P[slack_bus] = L_P[slack_bus] = 0;
    Dat.L_Q[slack_bus] = L_Q[slack_bus] = 0;
}
}
printf("Enter BASE of POWER : ");
scanf("%le",&BASE);
Dat.BASE = BASE;
printf("Enter BASE of VOLTAGE : ");
scanf("%le",&BASE_V);
Dat.BASE_V = BASE_V;
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} //end else1
out_VPQ: return(success);
}

```

```

void Make_KL (void)
{
    int i,d,j,m,k,check_pv;
    double MAG_PQ,ANG_PQ;
    double P[SP_MAX_BUS],Q[SP_MAX_BUS];
    double DATA_1,DATA_2,DATA_3;

    for(d=1 ; d < MAX_BUS ; d++)
    {
        Psch[d] = G_P[d] - L_P[d] ;
        Psch[d] = Psch[d] / BASE;
        Qsch[d] = G_Q[d] - L_Q[d];
        Qsch[d] = Qsch[d]/BASE;
    } //end loop1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่าในรูปแบบใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i=1 ; i< MAX_BUS; i++)
    if(i!=slack_bus)
    {
        check_pv = 1;
        for(m=1;m<(pv+1);m++)
            if(v_control[m] == i)
                check_pv = 0;

        if(check_pv!=0)
        {
            P[i] = Psch[i];
            Q[i] = (-1*Qsch[i]);
        }
        else
        {
            Q[i] = 0 ;
            for(k=1 ; k<MAX_BUS ; k++)
            {
                DATA_1 = MAGNI_V[m] * MAGNI_V[k] * MAGNI_Ybus[m][k] ;
                DATA_2 = ANGLE_V[m] - ANGLE_V[k] - ANGLE_Ybus[m][k] ;
                DATA_2 = ( DATA_2 * 3.141592654 ) / 180 ;

                DATA_3 = sin( DATA_2 ) ;
                Q[i] = Q[i] + ( DATA_1 * DATA_3 ) ;
            }
            Q[i] = (-1*Q[i]);
            P[i] = Psch[i];
        }
    } //end loop2

for( j=1; j<MAX_BUS; j++)
    if( j != slack_bus)
    {
        MAG_PQ = sqrt((P[j]*P[j]) + (Q[j]*Q[j]));
        if( P[j] != 0 )
            ANG_PQ = Q[j]/P[j];
        else
            ANG_PQ = 0;
        ANG_PQ = atan(ANG_PQ);
        ANG_PQ = ( ANG_PQ * 180 ) / 3.141592654 ;

        if (P[j] < 0)
            ANG_PQ = ANG_PQ + 180;
        MAG_KL[j] = MAG_PQ / MAGNI_Ybus[j][j];
        ANG_KL[j] = ANG_PQ - ANGLE_Ybus[j][j];
    } /*end if*/
}

```

```

void Make_YL(void)

```

```

{
    int p,q;

    for(p=1 ; p< MAX_BUS; p++)
        if(p != slack_bus)
        {
            for(q=1; q<MAX_BUS; q++)
                if(q != p)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 และไม่กรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        MAG_YL[p][q] = MAGNI_Ybus[p][q] / MAGNI_Ybus[p][p] ;
        ANG_YL[p][q] = ANGLE_Ybus[p][q] - ANGLE_Ybus[p][p] ;
    }
}

void Make_I(void)
{
    int k ;
    double DATA_1,DATA_3,MAG_I[SP_MAX_BUS],ANG_I[SP_MAX_BUS];
    double TEMP;

for(k=1;k<MAX_BUS;k++)
    if(k != slack_bus )
    {
        DATA_C[k]=0;
        DATA_D[k]=0;
        TEMP = -1*ANGLE_V[k];
        MAG_I[k] = MAG_KL[k] / MAGNI_V[k];
        ANG_I[k] = ANG_KL[k] - TEMP;
        DATA_1 = ( ANG_I[k] * 3.141592654 ) / 180 ;
        /*      DATA_1 *= -1; */

        DATA_3 = cos( DATA_1 );
        DATA_C[k] = MAG_I[k] * DATA_3 ;

        DATA_3 = sin( DATA_1 );
        DATA_D[k] = MAG_I[k] * DATA_3 ;
    }
}/*-----End of Make_I */

void Cal_V(void)
{
    double DATA_E , DATA_F , DATA_1 ;
    double DATA_2 , DATA_3 , DATA_4 , C[SP_MAX_BUS],D[SP_MAX_BUS];
    int I, k, m , check_pv;

for(I = 1 ; I < MAX_BUS ; I++)
    { if( I != slack_bus )
        {
            M[I] = MAGNI_V[I];
            A[I] = ANGLE_V[I];
            DATA_E = 0 ; DATA_F = 0 ;
            for(k = 1 ; k < MAX_BUS ; k++)
                if( k != I )
                {
                    DATA_1 = MAGNI_V[k] * MAG_YL[I][k] ;
                    DATA_2 = ANGLE_V[k] + ANG_YL[I][k] ;
                    DATA_2 = ( DATA_2 * 3.141592654 ) / 180 ;
                    /*      DATA_2 *= -1; */

                    DATA_3 = cos( DATA_2 );
                    DATA_E = DATA_E + ( DATA_1 * DATA_3 );
                    /* I not equal k */

                    DATA_3 = sin( DATA_2 );
                    DATA_F = DATA_F + ( DATA_1 * DATA_3 );
                }
            }
    }
}

```

```

    }
    C[I] = DATA_C[I] - DATA_E;
    D[I] = DATA_D[I] - DATA_F;

    ANGLE_V[I] = atan2(D[I],C[I]);
    ANGLE_V[I] = ( ANGLE_V[I] * 180)/ 3.141592654;

    check_pv = 1;
    for(m=1 ; m<(pv+1) ; m++)
        if(I != v_control[m])
            check_pv = 0;

    if(check_pv==1)
    {
        MAGNI_V[I] = (C[I]*C[I]) + (D[I]*D[I]) ;
        MAGNI_V[I] = sqrt(MAGNI_V[I]);
    }
    else
    {
        MAGNI_V[I] = M[I];
    }
}
}
} /*----- End of Cal_V-----*/

```

```

void Cal_Error()
{
    int k,m ;
    double SQ_V , SQ_A ;
    double DELTA_VOL[SP_MAX_BUS], DELTA_ANG[SP_MAX_BUS];

```

```

/*-----*/
ERROR = 0 ;
    for(k = 1 ; k < MAX_BUS ; k++ )
    { if ( k != slack_bus)
        {
            DELTA_VOL[k] = fabs( MAGNI_V[k] - M[k] );
            DELTA_ANG[k] = fabs( ANGLE_V[k] - A[k] );

            SQ_V = DELTA_VOL[k] ;
            SQ_A = DELTA_ANG[k] ;

            if(( SQ_V > ERR)|| ( SQ_A > ERR))
                ERROR =1;
            MAGNI_V[k] = M[k] + 1.4*(MAGNI_V[k] - M[k]);
            ANGLE_V[k] = A[k] + 1.4*(ANGLE_V[k] - A[k]);
        }
    }
}

```

```

} /*----- End of Cal_Error */

```

```

void Cal_PQO /* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */

```

```

{
    int p , q , k , Y , d , check_pv ;
    double e[SP_MAX_BUS] , f[SP_MAX_BUS] ;
    double REAL_1 , IMG_1 , TEMP_DATA ;
    /*----- Cal e f */
    for(p = 1 ; p < MAX_BUS ; p++)
    { TEMP_DATA = (ANGLE_V[p] * 3.141592654) / 180 ;

```

เอกสาร e[p] = MAGNI\_V[p] \* cos(TEMP\_DATA); เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    f[p] = MAGNI_V[p] * sin(TEMP_DATA);
}
/*----- Cal_PQ */
for(p = 1; p < MAX_BUS; p++)
{
    for(q = 1; q < MAX_BUS; q++)
    {
        if(q != p)
        {
            REAL_1 = e[p] * (e[p] - e[q]) + f[p] * (f[p] - f[q]);
            IMG_1 = e[p] * (f[p] - f[q]) - f[p] * (e[p] - e[q]);
            /* Ep^(Ep-Eq) = REAL_1 + jIMG_1 */

            TEMP_DATA = REAL_1 * Real_Ypq[p][q];
            FLOW_P[p][q] = TEMP_DATA - (IMG_1 * Img_Ypq[p][q]);
            FLOW_P[p][q] = BASE * FLOW_P[p][q];

            TEMP_DATA = REAL_1 * Img_Ypq[p][q];
            TEMP_DATA = TEMP_DATA + (IMG_1 * Real_Ypq[p][q]);
            FLOW_Q[p][q] = -1 * TEMP_DATA;
            TEMP_DATA = Img_Ypq2[p][q] * (e[p]*e[p] + f[p]*f[p]);
            FLOW_Q[p][q] = FLOW_Q[p][q] - TEMP_DATA;
            FLOW_Q[p][q] = BASE * FLOW_Q[p][q];
        }
    }
}
/*----- Cal G_PQ SLACK BUS */
for(k = 1, REAL_1 = 0, IMG_1 = 0; k < MAX_BUS; k++)
{
    if(k != slack_bus)
    {
        REAL_1 = REAL_1 + FLOW_P[slack_bus][k];
        G_P[slack_bus] = REAL_1;
        IMG_1 = IMG_1 + FLOW_Q[slack_bus][k];
        G_Q[slack_bus] = IMG_1;
    }
}
G_P[p] = G_P[p] + L_P[p];
G_Q[q] = G_Q[q] + L_Q[q];

/*----- Cal G_Q PV BUS */
for(k = 1, TEMP_DATA = 0; k < MAX_BUS; k++)
{
    check_pv = 1;
    for(d = 1; d < (pv+1); d++)
        if(v_control[d] == k)
            check_pv = 0;

    if(check_pv == 0)
    {
        for(q = 1; q < MAX_BUS; q++)
        {
            if(q != k)
            {
                TEMP_DATA = TEMP_DATA + FLOW_Q[k][q];
            }
        }
        G_Q[k] = TEMP_DATA + L_Q[k];
    }
}
}
/*----- End of Cal_PQ */
void Make_Table()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int p , q , k , Y , BUS ;
/*-----*/
printf("\n");
printf("BUS_____VOLTAGE_MAGNITUDE_____PHASE_ANGLE");
printf("\n\n");
for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
{ printf("[%d]",BUS) ; Y = wherey() ;
  gotoxy( 15,Y) ; printf("%f",MAGNI_V[BUS]) ;
  gotoxy(36,Y) ; printf("%9.6f",ANGLE_V[BUS]) ;
  printf("\n") ;
}
getch();
clrscr();
printf("\n");
printf("BUS_____G[MW]_____G[MVAR]_____L[MW___VAR]_");
printf(" ToBus_____MW_____MVAR_____ \n\n");
for(p = 1 ; p < MAX_BUS ; p++)
{ printf("[%d]",p) ; Y = wherey() ;
  gotoxy( 7,Y) ; printf("%9.6f",G_P[p]) ;
  gotoxy(20,Y) ; printf("%9.6f",G_Q[p]) ;
  gotoxy(34,Y) ; printf("%4.1f",L_P[p]) ;
  gotoxy(41,Y) ; printf("%4.1f",L_Q[p]) ;
  for(q = 1 ; q < MAX_BUS ; q++)
  { if(q != p)
    { gotoxy(49,Y) ; printf("%d",q) ;
      gotoxy(53,Y) ; printf("%11.7f",FLOW_P[p][q]) ;
      gotoxy(67,Y) ; printf("%11.7f",FLOW_Q[p][q]) ;
      Y++ ;
    }
    printf("\n") ;
  }
  printf("\n") ;
  if(( 25-Y ) < (MAX_BUS-2))
  {
  getch();
  clrscr();
  }
}
} /*----- End of Make_Table */
/*-----*/
/*
      END OF FILE
*/
/*-----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**FILE NEWTON-RAPHSON**  
**POLAR COORDINATES METHOD**

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <ctype.h>
#include <process.h>
#define SP_MAX_BUS 6
#define ERR 0.00000001
```

```
// declare function
int input_new_filedata_RX(void);
int input_new_filedata_VPQ(void);
int get_qld_filedata(void);
void calculate(void);
void getstring(void);
void Make_RX_to_GB(void);
void Make_Delta_PQ(void);
void Make_Curl_PQ(void);
void Make_Jacobian(void);
void Cal_Error(void);
void Cal_PQ(void);
void Make_Table(void);
void graph(void);
```

```
// declare global variable
int slack_bus, v_control[SP_MAX_BUS], pv;
char filename[80];
FILE *fp;
```

```
/*----- External variable for Make_Ybus */
```

```
struct Data{
    int Bus_num;
    int t[SP_MAX_BUS];
    double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
    double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
    double Img_Yp[SP_MAX_BUS];
    double DIVIDE , DATA_1 , DATA_2 , DATA_3;
    double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS] ,BASE;
    double L_P[SP_MAX_BUS] , L_Q[SP_MAX_BUS];
    double BASE_V,MAGNI_V[SP_MAX_BUS];
    double ANGLE_V[SP_MAX_BUS];
} Dat;
```

```
int MAX_BUS;
double MAGNI_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double ANGLE_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
```

ไม่ว่าการณีใดท้งสิ้น อีกท้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
/*----- Make_BVPQ */
double BASE, BASE_V;
int CHECK_LOAD_BUS[SP_MAX_BUS];
double Psch[SP_MAX_BUS], Qsch[SP_MAX_BUS];
double MAGNI_V[SP_MAX_BUS];
double ANGLE_V[SP_MAX_BUS];
double G_P[SP_MAX_BUS], G_Q[SP_MAX_BUS];
double L_P[SP_MAX_BUS], L_Q[SP_MAX_BUS];
/*----- Make_Delta_PQ */
double DELTA_P[SP_MAX_BUS];
double DELTA_Q[SP_MAX_BUS];
/*----- Make_Curl_PQ */
double CURL_P_DELTA[SP_MAX_BUS][SP_MAX_BUS];
double CURL_Q_DELTA[SP_MAX_BUS][SP_MAX_BUS];
double CURL_P_V[SP_MAX_BUS][SP_MAX_BUS];
double CURL_Q_V[SP_MAX_BUS][SP_MAX_BUS];
/*----- Make_Jacobian */
double DELTA_VOL[SP_MAX_BUS];
double DELTA_ANG[SP_MAX_BUS];
/*----- Cal_Error */
int ITERATION;
double ERROR;
/*----- Cal_PQ */
double FLOW_P[SP_MAX_BUS][SP_MAX_BUS];
double FLOW_Q[SP_MAX_BUS][SP_MAX_BUS];
/*----- */

```

```

void main(void)
{

```

```

    int ITERATION, PHASE_ABC;
    char c=0;

```

```

    while (1) // alway loop
    { // start while loop

```

```

        clrscr();
        printf("SELECT CHOICE:\n");
        printf("1 => Read old file\n");
        printf("2 => Write new file\n");
        printf("3 => Exit\n");

```

```

        c=getch();

```

```

        switch(c) {

```

```

            case '1':if (get_old_filedata()==1)

```

```

            {

```

```

                Make_RX_to_GB();

```

```

                calculate();

```

```

                getch();

```

```

            }

```

```

            break;

```

```

            case '2':if (input_new_filedata_RX()==1)

```

```

            {

```

```

                Make_RX_to_GB();

```

```

                if (input_new_filedata_VPQ()==1)

```

```

                {

```

```

                    calculate();

```

```

                    getch();

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    }
    break;
case '3':printf("\nBye\n");
    exit(1);
}
} // end while loop
}
//----- End of Main -----

void calculate(void)
{
    struct dostime_t t;

    clrscr();
    printf("NEWTON-RAPHSON POLAR COORDINATE METHOD");
    t.hour = 0;
    t.minute = 0;
    t.second = 0;
    t.hsecond = 0;
    _dos_settime(&t);
    _dos_gettime(&t);
    for(ITERATION = 1 ; ITERATION < 1000 ; ITERATION++)
    {
        //gotoxy(1,3);
        //printf("ITERATION = [%d] = TOTAL ",ITERATION);
        Make_Delta_PQ();
        Make_Curl_PQ();
        Make_Jacobian();
        Cal_Error();
        if( ERROR == 0 )
        { gotoxy(1,3);
          printf("ITERATION = [%d] = TOTAL ",ITERATION);
          printf("\t ERROR = [%10.8f]\n",ERR);
          Cal_PQ();
          ITERATION = 1100;
        }
    }
    _dos_gettime(&t);
    Make_Table();
    gotoxy(56,40);
    printf("Total time : %d:%d:%d.%d ",t.hour,t.minute,t.second,t.hsecond);
}

```

```

void getstring(void)
{
    int ch,X,Y;
    int i=0;
    while(ch!=13)
    {
        ch = getch();
        ch = tolower(ch);
        //if (((ch>=97)&&(ch<=122))||(ch=='.')||(ch=='\')|((ch>=30)&&(ch<=39)))
        if ((ch>=21)&&(ch<=176))
        {
            filename[i] = ch;
            printf("%c",ch);
            i++;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if (ch==47)
{
    filename[i] = '\\';
    i++;
}
if (ch==8)
{
    i--;
    X=wherex();
    Y=wherey();
    gotoxy(X-1,Y);
    printf(" ");
    gotoxy(X-1,Y);
}
if (ch==13)
{
    filename[i]='\000';
    printf("\n");
}
}
}

```

```

int get_old_filedata(void)

```

```

{
    int d=0,P,Q,n,m;
    int success = 1;

    printf("\nENTER file name: ");
    getstring();
    if((fp=fopen(filename,"r"))==NULL)
    {
        printf("\nCan not open file !!!\007\nPress any key to continue...");
        success = 0;
        getch();
    }
    else
    {
        // Clear array
        for(P = 1 ; P < MAX_BUS ; P++)
        { for(Q = 1 ; Q < MAX_BUS ; Q++)
            { Img_Ypq2[P][Q] = 0;
              Real_Zpq[P][Q] = 0;
              Img_Zpq[P][Q] = 0;
            }
        }

        // read data
        fread(&Dat,sizeof(struct Data),1,fp);
        // display data
        clrscr();
        MAX_BUS = Dat.Bus_num+1;
        printf("Number of bus : %d\n",MAX_BUS-1);
        for(P = 1; P < MAX_BUS ;P++)
        { // start for1 loop
            for(Q =1 ;Q < MAX_BUS ;Q++)
            { // start for2 loop

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];
    if(P!=Q)
    {
        printf("Impedance and line charging between bus %d and %d\n",P,Q);
        printf(" R   : %f\n",Real_Zpq[P][Q]);
        printf(" X   : %f\n",Img_Zpq[P][Q]);
        printf(" L chrg : %f\n",Img_Ypq2[P][Q]);
        getch();
    }
} // end for2 loop
} // end for1 loop

```

```

pv= 0;
m = 0;
n = 1;
for(d=1;d<MAX_BUS;d++)
{

```

```

    switch(Dat.t[d]) {

```

```

    case 1 : n--;

```

```

        slack_bus = d;

```

```

        break;

```

```

    case 2 : m++;

```

```

        v_control[m] = d;

```

```

        pv = m;

```

```

        break;

```

```

    case 3 : break;
    }

```

```

MAGNI_V[d] = Dat.MAGNI_V[d];

```

```

ANGLE_V[d] = Dat.ANGLE_V[d];

```

```

G_P[d] = Dat.G_P[d];

```

```

G_Q[d] = Dat.G_Q[d];

```

```

L_P[d] = Dat.L_P[d];

```

```

L_Q[d] = Dat.L_Q[d];

```

```

printf("\nMagnitude and angle of Bus no. %d\n",d);

```

```

printf(" Magnitude : %f\n",MAGNI_V[d]);

```

```

printf(" Angle   : %f\n",ANGLE_V[d]);

```

```

printf("Mw of Gen. of Bus no. %d : %f\n",d,G_P[d]);

```

```

printf("Mvar of Gen. of Bus no. %d : %f\n",d,G_Q[d]);

```

```

printf("Enter Mw of Load of Bus no. %d : %f\n",d,L_P[d]);

```

```

printf("Enter Mvar of Load of Bus no. %d : %f\n",d,L_Q[d]);

```

```

getch();
}

```

```

BASE = Dat.BASE;

```

```

BASE_V = Dat.BASE_V;

```

```

printf("\nBase of Power : %f\n",BASE);

```

```

printf("Base of Voltage : %f\n",BASE_V);

```

```

getch();

```

```

fclose(fp);
}

```

```

return(success);
}

```

```

int input_new_filedata_RX(void)
{

```

```

    char ch;

```

```

    int P,Q;

```

```

    int success=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\nEnter file name: ");
getstring();
if((fp=fopen(filename,"r"))!=NULL)
{
printf("\nWARNING !!\n");
printf(" Overwrite exist file ? (Y/N) ");
ch=getch();
if ((ch=='N')||(ch=='n'))
{
success = 0;
goto out_RX;
}
}
fclose(fp);
if((fp=fopen(filename,"w"))==NULL)
{ // start if3
success = 0;
printf("\nError !!!\007\nPress any to continue...");
getch();
} // end if3
else
{ // start else3
printf("\nEnter number of Bus :");
scanf("%d",&MAX_BUS);
Dat.Bus_num = MAX_BUS;
MAX_BUS = MAX_BUS+1;
for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
{ Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0 ; /* Clear all array */
Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q] = 0 ; /* for next loop */
Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0 ;
}
}
for(P = 1; P < MAX_BUS ;P++)
{ // start for1 loop
for(Q =1 ;Q < MAX_BUS ;Q++)
{ // start for2 loop
if(P<Q)
{ // start if4
printf("Is there Impedance and line charging between bus %d and %d(Y or N) : ",
P,Q);
ch = getch();
ch = toupper(ch);
if(ch == 'Y')
{
printf("\n");
printf(" R : ");
scanf("%le",&Dat.Real_Zpq[P][Q]);
Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q];
printf(" X : ");
scanf("%le",&Dat.Img_Zpq[P][Q]);
Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q];
printf(" L chrg : ");
scanf("%le",&Dat.Img_Ypq2[P][Q]);
Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];
}
}
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q] = 0;
        Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0;
        Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0;
    }
    Dat.Real_Zpq[Q][P] = Dat.Real_Zpq[P][Q];
    Dat.Img_Zpq[Q][P] = Dat.Img_Zpq[P][Q];
    Dat.Img_Ypq2[Q][P] = Dat.Img_Ypq2[P][Q];
    Real_Zpq[Q][P] = Real_Zpq[P][Q];
    Img_Zpq[Q][P] = Img_Zpq[P][Q];
    Img_Ypq2[Q][P] = Img_Ypq2[P][Q];
} // end if4
else
{ // start else4
    if(P==Q)
    { Real_Zpq[P][P] = Dat.Real_Zpq[P][P] = 0;
      Img_Zpq[P][P] = Dat.Img_Zpq[P][P] = 0;
      Img_Ypq2[P][P] = Dat.Img_Ypq2[P][P] = 0;
    }
} //end else4
} // end for2 loop
} // end for1 loop
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} // end else3
out_RX : return(success);
}

```

```

void Make_RX_to_GB(void)

```

```

{
    double DIVIDE , DATA_1 , DATA_2 , DATA_3;
    double Img_Yp[SP_MAX_BUS];
    double G[SP_MAX_BUS][SP_MAX_BUS],B[SP_MAX_BUS][SP_MAX_BUS];
    int P, Q, COUNT;
    char ch ;

    for(P = 1 ; P < MAX_BUS ; P++)
    { for(Q = 1 ; Q < MAX_BUS ; Q++)
        { DIVIDE = Real_Zpq[P][Q]*Real_Zpq[P][Q] ;
          DIVIDE = DIVIDE + Img_Zpq[P][Q]*Img_Zpq[P][Q] ;

          if ( DIVIDE != 0 )
          { Real_Ypq[P][Q] = Real_Zpq[P][Q] / DIVIDE ;
            Img_Ypq[P][Q] = -1*(Img_Zpq[P][Q]/DIVIDE) ;
          }
          /* -1 Because conjugate */
          else
          { Real_Ypq[P][Q] = 0 ;
            Img_Ypq[P][Q] = 0 ;
          }
        }
    }
}
/*----- Cal Yp */
for(P = 1 ; P < MAX_BUS ; P++) Img_Yp[P]=0 ;/*Protecthang*/

for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
    { Img_Yp[P] = Img_Yp[P] + Img_Ypq2[P][Q] ;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
/*----- Cal G_B */
for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
  { G[P][Q] = 0 ;
    B[P][Q] = 0 ;
  } /* Protect hang on next loop */
}

for(P = 1 ; P < MAX_BUS ; P++)
  B[P][P] = B[P][P] + Img_Yp[P] ;
/* Self admittances must include Yp */

for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
  { if( P == Q )
    { for(COUNT = 1 ; COUNT < MAX_BUS ; COUNT++)
      { G[P][P] = G[P][P] + Real_Ypq[P][COUNT] ;
        B[P][P] = B[P][P] + Img_Ypq[P][COUNT] ;
      }
    }
    else
    { G[P][Q] = -1*(Real_Ypq[P][Q]) ;
      B[P][Q] = -1*(Img_Ypq[P][Q]) ;
    } /* Mutual admittances value are negative */
  }
}

/*----- Cal Polar */
for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
  { DATA_1 = B[P][Q]*B[P][Q] + G[P][Q]*G[P][Q] ;
    MAGNI_Ybus[P][Q] = sqrt(DATA_1) ;

    if( G[P][Q] != 0 ) DATA_1 = B[P][Q] / G[P][Q] ;
    else DATA_1 = 0 ;
    DATA_1 = atan(DATA_1) ; /* Value are in radian */
    DATA_1 = ( DATA_1 * 180 ) / 3.141592654 ;
    if( G[P][Q] < 0 ) DATA_1 = DATA_1 + 180 ;
    ANGLE_Ybus[P][Q] = DATA_1 ;
  }
}

//----- End of Make_RX_to_GB -----

```

```

int input_new_filedata_VPQ(void)
{
  int k,d,m,n,t[SP_MAX_BUS];
  double TEMP_DATA;
  int success=1;

  clrscr();
  pv = 0;
  m = 0;
  n = 1;
  if((fp=fopen(filename,"w"))==NULL)
  { // start if1
    success = 0;
    printf("\nError !!!\007\nPress any to continue...");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

getch();
} // end if1
else
{ // start else1
printf("%52s\n", "<< TYPE OF BUS SELECTION >>");
printf("%44s\n", "1. SLACK BUS");
printf("%55s\n", "2. VOLTAGE CONTROLLED BUS");
printf("%43s\n", "3. LOAD BUS");
printf("%45s\n", "( SELECT 1-3 )");
for (d=1; d < MAX_BUS ;d++)
{
printf(" TYPE OF BUS [%d] SELECT ==> ",d);
scanf("%d",&t[d]);
Dat.t[d] = t[d];
}
clrscr();
for (d=1; d < MAX_BUS ;d++ )
{
switch(t[d]) {
case 1 : n--; -
slack_bus = d;
printf(" BUS [%d] SLACK BUS >> ",d);
break;
case 2 : m++;
v_control[m] = d;
pv = m ;
printf(" BUS [%d] PV BUS >> ",d);
break;
case 3 : printf(" BUS [%d] LOAD BUS >> ",d);
break;
default: printf("%35s", "<<< DATA ERROR ! >>>");
success = 0;
goto out_VPQ;
}
if(n<0)
{
printf(" <<< DATA ERROR ! >>>\007\n");
printf(" DUPLICATE SLACK BUS !!\n");
printf(" Press any key to continue...");
success = 0;
getch();
goto out_VPQ;
}
printf("%18s", "MAGNITUDE : ");
scanf("%le",&MAGNI_V[d]);
Dat.MAGNI_V[d] = MAGNI_V[d];
printf("%40s", "ANGLE : ");
scanf("%le",&ANGLE_V[d]);
Dat.ANGLE_V[d] = ANGLE_V[d];

if( d != slack_bus )
{ /*
if( d == v_control[m] )
{ /*
printf("%40s", "MW OF GEN : ");
scanf("%le",&G_P[d]);
Dat.G_P[d] = G_P[d];
printf("%40s", "MVAR OF GEN : ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

scanf("%le",&G_Q[d]);
Dat.G_Q[d] = G_Q[d];
/* }
else
{
G_P[d] = 0;
G_Q[d] = 0;
}*/
printf("%+0s", "MW OF LOAD : ");
scanf("%le",&L_P[d]);
Dat.L_P[d] = L_P[d];
printf("%+0s", "MVAR OF LOAD : ");
scanf("%le",&L_Q[d]);
Dat.L_Q[d] = L_Q[d];

```

```

}
else
{

```

```

Dat.G_P[slack_bus] = G_P[slack_bus] = 0;
Dat.G_Q[slack_bus] = G_Q[slack_bus] = 0;
Dat.L_P[slack_bus] = L_P[slack_bus] = 0;
Dat.L_Q[slack_bus] = L_Q[slack_bus] = 0;
}
}

```

```

printf("Enter BASE of POWER : ");
scanf("%le",&BASE);
Dat.BASE = BASE;
printf("Enter BASE of VOLTAGE : ");
scanf("%le",&BASE_V);
Dat.BASE_V = BASE_V;
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} //end else1
out_VPQ: return(success);
}

```

```

//----- End of input_new_filedata_VPQ -----

```

```

void Make_Delta_PQ(void)
{

```

```

double DATA_P, DATA_Q, DATA_1, DATA_2, DATA_3;
int I, K, d;
/*-----*/

```

```

for(d=1 ; d < MAX_BUS ; d++)
{

```

```

Psch[d] = G_P[d] - L_P[d];
Psch[d] = Psch[d] / BASE;
Qsch[d] = G_Q[d] - L_Q[d];
Qsch[d] = Qsch[d]/BASE;
}

```

```

for(I = 1 ; I < MAX_BUS ; I++) /*--- 1 Slack bus ---*/
if( I != slack_bus )
{ DATA_P = 0 ; DATA_Q = 0 ;
for(K = 1 ; K < MAX_BUS ; K++)
{ DATA_1 = MAGNI_V[I] * MAGNI_V[K] *

```

```

MAGNI_Ybus[I][K] ;

```

```

DATA_2 = ANGLE_V[I] - ANGLE_V[K] -

```

```

ANGLE_Ybus[I][K] ;

```

```

DATA_2 = (DATA_2 * 3.141592654) / 180 ;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA_3 = cos( DATA_2 );
DATA_P = DATA_P + ( DATA_1 * DATA_3 );

DATA_3 = sin( DATA_2 );
DATA_Q = DATA_Q + ( DATA_1 * DATA_3 );
}
DELTA_P[I] = Psch[I] - DATA_P;
DELTA_Q[I] = Qsch[I] - DATA_Q;
}
} /*----- End of Make_Delta_PQ */

void Make_Curl_PQ(void)
{
double DATA_P, DATA_Q, DATA_1;
double DATA_2, DATA_3, DATA_4;
int I, k;
/*----- Cal Curl_PQ_Delta */
for(I = 1; I < MAX_BUS; I++)
if( I != slack_bus )
{ DATA_P = 0; DATA_Q = 0;
for(k = 1; k < MAX_BUS; k++)
{ if( k == I ) k++;
DATA_1 = MAGNI_V[I] * MAGNI_V[k] *
MAGNI_Ybus[I][k];
DATA_2 = ANGLE_V[I] - ANGLE_V[k] -
ANGLE_Ybus[I][k];
DATA_2 = ( DATA_2 * 3.141592654 ) / 180;

DATA_3 = sin( DATA_2 );
DATA_P = DATA_P + ( DATA_1 * DATA_3 );
CURL_P_DELTA[I][k] = 1 * DATA_1 * DATA_3;
/* I not equal k */

DATA_3 = cos( DATA_2 );
DATA_Q = DATA_Q + ( DATA_1 * DATA_3 );
CURL_Q_DELTA[I][k] = -1 * DATA_1 * DATA_3;
}
CURL_P_DELTA[I][I] = -1 * DATA_P; /* for I == k */
CURL_Q_DELTA[I][I] = 1 * DATA_Q; /* for I == k */
}
} /*----- Cal Curl_PQ_V */
for(I = 1; I < MAX_BUS; I++)
if( I != slack_bus )
{ for(k = 1; k < MAX_BUS; k++)
{ DATA_1 = MAGNI_V[I] * MAGNI_Ybus[I][k];
DATA_2 = ANGLE_V[I] - ANGLE_V[k] -
ANGLE_Ybus[I][k];
DATA_2 = ( DATA_2 * 3.141592654 ) / 180;

DATA_3 = cos( DATA_2 );
CURL_P_V[I][k] = DATA_1 * DATA_3; /* I != k */

DATA_3 = sin( DATA_2 );
CURL_Q_V[I][k] = DATA_1 * DATA_3; /* I != k */
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ( I != slack_bus )
{ DATA_P=0; DATA_Q=0;
  for(k = 1; k < MAX_BUS; k++)
  { if(k == I) k++;
    DATA_1 = MAGNI_V[k] * MAGNI_Ybus[I][k];
    DATA_2 = ANGLE_V[I] - ANGLE_V[k] -
              ANGLE_Ybus[I][k];
    DATA_2 = ( DATA_2 * 3.141592654 ) / 180;

    DATA_3 = cos( DATA_2 );
    DATA_P = DATA_P + ( DATA_1 * DATA_3 );

    DATA_3 = sin( DATA_2 );
    DATA_Q = DATA_Q + ( DATA_1 * DATA_3 );
  }
  DATA_1 = MAGNI_V[I] * MAGNI_Ybus[I][I];

  DATA_2 = ( ANGLE_Ybus[I][I] * 3.141592654 ) / 180;

  DATA_3 = cos( -1 * DATA_2 );
  CURL_P_V[I][I] = ( 2 * DATA_1 * DATA_3 ) + DATA_P;
  /* I == k */

  DATA_3 = sin( -1 * DATA_2 );
  CURL_Q_V[I][I] = ( 2 * DATA_1 * DATA_3 ) + DATA_Q;
  /* I == k */
} /*----- End 2of Make_Curl_PQ */

```

```

void Make_Jacobian()

```

```

{
double JACOBIAN[SP_MAX_BUS*2][SP_MAX_BUS*2];
double TEMP_JACOB[SP_MAX_BUS*2];
int I, X, m, check_pv;
int k, J_X, J_Y;
int DIMENSION = 0; /* Number of equation */
double MULTI;
/*----- Jacob P & Del */

```

```

for(I = 1, J_Y = 1; I < MAX_BUS; I++, J_Y++)
{
  if ( I != slack_bus )
  { DIMENSION++;
    for(X = 1, J_X = 1; X < MAX_BUS; X++, J_X++)
    if ( X != slack_bus )
    { JACOBIAN[J_Y][J_X] = CURL_P_DELTA[I][X];
    }
    else
      J_X--;

    for(X = 1; X < MAX_BUS; X++)
    { check_pv = 1;
      for(m = 1; m < (pv + 1); m++)
        if( v_control[m] == X )
          check_pv = 0;

      if(( X != slack_bus ) && ( check_pv != 0 ))
        { JACOBIAN[J_Y][J_X] = CURL_P_V[I][X];
          J_X++;
        }
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

```

}
}
}
else
  J_Y--;
}
/*----- Jacob Q & Del */
for(I = 1 ; I < MAX_BUS ; I++)
{
  check_pv = 1;
  for(m=1 ; m < (pv+1) ; m++)
    if(v_control[m] == I)
      check_pv = 0;

  if(( I != slack_bus )&&( check_pv != 0 ))
  {
    DIMENSION++;
    for(X = 1 , J_X = 1 ; X < MAX_BUS ; X++ , J_X++)
      if ( X != slack_bus )
      {
        JACOBIAN[J_Y][J_X] = CURL_Q_DELTA[I][X] ;
      }
      /* C_Q_D FOR ONLY PQ_BUS */
    else
      J_X--;

    for(X = 1 ; X < MAX_BUS ; X++)
    {
      check_pv = 1;
      for( m=1 ; m < (pv+1) ; m++)
        if(v_control[m] == X)
          check_pv = 0;

      if(( X != slack_bus )&&( check_pv != 0 ))
      {
        JACOBIAN[J_Y][J_X] = CURL_Q_V[I][X] ;
        J_X++; /* C_Q_V for only PQ_BUS */
      }
    }
  }
  else
    J_Y--;

    J_Y++;
}
}

for(I=1,J_Y=1,m=0;I<DIMENSION+1;I++)
{
  if( I != slack_bus )
  {
    m++;
    JACOBIAN[J_Y][DIMENSION+1] = DELTA_P[I];
    J_Y++;
    if( m == MAX_BUS-2 )
      I = DIMENSION+1;
  }
}

for(I=1;I<DIMENSION+1;I++)
{
  check_pv = 1;
  for(m=1 ; m < (pv+1) ; m++)
    if(v_control[m] == I)
      check_pv = 0;

  if(( I != slack_bus )&&( check_pv != 0 ))

```

```

{
    JACOBIAN[J_Y][DIMENSION+1] = DELTA_Q[I];
    J_Y++;
}
}

```

```

/*----- Cal Gauss_jordan */

```

```

for(J_Y = 1 ; J_Y < DIMENSION+1 ; J_Y++)
{
    for(J_X = 1 ; J_X < DIMENSION+1 ; J_X++)
    {
        if(J_X != J_Y)
        {
            MULTI = JACOBIAN[J_X][J_Y]
                / JACOBIAN[J_Y][J_Y] ;
            for(k = 1 ; k < DIMENSION+2 ; k++)
            {
                JACOBIAN[J_X][k] = JACOBIAN[J_X][k] -
                    (MULTI * JACOBIAN[J_Y][k]) ;
            }
        }
    }
}

```

```

for(J_Y = 1 ; J_Y < DIMENSION+1 ; J_Y++)
{
    TEMP_JACOB[J_Y] = JACOBIAN[J_Y][DIMENSION+1]
        / JACOBIAN[J_Y][J_Y] ;
}

```

```

/*----- Make Del V_&_Ang */

```

```

for(k = 1 , J_Y = 1 , m = 0 ; k < DIMENSION+1 ; k++)
{
    if( k != slack_bus )
    {
        m++;
        DELTA_ANG[k] = TEMP_JACOB[J_Y] ;
        DELTA_ANG[k] = ( DELTA_ANG[k] * 180 ) / 3.141592654 ;
        J_Y++;
        if( m == MAX_BUS-2 )
            k = DIMENSION+1;
    }
}

```

```

for(k = 1 ; k < DIMENSION+1 ; k++)
{

```

```

    check_pv = 1;

```

```

    for(m=1 ; m < (pv+1) ; m++)

```

```

        if( v_control[m] == k )

```

```

            check_pv = 0;

```

```

    if(( k != slack_bus)&&( check_pv != 0 ))

```

```

        { DELTA_VOL[k] = TEMP_JACOB[J_Y] ;

```

```

          J_Y++;

```

```

        }

```

```

}
}
/*----- End of Make_Jacobian */

```

```

void Cal_Error(void)

```

```

{
    int k ;
    double SQ_V , SQ_A ;
    /*-----*/

```

```

    ERROR = 0 ;

```

```

    for(k = 1 ; k < MAX_BUS ; k++)

```

```

    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAGNI_V[k] = MAGNI_V[k] + DELTA_VOL[k] ;
ANGLE_V[k] = ANGLE_V[k] + DELTA_ANG[k] ;

```

```

if ( k != slack_bus)

```

```

{
    SQ_V = DELTA_VOL[k] * DELTA_VOL[k];
    SQ_V = sqrt(SQ_V);
    SQ_A = DELTA_ANG[k] * DELTA_ANG[k];
    SQ_A = sqrt(SQ_A);

```

```

    if(( SQ_V > ERR)||( SQ_A > ERR))
        ERROR =1;
}

```

```

}
} /*----- End of Cal_Error */

```

```

void Cal_PQO /* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */

```

```

{
    int p , q , k , Y , d , check_pv ;
    double e[SP_MAX_BUS] , f[SP_MAX_BUS] ;
    double REAL_1 , IMG_1 , TEMP_DATA ;
    /*----- Cal e f */
    for(p = 1 ; p < MAX_BUS ; p++)
    { TEMP_DATA = (ANGLE_V[p] * 3.141592654) / 180 ;
      e[p] = MAGNI_V[p] * cos(TEMP_DATA) ;
      f[p] = MAGNI_V[p] * sin(TEMP_DATA) ;
    }
}

```

```

/*----- Cal PQ */

```

```

for(p = 1 ; p < MAX_BUS ; p++)
{ for(q = 1 ; q < MAX_BUS ; q++)
  { if(q != p)
    {

```

```

      REAL_1 = e[p] * ( e[p] - e[q] ) + f[p] * ( f[p] - f[q] ) ;
      IMG_1 = e[p] * ( f[p] - f[q] ) - f[p] * ( e[p] - e[q] ) ;
      /* Ep^(Ep-Eq) = REAL_1 + jIMG_1 */

```

```

      TEMP_DATA = REAL_1 * Real_Ypq[p][q] ;
      FLOW_P[p][q] = TEMP_DATA - ( IMG_1 * Img_Ypq[p][q] ) ;
      FLOW_P[p][q] = BASE * FLOW_P[p][q] ;

```

```

      TEMP_DATA = REAL_1 * Img_Ypq[p][q] ;
      TEMP_DATA = TEMP_DATA + ( IMG_1 * Real_Ypq[p][q] ) ;
      FLOW_Q[p][q] = -1 * TEMP_DATA ;
      TEMP_DATA = Img_Ypq2[p][q] * ( e[p]*e[p] + f[p]*f[p] ) ;
      FLOW_Q[p][q] = FLOW_Q[p][q] - TEMP_DATA ;
      FLOW_Q[p][q] = BASE * FLOW_Q[p][q] ;
    }
  }
}

```

```

}
/*----- Cal G_PQ SLACK BUS */

```

```

for(k = 1 , REAL_1 = 0 , IMG_1 = 0 ; k < MAX_BUS ; k++)
{ if( k != slack_bus)
  { REAL_1 = REAL_1 + FLOW_P[slack_bus][k] ;
    G_P[slack_bus] = REAL_1 ;
    IMG_1 = IMG_1 + FLOW_Q[slack_bus][k] ;
    G_Q[slack_bus] = IMG_1 ;
  }
}

```

```

G_P[p] = G_P[p] + L_P[p] ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

G\_Q[q] = G\_Q[q] + L\_Q[q];

```
/*----- Cal G_Q PV BUS */
for(k = 1 , TEMP_DATA = 0 ; k < MAX_BUS ; k++)
{
    check_pv = 1 ;
    for(d = 1.; d < (pv+1) ; d++)
        if( v_control[d] == k )
            check_pv = 0;

    if( check_pv == 0 )
        {
            for(q = 1 ; q < MAX_BUS ; q++)
                {
                    if(q != k)
                        {
                            TEMP_DATA = TEMP_DATA + FLOW_Q[k][q] ;
                        }
                }
            G_Q[k] = TEMP_DATA + L_Q[k] ;
        }
}
} /*----- End of Cal_PQ */
```

```
void Make_Table(void)
{
    int p , q , k , Y , BUS ;
    /*-----*/
    printf("\n");
    printf("BUS_____VOLTAGE_MAGNITUDE_____PHASE_ANGLE");
    printf("\n\n");
    for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
    {
        printf("[%d]",BUS); Y = wherey();
        gotoxy( 15,Y) ; printf("%f",MAGNI_V[BUS]);
        gotoxy(36,Y) ; printf("%9.6f",ANGLE_V[BUS]);
        printf("\n");
    }
    getch();
    clrscr();
    printf("\n");
    printf("BUS___G[MW]___G[MVAR]___L[MW___VAR_]");
    printf("_ToBus___MW_____MVAR_____");
    for(p = 1 ; p < MAX_BUS ; p++)
    {
        printf("[%d]",p); Y = wherey();
        gotoxy( 7,Y) ; printf("%9.6f",G_P[p]);
        gotoxy(20,Y) ; printf("%9.6f",G_Q[p]);
        gotoxy(34,Y) ; printf("%4.1f",L_P[p]);
        gotoxy(41,Y) ; printf("%4.1f",L_Q[p]);
        for(q = 1 ; q < MAX_BUS ; q++)
        {
            if(q != p)
                {
                    gotoxy(49,Y) ; printf("%d",q);
                    gotoxy(53,Y) ; printf("%11.7f",FLOW_P[p][q]);
                    gotoxy(67,Y) ; printf("%11.7f",FLOW_Q[p][q]);
                    Y++;
                }
        }
        printf("\n");
    }
    printf("\n");
    if(( 25-Y) < (MAX_BUS-2))
```

```
{
  getch();
  clrscr();
}
}/*----- End of Make_Table */
```

```
/*-----*/
/*          END OF FILE          */
/*-----*/
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DECOUPLE NEWTON METHOD

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <ctype.h>
#include <process.h>

#define SP_MAX_BUS 6
#define ERR 0.000000001

// declare funciton
int input_new_filedata_RX(void);
int input_new_filedata_VPQ(void);
int get_old_filedata(void);

void calculate(void);
void getstring(void);
void Make_RX_to_GB(void);
void JACOB(void);
void Make_Delta_PQ(void);
void Sol_Matrx(void);
void Cal_Error(void);
void Cal_PQ(void);
void Make_Table(void);

// declare global variable
int slack_bus , v_control[SP_MAX_BUS], pv;
char filename[80];
FILE *fp;

/*----- External variable for Make_Ybus */
struct Data{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int t[SP_MAX_BUS];
double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Yp[SP_MAX_BUS];
double DIVIDE , DATA_1 , DATA_2 , DATA_3;
double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS] ,BASE;
double L_P[SP_MAX_BUS] , L_Q[SP_MAX_BUS];
double BASE_V,MAGNI_V[SP_MAX_BUS];
double ANGLE_V[SP_MAX_BUS];
} Dat;

int DIMEN_P,DIMEN_Q,MAX_BUS;
double pi = M_PI;
double MAGNI_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double ANGLE_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
/*----- Make_BVPQ */

double B[SP_MAX_BUS][SP_MAX_BUS];
int pv, X, Y, Z;
int B_X,B_Y;
double DAT;
double JCBN_P[SP_MAX_BUS][SP_MAX_BUS],JCBN_Q[SP_MAX_BUS][SP_MAX_BUS];
double BASE,BASE_V;
int CHECK_LOAD_BUS[SP_MAX_BUS];
double Psch[SP_MAX_BUS] , Qsch[SP_MAX_BUS];
double MAGNI_V[SP_MAX_BUS];
double ANGLE_V[SP_MAX_BUS];
double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double L_P[SP_MAX_BUS] . L_Q[SP_MAX_BUS];
/*----- Global variable for Make_Delta_PQ */

double DELTA_P[SP_MAX_BUS], DEL_PbyV[SP_MAX_BUS];
double DELTA_Q[SP_MAX_BUS], DEL_QbyV[SP_MAX_BUS];
/*----- Global variable for Make_Curl_PQ */

double DELTA_VOL[SP_MAX_BUS];
double DELTA_ANG[SP_MAX_BUS];
/*----- Global variable for Cal_PQ */

double FLOW_P[SP_MAX_BUS][SP_MAX_BUS] ;
double FLOW_Q[SP_MAX_BUS][SP_MAX_BUS] ;
double B[SP_MAX_BUS][SP_MAX_BUS];
double DELTA_P[SP_MAX_BUS];
double DELTA_Q[SP_MAX_BUS];
/*----- Make_Curl_PQ */

int ITERATION;
double ERROR;
/*----- Cal_PQ */

void main(void)
{

// int ITERATION ;
char c=0;

while (1) // always loop
{ // start while loop
clrscr();
printf("SELECT CHOICE:\n");
printf("1 => Read old file\n");
printf("2 => Write new file\n");
printf("3 => Exit\n");
c=getch();
switch(c) {
case '1':if (get_old_filedata()==1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

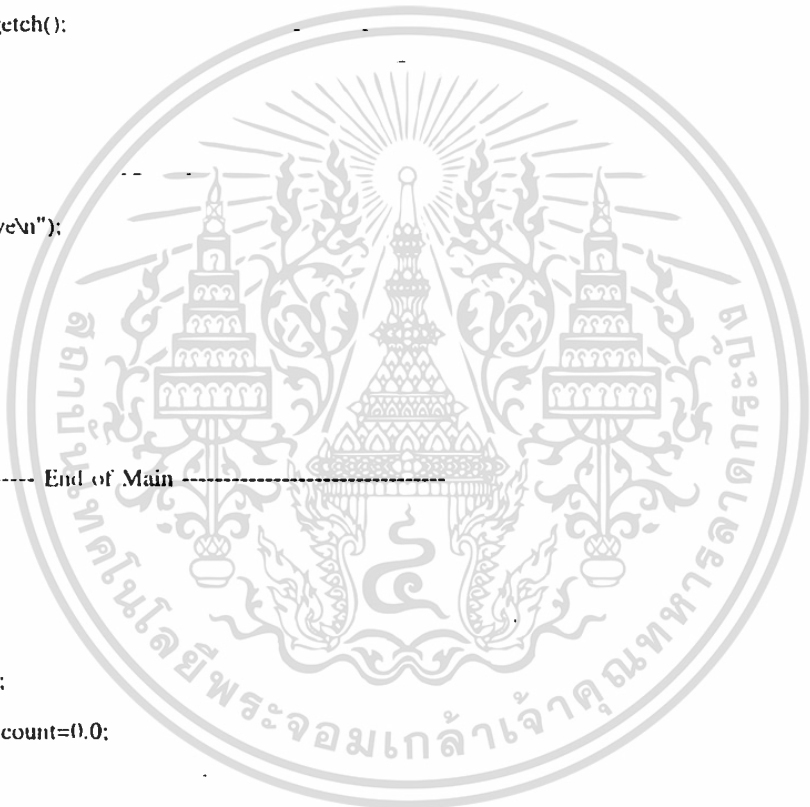
```

        Make_RX_to_GB();
        calculate();
        getch();
    }
    break;
case '2':if (input_new_filedata_RX()==1)
    {
        Make_RX_to_GB();
        if (input_new_filedata_VPQO==1)
            {
                calculate();
                getch();
            }
        }
    break;
case '3':printf("\nBye\n");
    exit(1);
} // end while loop
}
//----- End of Main -----

void calculate(void)
{
    struct dostime_t t;
    double t_iter=0.0,count=0.0;

    clrscr();
    printf("DECOUPLED NEWTON METHOD");
    t.hour = 0;
    t.minute = 0;
    t.second = 0;
    t.hsecond = 0;
    _dos_settime(&t);
    _dos_gettime(&t);
    for(ITERATION = 1 ; ITERATION < 1000 ; ITERATION++)

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    // gotoxy(1,3) ;
    // printf("ITERATION = [%d] = TOTAL ".ITERATION) ;
    JACOB() ;
    count++;
    Make_Delta_PQ();
    Sol_Matrx();
    Cal_Error() ;
    if( ERROR == 0 )
    {
        _dos_gettime(&t);
        gotoxy(1,3) ;
        printf("ITERATION = [%d] = TOTAL ".ITERATION) ;
        printf(" \ ERROR = [%10.8f] \n",ERR) ;
        Cal_PQ() ;
        ITERATION = 1002;
    }
}
Make_Table();
gotoxy(56,40);
t_iter = (t.second + (t.hsecond*0.01))/count;
// printf("Total time : %d:%d:%d.%d ",t.hour,t.minute,t.second,t.hsecond);
printf("Time per Iteration : %f ",t_iter);
}

void getstring(void)
{
    int ch,X,Y;
    int i=0;
    while(ch!=13)
    {
        ch = getch();
        ch = tolower(ch);
        if ((ch>=21)&&(ch<=176))
        {
            filename[i] = ch;
            printf("%c",ch);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i++;
    }
    if (ch==47)
    {
        filename[i] = '\\';
        i++;
    }
    if (ch==8)
    {
        i--;
        X=wherex();
        Y=wherey();
        gotoxy(X-1,Y);
        printf(" ");
        gotoxy(X-1,Y);
    }
    if (ch==13)
    {
        filename[i]='\000';
        printf("\n");
    }
}

```

```

int get_old_filedata(void)

```

```

{
    int d=0,P,Q,n,m;
    int success = 1;

    printf("\nEnter file name: ");
    getstring();
    if((fp=fopen(filename,"r"))==NULL)
    {
        printf("\nCan not open file !!!\nPress any key to continue...");
        success = 0;

        getch();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else
{
    // Clear array
    for(P = 1 ; P < MAX_BUS ; P++)
    { for(Q = 1 ; Q < MAX_BUS ; Q++)
        { Img_Ypq2[P][Q] = 0;
          Real_Zpq[P][Q] = 0;
          Img_Zpq[P][Q] = 0;
        }
    }

    // read data
    fread(&Dat,sizeof(struct Data),1,fp);
    // display data
    clrscr();
    MAX_BUS = Dat.Bus_num+1;
    printf("Number of bus : %d\n",MAX_BUS-1);
    for(P = 1; P < MAX_BUS ;P++)
    { // start for1 loop
      for(Q =1 ;Q < MAX_BUS ;Q++)
      { // start for2 loop
        Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q];
        Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q];
        Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];
        if(P!=Q)
        {
            printf("Impedance and line charging between bus %d and %d\n",P,Q );
            printf(" R      : %f\n",Real_Zpq[P][Q]);
            printf(" X      : %f\n",Img_Zpq[P][Q]);
            printf(" L chrg : %f\n",Img_Ypq2[P][Q]);
            getch();
        }
      } // end for2 loop
    } // end for1 loop
    pv= 0;
    mv = 0;

```

```

n = 1;
for(d=1;d<MAX_BUS;d++)
{
    switch(Dat.t[d]) {
        case 1 : n--;
            slack_bus = d;
            break;

        case 2 : m++;
            v_control[m] = d;
            pv = m ;
            break;

        case 3 : break;
    }
    MAGNI_V[d] = Dat.MAGNI_V[d];
    ANGLE_V[d] = Dat.ANGLE_V[d];
    G_P[d] = Dat.G_P[d];
    G_Q[d] = Dat.G_Q[d];
    L_P[d] = Dat.L_P[d];
    L_Q[d] = Dat.L_Q[d];
    printf("\nMagnitude and angle of Bus no. %d \n",d);
    printf("  Magnitude : %f\n",MAGNI_V[d]);
    printf("  Angle      : %f\n",ANGLE_V[d]);
    printf(" Mw of Gen. of Bus no. %d : %f\n",d,G_P[d]);
    printf(" Mvar of Gen. of Bus no. %d : %f\n",d,G_Q[d]);
    printf(" Mw of Load of Bus no. %d : %f\n",d,L_P[d]);
    printf(" Mvar of Load of Bus no. %d : %f\n",d,L_Q[d]);
    getch();
}

```

BASE = Dat.BASE;

BASE\_V = Dat.BASE\_V;

printf("\nBase of Power : %f\n",BASE);

printf("Base of Voltage : %f\n",BASE\_V);

getch();

fclose(fp);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(success);
}

int input_new_filedata_RX(void)
{
char ch;
int P,Q;
int success=1;

printf("\nEnter file name: ");
getstring();
if((fp=fopen(filename,"r"))!=NULL)
{
printf("\nWARNING !\n");
printf(" Overwrite exist file ? (Y/N) ");
ch=getch();
if ((ch=='N')||(ch=='\n'))
{
success = 0;
goto out_RX;
}
}
fclose(fp);
if((fp=fopen(filename,"w"))==NULL)
{ // start if3
success = 0;
printf("\nError !!!\nPress any to continue...");
getch();
} // end if3
else
{ // start else3
printf("\nEnter number of Bus :");
scanf("%d",&MAX_BUS);
Dat.Bus_num = MAX_BUS;
MAX_BUS = MAX_BUS+1;
for(P = 1 ; P < MAX_BUS ; P++)

```

```

{ for(Q = 1 ; Q < MAX_BUS ; Q++)
    { Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0 ; /* Clear all array */
      Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q] = 0 ; /* for next loop */
      Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0 ;
    }
}

```

```

for(P = 1; P < MAX_BUS :P++)

```

```

{ // start for1 loop

```

```

    for(Q =1 :Q < MAX_BUS ;Q++)

```

```

    { // start for2-loop

```

```

        if(P<Q)

```

```

            { // start if4

```

```

                printf("Is there Impedance and line charging between bus %d and %d(Y or N) : \n",P,Q )

```

```

                ch = getch();

```

```

                ch = toupper(ch);

```

```

                if(ch == 'Y')

```

```

                {

```

```

                    // printf("\n");

```

```

                    printf(" R   : ");

```

```

                    scanf("%le",&Dat.Real_Zpq[P][Q]);

```

```

                    Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q];

```

```

                    printf(" X   : ");

```

```

                    scanf("%le",&Dat.Img_Zpq[P][Q]);

```

```

                    Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q];

```

```

                    printf(" L chrg : ");

```

```

                    scanf("%le",&Dat.Img_Ypq2[P][Q]);

```

```

                    Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];

```

```

                }

```

```

            } else

```

```

            {

```

```

                Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q] = 0;

```

```

                Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0;

```

```

                Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dat.Real_Zpq[Q][P] = Dat.Real_Zpq[P][Q];
Dat.Img_Zpq[Q][P] = Dat.Img_Zpq[P][Q];
Dat.Img_Ypq2[Q][P] = Dat.Img_Ypq2[P][Q];
Real_Zpq[Q][P] = Real_Zpq[P][Q];
Img_Zpq[Q][P] = Img_Zpq[P][Q];
Img_Ypq2[Q][P] = Img_Ypq2[P][Q];
} // end if4
else
{ // start else4
    if(P==Q)
    { Real_Zpq[P][P] = Dat.Real_Zpq[P][P] =0;
      Img_Zpq[P][P] = Dat.Img_Zpq[P][P] =0;
      Img_Ypq2[P][P] = Dat.Img_Ypq2[P][P] =0;
    }
} //end else4
} // end for2 loop
} // end for1 loop
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} // end else3
out_RX : return(success);
}

```

```

void Make_RX_to_GB(void)

```

```

{
    double DIVIDE , DATA_1 , DATA_2 , DATA_3;
    double Img_Yp[SP_MAX_BUS];
    double G[SP_MAX_BUS][SP_MAX_BUS];
    int P, Q, COUNT;
    char ch ;

    for(P = 1; P < MAX_BUS ; P++)
    { for(Q = 1 ; Q < MAX_BUS ; Q++)
        { DIVIDE = Real_Zpq[P][Q]*Real_Zpq[P][Q] ;
          DIVIDE = DIVIDE + Img_Zpq[P][Q]*Img_Zpq[P][Q] ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ( DIVIDE != 0 )
(
    Real_Ypq[P][Q] = Real_Zpq[P][Q] / DIVIDE ;
    Img_Ypq[P][Q] = -1*(Img_Zpq[P][Q]/DIVIDE) ;
)
/* -1 Because conjugate */
else
(
    Real_Ypq[P][Q] = 0 ;
    Img_Ypq[P][Q] = 0 ;
)
}
}
/*----- Cal Yp */
for(P = 1 ; P < MAX_BUS ; P++) Img_Yp[P]=0 /*Protecthang*/

for(P = 1 ; P < MAX_BUS ; P++)
{
    for(Q = 1 ; Q < MAX_BUS ; Q++)
    {
        Img_Yp[P] = Img_Yp[P] + Img_Ypq2[P][Q] ;
    }
}
/*----- Cal G_B */
for(P = 1 ; P < MAX_BUS ; P++)
{
    for(Q = 1 ; Q < MAX_BUS ; Q++)
    {
        G[P][Q] = 0 ;
        B[P][Q] = 0 ;
    }
    /* Protect hang on next loop */
}

for(P = 1 ; P < MAX_BUS ; P++)
    B[P][P] = B[P][P] + Img_Yp[P] ;
/* Self admittances must include Yp */

for(P = 1 ; P < MAX_BUS ; P++)
{
    for(Q = 1 ; Q < MAX_BUS ; Q++)
    {
        if( P == Q )
        {
            for(COUNT = 1 ; COUNT < MAX_BUS ; COUNT++)
            {
                G[P][P] = G[P][P] + Real_Ypq[P][COUNT];
                B[P][P] = B[P][P] + Img_Ypq[P][COUNT] ;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        }
    }
else
    {
        G[P][Q] = -1*(Real_Ypq[P][Q]) ;
        B[P][Q] = -1*(Img_Ypq[P][Q]) ;
    } /* Mutual admittances value are negative */
}
}
/*----- Cal' Polar */
for(P = 1 ; P < MAX_BUS ; P++)
{
    for(Q = 1 ; Q < MAX_BUS ; Q++)
    {
        DATA_1 = B[P][Q]*B[P][Q] + G[P][Q]*G[P][Q] ;
        MAGNI_Ybus[P][Q] = sqrt(DATA_1) ;

        if( G[P][Q] != 0 ) DATA_1 = B[P][Q] / G[P][Q] ;
        else DATA_1 = 0 ;

        DATA_1 = atan(DATA_1) ; /* Value are in radian */
        DATA_1 = ( DATA_1 * 180 ) / pi ;
        if( G[P][Q] < 0 ) DATA_1 = DATA_1 + 180 ;
        ANGLE_Ybus[P][Q] = DATA_1 ;
    }
}
}
//----- End of Make_RX_to_GB -----

```

```

int input_new_filedata_VPQ(void)

```

```

{
    int k,d,m,n,t(SP_MAX_BUS);
    double TEMP_DATA;
    int success=1;

    clrscr();
    pv = 0;
    m = 0;
    n = 1;

    if((fp=fopen(filename,"w"))==NULL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ // start if1
    success = 0;
    printf("\nError !!!\nPress any to continue...");
    getch();
} // end if1
else
{ // start else1
    printf("%52s\n","<< TYPE OF BUS SELECTION >>");
    printf("%44s\n","1. SLACK BUS");
    printf("%57s\n","2. VOLTAGE CONTROLLED BUS");
    printf("%43s\n","3. LOAD BUS");
    printf("%45s\n","( SELECT 1-3 )");
    for (d=1; d < MAX_BUS ;d++)
    {
        printf(" TYPE OF BUS [%d] SELECT ==>> ",d);
        scanf("% d",&t[d]);
        Dat.t[d] = t[d];
    }
    clrscr();
    for (d=1; d < MAX_BUS ;d++ )
    {
        switch(t[d]) {
            case 1 : n--;
                slack_bus = d;
                printf(" BUS [%d] SLACK BUS >> ",d);
                break;
            case 2 : m++;
                v_control[m] = d;
                pv = m ;
                printf(" BUS [%d] PV BUS  >> ",d);
                break;
            case 3 : printf(" BUS [%d] LOAD BUS >> ",d);
                break;
            default: printf("%35s","<<< DATA ERROR ! >>>");
                success = 0;
                goto out_VPQ;
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if(n<0)
    {
        printf(" <<< DATA ERROR ! >>>\007\n");
        printf(" DUPLICATE SLACK BUS !\n");
        printf(" Press any key to continue...");
        success = 0;
        getch();
        goto out_VPQ;
    }

```

```

printf("%18s","MAGNITUDE : ");
scanf("%le",&MAGNI_V[d]);
Dat.MAGNI_V[d] = MAGNI_V[d];
printf("%40s","ANGLE : ");
scanf("%le",&ANGLE_V[d]);
Dat.ANGLE_V[d] = ANGLE_V[d];

```

```

if( d != slack_bus )

```

```

{ /*
    if( d == v_control[m] )
    { /*
        printf("%40s","MW OF GEN : ");
        scanf("%le",&G_P[d]);
        Dat.G_P[d] = G_P[d];
        printf("%40s","MVAR OF GEN : ");
        scanf("%le",&G_Q[d]);
        Dat.G_Q[d] = G_Q[d];
    /* }
    else
    {
        G_P[d] = 0 ;
        G_Q[d] = 0 ;
    /* }
        printf("%40s","MW OF LOAD : ");
        scanf("%le",&L_P[d]);
        Dat.L_P[d] = L_P[d];

```

```

printf("%-40s","MVAR OF LOAD : ");
scanf("%le",&L_Q[d]);
Dat.L_Q[d] = L_Q[d];
}
else
{
Dat.G_P[slack_bus] = G_P[slack_bus] = 0;
Dat.G_Q[slack_bus] = G_Q[slack_bus] = 0;
Dat.L_P[slack_bus] = L_P[slack_bus] = 0;
Dat.L_Q[slack_bus] = L_Q[slack_bus] = 0;
}
}
printf("Enter BASE of POWER : ");
scanf("%le",&BASE);
Dat.BASE = BASE;
printf("Enter BASE of VOLTAGE : ");
scanf("%le",&BASE_V);
Dat.BASE_V = BASE_V;
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} //end else1
out_VPQ: return(succes);
}

void JACOB(void)
{
int d,i,m,k,l,check_pv ;
double CURL_P_DELTA[SP_MAX_BUS][SP_MAX_BUS],
CURL_Q_V[SP_MAX_BUS][SP_MAX_BUS];
double DATA_1,DATA_2,DATA_3,DATA_P,DATA_Q v,u;

for(d=1 ; d < MAX_BUS ; d++)
{
Psch[d] = G_P[d] - L_P[d] ;
Psch[d] = Psch[d] / BASE;

```

```
Qsch[d] = (i_Q[d] - L_Q[d]);
```

```
Qsch[d] = Qsch[d]/BASE;
```

```
}
```

```
if(ITERATION>200)
```

```
u=100;
```

```
else
```

```
u=40;
```

```
v=(ITERATION-1)/u;
```

```
if(ceil(v)==(v)) {
```

```
DIMEN_P=0;
```

```
Y'= 1;
```

```
for(I = 1 ; I < MAX_BUS ; I++)
```

```
if( I != slack_bus )
```

```
{ DATA_P = 0 ;
```

```
for(k = 1 ; k < MAX_BUS ; k++)
```

```
{ if( k == I ) k++ ;
```

```
DATA_1 = MAGNI_V[I] * MAGNI_V[k] *  
MAGNI_Ybus[I][k] ;
```

```
DATA_2 = ANGLE_V[I] - ANGLE_V[k] -  
ANGLE_Ybus[I][k] ;
```

```
DATA_2 = ( DATA_2 * 3.141592654 ) / 180 ;
```

```
DATA_3 = sin( DATA_2 ) ;
```

```
DATA_P = DATA_P + ( DATA_1 * DATA_3 ) ;
```

```
CURL_P_DELTA[I][k] = 1 * DATA_1 * DATA_3 ;
```

```
/* I not equal k */
```

```
}
```

```
CURL_P_DELTA[I][I] = -1 * DATA_P ; /* for I == k */
```

```
}
```

```

{ if (i != slack_bus )
  { DIMEN_P++;
    for ( B_X=1, X=1 ; B_X < MAX_BUS ; B_X++,X++)
      { if ( B_X != slack_bus)
        {
          JCBN_P[Y][X] = CURL_P_DELTA[i][B_X];
        }
        else
          X--;
      }
      Y++;
    }
  }
DIMEN_P++;

```

```

/*-----JCBN_Q--*/

```

```

for(I = 1 ; I < MAX_BUS ; I++)
  if ( I != slack_bus )
  { DATA_Q = 0 ;
    for(k = 1 ; k < MAX_BUS ; k++)
      { if(k == I) k++ ;
        DATA_1 = MAGNI_V[k] * MAGNI_Ybus[I][k] ;
        DATA_2 = ANGLE_V[I] - ANGLE_V[k] -
          ANGLE_Ybus[I][k] ;
        DATA_2 = ( DATA_2 * 3.141592654 ) / 180 ;

        DATA_3 = sin( DATA_2 ) ;
        CURL_Q_V[I][k] = ( DATA_1 * DATA_3 ) ;
        DATA_Q = DATA_Q + ( DATA_1 * DATA_3 ) ;
      }
    DATA_1 = MAGNI_V[I] * MAGNI_Ybus[I][I] ;

    DATA_2 = ( ANGLE_Ybus[I][I] * 3.141592654 ) / 180 ;

    DATA_3 = sin( -1 * DATA_2 ) ;
    CURL_Q_V[I][I] = ( 2 * DATA_1 * DATA_3 ) + DATA_Q ;

```

```

}                                     /* I == k */

DIMEN_Q = 0;
for(i = 1 , Y = 1 ; i < MAX_BUS ; i ++ ,Y++)
{ check_pv = 1 ;
  for(m=1 ;m < (pv+1) ; m++)
    if(v_control[m] == i)
      check_pv = 0 ;
  if ((i != slack_bus)&&(check_pv != 0))
  { DIMEN_Q++;
    for(B_X = 1 , X = 1 ; B_X < MAX_BUS; B_X++,X++)
    { check_pv = 1 ;
      for(m=1 ;m < (pv+1) ; m++)
        if(v_control[m] == B_X)
          check_pv = 0 ;
      if ((B_X != slack_bus)&&(check_pv != 0 ))
      {
        JCBN_Q[Y][X] = CURL_Q_V[i][B_X];
      }
      else
        X--;
    }
  }
  Y--;
}
DIMEN_Q++;
}
}
/*-----END OF JACOB-----*/

```

```
void Make_Delta_PQ(void)
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double DATA_P, DATA_Q, DATA_1, DATA_2, DATA_3;
int I, K;
/*-----*/

for(I = 1 ; I < MAX_BUS ; I++) /*--- 1 Slack bus ---*/
if( I != slack_bus )
{ DATA_P = 0 ; DATA_Q = 0 ;
for(K = 1 ; K < MAX_BUS ; K++)
{ DATA_1 = MAGNI_V[I] * MAGNI_V[K] * MAGNI_Ybus[I][K] ;
DATA_2 = ANGLE_V[I] - ANGLE_V[K] - ANGLE_Ybus[I][K] ;
DATA_2 = ( DATA_2 * pi ) / 180 ;

DATA_3 = cos( DATA_2 ) ;
DATA_P = DATA_P + ( DATA_1 * DATA_3 ) ;

DATA_3 = sin( DATA_2 ) ;
DATA_Q = DATA_Q + ( DATA_1 * DATA_3 ) ;
}
DELTA_P[I] = Psch[I] - DATA_P ;
DEL_PbyV[I] = DELTA_P[I] / MAGNI_V[I];

DELTA_Q[I] = Qsch[I] - DATA_Q ;
DEL_QbyV[I] = DELTA_Q[I] / MAGNI_V[I];
}
}
/*----- End of Make_Delta_PQ */

```

```

void Sol_Matrx()
{
int i, m, check_pv;
int BUS;
double MULTANS[SP_MAX_BUS];
double JC_P[SP_MAX_BUS][SP_MAX_BUS], JC_Q[SP_MAX_BUS][SP_MAX_BUS];
/*-----JCBN <= DelPbyV */

```

```

for(i=1 ; i<DIMEN_P ; i++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JC_P[i][m] = JCBN_P[i][m];
```

```
for( Y=1 ,i=1; Y < DIMEN_P ; Y++,i++)
```

```
{ if ( i != slack_bus)
```

```
    JCBN_P[Y][DIMEN_P] = DEL_PbyV[i];
```

```
else
```

```
    Y = Y-1;
```

```
}
```

```
/*-----Cal Del_AngleV */
```

```
for(Y = 1 ; Y < DIMEN_P ; Y++)
```

```
{ for(X = 1 ; X < DIMEN_P ; X++)
```

```
{ if( X != Y)
```

```
    { MULTI= JCBN_P[X][Y] / JCBN_P[Y][Y];
```

```
      for(Z = 1 ; Z < (DIMEN_P+1);Z++)
```

```
        JCBN_P[X][Z] = JCBN_P[X][Z]-(MULTI * JCBN_P[Y][Z]);
```

```
    }
```

```
}
```

```
}
```

```
for (Y = 1 ;Y < DIMEN_P; Y++)
```

```
    ANS[Y] = JCBN_P[Y][DIMEN_P] / JCBN_P[Y][Y];
```

```
/*-----Angle_V + Del_AngleV */
```

```
for(Y = 1 ,Z = 1 ;Y < MAX_BUS; Y++,Z++)
```

```
{ if ( Y != slack_bus)
```

```
{
```

```
    DELTA_ANG[Y] = ANS[Z];
```

```
    ANGLE_V[Y] = ANGLE_V[Y] + 1.8*DELTA_ANG[Y];
```

```
}
```

```
else
```

```
    Z--;
```

```
}
```

```
for(i=1 ; i<DIMEN_P ; i++)
```

เอกสารถูกใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JCBN_P[i][m] = JC_P[i][m] ;
```

```
/*-----JCBN <= DelQbyV */
```

```
for(i=1 ; i<DIMEN_Q ; i++)
```

```
for(m=1 ; m<DIMEN_Q ; m++)
```

```
JC_Q[i][m] = JCBN_Q[i][m];
```

```
for (Y = 1,i = 1 ; i< MAX_BUS ; Y++, i ++)
```

```
{ check_pv = 1 ;
```

```
for(m=1 ;m < (pv+1) ; m++)
```

```
if(v_control[m] == i )
```

```
check_pv = 0;
```

```
if (( i !=slack_bus)&&(check_pv != 0))
```

```
JCBN_Q[Y][DIMEN_Q] = DEL_QbyV[i];
```

```
else
```

```
Y--;
```

```
}
```

```
/*-----Cal Del_MagniV */
```

```
for(Y=1 ;Y < DIMEN_Q; Y++)
```

```
{ for(X = 1 ; X < DIMEN_Q ; X++)
```

```
{ if (X != Y)
```

```
{ MULTI = JCBN_Q[X][Y] / JCBN_Q[Y][Y];
```

```
for(Z = 1; Z < DIMEN_Q+1;Z++)
```

```
JCBN_Q[X][Z] = JCBN_Q[X][Z]-(MULTI * JCBN_Q[Y][Z]);
```

```
}
```

```
}
```

```
}
```

```
for(Y = 1 ; Y < DIMEN_Q;Y++)
```

```
ANS[Y] = JCBN_Q[Y][DIMEN_Q] / JCBN_Q[Y][Y] ;
```

```
/* -----Magni_V+ Del_MagniV */
```

```
for(Y = 1 , Z = 1 ;Y < MAX_BUS ; Y++,Z++)
```

```
{ check_pv = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(m=1 ;m < (pv+1) ; m++)
    if(v_control[m] == Y )
        check_pv = 0 ;
if((Y != slack_bus)&&(check_pv != 0))
{
    DELTA_VOL[Y] = ANS[Z];
    MAGNI_V[Y] = MAGNI_V[Y] + 1.8*DELTA_VOL[Y];
}
else
    Z--;
}

for(i=1 ; i<DIMEN_Q : i++)
for(m=1 ; m<DIMEN_Q ; m++)
    JCBN_Q[i][m] = JC_Q[i][m] ;
}
/* -----END OF SOL_Matrx----- */

void Cal_Error()
{
    int k;
    double SQ_V , SQ_A ;

    ERROR=0;
    for(k = 1 ; k < MAX_BUS : k++ )
        if(k!= slack_bus)
        {
            SQ_V = DELTA_VOL[k] * DELTA_VOL[k];
            SQ_V = sqrt(SQ_V);
            SQ_A = DELTA_ANG[k] * DELTA_ANG[k];
            SQ_A = sqrt(SQ_A);

            if((SQ_V>ERR)||(SQ_A>ERR))
                ERROR=1;
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*----- End of Cal_Error */
```

```
void Cal_PQ(void)
```

```
/* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
```

```
{
```

```
int p, q, k, Y, d, check_pv;
```

```
double e[SP_MAX_BUS], f[SP_MAX_BUS];
```

```
double REAL_1, IMG_1, TEMP_DATA;
```

```
/*----- Cal e_f */
```

```
for(p = 1; p < MAX_BUS; p++)
```

```
{ TEMP_DATA = (ANGLE_V[p] * pi) / 180;
```

```
e[p] = MAGNI_V[p] * cos(TEMP_DATA);
```

```
f[p] = MAGNI_V[p] * sin(TEMP_DATA);
```

```
}
```

```
/*----- Cal PQ */
```

```
for(p = 1; p < MAX_BUS; p++)
```

```
{ for(q = 1; q < MAX_BUS; q++)
```

```
{ if(q != p)
```

```
{
```

```
REAL_1 = e[p] * ( e[q] - e[q] ) + f[p] * ( f[p] - f[q] );
```

```
IMG_1 = e[p] * ( f[p] - f[q] ) - f[p] * ( e[p] - e[q] );
```

```
/* Ep^(Ep-Eq) = REAL_1 + jIMG_1 */
```

```
TEMP_DATA = REAL_1 * Real_Ypq[p][q];
```

```
FLOW_P[p][q] = TEMP_DATA - ( IMG_1 * Img_Ypq[p][q] );
```

```
FLOW_P[p][q] = BASE * FLOW_P[p][q];
```

```
TEMP_DATA = REAL_1 * Img_Ypq[p][q];
```

```
TEMP_DATA = TEMP_DATA + ( IMG_1 * Real_Ypq[p][q] );
```

```
FLOW_Q[p][q] = -1 * TEMP_DATA;
```

```
TEMP_DATA = Img_Ypq2[p][q] * ( e[p]*e[p] + f[p]*f[p] );
```

```
FLOW_Q[p][q] = FLOW_Q[p][q] - TEMP_DATA;
```

```
FLOW_Q[p][q] = BASE * FLOW_Q[p][q];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

/*----- Cal G_PQ SLACK BUS */
for(k = 1 , REAL_1 = 0 , IMG_1 = 0 ; k < MAX_BUS ; k++)
(
if( k != slack_bus )
(
REAL_1 = REAL_1 + FLOW_P[slack_bus][k] ;
G_P[slack_bus] = REAL_1 ;
IMG_1 = IMG_1 + FLOW_Q[slack_bus][k] ;
G_Q[slack_bus] = IMG_1 ;
)
)
)
)
G_P[p] = G_P[p] + L_P[p] ;
G_Q[q] = G_Q[q] + L_Q[q] ;

/*----- Cal G_Q PV BUS */
for(k = 1 , TEMP_DATA = 0 ; k < MAX_BUS ; k++)
(
check_pv = 1 ;
for(d = 1 ; d < (pv+1) ; d++)
if( v_control[d] == k )
check_pv = 0;

if( check_pv == 0 )
(
for(q = 1 ; q < MAX_BUS ; q++)
(
if(q != k)
TEMP_DATA = TEMP_DATA + FLOW_Q[k][q] ;
)
)
G_Q[k] = TEMP_DATA + L_Q[k] ;
)
)
)
)
)
)

```

```

void Make_Table(void)
{
    int p , q , k , BUS;
    int y_pos;          /* position in y-axis */

    printf("\nBUS_____VOLTAGE_MAGNITUDE_____PHASE_ANGLE\n\n");
    for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
    {
        printf("[%d]".BUS) :
        y_pos = wherey() ;
        gotoxy( 15,y_pos) ; printf("%f",MAGNI_V[BUS]) ;
        gotoxy(36,y_pos) ; printf("%9.6f",ANGLE_V[BUS]) ;
        printf("\n") ;
    }
    getch();
    clrscr();
    printf("\n");
    printf("BUS_____G[MW]_____G[MVAR]_____L[MW]_____VAR]_");
    printf(" _ToBus_____MW_____MVAR_____ \n\n") ;
    for(p = 1 ; p < MAX_BUS ; p++)
    {
        printf("[%d]".p) ; y_pos = wherey() ;
        gotoxy( 7,y_pos) ; printf("%9.6f",G_P[p]) ;
        gotoxy(20,y_pos) ; printf("%9.6f",G_Q[p]) ;
        gotoxy(34,y_pos) ; printf("%4.1f",L_P[p]) ;
        gotoxy(41,y_pos) ; printf("%4.1f",L_Q[p]) ;
        for(q = 1 ; q < MAX_BUS ; q++)
        {
            if(q != p)
            {
                gotoxy(49,y_pos) ; printf("%d",q) ;
                gotoxy(53,y_pos) ; printf("%11.7f",FLOW_P[p][q]) ;
                gotoxy(67,y_pos) ; printf("%11.7f",FLOW_Q[p][q]) ;
                y_pos++ ;
            }
        }
        printf("\n") ;
    }
    printf("\n") ;
    if(( 25-y_pos ) < (MAX_BUS-2))

```

```
{  
    getch();  
    clrscr();  
}  
}  
/*----- End of Make_Table -----*/
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Fast Decouple Load Flow Analysis

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <ctype.h>
#include <process.h>
#include <graphics.h>

#define SP_MAX_BUS 6
#define ERR 0.00000001
#define r 11
#define span 36
#define SEX 37
#define position 10

// declare x,y for charting
//int x[6],y[6];
int x[6] = {0,180,390,310,332,387};
int y[6] = {0,80,80,180,295,420};

// declare functon
int input_new_filedata_RX(void);
int input_new_filedata_VPQ(void);
int get_old_filedata(void);

void calculate(void);
void getstring(void);
void Make_RX_to_GB(void);
void JACOB(void);
void Make_Delta_PQ(void);
void Sol_Matrx(void);
void Cal_Error(void);
void Cal_PQ(void);
void Make_Table(void);
void Setgraph(void);
void Line_main1(void);
void Line_main2(void);
void Line_main3(void);
void Line_main4(void);
void Line_main5(void);

// declare global variable
int MAX_BUS,slack_bus , v_control[SP_MAX_BUS], pv.O[SP_MAX_BUS],
n1,n2,n3,n4,n5;
char filename[80],Relat[SP_MAX_BUS][SP_MAX_BUS] , Relat_G[SP_MAX_BUS],
Relat_L[SP_MAX_BUS];
FILE *fp;

/*----- External variable for Make_Ybus */
struct Data{
int Bus_num;
int t[SP_MAX_BUS];
int BUBB[SP_MAX_BUS];
int O[SP_MAX_BUS];
```

```

char Relat[SP_MAX_BUS][SP_MAX_BUS];
char Relat_G[SP_MAX_BUS];
char Relat_L[SP_MAX_BUS];
double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Yp[SP_MAX_BUS];
double DIVIDE , DATA_1 , DATA_2 , DATA_3;
double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS] ,BASE;
double L_P[SP_MAX_BUS] , L_Q[SP_MAX_BUS];
double BASE_V,MAGNI_V[SP_MAX_BUS];
double ANGLE_V[SP_MAX_BUS];
} Dat;

int DIMEN_P,DIMEN_Q,MAX_BUS;
double pi = M_PI;
double MAGNI_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double ANGLE_Ybus[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS];
double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS];
double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS];
double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS];
//----- Make_BVPQ

double B[SP_MAX_BUS][SP_MAX_BUS];
int BUBB[5],pv, X, Y, Z;
int B_X,B_Y;
double DAT;
double JCBN_P[SP_MAX_BUS][SP_MAX_BUS],JCBN_Q[SP_MAX_BUS][SP_MAX_BUS];
double BASE,BASE_V;
int CHECK_LOAD_BUS[SP_MAX_BUS];
double Psch[SP_MAX_BUS] , Qsch[SP_MAX_BUS];
double MAGNI_V[SP_MAX_BUS];
double ANGLE_V[SP_MAX_BUS];
double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS];
double L_P[SP_MAX_BUS] , L_Q[SP_MAX_BUS];
//----- Global variable for Make_Delta_PQ

double DELTA_P[SP_MAX_BUS], DEL_PbyV[SP_MAX_BUS];
double DELTA_Q[SP_MAX_BUS], DEL_QbyV[SP_MAX_BUS];
//----- Global variable for Make_Curl_PQ
double DELTA_VOL[SP_MAX_BUS];
double DELTA_ANG[SP_MAX_BUS];
//----- Global variable for Cal_PQ
double FLOW_P[SP_MAX_BUS][SP_MAX_BUS] ;
double FLOW_Q[SP_MAX_BUS][SP_MAX_BUS] ;
double B[SP_MAX_BUS][SP_MAX_BUS];
double DELTA_P[SP_MAX_BUS];
double DELTA_Q[SP_MAX_BUS];
//----- Make_Curl_PQ
int ITERATION;
double ERROR;
//----- Cal_PQ

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void main(void)
{
int ITERATION;
char c=0;

while(1) // always loop
{ // start while loop
clrscr();
printf("SELECT CHOICE:\n");
printf("1 => Read old file\n");
printf("2 => Write new file\n");
printf("3 => Exit\n");
c=getch();
switch(c)
{
case '1':if (get_old_filedata()==1)
{
Make_RX_to_GB();
Setgraph();
calculate();
getch();
}
break;
case '2':if (input_new_filedata_RX()==1)
{
Make_RX_to_GB();
if (input_new_filedata_VPQ()==1)
{
Setgraph(); // Draw Bus
calculate();
getch();
}
}
break;
case '3':printf("\nBye\n");
exit(1);
}
} // end infinite while loop
}
//----- End of Main -----

```

```

void calculate(void)
{
struct dostime_t t;

clrscr();
printf("FAST DECOUPLE METHOD");
t.hour = 0;
t.minute = 0;
t.second = 0;
t.hsecond = 0;
_dos_settime(&t);
_dos_gettime(&t);
for(ITERATION = 1 ; ITERATION < 1000 ; ITERATION++)
{
gotoxy(1,3);
printf("ITERATION = [%d] = TOTAL ".ITERATION);
JACOB();
}
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Make_Delta_PQ():
Sol_Matrx():
Cal_Error():
if( ERROR == 0 )
{
_dos_gettime(&t);
gotoxy(1,3);
printf("ITERATION = [%d] = TOTAL ".ITERATION);
printf("\t ERROR = [%10.8f] \n".ERR);
Cal_PQ();
ITERATION = 1002;
}
}
Make_Table():
gotoxy(56,40);
printf("Total time : %d:%d:%d.%d ",t.hour,t.minute,t.second,t.hsecond);
}
}

void getstring(void)
{
int ch,X,Y;
int i=0;

while(ch!=13)
{
ch = getch();
ch = tolower(ch);
if ((ch>=21)&&(ch<=176))
{ filename[i] = ch;
printf("%c",ch);
i++;
}
if (ch==47)
{ filename[i] = "\ ";
i++;
}
if (ch==8)
{ i--;
X=wherex();
Y=wherey();
gotoxy(X-1,Y);
printf(" ");
gotoxy(X-1,Y);
}
if (ch==13)
{ filename[i]='\000';
printf("\n");
}
}
}

int get_old_filedata(void)
{
int d=0,P,Q,n,m;
int success = 1;

```

printf("\nENTER file name: ");
 getstring();
 ไม่รับประกันใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((fp=fopen(filename,"r"))==NULL)
{ printf("\nCan not open file !!!\nPress any key to continue...");
  success = 0;
  getch();
}
else
{
  // Clear array
  for(P = 1 ; P < MAX_BUS ; P++)
  { for(Q = 1 ; Q < MAX_BUS ; Q++)
    { Img_Ypq2[P][Q] = 0;
      Real_Zpq[P][Q] = 0;
      Img_Zpq[P][Q] = 0;
    }
  }

  // read data
  fread(&Dat,sizeof(struct Data),1,fp);
  // display data
  clrscr();
  MAX_BUS = Dat.Bus_num+1;
  printf("Number of bus : %d\n",MAX_BUS-1);

  for(P= 1; P < MAX_BUS ;P++)
  { // start for1 loop
    for(Q=1 ;Q < MAX_BUS ;Q++)
    { // start for2 loop
      Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q];
      Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q];
      Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];
      Relat[P][Q] = Dat.Relat[P][Q];
      Relat[Q][P] = Dat.Relat[Q][P];
      if(P!=Q)
      { printf("Impedance and line charging between bus %d and %d\n",P,Q);
        printf(" R   : %f\n",Real_Zpq[P][Q]);
        printf(" X   : %f\n",Img_Zpq[P][Q]);
        printf(" L chrg : %f\n",Img_Ypq2[P][Q]);
        getch();
      }
    } // end for2 loop
  } // end for1 loop

  pv=0;
  m = 0;
  n = 1;
  for(d=1;d<MAX_BUS;d++)
  {
    switch(Dat.t[d]) {
      case 1 : n--;
                slack_bus = d;
                break;
      case 2 : m++;
                v_control[m] = d;
                pv = m ;
                break;
      case 3 : break;
    }
  }
  MAGNI_V[d] = Dat.MAGNI_V[d];
  ANGLE_V[d] = Dat.ANGLE_V[d];
}

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    G_P[d] = Dat.G_P[d];
    G_Q[d] = Dat.G_Q[d];
    L_P[d] = Dat.L_P[d];
    L_Q[d] = Dat.L_Q[d];
    printf("\nMagnitude and angle of Bus no. %d \n",d);
    printf("  Magnitude : %f\n",MAGNI_V[d]);
    printf("  Angle   : %f\n",ANGLE_V[d]);
    printf(" Mw of Gen. of Bus no. %d : %f\n",d,G_P[d]);
    printf(" Mvar of Gen. of Bus no. %d : %f\n",d,G_Q[d]);
    printf(" Mw of Load of Bus no. %d : %f\n",d,L_P[d]);
    printf(" Mvar of Load of Bus no. %d : %f\n",d,L_Q[d]);
    Relat_G[d] = Dat.Relat_G[d];
    Relat_L[d] = Dat.Relat_L[d];
    O[d-1] = Dat.O[d-1];
    BUBB[d-1] = Dat.BUBB[d-1];
    getch();
}
BASE = Dat.BASE;
BASE_V = Dat.BASE_V;
printf("\nBase of Power   : %f\n",BASE);
printf("Base of Voltage : %f\n",BASE_V);

getch();
fclose(fp);
}
return(success);
}

```

```

int input_new_filedata_RX(void)
{
    char ch;
    int N,n,change,k,i,j,num,check_bubb,P,Q;
    int B_MAX,MAX; //int M=0;
    int G_BUBB[5];
    int success=1;
    int order[5];

    printf("\nEnter file name: ");
    getstring();
    if((fp=fopen(filename,"r"))!=NULL)
    {
        printf("\nWARNING !!\n");
        printf("  Overwrite exist file ? (Y/N) ");
        ch=getch();
        if ((ch=='N')||(ch=='n'))
        { success = 0;
          goto out_RX;
        }
    }
    fclose(fp);
    if((fp=fopen(filename,"w"))==NULL)
    { // start if3
      success = 0;
      printf("\nError !!!\007\nPress any to continue...");
      getch();
    } // end if3
    else
    { // start else3

```

```

printf("\nEnter number of Bus :");
scanf("%d",&MAX_BUS);
Dat.Bus_num = MAX_BUS;
MAX_BUS = MAX_BUS+1;
for(P = 1 ; P < MAX_BUS ; P++)
{
    for(Q = 1 ; Q < MAX_BUS ; Q++)
    {
        Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0 ; /* Clear all array */
        Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q] = 0 ; /* for next loop */
        Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0 ;
    }
}
for (num=1;num<MAX_BUS:num++)
    BUBB[num] = 0; //protect hang
for(P = 1; P < MAX_BUS :P++)
{ // start for1 loop
    for(Q =1; Q < MAX_BUS :Q++)
    { // start for2 loop
        if(P<Q)
        { // start if4
            printf("Is there Impedance and line charging between bus %d and %d(Y or N) : \
n",P,Q);
            ch = getch();
            ch = toupper(ch);
            if(ch == 'Y')
            {
                Dat.Relat[P][Q] = Relat[P][Q] = 'Y'; //check relation between bus
                // printf("\n");
                printf(" R : ");
                scanf("%le",&Dat.Real_Zpq[P][Q]);
                Real_Zpq[P][Q] = Dat.Real_Zpq[P][Q];
                printf(" X : ");
                scanf("%le",&Dat.Img_Zpq[P][Q]);
                Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q];
                printf(" L chrg : ");
                scanf("%le",&Dat.Img_Ypq2[P][Q]);
                Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q];

                switch(P)
                {
                    case 1 : BUBB[0]++;break;
                    case 2 : BUBB[1]++;break;
                    case 3 : BUBB[2]++;break;
                    case 4 : BUBB[3]++;break;
                    case 5 : BUBB[4]++;break;
                }
                switch(Q)
                {
                    case 1 : BUBB[0]++;break;
                    case 2 : BUBB[1]++;break;
                    case 3 : BUBB[2]++;break;
                    case 4 : BUBB[3]++;break;
                    case 5 : BUBB[4]++;break;
                }
            }
        }
    }
}
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Img_Zpq[P][Q] = Dat.Img_Zpq[P][Q] = 0;
    Img_Ypq2[P][Q] = Dat.Img_Ypq2[P][Q] = 0;
}
Dat.Real_Zpq[Q][P] = Dat.Real_Zpq[P][Q];
Dat.Img_Zpq[Q][P] = Dat.Img_Zpq[P][Q];
Dat.Img_Ypq2[Q][P] = Dat.Img_Ypq2[P][Q];
Real_Zpq[Q][P] = Real_Zpq[P][Q];
Img_Zpq[Q][P] = Img_Zpq[P][Q];
Img_Ypq2[Q][P] = Img_Ypq2[P][Q];
Dat.Relat[Q][P] = Relat[Q][P] = Relat[P][Q];

} // end if4
else
{ // start else4
    if(P==Q)
    { Real_Zpq[P][P] = Dat.Real_Zpq[P][P] =0;
      Img_Zpq[P][P] = Dat.Img_Zpq[P][P] =0;
      Img_Ypq2[P][P] = Dat.Img_Ypq2[P][P] =0;
    }
} //end else4
} // end for2 loop
} // end for1 loop

for ( n=0 ; n<MAX_BUS ;n++ )
    O[n] = -1;

for(i=1 ; i< MAX_BUS ; i++)
{
    order[i-1] = 0;
    for ( j=1 ; j< MAX_BUS ; j++ )
    {
        check_bubb = 1;
        for ( n=0 ; n<MAX_BUS ; n++ )
            if ( j == O[n] )
                check_bubb = 0;

        if (( BUBB[j-1] >= order[i-1] )&&( check_bubb != 0 ))
        {
            order[i-1] = BUBB[j-1];
            change = j;
        }
    }
    O[i-1] = change;
}

for(i=0;i<MAX_BUS-1;i++)
{ Dat.BUBB[i] = BUBB[i];
  Dat.O[i] = O[i];
  printf("%d\n", O[i]);
}

getch();
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} // end else3
out_RX : return(success);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Make_RX_to_GB(void)
```

```
{  
  double DIVIDE , DATA_1 , DATA_2 , DATA_3;  
  double Img_Yp[SP_MAX_BUS];  
  double G[SP_MAX_BUS][SP_MAX_BUS];  
  int P, Q, COUNT;  
  char ch ;  
  
  for(P = 1 ; P < MAX_BUS ; P++)  
  {  
    for(Q = 1 ; Q < MAX_BUS ; Q++)  
    {  
      DIVIDE = Real_Zpq[P][Q]*Real_Zpq[P][Q] ;  
      DIVIDE = DIVIDE + Img_Zpq[P][Q]*Img_Zpq[P][Q] ;  
      if ( DIVIDE != 0 )  
      {  
        Real_Ypq[P][Q] = Real_Zpq[P][Q] / DIVIDE ;  
        Img_Ypq[P][Q] = -1*(Img_Zpq[P][Q]/DIVIDE) ;  
        /* -1 Because conjugate */  
      }  
      else  
      {  
        Real_Ypq[P][Q] = 0 ;  
        Img_Ypq[P][Q] = 0 ;  
      }  
    }  
  }  
  /*----- Cal Yp */  
  for(P = 1 ; P < MAX_BUS ; P++)  
    Img_Yp[P]=0 ; /*Protecthang*/  
  for(P = 1 ; P < MAX_BUS ; P++)  
    for(Q = 1 ; Q < MAX_BUS ; Q++)  
      Img_Yp[P] = Img_Yp[P] + Img_Ypq2[P][Q] ;  
  /*----- Cal G_B */  
  for(P = 1 ; P < MAX_BUS ; P++)  
  {  
    for(Q = 1 ; Q < MAX_BUS ; Q++)  
    {  
      G[P][Q] = 0 ;  
      B[P][Q] = 0 ;  
    } /* Protect hang on next loop */  
  }  
  
  for(P = 1 ; P < MAX_BUS ; P++)  
    B[P][P] = B[P][P] + Img_Yp[P] ;  
    /* Self admittances must include Yp */  
  
  for(P = 1 ; P < MAX_BUS ; P++)  
  {  
    for(Q = 1 ; Q < MAX_BUS ; Q++)  
    {  
      if ( P == Q )  
      {  
        for(COUNT = 1 ; COUNT < MAX_BUS ; COUNT++)  
        {  
          G[P][P] = G[P][P] + Real_Ypq[P][COUNT];  
          B[P][P] = B[P][P] + Img_Ypq[P][COUNT] ;  
        }  
      }  
      else  
      {  
        G[P][Q] = -1*(Real_Ypq[P][Q]) ;  
        B[P][Q] = -1*(Img_Ypq[P][Q]) ;  
      } /* Mutual admittances value are negative */  
    }  
  }  
  }  
  /*----- Cal Polar */  
  for(P = 1 ; P < MAX_BUS ; P++)  
  {  
    for(Q = 1 ; Q < MAX_BUS ; Q++)  
    {  
      DATA_1 = B[P][Q]*B[P][Q] + G[P][Q]*G[P][Q];
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAGNI_Ybus[P][Q] = sqrt(DATA_1);
if( G[P][Q] != 0 ) DATA_1 = B[P][Q] / G[P][Q];
else DATA_1 = 0;
DATA_1 = atan(DATA_1); /* Value are in radian */
DATA_1 = ( DATA_1 * 180 ) / pi;
if( G[P][Q] < 0 ) DATA_1 = DATA_1 + 180;
ANGLE_Ybus[P][Q] = DATA_1;
}
}
}

```

//----- End of Make\_RX\_to\_GB -----

```

int input_new_filedata_VPQ(void)
{
    int k,d,m,n,t[SP_MAX_BUS];
    double TEMP_DATA;
    int success=1;

    clrscr();
    pv = 0;
    m = 0;
    n = 1;
    if((fp=fopen(filename,"w"))==NULL)
    { // start if1
        success = 0;
        printf("\nError !!!\nPress any to continue...");
        getch();
    } // end if1
    else
    { // start,else1
        printf("%52s\n","<< TYPE OF BUS SELECTION >>");
        printf("%44s\n","1. SLACK BUS");
        printf("%57s\n","2. VOLTAGE CONTROLLED BUS");
        printf("%43s\n","3. LOAD BUS");
        printf("%45s\n","( SELECT 1-3 )");
        for (d=1; d < MAX_BUS ;d++)
        { printf(" TYPE OF BUS [%d] SELECT ==>> ".d);
          scanf("%d",&t[d]);
          Dat.t[d] = t[d];
        }
        clrscr();
        for (d=1; d < MAX_BUS ;d++)
        {
            switch(t[d]) {
                case 1 : n--;
                    slack_bus = d;
                    printf(" BUS [%d] SLACK BUS >> ".d);
                    break;
                case 2 : m++;
                    v_control[m] = d;
                    pv = m;
                    printf(" BUS [%d] PV BUS >> ".d);
                    break;
                case 3 : printf(" BUS [%d] LOAD BUS >> ".d);
                    break;
                default: printf("%35s","<<< DATA ERROR ! >>>");
                    success = 0;
                    goto out_VPQ;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(n<0)
{
printf(" <<< DATA ERROR ! >>>\007\n");
printf(" DUPLICATE SLACK BUS !!\n");
printf(" Press any key to continue...");
success = 0;
getch();
goto out_VPQ;
}
printf("%18s", "MAGNITUDE : ");
scanf("%le", &MAGNI_V[d]);
Dat.MAGNI_V[d] = MAGNI_V[d];
printf("%10s", "ANGLE : ");
scanf("%le", &ANGLE_V[d]);
Dat.ANGLE_V[d] = ANGLE_V[d];

if( d != slack_bus )
{ /*
if( d == v_control[m] )
{ /*
printf("%40s", "MW OF GEN : ");
scanf("%le", &G_P[d]);
if(G_P[d] != 0)
Dat.Relat_G[d] = Relat_G[d] = 'Y'; // Is there Generator?
Dat.G_P[d] = G_P[d];
printf("%40s", "MVAR OF GEN : ");
scanf("%le", &G_Q[d]);
Dat.G_Q[d] = G_Q[d];
/* }
else
{
G_P[d] = 0;
G_Q[d] = 0;
}*/
printf("%40s", "MW OF LOAD : ");
scanf("%le", &L_P[d]);
if(L_P[d] != 0)
Dat.Relat_L[d] = Relat_L[d] = 'Y'; //Is there Load?
Dat.L_P[d] = L_P[d];
printf("%40s", "MVAR OF LOAD : ");
scanf("%le", &L_Q[d]);
Dat.L_Q[d] = L_Q[d];
}
else
{
Dat.G_P[slack_bus] = G_P[slack_bus] = 0;
Dat.G_Q[slack_bus] = G_Q[slack_bus] = 0;
Dat.L_P[slack_bus] = L_P[slack_bus] = 0;
Dat.L_Q[slack_bus] = L_Q[slack_bus] = 0;
Dat.Relat_G[slack_bus] = 'Y';
}
}
printf("Enter BASE of POWER : ");
scanf("%le", &BASE);
Dat.BASE = BASE;
printf("Enter BASE of VOLTAGE : ");
scanf("%le", &BASE_V);

```

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dat.BASE_V = BASE_V;
fwrite(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
} //end else1
out_VPQ: return(success);
}

```

```

void JACOB(void)

```

```

{
int d,i,m,check_pv;

DIMEN_P=0;
Y = 1;

for(d=1 ; d < MAX_BUS : d++)
{
    Psch[d] = G_P[d] - L_P[d] ;
    Psch[d] = Psch[d] / BASE;
    Qsch[d] = G_Q[d] - L_Q[d];
    Qsch[d] = Qsch[d]/BASE;
}

for (i=1 ; i<MAX_BUS ; i++)
{
    if (i != slack_bus)
    {
        DIMEN_P++;
        for ( B_X=1, X=1 ; B_X < MAX_BUS ; B_X++,X++)
        {
            if ( B_X != slack_bus)
            {
                DAT = B[i][B_X];
                JCBN_P[Y][X] = -1 * DAT;
            }
            else
                X--;
        }
        Y++;
    }
}
DIMEN_P++;

```

```

/*-----JCBN_Q--*/
DIMEN_Q = 0;
for(i = 1, Y = 1 ; i < MAX_BUS ; i ++, Y++)
{
    check_pv = 1 ;
    for(m=1 ; m < (pv+1) ; m++)
        if(v_control[m] == i)
            check_pv = 0 ;
    if ((i != slack_bus)&&(check_pv != 0))
    {
        DIMEN_Q++;
        for(B_X = 1, X = 1 ; B_X < MAX_BUS ; B_X++,X++)
        {
            check_pv = 1 ;
            for(m=1 ; m < (pv+1) ; m++)
                if(v_control[m] == B_X)
                    check_pv = 0 ;
            if ((B_X != slack_bus)&&(check_pv != 0 ))
            {
                DAT = B[i][B_X];
                JCBN_Q[Y][X] = -1 * DAT ;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        X--;
    }
}
else
    Y--;
}
DIMEN_Q++;
}
/*-----END OF JACOB-----*/

```

```
void Make_Delta_PQ(void)
```

```

{
    double DATA_P, DATA_Q, DATA_1, DATA_2, DATA_3;
    int I, K;
}
/*-----*/

```

```

for(I = 1; I < MAX_BUS; I++) /*--- 1 Slack bus ---*/
if( I != slack_bus )
{ DATA_P = 0; DATA_Q = 0;
  for(K = 1; K < MAX_BUS; K++)
  { DATA_1 = MAGNI_V[I] * MAGNI_V[K] * MAGNI_Ybus[I][K];
    DATA_2 = ANGLE_V[I] - ANGLE_V[K] - ANGLE_Ybus[I][K];
    DATA_2 = ( DATA_2 * pi ) / 180;

    DATA_3 = cos( DATA_2 );
    DATA_P = DATA_P + ( DATA_1 * DATA_3 );

    DATA_3 = sin( DATA_2 );
    DATA_Q = DATA_Q + ( DATA_1 * DATA_3 );
  }
  DELTA_P[I] = Psch[I] - DATA_P;
  DEL_PbyV[I] = DELTA_P[I] / MAGNI_V[I];

  DELTA_Q[I] = Qsch[I] - DATA_Q;
  DEL_QbyV[I] = DELTA_Q[I] / MAGNI_V[I];
}
}
/*----- End of Make_Delta_PQ */

```

```
void Sol_Matrx()
```

```

{
    int i, m, check_pv;
    int BUS;
    double MULTI.ANS[SP_MAX_BUS];
}
/*-----JCBN <= DelPbyV */

for( Y=1, i=1; Y < DIMEN_P; Y++, i++)
{ if( i != slack_bus )
    JCBN_P[Y][DIMEN_P] = DEL_PbyV[i];
else
    Y = Y-1;
}

```

```
/*-----Cal Del_AngleV */
```

```

for(Y = 1; Y < DIMEN_P; Y++)
{ for(X = 1; X < DIMEN_P; X++)
  { if( X != Y )

```

การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่มีการเห็นแก่ตัวทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { MULTI= JCBN_P[X][Y] / JCBN_P[Y][Y];
      for(Z = 1 ; Z < (DIMEN_P+1);Z++)
        JCBN_P[X][Z] = JCBN_P[X][Z]-(MULTI * JCBN_P[Y][Z]);
    }
  }
}
for (Y = 1 ;Y < DIMEN_P; Y++)
  ANS[Y] = JCBN_P[Y][DIMEN_P] / JCBN_P[Y][Y];

/*-----Angle_V + Del_AngleV */
for(Y = 1, Z = 1 ;Y < MAX_BUS; Y++,Z++)
{ if ( Y != slack_bus)
  {
    DELTA_ANG[Y] = ANS[Z];
    ANGLE_V[Y] = ANGLE_V[Y] + DELTA_ANG[Y];
  }
  else
    Z--;
}

/*-----JCBN <= DelQbyV */

for (Y = 1,i = 1 ; i < MAX_BUS ; Y++, i ++ )
{ check_pv = 1 ;
  for(m=1 ; m < (pv+1) ; m++)
    if(v_control[m] == i )
      check_pv = 0;
  if (( i !=slack_bus)&&(check_pv != 0))
    JCBN_Q[Y][DIMEN_Q] = DEL_QbyV[i];
  else
    Y--;
}

/*-----Cal Del_MagniV */
for(Y=1 ;Y < DIMEN_Q; Y++)
{ for(X = 1 ; X < DIMEN_Q ; X++)
  { if (X != Y)
    { MULTI = JCBN_Q[X][Y] / JCBN_Q[Y][Y];
      for(Z = 1; Z < DIMEN_Q+1;Z++)
        JCBN_Q[X][Z] = JCBN_Q[X][Z]-(MULTI * JCBN_Q[Y][Z]);
    }
  }
}

for(Y = 1 ; Y < DIMEN_Q;Y++)
  ANS[Y] = JCBN_Q[Y][DIMEN_Q] / JCBN_Q[Y][Y] ;

/* -----Magni_V+ Del_MagniV */
for(Y = 1 , Z = 1 ;Y < MAX_BUS ; Y++,Z++)
{ check_pv = 1;
  for(m=1 ; m < (pv+1) ; m++)
    if(v_control[m] == Y )
      check_pv = 0 ;
  if((Y != slack_bus)&&(check_pv != 0))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 DELTA\_VOL[Y] = ANS[Z];  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MAGNI_V[Y] = MAGNI_V[Y] + DELTA_VOL[Y];
}
else
    Z--;
}
}
/* -----END OF SOL_Matrix-----*/

void Cal_Error()
{
    int k;
    double SQ_V, SQ_A;

    ERROR=0;
    for(k = 1 ; k < MAX_BUS ; k++ )
        if(k!= slack_bus)
        {
            SQ_V = DELTA_VOL[k] * DELTA_VOL[k];
            SQ_V = sqrt(SQ_V);
            SQ_A = DELTA_ANG[k] * DELTA_ANG[k];
            SQ_A = sqrt(SQ_A);

            if((SQ_V>ERR)||(SQ_A>ERR))
                ERROR=1;
        }
}
/* ----- End of Cal_Error */

void Cal_PQ(void)
    /* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
{
    int p, q, k, Y, d, check_pv;
    double c[SP_MAX_BUS], f[SP_MAX_BUS];
    double REAL_1, IMG_1, TEMP_DATA;

    /*----- Cal c, f */
    for(p = 1 ; p < MAX_BUS ; p++)
    {
        TEMP_DATA = (ANGLE_V[p] * pi) / 180 ;
        c[p] = MAGNI_V[p] * cos(TEMP_DATA);
        f[p] = MAGNI_V[p] * sin(TEMP_DATA);
    }

    /*----- Cal_PQ */
    for(p = 1 ; p < MAX_BUS ; p++)
    {
        for(q = 1 ; q < MAX_BUS ; q++)
        {
            if(q != p)
            {
                REAL_1 = c[p] * ( c[p] - c[q] ) + f[p] * ( f[p] - f[q] );
                IMG_1 = c[p] * ( f[p] - f[q] ) - f[p] * ( c[p] - c[q] );
                /* Ep^(Ep-Eq) = REAL_1 + jIMG_1 */

                TEMP_DATA = REAL_1 * Real_Ypq[p][q];
                FLOW_P[p][q] = TEMP_DATA - ( IMG_1 * Img_Ypq[p][q] );
                FLOW_P[p][q] = BASE * FLOW_P[p][q];

                TEMP_DATA = REAL_1 * Img_Ypq[p][q];
                TEMP_DATA = TEMP_DATA + ( IMG_1 * Real_Ypq[p][q] );
            }
        }
    }
}

```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุณาให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        FLOW_Q[p][q] = -1 * TEMP_DATA ;
        TEMP_DATA = Img_Ypq2[p][q] * (e[p]*e[p] + f[p]*f[p]) ;
        FLOW_Q[p][q] = FLOW_Q[p][q] - TEMP_DATA ;
        FLOW_Q[p][q] = BASE * FLOW_Q[p][q] ;
    }
}

/*----- Cal G_PQ SLACK BUS */
for(k = 1, REAL_1 = 0, IMG_1 = 0; k < MAX_BUS; k++)
{ if( k != slack_bus )
  { REAL_1 = REAL_1 + FLOW_P[slack_bus][k] ;
    G_P[slack_bus] = REAL_1 ;
    IMG_1 = IMG_1 + FLOW_Q[slack_bus][k] ;
    G_Q[slack_bus] = IMG_1 ;
  }
}
G_P[p] = G_P[p] + L_P[p] ;
G_Q[q] = G_Q[q] + L_Q[q] ;

/*----- Cal G_Q PV BUS */
for(k = 1, TEMP_DATA = 0; k < MAX_BUS; k++)
{
  check_pv = 1 ;
  for(d = 1; d < (pv+1); d++)
    if( v_control[d] == k )
      check_pv = 0;

  if( check_pv == 0 )
  { for(q = 1; q < MAX_BUS; q++)
    { if(q != k)
      TEMP_DATA = TEMP_DATA + FLOW_Q[k][q] ;
    }
    G_Q[k] = TEMP_DATA + L_Q[k] ;
  }
}
}
/*----- End of Cal_PQ -----*/

```

```
void Make_Table(void)
```

```

{
  int p, q, k, BUS;
  int y_pos;          /* position in y-axis */

  printf("\nBUS_____VOLTAGE_MAGNITUDE_____PHASE_ANGLE\n\n");
  for(BUS = 1; BUS < MAX_BUS; BUS++)
  { printf("[%d].BUS) :
    y_pos = wherex() ;
    gotoxy( 15,y_pos) : printf("%f".MAGNI_V[BUS]) ;
    gotoxy(36,y_pos) : printf("%9.6f".ANGLE_V[BUS]) ;
    printf("\n") ;
  }
  getch();
  clrscr();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\n");
printf("BUS___G[MW]___G[MVAR]___L[MW___VAR]___");
printf("  TòBus___MW___MVAR___\n\n");
for(p = 1 ; p < MAX_BUS ; p++)
{ printf("[%d]" ,p) ; y_pos = wherey() ;
gotoxy( 7,y_pos) ; printf("%9.6f",G_P[p]) ;
gotoxy(20,y_pos) ; printf("%9.6f",G_Q[p]) ;
gotoxy(34,y_pos) ; printf("%4.1f",L_P[p]) ;
gotoxy(41,y_pos) ; printf("%4.1f",L_Q[p]) ;
for(q = 1 ; q < MAX_BUS ; q++)
{ if(q != p)
{ gotoxy(49,y_pos) ; printf("%d",q) ;
gotoxy(53,y_pos) ; printf("%11.7f",FLOW_P[p][q]) ;
gotoxy(67,y_pos) ; printf("%11.7f",FLOW_Q[p][q]) ;
y_pos++ ;
}
}
printf("\n") ;
}
printf("\n") ;
if(( 25-y_pos ) < (MAX_BUS-2))
{
getch();
clrscr();
}
}
}
/*----- End of Make_Table -----*/

```

```

//Begin set graphics
void Setgraph(void)
{
int i;
int driver=DETECT,mode=VGAHI;

initgraph(&driver,&mode,"");

setbkcolor(1);
setcolor(15);

for(i=0;i<MAX_BUS-1; i++)
{
switch(i)
{
case 0: n1 = O[i];
Line_main1();
break;
case 1: n2 = O[i];
Line_main2();
break;
case 2: n3 = O[i];
Line_main3();
break;
case 3: n4 = O[i];
Line_main4();
break;
case 4: n5 = O[i];
Line_main5();
break;
}
}
}

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

getch();
closegraph();
}
/* Begin Draw Line Between Bus */

void Line_main1(void)
{
    int a,b;

    // Setup Color
    // 1st circle
    settextstyle( SANS_SERIF_FONT , HORIZ_DIR,2);
    outtextxy(40,3 , "    ** THREE PHASE LOAD FLOW ** ");
    gotoxy(SEX-3,6);
    printf("%d",O[0]);
    if (Relat_G[n1]=='Y')
    { setlinestyle(SOLID_LINE,1,1);
      circle(x[1],y[1],r);
      setlinestyle(SOLID_LINE,1,3);
      line(x[1],y[1]+r,x[1],y[1]+r+10); //1.2
    }

    // Lines over/under circle
    setlinestyle(SOLID_LINE,1,3);
    // line(x[1],y[1]+r,x[1],y[1]+r+10); //2
    // line(x[1],y[1]+r+10,x[1],y[1]+r+20); //5
    line(x[1]-span,y[1]+r+10,x[1]+span,y[1]+r+10);
    line(x[1]-(span*2/3),y[1]+r,x[1]-(span*2/3),y[1]+r+10); //1

    setlinestyle(SOLID_LINE,1,2);
    a = x[1]-(span*2/3);
    b = y[1]+r;
    line(a-3,b,a+3,b);
    line(a+3,b,a,b-9);
    line(a-3,b,a,b-9);

    // line(x[1]-(span*2/3),y[1]+r+10,x[1]-(span*2/3),y[1]+r+20); //4
    // line(x[1]+(span*2/3),y[1]+r,x[1]+(span*2/3),y[1]+r+10); //3
    // line(x[1]+(span*2/3),y[1]+r+10,x[1]+(span*2/3),y[1]+r+20); //6
}

```

```

void Line_main2(void)

```

```

{
    int a,b;

```

```

    //2nd circle

```

```

    gotoxy(SEX+35,6);

```

```

    printf("%d",O[1]);

```

```

    if(Relat_G[n2] == 'Y')

```

```

        setlinestyle(SOLID_LINE,1,1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

circle(x[2],y[2],r);
setlinestyle(SOLID_LINE,1,3);
line(x[2],y[2]+r,x[2],y[2]+r+10);           //2.2

```

```

}

```

```

// Lines over/under circle
setlinestyle(SOLID_LINE,1,3);
// line(x[2],y[2]+r,x[2],y[2]+r+10);           //2
// line(x[2],y[2]+r+10,x[2],y[2]+r+20);       //5
line(x[2]-span,y[2]+r+10,x[2]+span,y[2]+r+10);
// line(x[2]-(span*2/3),y[2]+r,x[2]-(span*2/3),y[2]+r+10); //1
// line(x[2]-(span*2/3),y[2]+r+10,x[2]-(span*2/3),y[2]+r+20); //4
line(x[2]+(span*2/3),y[2]+r,x[2]+(span*2/3),y[2]+r+10); //3.
// line(x[2]+(span*2/3),y[2]+r+10,x[2]+(span*2/3),y[2]+r+20); //6

```

```

setlinestyle(SOLID_LINE,1,2);
a = x[2]+(span*2/3);
b = y[2]+r;
line(a-3,b,a+3,b);
line(a+3,b,a,b-9);
line(a-3,b,a,b-9);
setlinestyle(SOLID_LINE,1,3);

```

```

//if( n1 < n2 )
if (Relat[n1][n2] == 'Y')
{
gotoxy(3,position);
printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n1,n2,Real_Zpq[n1][n2],Img_Zpq[n1][n2] );
if( Img_Ypq2[n1][n2] != 0 )
printf(" Y2[%d][%d]= %2.2fj P.U",n1,n2,Img_Ypq2[n1][n2]);

line(x[1]+(span*2/3),y[1]+r,x[2]-(span*2/3),y[2]+r); //L12
line(x[1]+(span*2/3),y[1]+r,x[1]+(span*2/3),y[1]+r+10); //1.3
line(x[2]-(span*2/3),y[2]+r,x[2]-(span*2/3),y[2]+r+10); //2.1
}
}

```

```

void Line_main3(void)

```

```

{
int a,b;

// 4th circle

gotoxy(SEX+19,10);
printf("%d".O[3]);

if(Relat_G[n3] == 'Y')
{
setlinestyle(SOLID_LINE,1,1);
circle(x[4]+(span*2/3),y[4],r);
setlinestyle(SOLID_LINE,1,3);
line(x[4]+(span*2/3),y[4]-r,x[4]+(span*2/3),y[4]-r-10); //3.5
// line(x[4],y[4]-r-10,x[4],y[4]-r-20); //3.5
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Lines over/under circle
setlinestyle(SOLID_LINE,1,3);
// line(x[4],y[4]-r,x[4],y[4]-r-10); //2
// line(x[4],y[4]-r-10,x[4],y[4]-r-20); //5
line(x[4]-span,y[4]-r-10,x[4]+span,y[4]-r-10);
line(x[4]-(span*2/3),y[4]-r,x[4]-(span*2/3),y[4]-r-10); //4
// line(x[4]-(span*2/3),y[4]-r-10,x[4]-(span*2/3),y[4]-r-20); //1
// line(x[4]+(span*2/3),y[4]-r,x[4]+(span*2/3),y[4]-r-10); //3
// line(x[4]+(span*2/3),y[4]-r-10,x[4]+(span*2/3),y[4]-r-20); //6

```

```

setlinestyle(SOLID_LINE,1,2);
a = x[4]-(span*2/3);
b = y[4]-r;
line(a-3,b,a+3,b);
line(a+3,b,a,b+9);
line(a-3,b,a,b+9);
setlinestyle(SOLID_LINE,1,3);

```

```

//if( n1 < n3)
if (Relat[n1][n3] == 'Y')
{
gotoxy(3,position+1);
printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n1,n3,Real_Zpq[n1][n3],Img_Zpq[n1][n3] );
if (Img_Ypq2[n1][n3] != 0 )
printf(" Y/2[%d][%d]= %2.2fj P.U",n1,n3,Img_Ypq2[n1][n3]);

line(x[1],y[1]+r+20,x[4]-(span*2/3),y[4]-r-20); //L13
line(x[4]-(span*2/3),y[4]-r-10,x[4]-(span*2/3),y[4]-r-20); //3.1
line(x[1],y[1]+r+10,x[1],y[1]+r+20); //1.5
}

```

```

//if( n2 < n3)
if (Relat[n2][n3] == 'Y')
{
gotoxy(3,position+2);
printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n2,n3,Real_Zpq[n2][n3],Img_Zpq[n2][n3] );
if (Img_Ypq2[n2][n3] != 0 )
printf(" Y/2[%d][%d]= %2.2fj P.U",n2,n3,Img_Ypq2[n2][n3]);

line(x[2],y[2]+r+20,x[4]+(span*2/3),y[4]-r-20); //L23
line(x[4]+(span*2/3),y[4]-r-10,x[4]+(span*2/3),y[4]-r-20); //6
//line(x[4]+(span*2/3),y[4]-r,x[4]+(span*2/3),y[4]-r-10); //3.3
line(x[2],y[2]+r+10,x[2],y[2]+r+20); //2.5
}
}

```

```

void Line_main4(void)
{

```

```

int a,b;

```

```

// 3rd circle

```

```

gotoxy(SEX+28,19);
printf("%d".O[2]);

```

```

if(Relat_G[n4] == 'Y')

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

circle(x[3],y[3],r);
setlinestyle(SOLID_LINE,1,3);
line(x[3],y[3]+r,x[3],y[3]+r+10); //4.2
}

setlinestyle(SOLID_LINE,1,3);
// line(x[3],y[3]+r,x[3],y[3]+r+10); //2
// line(x[3],y[3]+r+10,x[3],y[3]+r+20); //5
line(x[3]-span,y[3]+r+10,x[3]+span,y[3]+r+10);
// line(x[3]-(span*2/3),y[3]+r,x[3]-(span*2/3),y[3]+r+10); //1
// line(x[3]+(span*2/3),y[3]+r,x[3]+(span*2/3),y[3]+r+10); //3
// line(x[3]+(span*2/3),y[3]+r+10,x[3]+(span*2/3),y[3]+r+20); //6
line(x[3]-(span*2/3),y[3]+r+10,x[3]-(span*2/3),y[3]+r+20); //4

setlinestyle(SOLID_LINE,1,2);
a = x[3]-(span*2/3) ;
b = y[3]+r+20 ;
line(a-3,b,a+3;b);
line(a+3,b,a,b+9);
line(a-3,b,a,b+9);
setlinestyle(SOLID_LINE,1,3);

//if( n1 < n4 )
if (Relat[n1][n4] == 'Y')
{
gotoxy(3,position+3);
printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n1,n4,Real_Zpq[n1][n4],Img_Zpq[n1][n4] );
if( Img_Ypq2[n1][n2] != 0 )
printf(" Y/2[%d][%d]= %2.2fj P.U",n1,n4,Img_Ypq2[n1][n4]);

line(x[1]+(span*2/3),y[1]+r+20,x[3]-(span*2/3),y[3]+r); //L14
line(x[1]+(span*2/3),y[1]+r+10,x[1]+(span*2/3),y[1]+r+20); //1.6
line(x[3]-(span*2/3),y[3]+r,x[3]-(span*2/3),y[3]+r+10); //4.1
}

//if( n2 < n4 )
if (Relat[n2][n4] == 'Y')
{
gotoxy(3,position+4);
printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n2,n4,Real_Zpq[n2][n4],Img_Zpq[n2][n4] );
if( Img_Ypq2[n1][n2] != 0 )
printf(" Y/2[%d][%d]= %2.2fj P.U",n2,n4,Img_Ypq2[n2][n4]);

line(x[2]-(span*2/3),y[2]+r+20,x[3]+(span*2/3),y[3]+r); //L24
line(x[2]-(span*2/3),y[2]+r+10,x[2]-(span*2/3),y[2]+r+20); //2.4
line(x[3]+(span*2/3),y[3]+r,x[3]+(span*2/3),y[3]+r+10); //4.3
}

//if( n3 < n4 )
if (Relat[n3][n4] == 'Y')
{
gotoxy(3,position+5);
printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n3,n4,Real_Zpq[n3][n4],Img_Zpq[n3][n4] );
if( Img_Ypq2[n1][n2] != 0 )
printf(" Y/2[%d][%d]= %2.2fj P.U",n3,n4,Img_Ypq2[n3][n4]);
}

```

```

line(x[3],y[3]+r+20,x[4],y[4]-r-20); //L34
line(x[4],y[4]-r-10,x[4],y[4]-r-20); //3.2
line(x[3],y[3]+r+10,x[3],y[3]+r+20); //4.5
}

```

```

}

```

```

void Line_main5(void)

```

```

{

```

```

    int a,b;

```

```

        // 5th circle

```

```

gotoxy( SEX+25,30);
printf("%d",O[4]);

```

```

if(Relat_G[n5] == 'Y')
{
    setlinestyle(SOLID_LINE,1,1);
    circle(x[5],y[5],r);
    setlinestyle(SOLID_LINE,1,3);
    line(x[5],y[5]-r,x[5],y[5]-r-10); //5.5
}

```

```

setlinestyle(SOLID_LINE,1,3);
// line(x[5],y[5]-r,x[5],y[5]-r-10); //2
// line(x[5],y[5]-r-10,x[5],y[5]-r-20); //5
line(x[5]-span,y[5]-r-10,x[5]+span,y[5]-r-10);
// line(x[5]-(span*2/3),y[5]-r-10,x[5]-(span*2/3),y[5]-r-20); //1
// line(x[5]+(span*2/3),y[5]-r,x[5]+(span*2/3),y[5]-r-10); //3
line(x[5]+(span*3/4),y[5]-r,x[5]+(span*3/4),y[5]-r-10); //6

```

```

setlinestyle(SOLID_LINE,1,2);
a = x[5]+(span*3/4) ;
b = y[5]-r ;
line(a-3,b,a+3,b);
line(a+3,b,a,b+9);
line(a-3,b,a,b+9);
setlinestyle(SOLID_LINE,1,3);

```

```

//if( n1 < n5)

```

```

if (Relat[n1][n5] == 'Y')

```

```

{

```

```

    gotoxy(3,position+6);
    printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n1,n5,Real_Zpq[n1][n5],Img_Zpq[n1][n5] );
    if( Img_Ypq2[n1][n2] != 0 )
    printf(" Y/2[%d][%d]= %2.2fj P.U",n1,n5,Img_Ypq2[n1][n5]);

```

```

    line(x[1]-(span*2/3),y[1]+r+20,x[5]-(span*3/4),y[5]-r-20); //L15
    line(x[5]-(span*3/4),y[5]-r-10,x[5]-(span*3/4),y[5]-r-20); //5.1
    line(x[1]-(span*2/3),y[1]+r+10,x[1]-(span*2/3),y[1]+r+20); //1.4
}

```

```

//if( n2'< n5)

```

if (Relat[n2][n5] == 'Y')
 หน้าที่สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  gotoxy(3,position+7);
  printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n2,n5,Real_Zpq[n2][n5],Img_Zpq[n2][n5] );
  if( Img_Ypq2[n1][n2] != 0 )
  printf(" Y/2[%d][%d]= %2.2fj P.U",n2,n5,Img_Ypq2[n2][n5]);

  line(x[2]+(span*2/3),y[2]+r+20,x[5]+(span*3/4),y[5]-r-10); //L25
  line(x[5]+(span*3/4),y[5]-r-10,x[5]+(span*3/4),y[5]-r-20); //5.3
  line(x[2]+(span*2/3),y[2]+r+10,x[2]+(span*2/3),y[2]+r+20); //2.6
}

//if( n3 < n5)
if (Relat[n3][n5] == 'Y')
{
  gotoxy(3,position+8);
  printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n3,n5,Real_Zpq[n3][n5],Img_Zpq[n3][n5] );
  if( Img_Ypq2[n1][n2] != 0 )
  printf(" Y/2[%d][%d]= %2.2fj P.U",n3,n5,Img_Ypq2[n3][n5]);

  line(x[4],y[4]-r,x[5]-(span*1/4),y[5]-r-20); //L35
  line(x[5]-(span*1/4),y[5]-r-10,x[5]-(span*1/4),y[5]-r-20); //5.2
  line(x[4],y[4]-r,x[4],y[4]-r-10); //3.6
}

if (Relat[n4][n5] == 'Y')
{
  gotoxy(3,position+9);
  printf(" Z[%d][%d] = %2.2f+%2.2fj P.U ",n4,n5,Real_Zpq[n4][n5],Img_Zpq[n4][n5] );
  if( Img_Ypq2[n1][n2] != 0 )
  printf(" Y/2[%d][%d]= %2.2fj P.U",n4,n5,Img_Ypq2[n4][n5]);

  line(x[3]+(span*2/3),y[3]+r+20,x[4]+26,y[4]-r-53); //1
  line(x[4]+35,y[4]-r-45,x[5]+(span*1/4),y[4]-28) ; //2
  line(x[5]+(span*1/4),y[4]-28,x[5]+(span*1/4),y[5]-r-20); //3
  line(x[5]+(span*1/4),y[5]-r-10,x[5]+(span*1/4),y[5]-r-20); //5.7
  line(x[3]+(span*2/3),y[3]+r+10,x[3]+(span*2/3),y[3]+r+20); //4.6
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## THREE-PHASE LOAD FLOW FAST DECOUPLE METHOD

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <ctype.h>
#include <process.h>
#include <string.h>
#include <graphics.h>
#include <graphics.h>
#include <dos.h>

#define SP_MAX_BUS 4
#define ERR 0.0000001
#define Vpv_LEVEL 0.03
#define VloadLEVEL 0.03
#define pi M_PI
#define ERROR 0.0000001
#define r 11
#define span 36
#define SEX 37

#define menu_bar_thick 22
#define first_button 200
#define button_width 35
#define button_length 550
#define buttons_gab 20
```

```
struct Data {
```

```
    int MAX_BUS;
```

```
    double Y_G[4][SP_MAX_BUS][4][SP_MAX_BUS];
```

```
    double Y_B[4][SP_MAX_BUS][4][SP_MAX_BUS];
```

```
    double MAGNI_V[4][SP_MAX_BUS];
```

```
    double ANGLE_V[4][SP_MAX_BUS];
```

```
    double G_P[4][SP_MAX_BUS], G_Q[4][SP_MAX_BUS];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double L_P[4][SP_MAX_BUS] , L_Q[4][SP_MAX_BUS];
double BASE_MVA,BASE_V,BASE ;
int t(SP_MAX_BUS);
int BUBB(SP_MAX_BUS);
int O(SP_MAX_BUS);
char Relat(SP_MAX_BUS)[SP_MAX_BUS];
char Relat_G(SP_MAX_BUS);
char Relat_L(SP_MAX_BUS);
} Dat;

```

```
//Declare Global variable
```

```

FILE *fp;
double Img_Ypq2[4][SP_MAX_BUS][4][SP_MAX_BUS];
int n1,n2,n3,n4,n5,iteration;
int position = 7;
int TRAN_X,TRAN_Y;
char convergen_p,convergen_q;
int MAX_Y ,MAX_X;
int PIK ,BAL;
double Y_G[4][SP_MAX_BUS][4][SP_MAX_BUS];
double Y_B[4][SP_MAX_BUS][4][SP_MAX_BUS];
double Y_Ggen[SP_MAX_BUS];
double Y_Bgen[SP_MAX_BUS];
double MAGNI_Y[4][SP_MAX_BUS][4][SP_MAX_BUS];
double MAGNI_Ygen[SP_MAX_BUS];
double ANG_Y[4][SP_MAX_BUS][4][SP_MAX_BUS];
double ANG_Ygen[SP_MAX_BUS];
int pv ,check_pv ,slack_bus ,v_control[SP_MAX_BUS];
int MAX_BUS;
double B[SP_MAX_BUS*3][SP_MAX_BUS*3]; //GEN BUS MUSTNT MORE THAN 3
double C[((SP_MAX_BUS-2)*3)+1][((SP_MAX_BUS-2)*3)+1];
double M[4][SP_MAX_BUS][4][SP_MAX_BUS];
double G_P[4][SP_MAX_BUS] ,G_Q[4][SP_MAX_BUS];
double L_P[4][SP_MAX_BUS] ,L_Q[4][SP_MAX_BUS];
double BASE,BASE_V;
int x[6] = {0,210,420,340,362,417};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int y[6] = {0,80,80,180,295,420};
int BUBB[5];
int O[SP_MAX_BUS];
char text_tmp[80];

int BASE_MVA ;
double Psch_gen[SP_MAX_BUS];
double GN_P[4][SP_MAX_BUS] , GN_Q[4][SP_MAX_BUS] ;
double LD_P[4][SP_MAX_BUS] , LD_Q[4][SP_MAX_BUS] ; /* 4 = Phase */
double Psch[4][SP_MAX_BUS] , Qsch[4][SP_MAX_BUS] ;
double MAGNI_V[4][SP_MAX_BUS] , ANGLE_V[4][SP_MAX_BUS] ;
double MAGNI_Vint[SP_MAX_BUS] , ANGLE_Vint[SP_MAX_BUS];
/*----- Cal_Del_PbyV */
double DEL_PbyV[4][SP_MAX_BUS], DEL_QbyV[4][SP_MAX_BUS] ;
/*----- Cal_Del_Pgen_byVint */
double DEL_Pgen_byVint[SP_MAX_BUS] ;
/*----- Cal Line Flow */
double FLOW_P[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
double FLOW_Q[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
char Relat[SP_MAX_BUS][SP_MAX_BUS] , Relat_G[SP_MAX_BUS],
      Relat_L[SP_MAX_BUS] ;
/*-----*/
double DELTA_P[4][SP_MAX_BUS] , DELTA_Q[4][SP_MAX_BUS]
      , DELTA_ANG[4][SP_MAX_BUS],DELTA_VOL[4][SP_MAX_BUS]
      ,JCBN_P[SP_MAX_BUS][SP_MAX_BUS] ,JCBN_Q[SP_MAX_BUS][SP_MAX_BUS];
double ERROR_2,ERROR_3;
int DIMEN_P , DIMEN_Q ;
char filename[80];
char number[80];
char string[200];
char text[200];

void getstring(char *str,int x1,int x2,int y);
double str_to_real(char *string);
int str_to_int(char *string);
void int_to_string(int num);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void double_to_text(double num,int point);
```

```
// Declare function
```

```
void outtextcomplex(double Re,double Im,int x ,int y);
```

```
void outtextcomplex_PQ_1(double T,double O,int c ,int d);
```

```
void outtextcomplex_PQ_2(double Y,double N,int u ,int v);
```

```
void outtextcomplex_PQ_3(double H,double K,int re ,int se);
```

```
void calculate(void);
```

```
int get_old_filedata(void);
```

```
int input_new_filedata_GB(void);
```

```
int input_new_filedata_VPQ(void);
```

```
void Clear_variable(void);
```

```
void DATA_OF_Y (void);
```

```
void Cal_Sch_PQ (void);
```

```
void Cal_Delta_PQ(void);
```

```
void Cal_Del_PbyV(void);
```

```
void Cal_Del_Pgen_byV(void);
```

```
void ERROR_P(void);
```

```
void Sol_Matrix_Delta(void);
```

```
void Cal_Del_QbyV(void);
```

```
void ERROR_Q(void);
```

```
void Sol_Matrix_V(void);
```

```
void Cal_Vint(void);
```

```
void Cal_V_slack(void);
```

```
void Cal_Flow_PQ_1(void);
```

```
void Cal_Flow_PQ_2(void);
```

```
void Cal_Flow_PQ_3(void);
```

```
void Cal_M(void);
```

```
void Cal_M_Q(void);
```

```
void Cal_Sum_Mil(void);
```

```
void Cal_Sum_Mjk(void);
```

```
void Cal_Sum2_M(void);
```

```
void Stable_of_sys(void);
```

```
void Make_Table_1(void);
```

```
void Make_Table_2(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Make_Table_3(void);
void Setgraph(void);
void Line_main1(void);
void Line_main2(void);
void Line_main3(void);
void Line_main4(void);
void Line_main5(void);
void Line_main1_1(void);
void Line_main2_1(void);
void Line_main3_1(void);
void Line_main4_1(void);
void Line_main5_1(void);
void Cal_Error_2(void);
void Cal_Error_3(void);
void Sol_Matrx_2(void);
void Sol_Matrx_3(void);
void Make_Delta_PQ_2(void);
void Make_Delta_PQ_3(void);
void JACOB(void);
void Make_MAGNI_ANG(void);
void compound(void);
void Setarrow(void);
void Setgraph_PQ(void);
void graphic_initialize(void); // Initialize graphics mode
void display(double error); // Display every values (x,y,Y/Go,Error,iteration)
void button(int x1,int y1,int x2,int y2,int thick,int push); // Draw a 3D button
void Win(int x1,int y1,int x2,int y2); // Draw a window
void BackGround(int color); // Set background color in Graphics mode
void frame(int x1,int y1,int x2,int y2); // Draw a 3D frame
void space(int x1,int y1,int x2,int y2);
void outtext3d(int x,int y,char str[],int font,int font_size,int fore_color,int bk_color);
void beep(void);
void dialog_box(int x1,int y1,int x2,int y2,int fillin,int fillcolor);
void clear_in_win(void);
void about_dialog(void);
void complete_cal_indicate(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void display_result_double(int x,int y,double num,int point,int font,int font_size,int fore_color,int bk_color);
```

```
void main(void)
```

```
{  
    int ITERATION;  
    char c=0;  
  
    while (1) // always loop  
    { // start while loop  
        // Prelude  
        graphic_initialize(); // Initialize graphics mode  
        Background(7); // Set Background color to GRAY  
        // Setup interface  
        Win(0,0,640,480); // Make main window  
        setcolor(14);  
        settextstyle(7,0,1);  
        outtextxy(230,2,"3 Phase Load Flow");  
        clear_in_win();  
        // Setup title dialog  
        dialog_box(130,130,510,350,0,0);  
        outtext3d(173,151,"3 Phase Load Flow",1,4,4,0);  
        outtext3d(172,210,"Please select one :",0,0,15,0);  
        outtext3d(220,245,"1: Read old data files",0,0,15,0);  
        outtext3d(220,260,"2: Write new data files",0,0,15,0);  
        outtext3d(220,275,"3: Exit",0,0,15,0);  
        // Make title dialog button  
        button(220,300,420,330,2,0);  
        outtext3d(300,312,"SELECT",0,0,15,0);  
        wrong_choice: // This is a GOTO label which will be used when the  
                        // user choose wrong choice  
        // Wait for keypressing  
        c=getch();  
        if((c=='1')||(c=='2')||(c=='3'))  
        {  
            // Make 1 animated button  
            button(220,300,420,330,2,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtext3d(301,313,"SELECT",0,0,15,0);
sound(1000);delay(120);nosound();
button(220,300,420,330,2,0);
outtext3d(300,312,"SELECT",0,0,15,0);
delay(200);

// Check - what key was pressed
switch(c) {
case '1':if (get_old_filedata()==1)
    {
        Setgraph();
        about_dialog();
        calculate();
        Setgraph_PQ();
    }
closegraph();
break;
case '2':if (input_new_filedata_GB()==1)
    {
        if (input_new_filedata_VPQ()==1)
        {
            Setgraph();
            about_dialog();
            calculate();
            Setgraph_PQ();
            getch();
        }
    }
closegraph();
break;
case '3':closegraph();
printf("\nBye bye\n");
exit(1);
}
}
else
{ beep();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        goto wrong_choice;
    }
} // end while loop
}

void Clear_variable(void)
{
    int i,j,m,n;

    TRAN_X = TRAN_Y = convergen_p = convergen_q = PIK = BAL = 0;
    MAX_Y = MAX_X = pv = check_pv = slack_bus = 0;

    for(i=0;i< SP_MAX_BUS;i++)
    {
        Y_Ggen[i] = MAGNI_Ygen[i] = Psch_gen[i] = DEL_Pgen_byVint[i] = 0;
        Y_Bgen[i] = ANG_Ygen[i] = v_control[i] = MAGNI_Vint[i] = 0;
        ANGLE_Vint[i] = 0;
    }
    for(i=0;i<SP_MAX_BUS;i++)
    for(j=0;j<SP_MAX_BUS;j++)
    {
        B[i][j] = C[i][j] = 0;
        GN_P[i][j] = GN_Q[i][j] = 0;
        LD_P[i][j] = LD_Q[i][j] = 0;
        Psch[i][j] = Qsch[i][j] = 0;
        DEL_PbyV[i][j] = DEL_QbyV[i][j] = 0;
    }

    for(i=0;i<4;i++)
    for(j=0;j<SP_MAX_BUS;j++)
    for(m=0;m<4;m++)
    for(n=0;n<SP_MAX_BUS;n++)
    {
        Img_Ypq2[i][j][m][n] = MAGNI_Y[i][j][m][n] = 0;
        ANG_Y[i][j][m][n] = M[i][j][m][n] = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FLOW_P[i][j][m][n] = FLOW_Q[i][j][m][n] = 0;
```

```
}
```

```
}
```

```
void calculate(void)
```

```
{
```

```
int KP,KQ;
```

```
int i;
```

```
setfillstyle(1,3);
```

```
bar(10,40,630,321);
```

```
frame(10,40,629,320);
```

```
outtext3d(220,160,"Calculating ...",1.4,14.6);
```

```
for(i=9000;i>=10;i--)
```

```
{
```

```
sound(i);
```

```
sound(i+1000);
```

```
sound(i+2000);
```

```
}
```

```
nosound();
```

```
DATA_OF_Y();
```

```
Cal_M();
```

```
Cal_M_Q();
```

```
Cal_Sum_Mil();
```

```
Cal_Sum_Mjk();
```

```
Cal_Sum2_M();
```

```
KP=KQ=1;
```

```
for(iteration=1;iteration<=999;iteration++)
```

```
if((KP==1)||(KQ==1))
```

```
{
```

```
Cal_Sch_PQ();
```

```
Cal_Vint();
```

```
Cal_Del_PbyV();
```

```
Cal_Del_Pgen_byV();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ERROR_P():
if( convergen_p == 'F' )
{ //con_p
  Sol_Matrix_Delta();
  KQ=1;
  Cal_Det_QbyV();
  ERROR_Q();
  if( convergen_q == 'F' )
  {
    Sol_Matrix_V();
    KP=1;
  }
else
{
  KQ=0;
  if (KP==0)
  {
    Cal_V_slack();
    Stable_of_sys();
    Cal_Flow_PQ_1();
    compound();
    Setarrow();
    Make_Table_1();
    Make_Table_2();
    Make_Table_3();
    iteration=1000;
  }
}
} //end con_p
else
{
  KP=0;
  if (KQ==0)
  {
    Cal_V_slack();
    Stable_of_sys();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

Cal_V_slack();
Stable_of_sys();
Cal_Flow_PQ_1();
compound();
Setarrow();
Make_Table_1();
Make_Table_2();
Make_Table_3();
iteration=1000;
}

void getString(char *str,int x1,int x2,int y)
{
int ch,X,Y;
int i=0;
char tmp[2];

X=x1;Y=y;
while(ch!=13)
{
setcolor(7);
ch = getch();
if ((ch>=21)&&(ch<=176))
{
X+=8;
if(X>x2) {beep();X-=8;}
else
{ str[i] = ch;
tmp[0] = ch;
tmp[1] = '\000';
outtextxy(X-8,Y,tmp);
}
i++;
}
if (ch==47)
{

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        str[i] = '\\';
        i++;
    }
    if (ch==8)
    {
        if(i>0) i--;
        X-=8;
        if(X<x1) {beep();X=x1;}
        else {setfillstyle(1,0);
            bar(X,Y,X+8,Y+8);}
    }

    if (ch==13)
        str[i]='\000';
}
// sound(900);delay(150);nosound();
}

int get_old_filedata(void)
{
    int B_2,B_4,P_1,P_3,d=0,P,Q,n,m;
    int success = 1;
    int PHASE;

    clear_in_win();
    about_dialog();

    frame(10,40,629,80);           // 1st Frame
    outtext3d(20,58,"ENTER file name: ",0,0,14,6);
    space(157,50,624,71);
    getstring(filename,165,620,57);

    if((fp=fopen(filename,"rb"))==NULL)
    {
        dialog_box(110,160,530,305,1,15);
        outtext3d(165,180,"Can not open file !!!",1,4,12,0);

```

```

button(144,240,495,270,2,0);
outtext3d(225,250,"Press any key to continue",0,0,15,0);
printf("\007");
getch();
button(144,240,495,270,2,1);
outtext3d(226,251,"Press any key to continue",0,0,15,0);
sound(1000);delay(120);nosound();
button(144,240,495,270,2,0);
outtext3d(225,250,"Press any key to continue",0,0,15,0);
delay(200);
success = 0;
}
else // clear array
{
for(B_2 = 1 ; B_2 < MAX_BUS ; B_2++) //clear array
for(B_4 = 1 ; B_4 < MAX_BUS ; B_4++)
for(P_1 = 1 ; P_1 < 4 ; P_1++)
for(P_3 = 1 ; P_3 < 4 ; P_3++)
{
Y_G[P_1][B_2][P_3][B_4] = 0 ;
Y_B[P_1][B_2][P_3][B_4] = 0 ;
}
// read data
fread(&Dat,sizeof(struct Data),1,fp);
fclose(fp);
// display data
setfillstyle(1,3);
bar(110,160,531,306);
frame(10,100,629,140); // 2nd Frame
outtext3d(20,118,"Number of bus: ",0,0,14,6);
space(140,110,624,131);
int_to_string(Dat.MAX_BUS-1);
MAX_BUS = Dat.MAX_BUS;
setcolor(7);
outtextxy(148,117,string);

```

เอกสารฉบับนี้จัดทำขึ้นไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(B_2 = 1 ; B_2 < MAX_BUS ; B_2++) //output data
```

```
{
```

```
for(B_4 = 1 ; B_4 < MAX_BUS ; B_4++)
```

```
{
```

```
setfillstyle(1,3);
```

```
bar(22,172,617,208);
```

```
frame(20,170,619,210);
```

```
frame(20,230,619,310);
```

```
setfillstyle(1,3);
```

```
bar(22,232,617,308);
```

```
if (B_2==B_4)
```

```
{
```

```
strcpy(text,"Self admittance and susceptance of bus ");
```

```
int_to_string(B_2);
```

```
strcat(text,string);
```

```
outtext3d(30,188,text,0,0,14,6);
```

```
}
```

```
else
```

```
{
```

```
strcpy(text,"Admittance and susceptance between bus ");
```

```
int_to_string(B_2);
```

```
strcat(text,string);
```

```
strcat(text," and ");
```

```
int_to_string(B_4);
```

```
strcat(text,string);
```

```
outtext3d(30,188,text,0,0,14,6);
```

```
Relat[B_2][B_4] = Dat.Relat[B_2][B_4];
```

```
Relat[B_4][B_2] = Dat.Relat[B_4][B_2];
```

```
}
```

```
for(P_1=1;P_1<4;P_1++)
```

```
{
```

```
for(P_3=1;P_3<4;P_3++)
```

```
{
```

เอกสารนี้เป็นเอกสารของ Y\_G[P\_1][B\_2][P\_3][B\_4] = Dat.Y\_G[P\_1][B\_2][P\_3][B\_4]; วัตถุประสงค์ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Y_B[P_1][B_2][P_3][B_4] = Dat.Y_B[P_1][B_2][P_3][B_4];
switch ( P_1 )
(
    case 1 :{
        switch (P_3)
        {
            case 1:strcpy(text,"Between Phase A and Phase A ");break;
            case 2:strcpy(text,"Between Phase A and Phase B ");break;
            case 3:strcpy(text,"Between Phase A and Phase C ");break;
        }
        break;
    case 2 :{
        switch ( P_3 )
        {
            case 1:strcpy(text,"Between Phase B and Phase A ");break;
            case 2:strcpy(text,"Between Phase B and Phase B ");break;
            case 3:strcpy(text,"Between Phase B and Phase C ");break;
        }
        break;
    case 3 :{
        switch ( P_3 )
        {
            case 1 :strcpy(text,"Between Phase C and Phase A ");break;
            case 2 :strcpy(text,"Between Phase C and Phase B ");break;
            case 3 :strcpy(text,"Between Phase C and Phase C ");break;
        }
        break;
    } // close switch phase 1
    setfillstyle(1,3);
    bar(30,248,614,305);
    outtext3d(30,248,text,0,0,15,0);
    outtext3d(300,248,"G :",0,0,15,0);
    space(340,240,614,261);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

case 3 : break;
}
for(PHASE =1;PHASE < 4; PHASE++)
{
'setfillstyle(1,3);
bar(60,155,627,318);
MAGNI_V[PHASE][d] = Dat.MAGNI_V[PHASE][d];
ANGLE_V[PHASE][d] = Dat.ANGLE_V[PHASE][d];
G_P[PHASE][d] = Dat.G_P[PHASE][d];
G_Q[PHASE][d] = Dat.G_Q[PHASE][d];
L_P[PHASE][d] = Dat.L_P[PHASE][d];
L_Q[PHASE][d] = Dat.L_Q[PHASE][d];
switch ( PHASE )
{
case 1 : outtext3d(60,155,"Phase A",0,0,14,6);break;
case 2 : outtext3d(60,155,"Phase B",0,0,14,6);break;
case 3 : outtext3d(60,155,"Phase C",0,0,14,6);break;
}
outtext3d(140,155,"- Voltage magnitude ",0,0,15,0);
double_to_text(MAGNI_V[PHASE][d],10);
space(300,147,619,168);
setcolor(7);
outtextxy(308,154,text);
outtext3d(140,180,"- Voltage angle ",0,0,15,0);
double_to_text(ANGLE_V[PHASE][d],10);
space(300,172,619,193);
setcolor(7);
outtextxy(308,179,text);
outtext3d(140,205,"- MW of Gen ",0,0,15,0);
double_to_text(G_P[PHASE][d],10);
space(300,197,619,218);
setcolor(7);
outtextxy(308,204,text);
outtext3d(140,230,"- MVAR of Gen ",0,0,15,0);
double_to_text(G_Q[PHASE][d],10);

```

```

setcolor(7);
outtextxy(308,229,text);
outtext3d(140,255,"- MW of Load   ",0,0,15,0);
double_to_text(L_P[PHASE][d],10);
space(300,247,619,268);
setcolor(7);
outtextxy(308,254,text);
outtext3d(140,280,"- MVAR of Load   ",0,0,15,0);
double_to_text(L_Q[PHASE][d],10);
space(300,272,619,293);
setcolor(7);
outtextxy(308,279,text);

Relat_G[d] = Dat.Relat_G[d];
Relat_L[d] = Dat.Relat_L[d];
O[d-1] = Dat.O[d-1];
BUBB[d-1] = Dat.BUBB[d-1];
getch();
}
}

BASE = Dat.BASE;
BASE_V = Dat.BASE_V;
setfillstyle(1,3);
bar(12,102,627,318);
outtext3d(170,181,"BASE of POWER   :",0,0,14,6);
space(314,173,470,194);
gotoxy(41,12);
printf("%.10f",BASE);
outtext3d(170,213,"BASE of VOLTAGE :",0,0,14,6);
space(314,205,470,226);
gotoxy(41,14);
printf("%.10f",BASE_V);
getch();
}
return(success);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
int input_new_filedata_GB(void
```

```
{
```

```
int i,n,Y,j,check_bubb;
```

```
int P_1, B_2, P_3, B_4,change;
```

```
char ch;
```

```
int success = 1;
```

```
int order[5];
```

```
clear_in_win();
```

```
about_dialog();
```

```
frame(10,40,629,80); // 1st Frame
```

```
outtext3d(20,58,"ENTER file name: ",0,0,14,6);
```

```
space(157,50,624,71);
```

```
getstring(filename,165,620,57);
```

```
if((fp=fopen(filename,"r"))!=NULL)
```

```
{
```

```
dialog_box(110,160,530,305,1,15);
```

```
outtext3d(230,185,"WARNING !!!",1,4,12,0);
```

```
outtext3d(160,235,"Overwrite exist file ? (Y/N)",1,3,12,0);
```

```
printf("\007");
```

```
ch=getch();
```

```
setfillstyle(1,3);
```

```
bar(110,160,531,306);
```

```
if ((ch=='N')||(ch=='n'))
```

```
{
```

```
success = 0;
```

```
closegraph();
```

```
goto out_RX;
```

```
}
```

```
}
```

```
fclose(fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((fp=fopen(filename,"w"))==NULL)
{
    dialog_box(110,160,530,305,1,15);
    outtext3d(142,180,"Error in writing file !!!",1,4,12,0);
    button(144,240,495,270,2,0);
    outtext3d(225,250,"Press any key to continue",0,0,15,0);
    printf("\007");
    getch();
    button(144,240,495,270,2,1);
    outtext3d(226,251,"Press any key to continue",0,0,15,0);
    sound(1000);delay(120);nosound();
    button(144,240,495,270,2,0);
    outtext3d(225,250,"Press any key to continue",0,0,15,0);
    delay(200);
    closegraph();
    success = 0;
}
else
{ //big else
    setfillstyle(1,3);
    bar(110,160,531,306);
    frame(10,100,629,140); // 2nd Frame
    outtext3d(20,118,"Enter number of bus: ",0,0,14,6);
    space(190,110,624,131);
    getstring(number,198,620,117);
    MAX_BUS = str_to_int(number)+1;
    Dat.MAX_BUS = MAX_BUS;
    frame(10,160,629,320); // 3rd Frame

    // Clear buffers of Y_B and Y_B
    for(P_1=1 ; P_1 < 4 ; P_1++)
        for(B_2=1 ; B_2 < MAX_BUS ; B_2++)
            for(P_3=1 ; P_3 < 4 ; P_3++)
                for(B_4=1 ; B_4 < MAX_BUS ; B_4++)

```

```

        Dat.Y_B[P_1][B_2][P_3][B_4] = Y_B[P_1][B_2][P_3][B_4] = 0;
    }

```

```

// Input data GB

```

```

for(B_2 = 1 ; B_2 < MAX_BUS ; B_2++)

```

```

{ // start for BUS 2

```

```

    for(B_4 = 1 ; B_4 < MAX_BUS ; B_4++)

```

```

    { // start for BUS 4

```

```

        if (B_2<=B_4)

```

```

        { // start if (B_2<=B_4)

```

```

            setfillstyle(1,3);

```

```

            bar(22,172,617,208);

```

```

            frame(20,170,619,210);

```

```

            frame(20,230,619,310);

```

```

            setfillstyle(1,3);

```

```

            bar(22,232,617,308);

```

```

            if (B_2==B_4)

```

```

            {

```

```

                strcpy(text,"Is there self admittance and susceptance of bus ");

```

```

                int_to_string(B_2);

```

```

                strcat(text,string);

```

```

                strcat(text," (Y or N)");

```

```

                outtext3d(30,188,text,0,0,15,0);

```

```

                space(510,180,614,201);

```

```

            }

```

```

        else

```

```

        {

```

```

            strcpy(text,"Is there admittance and susceptance between bus ");

```

```

            int_to_string(B_2);

```

```

            strcat(text,string);

```

```

            strcat(text," and ");

```

```

            int_to_string(B_4);

```

```

            strcat(text,string);

```

```

            strcat(text," (Y or N)");

```

```

            outtext3d(30,188,text,0,0,15,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
ch = getch();
ch = toupper(ch);

if (ch=='Y')
{ // if (ch=='Y')
  if (B_2==B_4) outtextxy(518,188,"Y");
  else outtextxy(561,188,"Y");

  if(B_2 != B_4) //mutual admittance
  {
    Dat.Relat[B_2][B_4] = Relat[B_2][B_4] = 'Y'; //check relation between bus
    switch(B_2)
    {
    case 1 : BU[BB[0]]++;break;
    case 2 : BU[BB[1]]++;break;
    case 3 : BU[BB[2]]++;break;
    case 4 : BU[BB[3]]++;break;
    case 5 : BU[BB[4]]++;break;
    }
    switch(B_4)
    {
    case 1 : BU[BB[0]]++;break;
    case 2 : BU[BB[1]]++;break;
    case 3 : BU[BB[2]]++;break;
    case 4 : BU[BB[3]]++;break;
    case 5 : BU[BB[4]]++;break;
    }
  }
}

```

```
for(P_1=1;P_1<4;P_1++)
```

```
{--// start for phase 1
```

```
  for(P_3=1;P_3<4;P_3++)
```

```
    { // start for phase 3
```

```
      switch (P_1)
```

เอกสารนี้เป็นเอกสารที่มอบให้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1 :{
    switch (P_3)
    {
        case 1:strcpy(text,"Between Phase A and Phase A ");break;
        case 2:strcpy(text,"Between Phase A and Phase B ");break;
        case 3:strcpy(text,"Between Phase A and Phase C ");break;
    }
    }
    break;
case 2 :{
    switch ( P_3 )
    {
        case 1:strcpy(text,"Between Phase B and Phase A ");break;
        case 2:strcpy(text,"Between Phase B and Phase B ");break;
        case 3:strcpy(text,"Between Phase B and Phase C ");break;
    }
    }
    break;
case 3 :{
    switch ( P_3 )
    {
        case 1 :strcpy(text,"Between Phase C and Phase A ");break;
        case 2 :strcpy(text,"Between Phase C and Phase B ");break;
        case 3 :strcpy(text,"Between Phase C and Phase C ");break;
    }
    }
    break;
} // close switch phase 1
setfillstyle(1,3);
bar(30,248,614,305);
outtext3d(30,248,text,0,0,15,0);
outtext3d(300,248,"G :",0,0,15,0);
space(340,240,614,261);
getstring(number,348,610,247);
Y_G[P_1][B_2][P_3][B_4]=str_to_real(number);
Dat.Y_G[P_1][B_2][P_3][B_4] = Y_G[P_1][B_2][P_3][B_4];

```

```

        outtext3d(300,287,"B :",0,0,15.0);
        space(340,279,614,300);
        getstring(number,348,610,286);
        Y_B[P_1][B_2][P_3][B_4]=str_to_real(number);
        Dat.Y_B[P_1][B_2][P_3][B_4] = Y_B[P_1][B_2][P_3][B_4];
    } // end for phase 3
} // end for phase 1
} // end if (ch=='Y').
else
{
//-----
for(P_1=1:P_1<4:P_1++)
    for(P_3=1:P_3<4:P_3++)
    {
        Dat.Y_G[P_1][B_2][P_3][B_4] = Y_G[P_1][B_2][P_3][B_4] = 0;
        Dat.Y_B[P_1][B_2][P_3][B_4] = Y_B[P_1][B_2][P_3][B_4] = 0;
    }
//-----
} // close else (ch=='Y')
} //close condition 1
else
{
//-----
for(P_1=1:P_1<4:P_1++)
    for(P_3=1:P_3<4:P_3++)
    {
        Y_G[P_1][B_2][P_3][B_4] = Y_G[P_1][B_4][P_3][B_2];
        Y_B[P_1][B_2][P_3][B_4] = Y_B[P_1][B_4][P_3][B_2];
        Dat.Y_G[P_1][B_2][P_3][B_4] = Y_G[P_1][B_2][P_3][B_4];
        Dat.Y_B[P_1][B_2][P_3][B_4] = Y_B[P_1][B_2][P_3][B_4];
        Dat.Relat[B_2][B_4] = Relat[B_2][B_4] = Relat[B_4][B_2];
    }
//-----
}
} // end for BUS 4
} // end for BUS 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(B_2 = 1 ; B_2 < MAX_BUS ; B_2++)
{
    for(B_4 = 1 ; B_4 < MAX_BUS ; B_4++)
    {
        if(B_2<=B_4)
        {
            for(P_1 = 1 ; P_1 < 4 ; P_1++)
            {
                for(P_3 = 1 ; P_3 < 4 ; P_3++)
                {
                    if(P_1<=P_3)
                    {
                        Y_G[P_3][B_2][P_1][B_4] = Y_G[P_1][B_2][P_3][B_4];
                        Y_B[P_3][B_2][P_1][B_4] = Y_B[P_1][B_2][P_3][B_4];
                        Dat.Y_G[P_3][B_2][P_1][B_4] = Dat.Y_G[P_1][B_2][P_3][B_4];
                        Dat.Y_B[P_3][B_2][P_1][B_4] = Dat.Y_B[P_1][B_2][P_3][B_4];
                    }
                    Y_G[P_1][B_4][P_3][B_2] = Y_G[P_1][B_2][P_3][B_4];
                    Y_B[P_1][B_4][P_3][B_2] = Y_B[P_1][B_2][P_3][B_4];
                    Dat.Y_G[P_1][B_4][P_3][B_2] = Y_G[P_1][B_2][P_3][B_4];
                    Dat.Y_B[P_1][B_4][P_3][B_2] = Y_B[P_1][B_2][P_3][B_4];
                }
            }
        }
    }
}

for ( n=0 ; n<MAX_BUS ;n++ )
    O[n] = -1;

for(i=1 ; i< MAX_BUS ; i++)
{
    order[i-1] = 0;
    for ( j=1 ; j< MAX_BUS ; j++ )
    {
        check_bubb = 1;

        for ( n=0 ; n<MAX_BUS ; n++ )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (j == O[n] )
            check_bubb = 0;

        if (( BUBB[j-1] >= order[i-1] )&&( check_bubb != 0 ))
        {
            order[i-1] = BUBB[j-1];
            change = j;
        }
    }
    O[i-1] = change;
}

```

```

for(i=0;i<MAX_BUS-1;i++)
{
    Dat.BUBB[i] = BUBB[i];
    Dat.O[i] = O[i];
}
// close big else
out_RX :
return(success);
} //-----end of function

```

```

int input_new_filedata_VPQ(void)
{
    int k,d,m,n, PHASE ,(SP_MAX_BUS);
    double TEMP_DATA;
    int success = 1;
    double MAGNI_V1[SP_MAX_BUS];
    double ANGLE_V1[SP_MAX_BUS];

```

```

    char ch;

```

```

    pv = 0;

```

```

    m = 0;

```

```

    n = 1;

```

```

    frame(10,100,629,320); // 2nd+3rd Frame

```

```

    setfillstyle(1,3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bar(13,103,628,318);
outtext3d(220,125,"<< TYPE OF BUS SELECTION >>",0,0,14,6);
outtext3d(220,165,"1. SLACK BUS",0,0,15,0);
outtext3d(220,190,"2. VOLTAGE CONTROLLED BUS",0,0,15,0);
outtext3d(220,215,"3. LOAD BUS",0,0,15,0);
outtext3d(220,245," ( SELECT 1-3 )",0,0,15,0);
for (d=1; d < MAX_BUS ;d++)
(
    setfillstyle(1,3);
    bar(220,280,430,294);
    strcpy(text,"SELECT TYPE OF BUS ");
    int_to_string(d);
    strcat(text,string);
    strcat(text," : ");
    outtext3d(220,280,text,0,0,14,6);
    button(410,272,430,294,2,0);
    ch=getch();
    if (!(ch=='1')||(ch=='2')||(ch=='3'))
    (
        closegraph();
        printf("Error in input data !!!\n");
        exit(1);
    )
    t[d]=ch-'0';
    int_to_string(t[d]);
    button(410,272,430,294,2,1);
    outtext3d(416,280,string,0,0,12,0);
    sound(900);delay(100);nosound();delay(50);
    button(410,272,430,294,2,0);
    outtext3d(416,280,string,0,0,12,0);
    delay(150);
    Dat.t[d] = t[d];
)
for (d=1; d < MAX_BUS ;d++ )
( //begin loop 1
    strcpy(text,"<< BUS ");

```

```

int_to_string(d);
strcat(text,string);
switch(t[d])
{
case 1:n--;
    slack_bus = d;
    strcat(text," SLACK BUS >>");
    break;

case 2:m++;
    v_control[m] = d;
    pv = m ;
    strcat(text," PV BUS >>");
    break;

case 3:strcat(text," LOAD BUS >>");
    break;

default:closegraph();
    clrscr();
    printf("%35s","<<< DATA ERROR ! >>>");
    success = 0;
    goto out_VPQ;
}

setfillstyle(1,3);
bar(12,102,627,318);
outtext3d(30,125,text,0.0,14,6);
if(n<0)
{
    closegraph();
    clrscr();
    printf(" <<< DATA ERROR ! >>>\n");
    printf(" DUPLICATE SLACK BUS !\n");
    printf(" Press any Key to continue...");
    success = 0;
    getch();
    goto out_VPQ;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ // start if d==slack_bus
    outtext3d(60,155,"VOLTAGE MAGNITUDE OF PHASE A : ",0,0,15,0);
    space(350,147,619,168);
    getstring(number,358,615,154);
    MAGNI_V1[d]=str_to_real(number);

    outtext3d(60,185,"ANGLE OF PHASE A : ",0,0,15,0);
    space(350,177,619,198);
    getstring(number,358,615,184);
    ANGLE_V1[d]=str_to_real(number);

    MAGNI_V1[1][d] = MAGNI_V1[d];
    Dat.MAGNI_V1[1][d] = MAGNI_V1[1][d];
    ANGLE_V1[1][d] = ANGLE_V1[d];
    Dat.ANGLE_V1[1][d] = ANGLE_V1[1][d];
    for(PHASE = 2 : PHASE < 4 ; PHASE++)
    {
        Dat.MAGNI_V[PHASE][d] = Dat.MAGNI_V1[1][d] ;
        MAGNI_V[PHASE][d] = Dat.MAGNI_V1[1][d] ;
    }
    Dat.ANGLE_V[2][d] = ANGLE_V[2][d] = -1*(120);
    Dat.ANGLE_V[3][d] = ANGLE_V[3][d] = 1*(120);
    Dat.G_P[PHASE][d] = G_P[PHASE][d] = 0;
    Dat.G_Q[PHASE][d] = G_Q[PHASE][d] = 0;
    Dat.L_P[PHASE][d] = L_P[PHASE][d] = 0;
    Dat.L_Q[PHASE][d] = L_Q[PHASE][d] = 0;
    Dat.Relat_G[d] = Relat_G[d] = 'Y';
} // end if d==slack_bus
else
{ // start else d==slack_bus <= pv bus
    if(d == v_control[m])
    {
        outtext3d(60,155,"VOLTAGE MAGNITUDE OF PHASE A : ",0,0,15,0);
        space(350,147,619,168);
        getstring(number,358,615,154);

```

```

MAGNI_V1[d]=str_to_real(number);
Dat.MAGNI_V[1][d] = MAGNI_V[1][d] = MAGNI_V1[d] ;
for(PHASE = 2 ; PHASE < 4 ; PHASE++)
{
    Dat.MAGNI_V[PHASE][d] = Dat.MAGNI_V[1][d] ;
    MAGNI_V[PHASE][d] = Dat.MAGNI_V[1][d] ;
}
Dat.ANGLE_V[1][d] = ANGLE_V[1][d] = 0;
Dat.ANGLE_V[2][d] = ANGLE_V[2][d] = -1*(120) ;
Dat.ANGLE_V[3][d] = ANGLE_V[3][d] = 1*(120) ;

outtext3d(60,185,"MW OF GEN OF PHASE A : ",0,0,15,0);
space(350,177,619,198);
getstring(number,358,615,184);
G_P[1][d]=str_to_real(number);
if(G_P[1][d] != 0)
    Dat.Relat_G[d] = Relat_G[d] = 'Y'; // Is there Generator?
Dat.G_P[1][d] = G_P[1][d];
Dat.G_P[2][d] = G_P[2][d] = G_P[1][d];
Dat.G_P[3][d] = G_P[3][d] = G_P[1][d];
setfillstyle(1,3);
bar(60,155,627,318);
for(PHASE=1:PHASE<4:PHASE++)
{
    switch (PHASE)
    {
        case 1 :outtext3d(60,155,"MW OF LOAD OF PHASE A : ",0,0,15,0);
            space(350,147,619,168);
            getstring(number,358,615,154);
            L_P[PHASE][d]=str_to_real(number);
            if(L_P[1][d] != 0)
                Dat.Relat_L[d] = Relat_L[d] = 'Y';//Is there Load?
            Dat.L_P[PHASE][d] = L_P[PHASE][d] ;
            outtext3d(60,180,"MVAR OF LOAD OF PHASE A : ",0,0,15,0);
            space(350,172,619,193);
            getstring(number,358,615,179);

```

```

L_Q[PHASE][d]=str_to_real(number);
Dat.L_Q[PHASE][d] = L_Q[PHASE][d] ;
break;
case 2 :outtext3d(60,205,"MW OF LOAD OF PHASE B : ",0,0,15,0);
space(350,197,619,218);
getstring(number,358,615,204);
L_P[PHASE][d]=str_to_real(number);
Dat.L_P[PHASE][d] = L_P[PHASE][d] ;
outtext3d(60,230,"MVAR OF LOAD OF PHASE B : ",0,0,15,0);
space(350,222,619,243);
getstring(number,358,615,229);
L_Q[PHASE][d]=str_to_real(number);
Dat.L_Q[PHASE][d] = L_Q[PHASE][d] ;
break;
case 3 :outtext3d(60,255,"MW OF LOAD OF PHASE C : ",0,0,15,0);
space(350,247,619,268);
getstring(number,358,615,254);
L_P[PHASE][d]=str_to_real(number);
Dat.L_P[PHASE][d] = L_P[PHASE][d] ;
outtext3d(60,280,"MVAR OF LOAD OF PHASE C : ",0,0,15,0);
space(350,273,619,293);
getstring(number,358,615,279);
L_Q[PHASE][d]=str_to_real(number);
Dat.L_Q[PHASE][d] = L_Q[PHASE][d] ;
break;
)
)
}
else //load bus
(
for(PHASE=1 :PHASE<4:PHASE++)
Dat.MAGNI_V[PHASE][d] = MAGNI_V[PHASE][d] = 1;
Dat.ANGLE_V[1][d] = ANGLE_V[1][d] = 0;
Dat.ANGLE_V[2][d] = ANGLE_V[2][d] = -1*(120) ;
Dat.ANGLE_V[3][d] = ANGLE_V[3][d] = 1*(120) ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(PHASE=1:PHASE<4;PHASE++)
{
switch(PHASE)
{
case 1 :outtext3d(60,155,"MW OF LOAD OF PHASE A : ",0,0,15,0);
space(350,147,619,168);
getstring(number,358,615,154);
L_P[PHASE][d]=str_to_real(number);
if(L_P[1][d] !=0)
Dat.Relat_L[d] = Relat_L[d] = 'Y';
Dat.L_P[PHASE][d] = L_P[PHASE][d] ;
outtext3d(60,180,"MVAR OF LOAD OF PHASE A : ",0,0,15,0);
space(350,172,619,193);
getstring(number,358,615,179);
L_Q[PHASE][d]=str_to_real(number);
Dat.L_Q[PHASE][d] = L_Q[PHASE][d] ;
break;
case 2 :outtext3d(60,205,"MW OF LOAD OF PHASE B : ",0,0,15,0);
space(350,197,619,218);
getstring(number,358,615,204);
L_P[PHASE][d]=str_to_real(number);
Dat.L_P[PHASE][d] = L_P[PHASE][d];
outtext3d(60,230,"MVAR OF LOAD OF PHASE B : ",0,0,15,0);
space(350,222,619,243);
getstring(number,358,615,229);
L_Q[PHASE][d]=str_to_real(number);
Dat.L_Q[PHASE][d] = L_Q[PHASE][d] ;
break;
case 3 :outtext3d(60,255,"MW OF LOAD OF PHASE C : ",0,0,15,0);
space(350,247,619,268);
getstring(number,358,615,254);
L_P[PHASE][d]=str_to_real(number);
Dat.L_P[PHASE][d] = L_P[PHASE][d] ;
outtext3d(60,280,"MVAR OF LOAD OF PHASE C : ",0,0,15,0);
space(350,273,619,293);

```

เอกสารนี้เป็นเอกสารที่ getstring(number,358,615,279); การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

L_Q[PHASE][d]=str_to_real(number);
Dat.L_Q[PHASE][d] = L_Q[PHASE][d] ;
break:

```

```

}
}
}
}
}

```

```
setfillstyle(1,3);
```

```
bar(12,102,627,318);
```

```
outtext3d(170,180,"Enter BASE of POWER : ",0,0,14,6);
```

```
space(370,172,470,193);
```

```
getstring(number,378,466,179);
```

```
BASE=str_to_real(number);
```

```
Dat.BASE = BASE;
```

```
outtext3d(170,210,"Enter BASE of VOLTAGE : ",0,0,14,6);
```

```
space(370,202,470,223);
```

```
getstring(number,378,466,209);
```

```
BASE_V=str_to_real(number);
```

```
Dat.BASE_V = BASE_V;
```

```
out_VPQ;
```

```
if(fwrite(&Dat,sizeof(struct Data),1,fp)!=1)
```

```
{
```

```
closegraph();
```

```
clrscr();
```

```
printf("\n\nError in writing file!!\N007");
```

```
getch();
```

```
exit(1);
```

```
}]
```

```
fclose(fp);
```

```
return(success);
```

```
}
```

```
void DATA_OF_Y(void)
```

{ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int PH_P,BS_I,PH_M,BS_K,m;
double DAT,DATA_1;

for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++)
{ // for BS_I
for(PH_P = 1 ; PH_P < 4 ; PH_P++)
{ // for PH_P
for(BS_K = 1 ; BS_K < MAX_BUS ; BS_K++)
{ // for BS_K
for(PH_M = 1 ; PH_M < 4; PH_M++)
{ // for PH_M
DAT = pow(Y_G[PH_P][BS_I][PH_M][BS_K],2) + pow(Y_B[PH_P][BS_I][PH_M]
[BS_K],2);
MAGN_Y[PH_P][BS_I][PH_M][BS_K] = sqrt(DAT);

if(Y_G[PH_P][BS_I][PH_M][BS_K]!=0)
DATA_1= Y_B[PH_P][BS_I][PH_M][BS_K]/Y_G[PH_P][BS_I][PH_M][BS_K];
else
DATA_1= 0;
DATA_1 = atan(DATA_1);//Value in radian
DATA_1 = DATA_1*180/pi;
if(Y_G[PH_P][BS_I][PH_M][BS_K]<0) DATA_1=DATA_1+180;
ANG_Y[PH_P][BS_I][PH_M][BS_K] = DATA_1;
PIK = 1;
}
}
}
}

for(BS_I=1;BS_I<MAX_BUS;BS_I++)
{
check_pv = 1 ;
for(m=1 ;m < (pv+1) ; m++)
if(v_control[m] == BS_I)
check_pv = 0 ;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ((check_pv == 0) || (BS_I == slack_bus))
```

```
{  
    Y_Ggen[BS_I] = 0;  
    Y_Bgen[BS_I] = -100;  
}
```

```
DAT = pow(Y_Ggen[BS_I],2) + pow(Y_Bgen[BS_I],2);
```

```
MAGNI_Ygen[BS_I] = sqrt(DAT);
```

```
if(Y_Ggen[BS_I] == 0)
```

```
{  
    if(Y_Bgen[BS_I] != 0)  
        ANG_Ygen[BS_I] = -90;  
    else  
        ANG_Ygen[BS_I] = 0;  
}
```

```
else
```

```
{  
    if(Y_Bgen[BS_I] != 0)  
        ANG_Ygen[BS_I] = atan2(Y_Bgen[BS_I], Y_Ggen[BS_I]);  
    PIK=1;  
}
```

```
void Cal_Sch_PQ (void)
```

```
{
```

```
int BS_I,m , BS_K , PH_P , PH_M ;
```

```
for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++)
```

```
{
```

```
if(BS_I != slack_bus)
```

```
{
```

```
for(PH_P = 1 ; PH_P < 4 ; PH_P++)
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Psch[PH_P][BS_I] = G_P[PH_P][BS_I]- L_P[PH_P][BS_I] ;
Psch[PH_P][BS_I] = Psch[PH_P][BS_I]/BASE ;
Qsch[PH_P][BS_I] = G_Q[PH_P][BS_I]- L_Q[PH_P][BS_I] ;
Qsch[PH_P][BS_I] = Qsch[PH_P][BS_I]/BASE ;
}
}
}
}
}
}

```

```
void Cal_Dcl_PbyV(void)
```

```

{
double DAT_P , DAT_Y , SUM_P ;
double DAT_COS , DAT_SIN , DAT_ANG , SUM_K ;
int BS_I , BS_K , PH_P , PH_M ;

for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++) // loop BUS I
{ // for BS_I
if(BS_I != slack_bus) // only if not slack_bus
{ // if BS_I
for(PH_P = 1 ; PH_P <= 3 ; PH_P++) // loop 3 Phase
{ // for PH_P
SUM_K = 0; // initial SUM_K
for(BS_K = 1 ; BS_K < MAX_BUS ; BS_K++)// loop BUS K
{ // for BS_K
SUM_P = 0; // initial SUM_P
for(PH_M = 1 ; PH_M <=3 ; PH_M++)// loop 3 Phase
{ // for PH_M

DAT_ANG = ANGLE_V[PH_P][BS_I];
DAT_ANG -= ANGLE_V[PH_M][BS_K];
DAT_ANG -= ANG_Y[PH_P][BS_I][PH_M][BS_K];
DAT_ANG = (DAT_ANG * pi)/180;
DAT_COS = cos( DAT_ANG ); // Real part of angle

DAT_P = MAGNI_Y[PH_P][BS_I][PH_M][BS_K];
DAT_P *= MAGNI_V[PH_M][BS_K];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DAT_P *= DAT_COS;
```

```
SUM_P += DAT_P;
```

```
PIK = 1;
```

```
} // end for PH_M
```

```
SUM_K += SUM_P;
```

```
}// end for BS_K
```

```
SUM_K *= MAGNI_V[PH_P][BS_I];
```

```
SUM_K = Psch[PH_P][BS_I] - SUM_K;
```

```
DEL_PbyV[PH_P][BS_I] = SUM_K / MAGNI_V[PH_P][BS_I];
```

```
PIK = 1;
```

```
} // end for PH_P
```

```
} // end if BS_I
```

```
} // end for BS_I
```

```
}
```

```
void Cal_Del_Pgen_byV(void)
```

```
{
```

```
double DAT_P , SUM_P , SUM_K , SUM_H , m ;
```

```
double DAT_SIN , DAT_ANG ;
```

```
int BS_J , BS_K , PH_P , PH_M ;
```

```
for(BS_J = 1 ; BS_J < MAX_BUS ; BS_J++)
```

```
{
```

```
check_pv = 1 ;
```

```
for(m=1 ; m < (pv+1) ; m++)
```

```
if(v_control[m] == BS_J)
```

```
check_pv = 0 ;
```

```
if(check_pv == 0)
```

```
for(PH_P = 1 ; PH_P < 4 ; PH_P++)
```

```
{
```

เอกสารนี้เป็น Psch[PH\_P][BS\_J] = G\_P[PH\_P][BS\_J] - L\_P[PH\_P][BS\_J] ; กรุณาให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Psch[PH_P][BS_J] = Psch[PH_P][BS_J]/BASE ;
Psch_gen[BS_J] += Psch[PH_P][BS_J];
}

```

```
SUM_H=DAT_P=0;
```

```
for(BS_J = 1 ; BS_J < MAX_BUS ; BS_J++)
```

```
{
```

```
check_pv = 1 ;
```

```
for(m=1 ; m < (pv+1) ; m++)
```

```
if(v_control[m] == BS_J)
```

```
check_pv = 0 ;
```

```
if(check_pv == 0)
```

```
{
```

```
for(PH_P=1;PH_P<4;PH_P++)
```

```
{
```

```
SUM_K=0;
```

```
for(BS_K=1;BS_K<MAX_BUS;BS_K++)
```

```
{
```

```
if(BS_K==BS_J)
```

```
{
```

```
SUM_P=0;
```

```
for(PH_M = 1 ; PH_M < 4; PH_M++)
```

```
{
```

```
DAT_ANG = ANGLE_Vint[BS_J];
```

```
DAT_ANG -= ANGLE_V[PH_M][BS_J];
```

```
DAT_ANG = (DAT_ANG * pi)/180;
```

```
DAT_SIN = sin( DAT_ANG );
```

```
DAT_P = Y_Bgen[BS_J];
```

```
DAT_P *= DAT_SIN;
```

```
DAT_P *= MAGNI_V[PH_M][BS_K];
```

```
SUM_P += DAT_P;
```

```
PIK = 1;
```

```
} // end for PH_M
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SUM_K += SUM_P;
    }
}
SUM_K *= MAGNI_Vint[BS_J];
SUM_H += SUM_K;
}
SUM_H = Psch_gen[BS_J] - SUM_H;
DEL_Pgen_byVint[BS_J] = SUM_H / MAGNI_Vint[BS_J];
PIK = 1;
}
}
}

```

```

void Cal_Del_QbyV(void)
{
    double DAT_Q , DAT_Y , SUM_Q, COL;
    double DAT_COS , DAT_SIN , DAT_ANG;
    int BS_I , BS_K , PH_P , PH_M , m;

    for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++)
    { // for BS_I

        check_pv = 1 ;
        for(m=1 ; m<(pv+1) ; m++)
            if(v_control[m] == BS_I) check_pv = 0;

        if ((BS_I!=slack_bus)&&(check_pv!=0))
        { // if'
            for(PH_P = 1 ; PH_P <=3 ; PH_P++)
            { // for PH_P
                COL = 0;
                for(BS_K = 1 ; BS_K < MAX_BUS ; BS_K++)
                { // for BS_K
                    SUM_Q = 0;
                    for(PH_M = 1 ; PH_M <=3 ; PH_M++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DAT_ANG = ANGLE_V[PH_P][BS_I];
DAT_ANG -= ANGLE_V[PH_M][BS_K];
DAT_ANG -= ANG_Y[PH_P][BS_I][PH_M][BS_K];
DAT_ANG = (DAT_ANG * pi)/180;
DAT_SIN = sin( DAT_ANG );

```

```

DAT_Q = MAGNI_Y[PH_P][BS_I][PH_M][BS_K];
DAT_Q *= MAGNI_V[PH_M][BS_K];
DAT_Q *= DAT_SIN;
SUM_Q += DAT_Q;

```

```

    PIK = 1;
} // end for PH_P
COL += SUM_Q ;
} // end for BS_K
COL *= MAGNI_V[PH_P][BS_I];
COL = Qsch[PH_P][BS_I] - COL;
DEL_QbyV[PH_P][BS_I] = COL / MAGNI_V[PH_P][BS_I];
PIK = 1;
} // end for PH_P
} // if
} // end for BS_I
}
//-----
void Cal_M(void)
(
double DATA_1 ,DATA_2 ,ANG ,Angle[4][SP_MAX_BUS][4][SP_MAX_BUS];
int BS_I ,BS_K ,PH_P ,PH_M ,x ,y ,stand;

y=0;
stand=3;
for(BS_I=1;BS_I<MAX_BUS;BS_I++)
{ // for BS_I
if(BS_I!=slack_bus)
{ // if BS_I

```

```

x=0;
for(BS_K = 1; BS_K < MAX_BUS; BS_K++)
{ // for BS_K
    if(BS_K != slack_bus)
    { // if BS_K
        for(PH_P = 1; PH_P < 4; PH_P++)
        { // for PH_P
            y++;
            for(PH_M = 1; PH_M < 4; PH_M++)
            { // for PH_M
                if((PH_P == 1) && (PH_M == 2)) Angle[PH_P][BS_I][PH_M][BS_K] = 120;
                if((PH_P == 1) && (PH_M == 3)) Angle[PH_P][BS_I][PH_M][BS_K] = -120;
                if((PH_P == 2) && (PH_M == 1)) Angle[PH_P][BS_I][PH_M][BS_K] = -120;
                if((PH_P == 2) && (PH_M == 3)) Angle[PH_P][BS_I][PH_M][BS_K] = -240;
                if((PH_P == 3) && (PH_M == 1)) Angle[PH_P][BS_I][PH_M][BS_K] = 120;
                if((PH_P == 3) && (PH_M == 2)) Angle[PH_P][BS_I][PH_M][BS_K] = 240;
                if(PH_P == PH_M) Angle[PH_P][BS_I][PH_M][BS_K] = 0;

                ANG = (Angle[PH_P][BS_I][PH_M][BS_K] * pi) / 180;
                DATA_1 = sin(ANG);
                DATA_1 *= Y_G[PH_P][BS_I][PH_M][BS_K];
                DATA_2 = cos(ANG);
                DATA_2 *= Y_B[PH_P][BS_I][PH_M][BS_K];
                M[PH_P][BS_I][PH_M][BS_K] = DATA_1 - DATA_2;
                B[y][++x] = M[PH_P][BS_I][PH_M][BS_K];

                if ((PH_M == 3) && (y != stand)) x = x - 3;
            } // end for PH_M
            if(PH_P == 3)
                if(BS_K == (MAX_BUS - 1))
                    stand = stand + 3;
                else
                    y = y - 3;
        } // end for PH_P
    } // end if BS_K
} // end for BS_K

```

```

    } // end if BS_I
} // end for BS_I
TRAN_Y = MAX_Y = y;
TRAN_X = MAX_X = x;
}

```

```
void Cal_M_Q(void)
```

```

{
    double DATA_1 ,DATA_2 ,ANG ,Angle[4][SP_MAX_BLS][4][SP_MAX_BUS];
    int BS_I ,BS_K ,PH_P ,PH_M ,x ,y ,m, stand;

    y=0;
    stand=3;
    for(BS_I=1;BS_I<MAX_BUS;BS_I++)
    { // for BS_I
        if(BS_I != slack_bus)
        { // if BS_I == slack bus
            check_pv = 1 ;
            for(m=1 ;m < (pv+1) ; m++)
                if(v_control[m] == BS_I)
                    check_pv = 0 ;
            if (check_pv != 0 )
            { // if check_pv 1
                x=0;
                for(BS_K =1;BS_K <MAX_BUS;BS_K++)
                { // for BS_K
                    if(BS_K != slack_bus)
                    { // if BS_K == slack bus
                        check_pv = 1 ;
                        for(m=1 ;m < (pv+1) ; m++)
                            if(v_control[m] == BS_K)
                                check_pv = 0 ;
                        if (check_pv != 0 )
                        { // if check_pv 2
                            for(PH_P =1;PH_P <4;PH_P++)
                                { // for PH_P

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y++;
for(PH_M=1;PH_M<4;PH_M++)
{ // for PH_M
  if((PH_P==1)&&(PH_M==2))Angle[PH_P][BS_I][PH_M][BS_K]= 120;
  if((PH_P==1)&&(PH_M==3))Angle[PH_P][BS_I][PH_M][BS_K]= -120;
  if((PH_P==2)&&(PH_M==1))Angle[PH_P][BS_I][PH_M][BS_K]= -120;
  if((PH_P==2)&&(PH_M==3))Angle[PH_P][BS_I][PH_M][BS_K]= -240;
  if((PH_P==3)&&(PH_M==1))Angle[PH_P][BS_I][PH_M][BS_K]= 120;
  if((PH_P==3)&&(PH_M==2))Angle[PH_P][BS_I][PH_M][BS_K]= 240;
  if(PH_P==PH_M)Angle[PH_P][BS_I][PH_M][BS_K] = 0;

  ANG = (Angle[PH_P][BS_I][PH_M][BS_K]*pi)/180;
  DATA_1 = sin(ANG);
  DATA_1 *= Y_G[PH_P][BS_I][PH_M][BS_K];
  DATA_2 = cos(ANG);
  DATA_2 *= Y_B[PH_P][BS_I][PH_M][BS_K];
  M[PH_P][BS_I][PH_M][BS_K] = DATA_1 - DATA_2;
  C[y][++x] = M[PH_P][BS_I][PH_M][BS_K] ;

  if ((PH_M==3)&&(y!=stand)) x=x-3;
} // end for PH_M
if(PH_P==3)
  if(BS_K==(MAX_BUS-1))
    stand = stand+3;
  else
    y=y-3;
} // end for PH_P
} // end if check_pv 2
} // end if BS_K
} // end for BS_K
} // end if check_pv 1
} // end if BS_I == slackbus
} // end for BS_I
}

```

void Cal\_Sum\_Mil(void)  
 ขอสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
int m,x,y;
int BS_I , BS_L , PH_P ;

x=TRAN_X;
y=0;

for (BS_L=1;BS_L<MAX_BUS;BS_L++)
{
check_pv = 1 ;
for(m=1 ;m < (pv+1) ; m++)
if(v_control[m] == BS_L)
check_pv = 0 ;
if(check_pv == 0)
{
for (BS_I=1;BS_I<MAX_BUS;BS_I++)
if(BS_I!=slack_bus)
{
x++;
for(PH_P=1; PH_P<4 ;PH_P++)
{
y++;
if(BS_I==BS_L)
B[y][x] = -Y_Bgen[BS_L];
else
B[y][x] = 0;
}
if(BS_I>=MAX_BUS)
y=0;
if(BS_I!=(MAX_BUS-1))
x--;
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Cal_Sum_Mjk(void)
{
    int x,y;

    for(x=(TRAN_X+1);x<(TRAN_X+pv+1);x++)
        for(y=1;y<=((MAX_BUS-2)*3);y++)
            {
                B[x][y] = B[y][x];
            }
    x=(TRAN_X+1);
    for(y=1;y<=((MAX_BUS-2)*3);y++)
        B[y][x] = 0;
}

```

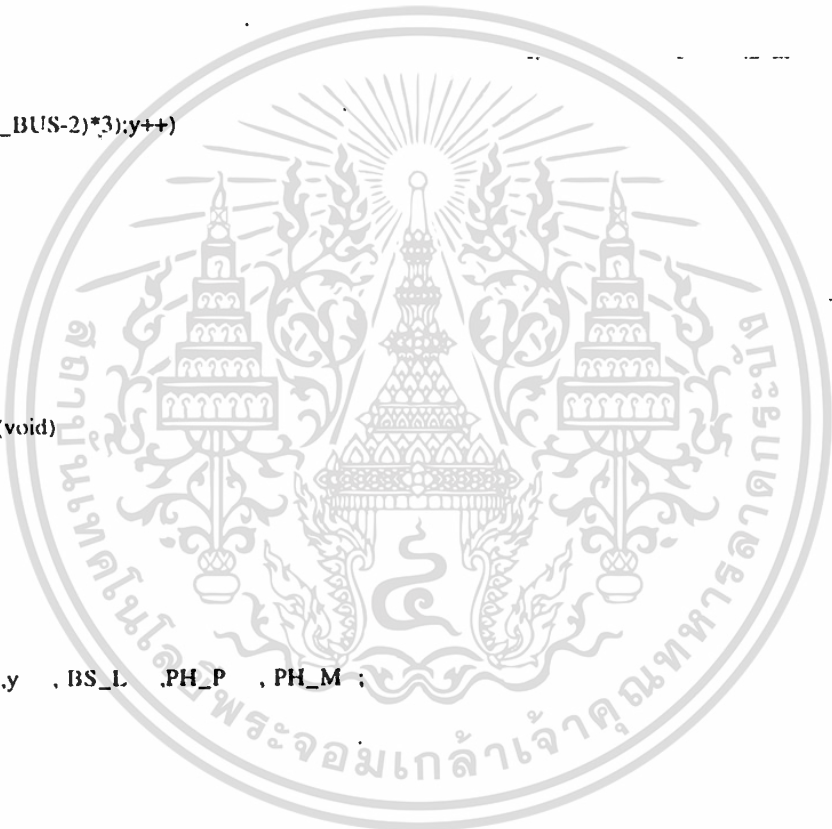
```

void Cal_Sum2_M(void)
{
    double data;
    int m;
    int BS_J ,x ,y , BS_L , PH_P , PH_M ;

    y=TRAN_Y;
    x=TRAN_X;
    for( BS_J=1; BS_J<MAX_BUS; BS_J++)
        {
            check_pv = 1 ;
            for(m=1 ;m < (pv+1) : m++)
                if(v_control[m] == BS_J)
                    check_pv = 0 ;

            if(check_pv == 0 )
                {

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

check_pv =0;

if (check_pv ==0)
{
    TRANS_Y++;
    DAT = DEL_Pgen_byVint[BS_J] ;
    JCBN[TRANS_Y][MAX_X+1] = DAT ;
}
}

/*----- DelP , SolMatrix */

for(Y = 1 ; Y < P_EQATN+1 ; Y++)
{
    for(X = 1 ; X < P_EQATN+1 ; X++)
    {
        if(X != Y)
        {
            MULTI = JCBN[X][Y] / JCBN[Y][Y] ;
            for(Z = 1 ; Z < P_EQATN+2 ; Z++)
            {
                JCBN[X][Z] = JCBN[X][Z] - ( MULTI * JCBN[Y][Z] ) ;
            }
        }
    }
}

for(Y = 1 ; Y < P_EQATN+1 ; Y++)
{
    ANS[Y] = JCBN[Y][P_EQATN+1] / JCBN[Y][Y] ;
}

/*----- Angle = Angle + Del */

Y=1;

for(BS_I=1;BS_I< MAX_BUS;BS_I++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(BS_I != slack_bus)

for( PH_P=1; PH_P < 4 ; PH_P++,Y++)

    if(Y <= TRAN_Y)
    {
        DAT = ANS[Y] ;
        ANGLE_V[PH_P][BS_I] = ANGLE_V[PH_P][BS_I] + DAT ;
    }

TRANS_Y = TRAN_Y;
for(BS_J=1;BS_J<MAX_BUS;BS_J++)
{
    check_pv=1;
for(s=1;s<(pv+1):s++)
    if(v_control[s]==BS_J)
        check_pv =0;

if(check_pv==0)
    (
        TRANS_Y++;
        DAT = ANS[TRANS_Y];
        DAT = DAT/3;
        ANGLE_Vint[BS_J] = ANGLE_Vint[BS_J]+DAT ;
    )
}

} /* END OF SOL_MATRIX_DELTA */

```

```

void ERROR_P(void)

```

```

{

int BS_I,PH_P,BS_J;
int s;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต  
 ไม่ควรนำเอกสารนี้ไปใช้ในการเรียนการสอนโดยไม่ได้รับอนุญาต  
 ไม่ควรนำเอกสารนี้ไปใช้ในการทำวิจัยโดยไม่ได้รับอนุญาต  
 ไม่ควรนำเอกสารนี้ไปใช้ในการทำธุรกิจโดยไม่ได้รับอนุญาต  
 ไม่ควรนำเอกสารนี้ไปใช้ในการทำโครงการโดยไม่ได้รับอนุญาต  
 ไม่ควรนำเอกสารนี้ไปใช้ในการทำสิ่งอื่นใดโดยไม่ได้รับอนุญาต

```

convergen_p = 'T' ;
for(BS_I=1;BS_I < MAX_BUS :BS_I++)
  if(BS_I!=slack_bus)
    for(PH_P=1;PH_P < 4 :PH_P++)
      (
        DAT_1 = sqrt(DEL_PbyV[PH_P][BS_I]*DEL_PbyV[PH_P][BS_I]) ;
        if(( DAT_1 > ERROR) || (convergen_p == 'F'))
          convergen_p = 'F' ;
        else
          convergen_p = 'T' ;
      )
}

```

```

void Sol_Matrix_V(void)

```

```

{
  double  DAT , MULTI ,MAX, ANS[SP_MAX_BUS*2];
  double  JCBN[(SP_MAX_BUS-1)*3][(SP_MAX_BUS-1)*3] ;
  int     s,FLX;
  int     Y , Z , X , P1 , B2 , P3 , B4 , BX , BY,BS_I,PH_P,BS_J ;
  int     P_EQATN , Q_EQATN ;
  /*----- Clear all */

```

```

MAX=((MAX_BUS-2-pv)*3);

```

```

for(Y = 1 ; Y <= MAX ; Y++)

```

```

{
  for(X = 1 ; X <= MAX+1 ; X++)
    {
      JCBN[Y][X] = 0 ;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*----- JCBN <=B */
Q_EQATN=0;
for( Y = 1 ; Y <= MAX ; Y++)
{
    Q_EQATN++;
    for( X = 1 ;X<= MAX ; X++)
    {
        DAT = C[Y][X] ;
        JCBN[Y][X] = DAT ;
    }
    FIX = X;
}

```

```

/*----- JCBN <= DelQbyV */
Y=1;
for( BS_I=1; BS_I < MAX_BUS ; BS_I++ )
{
    check_pv = 1;
    for (s=1;s<(pv+1);s++)
        if (v_control[s] ==BS_I)
            check_pv =0;
    if ((check_pv !=0)&&(BS_I != slack_bus))
    {
        for( PH_P=1; PH_P < 4 ; PH_P++,Y++)
        {
            DAT = DEL_QbyV[PH_P][BS_I] ;
            JCBN[Y][FIX] = DAT ;
        }
    }
}

```

```

/*----- Del_Q SolMatrix */

```

```

for(Y = 1 ; Y < Q_EQATN+1 ; Y++)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(X = 1 ; X < Q_EQATN+1 ; X++)
{
    if(X != Y)
    {
        MULTI = JCBN[X][Y] / JCBN[Y][Y] ;
        for(Z = 1 ; Z < Q_EQATN+2 ; Z++)
        {
            JCBN[X][Z] = JCBN[X][Z] - ( MULTI * JCBN[Y][Z] ) ;
        }
    }
}

for(Y = 1 ; Y < Q_EQATN+1 ; Y++)
{
    ANS[Y] = JCBN[Y][Q_EQATN+1] / JCBN[Y][Y] ;
}

/*----- Magnitude = Magnitude + Del_V */
Y=1;
for(BS_I=1;BS_I< MAX_BUS;BS_I++)
{
    check_pv=1;
    for(s=1;s<(pv+1);s++)
        if(v_control[s]==BS_I)
            check_pv =0;

    if((check_pv!=0)&&(BS_I != slack_bus))
        for( PH_P=1; PH_P < 4 ; PH_P++,Y++)
        {
            DAT = ANS[Y] ;
            MAGNI_V[PH_P][BS_I] =MAGNI_V[PH_P][BS_I] + DAT;
        }
}

} /* END OF SOL_MATRIX_V */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(X = 0 ; X <= MAX_X ; X++)
{
    JCBN[Y][X] = 0 ;
}
}

```

```

/*----- JCBN <=B */

```

```

P_EQATN=0;

```

```

for( Y = 1 ; Y <= MAX_Y ; Y++)

```

```

{
    P_EQATN++;
    for( X = 1 :X<= MAX_X ; X++)
    {
        DAT = B[Y][X] ;
        JCBN[Y][X] = DAT ;
    }
}

```

```

/*----- JCBN <= DelPbyV */

```

```

Y=1;

```

```

for( BS_I=1 ; BS_I < MAX_BUS ; BS_I++)

```

```

{
    if(BS_I != slack_bus)

```

```

    for( PH_P=1; PH_P < 4 ; PH_P++,Y++)

```

```

        DAT = DEL_PbyV[PH_P][BS_I] ;

```

```

        JCBN[Y][MAX_X+1] = DAT ;

```

```

    }
    TRANS_Y = TRAN_Y;

```

```

    for(BS_J=1; BS_J <MAX_BUS ; BS_J++)

```

```

    {
        check_pv = 1;
        for (s=1;s<(pv+1);s++)

```

```

            if (v_control[s] ==BS_J)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void ERROR_Q(void),
```

```
{
```

```
int BS_I,PH_P;
```

```
double DAT_1;
```

```
convergen_q = "T" ;
```

```
for(BS_I=1;BS_I < MAX_BUS ;BS_I++)
```

```
for(PH_P=1;PH_P < 4 :PH_P++)
```

```
{
```

```
DAT_1 = sqrt(DEL_QbyV[PH_P][BS_I]*DEL_QbyV[PH_P][BS_I]) ;
```

```
if(( DAT_1 > ERROR) || (convergen_q == 'F'))
```

```
convergen_q = 'F' ;
```

```
else
```

```
convergen_q = "T" ;
```

```
{
```

```
}
```

```
void Cal_Vint(void)
```

```
{
```

```
int BS_I,BS_K,s,PH_P;
```

```
double DAT,DAT_1,DAT_2,DAT_3,DAT_4,
```

```
temp_1,temp_2,temp_3,temp_4,temp_5,temp_6,temp_7,temp_8,
```

```
SUM_1,SUM_2,SUM_3,SUM_4,
```

```
TAN_1,TAN_2,TAN_3,TAN_4,TAN_5,TAN_6,TAN_7,
```

```
DATA_1,DATA_2,DATA_3,DATA_4,
```

```
DATA_5,DATA_6,DATA_7,DATA_8 ;
```

```
double Real_Cur,Img_Cur,MAGNI_Cur,ANGLE_Cur,REAL_V,IMG_V;
```

```
double Real_Vint,IMG_Vint ;
```

```
for(BS_I = 1 ;BS_I < MAX_BUS ;BS_I++)
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(s=1;s<(pv+1);s++)
    if(v_control[s]==BS_I)
        check_pv =0;
if(check_pv==0)
{
SUM_3=SUM_4=0;
for( BS_K = 1 :BS_K < MAX_BUS ;BS_K++)
{ //for
if( BS_K != BS_I)
{ // if
temp_1 = MAGNI_V[1][BS_I] * MAGNI_Y[1][BS_I][1][BS_K] ;
temp_2 = MAGNI_V[1][BS_K] * MAGNI_Y[1][BS_I][1][BS_K] ;
temp_3 = ANGLE_V[1][BS_I] + ANG_Y[1][BS_I][1][BS_K];
temp_3 = (temp_3*pi)/180;
temp_4 = ANGLE_V[1][BS_K] + ANG_Y[1][BS_I][1][BS_K];
temp_4 = (temp_4*pi)/180;
DAT = cos(temp_4);
temp_5 = temp_2*DAT;
DAT = sin(temp_4);
temp_6 = temp_2*DAT;
DAT = cos(temp_3);
temp_7 = temp_1*DAT;
DAT = sin(temp_3);
temp_8 = temp_1*DAT;
SUM_1 = (temp_8 - temp_6);
SUM_2 = (temp_7 - temp_5);
} // end if
SUM_3 = SUM_3 + SUM_2;
SUM_4 = SUM_4 + SUM_1;
SUM_1 = SUM_2 =0;

} //for
Real_Cur = SUM_3 ;
Img_Cur = SUM_4 ;
MAGNI_Cur = sqrt((Real_Cur*Real_Cur)+(Img_Cur*Img_Cur));

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้ทำซ้ำหรือเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายวิชาการ โทร. 0-2916-1111

```
ANGLE_Cur = 0;
```

```
else
```

```
ANGLE_Cur = atan2(Img_Cur,Real_Cur);
```

```
ANGLE_Cur = (ANGLE_Cur*180)/pi;
```

```
ANGLE_Vint[BS_I] = 0;
```

```
TAN_1 = MAGNI_Cur / MAGNI_Ygen[BS_I] ;
```

```
TAN_2 = ANGLE_Cur - ANG_Ygen[BS_I] - ANGLE_Vint[BS_I] ;
```

```
TAN_2 = (TAN_2*pi)/180;
```

```
TAN_3 = TAN_1 * cos(TAN_2);
```

```
TAN_4 = TAN_1 * sin(TAN_2);
```

```
TAN_5 = ANGLE_V[1][BS_I] - ANGLE_Vint[BS_I];
```

```
TAN_5 = (TAN_5*pi)/180;
```

```
TAN_6 = MAGNI_V[1][BS_I] * cos(TAN_5);
```

```
TAN_7 = MAGNI_V[1][BS_I] * sin(TAN_5);
```

```
Real_Vint = TAN_6 + TAN_3 ;
```

```
IMG_Vint = TAN_7 + TAN_4 ;
```

```
MAGNI_Vint[BS_I] = sqrt((Real_Vint * Real_Vint)+(IMG_Vint* IMG_Vint));
```

```
for(PH_P=2;PH_P < 4 ;PH_P++)
```

```
{
```

```
SUM_3=SUM_4=0;
```

```
for( BS_K = 1 ;BS_K < MAX_BUS ;BS_K++)
```

```
{ //for
```

```
if( BS_K != BS_I)
```

```
{ // if
```

```
temp_1 = MAGNI_V[PH_P][BS_I] * MAGNI_Y[PH_P][BS_I][PH_P][BS_K] ;
```

```
temp_2 = MAGNI_V[PH_P][BS_K] * MAGNI_Y[PH_P][BS_I][PH_P][BS_K] ;
```

```
temp_3 = ANGLE_V[PH_P][BS_I] + ANG_Y[PH_P][BS_I][PH_P][BS_K];
```

```
temp_3 = (temp_3*pi)/180;
```

```
temp_4 = ANGLE_V[PH_P][BS_K] + ANG_Y[PH_P][BS_I][PH_P][BS_K];
```

```
temp_4 = (temp_4*pi)/180;
```

```
DAT = cos(temp_4);
```

```
temp_5 = temp_2*DAT;
```

```
DAT = sin(temp_4);
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp_6 = temp_2*DAT;
DAT = cos(temp_3);
temp_7 = temp_1*DAT;
DAT = sin(temp_3);
temp_8 = temp_1*DAT;
SUM_1 = (temp_8 - temp_6);
SUM_2 = (temp_7 - temp_5);
} // end if

SUM_3 = SUM_3 + SUM_2;
SUM_4 = SUM_4 + SUM_1;
SUM_1 = SUM_2 =0;

} //for
Real_Cur = SUM_3 ;
Img_Cur = SUM_4 ;
MAGNI_Cur = sqrt((Real_Cur*Real_Cur)+(Img_Cur*Img_Cur));
if(Real_Cur==0)
    ANGLE_Cur = 0;
else
    ANGLE_Cur = atan2(Img_Cur,Real_Cur);
    ANGLE_Cur = (ANGLE_Cur*180)/pi;
if(PH_P==2) ANGLE_Cur += 120;
    else ANGLE_Cur += 240;
DATA_1 = ANGLE_Vint[BS_I] - ANGLE_V[PH_P][BS_I];
DATA_1 = (DATA_1*pi)/180;
DATA_2 = ANGLE_Cur - ANGLE_V[PH_P][BS_I]- ANG_Ygen[BS_I] ;
DATA_2 = (DATA_2*pi)/180;
DATA_3 = MAGNI_Vint[BS_I] * cos(DATA_1) ;
DATA_4 = MAGNI_Vint[BS_I] * sin(DATA_1) ;
DATA_5 = MAGNI_Cur / MAGNI_Ygen[BS_I] ;
DATA_6 = DATA_5 * cos(DATA_2) ;
DATA_7 = DATA_5 * sin(DATA_2) ;
REAL_V = DATA_3 - DATA_6 ;
IMG_V = DATA_4 - DATA_7 ;

```

```

        PIK = 1;
    }
}
}
} //END OF FUNCTION */

```

```
void Cal_V_slack(void)
```

```

{
    int    BS_I,BS_K,s,PH_P;
    double DAT,DAT_1,DAT_2,DAT_3,DAT_4,
           temp_1,temp_2,temp_3,temp_4,temp_5,temp_6,temp_7,temp_8,
           SUM_1,SUM_2,SUM_3,SUM_4,
           TAN_1,TAN_2,TAN_3,TAN_4,TAN_5,TAN_6,TAN_7,
           DATA_1,DATA_2,DATA_3,DATA_4,
           DATA_5,DATA_6,DATA_7,DATA_8 ;
    double Real_Cur,Img_Cur,MAGNI_Cur,ANGLE_Cur,REAL_V,IMG_V;
    double Real_Vint,IMG_Vint ;

    BS_I = slack_bus;

    SUM_3=SUM_4=0;
    for( BS_K = 1 ;BS_K < MAX_BUS ;BS_K++)
    { //for
        if( BS_K != BS_I)
        { // if
            temp_1 = MAGNI_V[1][BS_I] * MAGNI_Y[1][BS_I][1][BS_K] ;
            temp_2 = MAGNI_V[1][BS_K] * MAGNI_Y[1][BS_I][1][BS_K] ;
            temp_3 = ANGLE_V[1][BS_I] + ANG_Y[1][BS_I][1][BS_K];
            temp_3 = (temp_3*pi)/180;
            temp_4 = ANGLE_V[1][BS_K] + ANG_Y[1][BS_I][1][BS_K];
            temp_4 = (temp_4*pi)/180;
            DAT = cos(temp_4);
            temp_5 = temp_2*DAT;
            DAT = sin(temp_4);

```

```

temp_6 = temp_2*DAT;
DAT = cos(temp_3);
temp_7 = temp_1*DAT;
DAT = sin(temp_3);
temp_8 = temp_1*DAT;
SUM_1 = (temp_8 - temp_6);
SUM_2 = (temp_7 - temp_5);
}
SUM_3 = SUM_3 + SUM_2;
SUM_4 = SUM_4 + SUM_1;
SUM_1 = SUM_2 = 0;
}

```

```

Real_Cur = SUM_3 ;
Img_Cur = SUM_4 ;
MAGNI_Cur = sqrt((Real_Cur*Real_Cur)+(Img_Cur*Img_Cur));
if(Real_Cur==0)
    ANGLE_Cur = 0;
else
    ANGLE_Cur = atan2(Img_Cur,Real_Cur);
    ANGLE_Cur = (ANGLE_Cur*180)/pi;
TAN_1 = MAGNI_Cur / MAGNI_Ygen[BS_I] ;
TAN_2 = ANGLE_Cur - ANG_Ygen[BS_I] ;
TAN_2 = (TAN_2*pi)/180;
TAN_3 = TAN_1 * cos(TAN_2);
TAN_4 = TAN_1 * sin(TAN_2);
TAN_5 = ANGLE_V[1][BS_I] ;
TAN_5 = (TAN_5*pi)/180;
TAN_6 = MAGNI_V[1][BS_I] * cos(TAN_5);
TAN_7 = MAGNI_V[1][BS_I] * sin(TAN_5);
Real_Vint = TAN_6 + TAN_3 ;
IMG_Vint = TAN_7 + TAN_4 ;
MAGNI_Vint[BS_I] = sqrt((Real_Vint * Real_Vint)+(IMG_Vint* IMG_Vint));
ANGLE_Vint[BS_I] = atan2(IMG_Vint,Real_Vint);

```

```

for(PH_P=2;PH_P < 4 :PH_P++)
{
SUM_3=SUM_4=0;
for( BS_K = 1 ;BS_K < MAX_BUS ;BS_K++)
{ //for
if( BS_K != BS_1)
{ // if
temp_1 = MAGNI_V[PH_P][BS_I] * MAGNI_Y[PH_P][BS_I][PH_P][BS_K] ;
temp_2 = MAGNI_V[PH_P][BS_K] * MAGNI_Y[PH_P][BS_I][PH_P][BS_K] ;
temp_3 = ANGLE_V[PH_P][BS_I] + ANG_Y[PH_P][BS_I][PH_P][BS_K];
temp_3 = (temp_3*pi)/180;
temp_4 = ANGLE_V[PH_P][BS_K] + ANG_Y[PH_P][BS_I][PH_P][BS_K];
temp_4 = (temp_4*pi)/180;
DAT = cos(temp_4);
temp_5 = temp_2*DAT;
DAT = sin(temp_4);
temp_6 = temp_2*DAT;
DAT = cos(temp_3);
temp_7 = temp_1*DAT;
DAT = sin(temp_3);
temp_8 = temp_1*DAT;
SUM_1 = (temp_8 - temp_6);
SUM_2 = (temp_7 - temp_5);
}
SUM_3 = SUM_3 + SUM_2;
SUM_4 = SUM_4 + SUM_1;
SUM_1 = SUM_2 =0;
}
}
Real_Cur = SUM_3 ;
Img_Cur = SUM_4 ;
MAGNI_Cur = sqrt((Real_Cur*Real_Cur)+(Img_Cur*Img_Cur));
if(Real_Cur==0)
ANGIE_Cur = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ANGLE_Cur = atan2(Img_Cur,Real_Cur);
```

```
ANGLE_Cur = (ANGLE_Cur*180)/pi;
```

```
DATA_1 = ANGLE_Vint[BS_I] - ANGLE_V[PH_P][BS_I];
```

```
DATA_1 = (DATA_1*pi)/180;
```

```
DATA_2 = ANGLE_Cur - ANGLE_V[PH_P][BS_I] - ANG_Ygen[BS_I];
```

```
DATA_2 = (DATA_2*pi)/180;
```

```
DATA_3 = MAGNI_Vint[BS_I] * cos(DATA_1);
```

```
DATA_4 = MAGNI_Vint[BS_I] * sin(DATA_1);
```

```
DATA_5 = MAGNI_Cur / MAGNI_Ygen[BS_I];
```

```
DATA_6 = DATA_5 * cos(DATA_2);
```

```
DATA_7 = DATA_5 * sin(DATA_2);
```

```
REAL_V = DATA_3 - DATA_6;
```

```
IMG_V = DATA_4 - DATA_7;
```

```
MAGNI_V[PH_P][BS_I] = sqrt((REAL_V*REAL_V)+(IMG_V*IMG_V));
```

```
ANGLE_V[PH_P][BS_I] = atan2(IMG_V,REAL_V);
```

```
ANGLE_V[PH_P][BS_I] = -(ANGLE_V[PH_P][BS_I]*180)/pi;
```

```
}
```

```
}
```

```
void Cal_Flow_PQ_1(void)
```

```
/* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
```

```
{
```

```
int p, q, k, Y, d, check_pv;
```

```
double e[SP_MAX_BUS], f[SP_MAX_BUS];
```

```
double REAL_1, IMG_1, TEMP_DATA;
```

```
/*----- Cal e_f */
```

```
for(p = 1; p < MAX_BUS; p++)
```

```
{ TEMP_DATA = (ANGLE_V[1][p] * pi) / 180;
```

```
e[p] = MAGNI_V[1][p] * cos(TEMP_DATA);
```

```
f[p] = MAGNI_V[1][p] * sin(TEMP_DATA);
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*----- Cal_PQ */
for(p = 1 ; p < MAX_BUS ; p++)
{ for(q = 1 ; q < MAX_BUS ; q++)
  { if(q != p)
    {
      REAL_1 = e[p] * ( e[p] - e[q] ) + f[p] * ( f[p] - f[q] ) ;
      IMG_1 = c[p] * ( f[p] - f[q] ) - f[p] * ( e[p] - e[q] ) ;

      /* Ep^(Ep-Eq) = REAL_1 + jIMG_1 */

      TEMP_DATA = REAL_1 * Y_G[1][p][1][q] ;
      FLOW_P[1][p][1][q] = TEMP_DATA - ( IMG_1 * Y_B[1][p][1][q] ) ;
      FLOW_P[1][p][1][q] = -1*(BASE * FLOW_P[1][p][1][q]) ;

      TEMP_DATA = REAL_1 * Y_B[1][p][1][q] ;
      TEMP_DATA = TEMP_DATA + ( IMG_1 * Y_G[1][p][1][q] ) ;

      FLOW_Q[1][p][1][q] = -1 * TEMP_DATA ;
      Img_Ypq2[1][p][1][q] = 0;
      TEMP_DATA = Img_Ypq2[1][p][1][q] * (e[p]*e[p] + f[p]*f[p]) ;
      FLOW_Q[1][p][1][q] = FLOW_Q[1][p][1][q] - TEMP_DATA ;
      FLOW_Q[1][p][1][q] = -1*(BASE * FLOW_Q[1][p][1][q]) ;
    }
  }
}

```

```

/*----- Cal G_PQ SLACK BUS */

```

```

for(k = 1 , REAL_1 = 0 , IMG_1 = 0 ; k < MAX_BUS ; k++)
{ if( k != slack_bus )
  { REAL_1 = REAL_1 + FLOW_P[1][slack_bus][1][k] ;
    G_P[1][slack_bus] = REAL_1 ;
    IMG_1 = IMG_1 + FLOW_Q[1][slack_bus][1][k] ;
    G_Q[1][slack_bus] = IMG_1 ;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 G\_P[1][slack\_bus] = G\_P[1][slack\_bus] + L\_P[1][slack\_bus] ;  
 ไม่สามารถใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
G_Q[1][slack_bus] = G_Q[1][slack_bus] + L_Q[1][slack_bus] ;
```

```
/*----- Cal G_Q PV BUS */
```

```
for(k = 1 , TEMP_DATA = 0 ; k < MAX_BUS ; k++)
```

```
{
```

```
    check_pv = 1 ;
```

```
    for(d = 1 ; d < (pv+1) ; d++)
```

```
        if( v_control[d] == k )
```

```
            check_pv = 0;
```

```
    if( check_pv == 0 )
```

```
    { for(q = 1 ; q < MAX_BUS ; q++)
```

```
        { if(q != k)
```

```
            TEMP_DATA = TEMP_DATA + FLOW_Q[1][k][1][q] ;
```

```
        ;
```

```
    } 
```

```
    G_Q[1][k] = TEMP_DATA + L_Q[1][k] ;
```

```
    }
```

```
}
```

```
}
```

```
/*----- End of Cal_PQ -----*/
```

```
void Stable_of_sys(void)
```

```
{
```

```
    int i , k , d ;
```

```
    double DAT;
```

```
    BAL = 0;
```

```
    PIK = 1;
```

```
    for(k = 1 ; k < MAX_BUS ; k++)
```

```
        if(k!=slack_bus)
```

```
        {
```

```
            check_pv = 1 ;
```

```
            for(d = 1 ; d < (pv+1) ; d++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( v_control[d] == k )
    check_pv = 0;

if(check_pv==0)
{
    for(i=2;i<=3;i++)
    {
        DAT = sqrt(pow(MAGNI_V[1][k]-MAGNI_V[i][k],2));
        if(DAT>(Vpv_LEVEL*MAGNI_V[1][k]))
            PIK = 0;
    }
    if((L_P[1][k]==L_P[2][k])&&(L_P[2][k]==L_P[3][k])&&(L_P[3][k]==L_P[1][k])
        &&(L_Q[1][k]==L_Q[2][k])&&(L_Q[2][k]==L_Q[3][k])&&(L_Q[3][k]==L_Q[1][k]))
        BAL += 5;
}
else
{
    for(i=1;i<=3;i++)
    {
        DAT = sqrt(pow(L-MAGNI_V[i][k],2));
        if(DAT>VloadLEVEL)
            PIK = 0;
    }
    if((L_P[1][k]==L_P[2][k])&&(L_P[2][k]==L_P[3][k])&&(L_P[3][k]==L_P[1][k])
        &&(L_Q[1][k]==L_Q[2][k])&&(L_Q[2][k]==L_Q[3][k])&&(L_Q[3][k]==L_Q[1][k]))
        BAL += 5;
}
}
if(BAL==(5*(MAX_BUS-2)))
{
    for(i=1;i<MAX_BUS;i++)
        for(k=2;k<=3;k++)
        {
            MAGNI_V[k][i] = MAGNI_V[1][i];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ANGLE\_V[k][i] = ANGLE\_V[1][i]-120;  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    ANGLE_V[k][i] = ANGLE_V[1][i]+120;
}
}

void Make_Table_1(void)
{
    int p , q , k , BUS;
    int y_pos;          /* position in y-axis */

    complete_cal_indicate();
    outtext3d(60,110,"PHASE A",0,0,14,6);
    button(100,130,550,157,2,0);
    outtext3d(120,140,"BUS          VOLTAGE_MAGNITUDE          PHASE_ANGLE",0,0,14,6);

    for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
    {
        int_to_string(BUS);
        strcpy(text,string);
        outtext3d(130,180+((BUS-1)*25),text,0,0,14,6);
        space(217,171+((BUS-1)*25),380,192+((BUS-1)*25));
        space(410,171+((BUS-1)*25),540,192+((BUS-1)*25));
        display_result_double(225,178+((BUS-1)*25),MAGNI_V[1][BUS],10,0,0,15,8);
        display_result_double(418,178+((BUS-1)*25),ANGLE_V[1][BUS],10,0,0,15,8);
    }

    getch();
    setfillstyle(1,3);
    bar(20,125,615,317);
    button(25,130,615,157,2,0);
    outtext3d(30,140," BUS   G[MW]   G[MVAR]   L[MW MVAR]   ToBus   MW
MVAR",0,0,14,6) ;
    button(25,170,615,310,2,1);
    setfillstyle(1,0);
    bar(28,173,612,307);

    for(p = 1 ; p < MAX_BUS ; p++)
        int_to_string(p);
}

```

```

y_pos=180+((p-1)*25);
outtext3d(46,y_pos,string,0,0,14,6);
display_result_double(78,y_pos,G_P[1][p],6,0,0,15,8);
display_result_double(168,y_pos,G_Q[1][p],6,0,0,15,8);

display_result_double(151+14*8,y_pos,L_P[1][p],1,0,0,15,8);
display_result_double(150+14*8+56,y_pos,L_Q[1][p],1,0,0,15,8);

for(q = 1 ; q < MAX_BUS ; q++)
{
    if(q != p)
    {
        int_to_string(q);
        outtext3d(158+14*8+56+72,y_pos,string,0,0,15,8);
        display_result_double(158+14*8+56+112,y_pos,FLOW_P[1][p][1][q],6,0,0,15,8);
        display_result_double(150+14*8+72+192,y_pos,FLOW_Q[1][p][1][q],6,0,0,15,8);
        y_pos+=12 ;
    }
}
}
getch();
}
/*----- End of Make_Table -----*/

```

```
void Make_Table_2(void)
```

```

{
    int p , q , k , BUS;
    int y_pos;          /* position in y-axis */

    setfillstyle(1,3);
    bar(15,110,625,315);
    outtext3d(60,110,"PHASE B",0,0,14,6);
    button(100,130,550,157,2,0);
    outtext3d(120,140,"BUS          VOLTAGE_MAGNITUDE          PHASE_ANGLE",0,0,14,6);

```

```
for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
```

```
{ int_to_string(BUS);
```

```

strcpy(text,string);
outtext3d(130,180+((BUS-1)*25),text,0,0,14,6);
space(217,171+((BUS-1)*25),380,192+((BUS-1)*25));
space(410,171+((BUS-1)*25),540,192+((BUS-1)*25));
display_result_double(225,178+((BUS-1)*25),MAGNI_V[2][BUS],10,0,0,15,8);
display_result_double(418,178+((BUS-1)*25),ANGLE_V[2][BUS],10,0,0,15,8);
}

getch();
setfillstyle(1,3);
bar(20,125,615,317);
button(25,130,615,157,2,0);
outtext3d(30,140," BUS   G[MW]   G[MVAR]   L[MW  MVAR]   ToBus   MW
MVAR",0,0,14,6) ;
button(25,170,615,310,2,1);
setfillstyle(1,0);
bar(28,173,612,307);
for(p = 1 ; p < MAX_BUS ; p++)
{
  int_to_string(p);
  y_pos=180+((p-1)*25);
  outtext3d(46,y_pos,string,0,0,14,6);
  display_result_double(78,y_pos,G_P[2][p],6,0,0,15,8);
  display_result_double(168,y_pos,G_Q[2][p],6,0,0,15,8);

  display_result_double(151+14*8,y_pos,L_P[2][p],1,0,0,15,8);
  display_result_double(150+14*8+56,y_pos,L_Q[2][p],1,0,0,15,8);

  for(q = 1 ; q < MAX_BUS ; q++)
  {
    if(q != p)
    {
      int_to_string(q);
      outtext3d(158+14*8+56+72,y_pos,string,0,0,15,8);
      display_result_double(158+14*8+56+112,y_pos,FLOW_P[2][p][2][q],6,0,0,15,8);
      display_result_double(150+14*8+72+192,y_pos,FLOW_Q[2][p][2][q],6,0,0,15,8);
      y_pos+=12 ;
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
getch();
}
/*----- End of Make_Table -----*/

void Make_Table_3(void)
{
int p , q , k , BUS;
int y_pos;          /* position in y-axis */

setfillstyle(1,3);
bar(15,110,625,315);
outtext3d(60,110,"PHASE C",0,0,14,6);
button(100,130,550,157,2,0);
outtext3d(120,140,"BUS          VOLTAGE_MAGNITUDE          PHASE_ANGLE",0,0,14,6);

for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
{
int_to_string(BUS);
strcpy(text,string);
outtext3d(130,180+((BUS-1)*25),text,0,0,14,6);
space(217,171+((BUS-1)*25),380,192+((BUS-1)*25));
space(410,171+((BUS-1)*25),540,192+((BUS-1)*25));
display_result_double(225,178+((BUS-1)*25),MAGNI_V[3][BUS],10,0,0,15,8);
display_result_double(418,178+((BUS-1)*25),ANGLE_V[3][BUS],10,0,0,15,8);
}

getch();
setfillstyle(1,3);
bar(20,125,615,317);
button(25,130,615,157,2,0);
outtext3d(30,140," BUS   G[MW]   G[MVAR]   L[MW MVAR]   ToBus   MW
MVAR",0,0,14,6) ;
button(25,170,615,310,2,1);
setfillstyle(1,0);
bar(28,173,612,307);
for(p = 1 ; p < MAX_BUS ; p++)
{
int_to_string(p);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y_pos=180+((p-1)*25);
outtext3d(46,y_pos,string,0,0,14,6);
display_result_double(78,y_pos,G_P[3][p],6,0,0,15,8);
display_result_double(168,y_pos,G_Q[3][p],6,0,0,15,8);

display_result_double(151+14*8,y_pos,L_P[3][p],1,0,0,15,8);
display_result_double(150+14*8+56,y_pos,L_Q[3][p],1,0,0,15,8);

for(q = 1 ; q < MAX_BUS ; q++)
{ if(q != p)

int_to_string(q);
outtext3d(158+14*8+56+72,y_pos,string,0,0,15,8);
display_result_double(158+14*8+56+112,y_pos,FLOW_P[3][p][3][q],6,0,0,15,8);
display_result_double(150+14*8+72+192,y_pos,FLOW_Q[3][p][3][q],6,0,0,15,8);
y_pos+=12 ;
}
}
}
getch();
}
/*----- End of Make_Table -----*/

void graphic_initialize(void)
{
int GraphDriver;           // The Graphics device driver
int GraphMode;             // The Graphics mode value
int ErrorCode;             // Reports any graphics errors
int xasp, yasp;            // Used to read the aspect ratio
GraphDriver = DETECT;      // Request auto-detection
initgraph( &GraphDriver, &GraphMode, "" );
ErrorCode = graphresult(); // Read result of initialization
if( ErrorCode != grOk ){    // Error occurred during init
printf(" Graphics System Error: %s\n", grapherrormsg( ErrorCode ) );
exit(1);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void BackGround(int color)
```

```
// set backgound of screen
```

```
{  
    setfillstyle(SOLID_FILL,color);  
    setcolor(color);  
    bar(0,0,640,480);  
}
```

```
void Win(int x1,int y1,int x2,int y2)
```

```
// make window frame include title bar text, all buttons
```

```
{  
    int i;  
    int X1,Y1,X2,Y2;  
    // make frame  
    setlinestyle(0,0,1);  
    setcolor(BLACK);  
    for(i=0;i<=frame_thick;i+=frame_thick)  
    {  
        line(x1+i,y1+i,x2-i,y1+i);  
        line(x1+i,y1+i,x1+i,y2-i);  
        line(x1+i,y2-i,x2-i,y2-i);  
        line(x2-i,y2-i,x2-i,y1+i);  
    }  
    // make title bar  
    setfillstyle(SOLID_FILL,BLUE);  
    bar(x1+frame_thick+1,y1+frame_thick+1,x2-frame_thick-1,y1+frame_thick+menu_bar_thick);  
    // set window area color  
    setfillstyle(SOLID_FILL,WHITE);  
    bar(x1+frame_thick+1,y1+frame_thick+menu_bar_thick,x2-frame_thick-1,y2-frame_thick-1);  
}
```

```
void button(int x1,int y1,int x2,int y2,int thick,int push)
```

```
{  
    struct PTS {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
int x,y;  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} poly[7];
int color,X1,Y1;

// make button's highlight or vise versa
if (push==1) color=8;
else color=15;
setfillstyle(SOLID_FILL,color);
bar(x1,y1,x2,y2);
// make button's shadow or vise versa
poly[0].x = x1;
poly[0].y = y2;
poly[1].x = x1+thick;
poly[1].y = y2-thick;
poly[2].x = x2-thick;
poly[2].y = y2-thick;
poly[3].x = x2-thick;
poly[3].y = y1+thick;
poly[4].x = x2;
poly[4].y = y1;
poly[5].x = x2;
poly[5].y = y2;
if(push==1) color = 15;
else color=8;
setfillstyle(SOLID_FILL,color);
fillpoly(6,(int far*)poly);
// make betton's surface
setfillstyle(SOLID_FILL,7);
bar(x1+thick,y1+thick,x2-thick,y2-thick);
// make button's frame
setcolor(BLACK);
setlinestyle(0,0,1);
rectangle(x1,y1,x2,y2);
}

```

```
void frame(int x1,int y1,int x2,int y2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i;
setlinestyle(0,0,1);
for(i=0;i<=1;i++)
{
    setcolor(8+(i*3));
    line(x1+i,y1+i,x2+i,y1+i);
    line(x2+i,y1+i,x2+i,y2+i);
    line(x2+i,y2+i,x1+i,y2+i);
    line(x1+i,y2+i,x1+i,y1+i);
}
}

void space(int x1,int y1,int x2,int y2)
{
    button(x1,y1,x2,y2,2,1);
    setfillstyle(1,0);
    bar(x1+3,y1+3,x2-3,y2-3);
}

void outtext3d(int x,int y,char str[],int font,int font_size,int fore_color,int bk_color)
{
    settextstyle(font,0,font_size);
    setcolor(bk_color);
    outtextxy(x+1,y+1,str);
    setcolor(fore_color);
    outtextxy(x,y,str);
}

void beep(void)
{
    sound(1000);
    delay(50);
    nosound();
}

```

**void dialog\_box(int x1,int y1,int x2,int y2,int fillin,int fillcolor)** ซ้ำเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    frame(x1,y1,x2,y2);
    button(x1+5,y1+5,x2-5,y2-5,2,1);
    if(fillin==1)
    { setfillstyle(1,fillcolor);
      bar(x1+8,y1+8,x2-8,y2-8);
    }
}

```

```

void clear_in_win(void)

```

```

--{
    setfillstyle(SOLID_FILL,3);
    bar(5,26,635,475);
}

```

```

double str_to_real(char *string)

```

```

{
    double num=0,j;
    int i=0,isfloat=0,less_than_one=0,point_num=0,negative=0,negative_num=0;

    while (string[i]!='\0')
    { if (string[i]=='.')
      { isfloat = 1;
        point_num++;
      }

      if (!(((string[i]>='0')&&(string[i]<='9'))||(string[i]=='.')||(string[i]=='-'))))
      {
          printf("\007Error in inout data");
          exit(1);
      }

      if (string[i]=='-')
          negative_num++;

      i++;
    }
}

```

```

{
    negative=1;
    j=0;
    while(string[j]!='\0')
    {
        string[j]=string[j+1];
        j++;
    }
}

// If there are more than one points , exit
if ((point_num>1)|| (negative_num>1))
{
    clrscr();
    printf("\007Error in input data !!!");
    exit(1);
}

while ((string[0]!='0')&&(string[1]!='0'))
{
    j=1;
    while(string[j]!='\0')
    {
        string[j]=string[j+1];
        j++;
    }
}

if ((string[0]!='.')|| ((string[0]!='0')&&(string[1]!='.')))
    less_than_one=1;

if ((isfloat==1)&&(less_than_one==0))
{
    i=0;
    while (string[i]!='.')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

num=(num*10)+string[i]-'0';
i++;
}
j=0.1;
i++;
while (string[i]!='\0')
{
num=num+((string[i]-'0')*j);
i++;
j=j/10;
}
}
if(isfloat==0)
{
i=0;
while (string[i]!='\0')
{
num=(num*10)+string[i]-'0';
i++;
}
}
if ((isfloat==1)&&(less_than_one==1))
{
if(string[0]=='0')
{
j=0;
while(string[j]!='\0')
{
string[j]=string[j+1];
j++;
}
}
i=1;
j=0.1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (string[i]!='\0')
{
    num=num+((string[i]-'0')*j);
    i++;
    j=j/10;
}
}

```

```

if(negative==1) num=-num;

```

```

return(num);
}

```

```

int str_to_int(char *string)
{

```

```

    int num=0;

```

```

    int i=0,negative=0,negative_num=0;

```

```

    while (string[i]!='\0')
    {

```

```

        if (!(((string[i]>='0')&&(string[i]<='9')) || (string[i]=='-'))))
        {

```

```

            printf("\007Error in inout data");

```

```

            exit(1);
        }

```

```

        if (string[i]=='-')

```

```

            negative_num++;

```

```

        i++;
    }
}

```

```

if (string[0]=='-')
{

```

```

    negative=1;

```

```

    i=0;

```

```

    while(string[i]!='\0')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        string[i]=string[i+1];
        i++;
    }
}

i=0;
while (string[i]!='\0')
{
    num=(num*10)+string[i]-'0';
    i++;
}

if(negative==1) num=-num;

return(num);
}

```

```

void int_to_string(int num)
// not more than 9999
{
    int n;
    int i=0,j,w;

    while(num>=1)
    {
        n=1;
        if(num>=10) n=10;
        if(num>=100) n=100;
        if(num>=1000) n=1000;
        j=0;
        while(num>=n)
        {
            num=num/n;
            j++;
        }
        if((num<1)&&(n==10))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        string[++i]='0';
        i++;
    }
    else if((num<1)&&(n==100))
    { string[i]=j+'0';
        string[++i]='0';
        string[++i]='0';
        i++;
    }
    else if((num<1)&&(n==1000))
    { string[i]=j+'0';
        string[++i]='0';
        string[++i]='0';
        string[++i]='0';
        i++;
    }
    else if((num>1)&&(num<10)&&(n==100))
    { string[i]=j+'0';
        string[++i]='0';
        i++;
    }
    else if((num>10)&&(num<100)&&(n==1000))
    { string[i]=j+'0';
        string[++i]='0';
        i++;
    }
    else if((num>1)&&(num<10)&&(n==1000))
    { string[i]=j+'0';
        string[++i]='0';
        string[++i]='0';
        i++;
    }
    else
    { string[i]=j+'0';
        i++;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    string[i]='\0';
}

void double_to_text(double num,int point)
{

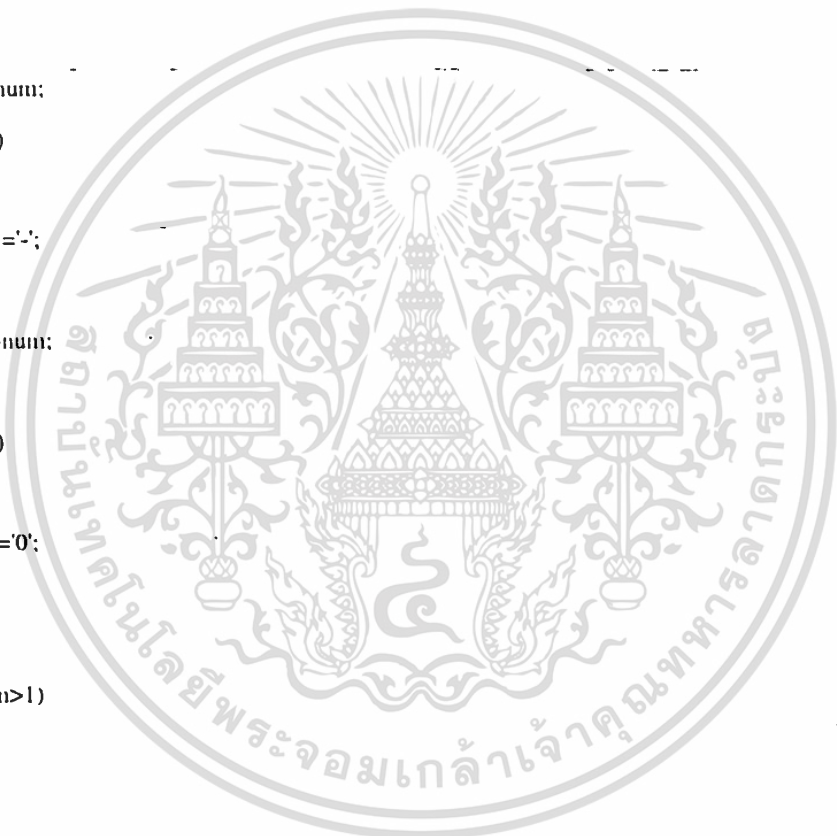
```

```

    double n;
    double chkzero;
    int i=0,j,w;

    chkzero=num;
    if(num<0)
    {
        text[0]='-';
        i++;
        num=-num;
    }
    if(num<1)
    {
        text[i]='0';
        i++;
    }
    while(num>1)
    {
        n=1;
        if(num>=10) n=10;
        if(num>=100) n=100;
        if(num>=1000) n=1000;
        j=0;
        while(num>=n)
        {
            num=num-n;
            j++;
        }
        if((num<1)&&(n==10))
        {
            text[i]=j+'0';
            text[++i]='0';

```



```

        i++;
    }
    else if((num<1)&&(n==100))
    { text[i]=j+'0';
      text[++i]='0';
      text[++i]='0';
      i++;
    }
    else if((num<1)&&(n==1000))
    { text[i]=j+'0';
      text[++i]='0';
      text[++i]='0';
      text[++i]='0';
      i++;
    }
    else if((num>1)&&(num<10)&&(n==100))
    { text[i]=j+'0';
      text[++i]='0';
      i++;
    }
    else if((num>10)&&(num<100)&&(n==1000))
    { text[i]=j+'0';
      text[++i]='0';
      i++;
    }
    else if((num>1)&&(num<10)&&(n==1000))
    { text[i]=j+'0';
      text[++i]='0';
      text[++i]='0';
      i++;
    }
    }
    else
    { text[i]=j+'0';
      i++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

text[i++]='.';
n=0.1;
if(num!=1)
{
for(w=0;w<=point;w++)
{
if(num<n)
{ text[i]='0';
i++;
num=num*10;
continue;
}
j=0;
while(num>n)
{ num=num-n;
j++;
}
text[i]=j+'0';
n=n/10;
i++;
}
// if(text[i-1]>='5') text[i-2]=text[i-2]+1
}
text[i-1]='\0';
if(chkzero==0)
{
strcpy(text,"0.");
for(i=2;i<=point+1;i++)
text[i]='0';
text[i]='\0';
}
if(num==1)
{
strcpy(text,"1.");
for(i=2;i<=point+1;i++)
text[i]='0';
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        text[i]='\0';
    }
}

void about_dialog(void)
{
    button(10,350,629,470,4,0);
    settextstyle(1,0,5);
    outtext3d(160,355,"3 PHASE LOAD FLOW",1,4,15,0);
    outtext3d(310,400,"by",0,0,15,0);
    outtext3d(153,420,"Kitichai Kiatlatchanon    34101025",0,0,15,0);
    outtext3d(153,435,"Pranithi Theanvanichpan    34104201",0,0,15,0);
    outtext3d(153,450,"Poolsak Toungsirikulwattana    34105248",0,0,15,0);
}

void complete_cal_indicate(void)
{
    setfillstyle(1,3);
    bar(15,45,625,300);
    outtext3d(230,160,"Complete ...",1,4,14,6);
    sound(900);delay(100);
    sound(1152);delay(100);
    sound(1358);delay(100);
    nosound();
    bar(15,45,625,300);
    outtext3d(270,60,"RESULT",1,2,12,0);
}

void display_result_double(int x,int y,double num,int point,int font,int font_size,int fore_color,int bk_color)
{
    if(num>0)
    { x+=8;
      if(num<10) x+=8;
    }
    if(num==0)

```

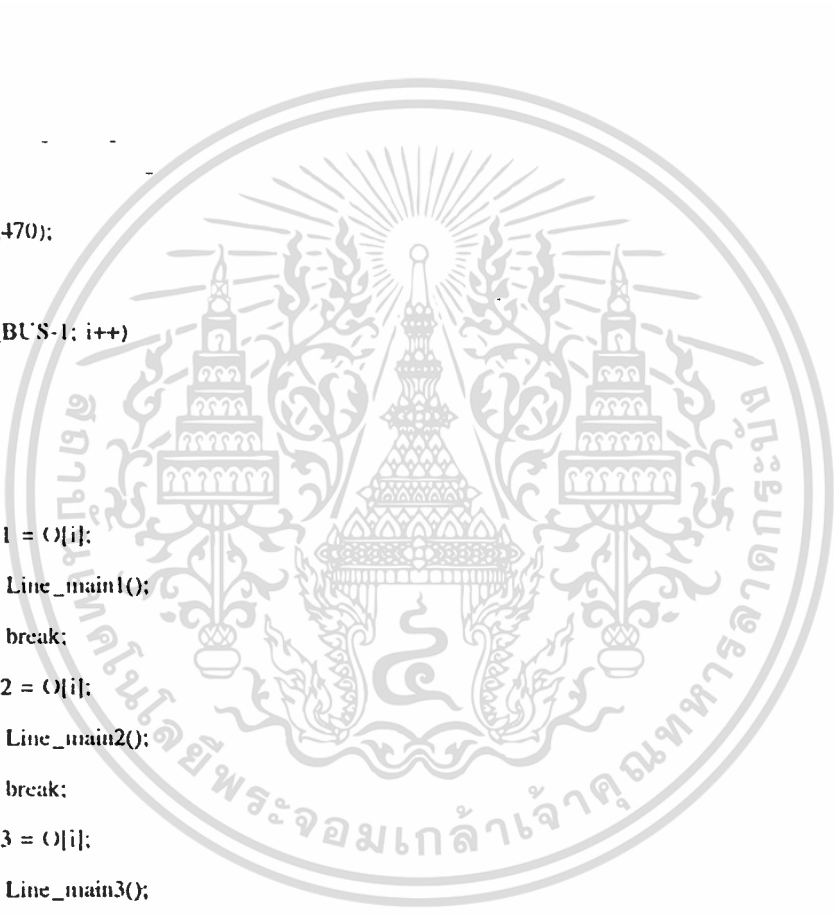
เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
    }  
    }  
    }  
    }  
}
```

void Setgraph(void)

```
{  
    int i,j,y_pos;  
  
    clear_in_win();  
    frame(10,40,629,470);  
  
    for(i=0;i<MAX_BUS-1; i++)  
    {  
        switch(i)  
        {  
            case 0: n1 = O[i];  
                Line_main1();  
                break;  
            case 1: n2 = O[i];  
                Line_main2();  
                break;  
            case 2: n3 = O[i];  
                Line_main3();  
                break;  
            case 3: n4 = O[i];  
                Line_main4();  
                break;  
            case 4: n5 = O[i];  
                Line_main5();  
                break;  
        }  
    }  
}
```



เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=1;i<MAX_BUS; i++)
{
    if(i==slack_bus)
    {
        int_to_string(i);
        strcat(string, " SLACK BUS");
        outtext3d(30,y_pos,string,0,0,15,0);
    }
    else
    {
        check_pv=1;
        for(j=1;j<=pv;j++)
            if(v_control[j]==i) check_pv = 0;
        if(check_pv==0)
        {
            int_to_string(i);
            strcat(string, " VOLTAGE CONTROL BUS");
            outtext3d(30,y_pos,string,0,0,15,0);
        }
        else
        {
            int_to_string(i);
            strcat(string, " LOAD BUS");
            outtext3d(30,y_pos,string,0,0,15,0);
        }
    }
    y_pos+=10;
}
getch();
// closegraph();
}

```

**/\* Begin Draw Line Between Bus \*/**

**void Line\_main1(void)**

**{**

**int a,b;**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int_to_string(O[0]);
outtext3d(x[1]-55,y[1]+10,string,0,0,14,6);
setcolor(15);
if (Relat_G[n1]== 'Y')
{
    setlinestyle(SOLID_LINE,1,2);
    circle(x[1],y[1],r);
    setlinestyle(SOLID_LINE,1,3);
    line(x[1],y[1]+r,x[1],y[1]+r+10);
}
setlinestyle(SOLID_LINE,1,3);
line(x[1]-span,y[1]+r+10,x[1]+span,y[1]+r+10);
line(x[1]-(span*2/3),y[1]+r,x[1]-(span*2/3),y[1]+r+10);

if(Relat_L[n1] == 'Y')
{
    setlinestyle(SOLID_LINE,1,2);
    a = x[1]-(span*2/3);
    b = y[1]+r
    line(a-3,b,a+3,b);
    line(a+3,b,a,b-9);
    line(a-3,b,a,b-9);
}
}

```

```
void Line_main2(void)
```

```

{
    int a,b;

    int_to_string(O[1]);
    outtext3d(x[2]+50,y[2]+10,string,0,0,14,6);
    setcolor(15);
    if(Relat_G[n2] == 'Y')
    {
        setlinestyle(SOLID_LINE,1,2);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

circle(x[2],y[2],r);
setlinestyle(SOLID_LINE,1,3);
line(x[2],y[2]+r,x[2],y[2]+r+10);
}

setlinestyle(SOLID_LINE,1,3);
line(x[2]-span,y[2]+r+10,x[2]+span,y[2]+r+10);
line(x[2]+(span*2/3),y[2]+r,x[2]+(span*2/3),y[2]+r+10);

if(Relat_L[n2] == 'Y')
{
setlinestyle(SOLID_LINE,1,2);
a = x[2]+(span*2/3);
b = y[2]+r;
line(a-3,b,a+3,b);
line(a+3,b,a,b-9);
line(a-3,b,a,b-9);
setlinestyle(SOLID_LINE,1,3);
}

if (Relat[n1][n2] == 'Y')
{
outtextcomplex(Y_G[1][n1][1][n2],Y_B[1][n1][1][n2],x[1]+65,y[1]-2); //display self G+j
setlinestyle(SOLID_LINE,1,3);
setcolor(15);
line(x[1]+(span*2/3),y[1]+r,x[2]-(span*2/3),y[2]+r);
line(x[1]+(span*2/3),y[1]+r,x[1]+(span*2/3),y[1]+r+10);
line(x[2]-(span*2/3),y[2]+r,x[2]-(span*2/3),y[2]+r+10);
}
}

```

```
void Line_main3(void)
```

```
{
int a,b;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int_to_string(O[2]);
outtext3d(x[4],y[4]+18,string(0,0,14,6));

if(Relat_G[n3] == 'Y')
{
    setcolor(15);
    setlinestyle(SOLID_LINE,1,2);
    circle(x[4]+(span*2/3),y[4],r);
    setlinestyle(SOLID_LINE,1,3);
    line(x[4]+(span*2/3),y[4]-r,x[4]+(span*2/3),y[4]-r-10);
}

```

```

// Lines over/under circle
setlinestyle(SOLID_LINE,1,3);
line(x[4]-span,y[4]-r-10,x[4]+span,y[4]-r-10);
line(x[4]-(span*2/3),y[4]-r,x[4]-(span*2/3),y[4]-r-10);

```

```

if(Relat_L[n3] == 'Y')
setlinestyle(SOLID_LINE,1,2);
a = x[4]-(span*2/3);
b = y[4]-r;
line(a-3,b,a+3,b);
line(a+3,b,a,b+9);
line(a-3,b,a,b+9);
setlinestyle(SOLID_LINE,1,3);

```

```

if (Relat[n1][n3] == 'Y')
{
    outtextcomplex(Y_G[1][n1][1][n3],Y_B[1][n1][1][n3],x[3]-170,y[3]); //display self G+j
    setlinestyle(SOLID_LINE,1,3);
    setcolor(15);
    line(x[1],y[1]+r+20,x[4]-(span*2/3),y[4]-r-20);
    line(x[4]-(span*2/3),y[4]-r-10,x[4]-(span*2/3),y[4]-r-20);
    line(x[1],y[1]+r+10,x[1],y[1]+r+20);
}

```

```

if (Relat[n2][n3] == 'Y')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    outtextcomplex(Y_G[1][n2][1][n3] ,Y_B[1][n2][1][n3] , x[3]+70, y[3]);
    setlinestyle(SOLID_LINE,1,3);
    setcolor(15);
    line(x[2],y[2]+r+20,x[4]+(span*2/3),y[4]-r-20);
    line(x[4]+(span*2/3),y[4]-r-10,x[4]+(span*2/3),y[4]-r-20);
    line(x[2],y[2]+r+10,x[2],y[2]+r+20);
}
}

```

```

void Line_main4(void)

```

```

{
    int a,b;

    int_to_string(O[3]);
    outtext3d(x[3]+41,y[3]+10,string,0,0,14,6);
    setcolor(15);

    if(Relat_G[n4] == 'Y')
    { setlinestyle(SOLID_LINE,1,2);
      circle(x[3],y[3],r);
      setlinestyle(SOLID_LINE,1,3);
      line(x[3],y[3]+r,x[3],y[3]+r+10);

    }

    setlinestyle(SOLID_LINE,1,3);
    line(x[3]-span,y[3]+r+10,x[3]+span,y[3]+r+10);
    line(x[3]-(span*2/3),y[3]+r+10,x[3]-(span*2/3),y[3]+r+20);

```

```

if(Relat_L[n4] == 'Y')

```

```

{
    setlinestyle(SOLID_LINE,1,2);

```

```

a = x[3]-(span*2/3) ;

```

```

b = y[3]+r+20 ;

```

```

line(a-3,b,a+3,b);

```

```
line(a+3,b,a,b+9);
```

```
line(a-3,b,a,b+9);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
}
```

```
if (Relat[n1][n4] == 'Y')
```

```
{
```

```
outtextcomplex(Y_G[1][n1][1][n4] ,Y_B[1][n1][1][n4] , x[3]-65, y[3]-120); //display self G+j
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[1]+(span*2/3),y[1]+r+20,x[3]-(span*2/3),y[3]+r);
```

```
line(x[1]+(span*2/3),y[1]+r+10,x[1]+(span*2/3),y[1]+r+20);
```

```
line(x[3]-(span*2/3),y[3]+r,x[3]-(span*2/3),y[3]+r+10);
```

```
}
```

```
if (Relat[n2][n4] == 'Y')
```

```
{
```

```
outtextcomplex(Y_G[1][n1][1][n4] ,Y_B[1][n1][1][n4] , x[3]+35, y[3]+20); //display self G+j
```

```
setcolor(15);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[2]-(span*2/3),y[2]+r+20,x[3]+(span*2/3),y[3]+r);
```

```
line(x[2]-(span*2/3),y[2]+r+10,x[2]-(span*2/3),y[2]+r+20);
```

```
line(x[3]+(span*2/3),y[3]+r,x[3]+(span*2/3),y[3]+r+10);
```

```
}
```

```
if (Relat[n3][n4] == 'Y')
```

```
{
```

```
outtextcomplex(Y_G[1][n3][1][n4] ,Y_B[1][n3][1][n4] , x[3]+20, y[3]); //display self G+j
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[3],y[3]+r+20,x[4],y[4]-r-20);
```

```
line(x[4],y[4]-r-10,x[4],y[4]-r-20);
```

```
line(x[3],y[3]+r+10,x[3],y[3]+r+20);
```

```
}
```

```
}
```

```
void Line_main5(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int a,b;
```

```
int_to_string(O[4]);
```

```
outtext3d(x[5]+50,y[5],string,0,0,14,6);
```

```
setcolor(15);
```

```
if(Relat_G[n5] == 'Y')
```

```
{ setlinestyle(SOLID_LINE,1,2);
```

```
circle(x[5],y[5],r);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[5],y[5]-r,x[5],y[5]-r-10);
```

```
}
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[5]-span,y[5]-r-10,x[5]+span,y[5]-r-10);
```

```
line(x[5]+(span*3/4),y[5]-r,x[5]+(span*3/4),y[5]-r-10);
```

```
if(Relat_L[n5] == 'Y')
```

```
{
```

```
setlinestyle(SOLID_LINE,1,2);
```

```
a = x[5]+(span*3/4);
```

```
b = y[5]-r;
```

```
line(a-3,b,a+3,b);
```

```
line(a+3,b,a,b+9);
```

```
line(a-3,b,a,b+9);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
}
```

```
if (Relat[n1][n5] == 'Y')
```

```
{
```

```
outtextcomplex(Y_G[1][n5][1][n5] .Y_B[1][n5][1][n5] , x[5]+35, y[5]-15); //display self G+j
```

```
setcolor(15);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[1]-(span*2/3),y[1]+r+20,x[5]-(span*3/4),y[5]-r-20);
```

```
line(x[5]-(span*3/4),y[5]-r-10,x[5]-(span*3/4),y[5]-r-20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
line(x[1]-(span*2/3),y[1]+r+10,x[1]-(span*2/3),y[1]+r+20);
```

```
}
```

```
if (Relat[n2][n5] == 'Y')
```

```
{
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
outtextcomplex(Y_G[1][n2][1][n5],Y_B[1][n2][1][n5],x[5]+35,y[5]-15); //display self G+j
```

```
line(x[2]+(span*2/3),y[2]+r+20,x[5]+(span*3/4),y[5]-r-10);
```

```
line(x[5]+(span*3/4),y[5]-r-10,x[5]+(span*3/4),y[5]-r-20);
```

```
line(x[2]+(span*2/3),y[2]+r+10,x[2]+(span*2/3),y[2]+r+20);
```

```
}
```

```
if (Relat[n3][n5] == 'Y')
```

```
{
```

```
outtextcomplex(Y_G[1][n3][1][n5],Y_B[1][n3][1][n5],x[3]+35,y[3]+20); //display self G+j
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[4],y[4]-r,x[5]-(span*1/4),y[5]-r-20);
```

```
line(x[5]-(span*1/4),y[5]-r-10,x[5]-(span*1/4),y[5]-r-20);
```

```
line(x[4],y[4]-r,x[4],y[4]-r-10);
```

```
}
```

```
if (Relat[n4][n5] == 'Y')
```

```
{
```

```
outtextcomplex(Y_G[1][n4][1][n5],Y_B[1][n4][1][n5],x[3]+35,y[3]+20); //display self G+j
```

```
setlinestyle(SOLID_LINE,1,3); line(x[3]+(span*2/3),y[3]+r+20,x[4]+26,y[4]-r-53); //1
```

```
line(x[4]+35,y[4]-r-45,x[5]+(span*1/4),y[4]-28); ;
```

```
line(x[5]+(span*1/4),y[4]-28,x[5]+(span*1/4),y[5]-r-20);
```

```
line(x[5]+(span*1/4),y[5]-r-10,x[5]+(span*1/4),y[5]-r-20);
```

```
line(x[3]+(span*2/3),y[3]+r+10,x[3]+(span*2/3),y[3]+r+20);
```

```
}
```

```
}
```

```
void Setgraph_PQ(void)
```

```
{
```

```
int i,j,y_pos;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clear_in_win();
frame(10,40,629,470);

for(i=0;i<MAX_BUS-1; i++)
{
    switch(i)
    {
        case 0: n1 = O[i];
                Line_main1_1();
                break;

        case 1: n2 = O[i];
                Line_main2_1();
                break;

        case 2: n3 = O[i];
                Line_main3_1();
                break;

        case 3: n4 = O[i];
                Line_main4_1();
                break;

        case 4: n5 = O[i];
                Line_main5_1();
                break;
    }
}

```

```
y_pos = 390;
```

```

for(i=1;i<MAX_BUS; i++)
{
    if(i==slack_bus)
    {
        int_to_string(i);
        strcat(string," SLACK BUS");
        outtext3d(30,y_pos,string,0,0,15,0);
    }
    else
    {
        check_pv=1;

```

```
        for(j=1;j<=pv;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

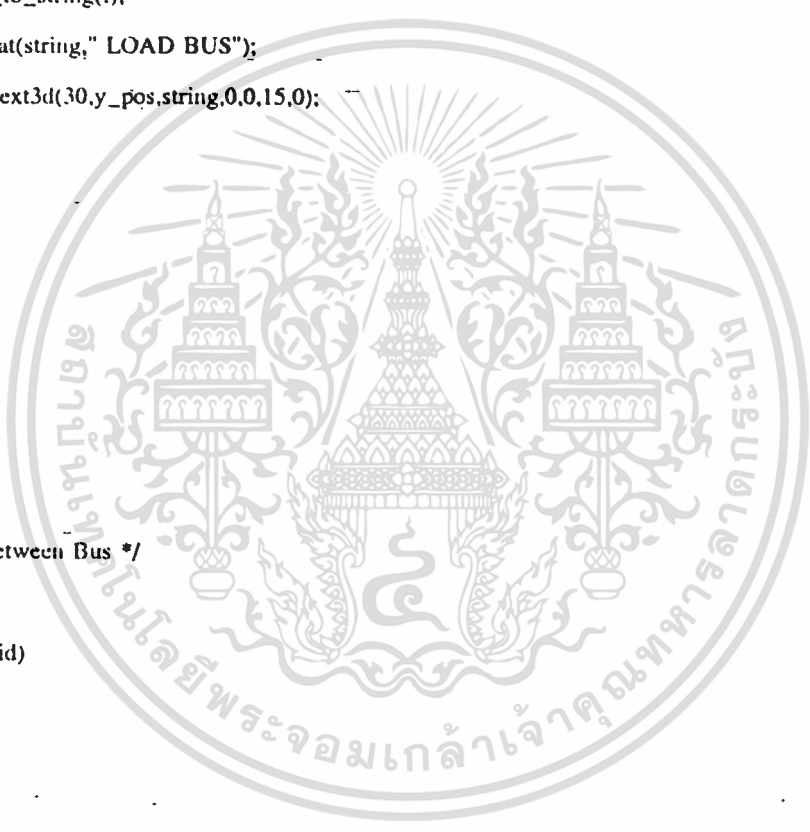
        if(v_control[j]==i) check_pv = 0;
    if(check_pv==0)
    {
        int_to_string(i);
        strcat(string," VOLTAGE CONTROL BUS");
        outtext3d(30,y_pos,string,0,0,15,0);
    }
    else
    {
        int_to_string(i);
        strcat(string," LOAD BUS");
        outtext3d(30,y_pos,string,0,0,15,0);
    }
}
y_pos+=10;
}
getch();
// closegraph();
}
/* Begin Draw Line Between Bus */

void Line_main1_1(void)
{
    int a,b;

    int_to_string(0[0]);
    outtext3d(x[1]-55,y[1]+10,string,0,0,14,6);

    setcolor(15);
    if (Relat_G[n1]=='Y')
    {
        setlinestyle(SOLID_LINE,1,2);
        circle(x[1],y[1],r);
        setlinestyle(SOLID_LINE,1,3);
        line(x[1],y[1]+r,x[1],y[1]+r+10);
    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Lines over/under circle
setlinestyle(SOLID_LINE,1,3);
line(x[1]-span,y[1]+r+10,x[1]+span,y[1]+r+10);
line(x[1]-(span*2/3),y[1]+r,x[1]-(span*2/3),y[1]+r+10);

if(Relat_L[n1] == 'Y')
{
setlinestyle(SOLID_LINE,1,2);
a = x[1]-(span*2/3);
b = y[1]+r;
line(a-3,b,a+3,b);
line(a+3,b,a,b-9);
line(a-3,b,a,b-9);
}
}

void Line_main2_1(void)
{
int a,b;

int_to_string(O[1]);
outtext3d(x[2]+50,y[2]+10,string,0,0,14,6);

setcolor(15);
if(Relat_G[n2] == 'Y')
{
setlinestyle(SOLID_LINE,1,2);
circle(x[2],y[2],r);
setlinestyle(SOLID_LINE,1,3);
line(x[2],y[2]+r,x[2],y[2]+r+10);
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Lines over/under circle
setlinestyle(SOLID_LINE,1,3);
line(x[2]-span,y[2]+r+10,x[2]+span,y[2]+r+10);
line(x[2]+(span*2/3),y[2]+r,x[2]+(span*2/3),y[2]+r+10);

if(Relat_L[n2] == 'Y')
{
setlinestyle(SOLID_LINE,1,2);
a = x[2]+(span*2/3) ;
b = y[2]+r ;
line(a-3,b,a+3,b);
line(a+3,b,a,b-9);
line(a-3,b,a,b-9);
setlinestyle(SOLID_LINE,1,3);
}

if (Relat[n1][n2] == 'Y')
{
outtextcomplex_PQ_1(FLOW_P[1][n1][1][n2] ,FLOW_Q[1][n1][1][n2] , x[1]+50, y[1]-20); //
display self G+j
outtextcomplex_PQ_2(FLOW_P[2][n1][2][n2] ,FLOW_Q[2][n1][2][n2] , x[1]+50, y[1]-10); //
display self G+j
outtextcomplex_PQ_3(FLOW_P[3][n1][3][n2] ,FLOW_Q[3][n1][3][n2] , x[1]+50, y[1]); //
display self G+j
setlinestyle(SOLID_LINE,1,3);
setcolor(11);
line(x[1]+80,y[1]+40,x[1]+130,y[1]+40);
setlinestyle(SOLID_LINE,1,2);
line(x[1]+130,y[1]+44,x[1]+130,y[1]+36);
line(x[1]+130,y[1]+36,x[1]+140,y[1]+40);
line(x[1]+130,y[1]+44,x[1]+140,y[1]+40);
setlinestyle(SOLID_LINE,1,3);
setcolor(15);

```

```

line(x[1]+(span*2/3),y[1]+r,x[2]-(span*2/3),y[2]+r);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(x[1]+(span*2/3),y[1]+r,x[1]+(span*2/3),y[1]+r+10);
line(x[2]-(span*2/3),y[2]+r,x[2]-(span*2/3),y[2]+r+10);
}
}

```

```
void Line_main3_1(void)
```

```
{
```

```
int a,b;
```

```
int_to_string(O[2]);
```

```
outtext3d(x[4],y[4]+18,string.D,0,14,6);
```

```
;
```

```
if(Relat_G[n3] == 'Y')
```

```
{ setcolor(15);
```

```
setlinestyle(SOLID_LINE,1,2);
```

```
circle(x[4]+(span*2/3),y[4],r);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[4]+(span*2/3),y[4]-r,x[4]+(span*2/3),y[4]-r-10);
```

```
}
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[4]-span,y[4]-r-10,x[4]+span,y[4]-r-10);
```

```
line(x[4]-(span*2/3),y[4]-r,x[4]-(span*2/3),y[4]-r-10);
```

```
if(Relat_L[n3] == 'Y')
```

```
setlinestyle(SOLID_LINE,1,2);
```

```
a = x[4]-(span*2/3) ;
```

```
b = y[4]-r ;
```

```
line(a-3,b,a+3,b);
```

```
line(a+3,b,a,b+9);
```

```
line(a-3,b,a,b+9);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
;
```

```
if (Relat[n1][n3] == 'Y')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outtextcomplex_PQ_1(FLOW_P[1][n1][1][n3] ,FLOW_Q[1][n1][1][n3] , x[3]-260, y[3]); //
```

display sel ,

```
outtextcomplex_PQ_2(FLOW_P[2][n1][2][n3] ,FLOW_Q[2][n1][2][n3] , x[3]-260, y[3]+10); //
```

display sel

```
outtextcomplex_PQ_3(FLOW_P[3][n1][3][n3] ,FLOW_Q[3][n1][3][n3] , x[3]-260, y[3]+20); //
```

display sel

```
setlinestyle(SOLID_LINE,1,3);
```

```
setcolor(11);
```

```
line(x[1],y[1]+r+60,x[4]-(span*2/3)-65,y[4]-r-60);
```

```
setlinestyle(SOLID_LINE,1,2);
```

```
line(x[4]-(span*2/3)-68,y[4]-r-57,x[4]-(span*2/3)-59,y[4]-r-54);
```

```
line(x[4]-(span*2/3)-59,y[4]-r-54,x[4]-(span*2/3)-62,y[4]-r-63);
```

```
line(x[4]-(span*2/3)-62,y[4]-r-63,x[4]-(span*2/3)-68,y[4]-r-57);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
setcolor(15);
```

```
line(x[1],y[1]+r+20,x[4]-(span*2/3),y[4]-r-20);
```

```
line(x[4]-(span*2/3),y[4]-r-10,x[4]-(span*2/3),y[4]-r-20);
```

```
line(x[1],y[1]+r+10,x[1],y[1]+r+20);
```

```
}
```

```
if (Relat[n2][n3] == 'Y')
```

```
{
```

```
outtextcomplex_PQ_1(FLOW_P[1][n2][1][n3] ,FLOW_Q[1][n2][1][n3] , x[3]+90, y[3]);
```

```
outtextcomplex_PQ_2(FLOW_P[2][n2][2][n3] ,FLOW_Q[2][n2][2][n3] , x[3]+90, y[3]+10);
```

```
outtextcomplex_PQ_3(FLOW_P[3][n2][3][n3] ,FLOW_Q[3][n2][3][n3] , x[3]+90, y[3]+20);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
setcolor(11);
```

```
line(x[2]+10,y[2]+r+60,x[4]+(span*2/3)+29,y[4]-r-62);
```

```
setlinestyle(SOLID_LINE,1,2);
```

```
line(x[4]+(span*2/3)+24, y[4]-r-62,x[4]+(span*2/3)+33, y[4]-r-60);
```

```
line(x[4]+(span*2/3)+33, y[4]-r-60,x[4]+(span*2/3)+27, y[4]-r-53);
```

```
line(x[4]+(span*2/3)+27, y[4]-r-53,x[4]+(span*2/3)+24, y[4]-r-62);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
setcolor(15);
```

```
line(x[2],y[2]+r+20,x[4]+(span*2/3),y[4]-r-20);
```

```
line(x[4]+(span*2/3),y[4]-r-10,x[4]+(span*2/3),y[4]-r-20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
line(x[2],y[2]+r+10,x[2],y[2]+r+20);
```

```
}
```

```
void Line_main4_1(void)
```

```
{
```

```
int a,b;
```

```
int_to_string(0){3};
```

```
outtext3d(x[3]+1,y[3]+10,string,0,0,14,6);
```

```
setcolor(15);
```

```
if(Relat_G[n4] == 'Y')
```

```
{ setlinestyle(SOLID_LINE,1,2);
```

```
circle(x[3],y[3],r);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[3],y[3]+r,x[3],y[3]+r+10);
```

```
}
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[3]-span,y[3]+r+10,x[3]+span,y[3]+r+10);
```

```
line(x[3]-(span*2/3),y[3]+r+10,x[3]-(span*2/3),y[3]+r+20);
```

```
if(Relat_L[n4] == 'Y')
```

```
{
```

```
setlinestyle(SOLID_LINE,1,2);
```

```
a = x[3]-(span*2/3) ;
```

```
b = y[3]+r+20 ;
```

```
line(a-3,b,a+3,b);
```

```
line(a+3,b,a,b+9);
```

```
line(a-3,b,a,b+9);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    outtextcomplex_PQ_1(FLOW_P[1][n1][1][n4] ,FLOW_Q[1][n1][1][n4] , x[3]-30, y[3]-120); //
display
    outtextcomplex_PQ_2(FLOW_P[2][n1][2][n4] ,FLOW_Q[2][n1][2][n4] , x[3]-30, y[3]-110); //
displ
    outtextcomplex_PQ_3(FLOW_P[3][n1][3][n4] ,FLOW_Q[3][n1][3][n4] , x[3]-30, y[3]-100); //
displ

    setlinestyle(SOLID_LINE,1,3);
    line(x[1]+(span*2/3),y[1]+r+20,x[3]-(span*2/3),y[3]+r);
    line(x[1]-(span*2/3),y[1]+r+10,x[1]+(span*2/3),y[1]+r+20);
    line(x[3]-(span*2/3),y[3]+r,x[3]-(span*2/3),y[3]+r+10);
}

```

```

if (Relat[n2][n4] == 'Y')

```

```

{
    outtextcomplex_PQ_1(FLOW_P[1][n1][1][n4] ,FLOW_Q[1][n1][1][n4] , x[3]+35, y[3]+20);
    outtextcomplex_PQ_2(FLOW_P[2][n1][2][n4] ,FLOW_Q[2][n1][2][n4] , x[3]+35, y[3]+30);
    outtextcomplex_PQ_3(FLOW_P[3][n1][3][n4] ,FLOW_Q[3][n1][3][n4] , x[3]+35, y[3]+40);

    setcolor(15);
    setlinestyle(SOLID_LINE,1,3);
    line(x[2]-(span*2/3),y[2]+r+20,x[3]+(span*2/3),y[3]+r);
    line(x[2]-(span*2/3),y[2]+r+10,x[2]-(span*2/3),y[2]+r+20);
    line(x[3]+(span*2/3),y[3]+r,x[3]+(span*2/3),y[3]+r+10);
}

```

```

if (Relat[n3][n4] == 'Y')

```

```

{
    outtextcomplex_PQ_1(FLOW_P[1][n3][1][n4] ,FLOW_Q[1][n3][1][n4] , x[3]+20, y[3]);
    outtextcomplex_PQ_2(FLOW_P[2][n3][2][n4] ,FLOW_Q[2][n3][2][n4] , x[3]+20, y[3]+10);
    outtextcomplex_PQ_3(FLOW_P[3][n3][3][n4] ,FLOW_Q[3][n3][3][n4] , x[3]+20, y[3]+20);

    setlinestyle(SOLID_LINE,1,3);
    line(x[3],y[3]+r+20,x[4],y[4]-r-20);
    line(x[4],y[4]-r-10,x[4],y[4]-r-20);
    line(x[3],y[3]+r+10,x[3],y[3]+r+20);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

void Line_main5_1(void)
{
    int a,b;

    int_to_string(0+1);
    outtext3d(x[5]+50,y[5],string,0,0,14,6);
    setcolor(15);

    if(Relat_G[n5] == 'Y')
    {
        setlinestyle(SOLID_LINE,1,2);
        circle(x[5],y[5],r);
        setlinestyle(SOLID_LINE,1,3);
        line(x[5],y[5]-r,x[5],y[5]-r-10);
    }

    setlinestyle(SOLID_LINE,1,3);
    line(x[5]-span,y[5]-r-10,x[5]+span,y[5]-r-10);
    line(x[5]+(span*3/4),y[5]-r,x[5]+(span*3/4),y[5]-r-10);

    if(Relat_L[n5] == 'Y')
    {
        setlinestyle(SOLID_LINE,1,2);
        a = x[5]+(span*3/4) ;
        b = y[5]-r ;
        line(a-3,b,a+3,b);
        line(a+3,b,a,b+9);
        line(a-3,b,a,b+9);
        setlinestyle(SOLID_LINE,1,3);
    }

    if (Relat[n1][n5] == 'Y')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outtextcomplex_PQ_1(FLOW_P[1][n1][1][n5] ,FLOW_Q[1][n1][1][n5] , x[5]+35, y[5]-10); //
```

display se

```
outtextcomplex_PQ_2(FLOW_P[2][n1][2][n5] ,FLOW_Q[2][n1][2][n5] , x[5]+35, y[5]); //display
```

s

```
outtextcomplex_PQ_3(FLOW_P[3][n1][3][n5] ,FLOW_Q[3][n1][3][n5] , x[5]+35, y[5]+10); //
```

display s

```
setcolor(15);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[1]-(span*2/3),y[1]+r+20,x[5]-(span*3/4),y[5]-r-20);
```

```
line(x[5]-(span*3/4),y[5]-r-10,x[5]-(span*3/4),y[5]-r-20);
```

```
line(x[1]-(span*2/3),y[1]+r+10,x[1]-(span*2/3),y[1]+r+20);
```

```
}
```

```
if (Relat[n2][n5] == 'Y')
```

```
{
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
outtextcomplex_PQ_1(FLOW_P[1][n2][1][n5] ,FLOW_Q[1][n2][1][n5] , x[5]+35, y[5]-15);
```

```
outtextcomplex_PQ_2(FLOW_P[2][n2][2][n5] ,FLOW_Q[2][n2][2][n5] , x[5]+35, y[5]+5);
```

```
outtextcomplex_PQ_3(FLOW_P[3][n2][3][n5] ,FLOW_Q[3][n2][3][n5] , x[5]+35, y[5]+15);
```

```
line(x[2]-(span*2/3),y[2]+r+20,x[5]-(span*3/4),y[5]-r-10);
```

```
line(x[5]-(span*3/4),y[5]-r-10,x[5]-(span*3/4),y[5]-r-20);
```

```
line(x[2]-(span*2/3),y[2]+r+10,x[2]-(span*2/3),y[2]+r+20);
```

```
}
```

```
if (Relat[n3][n5] == 'Y')
```

```
{
```

```
outtextcomplex_PQ_1(FLOW_P[1][n3][1][n5] ,FLOW_Q[1][n3][1][n5] , x[3]+35, y[3]+20);
```

```
outtextcomplex_PQ_2(FLOW_P[2][n3][2][n5] ,FLOW_Q[2][n3][2][n5] , x[3]+35, y[3]+30);
```

```
outtextcomplex_PQ_3(FLOW_P[3][n3][3][n5] ,FLOW_Q[3][n3][3][n5] , x[3]+35, y[3]+40);
```

```
setlinestyle(SOLID_LINE,1,3);
```

```
line(x[4]-r,y[4]-r,x[5]-(span*1/4),y[5]-r-20);
```

```
line(x[5]-(span*1/4),y[5]-r-10,x[5]-(span*1/4),y[5]-r-20);
```

```
line(x[4]-r,y[4]-r,x[4]-r,y[4]-r-10);
```

```
}
```

if (Relat[n4][n5] == 'Y') หน่วยงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    outtextcomplex_PQ_1(FLOW_P[1][n4][1][n5] ,FLOW_Q[1][n4][1][n5] , x[3]+35, y[3]+20);
    outtextcomplex_PQ_2(FLOW_P[2][n4][2][n5] ,FLOW_Q[2][n4][2][n5] , x[3]+35, y[3]+30);
    outtextcomplex_PQ_3(FLOW_P[3][n4][3][n5] ,FLOW_Q[3][n4][3][n5] , x[3]+35, y[3]+40);
    setlinestyle(SOLID_LINE,1,3); line(x[3]+(span*2/3),y[3]+r+20,x[4]+26,y[4]-r-53); //1
    line(x[4]+35,y[4]-r-45,x[5]+(span*1/4),y[4]-28) ;
    line(x[5]+(span*1/4),y[4]-28,x[5]+(span*1/4),y[5]-r-20);
    line(x[5]+(span*1/4),y[5]-r-10,x[5]+(span*1/4),y[5]-r-20);
    line(x[3]+(span*2/3),y[3]+r+10,x[3]+(span*2/3),y[3]+r+20);
}
}

```

```

void outtextcomplex(double Re , double Im ,int x ,int y)

```

```

{ double_to_text (Re,2);
  strcat(text ,"+j");
  strcpy(text_tmp,text);
  double_to_text(Im,2);
  strcat(text_tmp,text);
  outtext3d(x,y,text_tmp,0,0,4,6);
}

```

```

void outtextcomplex_PQ_1(double T , double O ,int c ,int d)

```

```

{ double_to_text (T,4);
  strcat(text ,"+j");
  strcpy(text_tmp,text);
  double_to_text(O,4);
  strcat(text_tmp,text);
  outtext3d(c,d,text_tmp,0,0,10,0);
}

```

```

void outtextcomplex_PQ_2(double Y , double N ,int u ,int v)

```

```

{ double_to_text (Y,4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
strcpy(text_tmp,text);
double_to_text(N,4);
strcat(text_tmp,text);
outtext3d(u,v,text_tmp,0,0,9,0);
}

void outtextcomplex_PQ_3(double H , double K ,int re ,int se)

```

```

{ double_to_text (H,4);
  strcat(text ,"+j");
  strcpy(text_tmp,text);
  double_to_text(K,4);
  strcat(text_tmp,text);
  outtext3d(re,se,text_tmp,0,0,12,8);
}

```

```

void JACOB(void)

```

```

{
  int d,i,m,X,Y,B_X,DAT;
  DIMEN_P=0;
  Y = 1;

  for (i=1;i<MAX_BUS ; i++)
    for(m=1;m<MAX_BUS;m++)
      JCBN_P[i][m]=0;

  for (i=1 ; i<MAX_BUS ; i++)
    { if (i != slack_bus )
      { DIMEN_P++;
        for ( B_X=1, X=1 ; B_X < MAX_BUS ; B_X++,X++)
          { if ( B_X != slack_bus)
            { DAT = Y_B[1][i][1][B_X];
              JC'BN_P[Y][X] = -1 * DAT;
            }
          }
        }
    }
}
else

```

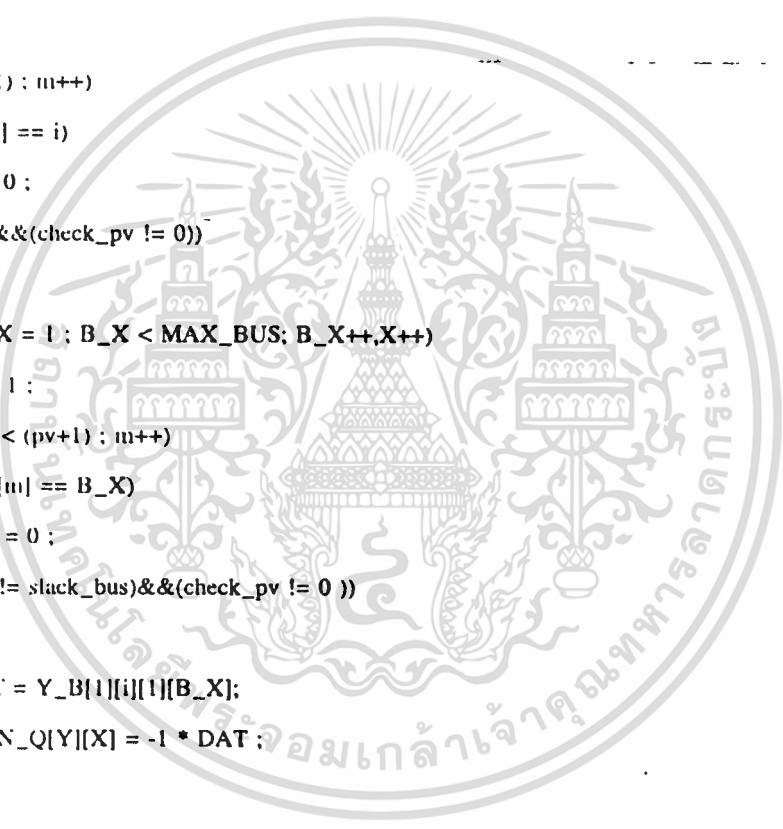
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดก็ตาม ห้ามนำไปใช้เพื่อการค้า และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        X--;
    }
    Y++;
}
}
DIMEN_P++;

/*-----JCBN_Q--*/
DIMEN_Q = 0;
for(i = 1, Y = 1; i < MAX_BUS; i ++, Y++)
{ check_pv = 1;
  _for(m=1; m < (pv+1); m++)
    if(v_control[m] == i)
      check_pv = 0;
  if ((i != slack_bus)&&(check_pv != 0))
  { DIMEN_Q++;
    for(B_X = 1, X = 1; B_X < MAX_BUS; B_X++, X++)
    { check_pv = 1;
      for(m=1; m < (pv+1); m++)
        if(v_control[m] == B_X)
          check_pv = 0;
      if ((B_X != slack_bus)&&(check_pv != 0))
      {
        DAT = Y_B[1][i][1][B_X];
        JCBN_Q[Y][X] = -1 * DAT;
      }
      else
        X--;
    }
  }
  else
    Y--;
}
DIMEN_Q++;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-----END OF JACOB-----*/
```

```
void Make_MAGNI_ANG(void)
```

```
{
```

```
int BS_I,BS_K,PH_P,PH_M;
```

```
double DAT_1,DAT_2;
```

```
double DAT_3,DAT_4,DAT_5;
```

```
for(BS_I = 1 ; BS_I< MAX_BUS; BS_I++)
```

```
for(BS_K = 1 ; BS_K< MAX_BUS; BS_K++)
```

```
for(PH_P=1;PH_P <4;PH_P++)
```

```
for(PH_M=1;PH_M < 4 ;PH_M++)
```

```
{
```

```
DAT_2=0;
```

```
if(Y_G[PH_P][BS_I][PH_M][BS_K] == 0)
```

```
{
```

```
DAT_1 = 0;
```

```
}
```

```
else
```

```
{
```

```
DAT_1 = (Y_B[PH_P][BS_I][PH_M][BS_K] / Y_G[PH_P][BS_I][PH_M][BS_K]);
```

```
}
```

```
DAT_2 = atan (DAT_1);
```

```
ANG_Y[PH_P][BS_I][PH_M][BS_K] = (DAT_2*180)/pi;
```

```
if( Y_G[PH_P][BS_I][PH_M][BS_K] < 0 )
```

```
ANG_Y[PH_P][BS_I][PH_M][BS_K] = ANG_Y[PH_P][BS_I][PH_M][BS_K]
```

```
+180;
```

```
DAT_3 = Y_B[PH_P][BS_I][PH_M][BS_K] ;
```

```
DAT_3 *= Y_B[PH_P][BS_I][PH_M][BS_K] ;
```

```
DAT_4 = Y_G[PH_P][BS_I][PH_M][BS_K] ;
```

```
DAT_4 *= Y_G[PH_P][BS_I][PH_M][BS_K] ;
```

```
DAT_5 = DAT_3+DAT_4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MAGNI_Y[PH_P][BS_I][PH_M][BS_K] = sqrt(DAT_5);
```

```
}
```

```
}
```

```
void Make_Delta_PQ_2(void)
```

```
{
```

```
double DATA_P, DATA_Q, DATA_1, DATA_2, DATA_3;
```

```
int I, K;
```

```
/*-----*/
```

```
for(I = 1 ; I < MAX_BUSES ; I++) /*--- 1 Slack bus ---*/
```

```
if( I != slack_bus )
```

```
{ DATA_P = 0 ; DATA_Q = 0 ;
```

```
for(K = 1 ; K < MAX_BUSES ; K++)
```

```
{ DATA_1 = (MAGNI_V[2][I] * MAGNI_V[2][K]) * MAGNI_Y[2][I][2][K] ;
```

```
DATA_2 = (ANGLE_V[2][I] - ANGLE_V[2][K]) - ANG_Y[2][I][2][K] ;
```

```
DATA_2 = ( DATA_2 * pi ) / 180 ;
```

```
DATA_3 = cos( DATA_2 ) ;
```

```
DATA_P = DATA_P + ( DATA_1 * DATA_3 ) ;
```

```
DATA_3 = sin( DATA_2 ) ;
```

```
DATA_Q = DATA_Q + ( DATA_1 * DATA_3 ) ;
```

```
}
```

```
DELTA_P[2][I] = Psch[2][I] - DATA_P ;
```

```
DEL_PbyV[2][I] = DELTA_P[2][I] / MAGNI_V[2][I];
```

```
DELTA_Q[2][I] = Qsch[2][I] - DATA_Q ;
```

```
DEL_QbyV[2][I] = DELTA_Q[2][I] / MAGNI_V[2][I];
```

```
}
```

```
}
```

```
/*----- End of Make_Delta_PQ */
```

```
void Make_Delta_PQ_3(void) นี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
double DATA_P, DATA_Q, DATA_1, DATA_2, DATA_3;
int I, K;

for(I = 1 ; I < MAX_BUS ; I++) /*--- 1 Slack bus ---*/
if( I != slack_bus )
{ DATA_P = 0 ; DATA_Q = 0 ;
for(K = 1 ; K < MAX_BUS ; K++)
{ DATA_1 = MAGNI_V[3][I] * MAGNI_V[3][K] * MAGNI_Y[3][I][3][K] ;
DATA_2 = ANGLE_V[3][I] - ANGLE_V[3][K] - ANG_Y[3][I][3][K] ;
DATA_2 = ( DATA_2 * pi ) / 180 ;

DATA_3 = cos( DATA_2 ) ;
DATA_P = DATA_P + ( DATA_1 * DATA_3 ) ;

DATA_3 = sin( DATA_2 ) ;
DATA_Q = DATA_Q + ( DATA_1 * DATA_3 ) ;
}
DELTA_P[3][I] = Psch[3][I] - DATA_P ;
DEL_PbyV[3][I] = DELTA_P[3][I] / MAGNI_V[3][I];

DELTA_Q[3][I] = Qsch[3][I] - DATA_Q ;
DEL_QbyV[3][I] = DELTA_Q[3][I] / MAGNI_V[3][I];
}
}
/*----- End of Make_Delta_PQ */

```

```
void Sol_Matrx_2()
```

```

{
int i, m, X, Y, Z;
int BUS;
double MULTI, ANS[SP_MAX_BUS];
/*-----JCBN <= DelPhyV */

```

การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ if ( i != slack_bus)
    JCBN_P[Y][DIMEN_P] = DEL_PbyV[2][i];
else
    Y = Y-1;
}

```

```

/*-----Cal Del_AngleV */

```

```

for(Y = 1 ; Y < DIMEN_P ; Y++)
{ for(X = 1 ; X < DIMEN_P ; X++)
    { if( X != Y)
        { MULTI= JCBN_P[X][Y] / JCBN_P[Y][Y];
          for(Z = 1 ; Z < (DIMEN_P+1);Z++)
              JCBN_P[X][Z] = JCBN_P[X][Z]-(MULTI * JCBN_P[Y][Z]);
        }
    }
}
for (Y = 1 ;Y < DIMEN_P; Y++)
    ANS[Y] = JCBN_P[Y][DIMEN_P] / JCBN_P[Y][Y];

```

```

/*-----Angle_V + Del_AngleV */

```

```

for(Y:= 1 ,Z = 1 ;Y < MAX_BUS; Y++,Z++)
{ if ( Y != slack_bus)
    {
        DELTA_ANG[2][Y] = ANS[Z];
        ANGLE_V[2][Y] = ANGLE_V[2][Y] + (1.5*DELTA_ANG[2][Y]);
    }
else
    Z--;
}

```

```

/*-----JCBN <= DelQbyV */

```

```

for (Y = 1,i = 1 ; i < MAX_BUS ; Y++, i ++)

```

```

{ check_pv = 1 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(m=1 ;m < (pv+1) ; m++)
    if(v_control[m] == i )
        check_pv = 0;
if (( i !=slack_bus)&&(check_pv != 0))
    JCBN_Q[Y][DIMEN_Q] = DEL_QbyV[2][i];
else
    Y--;
}

```

```

/*-----Cal Del_MagniV */

```

```

for(Y =1 ;Y < DIMEN_Q; Y++)
{
for(X = 1 ; X < DIMEN_Q ; X++)
{
if (X != Y)
{
MULTI = JCBN_Q[X][Y] / JCBN_Q[Y][Y];
for(Z = 1; Z < DIMEN_Q+1;Z++)
    JCBN_Q[X][Z] = JCBN_Q[X][Z]-(MULTI * JCBN_Q[Y][Z]);
}
}
}
}

```

```

for(Y = 1 ; Y < DIMEN_Q;Y++)
    ANS[Y] = JCBN_Q[Y][DIMEN_Q] / JCBN_Q[Y][Y] ;

```

```

/* -----Magni_V+ Del_MagniV */

```

```

for(Y = 1 , Z = 1 ;Y < MAX_BUS ; Y++,Z++)
{
check_pv = 1;
for(m=1 ;m < (pv+1) ; m++)
    if(v_control[m] == Y )
        check_pv = 0 ;
if((Y != slack_bus)&&(check_pv != 0))
{
DELTA_VOL[2][Y] = ANS[Z];
MAGNI_V[2][Y] = MAGNI_V[2][Y] + DELTA_VOL[2][Y];
}
}
else

```

```

Z--;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

}

/* -----END OF SOL_Matrx-----*/

```

```

void Cal_Error_2()

```

```

{
    int k;
    double SQ_V , SQ_A ;

    ERROR_2=0;
    for(k = 1 ; k < MAX_BUS ; k++)
        if(k!= slack_bus)
        {
            SQ_V = DELTA_VOL[2][k] * DELTA_VOL[2][k];
            SQ_V = sqrt(SQ_V);
            SQ_A = DELTA_ANG[2][k] * DELTA_ANG[2][k];
            SQ_A = sqrt(SQ_A);

            if((SQ_V>ERR)||(SQ_A>ERR))
                ERROR_2=1;
        }
}

/*----- End of Cal_Error */

```

```

void Sol_Matrx_3()

```

```

{
    int i, m, Y, X, Z;
    int BUS;
    double MULTANS[SP_MAX_BUS];

/*-----JCBN <= DelPbyV */

    for( Y=1 , i=1; Y < DIMEN_P ; Y++,i++)
        { if ( i != slack_bus)
            JCBN_P[Y][DIMEN_P] = DEL_PbyV[3][i];
        }
    else

```

แล  $Y = Y - 1$ ; เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}
/*-----Cal Del_AngleV */
for(Y = 1 ; Y < DIMEN_P ; Y++)
{
for(X = 1 ; X < DIMEN_P ; X++)
{
if( X != Y)
{
MULTI= JCBN_P[X][Y] / JCBN_P[Y][Y];
for(Z = 1 ; Z < (DIMEN_P+1);Z++)
JCBN_P[X][Z] = JCBN_P[X][Z]-(MULTI * JCBN_P[Y][Z]);
}
}
}
}

```

```

for (Y = 1 ; Y < DIMEN_P; Y++)
ANS[Y] = JCBN_P[Y][DIMEN_P] / JCBN_P[Y][Y];

```

```

/*-----Angle_V + Del_AngleV */
for(Y = 1 ; Z = 1 ; Y < MAX_BUS; Y++,Z++)
{
if (Y != slack_bus)
{
DELTA_ANG[3][Y] = ANS[Z];
ANGLE_V[3][Y] = ANGLE_V[3][Y] + (1.5*DELTA_ANG[3][Y]);
}
else
Z--;
}
}

```

```

/*-----JCBN <= DelQbyV */

```

```

for (Y = 1, i = 1 ; i < MAX_BUS ; Y++, i ++ )
{
check_pv = 1 ;
for(m=1 ;m < (pv+1) ; m++)
if(v_control[m] == i )
check_pv = 0;

```

if (( i != slack\_bus) && (check\_pv != 0))การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JCBN_Q[Y][DIMEN_Q] = DEL_QbyV[3][i];
```

```
else
```

```
Y--;
```

```
}
```

```
/*-----Cal Del_MagniV */
```

```
for(Y = 1 ; Y < DIMEN_Q; Y++)
```

```
{ for(X = 1 ; X < DIMEN_Q ; X++)
```

```
{ if (X != Y)
```

```
{ MULTI = JCBN_Q[X][Y] / JCBN_Q[Y][Y];
```

```
for(Z = 1; Z < DIMEN_Q+1;Z++)
```

```
JCBN_Q[X][Z] = JCBN_Q[X][Z]-(MULTI * JCBN_Q[Y][Z]);
```

```
}
```

```
}
```

```
}
```

```
for(Y = 1 ; Y < DIMEN_Q;Y++)
```

```
ANS[Y] = JCBN_Q[Y][DIMEN_Q] / JCBN_Q[Y][Y] ;
```

```
/* -----Magni_V+ Del_MagniV */
```

```
for(Y = 1 , Z = 1 ; Y < MAX_BUS ; Y++,Z++)
```

```
{ check_pv = 1;
```

```
for(m=1 ;m < (pv+1) ; m++)
```

```
if(v_control[m] == Y )
```

```
check_pv = 0 ;
```

```
if((Y != slack_bus)&&(check_pv != 0))
```

```
{
```

```
DELTA_VOL[3][Y] = ANS[Z];
```

```
MAGNI_V[3][Y] = MAGNI_V[3][Y] + DELTA_VOL[3][Y];
```

```
}
```

```
else
```

```
Z--;
```

```
)
```

```
}
```

```
/* -----END OF SOL_Matrix-----*/
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Cal_Error_3()
```

```
{  
    int k;  
    double SQ_V , SQ_A ;  
    ;  
    ERROR_3=0;  
    for(k = 1 ; k < MAX_BUS : k++)
```

```
        if(k!= slack_bus)
```

```
        {  
            SQ_V = DELTA_VOL[3][k] * DELTA_VOL[3][k];  
            SQ_V = sqrt(SQ_V);  
            SQ_A = DELTA_ANG[3][k] * DELTA_ANG[3][k];  
            SQ_A = sqrt(SQ_A);
```

```
            if((SQ_V>ERR)||(SQ_A>ERR))
```

```
                ERROR_3=1;
```

```
        }
```

```
}
```

```
/*----- End of Cal_Error */
```

```
void Cal_Flow_PQ_2(void)
```

```
/* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
```

```
{  
    int p , q , k , Y , d , check_pv;
```

```
    double c[SP_MAX_BUS] , f[SP_MAX_BUS] ;
```

```
    double REAL_1 , IMG_1 , TEMP_DATA ;
```

```
/*----- Cal e_f */
```

```
for(p = 1 ; p < MAX_BUS : p++)
```

```
{ TEMP_DATA = (ANGLE_V[2][p] * pi) / 180 ;
```

```
c[p] = MAGNI_V[2][p] * cos(TEMP_DATA) ;
```

```
f[p] = MAGNI_V[2][p] * sin(TEMP_DATA) ;
```

```
}
```

~~/\* เอกสารนี้เป็นลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้~~

```

for(p = 1 ; p < MAX_BUS : p++)
{
  for(q = 1 ; q < MAX_BUS : q++)
  {
    if(q != p)
    {
      REAL_1 = c[p] * ( c[p] - c[q] ) + f[p] * ( f[p] - f[q] ) ;
      IMG_1 = c[p] * ( f[p] - f[q] ) - f[p] * ( c[p] - c[q] ) ;
      /* Ep^(Ep-Eq) = REAL_1 + jIMG_1 */

      TEMP_DATA = REAL_1 * Y_G[2][p][2][q] ;
      FLOW_P[2][p][2][q] = TEMP_DATA - ( IMG_1 * Y_B[2][p][2][q] ) ;
      FLOW_P[2][p][2][q] = -1*(BASE * FLOW_P[2][p][2][q] ) ;

      TEMP_DATA = REAL_1 * Y_B[2][p][2][q] ;
      TEMP_DATA = TEMP_DATA + ( IMG_1 * Y_G[2][p][2][q] ) ;

      FLOW_Q[2][p][2][q] = -1 * TEMP_DATA ;
      Img_Ypq[2][p][2][q] = 0 ;
      TEMP_DATA = Img_Ypq[2][p][2][q] * (c[p]*c[p] + f[p]*f[p]) ;
      FLOW_Q[2][p][2][q] = FLOW_Q[2][p][2][q] - TEMP_DATA ;
      FLOW_Q[2][p][2][q] = -1*(BASE * FLOW_Q[2][p][2][q] ) ;
    }
  }
}

```

/\*----- Cal G\_PQ SLACK BUS \*/

```

for(k = 1 , REAL_1 = 0 , IMG_1 = 0 ; k < MAX_BUS ; k++)
{
  if( k != slack_bus )
  {
    REAL_1 = REAL_1 + FLOW_P[2][slack_bus][2][k] ;
    G_P[2][slack_bus] = REAL_1 ;
    IMG_1 = IMG_1 + FLOW_Q[2][slack_bus][2][k] ;
    G_Q[2][slack_bus] = IMG_1 ;
  }
}

```

G\_P[2][slack\_bus] = G\_P[2][slack\_bus] + L\_P[2][slack\_bus] ;

G\_Q[2][slack\_bus] = G\_Q[2][slack\_bus] + L\_Q[2][slack\_bus] ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----Cal G_Q PV BUS */
for(k = 1 , TEMP_DATA = 0 ; k < MAX_BUS ; k++)
{
    check_pv = 1 ;
    for(d = 1 ; d < (pv+1) . d++ )
        if( v_control[d] == k )
            check_pv = 0;

    if( check_pv == 0 )
    { for(q = 1 ; q < MAX_BUSES ; q++)
        { if(q != k)
            TEMP_DATA = TEMP_DATA + FLOW_Q[2][k][2][q] ;
        }
        G_Q[2][k] = TEMP_DATA + I_Q[2][k] ;
    }
}

```

/\*-----End of Cal\_PQ -----\*/

```

void Cal_Flow_PQ_3(void)
/* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
{
    int p , q , k , Y , d , check_pv;
    double e[SP_MAX_BUS] , i[SP_MAX_BUS] ;
    double REA[1] , IMG[1] , TEMP_DATA ;
}

```

```

/*----- Cal e_f */
for(p = 1 ; p < MAX_BUS , p++)
{ TEMP_DATA = (ANGLE_V[3][p] * pi) / 180 ;
    e[p] = MAGNI_V[3][p] * cos(TEMP_DATA) ;
    f[p] = MAGNI_V[3][p] * sin(TEMP_DATA) ;
}

```

/\*-----เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(p = 1 ; p < MAX_BUS ; p++)
{
  for(q = 1 ; q < MAX_BUS ; q++)
  {
    if(q != p)
    {
      REAL_1 = e[p] * ( e[q] - e[q] ) + f[p] * ( f[p] - f[q] ) ;
      IMG_1 = e[p] * ( f[p] - f[q] ) - f[p] * ( e[p] - e[q] ) ;
      /* Ep^(Ep-Eq) = REAL_1 + jIMG_1 */
      TEMP_DATA = REAL_1 * Y_G[3][p][3][q] ;
      FLOW_P[3][p][3][q] = TEMP_DATA - ( IMG_1 * Y_B[3][p][3][q] ) ;
      FLOW_P[3][p][3][q] = -1*(BASE * FLOW_P[3][p][3][q]) ;
      TEMP_DATA = REAL_1 * Y_B[3][p][3][q] ;
      TEMP_DATA = TEMP_DATA + ( IMG_1 * Y_G[3][p][3][q] ) ;
      FLOW_Q[3][p][3][q] = -1 * TEMP_DATA ;
      Img_Ypq2[3][p][3][q]=0;
      TEMP_DATA = Img_Ypq2[3][p][3][q] * (e[p]*e[p] + f[p]*f[p]) ;
      FLOW_Q[3][p][3][q] = FLOW_Q[3][p][3][q] - TEMP_DATA ;
      FLOW_Q[3][p][3][q] =-1*( BASE * FLOW_Q[3][p][3][q]) ;
    }
  }
}

```

/\*----- Calc G\_PQ SLACK BUS \*/

```

for(k = 1 , REAL_1 = 0 , IMG_1 = 0 ; k < MAX_BUS ; k++)
{
  if( k != slack_bus )
  {
    REAL_1 = REAL_1 + FLOW_P[3][slack_bus][3][k] ;
    G_P[3][slack_bus] = REAL_1 ;
    IMG_1 = IMG_1 + FLOW_Q[3][slack_bus][3][k] ;
    G_Q[3][slack_bus] = IMG_1 ;
  }
}
G_P[3][slack_bus] = G_P[3][slack_bus] + L_P[3][slack_bus] ;
G_Q[3][slack_bus] = G_Q[3][slack_bus] + L_Q[3][slack_bus] ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*----- Cal G_Q PV BUS */
for(k = 1 , TEMP_DATA = 0 ; k < MAX_BUS ; k++)

```

```

{
    check_pv = 1 ;
    for(d = 1 ; d < (pv+1) . d++)
        if( v_control[d] == k )
            check_pv = 0;

```

```

    if( check_pv == 0 )

```

```

        { for(q = 1 ; q < MAX_BUS ; q++)

```

```

            { if(q != k)

```

```

                TEMP_DATA = TEMP_DATA + FLOW_Q[3][k][3][q] ;

```

```

            }

```

```

        G_Q[3][k] = TEMP_DATA + L_Q[3][k] ;

```

```

    }
}

```

```

void compound(void)

```

```

{ int p;

```

```

    Make_MAGN_ANGLE();

```

```

    JACOB();

```

```

for(p=1;p<1000;p++)

```

```

{

```

```

    Make_Delta_PQ_2();

```

```

    Make_Delta_PQ_3();

```

```

    Sol_Matrx_2();

```

```

    Sol_Matrx_3();

```

```

    Cal_Error_2();

```

```

    Cal_Error_3();

```

```

    if((ERROR_2==0)&&(ERROR_3==0))

```

```

    {

```

```

        Cal_Flow_PQ_2();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cal\_Flow\_PQ\_3():

}

}

}

/\*----- End of Cal\_PQ -----\*/



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงด้วยดีตามวัตถุประสงค์เพราะได้รับความช่วยเหลือเป็นอย่างดีจากอาจารย์ที่ปรึกษา ศศ.ศิริวัฒน์ โพรเวชกุล และอาจารย์ อานันทวัฒน์ คุณากร ที่ช่วยให้ความอำนวยความสะดวกในด้านต่าง ๆ รวมทั้งคำแนะนำที่มีประโยชน์อย่างมากในการดำเนินงาน ตลอดจนอาจารย์ท่านอื่น ๆ และเพื่อน ๆ ทุกคนที่ส่วนให้ความช่วยเหลือจนทำให้ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วง ไว้ ณ โอกาสนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1]. ชำนาญ ห่อเกียรติ, รศ.ดร. "ระบบไฟฟ้ากำลัง." มหาวิทยาลัยเกษตรศาสตร์,2526
- [2]. สมโภชน์ ประไพ วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรบัณฑิต "Computer Aid Design For Power System" สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2524
- [3]. Glinn W Stag "Computer Method in Power System Analysis"  
McGRW-Hill Book Company,1987
- [4] Gonin,T, "Modern Power System Analysis " John Wiley & Sons., in 1988
- [5] J. Arrillaga and C.P.Arrold "Computer Modelling of Electrical power System"  
John Willey & Sons 1983
- [6] Stevenson,W.D, Jr , "Elements of Power System Analysis", 4th.ed
- [7] Weedy,B.M, "Electric Power Systems" 3nd ed. Wiley,New York,1979
- [8] Vincent Del toro, "Electri Power System"Prentice-hall,Inc 1992