



สามเฟสโหลดฟลว์
THREE-PHASE LOAD FLOW



ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032657

ปีการศึกษา 2535

การศึกษาเรื่องโหลดฟลว์ 3 เฟส
Three-Phase Load Flow

โดย

- | | |
|-------------------------------|---------|
| 1. นาย ไพสันต์ พิฒนะคุหา | 32.1223 |
| 2. นาย สุรินทร์ จินต์जारุพงศ์ | 32.1394 |
| 3. นาย อติเทพ ราชณรงค์ | 32.1411 |

อาจารย์ที่ปรึกษา

(อ.ศิริวัฒน์ โปษิเวชกุล)

(อ.อานันท์วัฒน์ คุณากร)

ปริญญาโท ประจำปีการศึกษา 2535


ภาควิชา วิศวกรรมไฟฟ้า

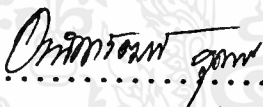
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาโพลดิพลว์ 3 เฟส

ผู้จัดทำ

1. นาย ไพสันต์ พัฒนะคุหา 32.1223
2. นายสุรินทร์ จินต์จารุพงศ์ 32.1394
3. นาย อติเทพ ราชรงค์ 32.1411


.....อาจารย์ที่ปรึกษา
(อ.ศิริวิวัฒน์ โปธิเวชกุล)


.....อาจารย์ที่ปรึกษา
(อ.อานันท์ คุณากร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพลโทพลว์ 3 เฟส

นาย ไพลินต์ พัฒนาคุหา
 นาย สุรินทร์ จินต์จารุงศ์
 นาย อติเทพ ราชณรงค์

อ.ศิริวัฒน์ โปธิเวชกุล อาจารย์ที่ปรึกษา
 อ.อานันท์วัฒน์ คุณากร อาจารย์ที่ปรึกษา

ปีการศึกษา 2535

บทคัดย่อ

โพลโทพลว์ เป็นการศึกษาดังภาวะการส่งและรับกำลังไฟฟ้าที่บัสต่าง ๆ ของระบบไฟฟ้ากำลังเป็นการคำนวณเพื่อหาค่าแรงดัน มุมของแรงดัน กำลังจริง กำลังรีแอกทีฟที่บัสต่างๆ ของระบบไฟฟ้ากำลัง โพลโทพลว์สามารถศึกษาได้ทั้งในระบบปกติหรือไม่ปกติ โพลโทพลว์มีความจำเป็นในการวางแผนพัฒนาระบบในอนาคต เช่นการเชื่อมต่อกับระบบไฟฟ้ากำลังอื่นๆ การต่อโพลเพิ่ม โรงจักรไฟฟ้าที่ผลิต และสายส่งก่อนที่จะมีการติดตั้ง รวมทั้งการศึกษาเสถียรภาพของระบบอีกด้วย

ในการคำนวณจะอยู่ในรูปของปริมาณเฟส (A, B, C) และจะใช้วิธีของ Newton Raphson เป็นหลัก หน่วยความจำที่จัดเก็บใน Three phase load flow โดยปกติจะมากกว่า single phase load flow และจะต้องมี algorithms ที่ดีกว่าแบบ single phase โดยปกติจะใช้ Three phase load flow ในการวางแผนมากกว่าใช้ในการปฏิบัติการ เมื่อนำโปรแกรมไปใช้ในการศึกษาการปฏิบัติงานในภาวะไม่สมดุลย์ของระบบไฟฟ้ากำลัง จะต้องเพิ่มลักษณะพิเศษอย่างอื่นที่จำเป็นคือ Automatic transformer tapping และ generator VAR limiting อีกด้วย

ในการวิเคราะห์ระบบไฟฟ้ากำลัง ในสภาวะ steady state ถ้าหากระบบเป็นแบบสมดุลย์ เราสามารถใช้ single-phase load flow ในการวิเคราะห์ได้ แต่ถ้าเกิดสภาวะ unbalance ขึ้นเราจำเป็นต้องใช้ three phase load flow โดยเราจะใช้นาความคิดของ automatic voltage regulators จะรวมอยู่ในสมการของ Jacobian matrix อีกด้วย บางครั้งอาจจะเป็นการไม่ประหยัดเลย ถ้าจะพยายามทำให้โพลสมดุลย์อย่างแท้จริง เช่นในกรณีของสายส่ง ผลของการ unbalance จะทำให้เกิด negative sequence currents ซึ่งทำให้ rotor ของ machine เกิด overheat ขึ้นมา และ zero sequence อาจทำให้ relay ทำงานผิดพลาด และยังทำให้เกิดความสูญเสียขึ้นในสายส่งที่ไม่ได้ทำการช้อวีส์สายช้อประโยชน์ด้านการค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในวงจำกัดเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Three-phase load flow

Paisan Pattanakuha

Surin Jinjarupong

Adithep Rajanaronk

Siriwat Photivejkul Advisor

Ananthawat Kunakorn Advisor

1992

Abstract

Load flow studies determine the supply or absorption power at various buses and calculate for voltages, phase angles, real power, reactive power at various buses of power systems at normal operation or not. Load flows are essential in the developing plans for the future systems. For examples, the connection with other power systems, new loading, new generating power stations and new transmission lines before installation and also stability studies in power systems.

Analysis power system on steady state condition, in balance system uses single phase load flow for analysing is adequate. But in unbalance system, three phase load flow is necessary. It takes a concept of single-phase fast-decoupled algorithm and modify to produce an efficient three-phase fast decoupled load flow. The compound-coil concept is used in the representation of power system components, and the effect of automatic voltage regulators is modelled as part of the Jacobian matrix equation. However, in practice it is uneconomical to balance the load completely, such as transmission system. Among the effects of power-system unbalance are negative sequence currents causing machine rotor overheating, zero sequence causing relay maloperation and increased losses due to parallel

เอกสารนี้ untransposed lines. รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solution are all done in phase quantities (A,B,C) and uses Newton-Raphson method for calculating. The storage and computational requirements of a three-phase load-flow program are much greater than those of the corresponding single-phase case. The need for efficient algorithms is therefore significant even though, in contrast to single-phase analysis, the three-phase load flow is likely to remain a planning rather than an operational exercise. When program is used for post-operational studies of important unbalanced situations on the power system, additional practical features such as automatic transformer tapping and generator VAR limiting are necessary.

บทนำ

โพลทไฟลว์ เป็นภาวะการส่งและการรับกำลังไฟฟ้าที่บีบคั้นต่างๆ ของระบบไฟฟ้ากำลัง การคำนวณโพลทไฟลว์เป็นการคำนวณเพื่อหาค่าแรงดัน มุมของแรงดัน กำลังงานจริง และกำลังงานรีแอกทีฟที่บีบคั้นต่างๆ ของระบบไฟฟ้ากำลัง การศึกษาโพลทไฟลว์สามารถศึกษาได้ขณะระบบไฟฟ้ากำลังอยู่ในสภาวะปกติ และในสภาวะผิดปกติ เช่น ลัดวงจรในระบบผลจากการคำนวณค่าทั้ง 4 ที่บีบคั้นต่างๆ จะนำไปตัดสินพิจารณาว่าเสถียรภาพของระบบยังมีอยู่หรือไม่ และจะต้องปฏิบัติงานอย่างไร

โพลทไฟลว์ มีประโยชน์ทั้งในแง่การสังเกตเสถียรภาพของระบบ และยังมีประโยชน์ในการวางแผนขยายระบบ อันได้แก่ การต่อระบบไฟฟ้ากำลังหนึ่งกับอีกระบบหนึ่ง การต่อโพลทเพิ่มขึ้น การต่อกับโรงจักรผลิตไฟฟ้าโรงใหม่ และการต่อสายส่งแรงสูงให้ยาวขึ้น เป็นต้น

ประโยชน์ของโพลทไฟลว์ ที่สำคัญคือ

1. ใช้ประกอบในการวางแผนของระบบ (system planning) ดังนี้คือ
 - การไหลของ MW และ Mvar ในบรรทัดของเน็คเวอด์
 - บัสบาร์โวลเตจ
 - ผลในการจัดเตรียมวงจรและการต่อวงจรใหม่เข้ากับโพลทของระบบ (system loading)
 - ผลในการปรับมุมและปรับแรงดันให้สูงขึ้นให้เข้ากับโพลทของระบบ (system loading)
 - ใช้ในการปรับเปลี่ยนขนาดของตัวนำ และแรงดันของระบบ
2. ใช้ประกอบในการพิจารณาการปฏิบัติงานของระบบ (system operation)
 - ผลของการสูญเสียของเครื่องกำเนิดไฟฟ้าและค่าส่งของระบบ
 - ช่วยในการเดินเครื่องและการจ่ายโพลทให้มีประสิทธิภาพ
 - ช่วยในการตรวจสอบและสูญเสียในระบบ
3. ใช้เป็นจุดเริ่มต้นในการศึกษาเรื่องอื่นได้ ดังนี้
 - ศึกษาถึงการป้องกันระบบทั้งกระแสและแรงดันโดยแรงดันจะถูกตรวจสอบด้วย protection relays และกระแสจะถูกอินเทอร์ลักซ์ โดยรีเลย์รีเลย์ที่แตกต่าง
 - ศึกษาถึงเสถียรภาพของระบบ ในการหาเวลาตอบสนอง (lime response) ของระบบไฟฟ้ากำลังที่มีต่อฟอลต์ (fault)

ขณะที่ระบบไฟฟ้ากำลังอยู่ในสภาวะสมดุล การศึกษาโพลทไฟลว์จะดูเพียงเฟสเดียวก็เพียงพอแล้ว แต่ถ้าระบบไม่สมดุล จะต้องศึกษาทั้ง 3 เฟส ซึ่งกล่าวไว้ในตอนท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1	1
NERMERICAL พื้นฐาน.....	1
1.1 วิธีการกำจัดตัวแปรของเกาส์.....	2
1.2 การเลือกแถวหลัก.....	7
บทที่ 2	9
NETWORK CALCULATION.....	9
2.1 Node equation.....	9
2.2 Node elimination by matrix algebra.....	11
2.3 การตัดทีละบัส.....	12
บทที่ 3	14
LOAD FLOW STUDY.....	14
3.1 Bus classification on power system.....	14
3.2 สมการการไหลของกำลังไฟฟ้าในสายส่ง.....	15
3.3 Control of power into network.....	17
3.4 การควบคุมระบบไฟฟ้ากำลัง.....	19
3.5 Gauss iteration method using Y bus.....	24
3.6 Gauss-Seidel iterative method using Y bus.....	27
3.7 Newton-Raphson method.....	32
3.8 Accelerating of convergence.....	44
3.9 Voltage controlled buses.....	44
3.10 Voltage control at a remote bus.....	48
3.11 วงจรสมมูลของหม้อแปลง.....	49
3.12 Tie line control.....	59
3.13 Comparison of load flow method.....	60
3.14 สรุปและเปรียบเทียบ.....	61
บทที่ 4	63
ระบบไฟสามเฟส.....	63
4.1 การวิเคราะห์ระบบสามเฟส.....	63
4.2 แบบจำลองของสายส่งสามเฟส.....	74
4.3 แบบจำลองของหม้อแปลงสามเฟส.....	84

	หน้า
บทที่ 5 THREE-PHASE LOAD FLOW.....	102
5.1 บทนำ.....	102
5.2 สัญลักษณ์ต่างๆ.....	102
5.3 ปัญหาของสูตรคำนวณที่ใช้ในระบบไฟสามเฟส.....	104
5.4 Fast decouple three-phase load flow.....	108
5.5 โปรแกรม Gauss-siedel.....	122
โปรแกรม Newton-Raphson แบบ rectangular.....	133
โปรแกรม Newton-Raphson แบบ polar.....	146
โปรแกรม Fast decouple single phase.....	162
โปรแกรม Fast decouple three-phase.....	174
5.6 สรุปข้อเปรียบเทียบระหว่างการคำนวณแบบเฟสเดียวและแบบสามเฟส.....	196

ภาคผนวก

รายชื่อเอกสารอ้างอิง

ตัวอย่างเนื้อเรื่องเอกสารอ้างอิง IEEE , Three-phase load flow program

ตัวอย่างเนื้อเรื่องเอกสารอ้างอิง IEEE , Fast-decouple three phase load flow

บทที่ 1

การหาผลเฉลยของระบบสมการเชิงเส้น

การหาผลเฉลยของระบบสมการเชิงเส้น เป็นสิ่งที่สำคัญมากอย่างหนึ่ง ในการประยุกต์คณิตศาสตร์ เพื่อใช้งานในสาขาวิชาอื่น ๆ เกือบทุกสาขา โดยเฉพาะสาขาวิชาที่มีการศึกษาในเชิงวิทยาศาสตร์ การหาผลเฉลยของระบบสมการเชิงเส้นคือ การหาค่า $x_1, x_2, x_3, \dots, x_n$ จากระบบสมการ

$$(1.1) \quad \begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n &= b_3 \\ &\vdots \\ &\vdots \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

ซึ่งเป็นระบบสมการ m สมการและตัวไม่รู้ค่า n ตัว หรือ เขียนอยู่ในรูป

$$(1.2) \quad AX = b$$

$$\text{เมื่อ } A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}_{m \times n}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}_{n \times 1}, \quad \text{และ } b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{bmatrix}_{m \times 1}$$

จากทฤษฎีทางพีชคณิตเชิงเส้น ระบบสมการเชิงเส้น (1.2) จะมีผลเฉลยหรือไม่เป็นไปตามเงื่อนไขต่อไปนี้

1. ถ้า $m < n$ แล้วระบบสมการเชิงเส้น (1.2) จะมีผลเฉลยเสมอ และมีผลเฉลยเอกสารถีนี้เป็นเอกสารถีที่สวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นจำนวนน้อยอันดับ

2. ถ้า $m > n$ แบ่งได้เป็น 3 กรณีดังนี้

2.1 ถ้าค่าลำดับชั้น $A =$ ค่าลำดับชั้น $[A:b] = n$ แล้วระบบสมการเชิงเส้น (2) จะมีผลเฉลยเพียงชุดเดียว

2.2 ถ้าค่าลำดับชั้น $A =$ ค่าลำดับชั้น $[A:b] < n$ แล้วระบบสมการเชิงเส้น (2) จะมีผลเฉลยเสมอและผลเฉลยเป็นจำนวนอนันต์ชุด

2.3 ถ้าค่าลำดับชั้น $A <$ ค่าลำดับชั้น $[A:b]$ แล้วระบบสมการเชิงเส้น (2) จะไม่มีผลเฉลยเลย

สำหรับในกรณีที่ $b = (0)$ จะได้ระบบสมการเชิงเส้นเป็น

$$(1.3) \quad Ax = 0$$

ระบบสมการเชิงเส้น (1.3) จะมีผลเฉลยเสมอ และผลเฉลยหนึ่งคือ $x=(0)$ นั่นคือ $x_1 = x_2 = x_3 = \dots = x_n = 0$ และระบบนี้จะมีผลเฉลยเดียว (คือ (0)) เมื่อค่าลำดับชั้น $A = n$ ถ้าค่าลำดับชั้น A น้อยกว่า n แล้วระบบนี้จะมีผลเฉลยเป็นจำนวนอนันต์ชุด
 สิ่งที่จะศึกษาต่อไปนี้จะเป็นการหาวิธีการเพื่อหาผลเฉลยของระบบสมการเชิงเส้นโดยจะกระทำรวมทั้งทั้งระบบ (1.2) และ (1.3) ซึ่งจะเน้นเฉพาะในกรณีที่ m เท่ากับ n

1.1 วิธีการกำจัดตัวแปรของเกาส์

วิธีการกำจัดตัวแปรของเกาส์ เป็นวิธีการที่เปลี่ยนรูปจากระบบสมการ $Ax = b$ เป็นระบบสมการ $Bx = c$ โดยที่ A เป็น $n \times n$ เมทริกซ์ใด ๆ และ B เป็น $n \times n$ เมทริกซ์สามเหลี่ยม จากระบบสมการ $Bx = c$ นี้จะสามารถหาค่า x_n ได้ทันที ต่อจากนั้นจึงใช้ค่า x_n แทนในสมการถัดขึ้นไปได้ค่า x_{n-1} และนำเอาค่าที่ได้แทนในสมการถัดขึ้นไปเรื่อย ๆ จนได้ค่าของ x_k 's ทุกค่า วิธีการแทนค่าในสมการถัดขึ้นไปนี้เรียกว่า การแทนค่าย้อนกลับ ด้วยเหตุนี้ จึงมีผู้เรียกวิธีนี้ว่า การกำจัดตัวแปรของเกาส์ และแทนค่าย้อนกลับ

วิธีการกำจัดตัวแปรของเกาส์จะเปลี่ยนรูปแบบที่ (1.1) เป็นรูปแบบที่ (1.2)
 ข้างล่างนี้

$$\begin{aligned}
 b_{11}x_1 + b_{12}x_2 + b_{13}x_3 + \dots + b_{1n}x_n &= c_1 \\
 b_{22}x_2 + b_{23}x_3 + \dots + b_{2n}x_n &= c_2 \\
 b_{33}x_3 + \dots + b_{3n}x_n &= c_3 \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 b_{n-1n-1}x_{n-1} + b_{n-1n}x_n &= c_{n-1} \\
 b_{nn}x_n &= c_n
 \end{aligned}$$

เมื่อแทนค่าย้อนกลับจะได้

$$x_n = \frac{c_n}{b_{nn}}$$

$$x_{n-1} = \frac{c_{n-1} - b_{n-1n}x_n}{b_{n-1n-1}}$$

...

...

$$x_e = \frac{c_e - b_{en}x_n - b_{e,n-1}x_{n-1} - \dots - b_{e3}x_3}{b_{ee}}$$

และ

$$x_1 = \frac{c_1 - b_{1n}x_n - b_{1,n-1}x_{n-1} - \dots - b_{1e}x_e}{b_{11}}$$

วิธีการกำจัดตัวแปรของเกาส์ อาจเขียนเป็นขั้นตอน เพื่อสะดวกในการเขียนเป็นโปรแกรมคอมพิวเตอร์ได้ โดยแบ่งออกเป็นสองตอน คือตอนกำจัดตัวแปร และตอนแทนค่าย้อนกลับ ดังนี้

ตอนกำจัดตัวแปร (ขั้นตอนนี้ให้รูปแบบที่ (1.4))

สำหรับ i จาก 1 ถึง $n-1$

ให้ $k = i + 1$

สำหรับ j จาก k ถึง n

ให้ $c = a_{i,k}/a_{i,i}$

สำหรับ 1 จาก k ถึง n

$$a_{i,j} = a_{i,j} - ca_{i,i}$$

กลับไปค่า 1 ตัวถัดไป

$$b_i = b_i - c \cdot b_i$$

กลับไปค่า j ตัวถัดไป

กลับไปค่า i ตัวถัดไป

ตอนแทนค่าย้อนกลับ

$$x(n) = b_n/a_{nn}$$

สำหรับ i จาก 1 ถึง $n - 1$

ให้ $m = n - 1$

$$k = n - i + 1$$

สำหรับ j จาก k ถึง n

$$b_m = b_m - x_j \cdot a_{m,j}$$

กลับไปค่า j ตัวถัดไป

$$x_m = b_m/a_{mm}$$

กลับไปค่า i ตัวถัดไป

การหาผลเฉลยของระบบสมการเชิงเส้น โดยวิธีการกำจัดตัวแปรของเกาส์มีปัญหาและสิ่งที่จะต้องปรับปรุงในหลายกรณี ซึ่งจะได้เริ่มศึกษาเป็นบางกรณีดังต่อไปนี้

กรณีแรก เกิดจากระหว่างที่ทำการกำจัดตัวแปรนั้น ค่าของ $a_{i,k}$ ที่ได้ใหม่บางค่าจะเป็นศูนย์ ซึ่งจะทำให้การหาค่า c และการแทนค่าย้อนกลับทำไม่ได้ ดังนั้นจะต้องหาทางแก้ไข โดยการสลับสมการกันเสียใหม่

กรณีที่สอง ระบบสมการเชิงเส้น $Ax = b$ ที่ให้มานี้ เมทริกซ์ A จะมีค่า $a_{i,i}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นศูนย์เป็นจำนวนมาก ทำให้การใช้วิธีการกำจัดตัวแปรของเกาส์ เสียเวลาคำนวณโดยไม่จำเป็นมาก (ทั้งนี้เพราะใช้เครื่องคอมพิวเตอร์ในการคำนวณ) ดังนั้น จึงควรจะตัดแปลงวิธีการกำจัดตัวแปรของเกาส์ให้เหมาะสมขึ้น และการตัดแปลงนี้จะทำให้ง่ายเมื่อเลขศูนย์ของค่า $a_{k,1}$ ปรากฏขึ้นอย่างสม่ำเสมอ ดังเช่นในระบบของรูปแบบ (1.5) ข้างล่างนี้ ซึ่งเรียกระบบนี้ว่าระบบสามแนวเฉียง

$$\begin{array}{rcl}
 d_1 x_1 + c_1 x_2 & = & b_1 \\
 a_2 x_1 + d_2 x_2 + c_2 x_3 & = & b_2 \\
 & a_3 x_2 + d_3 x_3 + c_3 x_4 & = & b_3 \\
 (1.5) & & a_4 x_3 + d_4 x_4 + c_4 x_5 & = & b_4 \\
 & & \cdot & & \cdot \\
 & & \cdot & & \cdot \\
 & & \cdot & & \cdot \\
 & & a_{n-1} x_{n-2} + d_{n-1} x_{n-1} + c_{n-1} x_n & = & b_{n-1} \\
 & & a_n x_{n-1} + d_n x_n & = & b_n
 \end{array}$$

จากรูปแบบ (1.5) เมื่อใช้วิธีการกำจัดตัวแปรของเกาส์ค่า $c_{k,s}$ จะไม่เปลี่ยนแปลง เลข ค่าที่จะต้องเปลี่ยนแปลงคือ ค่า $d_{k,s}$ และ $b_{k,s}$ และจะเปลี่ยนดังนี้

จาก $k = 2$ ถึง n

$$\text{ให้ } d_k = d_k - \frac{a_k \cdot c_{k-1}}{d_{k-1}} \quad \text{และ} \quad b_k = b_k - \frac{a_k \cdot b_{k-1}}{d_{k-1}}$$

ดังนั้นรูปแบบ (1.5) จะเปลี่ยนเป็นรูปแบบ (1.6)

$$\begin{aligned}
 d_1 x_1 + c_1 x_2 &= b_1 \\
 d_2 x_2 + c_2 x_3 &= b_2 \\
 d_3 x_3 + c_3 x_4 &= b_3 \\
 d_4 x_4 + c_4 x_5 &= b_4 \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 d_{n-1} x_{n-1} + c_{n-1} x_n &= b_{n-1} \\
 d_n x_n &= b_n
 \end{aligned}$$

ใช้การแทนค่าย้อนกลับ โดยให้ $x_n = \frac{b_n}{d_n}$ และ

จาก $k = n - 1$ ถึง 1

$$x_k = \frac{b_k - c_k \cdot x_{k+1}}{d_k}$$

วิธีการข้างต้นจะลดขั้นตอนในการคำนวณของคอมพิวเตอร์ จากวิธีการกำจัดตัวแปรของเกาส์ไปได้มาก

เมตริกซ์ที่มีสมาชิกเป็นศูนย์มากๆ และเป็นระเบียบ เรียกเมตริกซ์นี้ว่า สปาร์สเมตริกซ์ ซึ่งจะมีการศึกษาถึงการแก้ปัญหาระบบสมการเชิงเส้นเช่นนี้อย่างละเอียดโดยเฉพาะ

กรณีที่สาม เป็นการกำจัดตัวแปรในแต่ละสมการให้เหลือเพียงตัวแปรเดียว จากรูปแบบที่ (1.1) จะลดรูปแบบเป็น

$$\begin{array}{rcl}
 b_{11}x_1 & = & c_1 \\
 & & b_{22}x_2 & = & c_2 \\
 (1.7) & & b_{33}x_3 & = & c_3 \\
 & & \cdot & & \cdot \\
 & & \cdot & & \cdot \\
 & & \cdot & & \cdot \\
 & & b_{nn}x_n & = & c_n \\
 \text{และ} & & x_1 & = & \frac{c_1}{b_{11}}
 \end{array}$$

โดยวิธีนี้ไม่มีการแทนค่าย้อนกลับและวิธีนี้เรียกว่าวิธีลดทอนตัวแปรของเกาส์-จอร์แดน

1.2 การเลือกแถวหลัก

โดยทั่ว ๆ ไปการหาผลเฉลยของระบบสมการเชิงเส้น ตามแบบการกำจัดตัวแปรของเกาส์ หรือวิธีอื่น ๆ มักจะกระทำโดยไม่มีการสลับสมการ (ยกเว้นในบางกรณี) ซึ่งในบางครั้งจะให้ผลที่คลาดเคลื่อน ดังจะได้ยกตัวอย่างให้เห็นจริงต่อไป การสลับสมการคือ การสลับแถวของเมตริกซ์ ในรูปแบบที่ (1.2) นอกจากสลับแถวแล้วยังมีการสลับสดมภ์อีกด้วย ในกรณีที่มีการสลับเฉพาะแถวจะเรียกว่าการเลือกแถวหลัก ถ้ามีการสลับทั้งแถวและสดมภ์จะเรียกว่า การเลือกแถวและสดมภ์หลัก ทั้งนี้เนื่องจากจะเข้าใจถึงเหตุผลในหัวข้อเรื่องต่อไป

ข้างล่างนี้ เป็นตัวอย่างที่แสดงให้เห็นค่าคลาดเคลื่อน ของผลที่ได้จากการหาผลเฉลยของระบบสมการเชิงเส้นที่ไม่มีการเลือกแถวหลัก

ให้ระบบสมการเชิงเส้น

$$0.0003x_1 + 0.3371x_2 = 1.0073$$

$$0.0637x_1 + 0.0889x_2 = 1.0073$$

ซึ่งผลเฉลยของระบบสมการนี้คือ $x_1 = 20$ และ $x_2 = 3$ แต่ถ้าไม่มีการเลือกแถวหลัก และใช้วิธีการกำจัดตัวแปรของเกาส์โดยคิดเลขทศนิยมสี่ตำแหน่งแล้วจะได้

$$\begin{array}{rcl}
 0.0003x_1 + 0.3371x_2 & = & 1.0073 \\
 & & 0.3375x_2 & = & 1.0057 \\
 \dots & & x_2 & = & 2.6836 \\
 & & 0.0003x_1 & = & 0.2127 \\
 \dots & & x_1 & = & 709
 \end{array}$$

ผลเฉลยที่ได้คือ $x_1 = 709$ และ $x_2 = 2.6836$ ซึ่งคลาดเคลื่อนไปมาก แต่ถ้ามีการเลือก
แถวหลัก โดยใช้แถวที่สองเป็นแถวหลัก แล้วผลที่ได้จะเป็น

$$\begin{array}{rcl}
 0.0637x_1 - 0.0889x_2 & = & 1.0073 \\
 0.0003x_1 + 0.3371x_2 & = & 1.0073 \\
 \dots & & 0.0637x_1 - 0.0889x_2 & = & 1.0073 \\
 & & 0.0016x_2 & = & 0.0047 \\
 \dots & & x_2 & = & 2.9375 \\
 \dots & & 0.0637x_1 & = & 1.2684 \\
 \dots & & x_1 & = & 19.912
 \end{array}$$

และผลเฉลยที่ได้คือ $x_1 = 19.9122$ และ $x_2 = 2.9375$ ซึ่งใกล้เคียงกว่าเดิมมาก

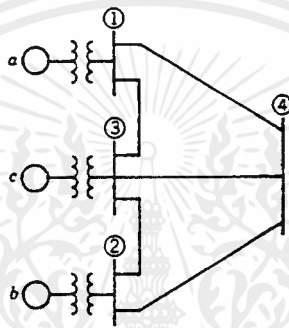
การเลือกแถวหลักข้างต้น เป็นการเลือกแถวหลัก ของระบบสมการเชิงเส้นสองสมการ
ซึ่งไม่ยุ่งยาก สำหรับการเลือกแถวหลักของระบบสมการเชิงเส้น ที่มีมากกว่าสองสมการจะยุ่งยากขึ้น



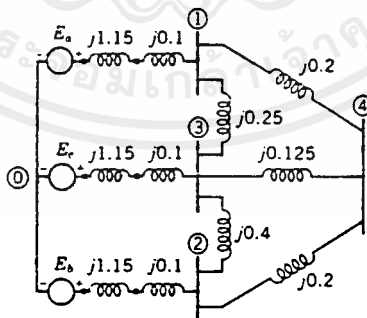
NETWORK CALCULATIONS

2.1 Node Equation

พิจารณารูปที่ 2.1 ซึ่งเป็น one line diagram ของระบบไฟฟ้ากำลังระบบหนึ่ง นำ มาเขียนเป็น reactance diagram ได้ดังรูปที่ 2.2

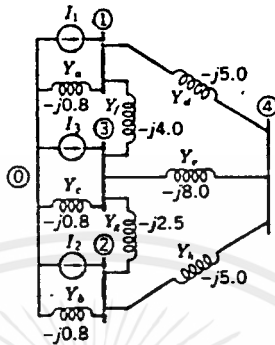


รูป 2.1 one line diagram



รูป 2.2 reactance diagram for system of fig 2.1 in per unit

โดยอาศัยการเปลี่ยน voltage source ไปเป็น current source และเปลี่ยนค่าอิมพีแดนซ์ไปเป็นค่าแอดมิตแตนซ์ จะได้ดังรูปที่ 3



รูป 2.3 circuit of fig 2.2 with current source replacing voltage source

ตัวห้อยท้ายแสดงถึง voltage แต่ละบัสเมื่อเทียบกับ neutral ซึ่งใช้เป็นจุดอ้างอิงคือ node 0 โดยใช้ Kirchhoff's current law ที่แต่ละ node ซึ่งต่อไปเราจะเรียกว่า บัส (bus) แทน จะได้สมการต่าง ๆ ดังนี้

$$\text{บัสที่ 1, } I_1 = VY_a + (V_1 - V_3)Y_f + (V_1 - V_4)Y_d$$

$$\text{บัสที่ 2, } I_2 = V_2Y_b + (V_2 - V_3)Y_g + (V_2 - V_4)Y_h$$

$$\text{บัสที่ 3, } I_3 = V_3Y_c + (V_3 - V_1)Y_f + (V_3 - V_2)Y_g + (V_3 - V_4)Y_e$$

$$\text{บัสที่ 4, } I_4 = (V_4 - V_1)Y_d + (V_4 - V_2)Y_h + (V_4 - V_3)Y_e$$

สมการจะมีอยู่ 4 สมการ เพราะไม่มีสมการที่ neutral bus สามารถเขียนใหม่ได้เป็น

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$$

หรือ

$$[I] = [Y] [V]$$

[Y] เรียกว่า bus admittance matrix

$Y_{11}, Y_{22}, Y_{33}, Y_{44}$ เรียกว่า self admittance ส่วนสมาชิกตัวอื่นๆ ในเมตริกซ์ เรียกว่า mutual admittance

self admittance หาได้จากผลบวกของ admittance ที่ต่อกับบัสนั้นๆ ส่วน mutual admittance หาได้จากค่าลบของ admittance ที่ต่อระหว่างบัสทั้งสอง

และจาก $[I] = [Y][V]$

และจะได้สมการทั่วไปดังนี้

$$I_k = Y_{kn} V_n$$

2.2 Node eliminaton by matrix algebra

จากสมการ $[I] = [Y][V]$ ถ้าหากบัสใดไม่มีกระแสเข้าหรือออก เราสามารถจะตัดสมการของบัสออกไปจากเมตริกซ์ได้ ดังนี้

$$\begin{bmatrix} I_A \\ I_x \end{bmatrix} = \begin{bmatrix} K & L \\ L^T & M \end{bmatrix} \begin{bmatrix} V_A \\ V_x \end{bmatrix}$$

เมื่อ I_A คือกระแสของบัสที่ต้องการให้คงอยู่ (retained bus) ส่วน I_x คือกระแสของบัสที่ตัดทิ้ง (eliminated bus) นั่นคือ I_x เป็นศูนย์

$$I_A = K V_A + L V_x$$

$$I_x = L^T V_A + L V_x \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่า $I_x = 0$ ดังนั้น $V_x = -M^{-1} L V_A$
 แทนค่า V_x ในสมการที่ (2.1) จะได้ว่า
 $I_A = K V_A - LM^{-1} L^T V_A$

ดังนั้น bus admittance ที่เหลืออยู่จะมีค่าดังนี้

$$Y_{bus} = K - LM^{-1} L^T$$

ซึ่งสามารถนำไปวางจรวดใหม่เพื่อใช้ในการคำนวณ

2.3 การตัดทีละบัส

จากที่กล่าวมาแล้ว จะเห็นว่าหากตัดบัสมากกว่า 1 บัสทิ้งพร้อม ๆ กันจะต้องเสียเวลาในการอินเวสเมตริกซ์ M ดังนั้น ถ้าหากตัดทีละบัส เมตริกซ์ M จะมีสมาชิกเพียงตัวเดียวทำให้หาอินเวสได้ง่าย และ admittance matrix เดิมซึ่งแบ่งเป็น submatrix K, L, L^T และ M จะได้

$$Y_{bus} = \begin{array}{c} K \\ \left[\begin{array}{cccc|c} Y_{11} & \dots & Y_{1j} & \dots & Y_{1n} \\ \vdots & & \vdots & & \vdots \\ Y_{k1} & \dots & Y_{kj} & \dots & Y_{kn} \\ \vdots & & \vdots & & \vdots \\ \hline Y_{n1} & \dots & Y_{nj} & \dots & Y_{nn} \\ & & & & \vdots \end{array} \right] \\ \begin{array}{cc} L^T & M \end{array} \end{array}$$

จากสมการ bus admittance จะลดเหลือ $(n-1) \times (n-1)$ matrix

$$Y_{bus} = \left[\begin{array}{cccc} Y_{11} & \dots & Y_{1j} & \dots \\ \vdots & & \vdots & \\ Y_{k1} & \dots & Y_{kj} & \dots \\ \vdots & & \vdots & \end{array} \right] - \frac{1}{Y_{nn}} \begin{bmatrix} Y_{1n} \\ \vdots \\ Y_{kn} \\ \vdots \end{bmatrix} [Y_{n1} \dots Y_{nj} \dots]$$

สมาชิกของเมตริกซ์ในแถว k และคอลัมน์ j ของ $(n-1) \times (n-1)$ matrix จะเป็น

$$Y_{k,j} \text{ (new)} = Y_{k,j} \text{ (orig)} - \frac{Y_{kn} Y_{nj}}{Y_{nn}}$$

เมื่อ $Y_{k,j} \text{ (new)}$ คือสมาชิกตัวใหม่ของ bus admittance matrix ขนาด $(n-1) \times (n-1)$
 $Y_{k,j} \text{ (orig)}$, Y_{kn} , Y_{nj} และ Y_{nn} คือสมาชิกค่าเดิมใน bus admittance matrix
 ขนาด $n \times n$



บทที่ 3

LOAD FLOW STUDY

3.1 Bus classification on Power System

ในการคำนวณเกี่ยวกับ load flow เรานิยมที่จะแบ่งบัสในระบบไฟฟ้ากำลังออกเป็น 3 บัสหลัก ๆ ดังนี้

1. Swing Bus หรือ Slack Bus

Swing bus หรือ slack bus เป็นบัสที่มีเครื่องกำเนิดไฟฟ้าต่ออยู่ เป็นบัสที่ควบคุมให้ voltage magnitude และ phase angle (V, δ) คงที่ตลอดเวลาโดยทั่วไปจะให้ ขนาด $|V| = 1.0$ p.u. และ phase angle $\delta = 0^\circ$ 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 (δ) ที่บัสนั้นออกมา

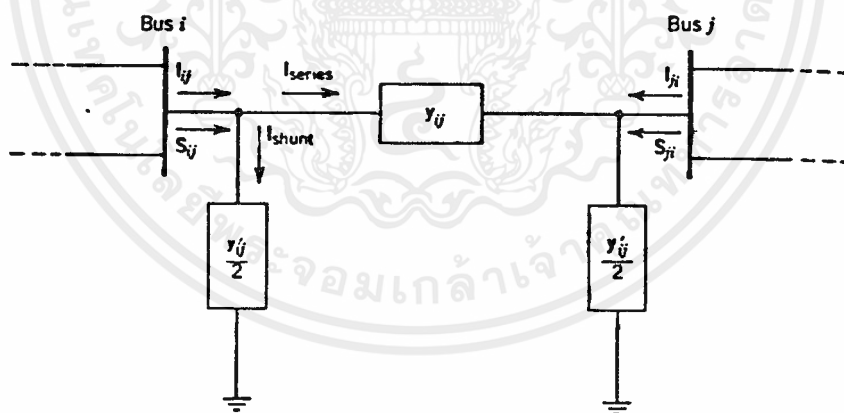
3. Voltage Control Bus

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	δ	P	Q
load bus	P	Q	V	δ
voltage control bus	P	V	Q	δ

3.2 สมการการไหลของกำลังไฟฟ้าในสายส่ง (Line Flow Computation)

หลังจากจบการอิตเทชันแล้ว ได้ค่าแรงดันที่บัสต่าง ๆ แล้ว จึงทำการคำนวณหากำลังไฟฟ้าที่ไหลเข้า-ออกจากบัสจากรูป พิจารณาสายส่งที่เชื่อมต่อระหว่างบัส i และ j



รูป 3.1 แสดง Power flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_{i,j} = I_{series} + I_{shunt} \quad (3.2.1)$$

$$I_{series} = (V_i - V_j) Y_{i,j} \quad (3.2.2)$$

$$I_{shunt} = V_i \frac{Y_{i,j}}{2} \quad (3.2.3)$$

ดังนั้น

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

เมื่อ

$$Y_{i,j} = \text{admittance of line } ij$$

$$Y'_{i,j} = \text{total line-charging admittance}$$

ค่า real และ reactive power ที่ไหลจากบัส i ไป j สามารถแสดงได้ดังนี้

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

โดยการแทน (3.2.4) ใน (3.2.5)

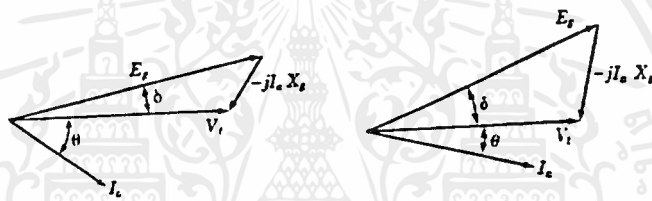
$$S_{i,j} = P_{i,j} + jQ_{i,j} = V_i (V_i^* - V_j^*) Y_{i,j} + V_i V_i^* \left(\frac{Y_{i,j}}{2} \right)^*$$

เมื่อ

$$S_{i,j} = \text{Line Power Flow จาก bus } i \text{ ไป bus } j$$

3.3 Control of Power into Network

จากรูปวงจรที่แสดงไว้ แสดงถึงการส่งจ่ายกำลังไฟฟ้าจาก synchronous generator ผ่านสายส่งไปยังระบบไฟฟ้ากำลังอย่างง่าย ๆ เราสามารถหาความสัมพันธ์ต่าง ๆ ได้ดังนี้



รูป 3.2 การควบคุมกำลังไฟฟ้า

$$\begin{aligned}
 I &= \frac{E_g / \angle \delta - V_t / 0}{Z / \theta} \\
 &= \frac{E_g / \angle \delta - \cancel{V_t} / -\theta}{Z} \\
 S_A &= E_g \cdot I^* \\
 &= E_g \frac{1}{Z} \left(\frac{E_g}{\angle \delta} - \frac{V_t}{\angle -\theta} \right)^* \\
 &= E_g \frac{1}{Z} \left(\frac{E_g}{\angle -\delta} - \frac{V_t}{\angle \theta} \right)
 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \frac{E_s^2}{Z} \frac{1}{\theta} - \frac{E_s V_s}{Z} \frac{1}{\theta + \phi}$$

ดังนั้น

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

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

ในทำนองเดียวกัน

$$S_B = \frac{V_s}{Z} \frac{1}{\theta} (\frac{V_s}{Z} \frac{1}{-\theta} - \frac{E_s}{Z} \frac{1}{-\theta})^*$$

$$= \frac{V_s^2}{Z} \frac{1}{\theta} - \frac{E_s}{Z} \frac{1}{\phi - \theta}$$

ดังนั้น

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

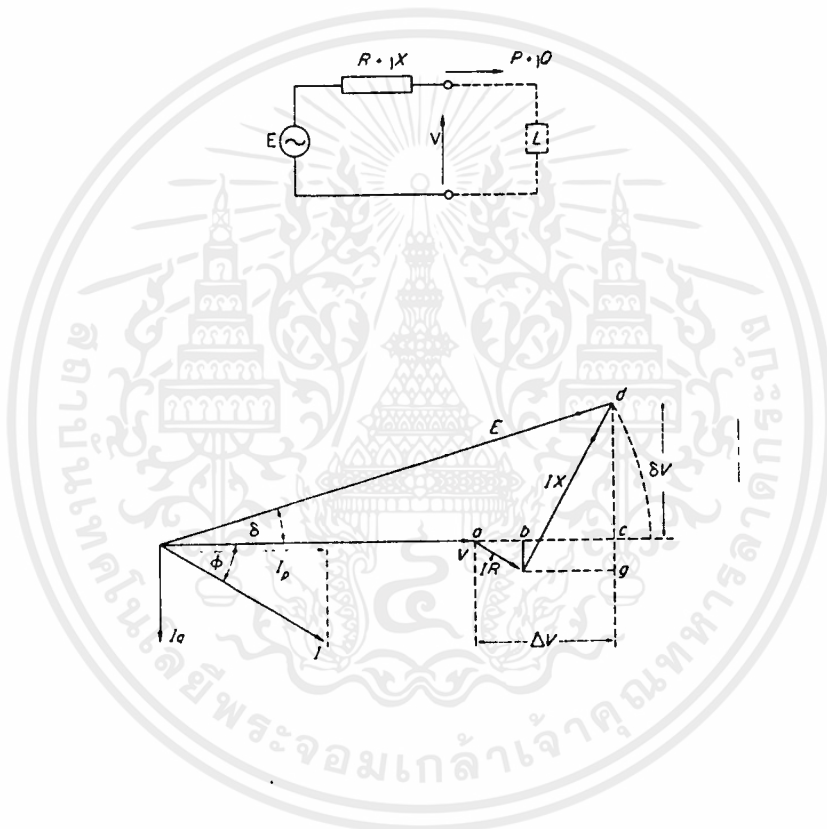
$$Q_B = \frac{V_s^2}{Z} \sin \theta - \frac{V_s E_s}{Z} \sin(\theta - \phi)$$

ค่า power output ที่จ่ายไปยัง B จะสูงสุดเมื่อ $\cos(\theta - \phi) = 1$ หรือ $\theta = \phi$

3.3 การควบคุมระบบไฟฟ้ากำลัง

Control Voltage and Reactive Power

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



รูป 3.3 Phasor diagram for transmission of power through a series impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 3.3

$$\begin{aligned}
 E^2 &= (V + \Delta V)^2 + \mathcal{I}V^2 \\
 &= (V + RI \cos\phi + XI \sin\phi)^2 + (XI \cos\phi - RI \sin\phi)^2 \\
 E^2 &= \left(V + \frac{RP}{V} + \frac{XQ}{V}\right)^2 + \left(\frac{XP}{V} - \frac{RQ}{V}\right)^2
 \end{aligned}$$

ดังนั้น

$$\Delta V = \frac{RP}{V} + \frac{XQ}{V}$$

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

$$\mathcal{I}V = \frac{XP}{V} - \frac{RQ}{V}$$

ถ้า R มีค่าน้อยมาก

$$\delta V \propto Q$$

การเปลี่ยนแปลงของ rotor ใน synchronous generator จะขึ้นกับ real power และมีผลต่อความถี่ด้วย

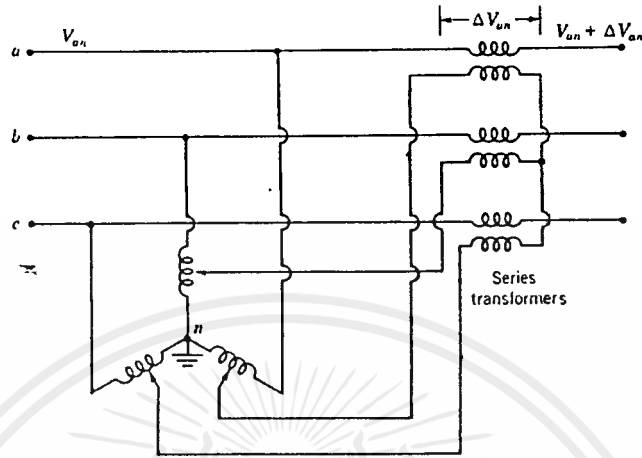
Control By Transformer

ในการปฏิบัติงานในระบบไฟฟ้ากำลัง เมื่อแรงดันด้าน primary ของหม้อแปลงต่ำลงไปต่ำกว่าเกณฑ์เนื่องจากภาระไฟฟ้าทางด้าน secondary ต้องการกระแสมาก จึงมีความจำเป็นต้องเปลี่ยนจำนวนรอบของขดลวดด้าน primary โดยใช้หม้อแปลงที่เปลี่ยนจำนวนรอบของขดลวดได้ (tap-changing transformer) ในการคำนวณโหลดโพล์ เมื่อคำนวณแล้วแรงดันที่บัสยังไม่เป็นที่ต้องการ หรือไม่มีโอกาสสะดวกการการอิทธิพลเรชั่น โปรแกรมจะทำการเปลี่ยน จำนวนรอบของขดลวด แล้วคำนวณใหม่จนกว่าจะได้แรงดันตามที่ต้องการ

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

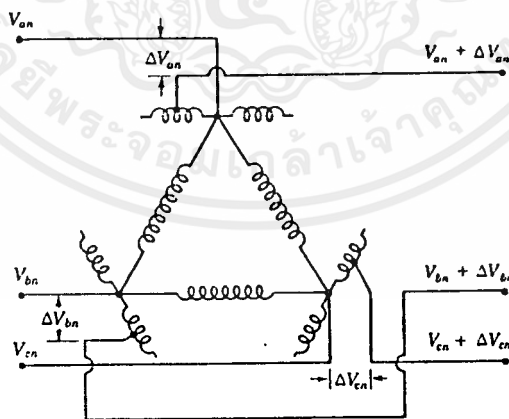
Regulating Transformer

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



รูป 3.4 Regulating transformer for control magnitude

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



รูป 3.5 Regulating transformer for control phase angle

และจากที่กล่าวมาแล้วว่า $\Delta V \propto P$ และ $\Delta V \propto Q$ ดังนั้น load-tap-changing transformer ซึ่งมีหน้าที่เปลี่ยนแปลงแรงดัน จึงทำหน้าที่ควบคุม real power ส่วน phase-shifting transformer ซึ่งทำหน้าที่เปลี่ยน phase จึงควบคุม reactive power

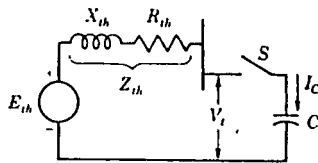
Capacitor Banks

วิธีที่ควบคุม bus voltages ที่สำคัญคือการใช้ shunt capacitor bank โดยอาจจะติดตั้งไว้ที่สายส่ง และระบบจำหน่าย หรือติดตั้งที่สถานีย่อยและโหลด capacitor จะเป็นตัวจ่าย vars ซึ่งอาจจะติดตั้งอย่างถาวรแต่ใช้ปรับ voltage โดยการ switched on หรือ off ออกจากระบบตามความต้องการของโหลด โดยสวิชต์นี้อาจตัดต่อด้วยมือหรือแบบอัตโนมัติ โดยใช้ในการตั้งเวลา หรือลดความต้องการของ voltage หรือ reactive power เมื่อต่อ capacitor หนานกับโหลด ซึ่งเป็นแบบตัวประกอบกำลังล้าหลัง capacitor จะเป็นแหล่งจ่าย reactive power ไปยังโหลด ซึ่งจะลด line current ที่จ่ายไปยังโหลดและลด voltage drop ในสายทำให้ power factor ดีมาก เพราะว่า capacitors จะไปลด reactive power ที่จ่ายออกมา generator

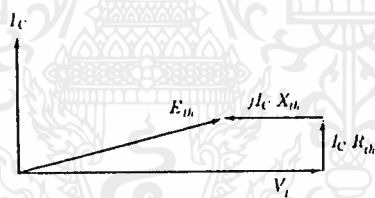
ใน load-flow computer program เราสามารถกำหนดค่าที่ slack bus ซึ่งเป็นแหล่งจ่าย reactive-power แต่ที่ load bus ซึ่งไม่ generator capacitor banks จะถูกติดตั้งไว้ และคอมพิวเตอร์จะคำนวณหาค่า Q ได้ตามต้องการ

ในการติดตั้ง capacitors ไว้ที่ node ใด ๆ ค่า voltage ที่เพิ่มขึ้นจะหาได้โดยใช้ Thevenin's theorem จากรูปเป็นการแทนระบบไฟฟ้ากำลังด้วย Thevenin equivalent ที่ node ซึ่งติดตั้ง capacitors โดยการใช้วิธีปิดสวิชต์ ขณะที่เปิดสวิชต์ node voltage V_n จะเท่ากับ Thevenin voltage E_{th} เมื่อปิดสวิชต์กระแสที่ผ่าน capacitors จะหาได้จาก

$$I_c = \frac{E_{th}}{Z_{th} - jX_c}$$



รูป 3.6 การต่อคาปาซิเตอร์เข้าไปในวงจร



รูป 3.7 เฟสเซอร์ไดอะแกรมของรูป 3.6

จาก phasor diagram การเพิ่มขึ้นของ V_t โดย capacitor มีค่าใกล้เคียงกับ I_c / X_{th} นอกจากนี้เรายังควบคุม voltage ได้ด้วยการควบคุมการกระตุ้นของ synchronous generator ซึ่งทำให้ควบคุมการจ่าย reactive power ระหว่าง generator กับระบบได้ เช่น เมื่อเป็น overexcited generator) จะจ่าย Q reactive power ให้กับระบบ ซึ่งจะทำหน้าที่คล้ายกับ capacitor คือ เพิ่มแรงดันขึ้น

3.5 Gauss iteration method using Y bus

การหาค่าตอบของไหลดโพล์ด้วยวิธีนี้ เริ่มต้นด้วยการสมมุติว่า voltages ทุก ๆ บัส นอกจาก slack bus ซึ่งได้กำหนดค่า voltage และกำหนดให้คงที่ เราสามารถคำนวณค่ากระแสสำหรับบัสทุกบัสนอกจาก slack บัส จากสมการเหล่านี้

$$I_p = \frac{P_p - jQ_p}{V_p^*}, \quad p = 1, 2, 3, \dots, n$$

$$p \neq s \quad (3.5.1)$$

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

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

ค่า bus currents คำนวณจากสมการ (3.5.1) แทนค่า slack bus voltage และ bus voltage ที่กำหนดค่าประมาณลงในสมการ (3.5.3) เพื่อคำนวณค่า bus voltage ชุดใหม่และค่า bus voltage ชุดใหม่นี้ จะนำไปแทนในสมการ (3.5.1) เพื่อคำนวณหาค่ากระแสชุดใหม่ เพื่อนำไปคำนวณในสมการ (3.5.3) และจะทำซ้ำไปเรื่อย ๆ จนกว่า ค่าความเปลี่ยนแปลงของ bus voltages ทุกแห่ง อยู่ในย่านที่ยอมรับได้ ภายหลังจากคำนวณหาค่า voltage แล้วจะทำให้คำนวณค่า power ของ slack bus และ line flow ได้

สมการของ Network (3.5.3) และ bus loading equation (3.5.1) สามารถ นำมารวมกันได้เป็น

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

(3.5.4)

สมการของ load flow เหล่านี้เป็นสมการแบบ nonlinear ซึ่งสามารถแก้โดยใช้วิธี iteration ในการคำนวณ เราจะลดเวลาที่ใช้ให้มากที่สุด โดยลดขั้นตอนทางคณิตศาสตร์ โดยให้

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

เขียนสมการ (3.5.4) ใหม่ได้ดังนี้

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

(3.5.5)

ให้

$$(P_p - jQ_p) L_p = K L_p$$

และ

$$Y_{pq} L_p = Y L_{pq}$$

และ bus voltage equation (3.5.5) จะกลายเป็น

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

(3.5.6)

ขั้นตอนของการศึกษา load flow โดยสมมุติว่าเป็นระบบสมดุล และใช้วงจรสมดุลแบบเฟสเดียวแบบ positive sequence network เพราะว่าไม่มี mutual coupling ค่า bus admittance matrix หาได้โดยการตรวจสอบ สมาชิกบางตัวเป็นศูนย์ เลือก bus 2 เป็น slack bus ในระบบในรูป สูตรในการคำนวณจะเป็นดังนี้

Busq						
Busp	1	2	3	4	5	6
1	Y_{11}	Y_{12}	Y_{13}	Y_{14}		
2	Y_{21}	Y_{22}			Y_{25}	Y_{26}
3	Y_{31}		Y_{33}		Y_{35}	
4	Y_{41}			Y_{44}		Y_{46}
5		Y_{52}	Y_{53}		Y_{55}	
6		Y_{62}		Y_{64}		Y_{66}

รูป 3.7 Single line diagram and bus admittance matrix of a power system

$$V_1^{k+1} = \frac{KL_1 - YL_{12}V_2 - YL_{13}V_3 - YL_{14}V_4}{(V_1^k)^*}$$

$$V_2 = \text{specified fixed value}$$

$$V_3^{k+1} = \frac{KL_3 - YL_{31}V_1 - YL_{35}V_5}{(V_3^k)^*}$$

$$V_4^{k+1} = \frac{KL_4 - YL_{41}V_1 - YL_{46}V_6}{(V_4^k)^*}$$

$$V_5^{k+1} = \frac{KL_5 - YL_{52}V_2 - YL_{53}V_3}{(V_5^k)^*}$$

$$V_6^{k+1} = \frac{KL_6 - YL_{62}V_2 - YL_{64}V_4}{(V_6^k)^*}$$

เมื่อ superscript k คือ จำนวนการ iteration ลำดับขั้นในการหาค่าตอบของ load flow โดยวิธี Gauss iteration แสดงในรูป

3.6 Gauss-Seidel iterative method using Y_{bus}

สมการ bus voltage (3.5.6) สามารถหาค่าตอบโดย Gauss-Seidel iterative method โดยวิธีนี้ค่า voltage V_b^{k+1} ที่คำนวณได้ใหม่ จากถูกแทนด้วย V_b^k และใช้ในสมการลำดับถัดมา สำหรับระบบที่แสดงในรูปสมการที่ใช้ในวิธีนี้ดังนี้คือ

$$V_1^{k+1} = \frac{K L_1 - YL_{12} V_2 - YL_{13} V_3^k - YL_{14} V_4^k}{(V_1^k)^*}$$

$$V_2 = \text{specified fixed value}$$

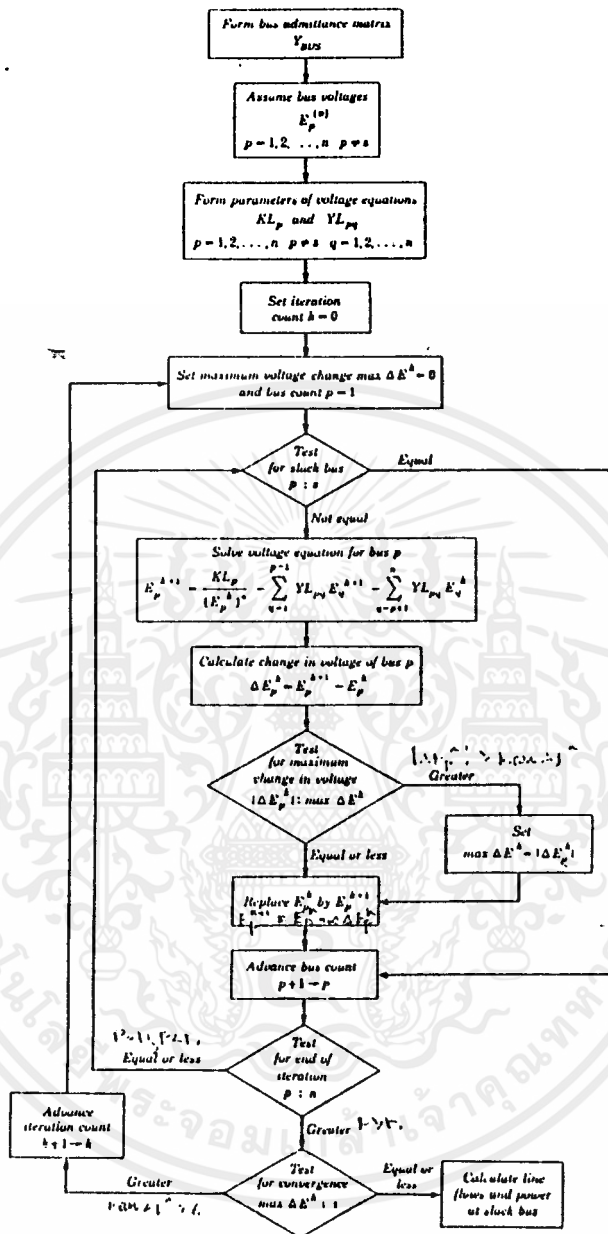
$$V_3^{k+1} = \frac{K L_3 - YL_{31} V_1^{k+1} - YL_{35} V_5^k}{(V_3^k)^*}$$

$$V_4^{k+1} = \frac{K L_4 - YL_{41} V_1^{k+1} - YL_{46} V_6^k}{(V_4^k)^*}$$

$$V_5^{k+1} = \frac{K L_5 - YL_{52} V_2 - YL_{53} V_3^{k+1}}{(V_5^k)^*}$$

$$V_6^{k+1} = \frac{K L_6 - YL_{62} V_2 - YL_{64} V_4^{k+1}}{(V_6^k)^*}$$

ลำดับขั้นในการหาค่าตอบของ load flow โดยวิธี Gauss-Seidel iterative ดังแสดงในรูป 3.9



รูป 3.9 Load Flow solution by the Gauss-seidel iterative method using Y_{bus}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Relaxation method using Y_{bus}

สมการของ bus currents จะใช้ในการหาค่าตอบของ load flow โดย relaxation method จากสมการ (2) กระแสที่ bus p เป็น

$$I_p = Y_{p1}V_1 + Y_{p2}V_2 + \dots + Y_{pp}V_p \dots + Y_{pn}V_n$$

สมการนี้เขียนใหม่ได้เป็น

$$Y_{p1}V_1 + Y_{p2}V_2 \dots + Y_{pp}V_p \dots + Y_{pn}V_n - I_p = R_p$$

เมื่อ R_p เป็นค่าที่เหลืออยู่และเป็นการแทนค่าผิดพลาดของกระแสที่บัส p ซึ่งเป็นผลจากการสมมติค่าตอบของ voltage สำหรับระบบที่แสดงในรูป สมการที่ใช้ใน relaxation method คือ

$$\begin{aligned} Y_{11}V_1^k + Y_{12}V_2^k + Y_{13}V_3^k + Y_{14}V_4^k - I_1^k &= R_1^k \\ Y_{31}V_1^k + Y_{33}V_3^k + Y_{35}V_5^k - I_3^k &= R_3^k \\ Y_{41}V_1^k + Y_{44}V_4^k + Y_{46}V_6^k - I_4^k &= R_4^k \\ Y_{52}V_2^k + Y_{53}V_3^k + Y_{55}V_5^k - I_5^k &= R_5^k \\ Y_{62}V_2^k + Y_{64}V_4^k + Y_{66}V_6^k - I_6^k &= R_6^k \end{aligned} \quad (3.6.1)$$

เมื่อสมการสำหรับ slack bus จะไม่รวมอยู่ด้วย เพราะว่า V_e ถูกกำหนดให้คงที่ ด้วยค่า bus voltage เหล่านี้

$$V_1^{(0)}, V_2^{(0)}, V_3^{(0)}, V_4^{(0)}, V_5^{(0)}, V_6^{(0)}$$

ค่า bus currents คำนวณได้จากสมการ (3.5.1) และบัสที่ยังเหลืออยู่คำนวณจาก สมการ (3.6.1) ค่า voltage correction ที่บัสซึ่ง residual R_p มีค่าสูงสุด ถ้ากระแสที่บัส p ยังคงที่ ค่า residual R_p จะลดลงจนเป็นศูนย์ตาม voltage correction

$$\Delta V_p^k = \frac{-R_p^k}{Y_{pp}}$$

และค่าประมาณของ voltage ที่ปรับปรุงแล้วสำหรับบัส p จะได้ว่า

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

และค่ากระแสที่คำนวณใหม่จะเป็น

$$I_p^{k+1} = \frac{P_p - jQ_p}{(V_p^{k+1})^*}$$

ขณะที่ผลของการเปลี่ยนแปลงกระแส ค่า actual residual ที่บัส p จะเป็น

$$R_p^{k+1} = I_p^k - I_p^{k+1}$$

ใช้ค่า V_p^{k+1} และ new residual สำหรับบัสอื่นนอกจาก p และ slack bus สามารถคำนวณได้จาก

$$R_q^{k+1} = R_q^k + Y_{qp} \Delta V_p^k \quad q = 1, 2, \dots, n$$

$$q \neq p, q \neq S$$

ขั้นตอนนี้จะถูกทำซ้ำ แต่ละครั้งที่ตรวจสอบค่า voltage ซึ่งเกี่ยวข้องกับค่า largest residual จนกระทั่งค่า residual ทั้งหมด จะน้อยกว่าหรือเท่ากับ specified tolerance ลำดับขั้นของการหาค่าตอบของ load flow โดยใช้ relaxation method แสดงในรูป 3.10

3.7 วิธี Newton-Raphson (Newton-Raphson method)

วิธีนิวตัน-ราฟสัน เป็นวิธีที่เร็วและมีประสิทธิภาพมากในการคำนวณโพลีโนเมียล หลักการของวิธีนิวตัน-ราฟสัน ได้จากการกระทบอนุกรมของเทย์เลอร์ (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 นั้นหมายความว่าถ้าหา $\Delta x_{1(0)}$ และ $\Delta x_{2(0)}$ 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

$$k_1 = f_1(x_{1(0)}, x_{2(0)} + \Delta x_{2(0)}) + \left. \frac{\partial f_1}{\partial x_1} \right|_{(0)} \Delta x_{1(0)} + \left. \frac{\partial f_1}{\partial x_2} \right|_{(0)} \Delta x_{2(0)} + \dots \quad (3.7.3)$$

$$k_2 = f_2(x_{1(0)}, x_{2(0)} + \Delta x_{2(0)}) + \left. \frac{\partial f_2}{\partial x_1} \right|_{(0)} \Delta x_{1(0)} + \left. \frac{\partial f_2}{\partial x_2} \right|_{(0)} \Delta x_{2(0)} + \dots \quad (3.7.4)$$

พาเชี่ยลเดอริเวทีฟ (partial derivative) ที่มีอันดับมากกว่า 1 ไม่ได้เขียนไว้ในสมการที่ (3.7.3) และ (3.7.4) ซึ่งถ้าตัดพาเชี่ยลเดอริเวทีฟที่มีอันดับมากกว่า 1 ออกจะเขียนสมการที่ (3.7.3) และ (3.7.4) ในรูปแบบเมตริกซ์ได้

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

เมตริกซ์จัตุรัสของพหุเชิงเส้นเดอริเวทีฟ 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

$$\begin{aligned} k_{1(0)} &= k_1 - f_1(x_{1(0)}, x_{2(0)}) \\ k_{2(0)} &= k_2 - f_2(x_{1(0)}, x_{2(0)}) \end{aligned}$$

นำไปแทนค่าในสมการที่ (3.7.5) ได้

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

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

$$\begin{aligned} x_{1(k)} &= x_{1(0)} + \Delta x_{1(0)} \\ x_{2(k)} &= x_{2(0)} + \Delta x_{2(0)} \end{aligned}$$

จะทำการคำนวณซ้ำจนกว่า x_k และ $x_{(k+1)}$ เข้าใกล้กันมาก คล้ายๆ กับการอิตเทอเรนซ์นั่นเอง

การประยุกต์วิธีนิวตัน-ราฟสัน ในการแก้ปัญหาโพลาร์

สมการสำหรับคำนวณกำลังที่บัส p หาได้จาก

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

ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

แต่เรามีสมการของกระแสคือ

$$I_p = \sum_{\alpha=1}^n Y_{p\alpha} V_\alpha$$

แทนค่าในสมการ (3.7.6) จะได้

$$P_p - jQ_p = V_p^* \sum_{\alpha=1}^n Y_{p\alpha} V_\alpha \quad (3.7.7)$$

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

$$P_p - jQ_p = (e_p - jf_p) \sum_{\alpha=1}^n (G_{p\alpha} - jB_{p\alpha}) (e_\alpha + jf_\alpha)$$

แยกส่วนจริงและส่วนจินตภาพออกจากกัน

$$P_p = \sum_{\alpha=1}^n e_p (e_\alpha G_{p\alpha} + f_\alpha B_{p\alpha}) + f_p (f_\alpha G_{p\alpha} - e_\alpha B_{p\alpha})$$

$$Q_p = \sum_{\alpha=1}^n f_p (e_\alpha G_{p\alpha} + f_\alpha B_{p\alpha}) + e_p (f_\alpha G_{p\alpha} - e_\alpha B_{p\alpha})$$

(3.7.8)

วิธีนิวตัน-ราฟสัน สามารถใช้จัดชุดของสมการไม่เชิงเส้นข้างต้น ให้เป็นสมการเชิงเส้นได้ ซึ่งมีการเปลี่ยนแปลงของกำลังจริงและกำลังเสมือนและส่วนประกอบของบัสโวลเตจ ดังนี้

ΔP_1	$\frac{\partial P_1}{\partial e_1} \dots \frac{\partial P_1}{\partial e_{n-1}}$	$\frac{\partial P_1}{\partial e_1} \dots \frac{\partial P_1}{\partial e_{n-1}}$	Δe_1
...
ΔP_{n-1}	$\frac{\partial P_{n-1}}{\partial e_1} \dots \frac{\partial P_{n-1}}{\partial e_{n-1}}$	$\frac{\partial P_{n-1}}{\partial e_1} \dots \frac{\partial P_{n-1}}{\partial e_{n-1}}$	Δe_{n-1}
ΔQ_1	$\frac{\partial Q_1}{\partial e_1} \dots \frac{\partial Q_1}{\partial e_{n-1}}$	$\frac{\partial Q_1}{\partial e_1} \dots \frac{\partial Q_1}{\partial e_{n-1}}$	Δf_1
...
ΔQ_{n-1}	$\frac{\partial Q_{n-1}}{\partial e_1} \dots \frac{\partial Q_{n-1}}{\partial e_{n-1}}$	$\frac{\partial Q_{n-1}}{\partial e_1} \dots \frac{\partial Q_{n-1}}{\partial e_{n-1}}$	Δf_{n-1}

เมื่อเมตริกซ์สัมประสิทธิ์คือ Jacobian และบัสที่ n เป็น swing bus ในรูปของเมตริกซ์สมการที่ (3.7.4) เขียนได้เป็น

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

สมการที่ใช้ในการหาสมาชิกของจาโคเบียน สามารถพิสูจน์จากสมการกำลังของบัส ค่ากำลังจริงของสมการที่ (3.7.9) คือ

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

คิฟเฟอเรนเชียล สมาชิก off-diagonal ของ J จะได้

$$\frac{\partial P_p}{\partial e_q} = e_p G_{pq} - f_p B_{pq}, \quad q = p$$

และสมาชิก diagonal ของ J_1 คือ

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

(3.7.11)

อย่างไรก็ตาม สมการของกระแสที่บัส p คือ

$$I_p = C_p + j d_p = (G_{pp} - j B_{pp})(e_p + j f_p) + \sum_{q=1}^n (G_{pq} - j B_{pq})(e_q + j f_q)$$

ซึ่งสามารถแยกเป็นส่วนจริงและจินตภาพได้ดังนี้

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

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

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

(3.7.12)

สมการของสมาชิก diagonal ของ J_1 หาโดยแทนส่วนจริงของกระแส C_p ในสมการ (3.7.11) จะได้

$$\frac{\partial P_p}{\partial e_p} = e_p G_{pp} - f_p B_{pp} + c_p$$

จากสมการ (3.7.10) สมาชิก off-diagonal ของ J_2 คือ

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

และสมาชิก diagonal ของ J_2 คือ

$$\frac{\partial P_{pp}}{\partial f_{pp}} = e_{pp} B_{pp} + 2f_{pp} G_{pp} - e_{pp} B_{pp} + \sum_{\substack{q=1 \\ q \neq p}}^n (f_{pq} G_{pq} - e_{pq} B_{pq})$$

(3.7.13)

นำส่วนจินตภาพของกระแสจากสมการที่ (3.7.12) แทนในสมการที่ (3.7.13) จะได้

$$\frac{\partial P_{pp}}{\partial f_{pp}} = e_{pp} B_{pp} + f_{pp} G_{pp} + d_{pp}$$

ค่ากำลังเสมือนจากสมการ (3.7.8) จะเป็น

$$Q_{pp} = f_{pp} (e_{pp} G_{pp} + f_{pp} B_{pp}) - e_{pp} (f_{pp} G_{pp} - e_{pp} B_{pp}) + \sum_{\substack{q=1 \\ q \neq p}}^n f_{pq} (e_{pq} G_{pq} + f_{pq} B_{pq}) - e_{pq} (f_{pq} G_{pq} - e_{pq} B_{pq})$$

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

(3.7.14)

หิพเพอเรนดีเอก สมาชิก off-diagonal ของ J^3 คือ

$$\frac{\partial P_{ppq}}{\partial f_{ppq}} = e_{ppq} P_{ppq} + f_{ppq} P_{ppq}, \quad p \neq q$$

และสมาชิก diagonal ของ J^3 คือ

$$\frac{\partial P_{pp}}{\partial f_{pp}} = e_{pp} P_{pp} - f_{pp} P_{pp} + e_{pp} P_{pp} - \sum_{\substack{q=1 \\ q \neq p}}^n (e_{pq} P_{ppq} - f_{pq} P_{ppq})$$

(3.7.15)

นำส่วนจินตภาพของกระแสจากสมการ (3.7.12) แทนลงในสมการ (3.7.15) จะได้

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

จากสมการที่ (3.7.14) สมาชิกของ off-diagonal ของ J_A คือ

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

และสมาชิก diagonal ของ J_A คือ

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

(3.7.16)

นำส่วนจริงของกระแสจากสมการ (3.7.12) แทนในสมการ (3.7.16) จะได้

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

(3.7.8) กำหนดค่าเริ่มต้นของบัสโวลเตจ ค่ากำลังจริงและกำลังปรากฏจะคำนวณได้จากสมการ การเปลี่ยนแปลงของกำลังบท ได้จากผลต่างค่าที่กำหนด (scheduled) กับค่าที่คำนวณ ได้

$$\begin{aligned} \Delta P_p^k &= P_{p(\text{scheduled})} - P_p^k \\ \Delta Q_p^k &= Q_{p(\text{scheduled})} - Q_p^k \end{aligned}$$

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

ค่าประมาณของโวลเตจที่บัสและกำลังจะใช้ในการคำนวณกระแสบัส เพื่อใช้ในการหาค่าสมาชิกของจาโคเบียน สมการที่ (3.7.9) สามารถให้หาค่า Δe_p และ Δf_p , $p = 1, 2, \dots, n-1$ โดยทางตรงหรือวิธีทำซ้ำ ดังนั้นค่าประมาณอันใหม่ของบัสโวลเตจจะเป็น

$$\begin{aligned} e_p^{k+1} &= e_p^k + \Delta e_p^k \\ f_p^{k+1} &= f_p^k + \Delta f_p^k \end{aligned}$$

ทำซ้ำจนกว่า ΔP_p^k และ ΔQ_p^k ของทุกบัสอยู่ในช่วงที่ยอมรับได้

วิธีนิวตัน-ราฟสัน ยังสามารถประยุกต์เพื่อให้หาค่าตอบของโวลต์โพลาร์เมื่อสมการอยู่

ในรูปพิกัดเชิงขั้วในระบบพิกัดเชิงขั้ว

$$V_p = |V_p| e^{j\theta_p} \quad \text{และ} \quad Y_{pq} = |Y_{pq}| e^{-j(\theta_p - \theta_q)}$$

แทนค่าลงในสมการ (3.7.7) ดังนั้นกำลังที่บัส p คือ

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

เพราะว่า

$e^{-j(\theta_p - \theta_q - \theta_{pq})} = \cos(\theta_p + \theta_{pq} - \theta_q) - j\sin(\theta_p + \theta_{pq} - \theta_q)$ ทั้งนี้ส่วนจริงและส่วนจินตภาพของกำลังจะเป็น

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

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

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

สำหรับ J_1 :

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

$$\frac{\partial P_p}{\partial \theta_p} = \sum_{q=1}^n |V_p V_q Y_{pq}| \sin(\theta_p + \theta_{pq} - \theta_q)$$

สำหรับ J_2 :

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

เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\partial P}{\partial |V_p|} = 2 \left| V_p Y_{pp} \right| \cos \theta_{pp} - \sum_{\substack{q=1 \\ q \neq p}}^n \left| V_q Y_{pq} \right| \cos(\theta_{pq} + \angle V_p - \angle V_q)$$

สำหรับ J_q :

$$\frac{\partial Q}{\partial \angle V_q} = - \left| V_p V_q Y_{pq} \right| \cos(\theta_{pq} + \angle V_p - \angle V_q) \quad q=p$$

$$\frac{\partial Q}{\partial \angle V_q} = - \sum_{\substack{q=1 \\ q \neq p}}^n \left| V_p V_q Y_{pq} \right| \cos(\theta_{pq} + \angle V_p - \angle V_q)$$

สำหรับ J_4 :

$$\frac{\partial Q}{\partial |V_p|} = \left| V_p Y_{pp} \right| \sin(\theta_{pp} + \angle V_p - \angle V_q) \quad q=p$$

$$\frac{\partial Q}{\partial \angle V_p} = 2 \left| V_p Y_{pp} \right| \sin \theta_{pp} - \sum_{\substack{q=1 \\ q \neq p}}^n \left| V_q Y_{pq} \right| \sin(\theta_{pq} + \angle V_p - \angle V_q)$$

ดังนั้น สมการที่แสดงความสัมพันธ์ของกำลังที่เปลี่ยนไป กับขนาดของโวลเตจและมุมเฟสสำหรับวิธีนิวตันราฟ-สันจะได้

$$\begin{array}{|c|} \hline \Delta P \\ \hline \Delta Q \\ \hline \end{array} = \begin{array}{|c|c|} \hline J_1 & J_2 \\ \hline J_3 & J_4 \\ \hline \end{array} \begin{array}{|c|} \hline \Delta \angle \\ \hline \Delta |E| \\ \hline \end{array}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมาณค่าของวิธีนิวตัน-ราฟสัน

โดยทั่วไป สำหรับการเปลี่ยนแปลงขนาดของบัสโวลเตจที่น้อย ๆ ค่ากำลังจริงที่บัสจะเปลี่ยนแปลงน้อยมาก ทำนองเดียวกันสำหรับการเปลี่ยนแปลงของมุมเฟสค่าน้อย ๆ ของบัสโวลเตจ ค่ากำลังเสมือน จะเปลี่ยนแปลงน้อยมาก ดังนั้น โดยใช้ระบบพิกัดเชิงขั้ว ค่าตอบของปัญหาโพลดิโพลว์ สามารถหาโดยสมมติว่าสมาชิกของเมตริกซ์ย่อย J_e และ J_o เป็นศูนย์ ดังนั้นจะได้สมการเมตริกซ์ ดังนี้

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

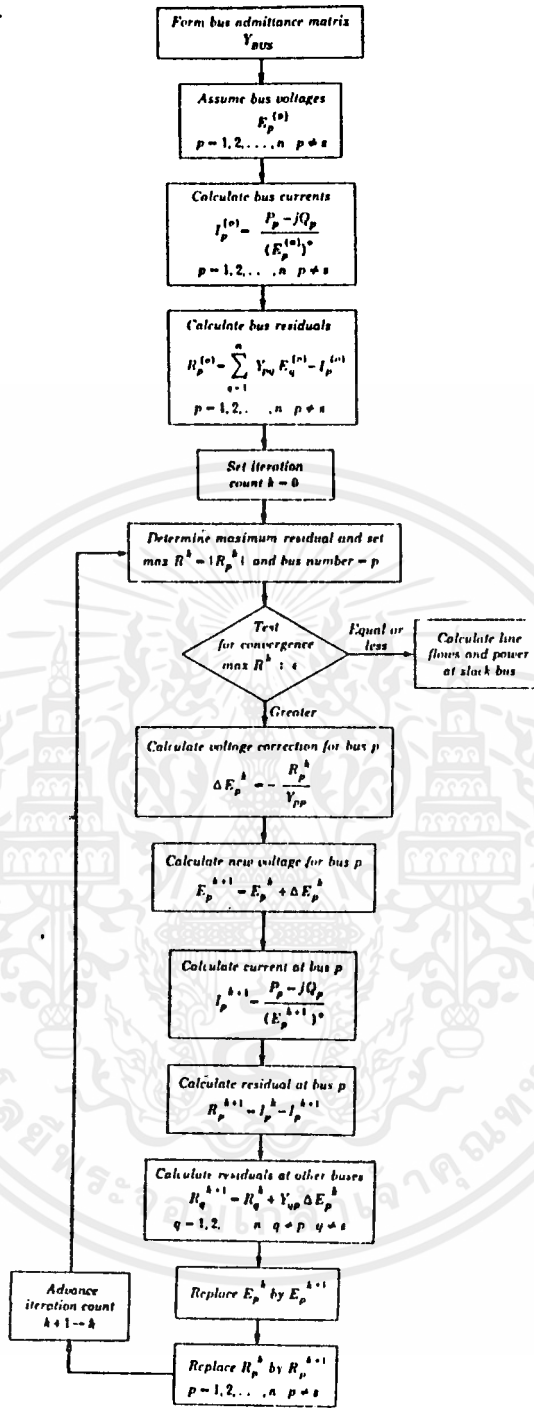
คำตอบสุดท้ายที่สมบูรณ์ จะหาได้โดยวิธีทำซ้ำเพียงหนึ่งถึงสองครั้งเท่านั้น ในการหาค่าจาโคเบียน เมื่อใช้ระบบพิกัดแกนมุมฉาก ค่าตอบของโพลดิโพลว์สามารถหาได้โดยละทิ้งค่าสมาชิก off-diagonal ของเมตริกซ์ย่อย J_1, J_2, J_3 และ J_4 ของจาโคเบียน ซึ่งจะได้สมการแสดงการเปลี่ยนแปลงของกำลังจริงและกำลังเสมือนที่บัส P ดังนี้

$$\begin{aligned} \Delta P_p &= \frac{\partial P_p}{\partial e_p} \Delta e_p + \frac{\partial P_p}{\partial f_p} \Delta f_p \\ &= \Delta e_p (e_p G_{pp} - f_p B_{pp} + C_p) + \Delta f_p (e_p B_{pp} + f_p G_{pp} + d_p) \\ \Delta Q_p &= \frac{\partial Q_p}{\partial e_p} \Delta e_p + \frac{\partial Q_p}{\partial f_p} \Delta f_p \\ &= e_p (e_p B_{pp} + f_p G_{pp} - d_p) + \Delta (-e_p G_{pp} + f_p B_{pp} + C_p) \end{aligned}$$

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

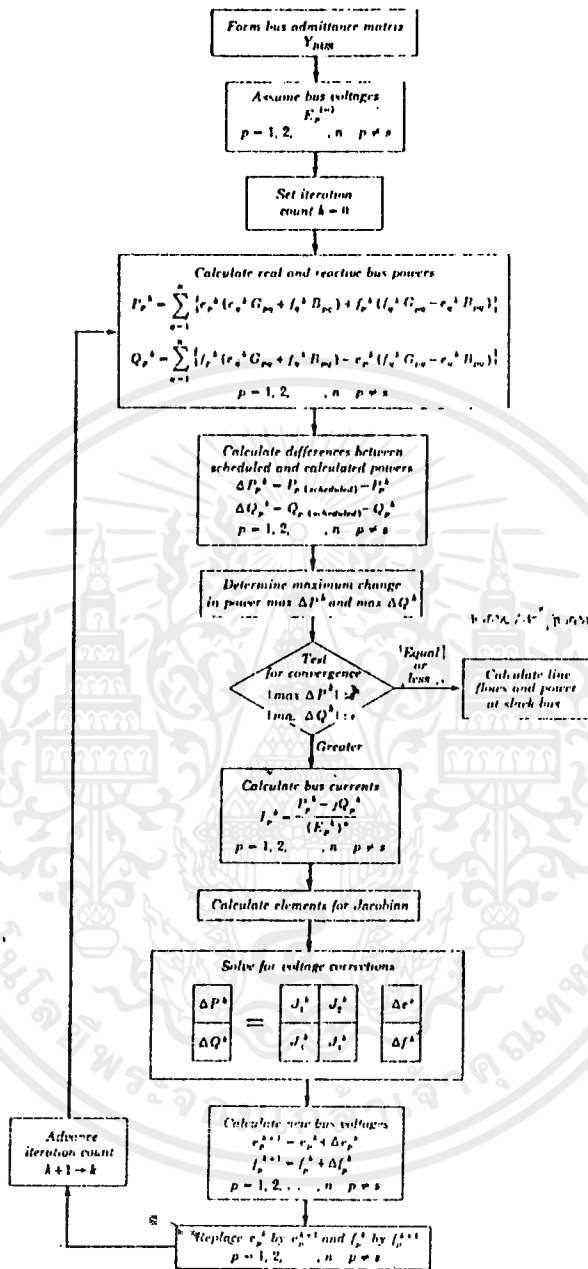
สมการเหล่านี้สามารถหาคำตอบได้โดยใช้วิธี Gauss-Seidel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.10 Load Flow solution by relaxation method using YBus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.11 Load Flow solution by Newton-Raphson method using Y_{BUS}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 Acceleration of convergence

ในบางกรณีอัตราการลู่เข้า (convergence) สำหรับกระบวนการทำซ้ำจะเร็วขึ้นโดยใช้ตัวเร่ง (acceleration factor) ในการประมาณค่าตอบจากการทำซ้ำแต่ละรอบให้ และ เป็น acceleration factors สำหรับส่วนจริงและส่วนจินตภาพของโวลเตจ ค่าที่ถูกเร่งจะเป็น

$$\begin{aligned} e_{p(\text{accelerated})}^{k+1} &= e_p^k + \alpha(e_p^{k+1} - e_p^k) \\ f_{p(\text{accelerated})}^{k+1} &= f_p^k + \alpha(f_p^{k+1} - f_p^k) \end{aligned}$$

และแทนการคำนวณของ e_p^{k+1} และ f_p^{k+1}

3.9 Voltage controlled buses

การควบคุมโวลเตจที่หัวของแหล่งจ่ายกำลังเสมือน

โดยปกติการคำนวณหาค่าตอบของโพลดิพล่าจะต้องมี voltage controlled buses ที่บัสเหล่านี้จะต้องหาค่าของกำลังจริงและกำลังเสมือน

วิธี Gauss และ Gauss-Seidel โดยที่ Y_{pq} ค่ากำลังเสมือนที่ voltage controlled bus p จะต้องคำนวณก่อนที่จะคำนวณหาโวลเตจที่บัสนั้น แยกส่วนจริงและส่วนจินตภาพในสมการกำลัง

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

ค่ากำลังเสมือนที่บัสคือ

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

(3.9.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ e_p และ f_p คือส่วนประกอบของโวลเตจที่บัส p ค่าของ e_p และ f_p จะต้องสอดคล้องกับความสัมพันธ์

$$e_p^2 + f_p^2 = V_{p(\text{scheduled})}^2 \quad (3.9.2)$$

เพื่อคำนวณค่ากำลังเสมือนที่บัส จะต้องหาค่าโวลเตจที่บัสที่กำหนด โดยประมาณ e_p^k และ f_p^k ดังนั้นตามสมการ (3.9.2)

$$\angle_p^k = \arctan \frac{f_p^k}{e_p^k}$$

สมมติว่า มุมที่ประมาณและโวลเตจที่กำหนดเท่ากัน และปรับค่าประมาณของ e_p^k และ f_p^k จะได้

$$\begin{aligned} e_{p(\text{new})}^k &= \left| V_{p(\text{scheduled})} \right| \cos \angle_p^k \\ f_{p(\text{new})}^k &= \left| V_{p(\text{scheduled})} \right| \sin \angle_p^k \end{aligned}$$

แทนค่า $e_{p(\text{new})}^k$ และ $f_{p(\text{new})}^k$ ในสมการ (3.9.1) จะหาค่าของกำลังเสมือน Q_p^k และใช้กับ $V_{p(\text{new})}^k$ สำหรับคำนวณโวลเตจค่าใหม่ V_p^{k+1}

ในทางปฏิบัติ จะมีขีดจำกัดแหล่งกำลังเสมือนที่ Voltage controlled bus ซึ่งจะต้องคำนึงถึงด้วย ถ้า Q_p^k ที่คำนวณได้เกินกว่าค่าความสามารถสูงสุด $Q_{p(\text{max})}$ ของแหล่งจ่ายค่าสูงสุดจะถูกแทนในกำลังเสมือนของบัส ถ้าค่าที่คำนวณได้น้อยกว่าค่าความสามารถต่ำสุด $Q_{p(\text{min})}$ ค่าต่ำสุดจะถูกใช้ ในแต่ละกรณีมันเป็นไปได้ที่จะค่าลบ ด้วยค่าโวลเตจที่กำหนด และดังนั้น $V_{p(\text{max})}^k$ ไม่สามารถใช้ในการคำนวณของ V_p^{k+1}

ในวิธี นิวตัน-ราฟสัน สมการสำหรับ voltage controlled bus p คือ

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

และ

$$\left| V_p \right|^2 = e_p^2 + f_p^2$$

(3.9.3)

เมื่อสมการที่ (3.9.3) แทนสมการสำหรับกำลังเสมือน สมการเมตริกซ์ ซึ่งมีความสัมพันธ์ของการเปลี่ยนแปลงกำลังของบัส และกำลังล่อง ขนาดของโวลเตจ ที่เปลี่ยนแปลงในส่วนตัวจริง และส่วนจินตภาพของโวลเตจ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{array}{|c|} \hline \Delta P \\ \hline \Delta Q \\ \hline \Delta |E|^2 \\ \hline \end{array} = \begin{array}{|c|c|} \hline J_1 & J_2 \\ \hline J_3 & J_4 \\ \hline J_5 & J_6 \\ \hline \end{array} \begin{array}{|c|} \hline \Delta e \\ \hline \Delta f \\ \hline \end{array}$$

สมาชิกของเมตริกซ์ย่อย J_1, J_2, J_3, J_4 ค่าวนเหมือนที่กล่าวมาแล้ว สมาชิก off-diagonal ของ J_5 จากสมการ (3.9.3) คือ

$$\frac{\partial |v_p|^2}{\partial e_q} = 0 \quad q \neq p$$

และสมาชิก diagonal คือ

$$\frac{\partial |v_p|^2}{\partial e_p} = 2e_p$$

ทำนองเดียวกัน สมาชิก off-diagonal ของ J_6 คือ

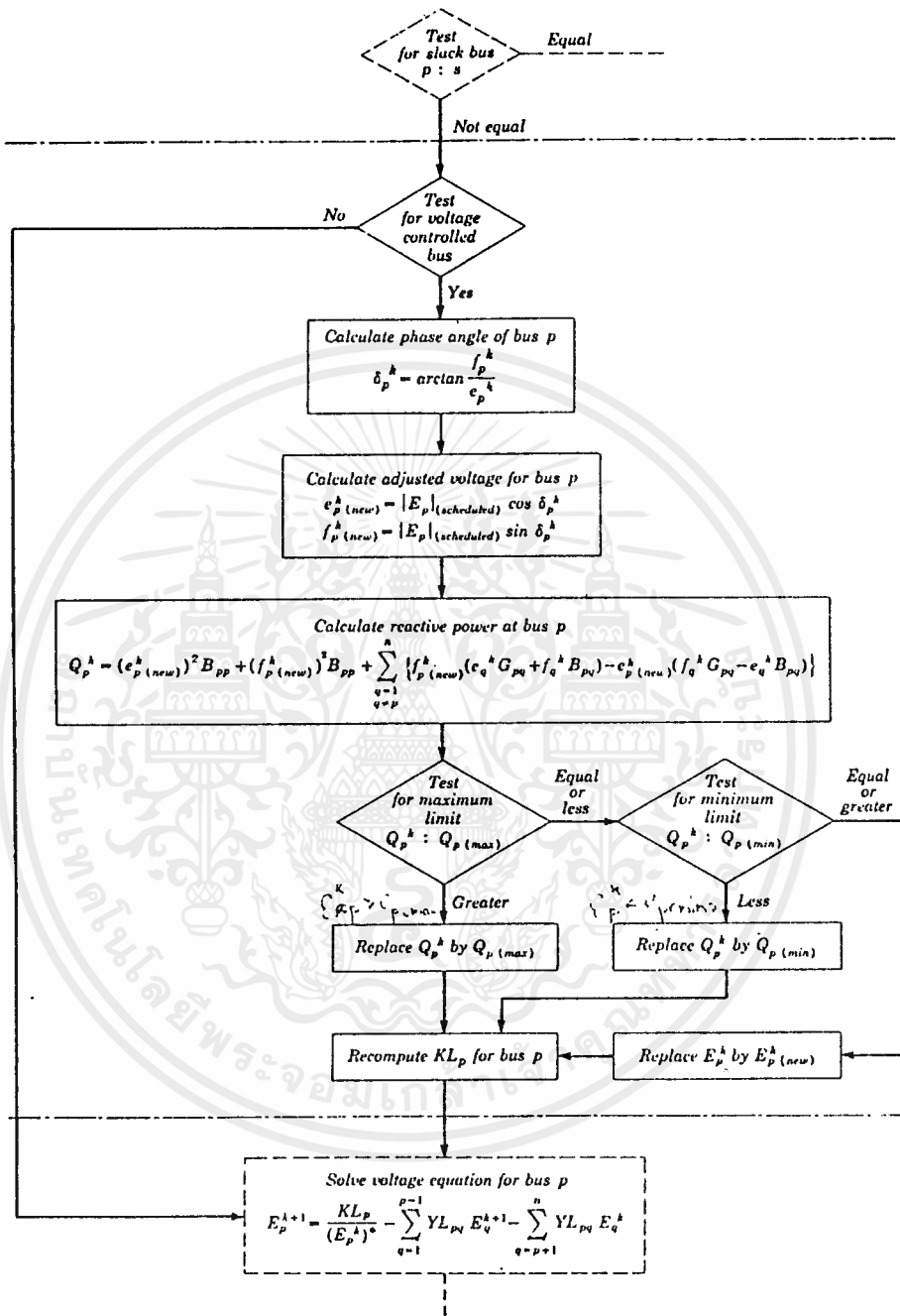
$$\frac{\partial |v_p|^2}{\partial f_q} = 0 \quad q \neq p$$

และสมาชิก diagonal คือ

$$\frac{\partial |v_p|^2}{\partial f_p} = 2f$$

ถ้ากำลังเสริมไม่สามารถคงอยู่ที่ขนาดของบัสโวลเตจที่ต้องการ ค่ากำลังเสริมจะต้องงัดที่จุดจำกัด ในกรณีนี้จะทำตัวเสริมโทลคิบส์ที่จำกัดกำลังเสริม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.12 calculation of reactive power at voltage controlled buses in the Gauss-Seidel iterative method using Y_{BUS}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 Voltage control at a remote bus

ในทางปฏิบัติ การควบคุมระบบไฟฟ้ากำลังจะควบคุมโวลเตจ ที่บัสมากกว่าที่จะควบคุมที่ซ้ำของแหล่งจ่ายกำลังเสมือนนี้ ทำให้มีความจำเป็นที่การหาค่าตอบของโวลตโพล์ ในการหาค่ากำลังเสมือนที่บัส p จะต้องคงค่าขนาดของโวลเตจที่หาในบัส q

ดังแสดงในรูป

วิธีหาค่าโวลเตจที่กำหนดสำหรับบัส p ในการประมาณครั้งที่หนึ่งคือ

$$\left| V_p \right|_{(\text{scheduled})}^{(0)} = \left| V_q \right|_{(\text{scheduled})}$$

ระหว่างการทำซ้ำของค่าตอบปรากฏที่บัส p จะถูกคำนวณตามปกติ โดยใช้โวลเตจที่กำหนดนี้ หลังจากการคำนวณหาโวลเตจที่บัส q ผลต่างระหว่างขนาดของ scheduled voltage หาได้จาก

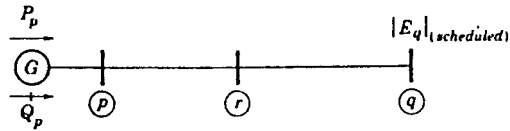
$$\Delta \left| V_q \right|^k = \left| V_q \right|_{(\text{scheduled})} - \left| V_q \right|^k$$

เมื่อ V_q^k เป็นค่าโวลเตจที่บัสที่คำนวณได้ ถ้าค่าของ $\Delta \left| V_q \right|^k$ นี้สูงกว่าค่าผิดพลาดที่ยอมรับได้ ค่าขนาดของ scheduled voltage สำหรับบัส p จะถูกประมาณใหม่เป็น

$$\left| V_p \right|_{(\text{scheduled})}^{k+1} = \left| V_p \right|_{(\text{scheduled})}^k + \Delta \left| V_q \right|^k$$

กระบวนการนี้จะใช้ใน Gauss-Seidel ที่ใช้ Y_{bus} ขณะที่การซ้ำของค่าตอบการเปลี่ยนค่า scheduled voltage ที่สมมุติที่บัส p จะไม่มีผลต่อการคำนวณโวลเตจที่บัส q อย่างทันทีทันใด จำเป็นที่จบการทำซ้ำให้สมบูรณ์ก่อนที่จะประมาณค่า scheduled voltage ใหม่สำหรับบัส p จากการทดสอบการทำซ้ำ 5 ครั้ง จะมีความแม่นยำเพียงพอในการเปลี่ยนค่าสำหรับคำนวณโวลเตจที่บัส q สำหรับการประมาณค่า scheduled voltage ค่าใหม่ที่บัส p ค่าโวลเตจที่ผิดพลาดเพียง 0.005 ต่อหน่วย เป็นค่าที่ยอมรับได้

วิธีการอื่นในการเปลี่ยน $\left| V_p \right|_{(\text{scheduled})}^{k+1}$ โดยที่ละน้อยในแต่ละทำซ้ำจนกระทั่งขนาดของ $\Delta \left| E_q \right|^k$ น้อยกว่าค่าผิดพลาด



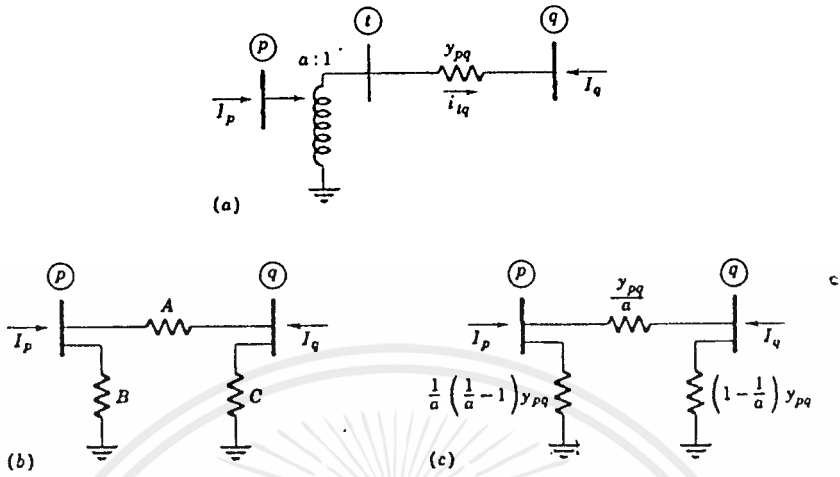
รูป 3.13 single line diagram of reactive power source and remote voltage controlled bus

3.11 วงจรสมมูลของหม้อแปลง

Fixed tap Setting transformers

หม้อแปลงซึ่งไม่มีการเปลี่ยนแปลงจำนวนรอบสามารถแทนได้ด้วยอิมพีแดนซ์ หรือแอดมิต-แดนซ์ ต่ออนุกรมกับหม้อแปลงแบบออโตในอุดมคติ ดังแสดงในรูป (a) วงจรสมมูลแบบ ที่ทำได้นี้จะใช้ในการศึกษาไหลตไฟฟ้า

พารามิเตอร์ของวงจรสมมูลแบบ π แสดงไว้ในรูป (b) ซึ่งสามารถหาได้จากการกำหนดให้กระแสที่ไหลในหม้อแปลงกับที่ไหลในวงจรสมมูลแบบ มีค่าเท่ากัน ที่บัส p กระแสที่เข้า I_p ของหม้อแปลง แสดงไว้ในรูป a คือ



รูป 3.14 Transformer representation

- a) equivalent circuit;
- b) equivalent π circuit;
- c) equivalent π circuit with parameter expressed in terms of admittance and off-nominal turns

$$I_p = \frac{i_{tq}}{a}$$

เมื่อ a เป็นอัตราส่วนจำนวนรอบของหม้อแปลงออโต ในอุดมคติ และ i_{tq} คือกระแสที่ไหลจาก t ไป q คือ

$$i_{tq} = (V_t - V_q) y_{pq}$$

ดังนั้น

$$I_p = (V_t - V_q) \frac{y_{pq}}{a}$$

(3.11.1)

เพราะว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_s = \frac{V_D}{a}$$

สมการที่ (3.11.1) จะกลายเป็น

$$I_D = (V_D - aV_s) \frac{y_{DQ}}{a^2} \quad (3.11.2)$$

ทำนองเดียวกัน สำหรับกระแสที่เข้า I_s ที่ขั้ว q คือ

$$I_s = (V_s - V_D) y_{DQ} \quad (3.11.3)$$

แทนค่า V_s สมการ (3.11.3) จะกลายเป็น

$$I_s = (aV_s - V_D) \frac{y_{DQ}}{a} \quad (3.11.4)$$

กระแสที่เข้าทั้งหมดของวงจรสมมูลแบบ π แสดงในรูป b คือ

$$I_D = (V_D - V_s) A + V_s B \quad (3.11.5)$$

$$I_s = (V_s - V_D) A + V_D C \quad (3.11.6)$$

ให้ $V_D = 0$ และ $V_s = 1$ ในสมการที่ (3.11.2)

$$I_D = \frac{-y_{DQ}}{a}$$

ให้ $V_D = 0$ และ $V_s = 1$ ในสมการที่ (3.11.5)

$$I_D = -A$$

เพราะว่ากระแสที่เข้าของหม้อแปลงและของวงจรสมมูลแบบ π ต้องเท่ากัน

$$A = \frac{y_{DQ}}{a} \quad (3.11.7)$$

ทำนองเดียวกัน แทนค่า $V_D = 0$ และ $V_s = 1$ ในทั้งสมการ (3.11.4) และ (3.11.6)

$$I_D = y_{DQ} \quad \text{และ} \quad I_s = A + C$$

อีกครั้ง เพราะว่ากระแสที่เข้าของหม้อแปลงและของวงจรสมมูลต้องเท่ากัน

$$y_{pq} = A + C$$

แทนค่าสำหรับ A จากสมการ (3.11.7) และแก้สมการหาค่า C

$$\begin{aligned} C &= y_{pq} - \frac{y_{pq}}{a} \\ &= (1 - \frac{1}{a}) y_{pq} \end{aligned}$$

กระแสในสมการ (3.11.2) และ (3.11.5) เท่ากัน และแทนค่า A จาก (3.11.7)

$$(V_p - aV_q) \frac{y_{pq}}{a^2} = (V_p - V_q) \frac{y_{pq}}{a} + V_p B$$

แก้สมการหาค่า B

$$\begin{aligned} B &= \frac{(V_p - aV_q) \frac{y_{pq}}{a^2} - (V_p - V_q) \frac{y_{pq}}{a}}{V_p} \\ &= \frac{y_{pq}}{a^2} - \frac{y_{pq}}{a} \\ &= \frac{1}{a} (1 - 1) y_{pq} \end{aligned}$$

วงจรสมมูลแบบ π และพารามิเตอร์ที่แสดงในเทอมของอัตราส่วนจำนวนรอบ a และค่าแอดมิตแตนซ์ของหม้อแปลง ได้แสดงไว้ในรูป C

เมื่ออัตราส่วนจำนวนรอบถูกแทนในบัส p ของหม้อแปลงที่ระหว่างบัส p และ q ค่า self admittance ที่บัส p คือ

$$\begin{aligned} Y_{pp} &= y_{p1} \dots + \frac{y_{pq} \dots}{a} + \frac{y_{pn}}{a} + \frac{1}{a} (1 - 1) y_{pq} \\ &= y_{p1} + y_{p2} \dots + \frac{y_{pq} \dots}{a^2} \dots + y_{pn} \end{aligned}$$

ค่า mutual admittance จาก p ไป q คือ

$$Y_{pq} = \frac{-Y_{qp}}{a}$$

ค่า self admittance ที่บัส q คือ

$$\begin{aligned} Y_{qq} &= y_{q1} \dots + \frac{y_{qp}}{a} \dots + Y_{qn} + (1 - \frac{1}{a}) y_{qp} \\ &= y_{q1} \dots + y_{qp} \dots + y_{qn} \end{aligned}$$

และไม่มีการเปลี่ยนแปลง ค่า mutual admittance จาก q ไป p คือ

$$Y_{qp} = \frac{-y_{qp}}{a}$$

วงจรสมมูลแบบ π แสดงในรูป C ถ้าไม่คิดรวมผลของ shunt ที่คอลลกราวด์ กระแสรวมที่บัส p และ q แสดงได้ดังนี้

$$\begin{aligned} I_p &= \frac{P_p - jQ_p}{V_p^*} - y_p V_p - \frac{1}{a} (1 - \frac{1}{a}) y_{pq} V_p \\ I_q &= \frac{P_q - jQ_q}{V_q^*} - y_q V_q - (1 - \frac{1}{a}) y_{qp} V_q \end{aligned}$$

การเปลี่ยนแทปในสภาวะโหลดของหม้อแปลง

ในการแทน tap changing under load (TCUL) transformer มันมีความจำเป็นที่ต้องเปลี่ยนอัตราส่วนจำนวนรอบเพื่อให้ได้ขนาดของโวลเตจที่บัสตามต้องการ ซึ่งทำได้โดยเปลี่ยนอัตราส่วนจำนวนรอบไปค่าเล็กๆ a ในการทำซ้ำเมื่อขนาดโวลเตจที่บัส q เป็นดังนี้

$$\left| \frac{|V_q^k|}{|V_q^i|} - \frac{|V_q^i|}{|V_q^i|} \right| > E$$

มาตรฐานของ tap setting ของหม้อแปลง TCUL คือ $\pm 5/8$ เปอร์เซ็นต์ต่อสแต็ป ซึ่งไม่จำเป็นที่จะต้องตรวจสอบขนาดของโวลเตจของบัสที่ถูกควบคุมด้วยหม้อแปลง TCUL ทุก ๆ รอบของการทำซ้ำ ตรวจสอบบางครั้งก็เพียงพอแล้ว ขนาดโวลเตจที่ยอมรับให้ผิดพลาดได้ E คือ 0.01 ต่อหน่วย จะเป็นค่าที่ยอมรับได้

ค่า self admittance Y_{pp} และ mutual admittances $Y_{pq} = Y_{qp}$ จะ

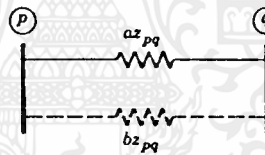
ต้องคำนวณใหม่ทุก ๆ รอบของการเปลี่ยน tap setting ของหม้อแปลงที่ต่อระหว่างบัส p และ q

ค่า Z_{pq} จะต้องถูกตัดแปลงทุกครั้งที่เปลี่ยน tap setting ซึ่งการเปลี่ยนแปลงนี้มีผลโดยเหมือนมีสมาชิกใหม่จากบัส p ไป q คือ อิมพีแดนซ์อนุกรมของวงจร π เป็นดังนี้

$$(a + \Delta a) Z_{pq}$$

เมื่อ a เป็นอัตราส่วนจำนวนรอบเดิมและ $a + \Delta a$ คือ ค่าที่เปลี่ยนไป ให้ bz_{pq} เป็นอิมพีแดนซ์ใหม่ที่เพิ่มเข้าไป ดังแสดงในรูป 3.15 และ

$$(a + \Delta a) Z_{pq} = \frac{abz_{pq}^2}{az_{pq} + bz_{pq}}$$



รูป 3.15 แสดงส่วนที่เพิ่มเข้าไปใน transformer tap setting

แก้สมการหาค่า bz_{pq}

$$bz_{pq} = - \frac{a(a + \Delta a) Z_{pq}}{\Delta a}$$

การเปลี่ยน tap setting ของหม้อแปลง ทำให้สมาชิกทุกตัวของ Z_{pq} ต้องคำนวณใหม่ เพื่อหลีกเลี่ยงข้อยุ่งยากนี้ จึงกำหนดให้อิมพีแดนซ์อนุกรม เท่ากับอิมพีแดนซ์เดิมของหม้อแปลง และส่วนของขั้วที่เปลี่ยนแปลงการเปลี่ยน tap

ให้ $A = y_{pq}$ และให้กระแสที่ขั้วเท่ากันจากสมการ (3.11.2) และ (3.11.5)

สำหรับหม้อ

แปลงและวงจรเสมือนแล้ว จะได้

$$(V_p - V_q) y_{pq} + V_p B = (V_p - aV_q) \frac{y_{pq}}{a^2}$$

แก้สมการหาค่า B

$$\begin{aligned} B &= \left\{ (V_p - aV_q) \frac{y_{pq}}{a^2} - (V_p - V_q) y_{pq} \right\} \frac{1}{V_p} \\ &= \left\{ \left(\frac{1}{a^2} - 1 \right) - \left(\frac{1}{a} - 1 \right) \frac{V_q}{V_p} \right\} y_{pq} \\ &= \frac{(1 - 1)}{a} \left\{ \frac{(1 + 1)}{a} - \frac{V_q}{V_p} \right\} y_{pq} \end{aligned} \quad (3.11.8)$$

ทำนองเดียวกัน กระแสที่เข้า I_p เท่ากันจากสมการ (3.11.4) และ (3.11.6) เมื่อ $A = y_{pq}$

$$(V_q - V_p) y_{pq} + V_q C = (aV_q - V_p) \frac{y_{pq}}{a}$$

แก้สมการหาค่า C

$$\begin{aligned} C &= \left\{ (aV_q - V_p) \frac{y_{pq}}{a} - (V_q - V_p) y_{pq} \right\} \frac{1}{V_q} \\ &= \frac{(1 - 1)}{a} y_{pq} \frac{V_p}{V_q} \end{aligned}$$

ค่า shunt admittance สมการ (3.11.8) และ (3.11.9) ที่มี p และ q ตามลำดับเป็นฟังก์ชันของโวลเตจ V_p และ V_q ดังนั้นจะได้ bus loading equations ดังนี้

$$I_p = \frac{P_p - jQ_p}{V_p^*} - y_{pq} V_p - \frac{(1 - 1)}{a} \left\{ \frac{(1 + 1)}{a} - \frac{V_q}{V_p} \right\} y_{pq} V_p$$

$$I_q = \frac{P_q - jQ_q}{V_q^*} - y_{pq} V_q - \frac{(1 - 1)}{a} y_{pq} V_p$$

Phase shifting transformer

phase shifting transformer จะถูกแทนในการศึกษาโหลดโฟลด์ด้วยอิมพีแดนซ์ หรือ แอดมิตแดนซ์ต่ออนุกรมกับหม้อแปลงออโตคัมคิต ซึ่งมีค่าอัตราส่วนของรอบ เป็นจำนวนเชิงซ้อน ดังแสดงในรูป และโวลเตจที่หัว V_p และ V_s มีความสัมพันธ์โดย

$$\frac{V_p}{V_s} = a_s + jb_s$$

(3.11.10)

เพราะว่าไม่มีการสูญเสียกำลังในหม้อแปลงออโตคัมคิต

$$V_p^* i_{pr} = V_s^* i_{sr}$$

(3.11.11)

จากสมการ (3.11.10) และ (3.11.11) จะได้

$$\begin{aligned} \frac{i_{pr}}{i_{sr}} &= \frac{V_s^*}{V_p^*} \\ &= \frac{1}{a_s - jb_s} \end{aligned}$$

เพราะว่า

$$i_{sr} = (V_p - V_s) y_{ps}$$

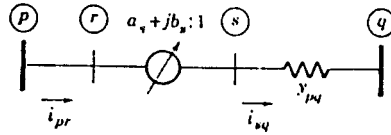
แล้ว

$$i_{pr} = (V_p - V_s) \frac{y_{ps}}{a_s - jb_s}$$

แทนค่าใน V_s จากสมการ (3.11.10)

$$i_{pr} = \{ V_p - (a_s + jb_s) V_p \} \frac{y_{ps}}{a_s^2 + b_s^2}$$

(3.11.12)



รูป3.16 phase shifting transformer representation

ทำนองเดียวกัน กระแสของหม้อแปลงที่บัส q, i_{qs} คือ

$$i_{qs} = (V_q - V_s) y_{pq}$$

แทนค่าของ V_s จะได้

$$i_{qs} = \left\{ (a_s + j b_s) V_q - V_p \right\} \frac{y_{pq}}{a_s + j b_s}$$

เมื่อ phase shifting transformer ถูกต่อระหว่างบัส p และ q ค่า self admittance ที่บัส p สามารถได้โดยให้ V_p เท่ากับ 1 per unit และลัดวงจรบัสเครื่องอื่น ๆ จะได้

$$Y_{pp} = i_{p1} + i_{p2} \dots + i_{pr} \dots + i_{pn}$$

แทนค่าสำหรับ i_{pr} จากสมการ (3.11.12) และเพราะว่า

$$i_{p1} = y_{p1}$$

$$i_{p2} = y_{p2}$$

.....

$$i_{pn} = y_{pn}$$

แล้ว

$$Y_{pp} = y_{p1} + y_{p2} \dots + \frac{y_{pq}^2}{a_s^2 + b_s^2} \dots + y_{pn}$$

กระแสที่ไหลออกจากบัส p เป็น $-i_{sp}$ ดังนั้น mutual admittance คือ

$$Y_{pq} = -i_{sp}$$

3.12 Tie line control

ในการศึกษาที่เกี่ยวกับการเชื่อมต่อของระบบไฟฟ้ากำลัง การหาค่าตอบของโพลดิพล์ จะต้องหาค่ากำลังสุทธิสำหรับแต่ละระบบ ขึ้นแรกของการแก้ปัญหาจะต้องคำนวณค่าโวลเตจของทั้งระบบก่อน ด้วยสมมติค่าการจ่ายกำลังในแต่ละระบบ ถัดมาใช้โวลเตจที่ได้ คำนวณหาผลรวมของระบบของกำลังสุทธิที่มีการเปลี่ยนแปลงภายในระบบ และเปรียบเทียบค่าที่ได้จริง กับค่าที่ตั้งไว้ของแต่ละระบบ เพื่อปรับค่า

ในทางปฏิบัติ การเปลี่ยนแปลงในระบบ จะถูกเลือกโดยเครื่องกำเนิดไฟฟ้าตัวหนึ่งในแต่ละระบบ ให้เป็น regulating generator ในแต่ละ regulating generator จะปรับกำลังสุทธิภายในให้ได้ใกล้ค่าที่กำหนดไว้ ดังนั้นสำหรับระบบ A ดังในรูป ค่ากำลังสุทธิจริง ๆ ในการเปลี่ยนภายในคือ

$$P_T^k = P_{F1}^k + P_{F2}^k - P_{F3}^k + P_{F4}^k$$

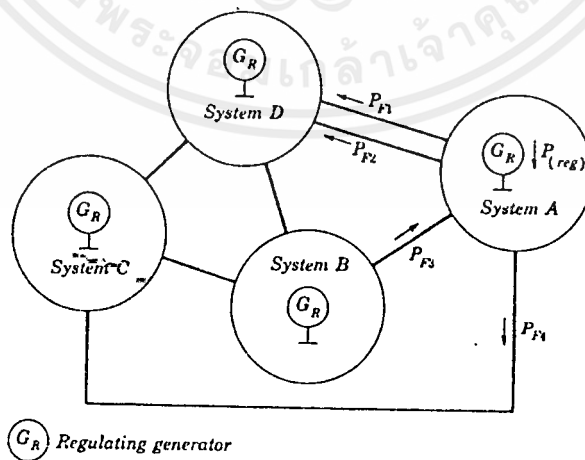
ผลต่างระหว่างค่าจริงและค่ากำหนด คือ

$$\Delta P_T^k = P_{T(\text{scheduled})} - P_T^k$$

ค่ากำลังที่จ่ายออกใหม่สำหรับ regulating generator ในระบบ A คือ

$$P_{(reg)}^{k+1} = P_{(reg)}^{k+1} + \Delta P_T^k$$

ใช้วิธีทำนองเดียวกันนี้คำนวณในระบบอื่น ๆ และทำซ้ำของการออกค่าตอบของโวลเตจด้วย กระบวนการนี้จะทำซ้ำจนกว่า ΔP_T^k ทั้งหมดน้อยกว่าหรือเท่ากับค่าผิดพลาดที่ยอมรับได้ โดยปกติ ประมาณ 5 เมกกะวัตต์



รูป 3.17 Simplified representation of interconnected power system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.13 Comparison of Load Flow Method

ในหัวข้อนี้จะเป็นการเปรียบเทียบระหว่างการใช้วิธี Gauss-Seidel และ Newton-Raphson ในการแก้ปัญหาโหลดฟลว์

1. Computer Storage, Ease of Programming

วิธี Gauss-Seidel จะง่ายไม่มีความซับซ้อนในตัวโปรแกรม และใช้หน่วยความจำของคอมพิวเตอร์ และโดยปกติจะเปลี่ยนแปลงไปตามขนาดของปัญหาที่พิจารณา

สำหรับวิธี Newton-Raphson การจัดเตรียมข้อมูลและการจัด square matrix storage จะต้องทำไปพร้อม ๆ กัน เพื่อเป็นลำดับในการลดเวลาในการคำนวณโดยใช้วิธี Gaussian elimination ในการแก้สมการ และในโปรแกรมของ Newton-Raphson จะมีรายละเอียด และความซับซ้อนมากกว่าในการคำนวณด้วย Gauss-Seidel โดยวิธีนี้ จะต้องพยายามลดหน่วยความจำที่ว่าง เพื่อให้มันแปรตามขนาดของปัญหา แต่ปัจจุบันปัญหาเกี่ยวกับที่ว่างของหน่วยความจำ จะมีผลน้อยมากเนื่องจากคอมพิวเตอร์สมัยใหม่

2. Solution Time

Gauss-Seidel จะใช้การ iteration ประมาณ 50 ถึง 500 ครั้งในการแก้ปัญหา และต้องใช้ accelerating factor เข้าช่วย จำนวน iteration ในการหาค่าตอบ จะเพิ่มมากขึ้นตามขนาดของปัญหา และถ้าความสัมพันธ์ของ components ใน networks มีจำนวนมาก จะทำให้รูปแบบทางคณิตศาสตร์ของ Gauss-Seidel program จะยากขึ้นด้วย เป็นการเพิ่มเวลาที่ใช้ในการหาค่าตอบ

Newton-Raphson programs จะใช้การ iteration เพียง 4 หรือ 5 ครั้งเท่านั้น (แต่ว่าการ iteration แต่ละครั้งจะใช้เวลานานกว่า) ซึ่งจะไม่ขึ้นกับขนาดของปัญหาหรือความซับซ้อนของมันเลย

Newton-Raphson จะใช้เวลาเร็วกว่า accelerated Gauss-Seidel สำหรับปัญหาเล็ก ๆ

3. Reliability

Reliability คือ ความสามารถของ algorithm ในการหาค่าตอบ

Gauss-Seidel programs อาจจะถูกหยุดจากการ converge ถ้าเป็นระบบจำนวนมาก และสามารถเลือก accelerating factor สำหรับ Gauss-Seidel ได้ก็ตามใจชอบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานาน นี ไม่นิยามให้เกินไปเชิงประเด็นด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละกรณี

Newton-Raphson program จะ converge ได้ง่ายกว่าได้ในการหาคำตอบ

3.14 สรุปและเปรียบเทียบ

ขั้นแรกในวิธีการของโพลด์โพลด์ คือ แปลงเน็ตเวิร์กทั้งหมดให้อยู่ในรูป เน็ตเวิร์คเมตริก โดยกำหนดบัสหนึ่งเป็นจุดอ้างอิง และแปลงบัสอื่น ๆ ที่ต่อส่วนประกอบต่าง ๆ เข้าหากัน ซึ่งจะต้องได้ออกมาเป็นข้อมูลที่คู่ด้วย การจัดบัสแอดมิตแตนซ์ จะไม่คำนึงถึงผลของการ คัปปลิงกัน ดังนั้น diagonal element Y_{pp} ของบัสเมตริกนี้ จะเท่ากับผลรวมของแอดมิตแตนซ์ ขององค์ประกอบที่ต่ออยู่กับบัส p และ off-diagonal element Y_{pq} จะเท่ากับค่าลบของแอดมิตแตนซ์ ขององค์ประกอบ (element) ที่ต่อระหว่างบัส p และ q อย่างไรก็ตาม เนื่องจาก บัส-แอดมิตแตนซ์เมตริกจะอยู่อย่างกระจัดกระจาย ดังนั้น จะมีสมาชิกของเมตริกซ์จำนวนมากที่เป็นศูนย์ ทำให้มีจำนวนที่คำนวณเหลือน้อยลง

เราสามารถเก็บค่าบัสแอดมิตแตนซ์เมตริก โดยตัดส่วนที่เป็นศูนย์ออก ทำให้ประหยัดเนื้อที่ในหน่วยความจำของคอมพิวเตอร์ วิธีหนึ่งคือเก็บส่วนที่ไม่เป็นศูนย์ด้วยหลายเลขบัส ซึ่งจะสมาชิกที่เป็นทั้งแถวและคอลัมน์ของเมตริก

เวลาที่คอมพิวเตอร์ใช้ทำอิทเทอเรชันในการหาคำตอบ จะขึ้นอยู่กับ

1. ความซับซ้อนทางคณิตศาสตร์ของสมการที่ใช้ในการอิทเทอเรชัน
2. อัตราการคอนเวจซ์ของเทคนิคที่ใช้
3. ขนาดและลักษณะของระบบไฟฟ้ากำลัง

ถ้าเปรียบเทียบกันระหว่างวิธี Gauss-Seidel และ Newton-Raphson ประโยชน์จากการกระจัดกระจายของเน็ตเวิร์คเมตริก จะช่วยความซับซ้อนทางคณิตศาสตร์ในการอิทเทอเรชันได้ โดย Gauss-Seidel จะใช้พิกัดแกนมุมจาก ส่วน Newton-Raphson จะใช้พิกัดเชิงขั้ว อย่างไรก็ตามจากที่กล่าวถึงความซับซ้อนทางคณิตศาสตร์ที่ลดลงไปของทั้ง 2 วิธี อันเนื่องจากเทคนิคในการหาคำตอบ วิธี Gauss-Seidel จะใช้เวลาน้อยกว่าของ Newton Raphson เนื่องจากวิธี Newton Raphson ต้องใช้เวลาคำนวณหาสมาชิกของจาโคเบียนในแต่ละครั้งของการอิทเทอเรชัน และเวลาที่ใช้ต่อการอิทเทอเรชันจะเพิ่มขึ้นตามจำนวนบัสของเน็ตเวิร์ค เพราะว่ามีจำนวนของสมาชิกที่ไม่เป็นศูนย์ จะเพิ่มเข้าไปในเน็ตเวิร์คเมตริก สำหรับบัสใหม่แต่ละบัสเท่า ๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับอัตราการคอนเวอร์จของวิธี Gauss-Seidel จะช้า เนื่องจากต้องการจำนวนครั้งของการอิทเทอเรชันมากกว่าวิธี Newton-Raphson ดังนั้นจำนวนครั้งของการอิทเทอเรชัน ในวิธี Gauss-Seidel จะเพิ่มขึ้นตามจำนวนบัสของเน็ตเวิร์ค ขณะที่จำนวนครั้งของการอิทเทอเรชัน ด้วยวิธีอื่นจะคงที่ โดยไม่ขึ้นกับขนาดของระบบ การเพิ่มอัตราการคอนเวอร์จของวิธี Gauss-Seidel จะเพิ่มด้วยการใช้ accelerating factors โดยจะขึ้นอยู่กับลักษณะของระบบและวิธีที่ใช้ในการคำนวณ

ค่า tolerance ที่ยอมรับได้จะขึ้นอยู่กับวิธีที่ใช้ สำหรับวิธี Gauss-Seidel ซึ่งมีอัตราการคอนเวอร์จช้า โดยมากจะใช้ค่าที่น้อย เช่น 0.0001 per unit สำหรับทั้งส่วนจริงและส่วนจินตภาพของโวลเตจ สำหรับวิธี Newton-Raphson จะใช้ tolerance ประมาณ 0.001 per unit

วิธี Newton-Raphson จะใช้เวลาในการอิทเทอเรชันน้อยที่สุด โดยในคอนเริ่มต้นจะยังไม่คิดที่ Voltage controlled buses เมื่อเริ่มคิดที่บัสนี้ เวลาที่ใช้ในการอิทเทอเรชันจะเพิ่มขึ้นตามเนื่องจากการที่มีส่วนที่ต้องคำนวณเพิ่มมากขึ้น ในขณะที่วิธี Newton-Raphson จะใช้เวลาลดลงเนื่องจากความซับซ้อนทางคณิตศาสตร์ลดลงสำหรับบัสนี้

ในการคำนวณของโพลคโพล์จะต้องตรวจสอบข้อมูลอยู่ตลอดเวลา เนื่องจากอาจมีการเปลี่ยนแปลงเกิดขึ้น เช่น การเพิ่มขึ้นหรือลดลงของสายส่ง และหม้อแปลง ซึ่งจำเป็นจะต้องเปลี่ยนแปลงเน็ตเวิร์คเมตริกตามไปด้วย ดังนั้น จึงมีความจำเป็นที่ต้องคำนวณสมาชิกของเมตริก ใหญ่ เพราะว่า การเปลี่ยนสมาชิกของเมตริก จะเกี่ยวข้องกับบัสหนึ่งบัส เน็ตเวิร์คจะได้รับผลจากการเปลี่ยนแปลงไปด้วย

การเลือกค่าเริ่มต้นสำหรับบัสโวลเตจ จะมีผลต่อเวลาในการหาค่าตอบด้วย เนื่องจากโดยปกติจะใช้ค่าที่ได้สุดท้ายของการคำนวณของแต่ละครั้ง เป็นค่าโวลเตจเริ่มต้นใหม่ในรอบถัดไป ซึ่งจะท่วลลดจำนวนการอิทเทอเรชันด้วย สำหรับเวลาที่ใช้ในคอมพิวเตอร์ จะขึ้นอยู่กับความเร็วของเครื่องและประสิทธิภาพของโปรแกรมด้วย

บทที่ 4

4.1 วิเคราะห์ระบบ 3 เฟส (Three phase system analysis)

การวิเคราะห์แกนอ้างอิง (Frame of reference analysis)

โดยปกติแล้วเรามักจะใช้ Symmetrical component transformation ซึ่งเป็น การวิเคราะห์ ทางคณิตศาสตร์ซึ่งนำเสนอโดย Fortescue ซึ่งกล่าวไว้ว่า "ระบบที่มี n เวกเตอร์ หรือปริมาณใดๆ ที่สามารถที่จะแยกเป็นอิสระจากกันได้ เมื่อระบบ n เวกเตอร์แยกเป็นระบบที่ สมมาตร n เฟส" ดังนั้นทั้งระบบ 3 เฟสของกระแส หรือโวลเตจ สามารถแปลงให้อยู่ในระบบที่ สมมาตรกันของแต่ละเวกเตอร์ได้ ซึ่งในการพิจารณาถึงข้อสมมุติฐานและวิธีการนี้ จะต้องระมัดระวัง โดยการตรวจสอบให้ดีเสียก่อน

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

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

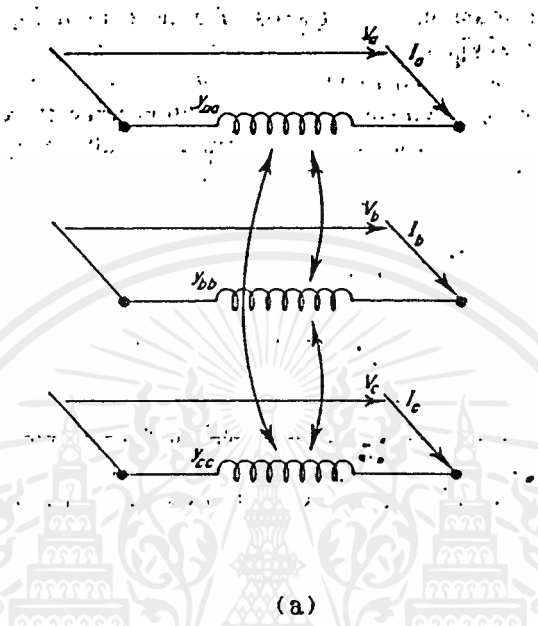
$$[V]_{abc} = [V_a \ V_b \ V_c]^T$$

และ

$$[Y]_{abc} = \begin{bmatrix} Y_{aa} & Y_{ab} & Y_{ac} \\ Y_{ba} & Y_{bb} & Y_{bc} \\ Y_{ca} & Y_{cb} & Y_{cc} \end{bmatrix} \quad (4.1.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



I_a	=	y_{aa}	y_{ab}	y_{ac}	*	V_a
I_b		y_{ba}	y_{bb}	y_{bc}		V_b
I_c		y_{ca}	y_{cb}	y_{cc}		V_c

(b)

รูป 4.1 แสดงการแทน admittance ของระบบ 3 เฟสในแต่ละ element
 (a) Series admittance element
 (b) Admittance matrix representation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริมาณที่ถูกแปลงแล้ว (โดยใช้ตัวทำย 012 เพื่อแสดงถึง zero, positive, negative sequence ตามลำดับ) ซึ่งมีความสัมพันธ์กับปริมาณที่เป็นเฟสดังนี้

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

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

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

$[T_{\underline{a}}]$ คือ transformation matrix

โวลเตจและกระแสที่ถูกแปลงแล้วจะมีความสัมพันธ์กับ admittance matrix ที่ถูกแปลงแล้วคือ

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

สมมติว่าแต่ละ element มีความสมดุล เราจะได้

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

และถ้าส่วนของ Invariant matrixes $[T_{\underline{a}}]$ เป็นจริง การแปลงใน (4.1.6) จะเป็น Diagonal matrix $[Y]_{012}$

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

แต่ถ้าในกรณีที่ admittance matrix $[y_{abc}]$ เป็นระบบไม่สมดุล เราไม่สามารถที่จะวิเคราะห์ด้วย แกนอ้างอิงของ Symmetrical component. ซึ่งแกนอ้างอิงของ Symmetrical component จะใช้ได้เพียงเมื่อ network element อยู่ในภาวะสมดุล เช่นในเครื่องกำเนิดไฟฟ้าแบบขั้วโรตอร์

อย่างไรก็ตาม ข้อสมมุติฐานบางอย่างก็อาจไม่เป็นจริงเสมอไป unsymmetrical inter-phase coupling ก็อาจจะมีในสายส่งและในเซตที่ใกล้กับหม้อแปลง และผลของการ coupling ระหว่าง sequence network ยิ่งกว่านั้นยังมีเฟสชิป phase shift ซึ่งเกิดจากการต่อหม้อแปลงแบบต่างๆ ซึ่งยากต่อการแทนใน sequence component model อีกด้วย

ประโยชน์ของการใช้ Phase coordinates ใน symmetrical component

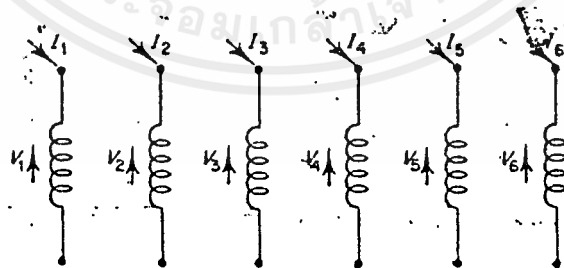
1. element ในระบบยังคงเหมือนเดิม
2. ลักษณะต่างๆ ของ asymmetric impedance, mutual coupling และระหว่าง element อื่นๆ และการสลับสายสามารถทำได้ง่ายขึ้น
3. ไม่มีปัญหาในเรื่อง phase shift ของหม้อแปลง

การวิเคราะห์ Compound admittance

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

กฎเกณฑ์และสมการของ network ทั่วๆ ไปจะยังใช้ได้ ใน compound network โดยอนุปริมาณเฟสเดียวด้วยเมทริกซ์ที่เหมาะสม

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



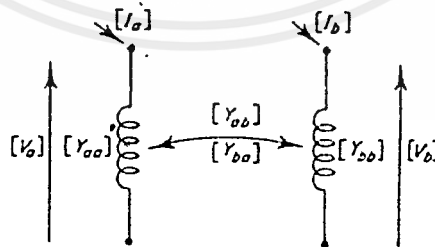
รูป 4.2 Primitive network of six couple admittances

$$[Y_{aa}] = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix} \quad [Y_{bb}] = \begin{bmatrix} y_{44} & y_{45} & y_{46} \\ y_{54} & y_{55} & y_{56} \\ y_{64} & y_{65} & y_{66} \end{bmatrix}$$

(4.1.10)

$$[Y_{ab}] = \begin{bmatrix} y_{14} & y_{15} & y_{16} \\ y_{24} & y_{25} & y_{26} \\ y_{34} & y_{35} & y_{36} \end{bmatrix} \quad [Y_{ba}] = \begin{bmatrix} y_{41} & y_{42} & y_{43} \\ y_{51} & y_{52} & y_{53} \\ y_{61} & y_{62} & y_{63} \end{bmatrix}$$

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



รูป 4.3 Two coupled compound admittances

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ระหว่าง primitive admittance matrix กับกระแสที่ไหลในแต่ละโหนด และโวลเตจในแต่ละ branch แสดงได้ดังนี้

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} & y_{16} \\ y_{21} & y_{22} & y_{23} & y_{24} & y_{25} & y_{26} \\ y_{31} & y_{32} & y_{33} & y_{34} & y_{35} & y_{36} \\ y_{41} & y_{42} & y_{43} & y_{44} & y_{45} & y_{46} \\ y_{51} & y_{52} & y_{53} & y_{54} & y_{55} & y_{56} \\ y_{61} & y_{62} & y_{63} & y_{64} & y_{65} & y_{66} \end{bmatrix} * \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix} \quad (4.1.8)$$

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

$$\begin{bmatrix} [I_a] \\ [I_b] \end{bmatrix} = \begin{bmatrix} [Y_{aa}] & [Y_{ab}] \\ [Y_{ba}] & [Y_{bb}] \end{bmatrix} * \begin{bmatrix} [V_a] \\ [V_b] \end{bmatrix} \quad (4.1.9)$$

เมื่อ

$$\begin{aligned} [I_a] &= [I_1 I_2 I_3]^T \\ [I_b] &= [I_4 I_5 I_6]^T \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 กัน

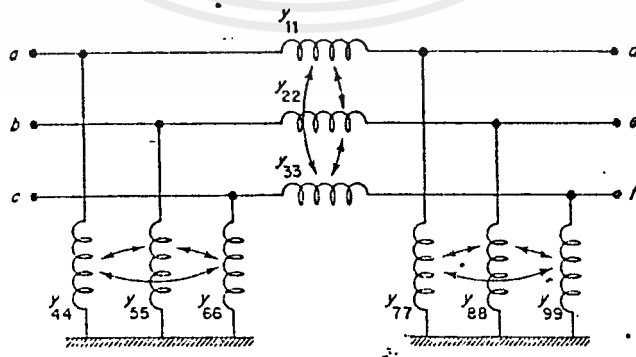
ในกรณีนี้สมการ (9) จะเขียนได้เป็น

$$\begin{bmatrix} [I_a] \\ [I_b] \end{bmatrix} = \begin{bmatrix} [Y_{aa}] & [Y_{ab}] \\ [Y_{ba}]^T & [Y_{bb}] \end{bmatrix} \begin{bmatrix} [V_a] \\ [V_b] \end{bmatrix} \quad (4.1.11)$$

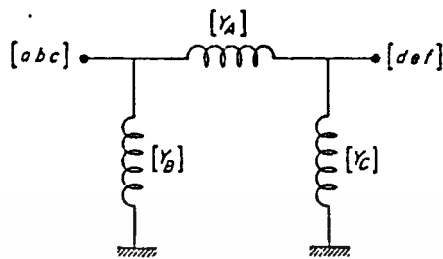
ดังนั้น Primitive network สำหรับ compound admittance จะถูกสร้างขึ้นโดยตรงจาก admittance แต่ละตัว และปริมาณต่างๆ ยังเป็นเมตริกซ์ที่อยู่ในตำแหน่งเดียวกับของ compound admittance

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

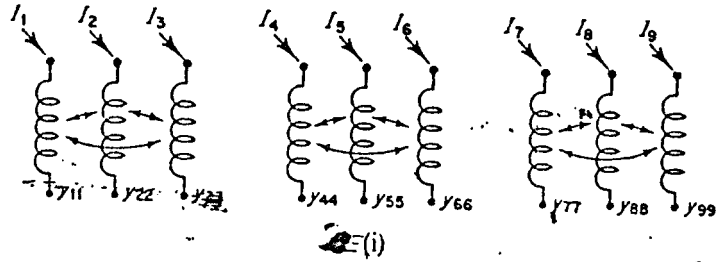


รูป 4.4 Sample network represented by single admittance



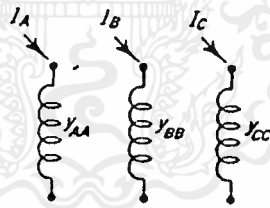
รูป 4.5 Sample network represented by compound admittance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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}

(ii)



(iii)

y_A		
	y_B	
		y_C

(iv)

รูป 4.6 Primitive networks และ admittance matrices

- (i) Primitive network โดยใช้ single admittance
- (ii) Primitive admittance matrix
- (iii) Primitive network โดยใช้ compound admittance
- (iv) Primitive admittance matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \\ V_9 \end{bmatrix} = \begin{bmatrix} -1 & & & 1 & & \\ & -1 & & & 1 & \\ & & -1 & & & 1 \\ 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} * \begin{bmatrix} V_a \\ V_b \\ V_c \\ V_d \\ V_e \\ V_f \end{bmatrix} \quad (4.1.12)$$

$$\begin{bmatrix} [V_A] \\ [V_B] \\ [V_C] \end{bmatrix} = \begin{bmatrix} -I & I \\ I & \\ & I \end{bmatrix} * \begin{bmatrix} [V_{abc}] \\ [V_{def}] \end{bmatrix} \quad (4.1.13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนตเวิร์คของ admittance matrix จะถูกกำหนดโดยสมการ Linear transformation ดังนี้

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

การคูณเมตริกซ์ไม่สามารถใช้ทั้งรูปเต็มของเมตริกซ์หรืออยู่ในรูปแบ่งกันแล้วก็ได้ ผลของการแบ่งกันจะได้

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

กฎในการสร้าง admittance matrix ในเนตเวิร์ค

วิธี Linear transformation อาจจะใช้หา admittance matrix ของเนตเวิร์คทั่วไปได้ แต่ในกรณีพิเศษคือเนตเวิร์คไม่มีการ mutual coupling ระหว่างกัน กฎเกณฑ์ง่าย ๆ ในการสร้าง admittance matrix โดยการตรวจสอบ โดยให้ compound network ที่ไม่มีการ mutual coupling ระหว่าง compound admittances โดยทำตามดังนี้ :

1. เทอม diagonal มาจาก ผลรวมของแต่ละ branch admittance ซึ่งค่าอยู่กับโหนดต่างๆ ที่เราพิจารณา
2. เทอม off-diagonal มาจากค่าลบของผลรวมของ branch admittances ซึ่งค่าอยู่ระหว่างโหนดที่พิจารณา

4.2 แบบจำลองของสายส่ง 3 เฟส (Three phase model of transmission lines)

ค่าต่างๆ ของสายส่ง มักจะคำนวณจากลักษณะทางเรขาคณิตของสาย ค่าพารามิเตอร์ที่คำนวณได้จะอยู่ในรูปของ series impedance และ shunt admittance ต่อหน่วยความยาวของสาย ผลของ ground currents และ earth wires จะรวมอยู่ในการคำนวณของพารามิเตอร์เหล่านี้

Series impedance

สายส่งแบบ 3 เฟสที่มี ground wire แสดงในรูป 4.7 (a)

สำหรับเฟส a สามารถเขียนสมการได้ดังนี้

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

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

แทนค่าด้วย

$$I_n = I_a + I_b + I_c + I_x$$

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

จัดกลุ่มและแทนค่า V_n

$$\begin{aligned} \Delta V_a &= V_a - V'_a \\ &= I_a(R_a + j\omega L_a - j\omega L_{an} + R_n + j\omega L_n - j\omega L_{na}) \\ &\quad + I_b(j\omega L_{ab} - j\omega L_{bn} + R_n + j\omega L_n - j\omega L_{nb}) \\ &\quad + I_c(j\omega L_{ac} - j\omega L_{cn} + R_n + j\omega L_n - j\omega L_{nc}) \\ &\quad + I_x(j\omega L_{ax} - j\omega L_{xn} + R_n + j\omega L_n - j\omega L_{nx}) \end{aligned}$$

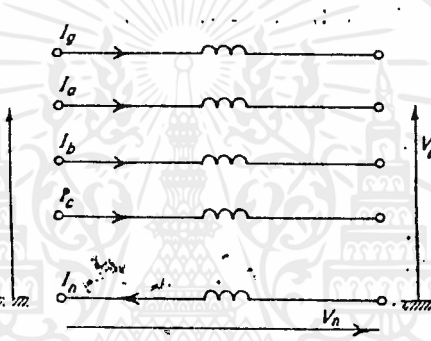
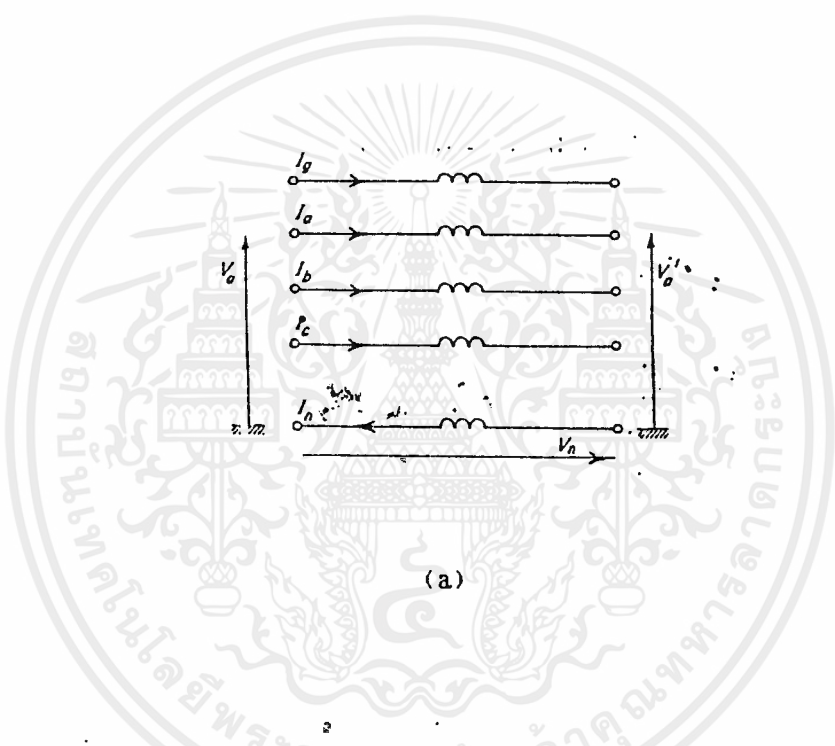
$$\begin{aligned} \Delta V_a &= I_a(R_a + j\omega L_a - 2j\omega L_{an} + R_n + j\omega L_n) \\ &\quad + I_b(j\omega L_{ab} - j\omega L_{bn} + j\omega L_{an} + R_n + j\omega L_n) \\ &\quad + I_c(j\omega L_{ac} - j\omega L_{cn} + j\omega L_{an} + R_n + j\omega L_n) \\ &\quad + I_x(j\omega L_{ax} - j\omega L_{xn} + j\omega L_{an} + R_n + j\omega L_n) \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้พิมพ์ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

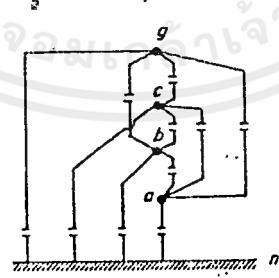
$$\Delta V_n = Z_{aa-n} I_a + Z_{ab-n} I_b + Z_{ac-n} I_c + Z_{ag-n} I_g \quad (4.2.1)$$

ΔV_a	Z_{aa-n} Z_{ab-n} Z_{ac-n}	Z_{ag-n}	I_a
ΔV_b	Z_{ba-n} Z_{bb-n} Z_{bc-n}	Z_{bg-n}	I_b
ΔV_c	Z_{ca-n} Z_{cb-n} Z_{cc-n}	Z_{cg-n}	I_c
ΔV_n	Z_{na-n} Z_{nb-n} Z_{nc-n}	Z_{ng-n}	I_g

(4.2.2)



(a)



(b)

รูป 4.7 (a) Three phase transmission series impedance equivalent
 (b) Three phase transmission shunt impedance equivalent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

$$\begin{array}{|c|} \hline \Delta V_{abc} \\ \hline \Delta V_g \\ \hline \end{array} = \begin{array}{|cc|} \hline Z_A & Z_B \\ \hline Z_C & Z_D \\ \hline \end{array} \begin{array}{|c|} \hline I_{abc} \\ \hline I_g \\ \hline \end{array} \quad (4.2.3)$$

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

$$Z_{abc} = \begin{array}{c} Z_a - Z_b Z_d^{-1} Z_c \\ \hline \end{array} = \begin{array}{|ccc|} \hline Z'_{aa-n} & Z'_{ab-n} & Z'_{ac-n} \\ \hline Z'_{ba-n} & Z'_{bb-n} & Z'_{bc-n} \\ \hline Z'_{ca-n} & Z'_{cb-n} & Z'_{cc-n} \\ \hline \end{array}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shunt admittance

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

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

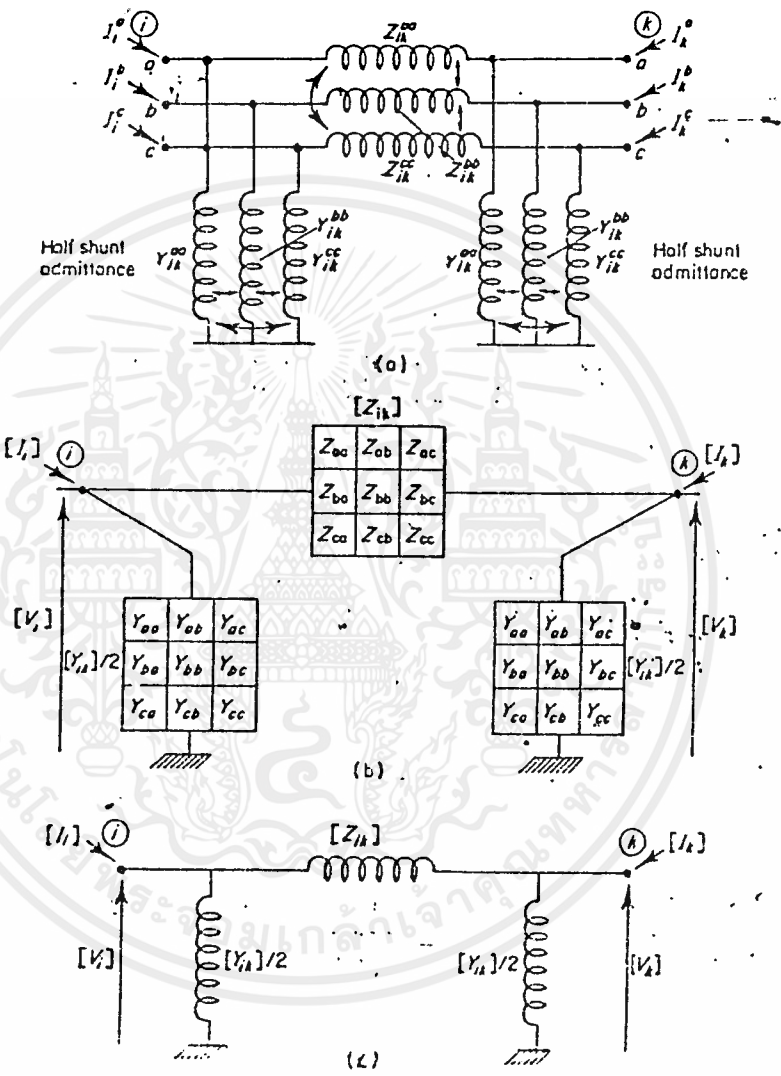
พิจารณาร่วมกับ series impedance matrix จะได้

$$V_{abc} = P'_{abc} Q_{abc} \tag{4.2.8}$$

เมื่อ P'_{abc} คือ 3*3 เมตริกซ์ ซึ่งผลรวมของ ground wire และ capacitance matrix ของ transmission line ในรูป 4.7 เขียนได้เป็น

$$C'_{abc} = P'^{-1}_{abc} = \begin{bmatrix} C_{aa} & -C_{ab} & -C_{ac} \\ -C_{ba} & C_{bb} & -C_{bc} \\ -C_{ca} & -C_{cb} & C_{cc} \end{bmatrix}$$

Series impedance และ shunt admittance แทนสายส่ง 3 เฟส ด้วยแบบจำลอง π ดังแสดงในรูปที่ 4.8 (a) และแทนสายส่งด้วย เมตริกซ์ในรูป 4.8 (b) ซึ่งเมตริกซ์ทั้งสองสามารถแทนด้วย compound admittances รูป 4.8 (c) ดังที่ได้อธิบายไว้แล้ว



รูป 4.8 Lumped- π model ของ short three-phase line series impedances
 (a) Full circuit representation
 (b) Matrix equivalent
 (c) ใช้ Three-phase compound admittance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้การแปลง admittance matrix ซึ่งจะแสดงกระแสที่ไหลเข้าโหนดในรูป 4.8(c) ซึ่งมีความสัมพันธ์กับ โวลเตจ ดังสมการ

$$\begin{array}{|c|} \hline [I_i] \\ \hline [I_k] \\ \hline \end{array} = \begin{array}{|c|c|} \hline [Z]^{-1} + [Y]/2 & -[Z]^{-1} \\ \hline -[Z]^{-1} & [Z]^{-1} + [Y]/2 \\ \hline \end{array} \begin{array}{|c|} \hline [V_i] \\ \hline [V_k] \\ \hline \end{array} \quad (4.2.9)$$

โดยสมการข้างต้นเป็นการร่าง admittance matrix แทนสายส่งระยะสั้นระหว่างบัส i และ k ในเทอมของ 3*3 เมตริกซ์

โดยแบบจำลอง นี้ อาจจะมีความแม่นยำน้อย ในกรณีสายส่งระยะไกล ซึ่งเป็นผลจากความยาวคลื่น ถ้าพิจารณาถึงความถี่อาร์โมนิคด้วยแล้ว ผลอันนี้จะน้อยมาก โดยใช้สายส่งและทฤษฎีของ wave propagation เราจะหาแบบจำลองได้ละเอียดยิ่งขึ้น อย่างไรก็ตามถ้าพิจารณาในย่านความถี่ปกติ เราสามารถใช้แบบจำลองของ π network ในสายส่งระยะยาว

Mutually coupled three phase lines

เมื่อสายส่ง 2 เส้นหรือมากกว่าวางอยู่ในแนวเดียวกัน จะต้องคำนึงถึงผลของ สนามไฟฟ้าสถิต (electrostatics) และ สนามแม่เหล็กไฟฟ้า (electromagnetic) พิจารณากรณีที่ง่ายที่สุดคือ มีสายส่ง 3 เฟส 2 ชุด ซึ่งมี mutually coupled ระหว่างกัน การ coupled ของสาย 2 ชุด ถูกแยกเป็นระบบย่อย (subsystem) ซึ่งประกอบด้วย บัสบาร์ของระบบ สายที่มีการ coupled กันแสดงในรูปที่ 4.9 เมื่อแต่ละอิลิเมนต์เป็น 3*3 compound admittance และทั้งกระแสและโวลเตจเป็นเวกเตอร์แบบ 3*1

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

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

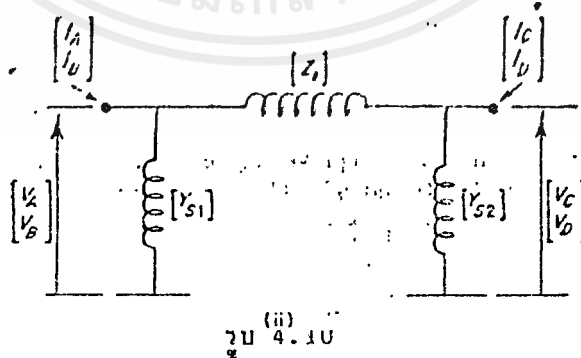
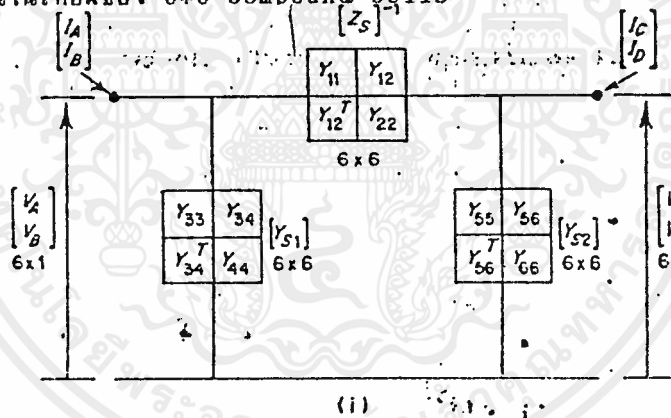
I_A	$Y_{11} + Y_{23}$	$Y_{12} + Y_{34}$	$-Y_{11}$	$-Y_{12}$	V_a
I_B	$Y_{12}^T + Y_{34}^T$	$Y_{22} + Y_{44}$	$-Y_{12}^T$	$-Y_{22}$	V_b
I_C	$-Y_{11}$	$-Y_{12}$	$Y_{11} + Y_{55}$	$Y_{12} + Y_{56}$	V_c
I_D	$-Y_{12}^T$	$-Y_{22}$	$Y_{12}^T + Y_{56}^T$	$Y_{22} + Y_{66}$	V_d

(4.2.10)

12×1
 12×12
 12×1

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

ระบบย่อยอาจจะเขียนใหม่ได้ดังในรูป 4.10 แต่ละคู่ของ couples 3*3 compound admittance จะถูกแทนด้วย 3*3 compound admittance และแสดงการแทนเมตริกซ์ไว้ในรูปด้วย โดยใช้แบบจำลองนี้และสัญลักษณ์ที่แสดงใน admittance blocks ในรูป ค่า admittance matrix อาจเขียนในเทอมของ 6*6 compound coils



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

(i) 6*6 Matrix representation

(ii) 6*6 Compound admittance representation

$$\begin{array}{c} \left[\begin{array}{c} I_A \\ I_B \end{array} \right] \\ \left[\begin{array}{c} I_C \\ I_D \end{array} \right] \end{array} = \begin{array}{|c|c|} \hline [Z_{u1}]^{-1} + [Y_{u1}] & -[Z_{u1}]^{-1} \\ \hline -[Z_{u1}]^{-1} & [Z_{u1}]^{-1} + [Y_{u2}] \\ \hline \end{array} \begin{array}{c} \left[\begin{array}{c} V_A \\ V_B \end{array} \right] \\ \left[\begin{array}{c} V_C \\ V_D \end{array} \right] \end{array} \quad (4.2.11)$$

12*1
12*12
12*1

ซึ่งจะเห็นได้ชัดว่ามีรูปแบบเหมือนสมการ (4.2.10) โดยการแบ่งกันเมตริกซ์

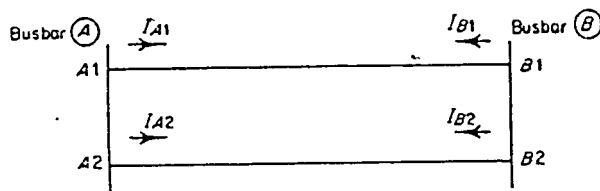
การแทนในรูป 4.10 มีความซับซ้อนและละเอียดแสดงโดยสมการ (4.2.11) ซึ่งคล้ายกับผลที่ใช้ 3*3 compound admittance ใน single three-phase line

ข้อมูลที่จะเป็นเพื่อสร้างเมตริกซ์ขึ้นนั้นจะคล้ายกับแบบ single line คือจะเป็น series impedance และ shunt admittance matrix และจะพบว่าเป็นแบบ 3*3 สำหรับ single line, 6*6 สำหรับ Two couple lines, 9*9 สำหรับ three coupled lines และ 12*12 สำหรับ four coupled lines

การหสเมตริกซ์ $[Z_u]$ และ $[Y_u]$ admittance matrix สำหรับระบบย่อยถูกสร้างโดยใช้สมการ (4.2.11)

เมื่อบัสบาร์ทั้งหมดของ couple line มีความแตกต่างกัน ระบบย่อยอาจรวมกันโดยตรงเป็น system admittance matrix อย่างไรก็ตาม ถ้าบัสบาร์ไม่มีความแตกต่างกันแล้ว admittance matrix อาจจะได้จากสมการ (4.2.11) ซึ่งจะพิจารณาต่อไป

การพิจารณา terminal connections admittance matrix ที่กล่าวมาแล้วสามารถลดรูปลงได้ถ้ามี elements ที่ต่างกันอยู่ติดอยู่ที่บัสบาร์เดียวกัน พิจารณาสายส่งที่ขนานกัน 2 เส้นในรูปที่ 4.11



รูป 4.11 Mutually coupled parallel transmission lines

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

$$\begin{bmatrix} I_{A1} \\ I_{A2} \\ I_{A3} \\ I_{A4} \end{bmatrix} = \begin{bmatrix} Y_{A1} & & & \\ & Y_{A2} & & \\ & & Y_{B1} & \\ & & & Y_{B2} \end{bmatrix} \begin{bmatrix} V_{A1} \\ V_{A2} \\ V_{A3} \\ V_{A4} \end{bmatrix} \quad (4.2.12)$$

nodal injected current ที่บัสบาร์ A (I_A) เขียนได้เป็น

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

ทำนองเดียวกัน

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

จากการสังเกตในรูปที่ 4.11

$$\begin{aligned} V_A &= V_{A1} = V_{A2} \\ V_B &= V_{B1} = V_{B2} \end{aligned}$$

เมตริกซ์ที่ต้องใช้ในการแสดงความสัมพันธ์ระหว่างกระแส และโวลเตจของบัสบาร์ หาได้จากสมการ (4.2.12) และเงื่อนไขที่ได้กล่าวไว้แล้วตอนต้นดังนี้

$$\begin{bmatrix} I_A \\ I_B \end{bmatrix} = \begin{bmatrix} Y_{AB} \end{bmatrix} \begin{bmatrix} V_A \\ V_B \end{bmatrix} \quad (4.2.13)$$

เมตริกซ์ $[Y_{AB}]$ เป็น nodal admittance matrix ของ subsystem

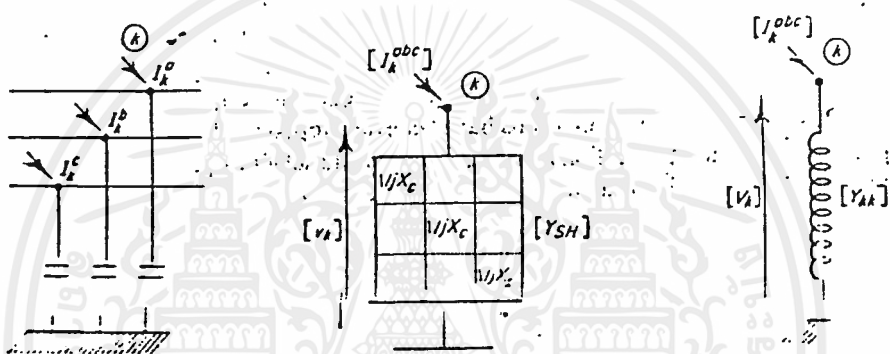
และสังเกตว่าเมตริกซ์ในสมการ (4.2.12) จะต้องนำไปใช้คำนวณหาค่า power line flows ในส่วนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shunt elements

shunt reactors และ capacitors จะใช้ในระบบไฟฟ้ากำลังเพื่อควบคุม reactive power ข้อมูลสำหรับ element เหล่านี้โดยทั่วไปจะอยู่ในรูปพิกัด MVA และ kV ของตัวมันเอง ค่า 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 โดยใช้แค่เพียงค่า self admittance ของแต่ละบัส



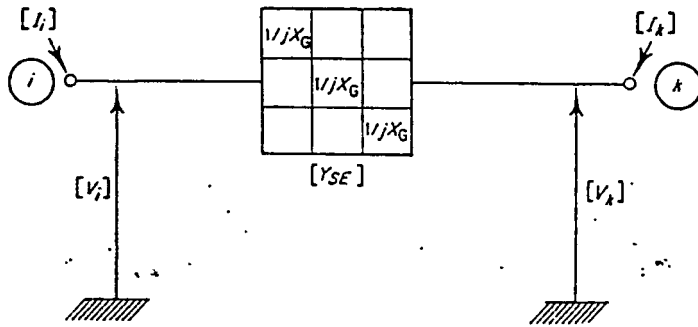
รูป 4.12 Representation of shunt capacitor bank

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

$$[Y] = \begin{bmatrix} [Y_{SE}] & -[Y_{SE}] \\ -[Y_{SE}] & [Y_{SE}] \end{bmatrix} \quad (4.2.14)$$



รูป 4.13 Graphic representation of series capacitor bank between nodes i, k

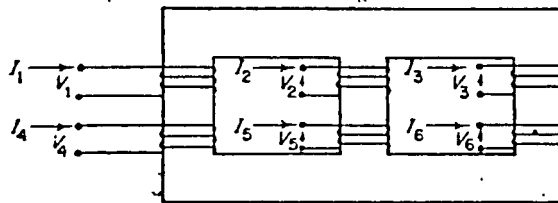
4.3 แบบจำลองของหม้อแปลง 3 เฟส (Three-phase models of transformers)

ข้อสมมติว่าหม้อแปลงเป็นระบบ 3 เฟสสมมูล เป็นเงื่อนไขที่สำคัญในระบบปฏิบัติงาน โดยหม้อแปลง 3 เฟสถูกแทนด้วยวงจรสมมูล

วิธีที่ใช้ในการหาแบบจำลองของหม้อแปลง 3 เฟส ก็โดยวิธี linear transformation ในการหา admittance matrix ขึ้นมา

Primitive admittance model ของ หม้อแปลง 3 เฟส

หม้อแปลง 3 เฟสโดยทั่วไปจะพันขดลวดอยู่ในแกน และมีการ coupling ระหว่างขดลวดทั้งหมด โดยทั่วไปหม้อแปลง 3 เฟสแบบ two winding จะมี primitive หรือ unconnected network ซึ่งประกอบด้วย coil ที่ couple กัน 6 ตัว ถ้าพิจารณาถึง tertiary winding จะได้เป็นแบบ primitive network ที่ประกอบด้วย coil ที่ couple กัน 9 ตัว พิจารณาหม้อแปลงแบบ two winding ได้แสดงไว้ในรูป 4.14 โดยใช้วิธีที่จะกล่าวต่อไปนี้

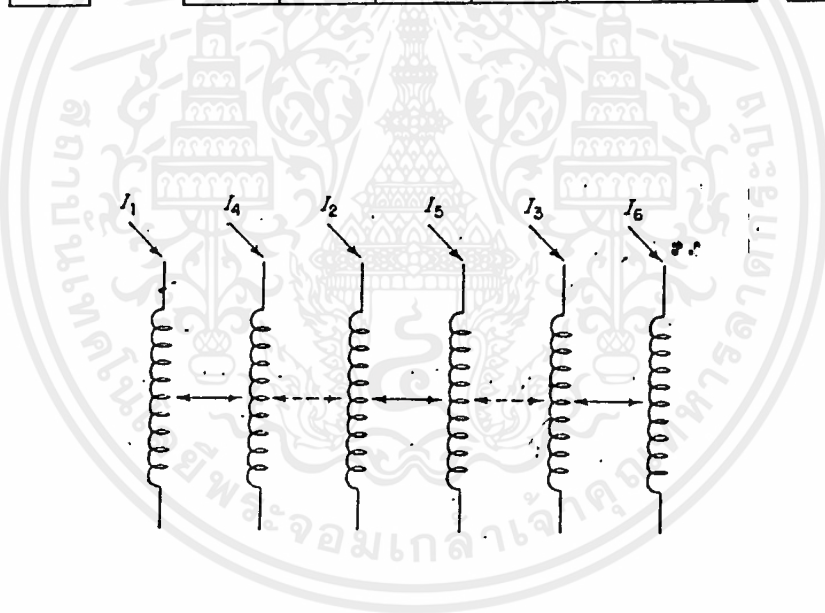


รูป 4.14 Diagrammatic representation of two-winding transformer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

(4.3.1)



รูป 4.15 Primitive network of two-winding transformer. Six coupled coil primitive network. (Note the dotted coupling represents parasitic coupling between phase)

element ของ matrix $[Y]$ สามารถวัดได้ โดยตัวอย่างเช่นจ่ายกระแสเข้าที่ coil i และ short-circuiting coil อื่นๆ แนวตั้ง i ของ $[Y]$ คำนวณได้จาก $y_{ki} = I_k/V_i$

พิจารณาส่วนกลับของ mutual coupling ในสมการที่ (1) การวัดค่า short circuit ทั้ง 21 ตัว เป็นสิ่งที่จำเป็นสำหรับการหา admittance matrix

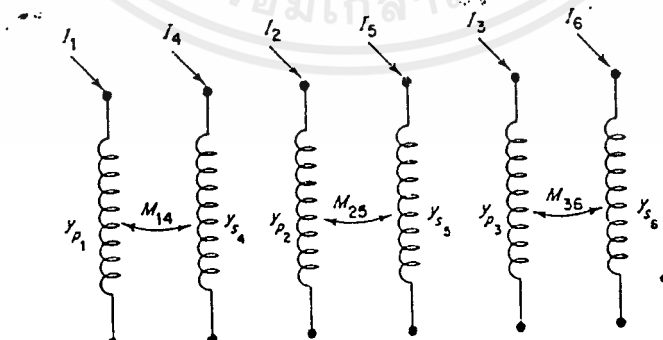
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสมมติให้ flux path มีความสมมาตรกันระหว่างขดลวด สมการ (1) อาจเขียน ได้ดังนี้

I_1	y_p	y'_m	y'_m	$-y_m$	y''_m	y''_m	V_1
I_2	y'_m	y_p	y'_m	y''_m	$-y_m$	y''_m	V_2
I_3	y'_m	y'_m	y_p	y''_m	y''_m	$-y_m$	V_3
I_4	$-y_m$	y''_m	y''_m	y_p	y'''_m	y'''_m	V_4
I_5	y''_m	$-y_m$	y''_m	y'''_m	y_p	y'''_m	V_5
I_6	y''_m	y''_m	$-y_m$	y'''_m	y'''_m	y_p	V_6

(4.3.2)

เมื่อ y'_m คือ mutual admittance ระหว่าง primary coils
 y''_m คือ mutual admittance ระหว่าง primary และ secondary coils ที่
 อยู่คนละแกนกัน
 y'''_m คือ mutual admittance ระหว่าง secondary coils
 สำหรับ three separate single phase units ซึ่งมีเครื่อง prim จะเป็นศูนย์
 และใน three phase unit ซึ่งมีเครื่องหมาย prim จะแสดงถึงมีการ coupling ระหว่างเฟส
 ซึ่งสามารถแสดงในรูปของ symmetrical component



where $y_p = y/\alpha^2$, $y_s = y/\beta^2$ and $M_{ij} = y/\alpha\beta$
 for $i = 1, 2$ or 3 and $j = 4, 5$ or 6

รูป 4.16 Primitive network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ (4.3.2) จะสอดคล้องกับ primitive network และหลักการ coupling ระหว่างเฟส การ coupling ระหว่าง primary และ secondary coil ซึ่งแทนด้วยแบบจำลองของ primitive network ในรูป 4.12
สมการ admittance matrix ใหม่จะเป็น

$$\begin{array}{c} I \\ I \\ I \\ I \\ I \\ I \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline y_{p1} & & & M_{14} & & \\ \hline & y_{p2} & & & M_{25} & \\ \hline & & y_{p3} & & & M_{36} \\ \hline M_{41} & & & y_{s1} & & \\ \hline & M_{52} & & & y_{s5} & \\ \hline & & M_{63} & & & y_{s6} \\ \hline \end{array} \begin{array}{c} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \quad (4.3.3)$$

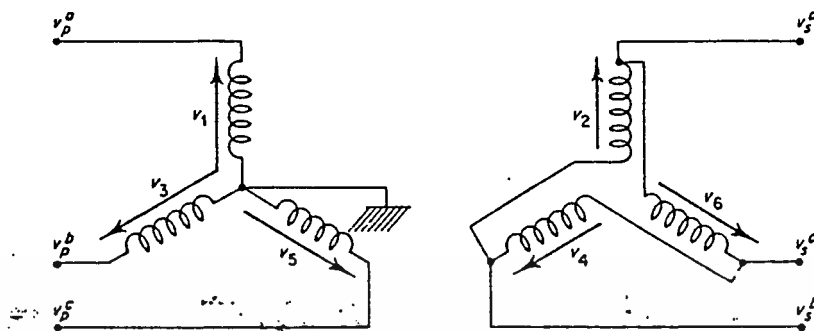
แบบจำลองของการต่อหม้อแปลงแบบต่างๆ

Network admittance matrix สำหรับหม้อแปลง 3 เฟสแบบ two-winding สามารถสร้างโดยวิธี linear transformer

พิจารณาการสร้าง admittance matrix สำหรับการต่อแบบ Y-Y ซึ่งมี solidly earthed โดยไม่มีการ mutuals ระหว่างเฟส ซึ่งเป็นวิธีที่วิเคราะห์ง่ายที่สุด

สำหรับหม้อแปลงที่ต่อแบบ Y-Y ในรูป 4.17 connection matrix [c] มีความสัมพันธ์กับ Branch voltages (โวลเตจของ primitive network) และ node voltage (voltage ของ actual network) คือเป็น 6*6 identity matrix

พิจารณาการต่อแบบ Wye G-Delta ดังในรูปที่ 18



รูป 4.18 Network connection diagram สำหรับ Wye G-Delta transformer

จากการตรวจ ทำให้สามารถเขียนแสดงความสัมพันธ์ระหว่าง primitive branch voltage และ node voltage ได้ดังนี้

V_1	1	0	0	0	0	0	V_p^a
V_2	0	0	0	1	-1	0	V_p^b
V_3	0	1	0	0	0	0	V_p^c
V_4	0	0	0	0	1	-1	V_s^a
V_5	0	0	1	0	0	0	V_s^b
V_6	0	0	0	-1	0	1	V_s^c

(4.3.7)

หรือ $[V]_{branch} = [C][V]_{node}$ (4.3.8)

เราเขียนได้ว่า $[Y]_{node} = [C]^T [Y]_{prim} [C]$ (4.3.9)

และใช้ $[Y]_{prim}$ จากสมการ (2)

$$[Y]_{\text{node}} =$$

y_p	y'_m	y'_m	$-(y_m+y''_m)$	$y_m+y''_m$	0	a
y'_m	y_p	y'_m	0	$-(y_m+y''_m)$	$(y_m+y''_m)$	b
y'_m	y'_m	y_p	$(y_m+y''_m)$	0	$-(y_m+y''_m)$	c
$-(y_m+y''_m)$	0	$(y_m+y''_m)$	$R(y_m-y''_m)$	$-(y_m-y''_m)$	$-(y_m-y''_m)$	A
$(y_m+y''_m)$	$-(y_m+y''_m)$	0	$-(y_m-y''_m)$	$2(y_m-y''_m)$	$-(y_m-y''_m)$	b
0	$(y_m+y''_m)$	$-(y_m+y''_m)$	$-(y_m-y''_m)$	$-(y_m-y''_m)$	$R(y_m-y''_m)$	C

$$(4.3.10)$$

ถ้า primitive admittances แสดงอยู่ในรูป per unit แล้วทั้ง primary และ secondary voltages จะอยู่ในรูป per unit ด้วย หม้อแปลงที่ต่อแบบ Y-Δ จะต้องรวมผลของ turn ratio ด้วยค่า $\sqrt{3}$ ควอดแรนท์ อยู่มุมขวาและล่างซ้ายของเมตริกซ์ สมการ 9.10 จะต้องถูกหารด้วย $\sqrt{3}$ และควอดแรนท์ด้านล่างขวาจะถูกหารด้วย 3

โดยทั่วไปหม้อแปลง 3 เฟสที่เกิดจากหม้อแปลงเฟสเดียวมาต่อกันเป็นแบบ Y-Δ ค่าทุกค่าของ Y' และ Y'' จะละทิ้งได้ ถ้าไม่คิด off monomial taps (แต่ต้องคิดถึงผลของ $\sqrt{3}$ turn ratio ในรูป per unit) สมการของ nodal admittance matrix จะมีความสัมพันธ์กับ nodal current และ nodal voltage ดังนี้

I^a_p	y		$y/\sqrt{3}$	$y/\sqrt{3}$		V^a_p
I^b_p		y		$-y/\sqrt{3}$	$y/\sqrt{3}$	V^b_p
I^c_p			y	$y/\sqrt{3}$	$-y/\sqrt{3}$	V^c_p
I^A_m	$-y/\sqrt{3}$		$y/\sqrt{3}$	$2y/3$	$-y/3$	V^A_m
I^B_m	$y/3$	$-y/\sqrt{3}$		$-y/3$	$2y/3$	V^B_m
I^C_m		$-y/\sqrt{3}$	$-y/\sqrt{3}$	$-y/3$	$-y/3$	V^C_m

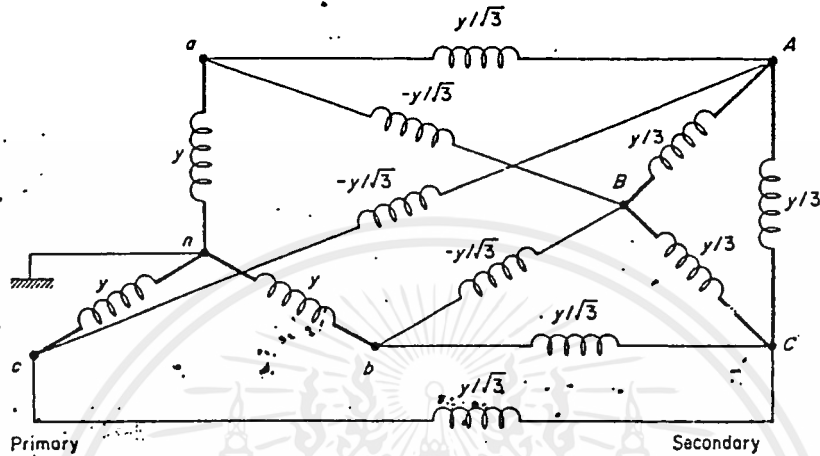
$$(4.3.11)$$

เมื่อ y เป็นค่า transformer leakage admittance in p.u.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และวงจรสมมูลสามารถเขียนใหม่ตามค่า admittance model ดังในรูปที่ 19

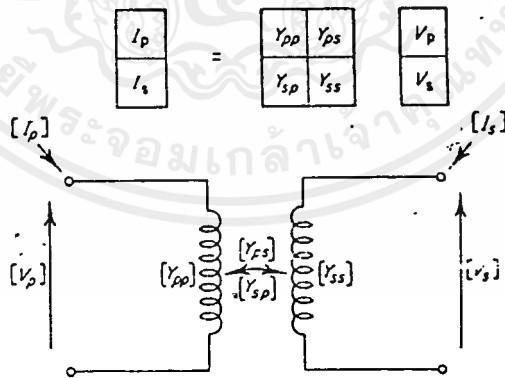
ค่า shunt admittance ที่ต่อลงดินที่มีขนาดใหญ่ จากโหนดของการต่อแบบ star ปรากฏอยู่ในวงจรสมมูล ค่า shunt นี้จะมีค่าประมาณ 10 p.u. (สำหรับหม้อแปลง ที่มี leakage reactance 10%)



รูป 4.19 วงจรสมมูลของหม้อแปลงแบบ Y-Δ

การต่อหม้อแปลงแบบนี้ก็สามารถหาแบบจำลองได้ด้วยวิธีเดียวกันนี้

โดยทั่วไปหม้อแปลง 3 เฟสแบบ two-winding จะแทนด้วย compound coil ที่ coupled กัน 2 ชุด วงจรสมมูลแบบ admittance matrix แสดงในรูป 4.20



รูป 4.20 Two-winding three-phase transformer as two coupled coils

และสังเกตว่า

$$[Y_{pp}] = [Y_{ss}]^T$$

และการ coupling ระหว่าง compound coils 2 ชุด จะเป็น bilateral กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีบางครั้งที่ไม่ต้องการความละเอียดมาก พารามิเตอร์ของทั้ง 3 เฟสจึงสมมุติให้สมดุล ในกรณีที่มีการต่อแบบ 3 เฟส จะถูกแทนด้วย เมตริกซ์ย่อย 3 ชุดแทน เมตริกซ์ย่อย $[Y_{pp}]$, $[Y_{ss}]$ จะถูกกำหนดในตาราง 1 สำหรับการต่อแบบต่างๆ ไป

ตาราง 4.1 Characteristic submatrix ที่ใช้สร้าง admittance matrix ของหม้อแปลง-

transformer connection		self admittance		mutual admittance
Bus P	Bus S	Y_{pp}	Y_{ss}	Y_{ps}, Y_{sp}
Wye-G	Wye-G	Y_1	Y_1	$-Y_1$
Wye-G	Wye	$Y_{11/3}$	$Y_{11/3}$	$-Y_{11/3}$
Wye-G	Delta	Y_1	Y_1	Y_{111}
Wye	Wye	$Y_{11/3}$	$Y_{11/3}$	$-Y_{11/3}$
Wye	Why	Y_{11}	Y_{11}	Y_{111}
Delta	Delta	Y_{11}	Y_{11}	$-Y_{11}$

$$Y_1 = \begin{bmatrix} y_\epsilon & & \\ & y_\epsilon & \\ & & y_\epsilon \end{bmatrix} \quad Y_{11} = \begin{bmatrix} 2y_\epsilon & -y_\epsilon & -y_\epsilon \\ -y_\epsilon & 2y_\epsilon & -y_\epsilon \\ -y_\epsilon & -y_\epsilon & 2y_\epsilon \end{bmatrix}$$

$$Y_{111} = \begin{bmatrix} -y_\epsilon & y_\epsilon & \\ & -y_\epsilon & y_\epsilon \\ y_\epsilon & & -y_\epsilon \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sequence components modelling of three-phase transformers

ในหัวข้อนี้จะพิจารณาถึงวงจรสมมูลของหม้อแปลงที่มี sequence components เมื่อพิจารณาหม้อแปลงที่ต่อแบบ Wyne G-Delta ในรูปที่ 4.18 ซึ่งแทนด้วยสมการ (4.3.10) และแบ่งเมตริกซ์นี้ออกโดยแยกเป็น self และ mutual elements โดยจะแปลงได้ดังนี้

ด้าน Primary

$$y_{120}^p = T_s^{-1} \begin{bmatrix} y_p & y'_m & y'_m \\ y'_m & y_p & y'_m \\ y'_m & y'_m & y_p \end{bmatrix} T_s$$

เมื่อ

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

และ

$$a = e^{j120^\circ/3}$$

ดังนั้น

$$y_{120}^p = \begin{bmatrix} y_p - y'_m & 0 & 0 \\ 0 & y_p - y'_m & 0 \\ 0 & 0 & y_p - y'_m \end{bmatrix} \tag{4.3.12}$$

ด้าน secondary

ด้าน secondary ที่ต่อแบบ delta จะมีผลของ $\sqrt{3}$ คูณกับ turn ratio และ sequence component admittance matrix จะเขียนได้เป็น

$$y_{120}^s = T_s^{-1} / 3$$

$2(y_s - y''''_m)$	$-(y_s - y''''_m)$	$-(y_s - y''''_m)$
$-(y_s - y''''_m)$	$2(y_s - y''''_m)$	$-(y_s - y''''_m)$
$-(y_s - y''''_m)$	$-(y_s - y''''_m)$	$2(y_s - y''''_m)$

 T_s

$y_s S_{-y \dots m}$	o	o
o	$y_s S_{-y \dots m}$	o
o	o	o

(4.3.13)

Mutual terms

Mutual admittance submatrix ในสมการ (4.3.10) จะถูกแปลงตามผลของ turn ratio ดังต่อไปนี้

$$y_{120}^m = \sqrt{3} T_s^{-1} / 3$$

$-(y_m^m - y''''_m)$	$(y_m^m - y''''_m)$	0
o	$-(y_m - y''''_m)$	$(y_m - y''''_m)$
$(y_m - y''''_m)$	o	$-(y_m - y''''_m)$

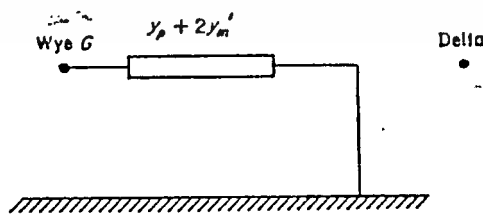
 T_s

$$= \begin{bmatrix} -(y_m + y''_m) 30^\circ & 0 & 0 \\ 0 & -(y_m + y''_m) -30^\circ & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.3.14)$$

รวม sequence component submatrix จะได้ดังนี้

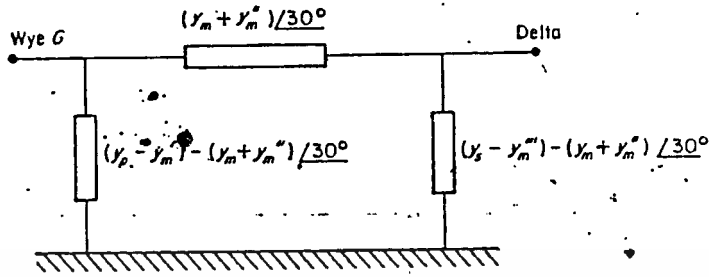
$$= \begin{bmatrix} I_1 & y_p - y'_m & & & -(y_m + y''_m)/30 & & 0 & V_1 \\ I_2 & & y_p - y'_m & & & & -(y_m + y''_m)/-30 & 0 & V_2 \\ I_0 & & & y_p + 2y'_m & & & & 0 & V_0 \\ I_3 & -(y_m + y''_m)/30 & & & y_s - y''_m & & & 0 & V_3 \\ I_4 & & -(y_m + y''_m)/-30 & & & y_s - y''_m & & 0 & V_4 \\ I_5 & & & & & & & 0 & V_5 \end{bmatrix}$$

สมการ (4.3.15) สมการเขียนแทนด้วย three sequence networks ของรูป 21, 22, 23

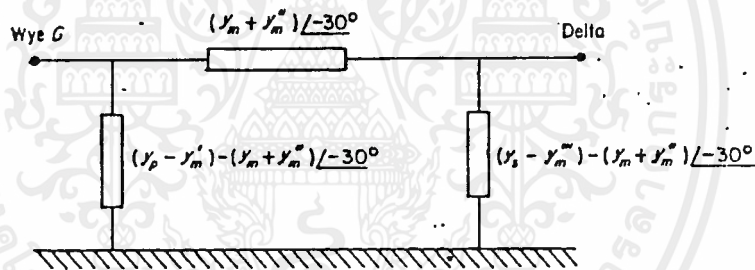


รูป 4.21 Zero-sequence node admittance model for a common-core grounded Wye - Delta transformer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.22 Positive-sequence node admittance model for a common-core grounded Wye-Delta transformer



รูป 4.23 Negative-sequence node admittance model for a common-core grounded Wye-Delta transformer

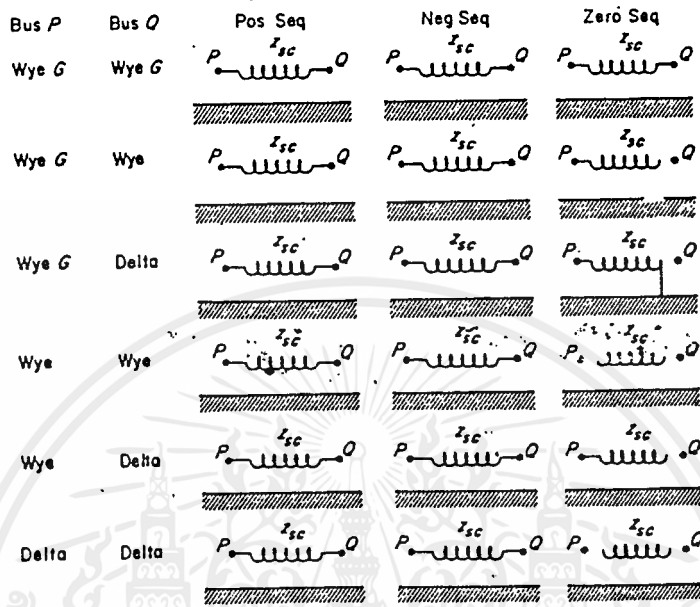
โดยทั่วไป three-phase impedance จะมีความแตกต่างกันใน common-core transformer เราสามารถลดความซับซ้อนของวงจรสมมูลย์ลงได้ด้วยวิธีต่อไปนี้

1. ละทิ้ง phase shift 90° ของการต่อแบบ Wye-Delta
2. สมมติให้ค่า mutual admittance ระหว่างเฟสมีค่าเท่ากัน คือ $y_m' = y_m'' = y_m'''$ ค่าเหล่านี้จะเป็นศูนย์เมื่อพิจารณากรณี 1 เฟส
3. ผลต่างของ $(y_p - y_m)$ และ $(y_s - y_m)$ มีค่าน้อยมากและสามารถตัดทิ้งได้

ตาราง 4.2 จะแสดง sequence impedance model ของหม้อแปลง 3 เฟส ในภาวะคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

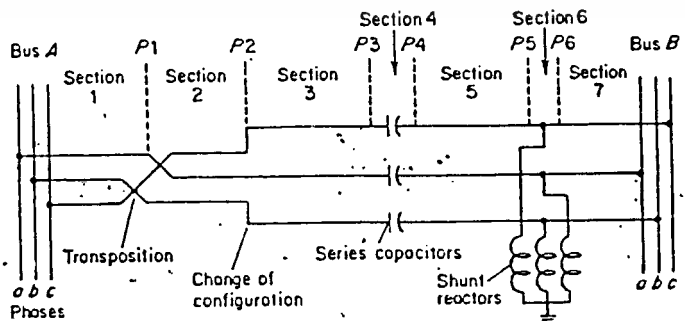
ตาราง 4.2 Typical symmetrical-component model for six most common connection of three-phase transformer



การแยกส่วนของสายส่ง (Line sectionalization)

สายส่งสามารถแยกเป็นส่วนย่อยได้ ตามเงื่อนไขต่อไปนี้

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



รูป 4.24 ตัวอย่างของสายส่งที่แยกกันเป็นส่วนย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

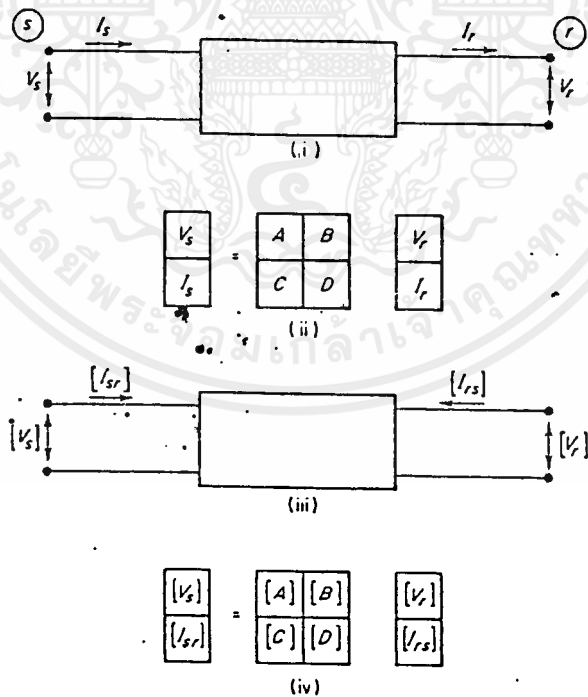
Network ในรูป 4.24 จะถูกพิจารณาเป็นส่วนย่อย ค่า admittance matrix ระหว่าง บัส A และ B หาได้โดยพิจารณาเป็นส่วนๆ ไป โดยหาค่า A B C D หรือ พารามิเตอร์ของ สายส่ง หลังจากนั้นเชื่อมค่าเหล่านี้จะได้ admittance matrix ที่เป็นผลลัพธ์ได้ในที่สุด

วิธีนี้จะใช้ทฤษฎีของ two port network รวกันหลายๆ 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

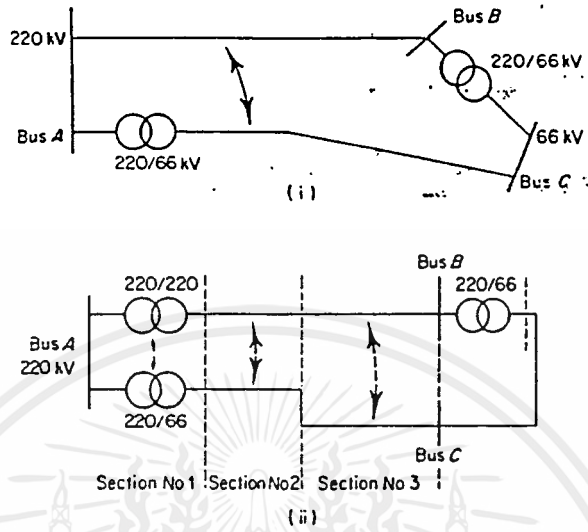
- (a) เพื่อความสะดวกของโปรแกรม จะให้หม้อแปลงเป็นแบบอุดมคติ
- (b) เครื่องหมาย dot แสดงการ coupling ซึ่งกำหนดให้เป็นศูนย์ เพื่อแสดงถึงขนาดของ matrix ได้อย่างถูกต้อง
- (c) ในระบบ per unit mutual coupling ระหว่างสาย 220 kV และ 66kV ถูกแสดง ในรูป voltage base ซึ่งกำหนดโดยค่าเฉลี่ยเรขาคณิต ของ base line neutral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



รูป 4.25 Two part network transmission parameters

- (i) Normal two port network
- (ii) transmisson parameters
- (iii) Multi-two-port network
- (iv) Matrix transmission parameters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.26 Sample system to illustrate line sectionalization

- (i) System single line diagram
- (ii) System redrawn to illustrate line sectionalization

Table 4.3 ABCD Parameter matrices for the common section types

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตาราง 4.3 [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

$$[Y] = \begin{bmatrix} [D][B]^{-1} & [C] - [D][B]^{-1}[A] \\ [B]^{-1} & -[B]^{-1}[A] \end{bmatrix} \quad (4.3.16)$$

Formation of the system admittance matrix

Subsystem admittance matrix สามารถรวมเข้ากันเพื่อ system admittance matrix ได้ดังนี้

1. ค่า self admittance ของบัสบาร์ใดคือผลรวมของ self admittance matrix ย่อยๆ ทั้งหมดที่บัสบาร์นั้น
2. ค่า mutual admittance ระหว่าง บัสบาร์ 2 จุดใดๆ จะเป็นผลรวมของ mutual admittance matrix ย่อยทั้งหมดของ subsystem ของโหนด 2 โหนดนั้น

บทที่ 5

Three Phase Load Flow

5.1 บทนำ

ในกรณีที่ศึกษาระบบไฟฟ้ากำลัง ที่อยู่ในภาวะ steady state ถ้าเราไม่คำนึงถึงความสมดุลของระบบ (System unbalance) แล้วเราสามารถจะใช้ single phase load flow ดังที่กล่าวมาแล้วก็เพียงพอ อย่างไรก็ตามในทางปฏิบัติ ก็อาจจะเป็นการเปลี่ยนแปลง ในการทำให้ โหลดสมดุล หรืออิมพีแดนซ์ของสายส่งสมดุล อันเนื่องจากสายส่งไฟฟ้าแรงสูงที่ไม่มีการใช้สาย

ผลของระบบไฟฟ้ากำลังที่ไม่สมดุล คือ จะมีกระแสลำดับเฟสลบ (negative sequence current) ซึ่งทำให้โรเตอร์ของเครื่องจักรเกิด overheat ขึ้นได้ และ กระแสลำดับเฟสลบ (Zero sequence current) ยังทำให้เกิดการทำงานผิดปกติ และยังเพิ่มความสูญเสียในสายส่ง ขนาดที่ไม่ได้ใช้สายไว้

แบบจำลองระบบ 3 เฟสที่ใช้หลักการรวม network admittance ทั้งหมด ดังได้กล่าวมาแล้ว เพื่อใช้สำหรับ three phase load flow ต่อไป และเวลาที่ใช้ในการคำนวณ และ storage program ที่ใช้จะมีค่ามากกว่าในกรณีของ single phase มาก ดังนั้น algorithms ที่ใช้ในการคำนวณ จะต้องมีประสิทธิภาพที่ดีกว่าใน single phase load flow โดยที่ไปการศึกษาเกี่ยวกับ three-phase load flow จะใช้ในการวางแผนและเตรียมการ มากกว่าที่จะใช้ขณะปฏิบัติการ

5.2 สัญลักษณ์ต่างๆ (Notation)

เพื่อความสะดวก จะกำหนดต่างๆ ต่อไปนี้แทน three phase vector และ matrix elements โดยใช้ตัวทอซท้าย และตัวยกขึ้นดังนี้

เมื่อพิจารณา a.c.system ที่มีบัสบาร์ทั้งหมด n จะได้

$$n = n_b + n_g$$

n_b คือจำนวนบัสบาร์ ที่มีอยู่ในระบบจริงๆ

n_g คือจำนวนเครื่องจักรไฟฟ้าแบบ ซิงโครนัส (Synchronous machine)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 subscript ต่อไปนี้แทนความหมายต่อไปนี้

reg แทน voltage regulator

int แทน internal busbar ของ generator

gen แทน generator

Subscripts p, m แทนความหมายว่า 3 เฟสของบัสบาร์ใดๆ

สัญลักษณ์ที่สำคัญ

$[I]$ = เวกเตอร์ของ nodal injected current

$[V]$ = เวกเตอร์ของ nodal voltage $[\begin{array}{c} V \\ e^{j\theta} \end{array}]$

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

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

ΔP_i^p = กำลังจริงที่บัสบาร์ i เฟส p

ΔQ_i^p = กำลังเสมือนที่บัสบาร์ i เฟส p

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

$(Q_i^p)^{sp}$ = 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

$|V|_{int, j}$ = มุมของโวลเตจที่ internal busbar ของ generator ตัวที่ j

$P_{gen, j}^{sp}$ = specified total power output ของ generator ตัวที่ j

$V_{term, j}^{sp}$ = voltage regulator specification สำหรับโวลเตจที่บัสทั้ง 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 $(Z_{\pi})_{012}$ ตามลำดับดังนี้

$$[Z_{\pi}]_{abc} = [T_{\pi}][Z_{\pi}]_{012}[T_{\pi}]^{-1} \tag{5.3.1}$$

$$= [T_{\pi}][Z_{\pi}]_{012}[T_{\pi}]^* \tag{5.3.2}$$

เมื่อ

$$[T_{\pi}]_{abc} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \tag{5.3.3}$$

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

$Z_0 + Z_1 + Z_2$	$Z_0 + aZ_1 + aZ_2$	$Z_0 + a^2Z_1 + aZ_2$	(5.3.4)
$Z_0 + a^2Z_1 + aZ_2$	$Z_0 + Z_1 + Z_2$	$Z_0 + aZ_1 + a^2Z_2$	
$Z_0 + aZ_1 + a^2Z_2$	$Z_0 + a^2Z_1 + aZ_2$	$Z_1 + Z_2 + Z_3$	

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

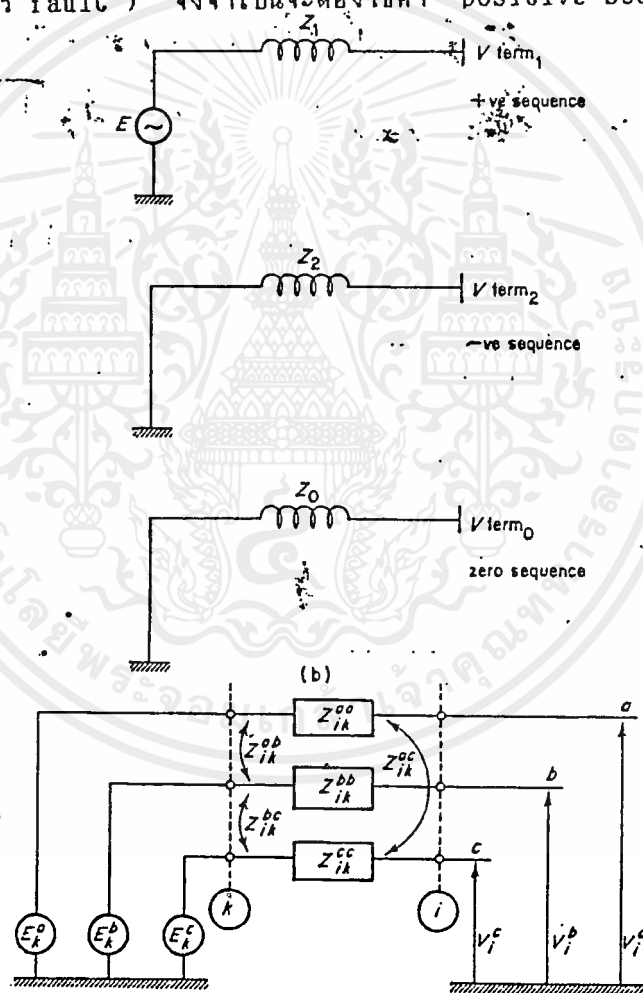
$$E^a_k = E^b_k = E^c_k \tag{5.3.5}$$

และ

$$e^a_k = e^b_k + 2\pi/3 = e^c_k - 2\pi/3 \tag{5.3.6}$$

ในกรณีของ three phase load flow ค่า voltage regulator จะต้องคำนวณให้ละเอียด เพราะมีผลอย่างมากต่อการปฏิบัติงานในภาวะไม่สมดุล เมื่อพิจารณารูป 5.1 (b) ค่า machine excitation จะเป็นแบบ 3 เฟส สมดุล 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



รูป 5.1 Synchronous machine models
 a) Phase component representation
 b) Symmetrical component representation

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

ดังนั้น ในการสร้างแบบจำลอง generator โดยใช้สมการ (5.3.4) ค่าที่กำหนดขึ้นมักใช้เป็น Z_1 แต่ในขณะค่าจริงๆ จะเป็น Z_0 และ Z_2 และค่าความสูญเสียอื่นเนื่องมาจากการไม่สมดุลจะถือว่าน้อยมาก

Nodal admittance matrix มีความสัมพันธ์กับ inject current ของ generator บัสนาร์กับ 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 $\delta_{int,j}$ เป็นมุมอ้างอิง)
- ขนาดของโวลเตจ internal busbar $V_{int,j}$ และมุม $\delta_{int,j}$ ที่ generator ตัวอื่น คือ $j = nb + 1, nb + ng - 1$
- ขนาดของโวลเตจทั้ง 3 เฟส (V^p_i) และมุม (δ^p_i) ของ generator terminal busbar และ Load busbar ทุกตัวในระบบคือ $i = 1, nb$ และ $p = 1, 3$

มีเพียง 2 ตัวแปรเท่านั้น ที่สัมพันธ์กับ generator internal busbar และถ้า 3 เฟส สมดุลย์ แล้วไม่มีความจำเป็นต้องรักษาโวลเตจและมุมเป็นตัวแปรอีก สมการที่ใช้สำหรับหาค่าตอบต่างๆ จะหาได้จากเงื่อนไขต่อไปนี้

- รวมค่ากำลังจริงและกำลังเสมือนที่ไหลทั้งหมดในแต่ละ System busbar
- กำหนดค่า Voltage regulator สำหรับ synchronous ทุกๆ ตัว
- รวมค่ากำลังจริงที่จ่ายออกจาก Synchronous machine แต่ละตัว โดยไม่รวม slack machine ทำโดยกำหนดให้โวลเตจแต่ละเฟสมีขนาดคงที่

Derivation of equation

ระบบ 3 เฟส โดยทั่วไปเขียนได้อยู่ในรูปสมการดังนี้

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

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

จะหาสมการของ system admittance matrix ได้คือ

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

โดยที่

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

$$\begin{aligned} \Delta P_{i,p}^D &= (P_{i,p}^D)^{sp} - P_{i,p}^D \\ &= (P_{i,p}^D)^{sp} - V_{i,p}^D \sum_{k=1}^n \sum_{m=1}^3 V_k^m [G_{ik}^{Dm} \cos \theta_{ik}^{Dm} + B_{ik}^{Dm} \sin \theta_{ik}^{Dm}] \end{aligned} \quad (5.3.8)$$

$$\begin{aligned} \Delta Q_{i,p}^D &= (Q_{i,p}^D)^{sp} - Q_{i,p}^D \\ &= (Q_{i,p}^D)^{sp} - V_{i,p}^D \sum_{k=1}^n \sum_{m=1}^3 V_k^m [G_{ik}^{Dm} \sin \theta_{ik}^{Dm} - B_{ik}^{Dm} \cos \theta_{ik}^{Dm}] \end{aligned} \quad (5.3.9)$$

2. สำหรับ generator j แต่ละตัว

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

เมื่อ k เป็นจำนวนบัสบาร์ของ generator ตัวที่ j

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

$$\begin{aligned} (\Delta P_{gen})_j &= (P_{gen}^{sp})_j - (P_{gen}^{act})_j \\ &= (P_{gen}^{sp})_j - \sum_{k=1}^n \sum_{m=1}^3 V_k^m [G_{jk}^{Dm} \cos \theta_{jk}^{Dm} + B_{jk}^{Dm} \sin \theta_{jk}^{Dm}] \end{aligned} \quad (5.3.11)$$

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

ค่าความสูญเสียของ generator จะมีผลน้อยมากต่อการทำงานของระบบ และสามารถคำนวณได้จาก sequence admittance ในตอนท้ายของการคำนวณ ผลรวมของกำลังที่ไหลออกจาก generator อาจคำนวณได้เนื่องจาก generator internal busbar ไม่ถูกต้องกับ อิลิเมนต์อื่นๆ ในระบบ การหาค่าตอบของโพลดิโพลว์ คือการพยายามทำให้ด้านซ้ายของสมการ (5.3.8) ถึง (5.3.11) มีค่าเป็นศูนย์

5.4 Fast decouple three-phase algorithm

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

$$\begin{bmatrix} \Delta P \\ \Delta P_{\text{gen}} \\ \Delta Q \\ \Delta V_{\text{ref}} \end{bmatrix} = \begin{bmatrix} A & E & I & M \\ B & F & J & N \\ C & G & K & P \\ D & H & L & R \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta \theta_{\text{int}} \\ \Delta V/V \\ \Delta V_{\text{int}}/V_{\text{int}} \end{bmatrix} \quad (5.5.1)$$

เมตริกซ์ทางด้านขวามือของสมการ (1) คือ Jacobian matrix ของ first order partial derivatives

จากกรณีของ decouple single-phase load flow ผลของ $\Delta \theta$ ที่มีต่อ reactive power และ ผลของ ΔV ที่มีต่อ real power สามารถละทิ้งได้ สมการ (5.5.1) อาจจะตัดแปลงโดย

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

และ

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

โดยที่ค่า voltage regulator specified จะถูกสมมติให้อยู่ในรูปของ terminal voltage magnitude ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ (5.5.1) อาจเขียนใหม่ในรูป decouple ได้ดังนี้

$$\begin{bmatrix} \Delta P^p_i \\ \Delta P_{gen..j} \end{bmatrix} = \begin{bmatrix} A & E \\ B & F \end{bmatrix} \begin{bmatrix} \Delta \theta^m_k \\ \Delta \theta_{int..l} \end{bmatrix} \quad (5.5.2)$$

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

$$\begin{bmatrix} \Delta Q^p_i \\ \Delta V_{gen..j} \end{bmatrix} = \begin{bmatrix} K & P \\ L & R \end{bmatrix} \begin{bmatrix} \Delta V^m_k / V^m_k \\ \Delta V_{int..l} / V_{int..l} \end{bmatrix} \quad (5.5.2)$$

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

พิจารณา Jacobian submatrix ในการคำนวณตั้งแต่สมการ (3.8) ถึง (3.11) จะต้องจำไว้ว่า

$$V^1_1 = V^2_1 = V^3_1 = V^3_{int..1}$$

$$\theta^1_1 = \theta^2_1 - 2\pi/3 = \theta^3_1 + 2\pi/3 = \theta_{int}$$

เมื่อ l คือ ตัวที่บอกถึง generator internal busbar

สัมประสิทธิ์ของสมการเมตริกซ์ (5.3.2) คือ

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

$$A^{pm}_{ik} = V^p_i V^m_k [G^{pm}_{ik} \sin \theta^{pm}_{ik} - B^{pm}_{ik} \cos \theta^{pm}_{ik}]$$

นอกจาก

$$A^{mm}_{kk} = -B^{mm}_{kk} (V^m_k)^2 - Q^m_k$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 [B_{jk}^m] &= [\Delta P_{gen,j} / \theta_k^m] \\
 &= \sum_{p=1}^3 V_{int,j} V_k^m [G_{jk}^{pm} \sin \theta_{jk}^{pm} - B_{jk}^{pm} \cos \theta_{jk}^{pm}]
 \end{aligned}$$

$$\begin{aligned}
 [E_{j1}^p] &= [\Delta P_{j1}^p / \theta_{int,j}] \\
 &= \sum_{p=1}^3 V_{int,j} V_{j1}^p [G_{j1}^{pm} \sin \theta_{j1}^{pm} - B_{j1}^{pm} \cos \theta_{j1}^{pm}]
 \end{aligned}$$

$$[F_{j1}] = [P_{gen,j} / \theta_{int,j}]$$

เมื่อ $[F_{j1}] = 0$ สำหรับทุกที่ ๆ $j \neq 1$ เพราะว่า generator ตัวที่ j ไม่ถูกต่อเข้ากับ internal busbar ของ generator ตัวที่ 1

$$\begin{aligned}
 [F_{11}] &= \sum_{p=1}^3 (-B_{11}^{pp} (V_{int,1})^2 - Q_{11}^p) \\
 &+ \sum_{m=1}^3 \sum_{p=1}^3 (V_{int,1})^2 [G_{11}^{pm} \sin \theta_{11}^{pm} - B_{11}^{pm} \cos \theta_{11}^{pm}]
 \end{aligned}$$

สัมประสิทธิ์ ของสมการเมตริกซ์ (5.4-3) คือ

$$-[k_{ik}^{pm}] = V_k^m [\Delta Q_{j1}^p / V_k^m]$$

เมื่อ

$$k_{ik}^{pm} = V_k^m V_{j1}^p [G_{ik}^{pm} \sin \theta_{ik}^{pm} - B_{ik}^{pm} \cos \theta_{ik}^{pm}]$$

นอกจาก

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ให้ $[L_{jk}^m] = V_k^m [L_{jk}^m]$ เมื่อ k คือ terminal busbar ของ generator ตัวที่ j และ $L_{jk}^m = 0$

$$-[P_{jj}^D] = V_{int,j} [\Delta Q_j^D / V_{int,j}]$$

$$= V_{int} \sum_{m=3}^3 V_j^D [G_{jj}^{Dm} \sin \theta_{jj}^{Dm} - B_{jj}^{Dm} \cos \theta_{jj}^{Dm}]$$

$$-[R_{jj}] = [\Delta V_{reg,j} / V_{int,j}]$$

= 0 สำหรับ j, l ซึ่งค่า voltage regulator specification ไม่ถูกรวมเข้ากับตัวแปร V_{int}

แม้ว่า สมการข้างต้นจะซับซ้อน แต่ความหมายจะคล้ายคลึงกับ single phase Jacobian elements โดยทั่วไป

การประมาณค่า Jacobian

การประมาณค่าจะคล้ายกับในกรณีของ single phase load flow โดยสามารถจะทำได้ดังนี้

1. ที่โหนดทั้งหมด (เฟสทั้งหมดของบัสทุกบัส) จะได้

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

2. ระหว่างโหนดที่ต่อกันและมีเฟสเดียวกันจะได้

$$\cos \theta_{ik}^{mm} \approx 1 \text{ คือ } \theta_{ik}^{mm} \text{ มีค่าน้อยมาก}$$

และ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ยิ่งกว่านั้นถ้า ค่า phase-angle unbalance ที่บัสบาร์ใดๆ มีค่าน้อยจะสามารถประมาณค่าสำหรับระบบ 3 เฟส ได้ดังนี้

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

4. จากผลของข้อ 2 และ 3 มุมระหว่างเฟสที่ต่างๆ กัน ที่ต่อระหว่างบัสบาร์ จะมีค่าประมาณ 120° คือ

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

หรือ

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

และ

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

ค่าเหล่านี้จะเปลี่ยนแปลงเป็นค่า phase shift 30° สำหรับการต่อหม้อแปลง 3 เฟส แบบ star-delta เปลี่ยนแปลงเร็วมาก และต่างไปจากมุม 120° เล็กน้อย ซึ่งทำให้การคอนเวอร์จ ของ phase unbalance ที่บัสบาร์ มีความช้ากว่า เมื่อเทียบกับ voltage magnitude และค่ามุม

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

$$\begin{aligned} \Delta P^p &= \left[\begin{array}{c} V^p_1 M^{pm}_{1k} V^m_k \\ \sum_{p=1}^3 V_{int..j} M^{pm}_{jk} V^m_k \end{array} \right] \left[\begin{array}{c} 0 \\ \sum_{m=1}^3 V^p_1 M^{pm}_{1j} V_{int..j} \\ \sum_{m=1}^3 \sum_{p=1}^3 V_{int..j} M^{pm}_{jk} V_{int..j} \end{array} \right] \left[\begin{array}{c} \Delta \theta^m_k \\ \Delta \theta_{int..j} \end{array} \right] \\ \Delta P_{gen..j} &= \left[\begin{array}{c} \sum_{p=1}^3 V_{int..j} M^{pm}_{jk} V^m_k \end{array} \right] \left[\begin{array}{c} 0 \\ \sum_{m=1}^3 \sum_{p=1}^3 V_{int..j} M^{pm}_{jk} V_{int..j} \end{array} \right] \left[\begin{array}{c} \Delta \theta^m_k \\ \Delta \theta_{int..j} \end{array} \right] \end{aligned} \quad (5.4.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{matrix} \Delta Q^p_i \\ \Delta P_{res.i} \end{matrix} = \begin{bmatrix} \begin{bmatrix} V^p_i & M^{pm}_{ik} & V^m_k \end{bmatrix} \\ \begin{bmatrix} V^m_k [L'] \end{bmatrix} \end{bmatrix} \begin{bmatrix} \sum_{m=1}^3 V^p_i M^{pm}_{ij} V_{int.j} \\ 0 \end{bmatrix} \begin{bmatrix} \Delta V^m_k / V^m_k \\ \Delta V_{int.j} / V_{int.j} \end{bmatrix}$$

(5.4.5)

เมื่อ

$$M^{pm}_{ik} = G^{pm}_{ik} \sin \theta^{pm}_{ik} - B^{pm}_{ik} \cos \theta^{pm}_{ik}$$

เมื่อ

$$G^{mm}_{kk} = 0$$

$$G^{pm}_{ik} = 0$$

$$\theta^{pm}_{ik} = \pm 120^\circ \quad \text{สำหรับ } p \neq m$$

ทุกเทอมในเมตริกซ์ [M] เป็นค่าคงที่ ซึ่งหาได้จาก system admittance matrix เมตริกซ์ [M] เหมือนกับเมตริกซ์ [-B] นอกจากเทอม off-diagonal ซึ่งต่อโหนดที่เฟสต่างกัน ซึ่งเทอมเหล่านี้จะถูกตัดแปลงโดยมุม 120° ที่เพิ่มเข้าไปยังเทอม $G^{pm}_{ik} \sin \theta^{pm}_{ik}$

การลดโครงสร้างของ Jacobian submatrix จะช่วยลดความซับซ้อนของโปรแกรมลงได้มาก จากที่ได้กล่าวมา Jacobian elements ในสมการ (4.4.4) และ (4.4.5) จะเป็นดังนี้

- ด้านซ้ายมือของสมการจะเปลี่ยนใหม่เป็น $[\Delta P^p_i / V^p_i]$, $[\Delta P_{res.i} / V_{int.i}]$, $[\Delta Q^p_i / V^p_i]$
- ในสมการ (4.4.4) ทางด้านขวามือจะเป็นค่าใน V เทอมให้เป็น 1 p.u.
- ในสมการ (4.4.5) ทางด้านขวามือ จะยกเลิกเทอม V ทางด้านขวา แต่เหลือค่าเวกเตอร์ไว้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการตัดแปลงจะได้สมการใหม่ดังนี้

$$\begin{bmatrix} \Delta P_1^D / V_1^D \\ \Delta P_{gen,i} / V_{int,i} \end{bmatrix} = \begin{bmatrix} M_{1k}^{Dm} & \sum_{m=1}^3 M_{1l}^{Dm} \\ \sum_{p=1}^3 M_{jk}^{Dm} & \sum_{p=1}^3 \sum_{m=1}^3 M_{jl}^{Dm} \end{bmatrix} \begin{bmatrix} \Delta \theta_k^D \\ \Delta \theta_{int,i} \end{bmatrix} \quad (5.4.6)$$

$$\begin{bmatrix} \Delta Q_1^D / V_1^D \\ \Delta V_{reg} \end{bmatrix} = \begin{bmatrix} M_{1k}^{Dm} & \sum_{m=1}^3 M_{1l}^{Dm} \\ [L_{jk}^m]' & 0 \end{bmatrix} \begin{bmatrix} \Delta V_k^m \\ \Delta V_{int,i} \end{bmatrix} \quad (5.4.7)$$

และ $[L_{jk}^m]'$ $[\Delta V_{reg} / V_k^m]$ ขณะที่ V_{reg} จะมีลักษณะเชิงเส้น ตาม terminal voltage $[L']$ เป็นเมตริกซ์ที่คงที่

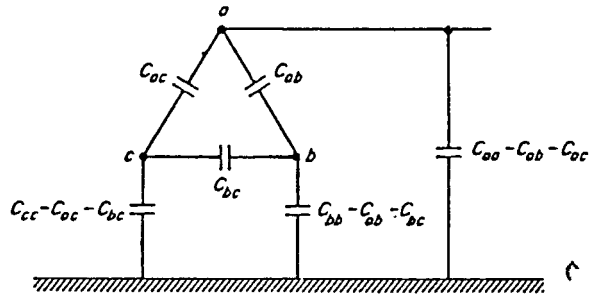
เนื่องจาก Jacobian matrix $[B']$ และ $[B'']$ ในสมการ (5.4.6) และ (5.4.7) จะประมาณค่าคงที่ได้ Zero diagonal element ในสมการ (5.4.7) เป็นผลมาจากการจัดเรียงสมการและตัวแปร

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 ซึ่งจะไม่ได้คิดค่า shunt capacitance matrix ในสูตรของ $[B']$ ซึ่งจะเป็นการเพิ่มอัตราความไวของกำลังจริง

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

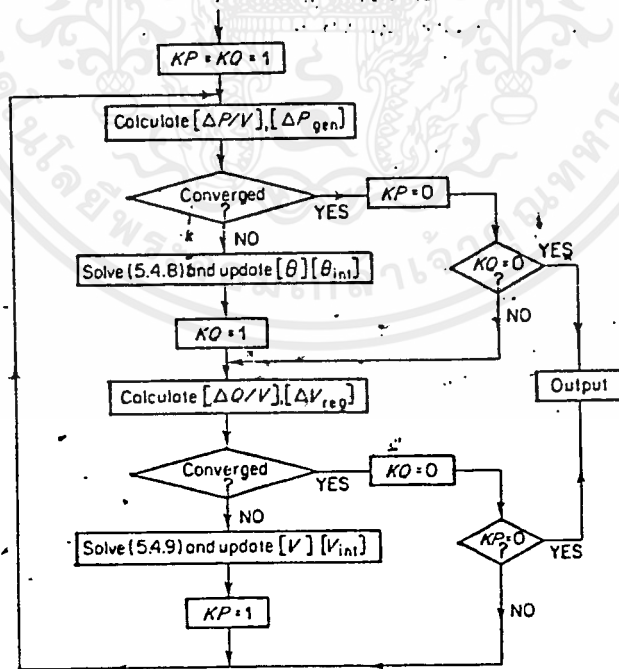


(i)

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

(ii)

รูป 5.2 Shunt capacitance matrix



รูป 5.3 Iteration sequence for three-phase a.c. load flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการตัดแปลงสมการดังที่กล่าวมาแล้ว จะได้ algorithmic 2 สมการสุดท้ายเป็น

$$\begin{bmatrix} \Delta P/V \\ \Delta P_{gen}/V_{int} \end{bmatrix} = \begin{bmatrix} B'_m \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta \theta_{int} \end{bmatrix} \quad (5.4.8)$$

$$\begin{bmatrix} \Delta Q/V \\ \Delta V_{reg} \end{bmatrix} = \begin{bmatrix} B''_m \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta V_{int} \end{bmatrix} \quad (5.4.9)$$

ค่าคงที่ Jacobian $[B'_m]$ และ $[B''_m]$ จะมีค่าคงที่ ความชันของพื้นที่ผิวขนาดหลายมิติ โดยใช้ฟังก์ชันทางด้านซ้ายมือของสมการ สมการ (5.4.8) และ (5.4.9) หาค่าตอบได้โดยการ iteration ตามลำดับ ดังใน รูป 5.3

Generator model และ fast decoupled algorithm

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

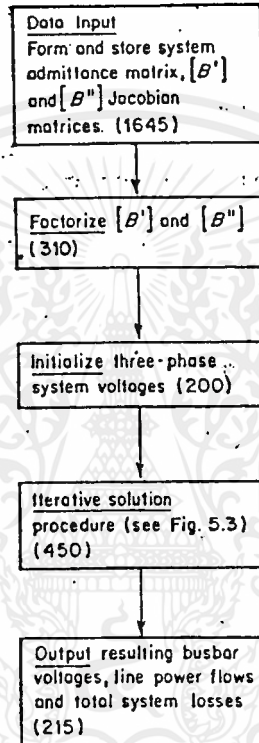
เมื่อใช้ค่าจริงของ positive sequence reactance ค่ามุมระหว่าง generator และขนาดของ excitation voltage มีค่าบวกภายใต้ภาวะ full load โดยมุมจะมีค่ามากกว่า 45 องศา และ excitation voltage จะมากกว่า 2 p.u. การคอนเวอร์จยากอาจเกิดขึ้นที่ slack generator และถ้าค่าแบบจำลองมีค่า Synchronous reactance สูง (1.5 p.u. บนฐานของพิกัดเครื่องจักร) และมากกว่า 70 เปอร์เซ็นต์ ของ full load ส่วน system generator เมื่อ real power ถูกกำหนด การคอนเวอร์จอาจช้า ภายใต้เงื่อนไขเดียวกัน

อัตราคอนเวอร์จที่แย่และข้อจำกัดของโหลดที่ติดต่อกับ generator อาจจะมีผลเล็กน้อย โดยตั้งค่า positive sequence reactance ของ generator ให้มีค่าต่ำ (ประมาณ 0.01 p.u. บนพิกัดของเครื่อง) โดยที่ไม่มีผลของความสูญเสียมาเกี่ยวข้องด้วย

โครงสร้างของโปรแกรมคอมพิวเตอร์ (Structure of the computer program)

ข้อมูลอินพุต (Input data)

ข้อมูลอินพุต จะใช้ข้อมูลดิบซึ่งได้แสดงในตัวอย่าง และจะนำข้อมูลดิบเหล่านี้ไปใช้ เพื่อหา 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 Program structure

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 โดยปราศจาก การประมาณค่าที่ต่ำกว่านี้ โดยการสมมติให้ค่า reactive power = 0 สำหรับ slack machine ค่า real power จะถูกประมาณให้เป็นผลต่าง ระหว่างค่าของโวลเตจทั้งหมด และค่า generator ทั้งหมด รวมทั้งค่าเปอร์เซ็นต์ของความสูญเสียที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 เฟส จะมีความซับซ้อนกว่าของ 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 conditions)

ระบบ 3 เฟสสมดุลย์ที่ปฏิบัติงานภายใต้ ภาวะ balanced load สามารถจะใช้แบบจำลอง ที่มีแต่ 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 3×3 และหม้อแปลงจะถูกแทนด้วย แบบจำลองที่มีพารามิเตอร์สมดุลย์กันทุกๆ 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 เฟส ภายใต้ ภาวะ มีแต่ค่า positive sequence network เท่านั้น ที่มีกำลังไหลผ่านและไม่มีการ coupling ระหว่าง sequence network ทั้ง negative และ zero sequence ในระบบ 3 เฟส ในภาวะสมดุลย์ จะไม่มีผลต่อการปฏิบัติงานและลักษณะของ algorithm

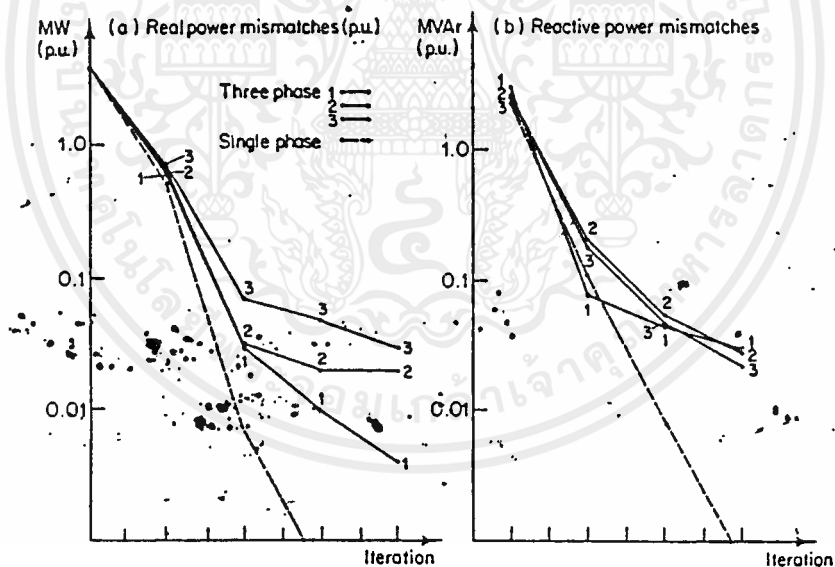
ตาราง 5.1 Convergence results

Case	Number of busbars	Single-phase load flow	Balanced three phase load flow		Typical three-phase unbalance
			$\lambda\lambda$	$\lambda\Delta$	
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:

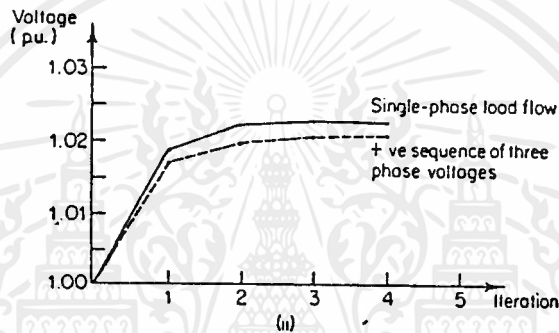
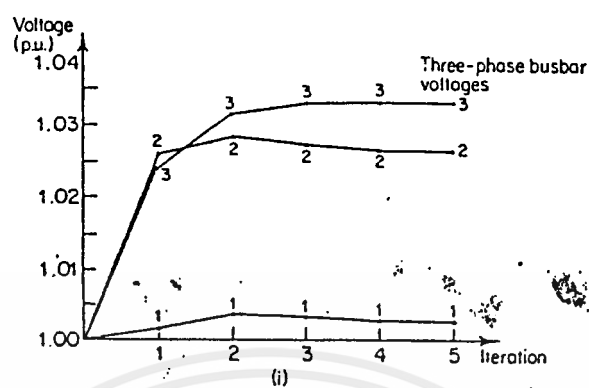
i—refers to the number of real power-angle update iterations.

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



รูป 5.5 Power convergence patterns for three phase and single phase load flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 5.6 Voltage convergence patterns for three phase and single phase load flow

(i) Three phase voltages

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

ลักษณะเมื่อเป็นระบบไม่สมดุล

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

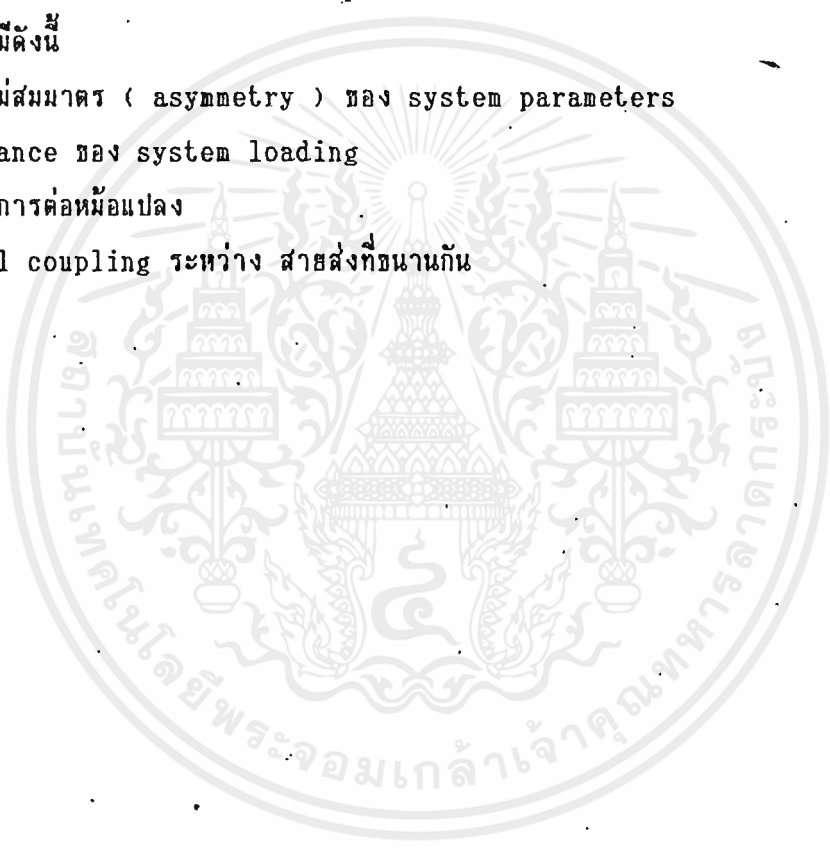
กราฟในรูป 5.5 แสดงถึงการเริ่มต้นการคอนเวอร์จ ของระบบ 3 เฟส ซึ่งมีค่าใกล้เคียงกับกรณี single phase load flow แต่อย่างไรก็ตาม ขณะที่เข้าใกล้ค่าตอบการคอนเวอร์จของระบบ 3 เฟส จะเริ่มช้ากว่า แม้ว่าโวลเตจและมุมที่ unbalance จะเริ่มตั้งแต่การ iteration ครั้งแรก ซึ่งมันจะมีผลต่อการคอนเวอร์จบ้าง จนกว่า positive sequence power flow จะเข้าใกล้การคอนเวอร์จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 จะมีลักษณะการคอนเวอร์จใกล้เคียงกับการแก้ของ 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



```

/* Load Flow Gauss-Seidel Iterative Method Using Y-Bus *****/
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define max_bus 5
#define base 100
#define CR 0xd
#define BK 0x8
typedef struct
{
    double real;
    double im;
} complex;
typedef struct
{
    int BusI, BusK;
    double Zre, Zim;
    double Lre, Lim;
} table44;
typedef struct
{
    int BusI;
    double Vre, Vim;
    double gMW, gMV;
    double lMW, lMV;
} table45;
complex sum_complex(complex, complex),
inverse_complex(complex),
multiply_complex(complex, complex),
divide_complex(complex, complex),
**get_matrix(int);
void free_matrix(complex **, int);
char *get_real(void);

```

```

main()
{
    FILE *fp;
    table44 data44;
    table45 data45;
    register i,j,k;
    static complex Z[max_bus+1][max_bus+1],
                  Y[max_bus+1][max_bus+1],
                  Ypq[max_bus+1][max_bus+1],
                  Yp[max_bus+1],
                  Y_bus[max_bus+1][max_bus+1],
                  V[max_bus+1],
                  power_gen[max_bus+1],
                  power_load[max_bus+1],
                  S[max_bus+1],
                  sum,
                  sum1,
                  sum2,
                  q,
                  z,
                  e,
                  x,
                  error;

    static int    bus,
                  count,
                  n;

    char          *s="",flnm[10],
                  ch;

    static double PV[max_bus],
                  ang,
                  tem,
                  fac,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

err;

clrscr();
printf("Enter file name of TABLE : ");
gets(s);
strcpy(flnm,s);
strcat(flnm, ".imp");
bus=0;
fp=fopen(flnm, "r");
while (fread(&data44, sizeof(table44), 1, fp) != 0)
{
    Z[data44.BusI][data44.BusK].real=data44.Zre;
    Z[data44.BusI][data44.BusK].im  =data44.Zim;
    Ypq[data44.BusI][data44.BusK].real=data44.Lre;
    Ypq[data44.BusI][data44.BusK].im  =data44.Lim;
}
fclose(fp);
strcpy(flnm,s);
strcat(flnm, ".gen");
fp=fopen(flnm, "r");
while (fread(&data45, sizeof(table45), 1, fp) != 0)
{
    V[data45.BusI].real=data45.Vre;
    V[data45.BusI].im  =data45.Vim;
    power_gen[data45.BusI].real=data45.gYW;
    power_gen[data45.BusI].im=data45.gMV;
    power_load[data45.BusI].real=data45.lMW;
    power_load[data45.BusI].im=data45.lMV;
    bus++; }
fclose(fp);
printf("%s%s%s%s\n",
        "bus code  ", "Impedance-real part  ",
        "Impedance-imaginary part  ", "Line-charging ");
window(1,3,80,25);
for(i=1;i<bus;i++)

```



```

cprintf("%2.3f",power_load[i].real);
gotoxy(67,wherey());
cprintf("%2.3f",power_load[i].im);
cprintf("\n\r");
    }
    putchar('\a');
    getch();
    window(1,1,80,25);
    clrscr();
    for(i=1;i<bus;i++)
    for(j=i+1;j<=bus;j++) {
if( Z[i][j].real == 0.0 && Z[i][j].im == 0.0 )
    continue;
Y[i][j] = inverse_complex(Z[i][j]);
    }
    for(k=1;k<=bus;k++)
    for(i=1;i<bus;i++)
    for(j=i+1;j<=bus;j++) {
        if( k==i || k==j )
            Yp[k].im += Ypq[i][j].im;
    }
    for(k=1;k<=bus;k++) {
    for(i=1;i<bus;i++)
    for(j=i+1;j<=bus;j++)
        if( k==i || k==j )
            Y_bus[k][k] = sum_complex(Y_bus[k][k],Y[i][j]);
    Y_bus[k][k].im += Yp[k].im;
    }
    for(i=1;i<bus;i++)
    for(j=i+1;j<=bus;j++) {
Y_bus[i][j].real = -1*Y[i][j].real;
Y_bus[i][j].im = -1*Y[i][j].im;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Y_bus[j][i].real = Y_bus[i][j].real;
Y_bus[j][i].im = Y_bus[i][j].im;
)
window(1,1,80,25);
clrscr();
printf("Has PV-Bus y/n? :");
while(ch= tolower(getche())=='y') {
printf("\nWhat is PV-Bus : ");
scanf("%d",&n);
printf("Fixed voltage at bus %d :",n);
scanf("%lf",&PV[n]);
printf("Another y/n? :");
)
clrscr();
printf("Acceleration factor = ? ");
scanf("%lf",&fac);
clrscr();
for(k=1;k<=bus;k++) {
S[k].real = (power_gen[k].real - power_load[k].real)/base;
if(k==n)
continue;
S[k].im = -1*(power_gen[k].im - power_load[k].im)/base;
}

for(i=1;i<80;printf("-"),i++);
printf("\nIteration\t\tBus 1\t\tBus 2\t\tBus 3\n");
for(i=1;i<80;printf("-"),i++);
e.real = V[n].real;
do {
for(i=2;i<=bus;i++) {
if(i==n) {
ang = atan2(V[n].im,V[n].real);
V[n].real = fabs(PV[n])*cos(ang);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

V[n].im = fabs(PV[n])*sin(ang);
tem = 0.0;
for(k=1;k<=bus;k++) {
    if(k==n)
        continue;
    tem += V[n].im*(V[k].real*Y_bus[n][k].real - \
V[k].im*Y_bus[n][k].im) - V[n].real*(V[k].im*\
Y_bus[n][k].real + V[k].real*Y_bus[n][k].im);
}
S[n].im = -1*(pow(V[n].real,2)*Y_bus[n][n].im + \
pow(V[n].im,2)*Y_bus[n][n].im) + tem;
S[n].im *= -1;
}
sum.real = 0.0; sum.im = 0.0;
for(j=1;j<=bus;j++) {
if (i==j)
    continue;
sum2 = multiply_complex(Y_bus[i][j],V[j]);
sum.real += sum2.real;
sum.im += sum2.im;
}
z = inverse_complex(Y_bus[i][i]);
q = divide_complex(S[i],V[i]);
q.real-=sum.real;
q.im -=sum.im;
sum1 = multiply_complex(q,z);
x.real = V[i].real;
x.im = V[i].im;
V[i].real += fac*(sum1.real - V[i].real);
V[i].im += fac*(sum1.im - V[i].im);
if(i==n) {
error.real = fabs(e.real - V[i].real);
error.im = fabs(e.im - V[i].im);
e.real = V[i].real;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

e.im = V[i].im;
}
else {
error.real = fabs(V[i].real - x.real);
error.im = fabs(V[i].im - x.im);
}
if(error.real > err)
    err = error.real;
if(error.im > err)
    err = error.im;
}
count++;
}
while( err > 0.001 );
}
char *get_real(void)
{
int i=0,j=0;
char ch,*s;
while( (ch = getch()) != CR ) {
if( isdigit(ch) || ch == '-' || ch == '.' || ch == BK ) {
if( ch == '-' && i != 0 )
    continue;
if( ch == '.' && j++ )
    continue;
if( ch == BK && i > 0 ) {
printf("\b \b");
if( s[i-1] == '.' )
j = 0;
i--;
continue;
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if ( ch == '\b' && i <= 0)
        continue;
    s[i++] = putchar(ch);
}
else continue;
}
s[i] = NULL;
if(!i) {
s = "0.0";
cprintf(s);
}

return s;
}
complex **get_matrix(int m)
{
register i;
complex **a;
a = calloc(m, sizeof(complex *));
--a;
for(i=1; i<=m; i++) {
a[i] = calloc(m, sizeof(complex));
--a[i];
}
return a;
}
void free_matrix(complex **a, int m)
{
register i;
for(i=1; i<=m; i++)
free(++a[i]);
free(++a);
}

```

```

complex sum_complex(complex a,complex b)
{
    complex ans;
    ans.real = a.real+b.real;
    ans.im   = a.im+b.im;
    return ans;
}

complex inverse_complex(complex c)
{
    double modulus;
    complex x;
    modulus = pow(c.real,2)+pow(c.im,2);
    x.real  = c.real/modulus;
    x.im    = -1*c.im/modulus;
    return x;
}

complex multiply_complex(complex a,complex b)
{
    complex ans;
    ans.real= (a.real*b.real)-(a.im*b.im);
    ans.im  = (a.real*b.im)+(a.im*b.real);
    return ans;
}

complex divide_complex(complex a,complex b)
{
    complex temp,ans;
    double modulus;
    modulus      = pow(b.real,2)+pow(b.im,2);
    temp.real    = b.real/modulus;
    temp.im      = -1*b.im/modulus;
    ans.real     = (a.real*temp.real)-(a.im*temp.im);
    ans.im       = (a.real*temp.im)+(a.im*temp.real);
    return ans;
}

```

ข้อมูลเริ่มต้นในการคำนวณโปรแกรม Gause seidel iterative method

Bus_Name	R_pq	X_pq	Img_Half_Line_Charging
1 2	0.08	0.24	0.0
1 3	0.02	0.06	0.0
2 3	0.06	0.18	0.0

Base_in_MVA 100

Bus	Voltage	G_MW	G_MVAR	L_MW	L_MVAR
1	1.05	0	0	0	0
2	1.03	20	0	50	20
-3	1	0	0	60	25

ผลการคำนวณของโปรแกรม Gause seidel

ITERATION = 7 = TOTAL

BUS	REAL VOLTAGE	IMAGINE VOLTAGE
1	1.05	0.00
2	1.028578	-0.0510117
3	1.02412	-0.034802

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*      LOAD FLOW CALCULATION BY NEWTON-RAPHSON      */
/*      RECTANGULAR COORDINATE METHOD                */
/*-----*/

#include <stdio.h>
#include <conio.h>
#include <math.h>

#define SP_MAX_BUS      6

/*----- External variable for Make_Ybus */
int      MAX_BUS ;
double  G[SP_MAX_BUS][SP_MAX_BUS] ;
double  B[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] ;
/*----- Make_BefPQ */
double  BASE ;
double  e[SP_MAX_BUS]      , f[SP_MAX_BUS]      ;
double  Psch[SP_MAX_BUS]  , Qsch[SP_MAX_BUS]  ;
double  GEN_MW[SP_MAX_BUS] , GEN_MVAR[SP_MAX_BUS] ;
double  L_MW[SP_MAX_BUS]  , L_MVAR[SP_MAX_BUS]  ;
/*----- Make_Delta_PQ */
double  P[SP_MAX_BUS]      , Q[SP_MAX_BUS]      ;
double  Delta_P[SP_MAX_BUS] , Delta_Q[SP_MAX_BUS] ;
double  Jacobian[SP_MAX_BUS*2][SP_MAX_BUS*2] ;
/*----- Make_cd */
double  c[SP_MAX_BUS] , d[SP_MAX_BUS] ;
/*----- Make_Jacobian */
double  Curl_P_e[SP_MAX_BUS][SP_MAX_BUS] ;
double  Curl_P_f[SP_MAX_BUS][SP_MAX_BUS] ;
double  Curl_Q_e[SP_MAX_BUS][SP_MAX_BUS] ;

```

```

double  Curl_Q_f[SP_MAX_BUS][SP_MAX_BUS] ;
int     DIMENSION ;
/*----- Make_Delta_ef */
double  Delta_e[SP_MAX_BUS] , Delta_f[SP_MAX_BUS] ;
/*----- Cal_PQ */
double  S_P[SP_MAX_BUS][SP_MAX_BUS] ;
double  S_Q[SP_MAX_BUS][SP_MAX_BUS] ;
double  P[SP_MAX_BUS] , Q[SP_MAX_BUS] ;
/*----- Make_table */
int     ITERATION ;
double  ERROR ;
/*-----*/
main()
{
void    Make_Ybus() , Make_BefPQ() , Make_Delta_PQ() ;
void    Make_cd() , Make_Jacobian() , Make_Delta_ef() ;
void    Make_Table() , Cal_PQ() ;
extern int    ITERATION ;
extern double ERROR ;
/*-----*/
clrscr() ;
Make_Ybus() ;
Make_BefPQ() ;
for(ITERATION = 1 ; ITERATION < 500 ; ITERATION++)
{
gotoxy(1,1) ; printf("ITERATION = [%d] ",ITERATION) ;
Make_Delta_PQ() ;
Make_cd() ;
Make_Jacobian() ;
Make_Delta_ef() ;
ERROR = ERROR * 10000 ;
ERROR = ERROR * ERROR ;
if(ERROR < 0.0001)

```

```

        Real_Zpq[P][Q] = 0 ; /* for next loop */
        Img_Zpq[P][Q] = 0 ;
    )
}

for(COUNT = 1 ; COUNT < NUM_BUS_NAME+1 ; COUNT++)
{
    fscanf(FP,"%d",&P) ;
    fscanf(FP,"%d",&Q) ;
    fscanf(FP,"%le",&Real_Zpq[P][Q]) ;
    Real_Zpq[Q][P] = Real_Zpq[P][Q] ;
    fscanf(FP,"%le", &Img_Zpq[P][Q]) ;
    Img_Zpq[Q][P] = Img_Zpq[P][Q] ;
    fscanf(FP,"%le", &Img_Ypq2[P][Q]);
    Img_Ypq2[Q][P] = Img_Ypq2[P][Q] ;
}
fclose(FP) ;
/*----- Cal Ypq */
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 ; /*ClearAll*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {   printf(" \t ERROR = [%f] ",ERROR) ;
    Cal_PQ() ;
    Make_Table() ;
    ITERATION = 600 ;
    printf("Program terminate press anykey  ") ;
    }
)
getch() ;
return(0) ;
) /*----- End of Main */

void Make_Ybus()
{
FILE *FP ;
auto int    NUM_BUS_NAME ;
auto double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS] ;
auto double   Img_Zpq[SP_MAX_BUS][SP_MAX_BUS] ;
extern double   Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS] ;
extern double   Real_Ypq[SP_MAX_BUS][SP_MAX_BUS] ;
extern double   Img_Ypq[SP_MAX_BUS][SP_MAX_BUS] ;
auto double   Img_Yp[SP_MAX_BUS] ;
auto double   DIVIDE , DATA_1 , DATA_2 , DATA_3 ;
auto int      P , Q , COUNT ;
auto char     NAME[22] ;
/*----- Input Zpq */
FP = fopen("IMP_5.DAT","r") ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%d",&NUM_BUS_NAME) ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%d",&MAX_BUS) ; MAX_BUS = MAX_BUS+1 ;
for(P = 1 ; P < 5 ; P++) fscanf(FP,"%s",NAME) ;
for(P = 1 ; P < MAX_BUS ; P++)
{   for(Q = 1 ; Q < MAX_BUS ; Q++)
    {   Img_Ypq2[P][Q] = 0 ; /* Clear all array */

```

```

FILE *FP ;
extern double BASE ;
extern double GEN_MW[SP_MAX_BUS] , GEN_MVAR[SP_MAX_BUS] ;
extern double L_MW[SP_MAX_BUS] , L_MVAR[SP_MAX_BUS] ;
auto int k ;
auto double TEMP_DATA ;
auto char NAME[22] ;
/*-----*/
FP = fopen("BefPQ_5.DAT","r") ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%le",&BASE) ;
for(k = 1 ; k < 7 ; k++) fscanf(FP,"%s",NAME) ;
for(k = 1 ; k < MAX_BUS ; k++)
{
    fscanf(FP,"%s",NAME) ;
    fscanf(FP,"%le",&e[k]) ;
    f[k] = 0 ;
    fscanf(FP,"%le",&Psch[k]) ; GEN_MW[k] = Psch[k] ;
    if(Psch[k] != 0) fscanf(FP,"%le",&Qsch[k]) ;
        else Qsch[k] = 0 ;
    GEN_MVAR[k] = Qsch[k] ;
    fscanf(FP,"%le",&TEMP_DATA) ; /* Scan Load MW */
    L_MW[k] = TEMP_DATA ;
    Psch[k] = ( Psch[k] - TEMP_DATA ) / BASE ;
    fscanf(FP,"%le",&TEMP_DATA) ; /* Scan Load MVA */
    L_MVAR[k] = TEMP_DATA ;
    Qsch[k] = ( Qsch[k] - TEMP_DATA ) / BASE ;
}
fclose(FP) ;
} /*----- End of Make_BefPQ */
void Make_Delta_PQ()
{
    auto int p , q , x , y , BUS ;
    /*-----*/

```

```

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 values are negative */
    }
}

} /*----- End of Make_Ybus */

void Make_BefPQ()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(BUS = 2 ; BUS < MAX_BUS ; BUS++) /* 1 Slack bus */
{
    P[BUS] = 0 ; /* Clear all array to protect */
    Q[BUS] = 0 ; /* computer hang on next loop */
}

for(p = 2 ; p < MAX_BUS ; p++) /* Bus 1 is slack bus */
{
    for(q = 1 ; q < MAX_BUS ; q++)
    {
        P[p] = P[p] + e[q] * ( e[q]*G[p][q] + f[q]*(-1*B[p][q]) ) ;
        P[p] = P[p] + f[p] * ( f[q]*G[p][q] - e[q]*(-1*B[p][q]) ) ;
        Delta_P[p] = Psch[p] - P[p] ;
        Q[p] = Q[p] + f[p] * ( e[q]*G[p][q] + f[q]*(-1*B[p][q]) ) ;
        Q[p] = Q[p] - e[p] * ( f[q]*G[p][q] - e[q]*(-1*B[p][q]) ) ;
        Delta_Q[p] = Qsch[p] - Q[p] ;
    }
}

for(p = 2 , x = 1 ; p < MAX_BUS ; p++ , x++) /* 1 slack */
{
    Jacobian[x][0] = Delta_P[p] ;
    Jacobian[x+(MAX_BUS-2)][0] = Delta_Q[p] ;
}

} /*----- End of Make_Delta_PQ */

void Make_cd()
{
    auto double DIVIDE ;
    auto int p ;
    /*-----*/
    for(p = 2 ; p < MAX_BUS ; p++) /*--- 1 Slack bus ---*/
    {
        DIVIDE = e[p]*e[p] ;
        DIVIDE = DIVIDE + f[p]*f[p] ;
        c[p] = ( P[p]*e[p] + Q[p]*f[p] )/DIVIDE ;
        d[p] = ( P[p]*f[p] - Q[p]*e[p] )/DIVIDE ;
    }
}

} /*----- End of Make_cd */

```

```

void Make_Jacobian()
{
    auto    int    p , q , x , y ;
    /*-----*/
    DIMENSION = (MAX_BUS - 2) * 2 ;
    for(p = 2 ; p < MAX_BUS ; p++) /*--- 1 SLACK BUS ---*/
    {
        for(q = 2 ; q < MAX_BUS ; q++)
        {
            if(p == q)
            {
                Curl_P_e[p][p] = e[p]*G[p][q] - f[p]*(-1*B[p][q]) + c[p] ;
                Curl_P_f[p][p] = e[p]*(-1*B[p][p]) + f[p]*G[p][p] + d[p] ;
                Curl_Q_e[p][p] = e[p]*(-1*B[p][p]) + f[p]*G[p][p] - d[p] ;
                Curl_Q_f[p][p] = -1*e[p]*G[p][p] + f[p]*(-1*B[p][p]) + c[p] ;
            }
            else
            {
                Curl_P_e[p][q] = e[p]*G[p][q] - f[p]*(-1*B[p][q]);
                Curl_P_f[p][q] = e[p]*(-1*B[p][q]) + f[p]*G[p][q];
                Curl_Q_e[p][q] = f[p]*G[p][q] + e[p]*(-1*B[p][q]);
                Curl_Q_f[p][q] = f[p]*(-1*B[p][q]) - e[p]*G[p][q];
            }
        }
    }
    for(p = 2 , y = 1 ; p < MAX_BUS ; p++ , y++)
    {
        for(q = 2 , x = 1 ; q < MAX_BUS ; q++ , x++)
        {
            Jacobian[y][x] = Curl_P_e[p][q] ;
        }
        for(q = 2 ; q < MAX_BUS ; q++ , x++)
        {
            Jacobian[y][x] = Curl_P_f[p][q] ;
        }
    }
    for(p = 2 ; p < MAX_BUS ; p++ , y++)

```

```

{   for(q = 2 , x = 1 ; q < MAX_BUS ; q++ , x++)
    {   Jacobian[y][x] = Curl_Q_e[p][q] ;
        }
    for(q = 2 ; q < MAX_BUS ; q++ , x++)
        {   Jacobian[y][x] = Curl_Q_f[p][q] ;
            }
}
) /*----- End of Make Jacobian */

```

```

void Make_Delta_ef()
{
auto   double   MULTI , Delta_ef[SP_MAX_BUS*2] ;
auto   int      i , y , x , k , yy ;
/*----- Delta PQ to right */
for(y = 1 ; y < DIMENSION+1 ; y++)
{   Jacobian[y][DIMENSION+1] = Jacobian[y][0] ;
}
/*----- cal Jacobian */
for (y = 1; y < DIMENSION+1 ; y++)
{   for (x = 1; x < DIMENSION+1 ; x++)
        {   if (x != y)
                {   MULTI = Jacobian[x][y]/Jacobian[y][y] ;
                    for (k = 1; k < DIMENSION+2 ; k++)
                        {   Jacobian[x][k] = Jacobian[x][k] -
                                MULTI*Jacobian[y][k] ;
                            }
                }
        }
}
}
for (y = 1; y < DIMENSION+1 ; y++)
{   Delta_ef[y] = Jacobian[y][DIMENSION+1]/Jacobian[y][y];
}
} /*----- Seperate Del e_f */

```

```

for(y = 1 , yy = 2 ; y < (DIMENSION/2) + 1 ; y++ , yy++)
{   Delta_e[yy] = Delta_ef[y] ;
}

for(y = (DIMENSION/2)+1 ,yy = 2; y < DIMENSION+1 ;y++,yy++)
{   Delta_f[yy] = Delta_ef[y] ;
}

/*----- Cal e_f */
for(i = 1 ; i < MAX_BUS ; i++)
{   e[i] = e[i] + Delta_e[i] ;
    f[i] = f[i] + Delta_f[i] ;
}

/*----- Cal ERROR */
ERROR = 0 ;
for(i = 2 ; i < MAX_BUS ; i++) /*--- 1 SLACK BUS ---*/
{   if(Delta_P[i]*Delta_P[i] > ERROR)
    ERROR = Delta_P[i]*Delta_P[i] ;
    if(Delta_Q[i]*Delta_Q[i] > ERROR) ERROR = Delta_Q[i]*Delta_Q[i] ;
}

) /*----- End of Make_Delta_ef */

void Cal_PQ() /* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
{
auto int    p , q ;
extern double S_P[SP_MAX_BUS][SP_MAX_BUS] ;
extern double S_Q[SP_MAX_BUS][SP_MAX_BUS] ;
extern double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS] ;
auto double REAL_1 , IMG_1 , TEMP_DATA ;
/*-----*/
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]) ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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] ;
S_P[p][q] = TEMP_DATA - (IMG_1*Img_Ypq[p][q]) ;
S_P[p][q] = BASE * S_P[p][q] ;
TEMP_DATA = REAL_1 * Img_Ypq[p][q] ;
TEMP_DATA = TEMP_DATA + (IMG_1*Real_Ypq[p][q]) ;
S_Q[p][q] = -1 * TEMP_DATA ;
TEMP_DATA = Img_Ypq2[p][q] * (e[p]*e[p] + f[p]*f[p]) ;
S_Q[p][q] = S_Q[p][q] - TEMP_DATA ;
S_Q[p][q] = BASE * S_Q[p][q] ;
    }
}
} /*----- End of Cal_PQ */
void Make_Table()
{
    auto int    p , q , X , Y ;
    /*----- Display Table */
    printf("\n\nBUS_____") ;
    printf("VOLTAGE_____TO_BUS_____MW_____MVAR_____ \n") ;
    for(p = 1 ; p < MAX_BUS ; p++)
    {   printf("[%d]  %f j%f",p,e[p],f[p]) ;
        for(q = 1 ; q < MAX_BUS ; q++)
        {   if( q != p && S_P[p][q] && S_Q[p][q] != 0 )
            {   Y = wherey() ;
                gotoxy(28,Y) ; printf("[%d]",q) ;
                gotoxy(35,Y) ; printf("%11.7f",S_P[p][q]) ;
                gotoxy(48,Y) ; printf("%11.7f \n",S_Q[p][q]) ;
            }
        }
    }
} /*----- end of make_Table */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเริ่มต้นในการคำนวณโปรแกรม Newton-Raphson Rectangular Coordinate Method

NUM_BUS_NAME 7

MAX_BUS 5

Bus_Name	R_pq	X_pq	Img_Half_Line_Charging
1 2	0.02	0.06	0.03
1 3	0.08	0.24	0.025
2 3	0.06	0.18	0.02
2 4	0.06	0.18	0.02
2 5	0.04	0.12	0.015
3 4	0.01	0.03	0.01
4 5	0.08	0.24	0.025

Base_in_MVA 100

Bus	Voltage	G_MW	G_MVAR	L_MW	L_MVAR
1	1.06	0	30	0	0
2	1	40	30	20	10
3	1	0		45	15
4	1	0		40	5
5	1	0		60	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการคำนวณของโปรแกรม Newton-Raphson Rectangular Coordinate Method

ITERATION = [3] = TOTAL

BUS	VOLTAGE	TO_BUS	MW	MVAR
[1]	1.060000 j0.000000	[2]	88.8638082	-8.5795106
		[3]	40.7230103	1.1582844
[2]	1.046182 -j0.051283	[1]	-87.4533583	6.1486825
		[3]	24.6943175	3.5464029
		[4]	27.9361166	2.9619632
		[5]	54.8229313	7.3429555
[3]	1.020283 -j0.089209	[1]	-39.5310532	-3.0138517
		[2]	-24.3428003	-6.7839736
		[4]	18.8738560	-5.2021685
[4]	1.019142 -j0.095066	[2]	-27.4947744	-5.9275647
		[3]	-18.8382512	3.2123595
		[5]	6.3330261	-2.2847874
[5]	1.012078 -j0.109058	[2]	-53.6976823	-7.1671907
		[4]	-6.3023154	-2.8327902

Program terminate press anykey

```

/*-----*/
/*      LOAD FLOW CALCULATION BY NEWTON-RAPHSON      */
/*      POLAR COORDINATES METHOD                      */
/*-----*/

#include <stdio.h>
#include <conio.h>
#include <math.h>

#define    SP_MAX_BUS    5

/*----- External variable for Make_Ybus */
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] ;
/*----- Make_BVPQ */
double  BASE ;
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 */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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] ;
/*-----*/
main()
{
void    Make_Ybus()      , Make_BVPQ() , Make_Delta_PQ() ;
void    Make_Curl_PQ() , Make_Jacobian() , Cal_Error() ;
void    Cal_PQ()      , Make_Table() ;
extern int    ITERATION ;
extern double ERROR ;
/*-----*/
clrscr() ;
Make_Ybus() ;
Make_BVPQ() ;
for(ITERATION = 1 ; ITERATION < 500 ; ITERATION++)
{
gotoxy(1,1) ; printf("ITERATION = [%d] ",ITERATION) ;
Make_Delta_PQ() ;
Make_Curl_PQ() ;
Make_Jacobian() ;
Cal_Error() ;
ERROR = ERROR * 10000 ;
ERROR = ERROR * ERROR ;
if( ERROR < 0.0001 )
{
printf("= TOTAL \n") ;
Cal_PQ() ;
Make_Table() ;
}
}
}

```

```

printf("Program terminate press anykey ");
ITERATION = 600 ;
}
)
getch() ;
return(0) ;
} /*----- End of Main */

```

```

void Make_Ybus()
{
FILE *FP ;
auto int NUM_BUS_NAME ;
auto double Real_Zpq[SP_MAX_BUS][SP_MAX_BUS] ;
auto double Img_Zpq[SP_MAX_BUS][SP_MAX_BUS] ;
extern double Img_Ypq2[SP_MAX_BUS][SP_MAX_BUS] ;
extern double Real_Ypq[SP_MAX_BUS][SP_MAX_BUS] ;
extern double Img_Ypq[SP_MAX_BUS][SP_MAX_BUS] ;
auto double Img_Yp[SP_MAX_BUS] ;
auto double G[SP_MAX_BUS][SP_MAX_BUS] ;
auto double B[SP_MAX_BUS][SP_MAX_BUS] ;
auto double DIVIDE , DATA_1 , DATA_2 , DATA_3 ;
auto int P , Q , COUNT ;
auto char NAME[22] ;
/*----- Input Zpq */
FP = fopen("IMP_6.DAT","r") ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%d",&NUM_BUS_NAME) ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%d",&MAX_BUS); MAX_BUS++;
for(P = 1 ; P < 5 ; P++) fscanf(FP,"%s",NAME) ;
for(P = 1 ; P < MAX_BUS ; P++)
{ for(Q = 1 ; Q < MAX_BUS ; Q++)
{ Img_Ypq2[P][Q] = 0 ; /* Clear all array */
Real_Zpq[P][Q] = 0 ; /* for next loop */

```

```

        Img_Zpq[P][Q] = 0 ;
    }
}
for(COUNT = 1 ; COUNT < NUM_BUS_NAME+1 ; COUNT++)
{
    fscanf(FP,"%d",&P) ;
    fscanf(FP,"%d",&Q) ;
    fscanf(FP,"%le",&Real_Zpq[P][Q]);
    Real_Zpq[Q][P] = Real_Zpq[P][Q];
    fscanf(FP,"%le",&Img_Zpq[P][Q]) ;
    Img_Zpq[Q][P] = Img_Zpq[P][Q] ;
    fscanf(FP,"%le",&Img_Ypq2[P][Q]);
    Img_Ypq2[Q][P] = Img_Ypq2[P][Q];
}
fclose(FP) ;
/*----- Cal Ypq */
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] ;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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_Ybus */

void Make_BVPQ()
{
FILE *FP ;
extern double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS] , BASE ;
extern double L_P[SP_MAX_BUS] , L_Q[SP_MAX_BUS] ;
auto int k ;
auto double TEMP_DATA ;
auto char NAME[22] ;
/*-----*/
FP = fopen("BVPQ_6.DAT","r") ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%le",&BASE) ;
for(k = 1 ; k < 7 ; k++) fscanf(FP,"%s",NAME) ;
for(k = 1 ; k < MAX_BUS ; k++)
{
fscanf(FP,"%d",&CHECK_LOAD_BUS[k]) ;
fscanf(FP,"%le",&MAGNI_V[k]) ;
ANGLE_V[k] = 0 ; /* BEGIN VALUE */
fscanf(FP,"%le",&Psch[k]) ; G_P[k] = Psch[k] ;
if(Psch[k] != 0) fscanf(FP,"%le",&Qsch[k]) ; else Qsch[k] = 0 ;
G_Q[k] = Qsch[k] ;
fscanf(FP,"%le",&TEMP_DATA) ; /* Scan Load MW */
L_P[k] = TEMP_DATA ;
}
}
}

```

```

    Psch[k] = ( Psch[k] - TEMP_DATA ) / BASE ;
    fscanf(FP,"%le",&TEMP_DATA) ; /* Scan Load MVA */
    L_Q[k] = TEMP_DATA ;
    Qsch[k] = ( Qsch[k] - TEMP_DATA ) / BASE ;
}
fclose(FP) ;
} /*----- End of Make_BVPQ */

void Make_Delta_PQ()
{
    auto double  DATA_P , DATA_Q , DATA_1 , DATA_2 , DATA_3 ;
    auto int      I      , K      ;
    /*-----*/
    for(I = 2 ; I < MAX_BUS ; I++) /*--- 1 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 ;
            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()
{

```

```

auto double DATA_P , DATA_Q , DATA_1 ;
auto double DATA_2 , DATA_3 , DATA_4 ;
auto int I , k ;
/*----- Cal Curl_PQ_Delta */
for(I = 2 ; I < MAX_BUS ; I++) /*--- 1 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 ; /* I not equal k */
    }
    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 = 2 ; I < MAX_BUS ; I++) /* 1 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 */
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA_3 = sin( DATA_2 ) ;
CURL_Q_V[I][k] = DATA_1 * DATA_3 ; /* I != k */
)
)
for(I = 2 ; I < MAX_BUS ; I++) /* 1 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 of Make_Curl_PQ */

void Make_Jacobian()
{
auto double JACOBIAN[SP_MAX_BUS*2][SP_MAX_BUS*2] ;
auto double TEMP_JACOB[SP_MAX_BUS*2] ;
auto int I , X ;
auto int k , J_X , J_Y ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

auto int    DIMENSION = 0 ; /* Number of equation */
auto double MULTI ;
/*----- Jacob P & Del */
for(I = 2 , J_Y = 1 ; I < MAX_BUS ; I++ , J_Y++) /*1 Slk */
{
    DIMENSION++ ;
    for(X = 2 , J_X = 1 ; X < MAX_BUS ; X++ , J_X++)
    {
        JACOBIAN[J_Y][J_X] = CURL_P_DELTA[I][X] ;
    }
    /* C_P_D for PV & PQ bus */
    for(X = 2 ; X < MAX_BUS ; X++)
    {
        if(CHECK_LOAD_BUS[X] < 0) /* C_P_V only PQ_BUS */
        {
            JACOBIAN[J_Y][J_X] = CURL_P_V[I][X] ;
            J_X++ ;
        }
    }
    JACOBIAN[J_Y][J_X] = DELTA_P[I] ; /*DEL_PforPV&PQbus*/
}
/*----- Jacob Q & Del */
for(I = 2 ; I < MAX_BUS ; I++) /*--- 1 SLACK BUS ---*/
{
    if(CHECK_LOAD_BUS[I] < 0) /*--- ONLY PQ_BUS ---*/
    {
        DIMENSION++ ;
        for(X = 2 , J_X = 1 ; X < MAX_BUS ; X++ , J_X++)
        {
            JACOBIAN[J_Y][J_X] = CURL_Q_DELTA[I][X] ;
        }
        /* C_Q_D FOR ONLY PQ_BUS */
        for(X = 2 ; X < MAX_BUS ; X++)
        {
            if(CHECK_LOAD_BUS[X] < 0)
            {
                JACOBIAN[J_Y][J_X] = CURL_Q_V[I][X] ;
                J_X++ ; /* C_Q_V for only PQ_BUS */
            }
        }
        JACOBIAN[J_Y][J_X] = DELTA_Q[I] ;
        J_Y++ ;
    }
}
}

```

if (i != slack bus)

if (x != slack bus)
on J_X

if (x != slack bus)

```

/*----- 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_JACOBI[J_Y] = JACOBIAN[J_Y][DIMENSION+1] /
        JACOBIAN[J_Y][J_Y] ;
}

/*----- Make Del V_& Ang */
for(k = 2 , J_Y = 1 ; k < DIMENSION+1 ; k++ , J_Y++) /* 1S */
{
    DELTA_ANG[k] = TEMP_JACOBI[J_Y] ;
    DELTA_ANG[k] = ( DELTA_ANG[k] * 180 ) / 3.141592654 ;
}

for(k = 2 ; k < DIMENSION+1 ; k++) /*- 1 SLACK BUS -*/
{
    if(CHECK_LOAD_BUS[k] < 0)
    {
        DELTA_VOL[k] = TEMP_JACOBI[J_Y] ;
        J_Y++ ; /* Value of J_Y begin from before loop */
    }
}

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

void Cal_Error()
{
    auto int    k ;

```

```

/*-----*/
for(k = 1 ; k < MAX_BUS ; k++ )
{
    MAGNI_V[k] = MAGNI_V[k] + DELTA_VOL[k] ;
    ANGLE_V[k] = ANGLE_V[k] + DELTA_ANG[k] ;
    ERROR = DELTA_VOL[k] * DELTA_ANG[k] ;
}
) /*----- End of Cal_Error */

void Cal_PQ() /* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
{
    auto    int    p , q , k , Y ;
    extern  double FLOW_P[SP_MAX_BUS][SP_MAX_BUS] ;
    extern  double FLOW_Q[SP_MAX_BUS][SP_MAX_BUS] ;
    extern  double G_P[SP_MAX_BUS] , G_Q[SP_MAX_BUS] ;
    auto    double e[SP_MAX_BUS] , f[SP_MAX_BUS] ;
    auto    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] ) ;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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 != 1)
  {
    REAL_1 = REAL_1 + FLOW_P[1][k] ;
    G_P[1] = REAL_1 ;
    IMG_1 = IMG_1 + FLOW_Q[1][k] ;
    G_Q[1] = 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 = 2 , TEMP_DATA = 0 ; k < MAX_BUS ; k++)
{
  if(CHECK_LOAD_BUS[k] > 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 */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Make_Table()
{
auto    int  p , q , k , Y , BUS ;
/*-----*/
printf("\nBUS_____VOLTAGE_MAGNITUDE_____PHASE_ANGLE\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") ;
}
for( k = 1 ; k < 60 ; k++ ) printf("-") ;
printf("\n\nBUS__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( 6,Y) ; printf("%5.2f",G_P[p]) ;
    gotoxy(13,Y) ; printf("%7.4f",G_Q[p]) ;
    gotoxy(23,Y) ; printf("%4.1f",L_P[p]) ;
    gotoxy(30,Y) ; printf("%4.1f",L_Q[p]) ;
    for(q = 1 ; q < MAX_BUS ; q++)
    {   if(q != p)
        {   gotoxy(37,Y) ; printf("%d",q) ;
            gotoxy(40,Y) ; printf("%9.5f",FLOW_P[p][q]) ;
            gotoxy(51,Y) ; printf("%9.5f",FLOW_Q[p][q]) ;
            Y++ ;
        }   printf("\n") ;
    }   printf("\n") ;
}
} /*----- End of Make_Table */

```

ข้อมูลเริ่มต้นในการคำนวณโปรแกรม Newton-Raphson Polar Coordinate Method

NUM_BUS_NAME 3

MAX_BUS 3

Bus_Name	R_pq	X_pq	Img_Half_Line_Charging
1 2	0.08	0.24	0.0
1 3	0.02	0.06	0.0
2 3	0.06	0.18	0.0

Base_in_MVA 100

Bus	Voltage	G_MW	G_MVAR	L_MW	L_MVAR
1	1.05	0	0	0	0
2	1.03	20	0	50	20
-3	1	0	0	60	25

ผลการคำนวณของโปรแกรม Newton-Raphson Polar Coordinate Method

ITERATION = [3] = TOTAL

BUS	VOLTAGE	MAGNITUDE	PHASE	ANGLE
[1]	1.050000		0.000000	
[2]	1.030000		-2.852037	
[3]	1.024752		-1.947019	

BUS	G[MW]	G[MVAR]	L[MW]	L[MVAR]	TO	BUS	MW	MVAR
[1]	91.37329	24.0691	0.00	0.0000	2		22.9719	1.650837
					3		68.4013	22.418264
[2]	20.00000	25.0507	50.00	20.0000	1		-22.5870	-0.496144
					3		-7.4129	5.546940
[3]	0.00000	0.0000	60.00	25.0000	1		-67.4614	-19.59849
					2		7.4614	-5.40150

Program terminate press anykey

```

/*-----*/
/*          FAST DECOUPLE METHOD          */
/*-----*/

#include <stdio.h>
#include <conio.h>
#include <math.h>

#define SP_MAX_BUS 4
#define MAX_Y      3 /*----- For      */
#define MAX_X      4 /*----- Sol_Matrx */

/*----- External variable for Read_Y_BUS */
int      MAX_BUS ;
double  Y_G[SP_MAX_BUS][SP_MAX_BUS], Y_B[SP_MAX_BUS][SP_MAX_BUS];
/*----- Read_BVPQ */
double  BASE_MVA ;
int      CHK_BUS[SP_MAX_BUS] ; /*----- 1 slk , 2 pv , -3 ld */
double  Psch[SP_MAX_BUS] , Qsch[SP_MAX_BUS] ;
double  MAGNI_V[SP_MAX_BUS] ;
double  ANGLE_V[SP_MAX_BUS] ;
double  GN_P[SP_MAX_BUS] , GN_Q[SP_MAX_BUS] ;
double  LD_P[SP_MAX_BUS] , LD_Q[SP_MAX_BUS] ;
/*----- Cal_Del_PQ */
double  DEL_PbyV[SP_MAX_BUS] ;
double  DEL_QbyV[SP_MAX_BUS] ;
/*----- Sol_Matrx */
double  ERROR , ERROR_OLD , ERROR_NEW ;
/*-----*/

main()
{
void      Read_Y_BUS() , Read_BVPQ() , Cal_Del_PQ() , Cal_HL() ;
void      Sol_Matrx() , Print_Vol() , Cal_FlowPQ() ;
auto      int.      LOOP ;

```

```

extern double ERROR ;
/*-----*/
clrscr() ; /*----- Clear screen first */
printf("FAST DECOUPLE METHOD") ;
Read_Y_BUS() ;
Read_BVPQ() ;
ERROR = 1.0 ;
for(LOOP = 1 ; ERROR > 0.01 ; LOOP++)
{
    gotoxy(1,3) ; printf("LOOP = [%d] \t ",LOOP) ;
    printf("ERROR = %f ",ERROR) ;
    Cal_Del_PQ() ;
    Sol_Matrx() ;
    Print_Vol() ;
    ERROR = ( (ERROR * 1000) * 1000 ) * 1000 ;
}
Cal_FlowPQ() ;
printf("Program termination press anykey ") ;
getch() ;
return(0) ;
} /*----- End of Main */

void Read_Y_BUS()
{
    FILE *FP ;
    extern int MAX_BUS ;
    auto int P , BUS_X , BUS_Y ;
    auto char NAME[20] ;
    auto double DAT ;
    /*----- Read G_B */
    FP = fopen("DP_Y_BUS.DAT","r") ;
    fscanf(FP,"%s",NAME) ; fscanf(FP,"%d",&MAX_BUS) ; MAX_BUS++ ;
    for(P = 1 ; P < 4 ; P++) fscanf(FP,"%s",NAME) ;
    for(BUS_Y = 1 ; BUS_Y < MAX_BUS ; BUS_Y++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{   fscanf(FP,"%s",NAME) ;
    for(BUS_X = 1 ; BUS_X < MAX_BUS ; BUS_X++)
    {   fscanf(FP,"%le",&DAT) ; Y_G[BUS_Y][BUS_X] = DAT ;
        fscanf(FP,"%le",&DAT) ; Y_B[BUS_Y][BUS_X] = DAT ;
    }
}

fclose(FP) ;

/*----- Display */
/*-----
clrscr() ;
printf("          Y_G ,   Y_B \n\n") ;
for(BUS_Y = 1 ; BUS_Y < MAX_BUS ; BUS_Y++)
{   for(BUS_X = 1 ; BUS_X < MAX_BUS ; BUS_X++)
    {   printf("[%d %d ,%8.4f,%8.4f]   ",BUS_Y,BUS_X,
            Y_G[BUS_Y][BUS_X] , Y_B[BUS_Y][BUS_X]) ;
    }
    printf("\n") ;
}

getch() ;
-----*/
} /*----- End of Read_Y_BUS */

void Read_BVPQ()
{
FILE   *FP ;

extern double  GN_P[SP_MAX_BUS] , GN_Q[SP_MAX_BUS] , BASE_MVA ;
extern double  LD_P[SP_MAX_BUS] , LD_Q[SP_MAX_BUS] ;
extern double  Psch[SP_MAX_BUS] , Qsch[SP_MAX_BUS] ;
auto   int     k ;
auto   char    NAME[20] ;

/*----- Read_BVPQ */

FP = fopen("DP_BVPQ.DAT","r") ;

fscanf(FP,"%s",NAME) ; fscanf(FP,"%le",&BASE_MVA) ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(k = 1 ; k < 7 ; k++) fscanf(FP,"%s",NAME) ;
for(k = 1 ; k < MAX_BUS ; k++)
{
    fscanf(FP,"%d",&CHK_BUS[k]) ;
    fscanf(FP,"%le",&MAGNI_V[k]) ;
    fscanf(FP,"%le",&ANGLE_V[k]) ;
    fscanf(FP,"%le",&Psch[k]) ; GN_P[k] = Psch[k] ;
    fscanf(FP,"%le",&Qsch[k]) ; GN_Q[k] = Qsch[k] ;
    fscanf(FP,"%le",&LD_P[k]) ;
    Psch[k] = ( Psch[k] - LD_P[k] ) / BASE_MVA ;
    fscanf(FP,"%le",&LD_Q[k]) ;
    Qsch[k] = ( Qsch[k] - LD_Q[k] ) / BASE_MVA ;
}
fclose(FP) ;
/*----- Display */
/*-----
clrscr() ;
printf("BASE_MVA = %f \n\n",BASE_MVA) ;
for(k = 1 ; k < MAX_BUS ; k++){
printf("CHK_BUS = %d \n",CHK_BUS[k]) ;
printf("MAGNI_V=%11.6f\tANGLE_V=%11.6f\n",MAGNI_V[k],ANGLE_V[k]);
printf("GN_P   = %11.6f \t GN_Q   = %11.6f \n",GN_P[k],GN_Q[k]) ;
printf("LD_P   = %11.6f \t LD_Q   = %11.6f \n",LD_P[k],LD_Q[k]) ;
printf("Psch   = %11.6f \t Qsch   = %11.6f \n",Psch[k],Qsch[k]) ;
printf("\n") ;
}
getch() ;
-----*/
} /*----- End of Read_BVPQ */

void Cal_Del_PQ()
{
extern double DEL_PbyV[SP_MAX_BUS] ;
extern double DEL_QbyV[SP_MAX_BUS] ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

auto    double    DAT_P , DAT_Q , DAT_COS , DAT_SIN ;
auto    double    SUM_P , SUM_Q , DAT_SUM , DAT_ANG ;
auto    int       P      , Q      ;

/*----- Cal_Del_PQbyV */
for(P = 1 ; P < MAX_BUS ; P++)
{
    SUM_P = 0 ; SUM_Q = 0 ;
    for(Q = 1 ; Q < MAX_BUS ; Q++)
    {
        DAT_P = 0 ;
        DAT_Q = 0 ;
        DAT_ANG = ANGLE_V[P] - ANGLE_V[Q] ;
        DAT_ANG = ( DAT_ANG * 3.141592654 ) / 180 ;
        DAT_COS = cos( DAT_ANG ) ;
        DAT_SIN = sin( DAT_ANG ) ;
        DAT_P = Y_G[P][Q] * DAT_COS ;
        DAT_P = DAT_P + ( Y_B[P][Q] * DAT_SIN ) ;
        DAT_P = DAT_P * MAGNI_V[Q] ;
        DAT_Q = Y_G[P][Q] * DAT_SIN ;
        DAT_Q = DAT_Q - ( Y_B[P][Q] * DAT_COS ) ;
        DAT_Q = DAT_Q * MAGNI_V[Q] ;
        SUM_P = SUM_P + DAT_P ;
        SUM_Q = SUM_Q + DAT_Q ;
    }
    SUM_P = SUM_P * MAGNI_V[P] ;
    SUM_Q = SUM_Q * MAGNI_V[P] ;
    SUM_P = Psch[P] - SUM_P ;
    DEL_PbyV[P] = SUM_P / MAGNI_V[P] ;
    SUM_Q = Qsch[P] - SUM_Q ;
    DEL_QbyV[P] = SUM_Q / MAGNI_V[P] ;
}

/*----- Display * /
clrscr() ;
for(P = 1 ; P < MAX_BUS ; P++)

```

```

{   printf("DEL_PbyV[%d] = %11.7f \t",P,DEL_PbyV[P]) ;
    printf("DEL_QbyV[%d] = %11.7f \n",P,DEL_QbyV[P]) ;
}

getch() ;

/*-----End of Display */

) /*----- End of Cal_Del_PQ */

void Sol_Matrx()
{
auto   int       P_EQATN , Q_EQATN ;
auto   double    DAT , MULTI , ANS[MAX_Y] , JCBN[MAX_Y][MAX_X] ;
auto   int       Y , X , Z ; BUS , B_X , B_Y ;
extern double    ERROR , ERROR_OLD , ERROR_NEW ;
/*-----*/
ERROR_OLD = MAGNI_V[3] ;
/*----- Cal_PQ_EQATN */
for(BUS = 1 , P_EQATN = 0 , Q_EQATN = 0 ; BUS < MAX_BUS ; BUS++)
{if(CHK_BUS[BUS] != 1) P_EQATN++ ; /*---- DelP use 2pv , -3ld */
if(CHK_BUS[BUS] < 0) Q_EQATN++ ; /*---- DelQ use      -3ld */
}

/*----- DelP , JCBN <= Y_B */
for(B_Y = 2 , Y = 1 ; B_Y < MAX_BUS ; B_Y++ , Y++)
{for(B_X = 2 , X = 1 ; B_X < MAX_BUS ; B_X++ , X++)
{DAT = Y_B[B_Y][B_X] ;
JCBN[Y][X] = -1 * DAT ;
}
}

/*----- JCBN <= DelPbyV */
for(Y = 1 , B_Y = 2 ; Y < 3 ; Y++ , B_Y++)
{JCBN[Y][3] = DEL_PbyV[B_Y] ;
BUS++ ;
}

/*----- Cal Del_AngleV */

```

```

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_V + Del_AngleV */
for(Y = 2 , Z = 1 ; Y < MAX_BUS ; Y++ , Z++)
{
  DAT = ANS[Z] ;
  ANGLE_V[Y] = ANGLE_V[Y] + DAT ;
}

/*----- Cal Del_MagniV */
/*----- Magni_V + Del_Magni */
DAT = Y_BC[3][3] ;
DAT = DEL_QbyV[3] / ( -1 * DAT ) ;
MAGNI_V[3] = MAGNI_V[3] + DAT ;
ERROR_NEW = MAGNI_V[3] ;
ERROR = ERROR_OLD - ERROR_NEW ;
if(ERROR < 0) ERROR = -1 * ERROR ;
/*----- Angle_240 = -120 */
for(BUS = 2 ; BUS < MAX_BUS ; BUS++)
{
  if( ( ANGLE_V[1] * ANGLE_V[BUS] ) < 0 )
  {
    if( ANGLE_V[1] < 0 ) ANGLE_V[BUS] = ANGLE_V[BUS] - 360 ;
    if( ANGLE_V[1] > 0 ) ANGLE_V[BUS] = 360 - ANGLE_V[BUS] ;
  }
}

```

```

    if( ANGLE_V[BUS] > 360 ) ANGLE_V[BUS] = ANGLE_V[BUS] - 360 ;
    if( ANGLE_V[BUS] < -360 ) ANGLE_V[BUS] = ANGLE_V[BUS] + 360 ;
}

} /*----- End of Sol_Matrx */

void Print_Vol()
{
extern double  MAGNI_V[SP_MAX_BUS] , ANGLE_V[SP_MAX_BUS] ;
auto  int      BUS ;
/*-----*/
printf("\n\n") ;
printf("BUS-----MAGNI_V-----ANGLE_V \n\n") ;
for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
{printf("[%d]",BUS) ;
printf(" %13.8f ",MAGNI_V[BUS]) ;
printf(" %13.8f ",ANGLE_V[BUS]) ;
printf("\n") ;
}
} /*----- End of Prn_V */

void Cal_FlowPQ() /*--Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
{
auto  int      p , q , k , X , Y ;
auto  double   FLOW_P[SP_MAX_BUS][SP_MAX_BUS] ;
auto  double   FLOW_Q[SP_MAX_BUS][SP_MAX_BUS] ;
extern double  GN_P[SP_MAX_BUS] , GN_Q[SP_MAX_BUS] ;
extern double  LD_P[SP_MAX_BUS] , LD_Q[SP_MAX_BUS] ;
extern double  Y_G[SP_MAX_BUS][SP_MAX_BUS] ;
extern double  Y_B[SP_MAX_BUS][SP_MAX_BUS] ;
extern double  BASE_MVA ;
extern int     CHK_BUS[SP_MAX_BUS] ;
auto  double   e[SP_MAX_BUS] , f[SP_MAX_BUS] ; /*----- V = e +jf */
auto  double   REL , IMG , DAT ;

```

```

/*----- Cal ef */
for(p = 1 ; p < MAX_BUS ; p++)
{
    DAT = ( ANGLE_V[p] * 3.141592654 ) /180 ;
    e[p] = MAGNI_V[p] * cos( DAT ) ;
    f[p] = MAGNI_V[p] * sin( DAT ) ;
}
/*----- Cal Flow PQ */
for(p = 1 ; p < MAX_BUS ; p++)
{
    for(q = 1 ; q < MAX_BUS ; q++)
    {
        if(q != p)
        {
            REL = e[p]*( e[p] - e[q] ) + f[p]*( f[p] - f[q] ) ;
            IMG = e[p]*( f[p] - f[q] ) - f[p]*( e[p] - e[q] ) ;
            /*----- Ep^(Ep-Eq) = REL + jIMG */
            DAT = REL * Y_G[p][q] ;
            DAT = DAT - ( IMG * Y_B[p][q] ) ;
            FLOW_P[p][q] = -1 * DAT ;
            FLOW_P[p][q] = BASE_MVA * FLOW_P[p][q] ;
            DAT = REL * Y_B[p][q] ;
            DAT = DAT + ( IMG * Y_G[p][q] ) ;
            FLOW_Q[p][q] = DAT ;
            FLOW_Q[p][q] = BASE_MVA * FLOW_Q[p][q] ;
        }
    }
}
/*----- Cal GN_PQ Slack bus */
for(k = 1 , REL = 0 , IMG = 0 ; k < MAX_BUS ; k++)
{
    if(k != 1)
    {
        REL = REL + FLOW_P[1][k] ;
        GN_P[1] = REL ;
        IMG = IMG + FLOW_Q[1][k] ;
        GN_Q[1] = IMG ;
    }
}

```

```

GN_P[p] = GN_P[p] + LD_P[p] ;
GN_Q[q] = GN_Q[q] + LD_Q[q] ;
/*----- Cal GN_Q PV bus */
for(k = 2 , DAT = 0 ; k < MAX_BUS ; k++)
{
    if(CHK_BUS[k] > 0)
    {
        for(q = 1 ; q < MAX_BUS ; q++)
        {
            if(q != k)
            {
                DAT = DAT + FLOW_Q[k][q] ;
            }
        }
        GN_Q[k] = DAT + LD_Q[k] ;
    }
}

/*----- Display Flow_PQ */
printf("\n");for(p=1;p<77;p++)printf("-");printf("\n\n") ;
printf("BUS_____G[MW]_____G[MVAR]_____L[MW]_____L[MVAR]") ;
printf("___TO_BUS_____MW_____MVAR_____\n\n ") ;
for(p = 1 ; p < MAX_BUS; p++)
{
    printf("[%d]",p) ; Y = wherey() ;
    gotoxy( 4,Y) ; printf(" %11.7f %11.7f  ",GN_P[p],GN_Q[p]) ;
    gotoxy(30,Y) ; printf(" %7.3f %7.3f  ",LD_P[p],LD_Q[p]) ;
    for(q = 1 ; q < MAX_BUS ; q++)
    {
        if(q != p)
        {
            gotoxy(51,Y) ; printf("%d",q) ;
            gotoxy(55,Y) ; printf("%11.7f",FLOW_P[p][q]) ;
            gotoxy(68,Y) ; printf("%11.7f",FLOW_Q[p][q]) ;
            Y++ ;
        }
        printf("\n") ;
    }
    printf("\n") ;
}

/*----- End of Cal_FlowPQ */
/*-----END OF FILE-----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเริ่มต้นในการคำนวณ

Base_in_MVA 100

Bus	Voltage	Angle	G_MW	G_MVAR	L_MW	L_MVAR
1	1.05	0	0	0	0	0
2	1.03	0	20	0	50	20
-3	1.00	0	0	0	60	25

MAX_BUS 3

	1	2	3
1	6.2500	-18.7500	-1.2500
2	-1.2500	3.7500	2.9167
3	-5.0000	15.0000	-1.6667

ผลการคำนวณของโปรแกรม Fats-decouple Method

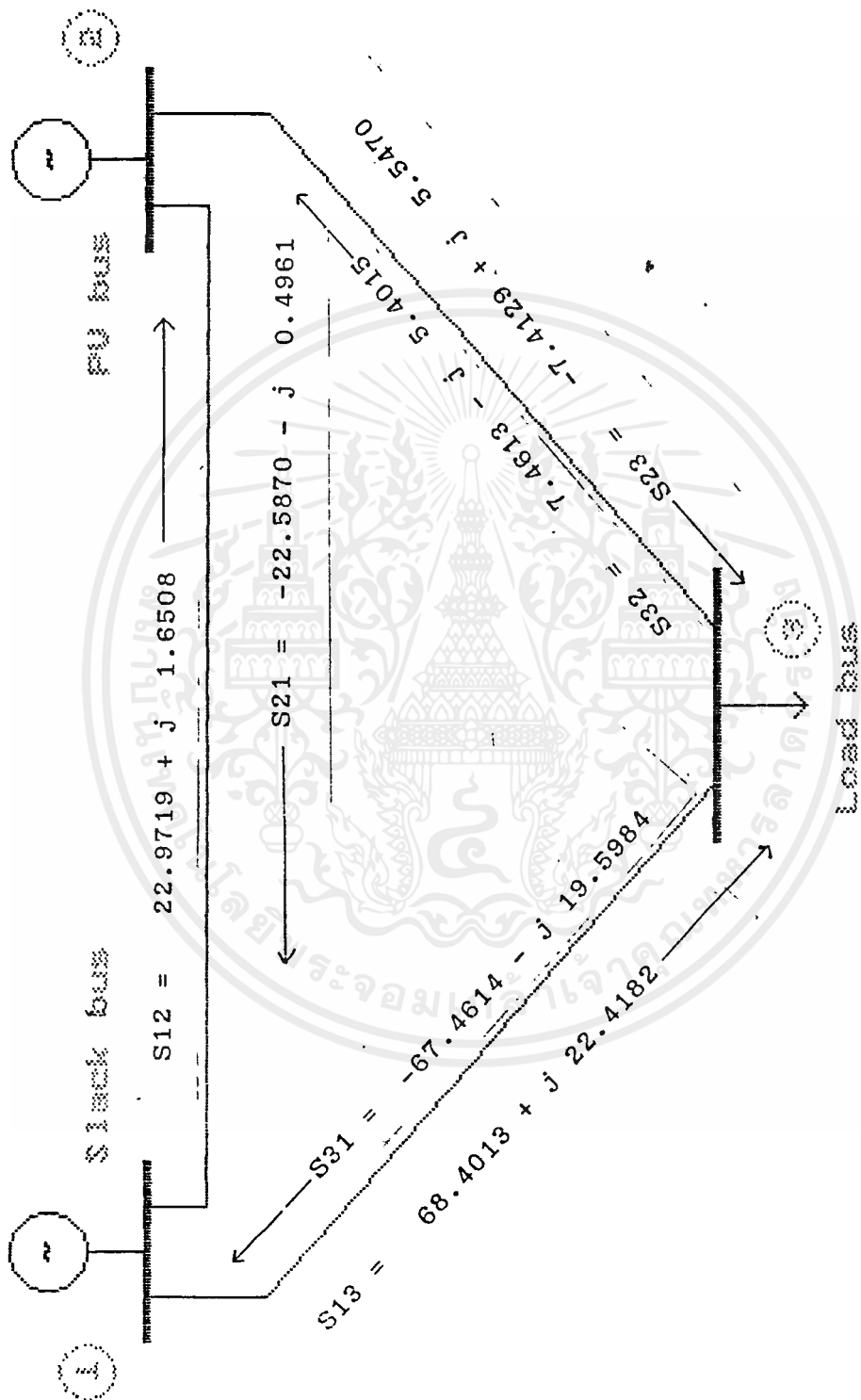
ITERATION = [822]

BUS	MAGNI_V	ANGLE_V
[1]	1.05000000	0.00000000
[2]	1.03000000	-2.85203714
[3]	1.02475239	-1.94701308

BUS	G[MW]	G[MVAR]	L[MW]	L[MVAR]	TO BUS	MW	MVAR
[1]	91.37283	24.069057	0.00	0.0000	2	22.9719	1.6508375
					3	68.4013	22.4182202
[2]	20.00000	25.05083	50.00	20.0000	1	-22.5870	-0.4961449
					3	-7.4129	5.5469820
[3]	0.00000	0.0000	60.00	25.0000	1	-67.4614	-19.5984560
					2	7.4613	-5.4015434

Program terminate press anykey

Load flow fast decouple method



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*          THREE-PHASE LOAD FLOW          */
/*          FAST DECOUPLE METHOD           */
/*-----*/
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define SP_MAX_BUS      4
#define SP_MAX_Y        7 /*----- For      */
#define SP_MAX_X        8 /*----- Sol_Matrix */
/*----- External Valibles for Read_Y_BUS */
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] ;
/*----- [phase_x][bus_x][phase_y][bus_y] */
/*----- Read_BVPQ */
int      BASE_MVA ;
int      CHK_BUS[SP_MAX_BUS] ; /*----- 1 slk , -2 pv , 3 ld */
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] ;
/*----- Cal_Del_PQbyV */
double  DEL_PbyV[4][SP_MAX_BUS] , DEL_QbyV[4][SP_MAX_BUS] ;
/*----- Cal_M */
double  M_[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
/*----- [phase_x][bus_x][phase_y][bus_y] */
/*----- Sol_Matrix */
double  ERROR , ERROR_OLD , ERROR_NEW ;
/*-----*/
main()
(

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void  Read_Y_BUS() , Read_BVPQ() , Cal_Del_PQbyV() , Cal_M() ;
void  Sol_Matrix() , Print_Vol() , Cal_Flow_PQ() ;
auto  int      LOOP ;
extern double  ERROR ;
/*-----*/
clrscr() ; /*----- Clear screen first */
printf("FAST DECOUPLE THREE-PHASE LOAD FLOW") ;
Read_Y_BUS() ;
Read_BVPQ() ;
ERROR = 0.1 ;
for(LOOP = 1 ; ERROR > 0.01 ; LOOP++)
{
    gotoxy(1,3) ; printf("LOOP = [%d] \t ",LOOP) ;
    printf("ERROR = %f      ",ERROR) ;
    Cal_Del_PQbyV() ;
    Cal_M() ;
    Sol_Matrix() ;
    Print_Vol() ;
    ERROR = ( (ERROR * 1000) * 1000 ) * 1000 ;
}
printf("Press anykey ") ;
getch() ;
Cal_Flow_PQ() ;
printf("Program terminate press anykey ") ;
getch() ;
return(0) ;
} /*----- End of Main */

void Read_Y_BUS()
{
    FILE      *FP ;
    auto  char  NAME[9] ;
    auto  int   Y , P_1 , B_2 , P_3 , B_4 ;
    auto  double  DAT ;

```

```

extern int      MAX_BUS ;
extern double   Y_G[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
extern double   Y_B[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
/*-----*/
FP = fopen("3P_Y_BUS.DAT","r") ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%d",&MAX_BUS) ;
MAX_BUS++ ; /*----- Array[4] = 0 to 3 */
for(B_2 = 1 ; B_2 < MAX_BUS ; B_2++)
{
    for(B_4 = 1 ; B_4 < MAX_BUS ; B_4++)
    {
        for(Y = 1 ; Y < 6 ; Y++) fscanf(FP,"%s",NAME) ;
        for(P_1 = 1 ; P_1 < 4 ; P_1++)
        {
            fscanf(FP,"%s",NAME) ;
            for(P_3 = 1 ; P_3 < 4 ; P_3++)
            {
                fscanf(FP,"%le",&DAT) ;
                Y_G[P_1][B_2][P_3][B_4] = DAT ;
                fscanf(FP,"%le",&DAT) ;
                Y_B[P_1][B_2][P_3][B_4] = DAT ;
            }
        }
    }
}
fclose(FP) ;
/*----- Display * /
clrscr() ;
for(B_2 = 1 ; B_2 < MAX_BUS ; B_2++)
{
    for(B_4 = 1 ; B_4 < MAX_BUS ; B_4++)
    {
        for(Y = 1 ; Y < 4 ; Y++)
            printf("-----PHASE_%d-----",Y) ;
        printf("\n") ;
        for(Y = 1 ; Y < 4 ; Y++)
            printf("-----G-----B-----") ;
        printf("\n") ;
    }
}

```

```

for(P_1 = 1 ; P_1 < 4 ; P_1++)
{
    for(P_3 = 1 ; P_3 < 4 ; P_3++)
        {
            printf("[%d%d%d%d,%8.4f,%8.4f] ",P_1,B_2,
                P_3,B_4,Y_G[P_1][B_2][P_3][B_4],
                Y_B[P_1][B_2][P_3][B_4]) ;
        }
    printf("\n") ;
}

getch() ; printf("\n") ;

}

} /*----- End of display */

} /*----- End of Read_Y_BUS */

void Read_BVPQ()
{
FILE      *FP ;
auto  int  Z , BUS , B , PHASE , P ;
auto  char  NAME[15] ;
auto  double  DAT ;
extern int  BASE_MVA , MAX_BUS , CHK_BUS[SP_MAX_BUS] ;
extern double  GN_P[4][SP_MAX_BUS] , GN_Q[4][SP_MAX_BUS] ;
extern double  LD_P[4][SP_MAX_BUS] , LD_Q[4][SP_MAX_BUS] ;
extern double  Psch[4][SP_MAX_BUS] , Qsch[4][SP_MAX_BUS] ;
extern double  MAGNI_VL[4][SP_MAX_BUS] , ANGLE_VL[4][SP_MAX_BUS] ;
/*-----*/
FP = fopen("3P_BVPQ.DAT","r") ;
fscanf(FP,"%s",NAME) ; fscanf(FP,"%d",&BASE_MVA) ;
/*----- Read Gen P Q */
for(Z = 1 ; Z < 11 ; Z++) fscanf(FP,"%s",NAME) ;
for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
{
    fscanf(FP,"%d",&CHK_BUS[BUS]) ;
    for(PHASE = 1 ; PHASE < 4 ; PHASE++)
        {
            fscanf(FP,"%le",&DAT) ; GN_PL[PHASE][BUS] = DAT ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        fscanf(FP,"%le",&DAT) ; GN_Q[PHASE][BUS] = DAT ;
    }
}
/*----- Read Load P Q */
for(Z = 1 ; Z < 11 ; Z++) fscanf(FP,"%s",NAME) ;
for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
{   fscanf(FP,"%s",NAME) ;
    for(PHASE = 1 ; PHASE < 4 ; PHASE++)
    {   fscanf(FP,"%le",&DAT) ; LD_P[PHASE][BUS] = DAT ;
        Psch[PHASE][BUS] = (GN_P[PHASE][BUS]-DAT)/BASE_MVA ;
        fscanf(FP,"%le",&DAT) ; LD_Q[PHASE][BUS] = DAT ;
        Qsch[PHASE][BUS] = (GN_Q[PHASE][BUS]-DAT)/BASE_MVA ;
    }
}
/*----- Read Voltage */
for(Z = 1 ; Z < 11 ; Z++) fscanf(FP,"%s",NAME) ;
for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
{   fscanf(FP,"%s",NAME) ;
    for(PHASE = 1 ; PHASE < 4 ; PHASE++)
    {   fscanf(FP,"%le",&DAT) ;
        MAGNI_V[PHASE][BUS] = DAT ;
        fscanf(FP,"%le",&DAT) ;
        ANGLE_V[PHASE][BUS] = DAT ;
    }
}
/*----- Display * /
clrscr() ;
printf("BASE_MVA = %d \n\n",BASE_MVA) ;
for(B = 1 ; B < MAX_BUS ; B++)
{   for(P = 1 ; P < 4 ; P++)
    {   printf("MAGNI_V[%d][%d]=[%.3f]",P,B,MAGNI_V[P][B]) ;
        if(P < 3) printf("  ") ; else printf("\n") ;
    }
}

```

```

}
for(P = 1 ; P < 4 ; P++)
{
    printf("ANGLE_V[%d][%d]=[%.3f]",P,B,ANGLE_V[P][B]) ;
    if(P < 3) printf("  ") ; else printf("\n") ;
}
for(P = 1 ; P < 4 ; P++)
{
    printf("  GN_P[%d][%d]=[%.3f]",P,B,GN_P[P][B]) ;
    if(P < 3) printf("  ") ; else printf("\n") ;
}
for(P = 1 ; P < 4 ; P++)
{
    printf("  GN_Q[%d][%d]=[%.3f]",P,B,GN_Q[P][B]) ;
    if(P < 3) printf("  ") ; else printf("\n") ;
}
for(P = 1 ; P < 4 ; P++)
{
    printf("  LD_P[%d][%d]=[%.3f]",P,B,LD_P[P][B]) ;
    if(P < 3) printf("  ") ; else printf("\n") ;
}
for(P = 1 ; P < 4 ; P++)
{
    printf("  LD_Q[%d][%d]=[%.3f]",P,B,LD_Q[P][B]) ;
    if(P < 3) printf("  ") ; else printf("\n") ;
}
for(P = 1 ; P < 4 ; P++)
{
    printf("  Psch[%d][%d]=[%.3f]",P,B,Psch[P][B]) ;
    if(P < 3) printf("  ") ; else printf("\n") ;
}
for(P = 1 ; P < 4 ; P++)
{
    printf("  Qsch[%d][%d]=[%.3f]",P,B,Qsch[P][B]) ;
    if(P < 3) printf("  ") ; else printf("\n") ;
}
printf("\n") ;
getch() ;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*----- End of display */
} /*----- End of Read_BVPQ */
void Cal_Del_PQbyV()
{
extern double DEL_PbyV[4][SP_MAX_BUS], DEL_QbyV[4][SP_MAX_BUS] ;
extern double MAGNI_VI[4][SP_MAX_BUS] , ANGLE_VI[4][SP_MAX_BUS] ;
extern int MAX_BUS ;
extern double Psch[4][SP_MAX_BUS] , Qsch[4][SP_MAX_BUS] ;
extern double Y_G[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
extern double Y_B[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
auto double DAT_P , DAT_Q , DAT_Y , SUM_P , SUM_Q ;
auto double DAT_COS , DAT_SIN , DAT_ANG ;
auto int BS_I , BS_K , PH_P , PH_M ;
/*-----*/
for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++)
{
for(PH_P = 1 ; PH_P < 4 ; PH_P++)
{
SUM_P = 0 ;
SUM_Q = 0 ;
for(BS_K = 1 ; BS_K < MAX_BUS ; BS_K++)
{
for(PH_M = 1 ; PH_M < 4 ; PH_M++)
{
DAT_ANG = ANGLE_VI[PH_P][BS_I] ;
DAT_ANG = DAT_ANG - ANGLE_VI[PH_M][BS_K] ;
DAT_ANG = ( DAT_ANG * 3.141592654 ) / 180 ;
DAT_SIN = sin( DAT_ANG ) ;
DAT_COS = cos( DAT_ANG ) ;
DAT_P = Y_G[PH_P][BS_I][PH_M][BS_K] ;
DAT_P = DAT_P * DAT_COS ;
DAT_Y = Y_B[PH_P][BS_I][PH_M][BS_K] ;
DAT_Y = DAT_Y * DAT_SIN ;
DAT_P = DAT_P + DAT_Y ;
DAT_Q = Y_G[PH_P][BS_I][PH_M][BS_K] ;
DAT_Q = DAT_Q * DAT_SIN ;
}
}
}
}
}
}

```

```

    DAT_Y = Y_B[PH_P][BS_I][PH_M][BS_K] ;
    DAT_Y = DAT_Y * DAT_COS ;
    DAT_Q = DAT_Q - DAT_Y ;
    DAT_P = DAT_P * MAGNI_V[PH_M][BS_K] ;
    DAT_Q = DAT_Q * MAGNI_V[PH_M][BS_K] ;
    SUM_P = SUM_P + DAT_P ;
    SUM_Q = SUM_Q + DAT_Q ;
}
}
SUM_P = SUM_P * MAGNI_V[PH_P][BS_I] ;
SUM_P = Psch[PH_P][BS_I] - SUM_P ;
DEL_PbyV[PH_P][BS_I] = SUM_P / MAGNI_V[PH_P][BS_I] ;
SUM_Q = SUM_Q * MAGNI_V[PH_P][BS_I] ;
SUM_Q = Qsch[PH_P][BS_I] - SUM_Q ;
DEL_QbyV[PH_P][BS_I] = SUM_Q / MAGNI_V[PH_P][BS_I] ;
}
}
/*----- Display */
clrscr() ;
for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++)
{
    for(PH_P = 1 ; PH_P < 4 ; PH_P++)
    {
        printf("DEL_PbyV[%d][%d]=[%.6f] ",PH_P,BS_I,
                DEL_PbyV[PH_P][BS_I]) ;
    }
    printf("\n") ;
}
printf("\n") ;
for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++)
{
    for(PH_P = 1 ; PH_P < 4 ; PH_P++)
    {
        printf("DEL_QbyV[%d][%d]=[%.6f] ",PH_P,BS_I,
                DEL_QbyV[PH_P][BS_I]) ;
    }
}

```

```

printf("\n") ;
}
printf("\n") ;
getch() ;
/*----- End of display */
} /*----- End of Cal_Del_PQbyV */
void Cal_M()
{
extern double M_[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
extern double Y_G[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
extern double Y_B[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
auto int PH_P , BS_I , PH_M , BS_K , BS_L , BS_J ;
auto int P_1 , B_2 , P_3 , B_4 ;
auto double DAT1 , DAT2 , DAT_ANG , DAT_SIN , DAT_COS ;
extern int MAX_BUS ;
/*----- Cal_M */
for(PH_P = 1 ; PH_P < 4 ; PH_P++)
{
for(BS_I = 1 ; BS_I < MAX_BUS ; BS_I++)
{
for(PH_M = 1 ; PH_M < 4 ; PH_M++)
{
for(BS_K = 1 ; BS_K < MAX_BUS ; BS_K++)
{
if(PH_P == 1 && PH_M == 2) DAT_ANG = 120 ;
if(PH_P == 1 && PH_M == 3) DAT_ANG = -120 ;
if(PH_P == 2 && PH_M == 1) DAT_ANG = -120 ;
if(PH_P == 2 && PH_M == 3) DAT_ANG = -240 ;
if(PH_P == 3 && PH_M == 1) DAT_ANG = 120 ;
if(PH_P == 3 && PH_M == 2) DAT_ANG = 240 ;
if(PH_P == PH_M) DAT_ANG = 0 ;
DAT_ANG = ( DAT_ANG * 3.141592654 ) / 180 ;
DAT_SIN = sin( DAT_ANG ) ;
DAT_COS = cos( DAT_ANG ) ;
DAT1 = Y_G[PH_P][BS_I][PH_M][BS_K] ;
DAT1 = DAT1 * DAT_SIN ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DAT2 = Y_B[PH_P][BS_I][PH_M][BS_K] ;
        DAT2 = DAT2 * DAT_COS ;
        DAT1 = DAT1 - DAT2 ;
        M_[PH_P][BS_I][PH_M][BS_K] = DAT1 ;
    }
}
}
}
/*----- Display */
clrscr() ;
for(B_2 = 1 ; B_2 < MAX_BUS ; B_2++)
{
    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++)
            {
                printf("M_[%d][%d][%d][%d][%8.4f] ", P_1,
                    B_2, P_3, B_4, M_[P_1][B_2][P_3][B_4]) ;
            }
            printf("\n") ;
        }
        getch() ; printf("\n") ;
    }
}
}
/*----- End of display */
} /*----- End of Cal_M */

void Sol_Matrix()
{
    auto double   DAT , MULTI , ANS[SP_MAX_Y] ;
    auto double   JCBN[SP_MAX_Y][SP_MAX_X] ;
    auto int      Y , Z , X , P1 , B2 , P3 , B4 , BX , BY ;
    auto int      P_EQATN , Q_EQATN ;
    extern double ERROR , ERROR_OLD , ERROR_NEW ;
    extern double M_[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
}

```

```

/*----- Clear all */
for(Y = 1 ; Y < SP_MAX_Y ; Y++)
{
  for(X = 1 ; X < SP_MAX_X ; X++)
    {
      JCBN[Y][X] = 0 ;
    }
}

/*----- DelP , JCBN <= M */
Y = 1 ;
for(B2 = 2 ; B2 < MAX_BUS ; B2++)
{
  for(P1 = 1 ; P1 < 4 ; P1++ , Y++)
    {
      for(B4 = 2 , X = 1 ; B4 < MAX_BUS ; B4++)
        {
          for(P3 = 1 ; P3 < 4 ; P3++ , X++)
            {
              DAT = M[P1][B2][P3][B4] ;
              JCBN[Y][X] = DAT ;
            }
        }
    }
}

/*----- JCBN <= DelPbyV */
Y = 1 ;
for(B2 = 2 ; B2 < MAX_BUS ; B2++)
{
  for(P1 = 1 ; P1 < 4 ; P1++ , Y++)
    {
      DAT = DEL_PbyV[P1][B2] ;
      JCBN[Y][7] = DAT ;
    }
}

/*----- DelP , SolMatrix */
P_EQATN = 6 ;
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(B2 = 2 ; B2 < MAX_BUS ; B2++)
  {
    for(P1 = 1 ; P1 < 4 ; P1++ , Y++)
      {
        DAT = ANS[Y] ;
        DAT = ANGLE_V[P1][B2] + DAT ;
        ANGLE_V[P1][B2] = DAT ;
      }
  }

/*----- DelQ , JCBN <= M */
for(P1 = 1 , Y = 1 ; P1 < 4 ; P1++ , Y++)
  {
    for(P3 = 1 , X = 1 ; P3 < 4 ; P3++ , X++)
      {
        DAT = M_[P1][3][P3][3] ;
        JCBN[Y][X] = DAT ;
      }
  }

}

/*----- JCBN <= DelQbyV */
for(Y = 1 , P1 = 1 ; Y < 4 ; Y++ , P1++)
  {
    DAT = DEL_QbyV[P1][3] ;
    JCBN[Y][4] = DAT ;
  }

}

/*----- DelQ , SolMatrix */
Q_EQATN = 3 ;

```

```

for(Y = 1 ; Y < Q_EQATN+1 ; Y++)
{
  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] ;
}

/*----- Magni = Magni + Del */
for(Y = 1 , P1 = 1 ; P1 < 4 ; Y++ , P1++)
{
  DAT = ANS[Y] ;
  DAT = DAT + MAGNI_V[P1][3] ;
  ERROR_OLD = MAGNI_V[P1][3] ;
  MAGNI_V[P1][3] = DAT ;
  ERROR_NEW = MAGNI_V[P1][3] ;
  ERROR = ERROR_OLD - ERROR_NEW ;
  if(ERROR < 0) ERROR = -1 * ERROR ;
}

/*----- Angle_240 = -120 */
for(B2 = 2 ; B2 < MAX_BUS ; B2++)
{
  for(P1 = 1 ; P1 < 4 ; P1++)
  {
    if((ANGLE_V[P1][1] * ANGLE_V[P1][B2]) < 0)
    {
      if(ANGLE_V[P1][1] < 0)
      {
        DAT = ANGLE_V[P1][B2] - 360 ;
        ANGLE_V[P1][B2] = DAT ;
      }
      if(ANGLE_V[P1][1] > 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {   DAT = 360 - ANGLE_VCP1][B2] ;
            ANGLE_VCP1][B2] = DAT ;
        }
    }
    if(ANGLE_VCP1][B2] > 360)
    {   DAT = ANGLE_VCP1][B2] - 360 ;
        ANGLE_VCP1][B2] = DAT ;
    }
    if(ANGLE_VCP1][B2] < -360)
    {   DAT = ANGLE_VCP1][B2] + 360 ;
        ANGLE_VCP1][B2] = DAT ;
    }
}
}
*/
} /*----- End of Sol_Matrix */
void Print_Vol()
{
extern double  MAGNI_V[4][SP_MAX_BUS] , ANGLE_V[4][SP_MAX_BUS] ;
auto  int      BUS , PHASE ;
/*-----*/
for(BUS = 1 ; BUS < MAX_BUS ; BUS++)
{   printf("\n\n") ;
    printf("BUS____PHASE____MAGNI_V____ANGLE_V\n") ;
    for(PHASE = 1 ; PHASE < 4 ; PHASE++)
    {   printf("[%d]      ",BUS) ; printf("[%d]",PHASE) ;
        printf("%13.8f",MAGNI_V[PHASE][BUS]) ;
        printf("      ") ;
        printf("%13.8f",ANGLE_V[PHASE][BUS]) ;
        printf("\n") ;
    }
}
} /*----- End of Print_Vol */

```

```

void Cal_Flow_PQ() /* Ppq - jQpq = Ep^(Ep-Eq)Ypq + Ep^EpYpq2 */
{
auto    int      p , q , k , X , Y , BUS , PH ;
auto    double   FLOW_PC[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
auto    double   FLOW_Q[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
extern  double   GN_PC[4][SP_MAX_BUS] , GN_Q[4][SP_MAX_BUS] ;
extern  double   LD_PC[4][SP_MAX_BUS] , LD_GC[4][SP_MAX_BUS] ;
extern  double   Y_GC[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
extern  double   Y_BC[4][SP_MAX_BUS][4][SP_MAX_BUS] ;
extern  int      BASE_MVA , CHK_BUS[SP_MAX_BUS] ;
auto    double   e[4][SP_MAX_BUS] , f[4][SP_MAX_BUS] ;
auto    double   REL , IMG , DAT ;

/*----- Cal ef */
for(PH = 1 ; PH < 4 ; PH++)
{
  for(p = 1 ; p < MAX_BUS ; p++)
  {
    DAT = ( ANGLE_V[PH][p] * 3.141592654 ) / 180 ;
    e[PH][p] = MAGNI_V[PH][p] * cos( DAT ) ;
    f[PH][p] = MAGNI_V[PH][p] * sin( DAT ) ;
  }
}

/*----- Cal Flow PQ */
for(PH = 1 ; PH < 4 ; PH++)
{
  for(p = 1 ; p < MAX_BUS ; p++)
  {
    for(q = 1 ; q < MAX_BUS ; q++)
    {
      if(q != p)
      {
REL = e[PH][p]*(e[PH][p]-e[PH][q])+f[PH][p]*(f[PH][p]-f[PH][q]);
IMG = e[PH][p]*(f[PH][p]-f[PH][q])-f[PH][p]*(e[PH][p]-e[PH][q]);
/*----- Ep^(Ep-Eq) = REL + jIMG */
          DAT = REL * Y_GC[PH][p][PH][q] ;
          DAT = DAT - ( IMG * Y_BC[PH][p][PH][q] ) ;
          DAT = -1 * DAT * BASE_MVA ;

```

```

        FLOW_P[PH][p][PH][q] = DAT ;
        DAT = REL * Y_B[PH][p][PH][q] ;
        DAT = DAT + ( IMG * Y_G[PH][p][PH][q] ) ;
        DAT =      DAT * BASE_MVA ;
        FLOW_Q[PH][p][PH][q] = DAT ;
    }
}
}
/*----- Cal GN_PQ Slack bus */
for(PH = 1 ; PH < 4 ; PH++)
{
    for(k = 1 , REL = 0 , IMG = 0 ; k < MAX_BUS ; k++)
    {
        if(k != 1)
        {
            REL = REL + FLOW_P[PH][1][PH][k] ;
            GN_P[PH][1] = REL ;
            IMG = IMG + FLOW_Q[PH][1][PH][k] ;
            GN_Q[PH][1] = IMG ;
        }
    }
    GN_P[PH][p] = GN_P[PH][p] + LD_P[PH][p] ;
    GN_Q[PH][q] = GN_Q[PH][q] + LD_Q[PH][q] ;
}
/*----- Cal GN_Q PV bus */
for(PH = 1 ; PH < 4 ; PH++)
{
    for(k = 2 , DAT = 0 ; k < MAX_BUS ; k++)
    {
        if(CHK_BUS[k] > 0)
        {
            for(q = 1 ; q < MAX_BUS ; q++)
            {
                if(q != k)
                {
                    DAT = DAT + FLOW_Q[PH][k][PH][q] ;
                }
            }
            GN_Q[PH][k] = DAT + LD_Q[PH][k] ;
        }
    }
}

```

```

    }
}
}
/*----- Display Flow_PQ */
for(PH = 1 ; PH < 4 ; PH++)
{
    clrscr() ;
    printf("PHASE = %d \n\n",PH) ;
    printf("BUS_____G[MW]_____G[MVAR]_____L[MW]___L[MVAR]") ;
    printf("___TO_BUS_____MW_____MVAR_____\n\n ") ;
    for(p = 1 ; p < MAX_BUS ; p++)
    {
        printf("[%d]",p) ; Y = wherey() ;
        gotoxy( 4,Y);printf(" %11.7f %11.7f ",
            GN_P[PH][p],GN_Q[PH][p]);
        gotoxy(30,Y);printf(" %7.3f %7.3f ",
            LD_P[PH][p],LD_Q[PH][p]);
        for(q = 1 ; q < MAX_BUS ; q++)
        {
            if(q != p)
            {
                gotoxy(51,Y) ; printf("%d",q) ;
                gotoxy(55,Y) ; printf("%11.7f",
                    FLOW_P[PH][p][PH][q]) ;
                gotoxy(68,Y) ; printf("%11.7f",
                    FLOW_Q[PH][p][PH][q]) ;
                Y++ ;
            }
        }
        printf("\n") ;
    }
    printf("\n") ;
}
getch() ;
}
} /*-----End of Cal_Flow_PQ */
/*-----END OF FILE-----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเริ่มต้นในการคำนวณ

BASE_MVA 100

BUS	GN_PHASE_A		GN_PHASE_B		GN_PHASE_C	
	P	Q	P	Q	P	Q
1	0	0	0	0	0	0
-2	20.00	00.00	20.00	00.00	20.00	00.00
3	0	0	0	0	0	0

BUS	LD_PHASE_A		LD_PHASE_B		LD_PHASE_C	
	P	Q	P	Q	P	Q
1	0	0	0	0	0	0
-2	50.00	20.00	50.00	20.00	50.00	20.00
3	60.00	25.00	60.00	25.00	60.00	25.00

BUS	VOLT_PHASE_A		VOLT_PHASE_B		VOLT_PHASE_C	
	MAGNI	ANG	MAGNI	ANG	MAGNI	ANG
1	1.05	0.0	1.05	-120	1.05	120
-2	1.03	0.0	1.03	-120	1.03	120
3	1.00	0.0	1.00	-120	1.00	120

MAX_BUS 3

		a		b		c	
1	a	6.25	-18.75	0.00	0.00	0.00	0.00
	b	0.00	0.00	6.25	-18.75	0.00	0.00
	c	0.00	0.00	0.00	0.00	6.25	-18.75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		2					
		a	b	c			
1	a	-1.25	3.75	0.00	0.00	0.00	0.00
	b	0.00	0.00	-1.25	3.75	0.00	0.00
	c	0.00	0.00	0.00	0.00	-1.25	3.75

		3					
		a	b	c			
1	a	-5.00	15.00	0.00	0.00	0.00	0.00
	b	0.00	0.00	-5.00	15.00	0.00	0.00
	c	0.00	0.00	0.00	0.00	-5.00	15.00

		1					
		a	b	c			
2	a	-1.25	3.75	0.00	0.00	0.00	0.00
	b	0.00	0.00	-1.25	3.75	0.00	0.00
	c	0.00	0.00	0.00	0.00	-1.25	3.75

		2					
		a	b	c			
2	a	2.9167	-8.75	0.00	0.00	0.00	0.00
	b	0.00	0.00	2.9167	-8.75	0.00	0.00
	c	0.00	0.00	0.00	0.00	2.9167	-8.75

		3					
		a	b	c			
2	a	-1.6667	5.00	0.00	0.00	0.00	0.00
	b	0.00	0.00	-1.6667	5.00	0.00	0.00
	c	0.00	0.00	0.00	0.00	-1.6667	5.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		1				
		a	b		c	
3	a	-5.00	15.00	0.00	0.00	0.00
	b	0.00	0.00	-5.00	15.00	0.00
	c	0.00	0.00	0.00	0.00	-5.00

		3				
		a	b		c	
2	a	-1.6667	5.00	0.00	0.00	0.00
	b	0.00	0.00	-1.6667	5.00	0.00
	c	0.00	0.00	0.00	0.00	-1.6667

		3				
		a	b		c	
3	a	6.6667	-20.00	0.00	0.00	0.00
	b	0.00	0.00	6.6667	-20.00	0.00
	c	0.00	0.00	0.00	0.00	6.6667

ผลการคำนวณ

FAST DECOUPLE THREE-PHASE LOAD FLOW

LOOP = [822] ERROR = 0.010126

BUS	PHASE	MAGN_V	ANGLE_V
[1]	[1]	1.05	0
[1]	[2]	1.05	-120
[1]	[3]	1.05	120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BUS	PHASE	MAGNI_V	ANGLE_V
[2]	[1]	1.03	-2.85203714
[2]	[2]	1.03	-122.85203714
[2]	[3]	1.03	117.14796286

BUS	PHASE	MAGNI_V	ANGLE_V
[3]	[1]	1.02475239	-1.94701908
[3]	[2]	1.02475239	-121.94701908
[3]	[3]	1.02475239	118.05298092

PHASE = 1

BUS	G[MW]	G[MVAR]	L[MW]	L[MVAR]	TO	BUS	MW	MVAR
[1]	91.37283	24.069057	0.00	0.0000	2		22.9719	1.6508375
					3		68.4013	22.4182202
[2]	20.00000	25.05083	50.00	20.0000	1		-22.5870	-0.4961449
					3		-7.4129	5.5469820
[3]	0.00000	0.0000	60.00	25.0000	1		-67.4614	-19.5984560
					2		7.4613	-5.4015434

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PHASE = 2

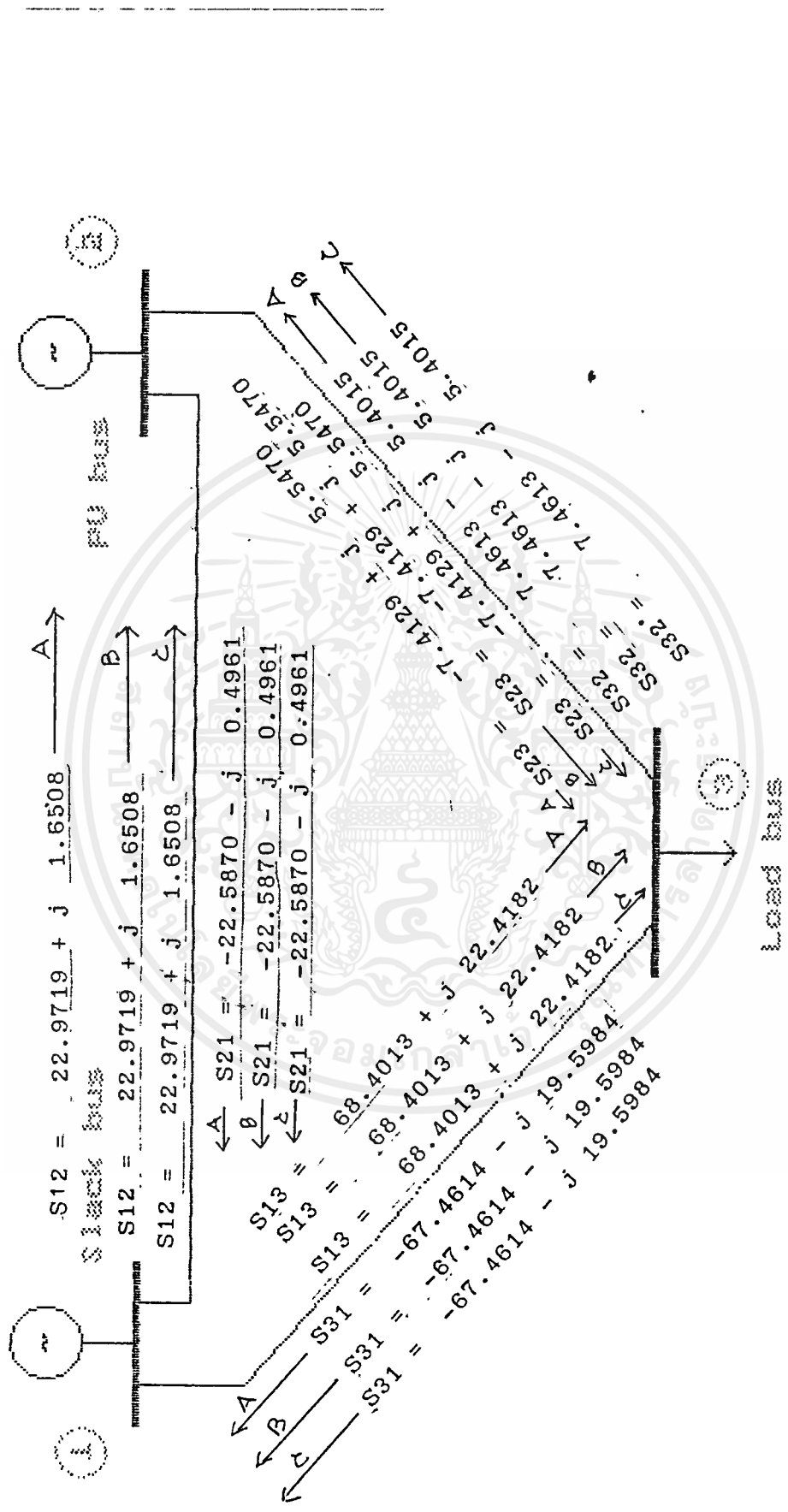
BUS	G[MW]	G[MVAR]	L[MW]	L[MVAR]	TO	BUS	MW	MVAR
[1]	91.37283	24.069057	0.00	0.0000	2		22.9719	1.6508375
					3		68.4013	22.4182202
[2]	20.00000	25.05083	50.00	20.0000	1		-22.5870	-0.4961449
					3		-7.4129	5.5469820
[3]	0.00000	0.0000	60.00	25.0000	1		-67.4614	-19.5984560
					2		7.4613	-5.4015434

PHASE = 3

BUS	G[MW]	G[MVAR]	L[MW]	L[MVAR]	TO	BUS	MW	MVAR
[1]	91.37283	24.069057	0.00	0.0000	2		22.9719	1.6508375
					3		68.4013	22.4182202
[2]	20.00000	25.05083	50.00	20.0000	1		-22.5870	-0.4961449
					3		-7.4129	5.5469820
[3]	0.00000	0.0000	60.00	25.0000	1		-67.4614	-19.5984560
					2		7.4613	-5.4015434

Program terminate press anykey

3-phase Load flow fast decouple method



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 ตารางแสดงข้อเปรียบเทียบระหว่างการคำนวณแบบเฟสเดียวกับแบบสามเฟส

แบบเฟสเดียว	แบบสามเฟส
1. ใช้เวลาในการคำนวณแต่ละรอบน้อย	1. ใช้เวลาในการคำนวณแต่ละรอบนาน
2. สมการในการคำนวณมีความยุ่งยากซับซ้อน	2. สมการในการคำนวณมีความยุ่งยากซับซ้อน
3. ระบบโครงข่ายจำเป็นต้องมีความสมดุลทั้งสามเฟส	3. ระบบโครงข่ายไม่จำเป็นต้องมีความสมดุลทั้งสามเฟส
4. คอมพิวเตอร์ใช้หน่วยความจำในการคำนวณน้อย	4. คอมพิวเตอร์ใช้หน่วยความจำในการคำนวณน้อยมาก
5. อัตราการลู่เข้าของคำตอบใช้เวลาน้อย	5. อัตราการลู่เข้าของคำตอบใช้เวลานาน
6. ไม่พิจารณาถึงค่าความเหนี่ยวนำเกี่ยวล้อที่เกิเกิดขึ้นระหว่างสาย	6. พิจารณาถึงค่าความเหนี่ยวนำเกี่ยวล้อที่เกิเกิดขึ้นระหว่างสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ชำนาญ ท่อเกียรติ , รัช.ดร. " ระบบไฟฟ้ากำลัง " มหาวิทยาลัยเกษตรศาสตร์, 2526
2. สมโภชน์ ประไพ วิทยาไพพันธ์ปริญญาวิศวกรรมศาสตรบัณฑิต "Computer Aid Design For Power System" สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังปีการศึกษา 2524
3. Glenn W. Stagg "Computer Method In Power System Analysis"
McGRW-Hill Book Company, 1987
4. Gonen, 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 & Son 1983
6. McGraw-Hill, New York, 1982
7. Stevenson, W.D., Jr., " Elements of Power System Analysis, " 4th ed.
8. Weedy, B.M., "Electric Power Systems" 3rd ed. Wiley, New York, 1979

THREE PHASE LOAD FLOW PROGRAM

Kenneth A. Birt,
Purdue University

James J. Graffy
Bonneville Power Administration

John D. McDonald
Bechtel

2 RT
1

A. H. El-Abiad
Purdue University

ABSTRACT

The paper describes a general three phase load flow program developed for the steady state analysis of any electric transmission network under all possible imbalance conditions such as untransposed EHV lines, single phase loading, single-pole switching, etc.

Input data preparation and output data interpretation are made easy for program users by using both phase and symmetrical components quantities; however, the load flow problem formulation and solution are all done in phase quantities (A, B, C). The method of solution is the Newton-Raphson method, using optimally ordered triangular factorization to take advantage of the sparsity of network equations.

The program has several new features to simulate real three phase operation of power systems and imbalance conditions. For example, the internal induced voltages of generators are balanced, while generator terminal voltages depend on internal machine impedances and the imbalance in machine currents. Also, the steady state performance of the voltage regulator is simulated to maintain some function of the machine terminal phase voltages constant, depending on the actual design of the voltage measurement. Another feature of the program is that it can handle all common transformer connections having two or three windings that are used in HV transmission networks.

INTRODUCTION

This paper describes a three phase load flow program and the models used for representing different components of the power system [1]. Of particular interest is the model for representing generators. Examples are given to show the degree of imbalance that can exist under normal operating conditions.

Previous work on this subject was done by Tarsi [2], but his program used the bus impedance matrix written with symmetrical components. This caused severe problems in the modeling of generators and transformers. The work presented here circumvents this problem by using the bus admittance matrix written in actual phase quantities. This has the added advantage of making data preparation easier.

For most purposes in the steady state analysis of power systems, the imbalance of a power system can be ignored and single phase analysis is adequate. However, some conditions such as very long untransposed lines, large single phase loads like induction furnaces and traction motors,

or single pole switching make more detailed analysis desirable. Although single pole switching is primarily a dynamic problem, it is desirable to know the imbalance that exists before operation of the breaker, and also the steady state condition that the system will approach while one phase is out.

Unbalanced currents can cause serious problems. Negative sequence currents may cause overheating of machinery. Ground currents (zero sequence) can cause improper operation of protective relaying. Zero sequence currents greatly increase the effect of inductive coupling between parallel transmission lines. This program could also be used to study the effect on system voltages of various phase arrangements on transmission lines.

METHOD OF SOLUTION

The three phase load flow is handled much like a single phase load flow - each single phase voltage, power, and current becomes a three element vector; and each single phase admittance is replaced with a three by three admittance matrix. In addition to this, some equipment is represented in more detail than would be used for a single phase study. The familiar Newton Raphson method [3,4,5,6] is used for solution of the system since it has been quite successful for single phase analysis, and it is most suited for handling the equations representing the generators.

To improve computational efficiency, sparsity-directed triangular factorization is used for the Jacobian matrix [7].

Transmission Lines

Transmission lines are represented with series and shunt admittances lumped to form a pi circuit as shown in Figure 1. The effects of ground resistance can be included in the admittances [9].

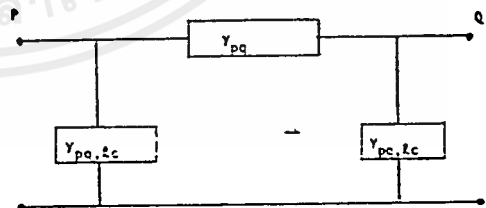


Fig. 1: Pi Model for 1 Circuit Transmission Line

Since the transmission line is a bilateral element, the matrices are symmetric. For the general case of an unbalanced line, all entries in the upper triangular part of the matrix will be different. Since this is true, the symmetrical component transformation will not diagonalize the matrix. Therefore there is no advantage

Paper F 75 423-4, recommended and approved by the IEEE Power System Engineering Committee of the IEEE Power Engineering Society for presentation at the IEEE PES Summer Meeting, San Francisco, Calif., July 20-25, 1975. Manuscript submitted January 31, 1975; made available for printing April 28, 1975.

in using symmetrical components, and actual phase quantities are used. This also simplified representation of unbalanced switching, e.g., single pole operations.

The bus admittance matrix for the line with respect to ground is shown in equation (1). Each entry of equation (1) is a three by three submatrix.

$$Y_{BUS} = \begin{matrix} & \begin{matrix} P & Q \end{matrix} \\ \begin{matrix} P \\ Q \end{matrix} & \begin{bmatrix} Y_{pq} + Y'_{pq,lc} & -Y_{pq} \\ (-Y_{pq})^T & Y_{pq} + Y'_{pq,lc} \end{bmatrix} \end{matrix} \quad (1)$$

For two parallel transmission lines, the inductive and capacitive coupling can be considered as shown in Figure 2. Each element in Figure 2 represents a three by three admittance matrix. The bus admittance matrix for these two lines is shown in (2).

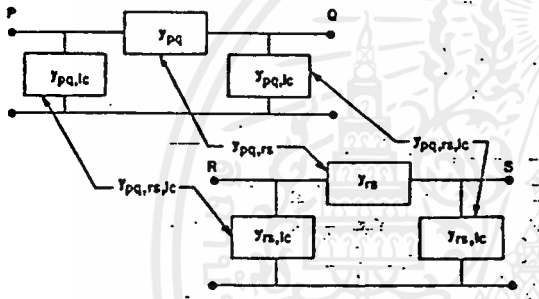


Fig. 2: Coupled Transmission Lines

$$Y_{BUS} = \begin{matrix} & \begin{matrix} P & Q & R & S \end{matrix} \\ \begin{matrix} P \\ Q \\ R \\ S \end{matrix} & \begin{bmatrix} Y_{pq} + Y'_{pq,lc} & -Y_{pq} & Y'_{pq,rs} + Y'_{rs,lc} & -Y_{pq,rs} \\ (-Y_{pq})^T & Y_{pq} + Y'_{pq,lc} & -Y_{pq,rs} & Y_{pq,rs} + Y'_{pq,rs,lc} \\ (Y'_{pq,rs} + Y'_{rs,lc})^T & (-Y_{pq,rs})^T & Y_{rs} + Y_{rs,lc} & -Y_{rs} \\ (-Y_{pq,rs})^T & (Y_{pq,rs} + Y'_{pq,rs,lc})^T & (-Y_{rs})^T & Y_{rs} + Y_{rs,lc} \end{bmatrix} \end{matrix} \quad (2)$$

In the event that a line changes construction between stations, a fictitious bus is introduced at that point. This fictitious bus is then included in the bus admittance matrix but eliminated during the iterative solution.

The bus admittance matrix for the whole system is constructed by combining the bus admittance matrices of individual lines as described in [8].

Transformers

Several common types of transformers can be represented. The program at present does not handle off-nominal taps or tap changing under load. The transformer representations are shown in Figure 6.

Load Buses

For three phase analysis, each bus will consist of three single phase buses. The program has provisions for reading either balanced or unbalanced loads, shunt reactors, or shunt capacitors.

The voltage magnitude and angle for each phase are to be found, therefore six equations are required for each bus. These equations are for the real and reactive bus powers for each phase.

Generator Buses

In single phase load flow studies, generators are commonly represented by injected powers with a specified bus voltage magnitude. If the effect of imbalance is to be studied, a more detailed model is needed. Therefore generators are represented by three, three phase buses connected by generator and transformer reactances as shown in Figure 3. Bus INT is an internal bus representing the internal induced voltages. All voltages at this bus are balanced, so only phase A is used as a variable, making the unknown quantities EINTA and δINTA. Total power (PA + PB + PC) generated by the machine is injected at this bus.

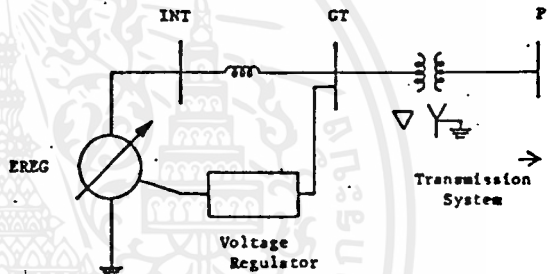


Fig. 3: Generator Model

Bus GT represents the generator terminals. The program assumes that there is no load at this bus (load is at bus P), therefore the injected real and reactive powers are zero. The six unknown quantities at this bus are the voltage magnitudes and angles.

Bus P is the high voltage bus connected to the transmission system. This is treated the same as any load bus.

The generator representation used requires that the steady state voltage regulator action be represented. It was assumed that the voltage regulator maintains the average of the three phase voltage magnitudes at EREG, however other representations can be introduced by making minor changes.

For the internal and generator terminal buses, the eight unknown quantities are three phase voltage magnitudes and three angles at bus GT and the phase A magnitude and angle at bus INT. Eight equations are written for these eight quantities: three reactive and three active powers at bus GT, voltage regulator setting and generator power setting as given in Table 1.

For the slack generator, the phase A voltage at bus GT is at a constant magnitude and a phase angle zero. This voltage was maintained because it was felt to be more realistic than regulating the internal voltage. Since there are two fewer unknowns for the slack generator, the equations for the total power and voltage regulator at this machine are deleted.

Table 1

Equations and Variables for Generator Representation

Equation	Variable
* $E_{REG} = 1/3(E_{GT}^A + E_{GT}^B + E_{GT}^C)$	E_{GT}^A
* $P_{MCH} = E_{INT} E_{GT} Y_G \cos(90 + \delta_{GT} - \delta_{INT})$	δ_{GT}^A
$Q_{GT}^A = f_{QA}(E, \delta)$	E_{INT}
$Q_{GT}^B = f_{QB}(E, \delta)$	E_{GT}^B
$Q_{GT}^C = f_{QC}(E, \delta)$	E_{GT}^C
$P_{GT}^A = f_{PA}(E, \delta)$	δ_{INT}^A
$P_{GT}^B = f_{PB}(E, \delta)$	δ_{GT}^B
$P_{GT}^C = f_{PC}(E, \delta)$	δ_{GT}^C

* For the slack generator, these equations are deleted and E_{GT}^A and δ_{GT}^A are held constant.

EXAMPLE

The load flow program was used to analyze the eight bus system shown in Figure 4. Two generators are included with their associated transformers. The seven buses numbered greater than 500 are fictitious buses to account for changes in line construction. Figure 5 and Table II show the dimensions of the different line constructions used.

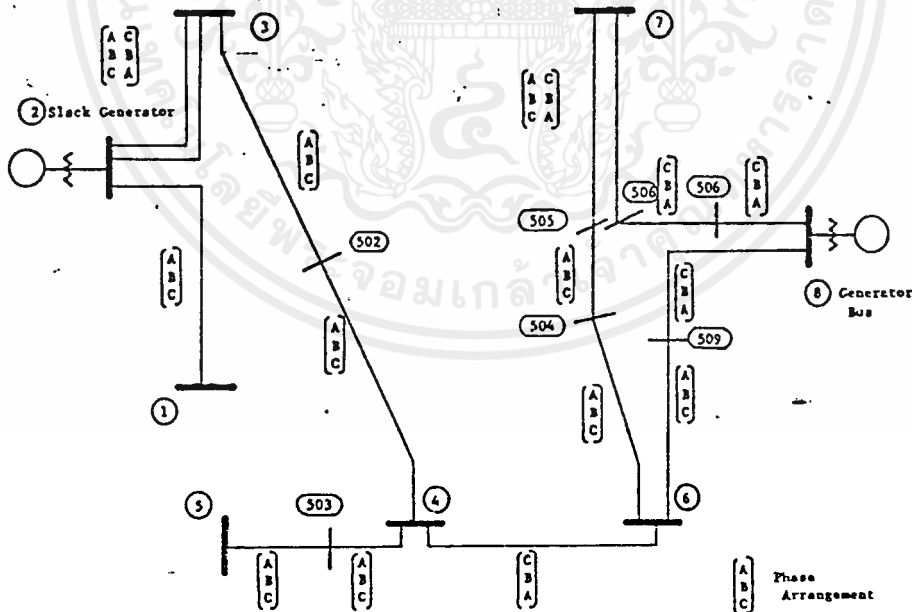


Fig. 4: Example System

The solution for this example is given in Table III and IV. The bus voltages in Table III show some imbalance. Negative sequence voltages are as high as 3.4% of the positive sequence voltage. Zero sequence voltages were as high as 2.8% of the positive sequence voltage. More important for relaying purposes, the currents show considerable imbalance. Negative sequence currents are as much as 10% of the positive sequence current and the zero sequence currents were as high as 30% of the positive sequence current as bus 504. This might be large enough to cause operational problems.

The currents induced in the generators are shown in Table V to have negative sequence components of 4% and 5% of the positive sequence current. This is of great interest because of its effect on heating of the rotor.

PROGRAM LIMITATIONS

Although the methods for three phase load flow analysis can include all the features found in single phase analysis, the present program does not include more advanced features.

Presently the program will handle a system of up to thirty buses with up to ten generators. A total of one hundred transmission system elements (lines, transformers, and coupled lines) can be included.

At present no facilities are provided for transformers with off-nominal taps, or phase shifting transformers.

CONCLUSION

A program has been developed for unbalanced three phase load flow analysis. The example given demonstrates that significant imbalance of line currents can exist during normal operation of a power system.

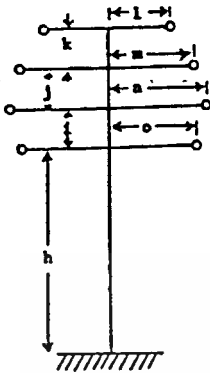


Fig. 5: Line Construction

Table II Transmission line dimensions in feet. See Figure 3 for explanation of letters.

TYPE	h	t	j	k	l	m	n	o	Conductor	Static
A	56.9	16.0	20.5	19.0	20.25	15.25	25.75	16.5	2156 MCM ACSR	7/16" steel
B	56.9	16.0	20.5	19.0	20.25	15.25	27.75	16.5	2156 MCM ACSR	7/16" steel
E	61.5	16.0	20.5	22.9	21.25	16.25	28.75	16.5	Two 954 MCM	7/16" steel
F	54.0	21.5	24.5	19.5	27.64	21.23	27.64	21.13	ACSR Bundled	159 MCM ACSR
G	61.5	16.0	20.5	22.9	21.25	16.25	28.75	17.5	18" Spacing	7/16" steel

* denotes double circuit line

ACKNOWLEDGEMENT

The authors would like to express their appreciation to the Purdue Electric Power Center for financial support of this work. The authors would also like to thank Messrs. H. L. Forgey, D. L. Hackett, D. V. Whiting, S. C. Bahl, and K. L. Burnett, of Consumers Power Company for their suggestions and assistance.

REFERENCES

- [1] K. A. Birt, J. J. Graffy, J. D. McDonald, and A. H. El-Abiad, "Three Phase Load Flow Program," TR-EE 75-19, Purdue University, June 1975.
- [2] D. C. Tarsi, "Analysis of Unbalanced Three-Phase Load Flows," M. S. Thesis, Purdue University, January 1967.
- [3] G. W. Stagg and A. H. El-Abiad, Computer Methods In Power System Analysis, New York: McGraw-Hill, 1968.
- [4] O. I. Elgerd, Electric Energy Systems Theory: An Introduction, New York: McGraw-Hill, 1971.
- [5] W. F. Tinney and C. E. Hart, "Power Flow Solutions by Newton's Method," IEEE Trans. Power Apparatus and Systems, Vol. PAS-86, pp. 1149-1460, November 1967.
- [6] J. E. VanNess and J. H. Griffin, "Elimination Methods for Load Flow Studies," Transactions AIEE, Vol. 80, June 1961.
- [7] W. F. Tinney and J. W. Walker, "Direct Solution of Sparse Network Equations by Optimally Order Triangular Factorization," PICA 1967, pp. 367-376.
- [8] B. J. Leon and P. A. Wintz, Basic Linear Networks for Electrical and Electronic Engineers, Holt, Rhinehart and Winston, 1970.
- [9] M. Chen and W. E. Dillon, "Power System Modeling," Proceedings of IEEE, Vol. 62, No. 7, July 1974.

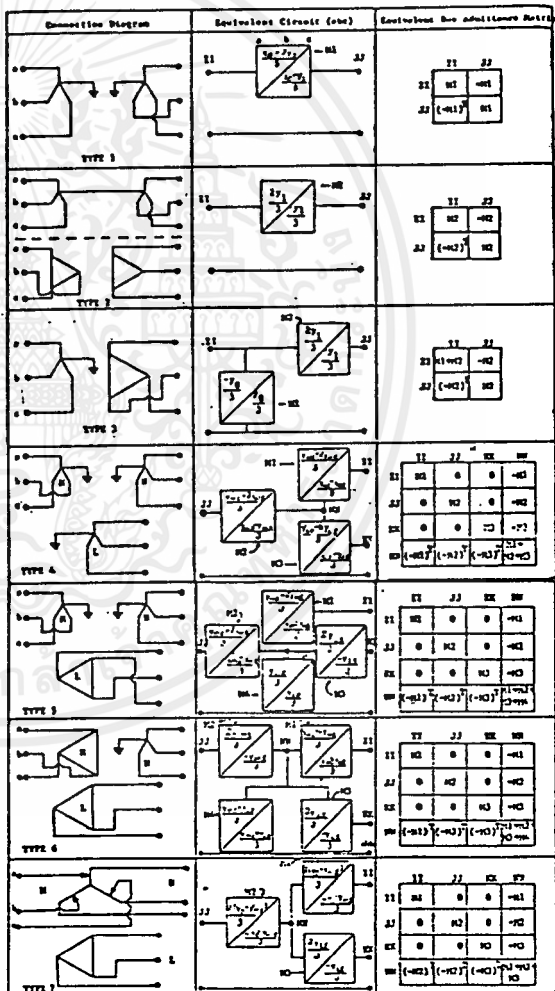


Fig. 6: Three Phase Representation of Transformer Types

Table III Bus voltages and loads

BUS	V(A,B,C)	V(0,1,2)	P(A,B,C)	Q(A,B,C)
1	1.02/-7.4	.01/-98.8	49.30	37.00
	1.06/-127.0	1.86/-7.7	49.30	37.00
	1.03/111.4	.03/142.9	49.30	37.00
2	-1.05/-5.8	.01/-88.4	0.00	0.00
	1.08/-125.6	1.83/-6.1	0.00	0.00
	1.05/113.1	.03/136.7	0.00	0.00
3	1.03/-10.6	.03/-102.2	140.60	19.30
	1.08/-129.3	1.82/-10.2	140.60	19.30
	1.04/109.0	.04/141.3	140.60	19.30
4	.99/-20.5	.05/-112.0	164.60	-3.30
	1.08/-136.8	1.79/-19.1	164.60	-3.30
	1.02/99.9	.06/152.2	164.60	-3.30
5	.99/-17.2	.05/-113.9	-54.00	21.00
	1.08/-134.4	1.77/-16.2	-54.00	21.00
	1.01/102.8	.06/143.4	-54.00	21.00
6	.99/-19.3	.05/-107.4	147.30	34.00
	1.07/-135.9	1.77/-18.0	147.30	34.00
	1.01/100.9	.06/149.7	147.30	34.00
7	.98/-18.7	.05/-115.3	95.30	32.00
	1.07/-133.8	1.76/-17.7	95.30	32.00
	1.00/101.3	.06/146.8	95.30	32.00
8	1.02/-11.4	.01/-98.1	7.66	-2.00
	1.06/-131.0	1.80/-11.7	7.66	-2.00
	1.03/107.2	.04/137.5	7.66	-2.00

Table IV Line currents and power flows

LINE NO.	LENGTH (MILES)	TYPE	LINE FLOWS			
			I(A,B,C)	I(0,1,2)	P(A,B,C)	Q(A,B,C)
2-3	40.	A	3.21/-11.4	.20/-7.4	111.36	10.92
			3.18/-123.6	5.57/-8.9	111.57	-3.78
			3.35/108.7	.34/-139.9	117.17	8.96
2-3	40.	A	3.25/-10.2	.12/.2	112.90	8.74
			3.11/-127.0	5.55/-9.1	111.91	2.83
			3.26/110.1	.14/-117.3	114.22	5.95
7-505	17.2	G	.50/18.4	.34/-47.4	12.93	-9.77
			.59/-67.5	.75/43.9	7.77	-15.51
			.30/-171.3	.04/-2	.47	-10.04
7-506	17.2	G	3.28/153.1	.32/-174.4	-106.13	-15.23
			2.84/35.1	5.36/153.0	-99.52	-15.82
			3.17/-89.1	.17/81.2	-104.34	-19.23
1-2	45.	B	1.91/135.7	.06/164.8	-49.30	-37.00
			1.75/16.1	3.09/135.4	-49.30	-37.00
			1.79/-105.5	.02/48.8	-49.30	-37.00
3-502	82.1	E	2.41/-10.7	.16/-4.9	82.66	.25
			2.33/-120.8	4.21/-7.4	83.05	-12.41
			2.58/109.8	.33/-138.8	88.97	-1.12
502-4	31.	F	2.47/-23.8	.18/-7.2	62.53	8.00
			2.30/-136.4	4.23/1.4	63.23	1.92
			2.56/96.4	.23/39.3	66.93	8.68
4-503	51.5	F	1.69/165.9	.11/-162.2	-55.58	6.29
			1.49/43.2	2.72/164.5	-53.75	.00
			1.53/-75.9	.11/171.8	-51.96	3.79
5-503	14.	B	1.76/4.1	.10/25.8	54.01	-21.03
			1.61/-113.2	2.94/5.0	53.98	-21.00
			1.73/124.1	.08/-75.4	54.02	-20.98
4-6	44.	F	.85/-178.9	.05/84.5	-26.03	10.33
			.86/69.6	1.50/-173.6	-27.87	13.80
			.89/-51.8	.09/76.9	-26.73	14.39
504-6	25.	F	.46/24.1	.18/28.9	11.14	-10.41
			.26/-73.7	.58/27.1	4.30	-8.17
			.30/135.4	.06/-24.3	8.44	-5.77
504-505	5.5	B	.46/-155.9	.18/-151.1	-11.14	10.41
			.26/106.3	.58/-152.9	-4.30	8.17
			.30/-44.6	.06/155.7	-8.44	5.77
506-507	1.3	B	3.24/155.7	.31/-174.9	-106.19	-13.51
			2.83/38.4	5.31/155.8	-99.75	-12.24
			3.14/-86.4	.16/73.5	-104.83	-16.96
507-8	53.6	F	3.24/155.9	.31/-174.9	-106.21	-13.37
			2.83/38.7	5.31/156.1	-99.74	-11.93
			3.14/-86.2	.17/73.3	-104.86	-16.77
8-509	19.6	F	5.29/-17.3	.71/2.3	178.94	18.49
			4.63/-129.8	8.74/-16.1	163.84	-3.39
			5.25/99.4	.48/-136.7	179.20	24.40
6-509	26.6	F	5.37/158.5	.72/-178.7	-176.43	-6.68
			4.62/44.7	8.78/159.2	-164.88	1.49
			5.26/-85.0	.41/46.3	-175.53	-18.12

Table V Machine Data

		Slack Gen at bus 2	Generator at bus 8
Terminal Voltage	V ^{ABC}	1.07/ 0.0 1.09/-120.3 1.07/118.1	1.04/ -5.2 1.07/-125.4 1.05/113.4
	V ⁰¹²	0 1.87/ -.5 .02/136.7	0 1.82/ -5.7 .03/137.5
Current from Machine	I ^{ABC}	7.73/ -15.3 7.73/-131.2 8.20/106.8	7.93/ -17.7 7.89/-132.5 8.52/105.1
	I ⁰¹²	0 13.65/ -13.3 .55/-133.2	0 14.04/ -15.0 .71/-132.4
Generated Power (MW & MVAR)	P ^{ABC}	265.34 275.60 287.32	267.46 278.06 294.47
	Q ^{ABC}	72.67 53.21 61.68	59.09 34.71 42.73
Generator & Transformer Reactances	XG0	.0105	.0105
	XG1	.0402	.0402
	XT0	.0267	.0267
	XT1	.0135	.0135

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Discussion

W. O. Stadlin (Leeds & Northrup Company, North Wales, Pa.): As a matter of application interest it would be desirable to compare the results of Tables III, IV, and V with those of the single phase equivalent. Perhaps the authors would care to comment on this comparison. Based on their investigation of three phase unbalances what are the authors' conclusions regarding the need for a three phase State Estimator?

Manuscript received July 31, 1975.

Luke Yu (Fluor Utah, Inc., San Mateo, CA.): The use of phase quantities (A, B, C), or phase coordinates in solving problems in electrical power engineering field appears to be an appropriate approach. This method has been introduced to study fault conditions instead of symmetrical components.

The application of phase quantities in the solution of load flow problems as contained in this paper has proved interesting.

For a precise representation of a transmission line, a hyperbolic function can be used to include the distributed parameters of transmission lines. This may contribute to the accuracy of the result while a long transmission line is involved.

I would suggest the authors include additional information concerning various electrical element representations such as reactors, condensers, synchronous motors, induction motors, cables, etc., which are always a part of the electrical system in question.

Manuscript received August 1, 1975.

P. A. Rusche and S. C. Bahl (Consumers Power Company, Jackson, Mich.): The authors' choice of actual phase quantities indeed simplified modeling of rotating machines in three-phase simulations. Our own studies using the method in authors' Reference 2 have confirmed that the assumption of a balanced voltage at a system bus unrealistically constrains the simulation to the extent that it may inhibit solution. The synchronous machine model and voltage control as suggested by the authors will, in our judgment, be required for every bus at which voltage control by a synchronous machine is to be simulated.

A three phase network model incorporating rotating machine simulations would be an effective tool to check propagation of steady-state unbalances, including negative sequence currents critical to synchronous machine capability.

Previous work by M. H. Hesse(2) identifying unbalances in untransposed transmission lines has assumed balanced phase voltage drops. For a double circuit 345 kV tower line with common terminals, approximately 80 miles long carrying a per circuit load of about 360 MW, we found the per phase voltage drop to vary by more than a factor of 3 as indicated in the following table:

Table of Phase Voltage Drops

Phase	Sending End Voltage-pu	Receiving End Voltage-pu	Phase Voltage Drop-pu
X	1.037	1.034	.003
Y	1.039	1.035	.004
Z	1.039	1.029	.010

We assume the authors have found similar results and probably have quantified the impact of such unbalanced voltage drops on phase current unbalances.

Field experience and subsequent three-phase load flow analysis has demonstrated the potential for a serious unbalance in paralleled double circuit lines as a function of a phase arrangement in the untransposed lines. The ground currents associated with these unbalances affect transmission line protective schemes and can cause line trips well

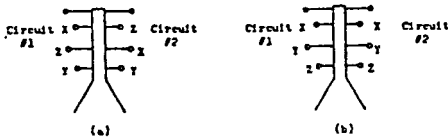


Fig. 1 Phase Arrangements of Double Circuit Tower Line (a) Existing Configuration (b) Mirror-Image Configuration

Manuscript received August 18, 1975.

below circuit ampacities because of reduced relay loadabilities. The following example of an actual 345 kV system consisting essentially of two parallel circuits connecting a 1850 MW pumped storage hydro site over a distance of about 75 miles to the network illustrates the severity of the problem and the importance of the three phase load-flow program in analyzing it.

There are 12 possible phase configurations(2) for a double circuit tower line which reduce to 6 when the two circuits are symmetrical about the vertical axis(3). Figure 1a is the existing configuration; Figure 1b a mirror image configuration. Figure 2 shows the per circuit ground currents for the two arrangements as a function of line loading. Relay settings have been superimposed and illustrate tripping of the existing configuration at about 50% of the 2000 A circuit thermal rating. The corresponding ground currents in the mirror image configuration are negligible.

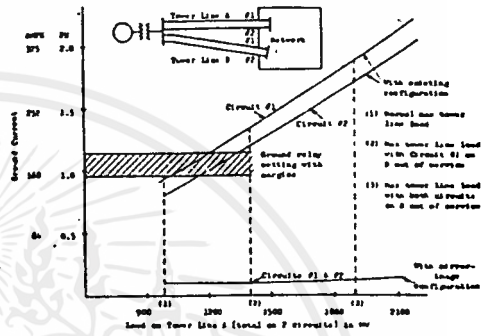


Fig. 2 Per Circuit Ground Currents on Tower Line A as a Function of Line Loading

The potential for this problem in paralleled double circuit tower was originally identified by Hesse(1). A network model is required to analyze propagation of such unbalances. Do the authors concur with our findings that the indicated ground current problem is unique to multicircuit lines with common terminals?

REFERENCES

- [1] M. H. Hesse, "Circulating Currents in Parallel Untransposed Multi-circuit Lines: I - Numerical Evaluations," IEEE Transactions on Power Apparatus and Systems, Vol 85, No 7, July 1966.
- [2] M. H. Hesse, "Circulating Currents in Parallel Untransposed Multi-circuit Lines: II - Methods for Estimating Current Unbalance," IEEE Transactions on Power Apparatus and Systems, Vol 85, No 7, July 1966.
- [3] Transmission Line Reference Book 345 kV and Above, Electric Power Research Institute, 1975, P 100.

K. A. Birt, J. J. Graffy, J. D. McDonald, and A. H. El-Abiad: The authors wish to express their gratitude to the discussors for their interest and most welcome comments and questions, all of which are deemed valuable additions to the paper.

The discussion by Messrs. Rusche and Bahl reflects their great experience with three-phase analysis of power systems. Unfortunately, most of the authors' work to date has been in the program development rather than analysis of systems; and therefore our comments are based on limited experience. Obviously the imbalance in the per phase voltage drop depends on the amount of imbalance in both line impedance and currents as well as the degree of loading in the line. The example given in this paper shows that the per phase voltage drops on line 3-502 (82.1 miles long and carrying 255 MW) is as follows:

Phase	Sending End Voltage - p.u.	Receiving End Voltage - p.u.	Phase Voltage Drop - p.u.
A	1.030	1.005	.025
B	1.082	1.088	-.006
C	1.037	1.025	.012

Manuscript received September 22, 1975.

A better phase voltage drop comparison would be between the phasor quantities. For the above line these are:

Phase	Sending End	Receiving End	Phase Voltage Drop
A	1.030/-10.56°	1.005/18.27°	0.139/65.25°
B	1.082/-129.26°	1.088/-135.11°	0.111/-39.09°
C	1.037/109.03°	1.025/102.13°	0.125/-169.93°

The phase relationships are $\angle a-b = 124.34^\circ$; $\angle b-c = 130.84^\circ$; and $\angle c-a = 104.68^\circ$

In other words, the imbalance in phase voltage drops is not as high as would appear from comparison of voltage magnitudes at both ends of the line. Also, the imbalance in phase angle relations can overshadow the imbalance in magnitudes. The example by Messrs. Rusche and Bahl, relating ground (zero sequence) currents to phase arrangements, illustrates one of the intended uses of our program. We are unable to comment on whether or not ground current problems are unique to multicircuit lines with common terminals. However, it seems advisable to check this possibility whenever new untransposed UHV lines are added or whenever unbalanced loads are expected (e.g. single pole switching, large single phase loads, etc.). Also, other considerations besides ground currents may constrain the phase arrangement choice.

Mr. Yu suggests that a hyperbolic function can be used to include the distributed parameter effects of long transmission lines for better accuracy. The user of our program can accomplish the same thing in one of two ways:

1. By representing the long line by several short segments in series.
2. By representing the long line by its equivalent pi matrices [1] as follows:

$$Z_{\pi} = x Z M S M^{-1}$$

$$\text{and } \frac{1}{2} Y_{\pi} = \frac{x}{2} M T M^{-1} Y$$

where

Z_{π} = series impedance matrix of the equivalent pi.

$\frac{1}{2} Y_{\pi}$ = shunt admittance matrix of the equivalent pi.

x = length of the line.

Z = series impedance matrix per unit length of the line.

Y = shunt admittance matrix per unit length of the line.

M = a matrix with the eigenvectors of YZ as its columns.

S = a diagonal matrix with $\frac{\sinh(w_j x)}{w_j x}$ as the j th diagonal element.

T = a diagonal matrix with $\frac{\tanh(\frac{w_j x}{2})}{\frac{w_j x}{2}}$ as the j th diagonal element.

w_j = square root of the j th eigenvalue of YZ .

The program also accepts any series or shunt, balanced or unbalanced lumped impedance to represent such things as reactors, condensers and special loads. Synchronous motors can be represented in the same way as generators. There is no special representation for induction motors. However, in load flow studies, it is customary to include the induction motor with other loads represented as constant active and reactive powers.

Mr. Stadlin asks about comparing the results of the three-phase load flow program with those of the single phase equivalent. Of course, if such a comparison is made, we would find close agreement between the values of positive sequence variables and their equivalents in the single phase case. As a matter of fact, it is highly recommended that single phase studies always precede three-phase studies and that the bus voltages of the single phase study be used as the initial values (assuming three-phase balance) for the three-phase studies. As for the need for a three-phase estimator, we believe that the effort and cost would be too high to justify its development at the present time.

REFERENCES

- [1] Sarosh Talukdar, PEREC Reports No. V-a and V-b, Users Manual and Programmers Manual for Equivalent Pi Program, Purdue Energy Research and Education Center, Purdue University, Lafayette, Indiana, November, 1965.

Fast-decoupled three-phase load flow

Prof. J. Arrillaga, M.Sc. Tech., Ph.D., C.Eng., F.I.E.E., and B.J. Harker, M.E.

Indexing terms: Load (electric), Modelling, Power-transmission lines, Reactive power, Transmission-network calculations

Abstract

Using as a reference the single-phase fast-decoupled algorithm, this paper describes the modifications required to produce an efficient three-phase fast-decoupled load flow. The compound-coil concept is used in the representation of power-system components, and the effect of automatic voltage regulators is modelled as part of reactive-power Jacobian-matrix equation. It is shown that the three-phase modified fast-decoupled load flow displays all the characteristics of the original single-phase version.

List of principal symbols

- $[I]$ = vector of nodal injected currents
- $[V]$ = vector of nodal voltages $\{|V|e^{j\theta}\}$
- $[Z] = [R] + j[X]$ = matrix of impedances
- $[Y] = [G] + j[B]$ = matrix of admittances
- ΔPP = real power mismatch at busbar i with phase p
- ΔQP = reactive power mismatch at busbar i with phase p
- $(PP)^{SP}$ = specified real power at busbar i with phase p
- $(QP)^{SP}$ = specified reactive power at busbar i with phase p
- $|V|_i$ = voltage magnitude at busbar i with phase p
- $\theta_{ik}^{pm} = \theta_i^p - \theta_k^m$ = angle between busbar i with phase p , and busbar k with phase m
- G_{ik}^{pm} = value of matrix G relating busbar i with phase p , and busbar k with phase m
- $|V|_{int,j}$ = voltage magnitude at the internal busbar of the j th generator
- $\theta_{int,j}$ = voltage phase angle at the internal busbar of the j th generator
- $P_{gen,j}^{SP}$ = specified total power output of j th generator
- $V_{term,j}^{SP}$ = voltage regulator specification for the three terminal voltages for generator j
- nb = number of system busbars, excluding generator internal busbars
- n = total number of system busbars
- ng = number of system generators

1 Introduction

Accurate models of power-system components using phase parameters are available in the literature.¹⁻³ Two recent publications^{4,5} have integrated such models with a Newton-Raphson method for the solution of three-phase load flows. In balanced studies however the conventional Newton algorithm has been displaced by the computationally superior fast-decoupled method. Because computational requirements (time and storage) are more demanding with three-phase models, it would appear that the conventional Newton method should give way to the more powerful decoupled algorithms.

A preliminary investigation carried out to assess the applicability of the fast-decoupled algorithm to three-phase studies gave negative results. Owing to the greater degree of representation used in three-phase models (transformer connections, mutual inductances etc.), the simplifications made in the development of the single-phase fast-decoupled algorithm could not be justified for three-phase load flows, and convergence was poor.

A deeper investigation into decoupling techniques and three-phase power system components indicated that, with suitably modified simplifications, an efficient fast-decoupled three-phase load flow could be achieved with similar characteristics to the single-phase fast-decoupled method.

The development of such an algorithm is described in this paper, emphasis being placed on those features peculiar to the three-phase case.

2 Three-phase power-system modelling

The compound-coil concept provides the basis for a systematic modelling procedure. Linear transformations can be applied to compound coils by simply replacing the single quantities of ordinary networks with appropriate admittance matrices.

The network is first subdivided into subsystems (e.g. generator transformers, transmission lines), with the restriction that should not be mutual coupling between the branches of different systems. With this restriction the subsystem admittance matrices can be combined as follows:

- (a) The self-admittance matrix of any busbar is the sum of all individual self-admittance matrices at that busbar
- (b) The mutual-admittance matrix between any two busbars is the sum of the individual mutual-admittance matrices from all systems containing those two busbars.

A transmission-line subsystem may itself need sectionalising to account for transpositions, changes of line parameters, series capacitors etc. In such cases, the parameters for each section are first found and combined by matrix multiplication to obtain the overall subsystem parameters.

2.1 Transmission lines

For power frequency analysis, a three-phase transmission line can be modelled by three lumped π circuits, with mutual coupling between both the series and shunt branches of all three circuits. This is illustrated by the matrix equivalent of Fig. 1a and its compound-coil equivalent of Fig. 1b.

The subsystem of Fig. 1b can be represented by matrix eqn. 1

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

The effect of earth wires and the influence of ground conductors included in the self and mutual reactances of the phase conductors. For a long line it may be necessary to consider the line as composed of two or three lumped π sections in series.

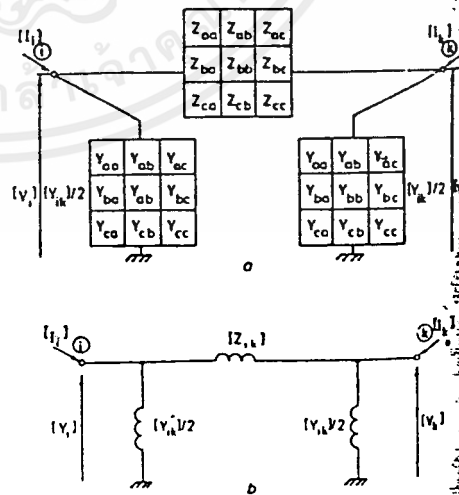


Fig. 1 Transmission-line model

- a Matrix representation
- b Equivalent circuit using three-phase compound coils

Paper 8098 P, first received 17th May 1977 and in revised form 23rd January 1978

Prof. Arrillaga is with the Department of Electrical Engineering, University of Canterbury, Christchurch 1, New Zealand, and Mr. Harker is with the New Zealand Electricity Department, Christchurch 1, New Zealand

2.1.1 Mutually-coupled three-phase lines

It is common for up to four transmission lines to occupy the same right-of-way for a considerable length. Their electrostatic and electromagnetic coupling must then be considered.

Applying the method of linear transformation for compound coils, and assuming bilateral mutual couplings (i.e. $Y_{21} = Y_{12}^T$ etc.), we can write the following matrix equation for the case of two coupled lines, shown in Fig. 2:

$$\begin{bmatrix} I_A \\ I_B \\ I_C \\ I_D \end{bmatrix} = \begin{bmatrix} Y_{11} + Y_{33} & Y_{12} + Y_{34} & -Y_{11} & -Y_{12} \\ Y_{12}^T + Y_{34}^T & Y_{22} + Y_{44} & -Y_{12}^T & -Y_{22} \\ -Y_{11} & -Y_{12} & Y_{11} + Y_{55} & Y_{12} + Y_{56} \\ -Y_{12}^T & -Y_{22} & Y_{12}^T + Y_{56}^T & Y_{22} + Y_{66} \end{bmatrix} \begin{bmatrix} V_A \\ V_B \\ V_C \\ V_D \end{bmatrix}$$

12 x 12

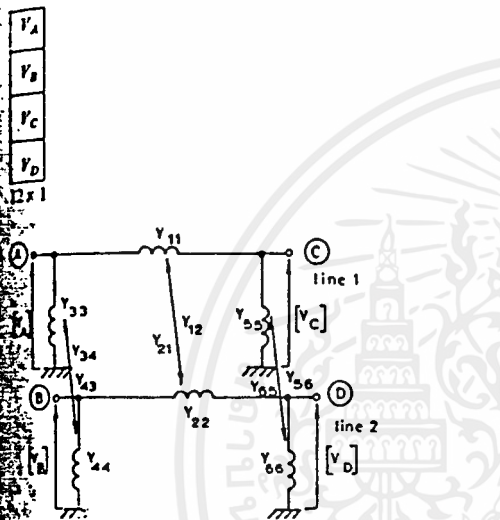


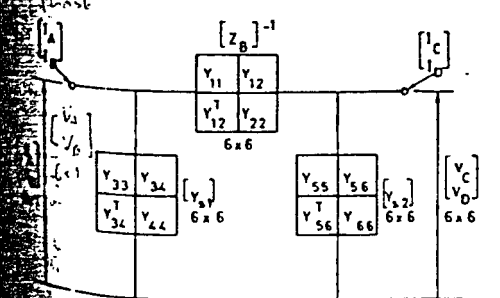
Fig. 2 Mutually-coupled three-phase lines

The combination of pairs of coupled 3 x 3 coils as a single 6 x 6 compound coil, the matrix equivalent of Fig. 3 results. The corresponding matrix equation is as follows:

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

(3)

Eq. 3 now has the same form as eqn. 1, the series impedance $[Z_S]$ and shunt admittance $[Y_S]$ matrices being of orders 3 x 3, 6 x 6, 6 x 6, or 12 x 12 for cases of one, two, three or four coupled three-phase lines, respectively.



Matrix representation of Fig. 2

2.2 Synchronous machines

With reference to Fig. 4, the synchronous machine is represented by a compound coil Y_{gen} , which interconnects an internal busbar of balanced three-phase voltages and a terminal busbar. The three terminal busbar voltages are used to control the balanced internal voltages according to some prespecified relationship.

The following matrix equation applies:

$$\begin{bmatrix} I_i \\ I_h \end{bmatrix} = \begin{bmatrix} [Y_{gen}] & -[Y_{gen}] \\ -[Y_{gen}] & [Y_{gen}] \end{bmatrix} \begin{bmatrix} V_i \\ V_h \end{bmatrix}$$

(4)

where Y_{gen} is a 3 x 3 admittance matrix derived from the sequence component impedances of the machine.

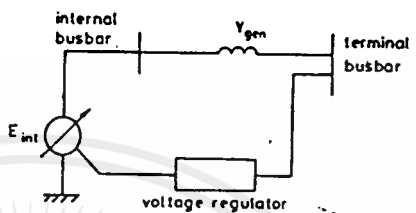


Fig. 4 Synchronous-generator model

2.3 Transformers

In general, any two-winding three-phase transformer can be represented using two coupled compound coils $[Y_{pp}]$, $[Y_{ss}]$, i.e.

$$\begin{bmatrix} I_p \\ I_s \end{bmatrix} = \begin{bmatrix} [Y_{pp}] & [Y_{ps}] \\ [Y_{sp}] & [Y_{ss}] \end{bmatrix} \begin{bmatrix} V_p \\ V_s \end{bmatrix}$$

(5)

This procedure is carried out for the common connections in Reference 1, and it is shown that three basic types of submatrix result. For each type of transformer connection, Table 9 in Appendix 10.1 indicates the appropriate submatrix to be used in the corresponding position of the nodal admittance matrix of eqn. 5. The generality of the model can be increased by modifying these submatrices to include primary- and secondary-winding off nominal taps, which is achieved by—

- (a) dividing the primary self-admittance matrix by α^2
- (b) dividing the secondary self-admittance matrix by β^2
- (c) dividing the mutual admittance matrix by $\alpha\beta$.

where α and β are the primary and secondary tap ratios, respectively. In the p.u. system, a delta winding has an effective tap of $\sqrt{3}$.

3 Newton-Raphson solution

To find the state of the system (i.e. the voltage magnitudes and their phase-angle relationships) so that the specified conditions at the busbars are satisfied, a Newton-Raphson solution requires the following mismatch equations:

- (i) For each of the three phases (p) at every load or generator terminal busbar (i),

$$\Delta P_i^p = (P_i^p)^{SP} - |V_i|^p \sum_{k=1}^n \sum_{m=1}^3 |V_k|^m [G_{ik}^{pm} \cos \theta_{ik}^{pm} + B_{ik}^{pm} \sin \theta_{ik}^{pm}]$$

(6)

$$\Delta Q_i^p = (Q_i^p)^{SP} - |V_i|^p \sum_{k=1}^n \sum_{m=1}^3 |V_k|^m [G_{ik}^{pm} \sin \theta_{ik}^{pm} - B_{ik}^{pm} \cos \theta_{ik}^{pm}]$$

(7)

- (ii) For each internal busbar of each generator, where the nodal voltages are constrained to form a balanced three-phase set, only two equations are required for the unknown voltage ($|V_{int.}^m|$) and unknown angle ($\theta_{int.}^m$), i.e.

$$\Delta P_{gen.J} = P_{gen.J}^{SP} - \sum_{p=1}^3 |V_{int.J}^p| \sum_{k=1}^n \sum_{m=1}^3 |V_k|^m [G_{ik}^{pm} \cos \theta_{ik}^{pm} + B_{ik}^{pm} \sin \theta_{ik}^{pm}]$$

(8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Delta |V|_{reg,j} = |V|_{term,j}^{SP} - |V|_{term,j} (|V|_j^A) \quad (9)$$

where $|V|_j^A$ are the three terminal voltages at busbar j .

The simplest equation results when the voltage regulator monitors one phase ($|V|^A$) only, i.e.

$$\Delta |V|_{reg,j} = |V|_{term,j}^{SP} - |V|_j^A \quad (10)$$

These sets of nonlinear equations are expressed by the equation

$$\begin{bmatrix} \Delta P \\ \Delta P_{gen} \\ \Delta Q \\ \Delta |V|_{reg} \end{bmatrix} = \begin{bmatrix} [A] & [E] & [J] & [M] \\ [B] & [F] & [U] & [N] \\ [C] & [G] & [K] & [P] \\ [D] & [H] & [L] & [R] \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta \theta_{int} \\ \Delta |V|/|V| \\ \Delta |V|_{int} / |V|_{int} \end{bmatrix} \quad (11)$$

The right-hand-side matrix is the usual Jacobian matrix of partial derivatives. Eqn. 11 is solved iteratively for the right-hand-side vector, the corresponding variables are updated, and the Jacobians are then re-evaluated. The procedure continues until the left-hand-side mismatches are within tolerance.

4 Decoupled algorithm

In decoupled solutions, the effects of $\Delta \theta$ on reactive-power flows and $\Delta |V|$ on real-power flows are ignored, i.e. we can simplify eqn. 11 by making

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

and

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

Moreover, it follows from eqn. 10 that

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

Eqn. 11 can thus be written in decoupled form, i.e.

$$\begin{bmatrix} \Delta P \\ \Delta P_{gen,j} \end{bmatrix} = \begin{bmatrix} [A] & [E] \\ [B] & [F] \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta \theta_{int} \end{bmatrix} \quad (12)$$

for $i = 1, nb$ and $j = 1, ng - 1$ (excluding the slack generator)

$$\begin{bmatrix} \Delta Q \\ \Delta |V|_{reg,j} \end{bmatrix} = \begin{bmatrix} [K] & [P] \\ [L] & [R] \end{bmatrix} \begin{bmatrix} \Delta |V|/|V| \\ \Delta |V|_{int,j} / |V|_{int,j} \end{bmatrix} \quad (13)$$

for $i = 1, nb$ and $j = 1, ng$ (including the slack generator).

The coefficients of the Jacobian submatrices for eqns. 12 and 13 are given in Appendix 10.2.

5 Three-phase fast-decoupled algorithm

The basis of a fast-decoupled algorithm is the use of constant Jacobian matrices. Approximations similar to those used in the single-phase-load-flow case are justifiable in eqns. 12 and 13 as follows:

(a) at all nodes

$$Q_k^m \ll B_{kk}^{mm} (|V|_k^m)^2$$

(b) between connected nodes of the same phase

$$\cos \theta_{ik}^{mm} \approx 1$$

and

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

The three-phase Jacobian submatrices in eqns. 12 and 13 require a further approximation to remain constant, namely, ignoring the phase-angle unbalance

$$\theta_{ik}^{pm} = \pm 120^\circ \text{ for } m \neq p$$

The above 'procedural' approximations, however, must not be confused with the 'actual' phase-angle differences or degrees of angle unbalance that the algorithm can handle.

Substituting these values into the Jacobian submatrices of Appendix 10.2 yields

$$\begin{bmatrix} \Delta P \\ \Delta P_{gen,j} \end{bmatrix} = \begin{bmatrix} |V|_i^p M_{ik}^{pm} |V|_k^m \\ \sum_{p=1}^3 |V|_{int,j} M_{ik}^{pm} |V|_k^m \end{bmatrix}$$

$$\begin{bmatrix} \Delta \theta \\ \Delta \theta_{int,i} \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^3 |V|_i^p M_{ik}^{pm} |V|_{int,i} \\ \sum_{m=1}^3 \sum_{p=1}^3 |V|_{int,j} M_{ik}^{pm} |V|_{int,i} \end{bmatrix}$$

$$\begin{bmatrix} \Delta Q \\ \Delta |V|_{reg,j} \end{bmatrix} = \begin{bmatrix} |V|_i^p M_{ik}^{pm} |V|_k^m & \sum_{m=1}^3 |V|_i^p M_{ik}^{pm} |V|_{int,i} \\ [L] & [O] \end{bmatrix} \begin{bmatrix} \Delta |V|_k^m / |V|_k^m \\ \Delta |V|_{int,i} / |V|_{int,i} \end{bmatrix}$$

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

All terms in the matrix $[M]$ are constant, and their values are obtained by substitution of the following:

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

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

and

$$\theta_{ik}^{pm} = \pm 120^\circ \text{ for } m \neq p$$

The matrix $[M]$ is the same as matrix $[B]$, except for the off-diagonal terms that connect nodes of different phases, which are modified allowing for the $\pm 120^\circ$ angle, and adding the $G_{ik}^{pm} \sin \theta_{ik}^{pm}$ term. Eqns. 14 and 15 are then modified as follows:

(i) The left-hand-side defining functions are redefined as $[\Delta P]$, $[\Delta P_{gen,j}/|V|_{int,j}]$ and $[\Delta Q]/|V|_i^p$.

(ii) In eqn. 14, the remaining right-hand-side $|V|$ terms are in p.u., removing the influence of reactive-power (MVar) flows on calculation of $\Delta \theta$ and $\Delta \theta_{int}$.

(iii) In eqn. 15, the remaining right-hand-side $|V|$ terms are constant with the corresponding ones at the right-hand-side vector.

This yields

$$\begin{bmatrix} \Delta P \\ \Delta P_{gen,j} \end{bmatrix} = \begin{bmatrix} M_{ik}^{pm} \sum_{m=1}^3 M_{ik}^{pm} \\ \sum_{p=1}^3 M_{ik}^{pm} \sum_{m=1}^3 \sum_{p=1}^3 M_{ik}^{pm} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta \theta_{int} \end{bmatrix}$$

$$\begin{bmatrix} \Delta Q \\ \Delta |V|_{reg,j} \end{bmatrix} = \begin{bmatrix} M_{ik}^{pm} \sum_{m=1}^3 M_{ik}^{pm} \\ [L^*] & [O] \end{bmatrix} \begin{bmatrix} \Delta |V|_k^m \\ \Delta |V|_{int,i} \end{bmatrix}$$

where $[B^*]$ and $[B^*]$ are constant approximated Jacobian matrices

5.1 Modification of constant Jacobian matrices

The decoupling process is completed by further approximating matrices $[B^*]$ and $[B^*]$ as follows.

It is important that the constant Jacobian matrices reliably approximate tangent slope to the corresponding surface independent of the minor changes in shape that occur during the process. For the surface defined by $[\Delta P/V]$, these changes can be viewed as localised deformities of the surface. The deformities are most pronounced along the axis that corresponds to busbars where shunt admittances are present. If the terms that cause these localised deformities into the tangent hyperplane, a reliably approximate tangent slope is obtainable. This is done in single-phase fast-decoupled load flows⁸ by removing $[B^*]$ the representation of the network elements that predominantly affect reactive-power (MVar) flows.

In the modelling of some three-phase transformer connections (the most common example being the star-delta connection), large shunt admittances are effectively introduced into the system. This is illustrated for the case of a star-delta transformer in Fig. 5.

When the shunts are excluded from the $[B']$ matrix, convergence is inhibited. With the inclusion of these shunts, an excellent convergence is obtained, consistent with the usual single-phase load-flow characteristics.

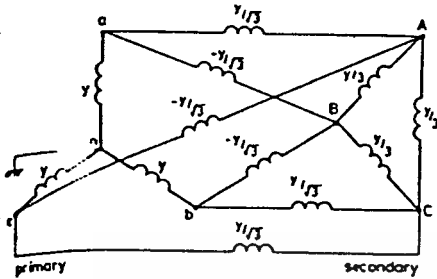


Fig. 5
Equivalent circuit for the star-delta transformer

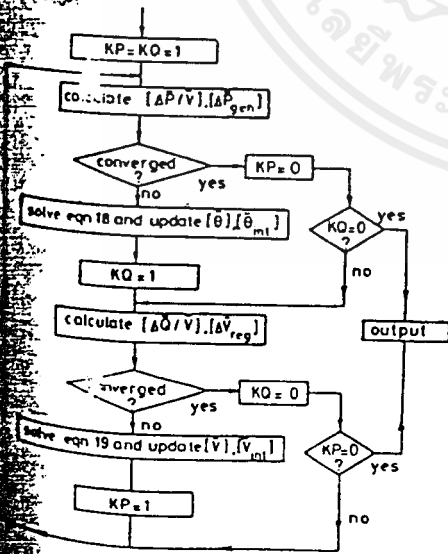
The difference with respect to the normal system shunt elements is purely one of magnitude. The shunt admittances from the transformer model are extremely large (20 p.u. for a 5% leakage reactance transformer), and will therefore alter the entire shape of the multidimensional surface defined by $[\Delta P/V]$. These shunts must be included in $[B']$ if this is to be representative of the tangent hyperplane to the surface. All other shunts should be excluded from $[B']$ to avoid the localised-deformity problem discussed earlier.

This yields the following two three-phase fast-decoupled matrix equations:

$$\begin{bmatrix} \Delta P/V \\ \Delta P_{gen}/V_{int} \end{bmatrix} = [B_M'] \begin{bmatrix} \Delta \theta \\ \Delta \theta_{int} \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} \Delta Q/V \\ \Delta V_{reg} \end{bmatrix} = [B_M''] \begin{bmatrix} \Delta |V| \\ \Delta |V_{int}| \end{bmatrix} \quad (19)$$

The constant approximated Jacobians $[B_M']$ and $[B_M'']$ correspond to the approximated tangent slopes to the multidimensional surfaces defined by the right-hand-side defining functions. Eqs. 18 and 19 are then solved successively, as in the single-phase fast-decoupled method, i.e. according to the flow diagram illustrated in Fig. 6.



Flow diagram of basic iterative procedure

IEE, Vol. 105, No. 8, AUGUST 1978

The solution of eqns. 18 and 19 is carried out using sparsity techniques and near-optimal ordering, as embodied in Zollenkopf's bifactorisation⁹ technique.

In decoupled load flows, the multidimensional surface defined by $[\Delta P/V]$ is considered to be independent of the values of $[V]$. This is not strictly correct, and the changing $[V]$ values will alter the surface defined by $[\Delta P/V]$.

6 Test system

The developed algorithm has been applied to the power system shown in Fig. 7, which includes synchronous generators, a synchronous condenser, a section of four mutually-coupled three-phase lines, star-delta-connected transformers with earthed neutrals, and both primary and secondary taps.

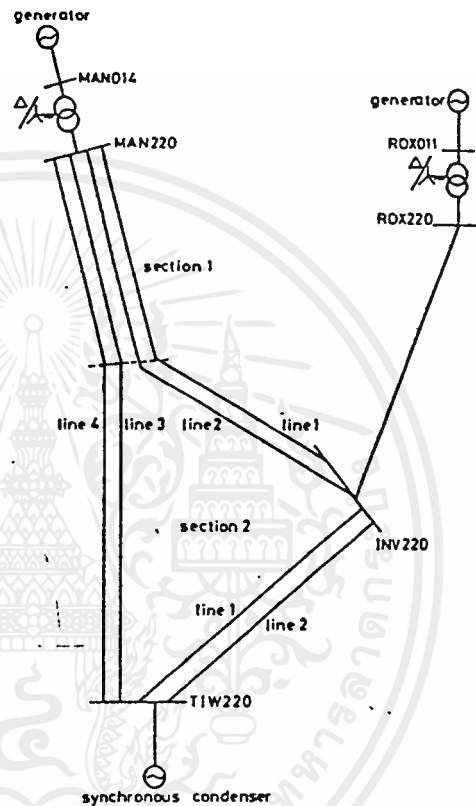


Fig. 7
Single-line diagram of the test system

Fig. 8 illustrates the system divided into eight natural subsystems, each in terms of 3×3 , 6×6 , and 12×12 matrix blocks, representing the various elements and sections. The nodal admittance matrix is formed for each section, and these are then combined as discussed earlier.

The matrix blocks are formed from the input data illustrated in Tables 1-3. Because the input data for the coupled lines consist of various full matrices (up to and including two 12×12 matrices) representative data only are given in Table 3.

To investigate the convergence properties of the algorithm, the following studies were compared:

- Balanced operation of the balanced power system
 - with the generators effectively excluded by setting their sequence impedances to a low value (this is the three-phase equivalent of the usual single-phase load flow)
 - with realistic generator modelling
- Unbalanced operation of the unbalanced power system (with busbar loading conditions as shown in Table 4)
 - generators effectively excluded
 - realistic generator modelling
- As for case (b) (ii), except with the 30° phase shift due to the transformer connections ignored in the starting values

- (d) For the unbalanced system under abnormal operating conditions
- (i) large unbalanced real-power loading (as for case (b) (ii), except for INV220, which now has 300 MW in phase A)
 - (ii) large unbalanced reactive-power loading (as for case (b) (ii), except for INV220, which now has 500 MVAR reactive power in phase A)
 - (iii) large unbalanced real and reactive-power loading ((d) (i) and (d) (ii) loading conditions applied together).

6.1 Discussion of results

The number of iterations required for the maximum mismatches to be within the specified tolerance is shown in Table 5. These results and the authors' experience with other cases leads to the following conclusions as regards convergence properties of the developed algorithm:

- (i) For a balanced system, convergence is similar to that of a single-phase load flow
- (ii) The inclusion of generator plant and the modelling of the voltage regulator does not cause any significant deterioration in convergence
- (iii) Starting values are not critical. Flat voltage and angle initial conditions are perfectly adequate
- (iv) Convergence is achieved even in cases of extreme unbalance.

The resulting system voltages and angles for cases (b) (ii) and (d) (ii) are given in Tables 6 and 7, respectively, and the line power flows for case (b) (ii) in Table 8.

In addition to the significant unbalance, the following features are noticed in Tables 6-8:

- (a) There is an approximate 30° phase shift due to the star-delta-connected transformers
- (b) Balanced voltages occur at the generator internal busbars

Table 1
GENERATOR DATA

Generator	Sequence reactances			Power	Voltage
Name	X_0	X_1	X_2	P	V
MAN14	0.050	0.100	0.010	6.900	1.045
TIW220	0.050	0.100	0.010	0.000	1.020
ROX011	0.050	0.100	0.010	slack	1.050

Table 3
LINE DATA: INV220-TIW220 DOUBLE CIRCUIT [Z_{ij}] SERIES-IMPEDANCE MATRIX

	a	b	c	a	b	c
a	0.0069 + j0.0440	0.0045 + j0.0209	0.0043 + j0.0219	0.0043 + j0.0212	0.0045 + j0.0185	0.0043 + j0.0182
b	0.0045 + j0.0209	0.0066 + j0.0441	0.0044 + j0.0207	0.0045 + j0.0185	0.0046 + j0.0182	0.0044 + j0.0167
c	0.0043 + j0.0219	0.0044 + j0.0207	0.0065 + j0.0442	0.0043 + j0.0182	0.0044 + j0.0167	0.0043 + j0.0167
a	0.0043 + j0.0212	0.0045 + j0.0185	0.0043 + j0.0182	0.0069 + j0.0440	0.0045 + j0.0209	0.0043 + j0.0182
b	0.0045 + j0.0185	0.0046 + j0.0182	0.0044 + j0.0167	0.0045 + j0.0209	0.0066 + j0.0441	0.0044 + j0.0167
c	0.0043 + j0.0182	0.0044 + j0.0167	0.0043 + j0.0163	0.0043 + j0.0219	0.0044 + j0.0207	0.0065 + j0.0442

$[Y_{ij}]$ shunt-admittance matrix

	a	b	c	a	b	c
a	j0.0152	-j0.0021	-j0.0024	-j0.0020	-j0.0009	-j0.0005
b	-j0.0021	j0.0142	-j0.0022	-j0.0009	j0.0111	-j0.0004
c	-j0.0024	-j0.0022	j0.0147	-j0.0008	-j0.0004	j0.0147
a	-j0.0020	-j0.0008	-j0.0008	j0.0152	-j0.0021	-j0.0005
b	-j0.0009	-j0.0011	-j0.0004	-j0.0021	j0.0142	-j0.0004
c	-j0.0005	-j0.0004	-j0.0001	-j0.0024	-j0.0022	j0.0147

Table 4
BUSBAR LOADING DATA

Bus name	Phase A		Phase B		Phase C	
	P	Q	P	Q	P	Q
INV220	50.000	15.000	45.000	14.000	48.300	16.600
MAN014	0.000	0.000	0.000	0.000	0.000	0.000
MAN220	0.000	0.000	0.000	0.000	0.000	0.000
ROX011	0.000	0.000	0.000	0.000	0.000	0.000
ROX220	48.000	20.000	47.000	12.000	51.300	26.300
TIW220	150.000	80.000	157.000	78.000	173.000	72.000
MAN.GN	0.000	0.000	0.000	0.000	0.000	0.000
TIW.GN	0.000	0.000	0.000	0.000	0.000	0.000
ROX.GN	0.000	0.000	0.000	0.000	0.000	0.000

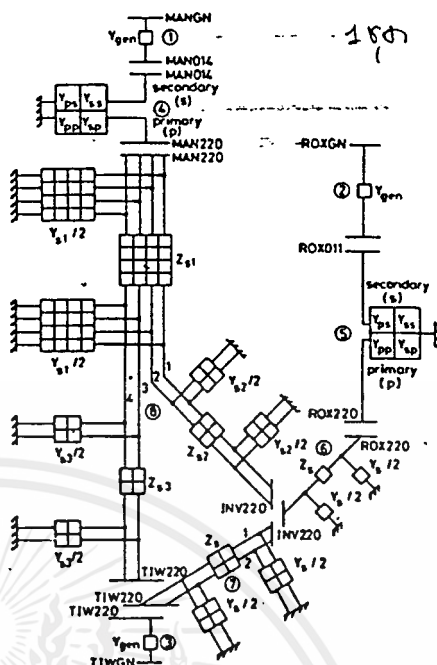


Fig. 8
Test system exploded into eight subsystems

Table 2
TRANSFORMER DATA

Busbar names	ROX220-ROX011	MAN220-MAN014
	Star delta	Star delta
Leakage impedance	0.0060 + j0.1164	0.0018 + j0.0018
Primary tap (%)	0.022	0.025
Secondary tap (%)	0.732	0.732

Table 5
NUMBER OF ITERATIONS TO CONVERGENCE

Case study	Maximum initial mismatch*	Convergence tolerance			
		0.1	0.01	0.001	0.0001
a (i)	2.45	2	1	3	2
a (ii)	2.45	2	1	4	3
b (i)	2.51	2	1	4	4
b (ii)	2.51	2	1	4	4
c	10.34	4	3	6	5
d (i)	2.88	4	4	7	7
d (ii)	3.14	6	6	9	9
d (iii)	3.14	6	6	10	10

*Tolerances and mismatches are in p.u. on a 100 MVA base

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 6
SYSTEM VOLTAGES (V) AND ANGLES (DEGREES) FOR CASE b (ii)

Busbar name	Phase A		Phase B		Phase C		Generation	
	Voltage	Angle	Voltage	Angle	Voltage	Angle	MW	MVA _r
INV220	1-021	29-447	1-022	-90-520	1-026	149-152	0-000	0-000
MAN014	1-045	11-235	1-046	-108-873	1-044	131-112	0-000	0-000
MAN220	1-059	35-399	1-053	-85-244	1-042	155-267	0-000	0-000
ROX011	1-050	-1-573	1-051	-121-550	1-051	118-412	0-000	0-000
ROX220	1-038	26-646	1-048	-93-247	1-035	146-354	0-000	0-000
TIW220	1-020	29-164	1-018	-90-908	1-023	148-845	0-000	0-000
MAN GN	1-111	22-584	1-111	-97-416	1-111	142-584	690-000	288-124
TIW GN	1-056	29-033	1-056	-90-967	1-056	149-033	0-000	114-641
ROX GN	1-078	0-000	1-078	-120-000	1-078	120-000	93-065	89-246

Table 7
SYSTEM VOLTAGES (V) AND ANGLES (DEGREES) FOR CASE d (ii)

Busbar name	Phase A		Phase B		Phase C		Generation	
	Voltage	Angle	Voltage	Angle	Voltage	Angle	MW	MVA _r
INV220	0-884	28-782	1-162	-97-506	1-190	152-450	0-000	0-000
TIW220	1-020	27-345	1-137	-94-500	1-142	149-480	0-000	0-000
ROX220	1-013	25-944	1-084	-94-405	1-070	146-723	0-000	0-000
ROX011	1-050	-1-849	1-050	-121-739	1-052	118-190	0-000	0-000
MAN220	1-057	33-828	1-104	-87-051	1-095	153-583	0-000	0-000
MAN014	1-045	9-280	1-047	-110-561	1-049	129-263	0-000	0-000
MAN GN	1-049	21-416	1-049	-98-584	1-049	141-416	690-000	80-021
TIW GN	1-382	27-448	1-382	-92-552	1-382	147-448	0-000	1174-450
ROX GN	1-065	0-000	1-065	-120-000	1-065	120-000	105-447	48-534

Table 8
LINE POWER FLOWS FOR CASE (b) (ii)

Sending-end busbar name	Receiving-end busbar name	Sending end		Receiving end	
		MW	MVA _r	MW	MVA _r
MAN014	MAN GN	-215-188	-47-660	214-317	94-689
		-236-239	-32-027	239-036	83-135
		-238-577	-57-358	236-651	110-300
TIW220	TIW GN	1-022	-48-856	-0-943	50-603
		13-968	-39-879	-14-459	41-417
		-14-990	-21-861	15-402	22-622
ROX011	ROX GN	-31-477	-32-397	31-380	34-123
		-27-102	-26-447	27-056	27-870
		-34-486	-25-625	34-629	27-253
MAN220	INV220	61-717	-0-449	-59-959	-3-060
		63-923	8-780	-63-029	-12-525
		55-884	9-433	-55-938	-12-890
MAN220	INV220	60-104	3-148	-58-834	-6-633
		60-697	8-862	-59-913	-12-893
		55-386	8-308	-55-251	-11-644
MAN220	TIW220	50-372	4-489	-50-036	-10-361
		45-195	8-696	-44-865	-15-661
		45-799	-2-050	-44-387	-4-002
MAN220	TIW220	50-280	1-766	-49-601	-7-803
		71-153	-10-392	-68-015	5-177
		66-716	21-838	-67-520	-25-218
INV220	TIW220	25-741	4-940	-25-740	-6-486
		29-094	12-418	-29-016	-13-818
		22-994	8-775	-23-049	-10-464
INV220	TIW220	25-755	4-932	-25-727	-6-494
		29-094	12-418	-29-016	-13-818
		23-008	8-782	-23-035	-10-456
INV220	ROX220	17-294	-15-180	-17-761	7-742
		19-699	-13-417	-18-814	8-668
		16-931	-9-624	-16-981	2-118
MAN220	MAN014	-222-495	-8-953	215-204	47-661
		-240-999	-15-945	236-242	32-027
		-223-842	-37-530	238-620	57-358
ROX220	ROX011	-30-247	-27-743	31-473	32-397
		-28-193	-20-668	27-099	26-447
		-34-329	-30-418	34-487	25-625

Total generation 783-0651 MW 492-0122 MVA_r
 Total load 769-6000 MW 335-9000 MVA_r
 Line losses 13-3693 MW 156-1128 MVA_r
 Net loss 0-0959 MW -0-0006 MVA_r

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (v) Balanced angles occur at the generator internal busbars
- (vi) There is an apparent gain in active power flow in any one phase. This power flows through the mutual-coupling terms between phases. The overall active power shows a net loss as expected for a realistic system
- (vii) For the synchronous compensator the individual phases may carry real power; although the total three-phase power always sums to zero.

7 Conclusions

The high efficiency of fast-decoupled algorithms over the conventional Newton-Raphson (roughly six fast-decoupled iterations are equivalent to one full Newton-Raphson iteration) method has been extended in this paper to the solution of three-phase load flows. Such extension is not straightforward, and various modifications to the basic algorithm have been developed to cope with mutual effects between phases and between parallel transmission lines. Also, the presence of large shunt admittances in some transformer equivalent circuits have been found to influence convergence, and need to be included in the active power-mismatch Jacobian matrix. Sparsity techniques and near-optimal ordering have been used to provide fast-repeat solutions. The algorithm provides fast and reliable convergence even with extreme conditions of steady-state unbalance.

8 Acknowledgments

The authors are grateful to P.W. Blakeley, General Manager of the New Zealand Electricity Department and to the Computer Centre of the University of Canterbury for their help.

9 References

- 1 CHEN, M.S., and DILLON, W.E.: 'Power system modelling', *Proc. IEEE*, **62**, 1974, pp. 901-915
- 2 LAUGHTON, M.A.: 'Analysis of unbalanced polyphase networks by the method of phase co-ordinates - Part 1. System representation in the phase frame of reference', *Proc. IEE*, **1968**, **115**, (8), pp. 1163-1172
- 3 BRAMLELLER, A., and PANDEY, R.S.: 'General fault analysis using phase frame of reference', *ibid.* **121**, (5), 1974, pp. 366-368
- 4 WASLEY, R.G., and SLASH, M.A.: 'Newton-Raphson algorithm for 3-phase load flow', *ibid.* **121**, (7), 1974, pp. 630-638
- 5 BIRT, K.A., GRAFFY, J.J., McDonald, J.D., and El-Abiad, A.H.: 'Three-phase load flow program', *IEEE Trans.*, **PAS-95**, 1976, pp. 59-65
- 6 KRON, G.: *Tensor analysis of networks* (MacDonald, London, republished 1965)
- 7 CLARKE, E.: *Circuit analysis of a.c. power systems - Vol. 1* (Wiley, 1943)
- 8 STOTT, B., and Alsac, D.: 'Fast decoupled load flow', *IEEE Trans.*, **PAS-93**, 1974, pp. 859-867
- 9 ZOLLENKOPF, K.: 'Bi-factorisation - Basic computational algorithm and programming techniques'. Conference on large sets of sparse linear equations, Oxford, 1970, pp. 75-96

10 Appendix

10.1 Submatrices for different transformer connections

Table 9
BASIC SUBMATRICES USED IN NODE-ADMITTANCE FORMULATION OF COMMON THREE-PHASE TRANSFORMER CONNECTIONS

Transformer connection	Self admittance	Mutual admittance
Bus P	Bus S	Y_{PS}, Y_{SP}^T
Wye - G	Wye - G	$Y_I, Y_I, -Y_I$
Wye - G	Wye	$Y_{II/3}, Y_{II/3}, -Y_{II/3}$
Wye - G	Delta	$Y_I, Y_{II}, +Y_{III}$
Wye	Wye	$Y_{II/3}, Y_{II/3}, -Y_{II/3}$
Wye	Wye	$Y_{II/3}, Y_{II}, Y_{III}$
Delta	Delta	$Y_{II}, Y_{II}, -Y_{II}$

The characteristic submatrices used in forming the transformer admittance matrices are as follows:

$$Y_I = \begin{bmatrix} Y_t & & \\ & Y_t & \\ & & Y_t \end{bmatrix}$$

$$Y_{II} = \begin{bmatrix} 2Y_t & -Y_t & -Y_t \\ -Y_t & 2Y_t & -Y_t \\ -Y_t & -Y_t & 2Y_t \end{bmatrix}$$

$$Y_{III} = \begin{bmatrix} -Y_t & Y_t & \\ & -Y_t & Y_t \\ Y_t & & -Y_t \end{bmatrix}$$

10.2 Jacobian submatrices

Coefficients of matrix eqn. 12 are as follows:

$$(i) [A_{ik}^{pm}] = [\partial \Delta P_i^p / \partial \theta_k^m]$$

where

$$A_{ik}^{pm} = |V_i^p| |V_k^m| [C_{ik}^{pm} \sin \theta_{ik}^{pm} - B_{ik}^{pm} \cos \theta_{ik}^{pm}]$$

except

$$A_{kk}^{mm} = -B_{kk}^{mm} |V_k^m|^3 - Q_k^m$$

$$(ii) [B_{jk}^{pm}] = [\partial \Delta P_{GEN,j} / \partial \theta_k^m]$$

$$= \sum_{p=1}^3 |V_{int,j}| |V_k^m| [C_{jk}^{pm} \sin \theta_{jk}^{pm} - B_{jk}^{pm} \cos \theta_{jk}^{pm}]$$

$$(iii) [E_{ij}^p] = [\partial \Delta P_i^p / \partial \theta_{int,i}]$$

$$= \sum_{m=1}^3 |V_{int,i}| |V_i^m| [C_{ii}^{pm} \sin \theta_{ii}^{pm} - B_{ii}^{pm} \cos \theta_{ii}^{pm}]$$

$$(iv) [F_{ij}] = [\partial \Delta P_{gen,j} / \partial \theta_{int,i}]$$

where $[F_{ij}] = 0$ for $j \neq i$ because the j th generator has no common bus with the i th generator internal busbar

and

$$[F_{ii}] = \sum_{m=1}^3 -B_{ii}^{mm} |V_i^m|^3 - Q_i^m + \sum_{m=1}^3 \sum_{p=1}^3 |V_{int,i}| |V_i^m| [C_{ii}^{pm} \sin \theta_{ii}^{pm} - B_{ii}^{pm} \cos \theta_{ii}^{pm}]$$

Coefficients of matrix 13 are as follows

$$(i) [K_{ik}^{pm}] = |V_i^m| [\partial \Delta Q_i^p / \partial |V_k^m|]$$

where

$$K_{ik}^{pm} = |V_i^m| |V_k^p| [C_{ik}^{pm} \sin \theta_{ik}^{pm} - B_{ik}^{pm} \cos \theta_{ik}^{pm}]$$

except

$$K_{kk}^{mm} \equiv -B_{kk}^{mm} |V_k^m|^3 + Q_k^m$$

$$(ii) [L_{jk}^m] = |V_k^m| [\partial \Delta P_{gen,j} / \partial |V_k^m|]$$

where $L_{jk}^m = -|V_k^m|$ where k is the terminal busbar of the j th generator

and $L_{jk}^m = 0$ otherwise

$$(iii) [P_{ij}^p] = |V_{int,i}| [\partial \Delta Q_i^p / \partial |V_{int,i}|]$$

$$= |V_{int,i}| \sum_{m=1}^3 |V_i^m| [C_{ii}^{pm} \sin \theta_{ii}^{pm} - B_{ii}^{pm} \cos \theta_{ii}^{pm}]$$

$$(iv) [R_{ij}] = 0 \text{ for all } j, i$$