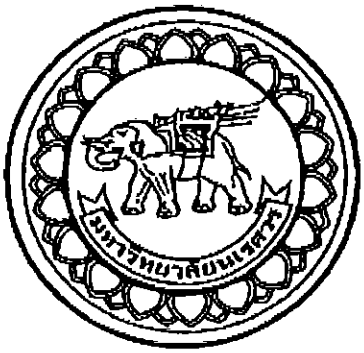


การวิเคราะห์ค่าองค์ประกอบกำลังไฟฟ้าโดยใช้คอมพิวเตอร์ส่วนบุคคล

POWER FACTOR ANALYSIS BY USING A PERSONAL COMPUTER

นายจักรพงศ์ สงเคราะห์ รหัส 44370088

นายณัฏฐ์ ภูวสันติ รหัส 44370146

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 29 มิ.ย. 2548
เลขทะเบียน 4800034
เลขเรียกหนังสือ
มหาวิทยาลัยนเรศวร

T5067145 e.2
ม5.
92220
2547
e.2

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2547

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	**การวิเคราะห์ค่าองค์ประกอบกำลังไฟฟ้าโดยใช้คอมพิวเตอร์ส่วนบุคคล**
ผู้เสนอโครงงาน	นายจักรพงศ์ สงเคราะห์ รหัส 44370088
	นายณัฏฐ์ ภูวสันติ รหัส 44370146
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2547

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

.....กรรมการ

(ดร.สมยศ เกียรติวนิชวิไล)

.....กรรมการ

(อาจารย์ สราวุฒิ วัฒนวงค์พิทักษ์)

หัวข้อโครงงาน	**การวิเคราะห์ค่าองค์ประกอบกำลังไฟฟ้าโดยใช้คอมพิวเตอร์ส่วนบุคคล**
ผู้ดำเนินโครงงาน	**นายจักรพงศ์ สงเคราะห์ รหัส 44370088**
	นายณัฏฐ์ ภูวสันติ รหัส 44370146
อาจารย์ที่ปรึกษา	**ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น**
สาขาวิชา	**วิศวกรรมคอมพิวเตอร์**
ภาควิชา	**วิศวกรรมไฟฟ้าและคอมพิวเตอร์**
ปีการศึกษา	2547

.....

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาโปรแกรมแมทแล็ปเพื่อหาค่าองค์**ประกอบกำลังไฟฟ้าในระบบไฟฟ้าด้วยวิธีการ ดี เอฟ ที โดยนำผลที่ได้มาแสดงทางจอคอมพิวเตอร์**

Project Title	Power factor analysis by using a personal computer
Name	Mr.Chakkrapong Songkhror ID 44370096
	Mr.Nat Phuwasanti ID 44370146
Project Advisor	Assistant Professor Suchart Yammen , Ph. D.
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic Year	2547

.....

ABSTRACT

This project is to study and develop a Matlab program for determining a power factor in the system by using the Discrete Fourier Transform (DFT) method. In addition, the result is displayed on the computer screen.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์นี้ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงงานคือ ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น ที่คอยให้ความช่วยเหลือและคำปรึกษาในทุก ๆ ด้าน และในโอกาสนี้ทางคณะผู้จัดทำโครงงานใคร่ขอขอบคุณเป็นพิเศษกับ ดร.สมยศ เกียรติวนิชวิไล ในคำปรึกษาในด้านต่างๆ รวมทั้งทุก ๆ ท่านที่มีส่วนช่วยทำให้โครงงานนี้ประสบความสำเร็จได้ด้วยดี

นายจักรพงศ์ สงเคราะห์
นายณัฏฐ์ ภูวสันติ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 หลักการและเหตุผล 1

1.2 วัตถุประสงค์ 1

1.3 ขอบข่ายของโครงงาน 1

1.4 ขั้นตอนการดำเนินงาน 2

1.5 ผลที่คาดว่าจะได้รับ 2

1.6 งบประมาณ 2

บทที่ 2 หลักการและทฤษฎี

2.1 กำลังไฟฟ้าขณะใดขณะหนึ่ง 3

2.2 กำลังไฟฟ้าเฉลี่ย 3

2.3 ค่าประสิทธิผลของสัญญาณแบบมีคาบ 6

2.4 องค์ประกอบกำลังไฟฟ้า 9

2.5 การพัฒนาโปรแกรม 10

บทที่3 วิธีการดำเนินโครงงาน

3.1 ออกแบบโปรแกรม 14

3.1.1 โปรแกรมการจำลองค่าในการสุ่มตัวอย่าง 15

3.1.2 โปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า 16

สารบัญ (ต่อ)

หน้า

3.2 สร้างโปรแกรม GUI 17
3.2.1 โปรแกรมจำลองค่าในการสุ่มตัวอย่าง 17
3.2.2 โปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า 18

บทที่ 4 ผลการทดลองและผลการวิเคราะห์

4.1 ผลการทดลอง 20
4.1.1 การทดลองในการใช้โปรแกรมสุ่มค่าตัวอย่าง 20
4.1.2 การทดลองในการใช้โปรแกรมคำนวณค่าองค์ประกอบทางไฟฟ้า 24
4.2 ผลการวิเคราะห์ 28

บทที่ 5 บทสรุป

5.1 สรุปผลการทดลอง 29
5.2 ปัญหาในการทดลองและแนวทางแก้ไข 30
5.3 แนวทางในการพัฒนาโปรแกรมต่อไป 30

เอกสารอ้างอิง 31
ภาคผนวก 32
ประวัติผู้เขียนโครงงาน 40

สารบัญตาราง

ตารางที่ หน้า

4.1 ค่าที่ได้จากการคำนวณในโปรแกรมเปรียบเทียบกับค่าการสุ่มตัวอย่าง 26

5.1 ตารางแสดงผลการเปรียบเทียบโปรแกรม 28

สารบัญรูป

รูปที่		หน้า
2.1	อุปกรณ์ในวงจรไฟฟ้า	3
2.2	กำลังไฟฟ้าสลับที่ถูกส่งผ่านโดยแรงดันสลับ	6
2.3	ภาพแสดงหน้าสำหรับสร้าง GUI	10
2.4	ภาพแสดงการเขียนไฟล์	11
2.5	ภาพแสดงการเปิดไฟล์	11
3.1	แผนผังการทำงานของโปรแกรมทั้งหมด	14
3.2	แผนผังการทำงานของโปรแกรมการจำลองค่าในการสุ่มตัวอย่าง	15
3.3	แผนผังการทำงานของโปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า	16
3.4	หน้าจอการออกแบบโปรแกรมจำลองค่าในการสุ่มตัวอย่าง	17
3.5	หน้าจอการออกแบบโปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า	18
4.1	การทำงานของโปรแกรมในการสุ่มค่าตัวอย่าง	19
4.2	การทำงานของโปรแกรมเมื่อผู้ใช้ใส่ค่าแล้ว	20
4.3	การทำงานของโปรแกรมให้ผู้ใช้กำหนดชื่อไฟล์	21
4.4	การทำงานของโปรแกรมให้ผู้ใช้กำหนดชื่อไฟล์	22
4.5	การทำงานของโปรแกรมคำนวณค่าองค์ประกอบทางไฟฟ้า	23
4.6	การทำงานของโปรแกรมเมื่อกดปุ่ม “กดเมื่อเริ่มต้นเปลี่ยนค่า”	25
4.7	การทำงานของโปรแกรมเพื่อให้กำหนดไฟล์กระแสไฟฟ้า	27
4.8	หน้าต่างโปรแกรมที่ทำการคำนวณแล้ว	28

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

โครงงานนี้เป็นการจำลองวัดค่าองค์ประกอบกำลังไฟฟ้าโดยแสดงผลทางคอมพิวเตอร์ เพื่อที่จะนำผลที่ได้มาคำนวณหาค่าตัวประกอบกำลังไฟฟ้าและคิดคำนวณค่ากำลังไฟฟ้าออกมาโดย ปัจจุบันอุปกรณ์ชนิดนี้มีราคาค้อนข้างสูงจึงทำให้เกิดแนวคิดที่จะจำลองโปรแกรมการวัดตัว ประกอบกำลังไฟฟ้าด้วยคอมพิวเตอร์พร้อมแสดงผลทางจอภาพ

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อจำลองการวัดค่าองค์ประกอบกำลังไฟฟ้า
2. เพื่อนำค่าองค์ประกอบกำลังไฟฟ้ามาแสดงผลในคอมพิวเตอร์
3. เพื่อเป็นแนวทางในการพัฒนาการวัดค่าองค์ประกอบกำลังไฟฟ้า

1.3 ขอบข่ายของโครงงาน

1. ศึกษาทฤษฎีที่เกี่ยวข้องกับการจำลองการวัดค่าองค์ประกอบกำลังไฟฟ้า
2. ศึกษาการเขียนโปรแกรมและออกแบบโปรแกรมสำหรับการวัดค่าองค์ประกอบกำลังไฟฟ้า

1.4 ขั้นตอนการดำเนินงาน

	2546		2547									
	พย	ธค	มค	กพ	มีค	เมย	พค	มิย	กค	สค	กย	ตค
ศึกษาค้นคว้าในเรื่องตัวประกอบกำลังทางไฟฟ้า	←				→							
ศึกษาค้นคว้าการใช้โปรแกรม Matlab				←		→						
ออกแบบการเขียนโปรแกรม						←		→				
พัฒนาทดสอบและแก้ไข								←		→		
สรุปและจัดทำรายงาน											←	→

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถจำลองการวัดค่าองค์ประกอบกำลังไฟฟ้าได้
2. สามารถสร้างโปรแกรมสำหรับการใช้งานในการหาค่าองค์ประกอบกำลังไฟฟ้าโดยคอมพิวเตอร์ได้

1.6 งบประมาณ

1. ค่าวัสดุอุปกรณ์	1,000	บาท
2. ค่าถ่ายเอกสาร	1,000	บาท
รวมทั้งสิ้น	**2,000**	**บาท (สองพันบาทถ้วน)**

(หมายเหตุ) ถัวเฉลี่ยทุกรายการ

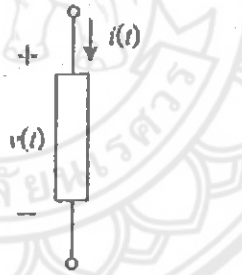
บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 กำลังไฟฟ้าขณะใดขณะหนึ่ง

โดยทั่วไป กำลังไฟฟ้าขณะใดขณะหนึ่ง คือ ผลคูณของกระแสสลับและแรงดันในมิติของเวลาที่มีความเกี่ยวข้องกับอุปกรณ์ใดๆ ในวงจรไฟฟ้า เมื่อพิจารณาอุปกรณ์ประเภทเฉื่อยที่มีทิศทางอ้างอิงกระแส $i(t)$ ที่ไหลผ่านและแรงดัน $v(t)$ ที่ตกคร่อมสำหรับวงจรไฟฟ้าดังที่แสดงไว้ในรูป 2.1 จะพบว่า

$$p(t) = v(t)i(t) \tag{2.1}$$

โดย $p(t)$ คือกำลังไฟฟ้าขณะใดขณะหนึ่งที่ส่งผ่านพลังงานไปยังอุปกรณ์ไฟฟ้า มีหน่วยเป็น วัตต์ [W] เนื่องด้วยแรงดัน $v(t)$ และกระแส $i(t)$ เป็นฟังก์ชันของเวลา ดังนั้น กำลังไฟฟ้าขณะใดขณะหนึ่งจึงเป็นฟังก์ชันของเวลาด้วย



รูป 2.1 อุปกรณ์ในวงจรไฟฟ้า

2.2 กำลังไฟฟ้าเฉลี่ย

จากอุปกรณ์ทางไฟฟ้าในรูป 2.1 สมมุติให้แรงดัน $v(t)$ เป็นสัญญาณที่มีคาบ T วินาที เราสามารถเขียนความสัมพันธ์ของแรงดันได้เป็น

$$v(t) = v(t+T)$$

สำหรับวงจรเชิงเส้น เราทราบว่ากระแส $i(t)$ จะเป็นสัญญาณที่มีคาบเดียวกันด้วยที่มีความสัมพันธ์เป็น

$$i(t) = i(i+T)$$

ดังนั้น กำลังไฟฟ้าขณะใดขณะหนึ่ง มีค่าเป็น

$$p(t) = v(t)i(t) = v(t+T)i(t+T)$$

เราทราบว่า ค่าเฉลี่ยของสัญญาณที่มีคาบตลอดช่วงเวลา T วินาที จะได้เป็น

$$P = \frac{1}{T}\int_{t_0}^{t_0+T} p(t)dt \qquad (2.2)$$

โดยที่ t_0 คือเวลาเริ่มต้นใดๆ และ P เป็นสัญลักษณ์ของค่าเฉลี่ยของกำลังไฟฟ้าขณะใดขณะหนึ่ง ต่อมา กำหนดให้แรงดัน $v(t)$ เป็นสัญลักษณ์แบบซายน์ที่มีค่าเป็น

$$v(t) = v_m \cos(\omega t + \theta_v)$$

และในกรณีวงจรเชิงเส้น ณ. สภาวะคงตัวผลตอบลัพธ์กระแสก็จะเป็นสัญญาณแบบซายน์ที่มีความถี่เชิงมุม ω [rad/sec] ด้วยดังนั้น จะพบว่า

$$i(t) = i_m \cos(\omega t + \theta_I)$$

เราทราบว่าความถี่และคาบของสัญญาณแรงดันและกระแสมีความสัมพันธ์เป็นดังนี้ คือ

$$\omega = \frac{2\pi}{T}$$

ดังนั้น กำลังไฟฟ้าขณะใดขณะหนึ่งที่มีคาบส่งผ่านไปยังอุปกรณ์ทางไฟฟ้าซึ่งมีค่าเป็น

$$p(t) = V_m I_m \cos(\omega t + \theta_v)\cos(\omega t + \theta_I)$$

โดยอาศัยเอกลักษณ์ตรีโกณสำหรับผลคูณของฟังก์ชันโคซายน์ทั้งสอง จะพบว่า

$$p(t) = \frac{V_m I_m}{2}\left[\cos(\theta_V - \theta_I) + \cos(2\omega t + \theta_V + \theta_I)\right]$$

จะเห็นได้ว่ากำลังไฟฟ้าขณะใดขณะหนึ่งมีสองพจน์ที่เกี่ยวข้อง พจน์แรก $\cos(\theta_V - \theta_I)$ เป็นค่าคงที่ไม่ขึ้นอยู่กับเวลา ส่วนพจน์ที่เหลือ $\cos(\omega t + \theta_V + \theta_I)$ แปรเปลี่ยนตามสัญญาณแบบซายน์ที่มีความถี่เชิงมุมเป็นสองเท่าของความถี่เชิงมุมของแรงดัน v *(t)* ฉะนั้น ***กำลังไฟฟ้าเฉลี่ย*** (Average Power) ที่ส่งผ่านพลังงานไปยังอุปกรณ์ทางไฟฟ้าโดยที่เราเลือกค่า $t_0 = 0$ วินาที มีค่าเป็น

$$P = \frac{1}{T}\int_0^T \frac{V_m I_m}{2}\left[\cos(\theta_V - \theta_I) + \cos(2\omega t + \theta_V + \theta_I)\right]dt$$

หรือ

$$P = \frac{1}{T}\int_0^T \frac{V_m I_m}{2}\cos(\theta_V - \theta_I)dt + \frac{1}{T}\int_0^T \frac{V_m I_m}{2}\cos(2\omega t + \theta_V + \theta_I)dt$$

$$= \frac{V_m I_m \cos(\theta_V - \theta_T)}{2T}\int_0^T dt + \frac{V_m I_m}{2T}\int_0^T \cos(2\omega t + \theta_V + \theta_I)dt$$

ทราบว่า อินติกรัลพจน์ที่สองมีค่าเป็นศูนย์ เพราะค่าเฉลี่ยตามช่วงเวลาหนึ่งคาบเวลาของฟังก์ชั่นโคซายน์ค่าเป็นศูนย์เสมอ ดังนั้น จะพบว่า

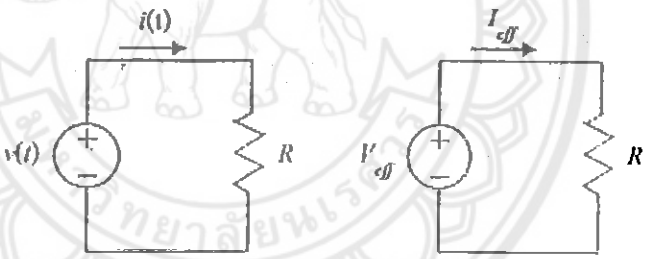
$$p(t) = \frac{V_m I_m}{2}\cos(\theta_V - \theta_I) \qquad (2.3)$$

โดย $\cos(\theta_v - \theta_i)$ เรียกว่า องค์ประกอบกำลัง (Power Factor) ซึ่งเป็นค่าโคซายน์ของผลต่างเฟสระหว่างแรงดัน $v(t)$ และกระแส $i(t)$ นั่นเอง

2.3 ค่าประสิทธิผลของสัญญาณแบบมีคาบ

โดยทั่วไป ค่าแรงดันฟ้าที่วัดได้จากปลั๊กไฟฟ้าในบ้านโดยใช้เครื่องวัดแรงดัน มีค่าเท่ากับ 220 [V] ค่าที่อ่านได้จากเครื่องมือวัดแรงดันนี้ไม่ใช่ค่าเฉลี่ยของสัญญาณแบบซายน์เนื่องจากเราทราบว่าค่าเฉลี่ยของสัญญาณซายน์มีความถี่ใดๆ มีค่าเป็นศูนย์เสมอ นอกจากนี้ก็ไม่ใช่ค่าขณะใดขณะหนึ่งหรือค่าสูงสุดของแรงดัน $v(t) = V_m \cos \omega t$ [V] เช่นกัน

ค่าประสิทธิผลของแรงดันเป็น การวัดประสิทธิผล ในการส่งผ่านกำลังไฟฟ้าไปยังความต้านทาน แนวคิดของค่าประสิทธิผลมาจากความต้องการที่ทำให้แรงดันสลับ (หรือกระแสสลับ) ส่งผ่านกำลังไฟฟ้าเฉลี่ยต่อความต้านทานเหมือนกับการใช้แรงดันตรงสมมูล V_{eff} (หรือกระแสตรงสมมูล I_{eff}) วัตถุประสงค์ ณ.ที่นี้ คือ ต้องการทราบค่า V_{eff} (หรือ I_{eff}) ที่จะส่งผ่านกำลังไฟฟ้าเฉลี่ยต่อความต้านทานเท่ากับกำลังไฟฟ้าสลับที่ถูกส่งผ่านโดยแรงดันสลับ ต่อความต้านตัวเดียวกันดังที่แสดงไว้ในรูป 2.2



รูป 2.2 กำลังไฟฟ้าสลับที่ถูกส่งผ่านโดยแรงดันสลับ

เราทราบว่า พลังงานที่ส่งผ่านไปยังความต้านทาน R ในช่วงหนึ่งคาบเวลา T วินาที คือ $w = PT$ โดยที่ P เป็นกำลังไฟฟ้าเฉลี่ย

เมื่อกระแสสลับ $i(t)$ ไหลผ่านความต้านทาน R เราจะได้กำลังไฟฟ้าเฉลี่ยที่ R ได้รับมีค่าเป็น

$$P = \frac{1}{T}\int_0^T i^2(t)Rdt \qquad (2.4)$$

และถ้ากระแสตรงสมมูล I_{eff} ไหลผ่านความต้านทาน R ในช่วงเวลา T วินาที ทำให้กำลังไฟฟ้าเฉลี่ยที่ R ได้รับมีค่าเป็น

$$P = I_{eff}^2 R \tag{2.5}$$

ฉะนั้น ถ้ากำลังไฟฟ้าเฉลี่ยในสมการ (7-4) เท่ากับกำลังไฟฟ้าเฉลี่ยในสมการ (7-5) จะได้ความสัมพันธ์

$$I_{eff}^2 R = \frac{1}{T}\int_0^T i^2(t)Rdt$$

หรือ

$$I_{eff} = \sqrt{\frac{1}{T}\int_0^T i^2(t)dt} \tag{2.6}$$

จะพบว่าค่า I_{eff} เป็นรากที่สองของค่ายกกำลังสองเฉลี่ย (Root Mean Square,RMS) ของกระแส $i(t)$ ดังนั้น ค่ากระแสประสิทธิผล I_{eff} อาจเรียกว่า กระแสรากที่สองของยกกำลังสองเฉลี่ย ซึ่งให้สัญลักษณ์ย่อว่า I_{rms} หรือกล่าวได้ว่า ค่าประสิทธิผลของกระแส คือกระแสตรงที่ส่งผ่านกำลังไฟฟ้าเฉลี่ยเช่นเดียวกับในกรณีที่กระแสสลับ

ในทำนองเดียวกับกรณีของการหาค่าประสิทธิผลของกระแส ค่าประสิทธิผลของแรงดันสามารถเขียนความสัมพันธ์ได้ว่า

$$V_{eff}^2 = V_{rms}^2 = \frac{1}{T}\int_0^T v^2(t)dt$$

หรือ

$$V_{rms} = \sqrt{\frac{1}{T}\int_0^T v^2(t)dt}$$

ในกรณีที่ กำหนดให้กระแสสลับ $i(t)$ เป็นสัญญาณแบบซายน์ซึ่งมีค่าเป็น $i(t) = I_m \cos \omega t [A]$ เราสามารถหาค่าประสิทธิผลของกระแสสลับ $i(t)$ ได้เป็น

$$I_{rms} = \sqrt{\frac{1}{T}\int_0^T I_m^2 \cos^2 \omega t dt}$$

จากเอกลักษณ์ของตรีโกณ $\cos^2 \theta = (1+\cos 2\theta)/2$ จะพบว่า

$$I_{rms} = \sqrt{\frac{I_m^2}{2T}\int_0^T [1+\cos 2\omega t]dt}$$

$$= \sqrt{\frac{I_m^2}{2T}\int_0^T 1dt + \frac{I_m^2}{2T}\int_0^T \cos 2\omega t dt}$$

เนื่องด้วยอินทิกรัลของ $\cos 2\omega t$ ตลอดช่วงเวลาหนึ่งคาบ T วินาที มีค่าเท่ากับศูนย์จะได้รับความสัมพันธ์เป็น

$$I_{rms} = \sqrt{\frac{I_m^2}{2T}\int_0^T 1dt}$$

หรือ

$$I_{rms} = \frac{I_m}{\sqrt{2}} \qquad (2.7)$$

นั่นคือ ค่าประสิทธิผลของสัญญาณแบบซายน์จะมีค่าเท่ากับ $1/\sqrt{2}$ เท่าของค่าสูงสุดของสัญญาณ เสมอซึ่งคุณสมบัตินี้สามารถใช้เป็นมาตรฐานในการเปรียบเทียบปริมาณของสัญญาณซายน์ต่างๆได้ โดยทั่วไปเมื่อพูดถึงค่าของแรงดันหรือกระแสสลับที่ไม่มีการบอกกล่าวอย่างชัดแจ้งว่าเป็นค่าใด เรามักหมายถึงค่าประสิทธิผลนี้เสมอและจากเครื่องวัดชนิดที่ใช้วัดปริมาณกระแสสลับ ค่าที่อ่านได้ก็จะเป็นค่าประสิทธิผลนี้เช่นกัน แต่อย่างไรก็ตามสำหรับวงจรอิเล็กทรอนิกส์หรือวงจรทางด้านสื่อสาร แรงดันหรือกระแสมักจะระบุเป็นค่าสูงสุดของสัญญาณ ดังนั้นจำเป็นอย่างยิ่งที่ต้องระบุความแตกต่างของสัญญาณที่วัดได้ว่าเป็นค่าประสิทธิผลหรือค่าสูงสุดของสัญญาณ

2.4 องค์ประกอบกำลังไฟฟ้า

กำลังไฟฟ้าเฉลี่ยที่ป้อนให้กับโหลดของวงจรในสภาวะคงตัว

$$P = \frac{V_m I_m}{2}\cos(\theta_V - \theta_I)$$

จากสมการ ผลคูณของ $\frac{V_m I_m}{2}$ เรียกว่า กำลังไฟฟ้าปรากฏ (Apparent Power) ที่มีหน่วยเป็นโวลท์แอมแปร์ (Volt-Ampere; VA) โดยที่ส่วนเทอม $\cos(\theta_V - \theta_I)$ เรียกว่า องค์ประกอบกำลังทางไฟฟ้าแบบตาม (Lagging Power Factor) เมื่อ $\theta_V > \theta_I$ นอกจากนี้ถ้า $\theta_V < \theta_I$ แล้ว $\cos(\theta_V - \theta_I)$ เรียกว่าองค์ประกอบกำลังทางไฟฟ้าแบบนำ (Leading Power Factor)

ในกรณีที่ มุม $\theta_V - \theta_I$ มีค่าเป็น 0° เนื่องมาจากค่ามุมของความต่างศักย์และค่ามุมของกระแสไฟฟ้ามีค่าเดียวกัน นั่นคือโหลดเป็นความต้านทานเพียงอย่างเดียว

ในกรณีที่ มุม $\theta_V - \theta_I$ มีค่าเป็น +90° เนื่องมาจากค่ามุมของความต่างศักย์มีค่ามากกว่าค่ามุมของกระแสไฟฟ้า นั่นคือโหลดที่เป็นตัวเหนี่ยวนำไฟฟ้าเพียงอย่างเดียว

ในกรณีที่ มุม $\theta_V - \theta_I$ มีค่าเป็น -90° เนื่องมาจากค่ามุมของความต่างศักย์มีค่าน้อยกว่าค่ามุมของกระแสไฟฟ้า นั่นคือโหลดที่เป็นตัวเก็บประจุไฟฟ้าเพียงอย่างเดียว

ในการหาค่าองค์ประกอบกำลังไฟฟ้า มีขั้นตอนดำเนินงานดังนี้สมมุติว่าสัญญาณค่าแรงดันไฟฟ้าเป็น $v[n]$ และสัญญาณค่ากระแสไฟฟ้าเป็น $i[n]$

ขั้นตอนที่ 1: ทำการหาค่า DFT (Discrete Fourier Transform) ของสัญญาณทั้งสองได้เป็นดังนี้

การหา FFT ของค่าแรงดันไฟฟ้า

$$v[n] \overset{DFT}{\Leftrightarrow} V[k]$$

$$i[n] \overset{DFT}{\Leftrightarrow} I[k]$$

ขั้นตอนที่ 2: การหาค่าของผลต่างเฟสของสัญญาณแรงดันไฟฟ้าและกระแสไฟฟ้า โดยการคำนวณผลต่างของมุมเฟส DFT ของแรงดันไฟฟ้า และ มุมเฟสของ DFT ของกระแส I[k] ที่ตำแหน่งที่ k=1

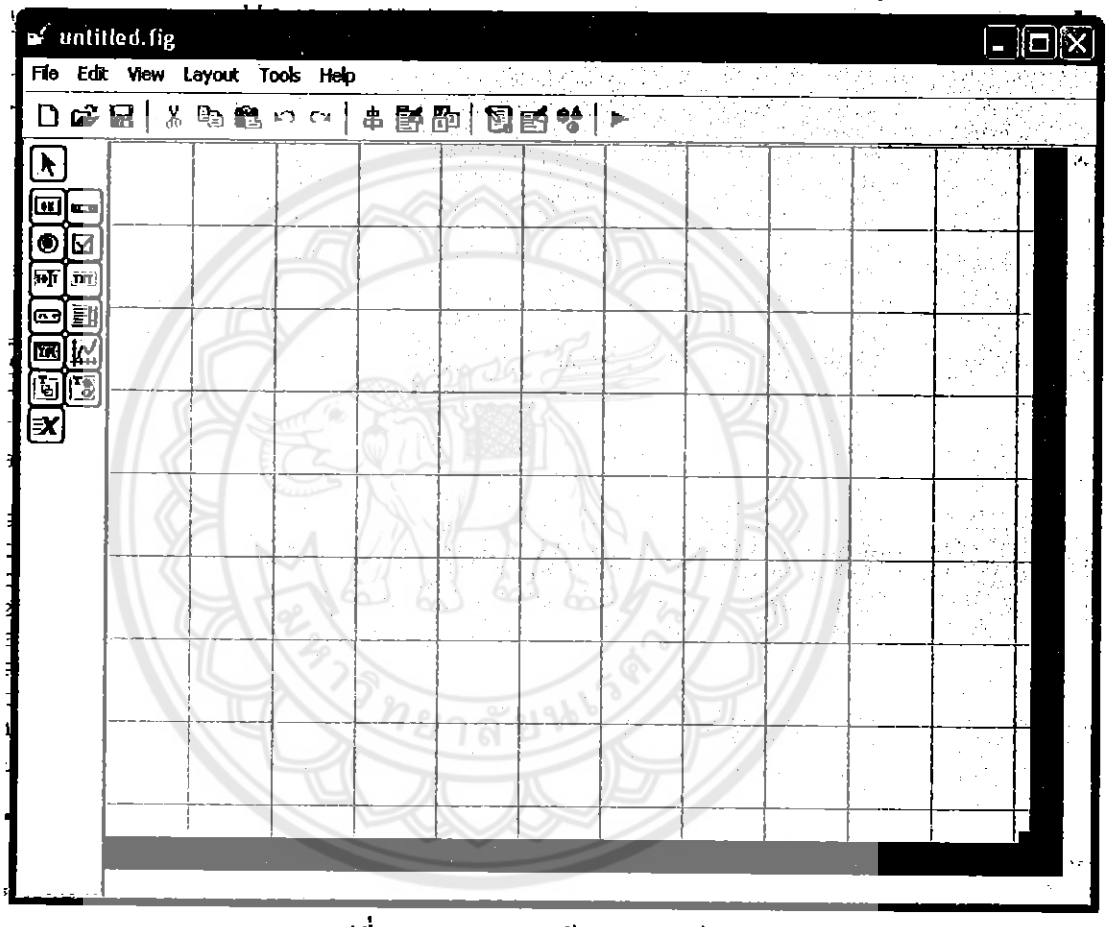
$$\phi = angle\{V[1]\} - angle\{I[1]\}$$

ขั้นตอนที่ 3: คำนวณค่าองค์ประกอบกำลังไฟฟ้า

$$\cos\phi = \cos(angle\{V[1]\} - angle\{I[1]\})$$

2.5 การพัฒนาโปรแกรม

ในที่นี้ได้พัฒนาโปรแกรมเป็นMatlab GUI โดยใช้หลักการ และทฤษฎี matlab ดั้งนี้



รูปที่ 2.3 ภาพแสดงหน้าสำหรับสร้าง GUI

ในโปรแกรม matlab นั้นมีการใช้งานด้านกราฟฟิกอินเตอร์เฟสผนวกอยู่ด้วยคือ ระบบ GUI ซึ่งมีการใช้งานได้อย่างสะดวกและง่ายต่อการใช้งานเนื่องจากเป็นระบบที่ทำงานแบบ Visual ซึ่งสามารถลากออปเจคต์ต่างมาวางบนฟอร์มเพื่อกำหนดหน้าตาของโปรแกรมจนเสร็จเรียบร้อยแล้ว ก็เพียงแต่เข้าไปที่view-> M-file Editor เพื่อเพิ่มโค้ต M-file ลงไปได้เลย หรือจะใช้ภาษาซีก็ได้เช่นกันแต่ในโครงงานนี้ขออธิบายในรูปแบบ M-file

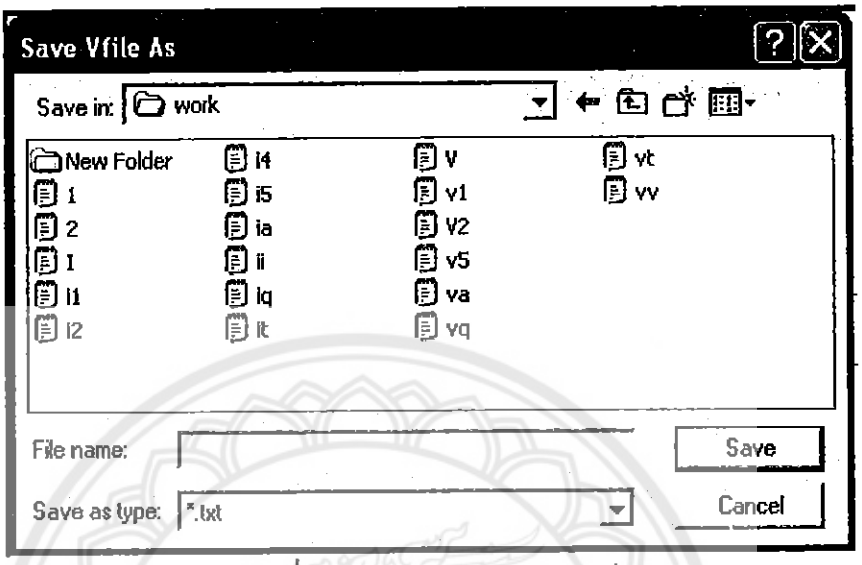
ในส่วนแรกนั้นเพื่อที่จะเริ่มเขียนโปรแกรมในโครงงานนี้นั้นก็จะต้องทำการเขียนไฟล์อ่านไฟล์โดยในการเขียน GUI นั้นจะมีฟอร์มการใช้งานดั้งนี้

การเขียนไฟล์

[file,path] = uiputfile('*.txt','Save Vfile As');

save(fullfile(path, file), 'v','-ASCII');

ซึ่งมีรูปแบบดั้งนี้



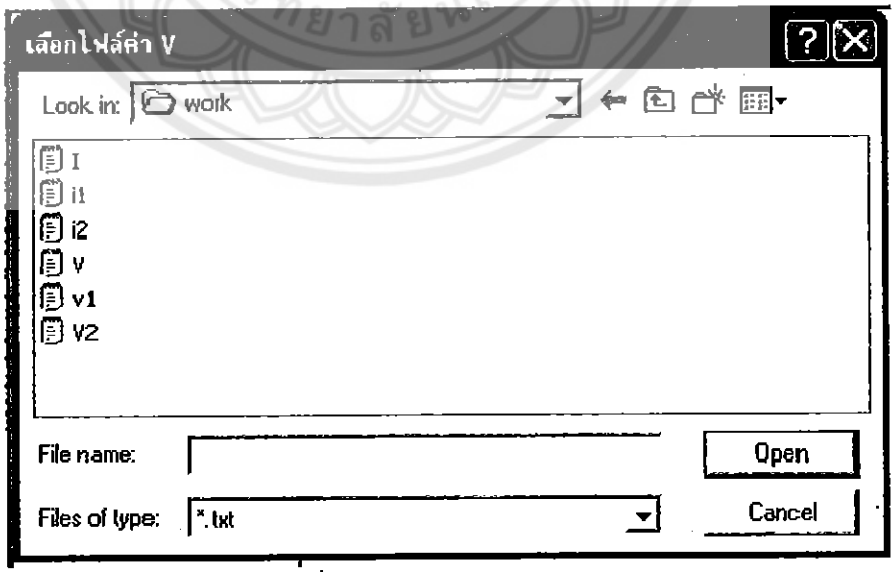
รูปที่ 2.4 ภาพแสดงการเขียนไฟล์

การเปิดไฟล์

[filename, pathname] = uigetfile('*.txt', 'เลือกไฟล์ค่า I');

Iin = load(fullfile(pathname, filename));

ซึ่งมีรูปแบบดั้งนี้



รูปที่ 2.5 ภาพแสดงการเปิดไฟล์

การใช้งานEdit

ให้ทำการโหลดฟังชั่นการใช้งานการรับค่าจาก Edit โดยการนำค่าดั้งต่อไปนี้มาใส่ในส่วนของ

function edit4_Callback(hObject, eventdata, handles)

โดยค่าที่ทำการโหลดใช้งานคือ

```
edit= str2double(get(hObject, 'String'));
if isnan(edit)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
```

และหลังจากเช็คค่าว่ามีค่าในeditแล้วให้นำมาใส่ในตัวแปรดังนี้

```
handles.metricdata.edit = edit;
guidata(hObject,handles)
```

ในการนำไปใช้นั้นสามารกำหนดตัวแปรที่จะรับค่าเข้าจากeditได้โดย

กรอกตัวแปรที่ต้องการ= handles.metricdata.edit

เป็นต้น

การใช้งาน Text

```
set(handles.text16, 'String', กรอกตัวแปรที่ต้องการแสดง)
```

การใช้งาน Axes

```
axes(handles.axes); cla;
plot(กรอกตัวแปรที่ต้องการแสดง,'r');
```

การใช้งาน Load

กรอกตัวแปรที่ต้องการ = load('กรอกไฟล์ที่ต้องการโหลด');

การใช้งาน Save

```
save ('กรอกไฟล์ที่ต้องการโหลด');
save('กรอกไฟล์ที่ต้องการเก็บลง', 'กรอกค่าตัวแปรที่ต้องการเก็บลงไฟล์','-ASCII');
```

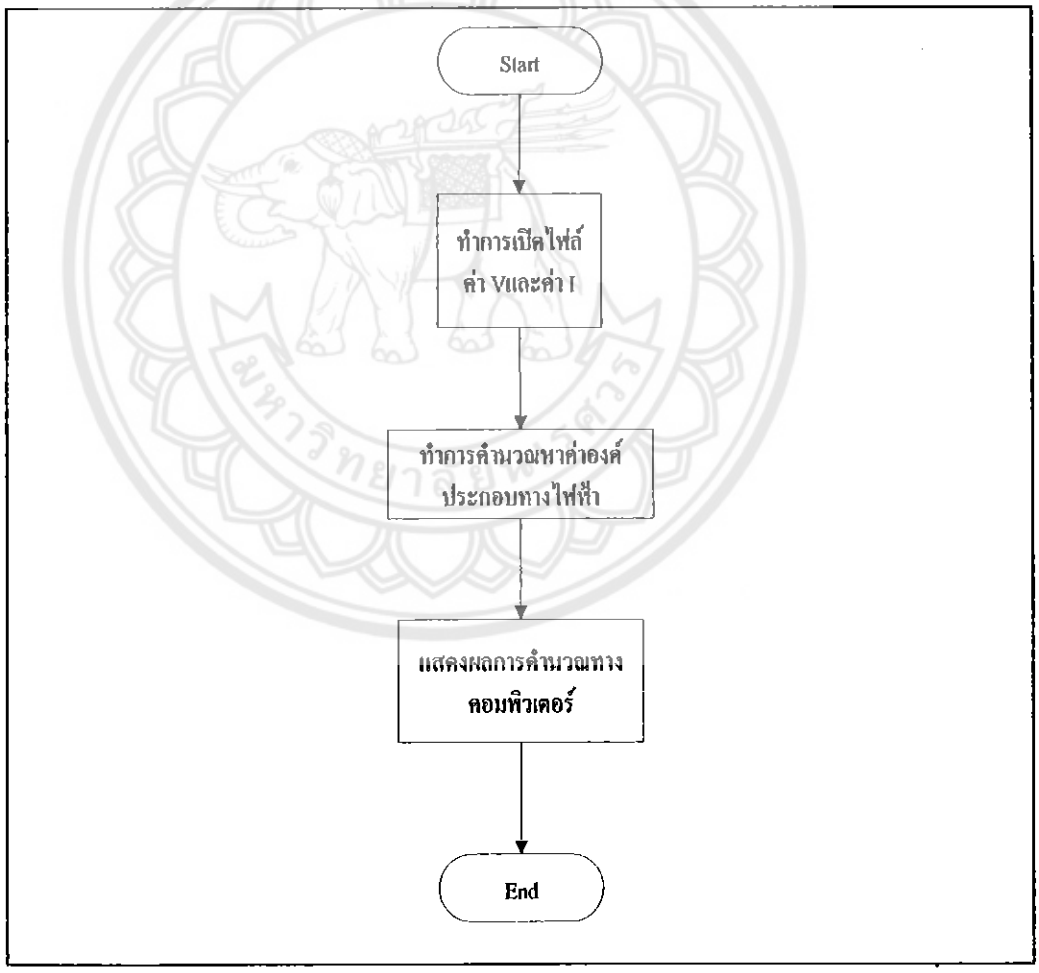
การใช้งาน serial

```
S1 = serial ('com2', BaudRate', 9600,'DataBit', 7)
```

บทที่3
วิธีการดำเนินโครงงาน

3.1 ออกแบบโปรแกรม

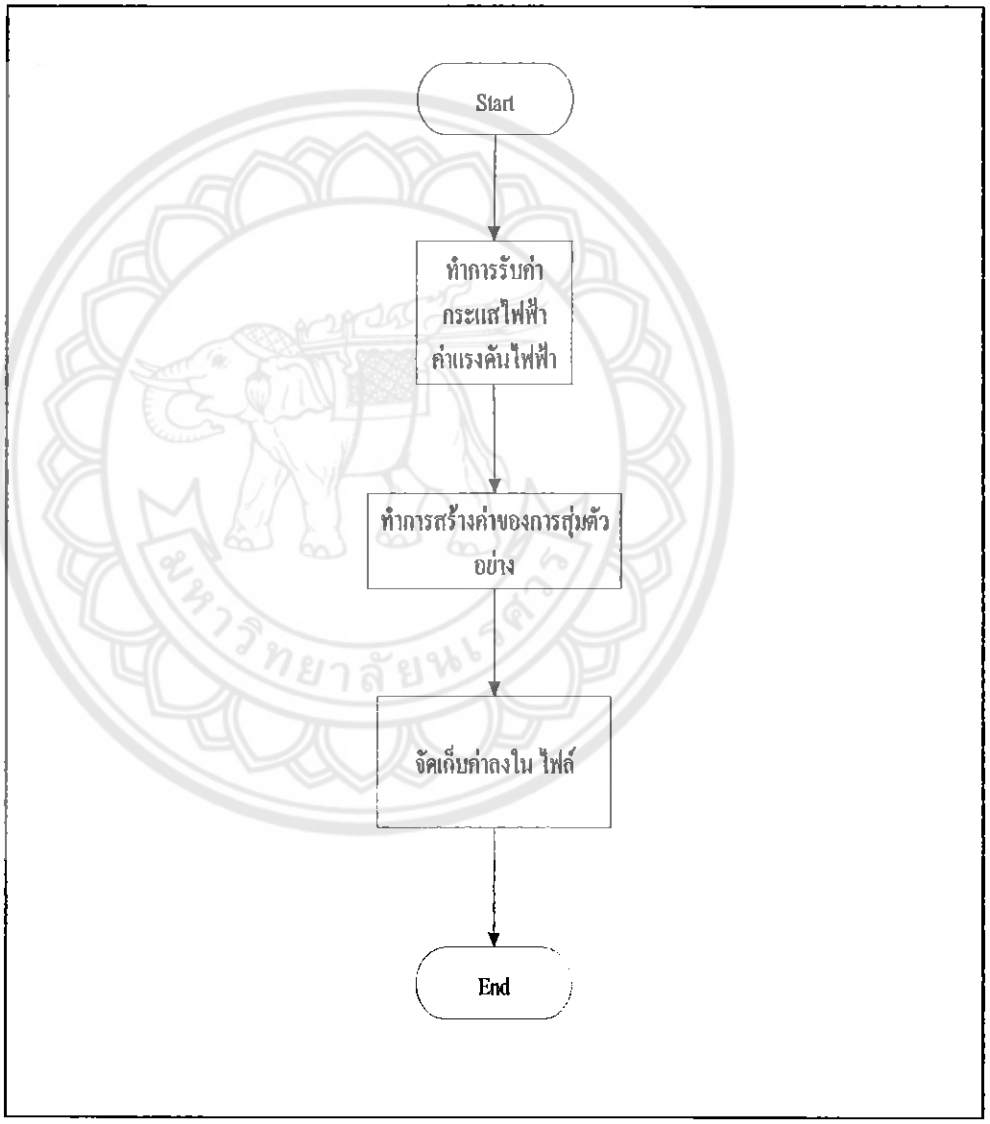
ในโปรแกรมจำลองการคำนวณค่าองค์ประกอบทางไฟฟ้านี้พัฒนาขึ้นภายใต้ข้อจำกัดของการทำงานร่วมกับอุปกรณ์ภายนอกซึ่งจะติดตั้งผ่านพอร์ท อนุกรม และนำค่าเข้ามาประมวลผลโดยในการจำลองนี้ได้จำลองค่าของ ความต่างศักย์ไฟฟ้า และ กระแสไฟฟ้าที่เข้ามาจากอุปกรณ์ภายนอกเป็นระบบไฟล์ เมตริกซ์ ซึ่งจะเท่ากับจำนวนการสุ่มตัวอย่าง โดยมีค่าเวลาการสุ่ม 2msโดยที่ค่าความถี่ของกระแสไฟฟ้า มีค่า ความถี่ 50 ครั้ง ใน 1 วินาที จะทำให้เราได้จำนานการสุ่มตัวอย่าง 500 ค่า ใน ค่าความต่างศักย์และค่ากระแสไฟฟ้า จากนั้นเราจะนำค่าเข้ามาประมวลผลในโปรแกรม



รูปที่ 3.1 แผนผังการทำงานของโปรแกรมทั้งหมด

3.1.1 โปรแกรมการจำลองค่าในการสุ่มตัวอย่าง

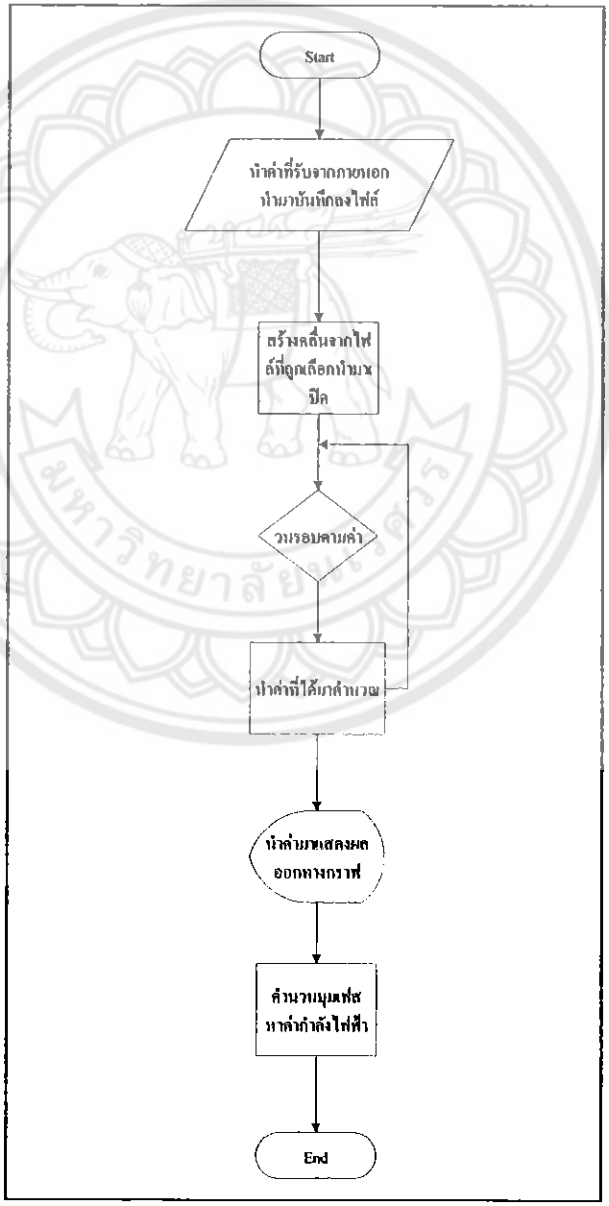
ในโปรแกรมจำลองค่านั้นถูกจัดทำโดยการพัฒนาด้วยโปรแกรมแมทแล็บ ซึ่งจัดทำในรูปแบบที่ง่ายต่อการใช้งานโดยผู้ใช้ทำการใส่ข้อมูล ค่าแอมพิจูดของค่าความต่างศักย์ ค่าแอมพิจูดของค่ากระแสไฟฟ้า ค่ามุมเฟสของค่าความต่างศักย์ และ ค่ามุมเฟสของค่ากระแสไฟฟ้า แล้วโปรแกรมจะทำการสร้างค่าการสุ่มตัวอย่างที่ต้องการให้อยู่ในรูปของสัญญาณ ซายน์ ที่ค่าความถี่ 50 คาบ ในเวลา 1 วินาที ซึ่งจะมีตัวอย่างในการสุมที่ 500ค่าใน 1 วินาที จากนั้นโปรแกรมจะทำการเก็บลงในไฟล์ เพื่อที่จะใช้ในการคำนวณต่อไป



รูปที่ 3.2 แผนผังการทำงานของโปรแกรมการจำลองค่าในการสุ่มตัวอย่าง

3.1.2 โปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า

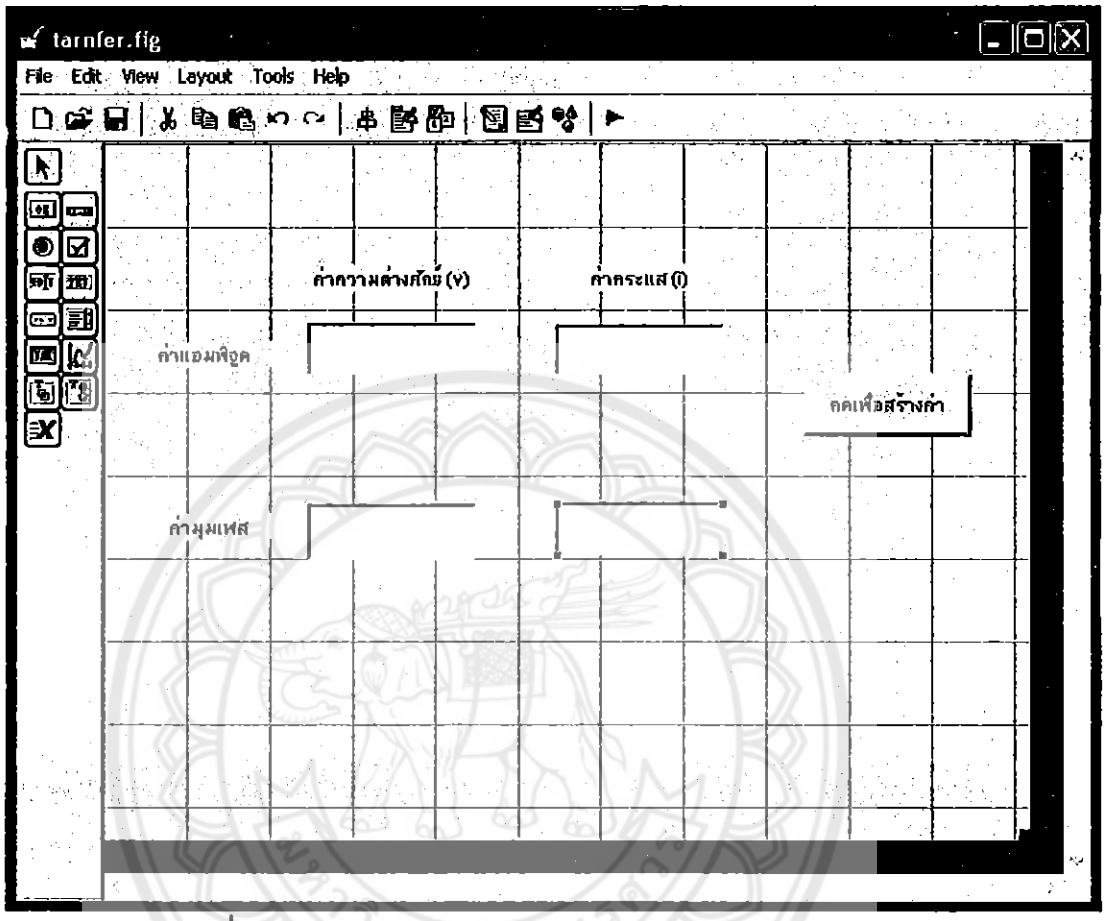
ในโปรแกรมการจำลองหาค่าองค์ประกอบทางไฟฟ้าถูกจัดทำโดยการพัฒนาด้วยโปรแกรมแมทแล็บ จัดทำขึ้นเพื่อให้อยู่ในรูปแบบที่ง่ายต่อการใช้งานในรูปแบบของ Graphic User Interface (GUI) โดยผู้ใช้สามารถใช้งานได้โดยการกดปุ่มเพียงปุ่มเดียวในการเปิดไฟล์และทดสอบค่าโดยโปรแกรมจะทำการรับค่า จากไฟล์ ที่ผู้ใช้ได้ทำการกำหนดค่าไว้จากโปรแกรมการจำลองค่า แล้วนำค่าที่ได้นำมาคำนวณหาค่า ของมุมเฟส ค่ากระแสไฟฟ้า ค่าความต่างศักย์ไฟฟ้า จากนั้นโปรแกรมจะแสดงกราฟที่แสดง ค่ากระแสไฟฟ้ากับค่าความต่างศักย์ ขั้นตอนต่อไป จะทำการคำนวณมุมเฟส และค่ากำลังไฟฟ้า ในขั้นตอนสุดท้าย จะทำการแสดงผลของค่าที่ได้จากการคำนวณ ขึ้นมาที่หน้าจอคอมพิวเตอร์



รูปที่ 3.3 แผนผังการทำงานของโปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า

3.2 สร้างโปรแกรม GUI

3.2.1 โปรแกรมจำลองค่าในการสุ่มตัวอย่าง



รูปที่3.4 หน้าจอการออกแบบโปรแกรมจำลองค่าในการสุ่มตัวอย่าง

การออกแบบโปรแกรมนี้ใช้ Matlab GUL ในการออกแบบโดยใช้ส่วนประกอบดั้งนี้

Text1 ใช้แสดงข้อความ "ค่าความต่างศักย์ V"

Text2 ใช้แสดงข้อความ "ค่ากระแส I"

Text3 ใช้แสดงข้อความ "ค่าแอมพิจูด V"

Text4 ใช้แสดงข้อความ "ค่ามุมเฟส"

Edit1 กล่องรับข้อความแอมพิจูด V

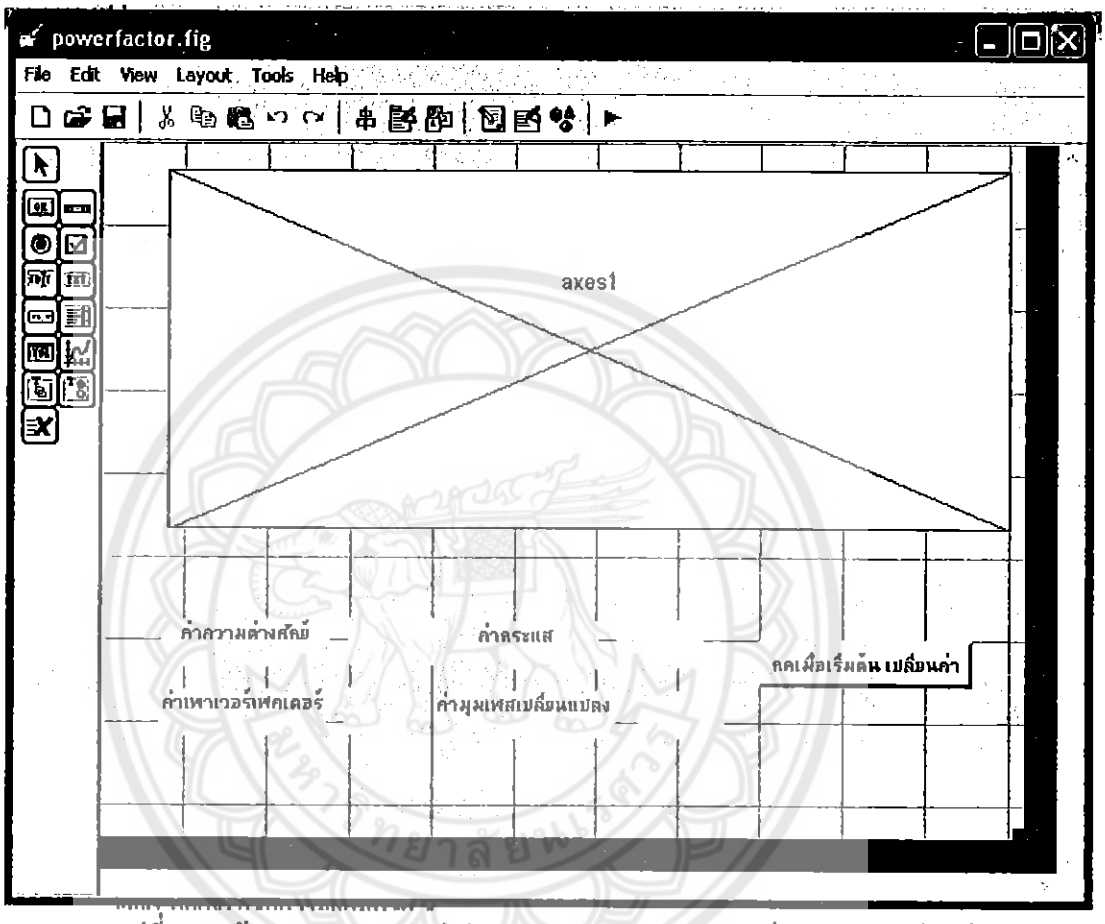
Edit2 กล่องรับมุมเฟส V

Edit3 กล่องรับข้อความแอมพิจูด I

Edit4 กล่องรับมุมเฟส I

Pushbutton1 = แสดงข้อความ "กดเพื่อสร้างค่า"และใช้สำหรับกดเพื่อเริ่มต้นการทำงานของโปรแกรม

3.2.2 โปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า



รูปที่3.5 หน้าจอการออกแบบโปรแกรมจำลองการหาค่าองค์ประกอบทางไฟฟ้า

การออกแบบโปรแกรมนี้ใช้ Matlab GUL ในการออกแบบโดยใช้ส่วนประกอบดั้งนี้

Axes1 ใช้แสดงกราฟความต่างศักย์ V กับกระแส I

Text7 ใช้แสดงค่าเพาเวอร์เฟคเตอร์

Text8 ใช้แสดงค่ามุมเฟสเปลี่ยนแปลง

Text9 ใช้แสดงข้อความ "ค่าเพาเวอร์เฟคเตอร์"

Text10 ใช้แสดงข้อความ "ค่ามุมเฟสเปลี่ยนแปลง"

Text11 ใช้แสดงค่าความต่างศักย์ V

Text12 ใช้แสดงข้อความ "ค่าความต่างศักย์ V"

Text13 ใช้แสดงค่ากระแส I

Text14 ใช้แสดงข้อความ "ค่ากระแส I"

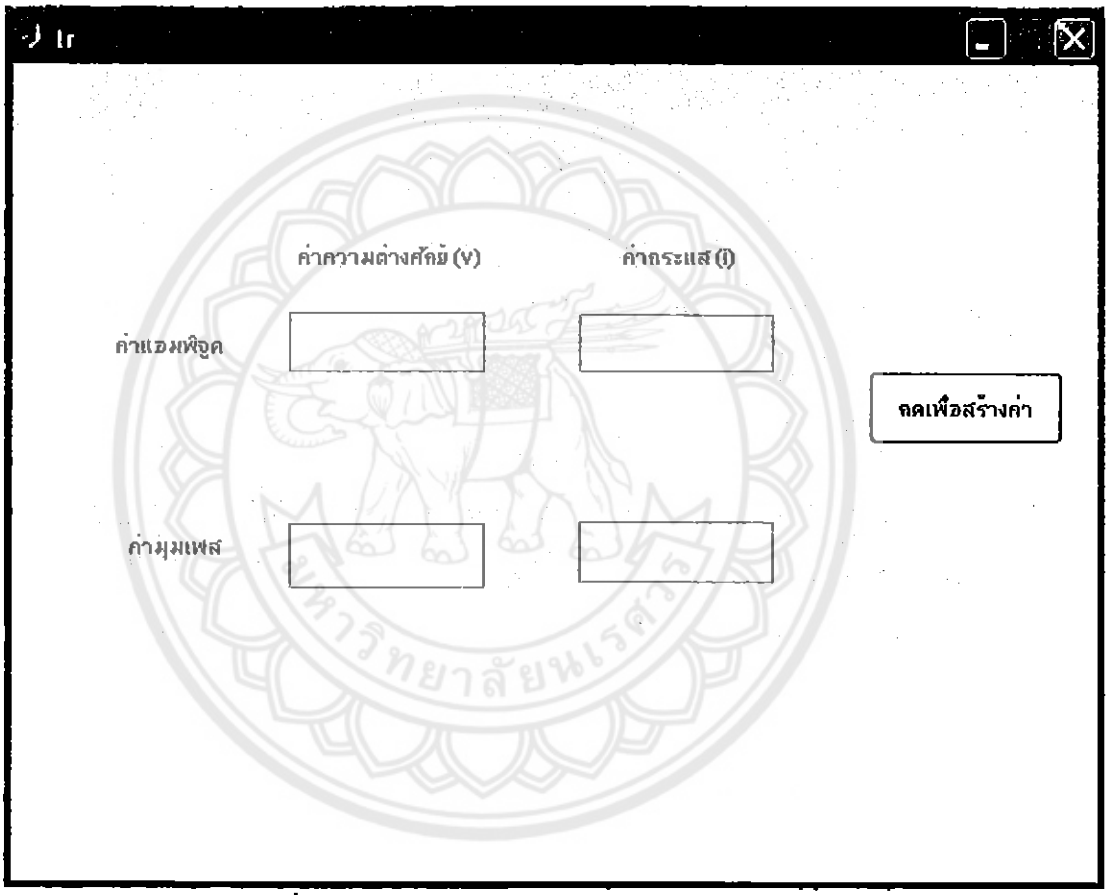
Pushbutton1 = แสดงข้อความ "กดเมื่อเริ่มต้น เปลี่ยนค่า"และใช้สำหรับกดเพื่อเริ่มต้นการทำงานของโปรแกรม

บทที่ 4
ผลการทดลองและผลการวิเคราะห์

4.1 ผลการทดลอง

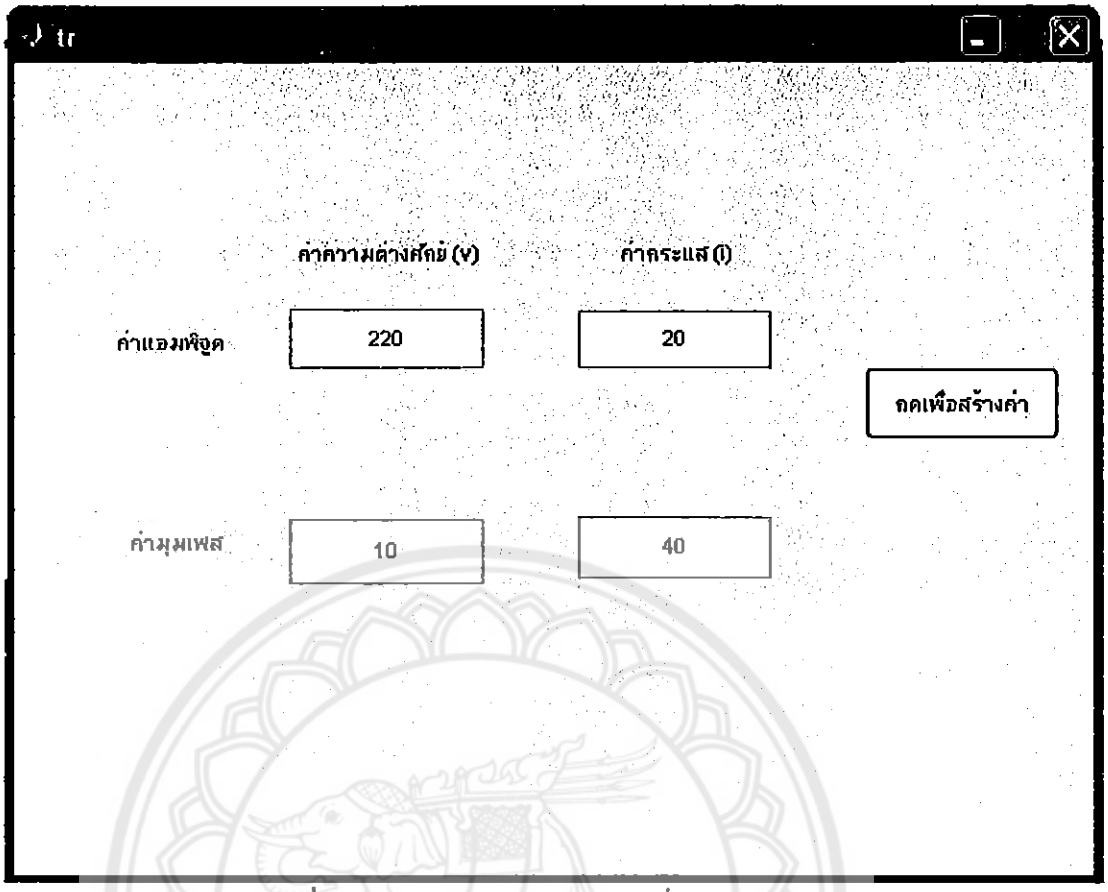
4.1.1 การทดลองในการใช้โปรแกรมสุ่มค่าตัวอย่าง

โปรแกรมการสุ่มค่าตัวอย่าง โดยเปิดได้จากไฟล์ tr.fig ดังรูปที่ 4.1



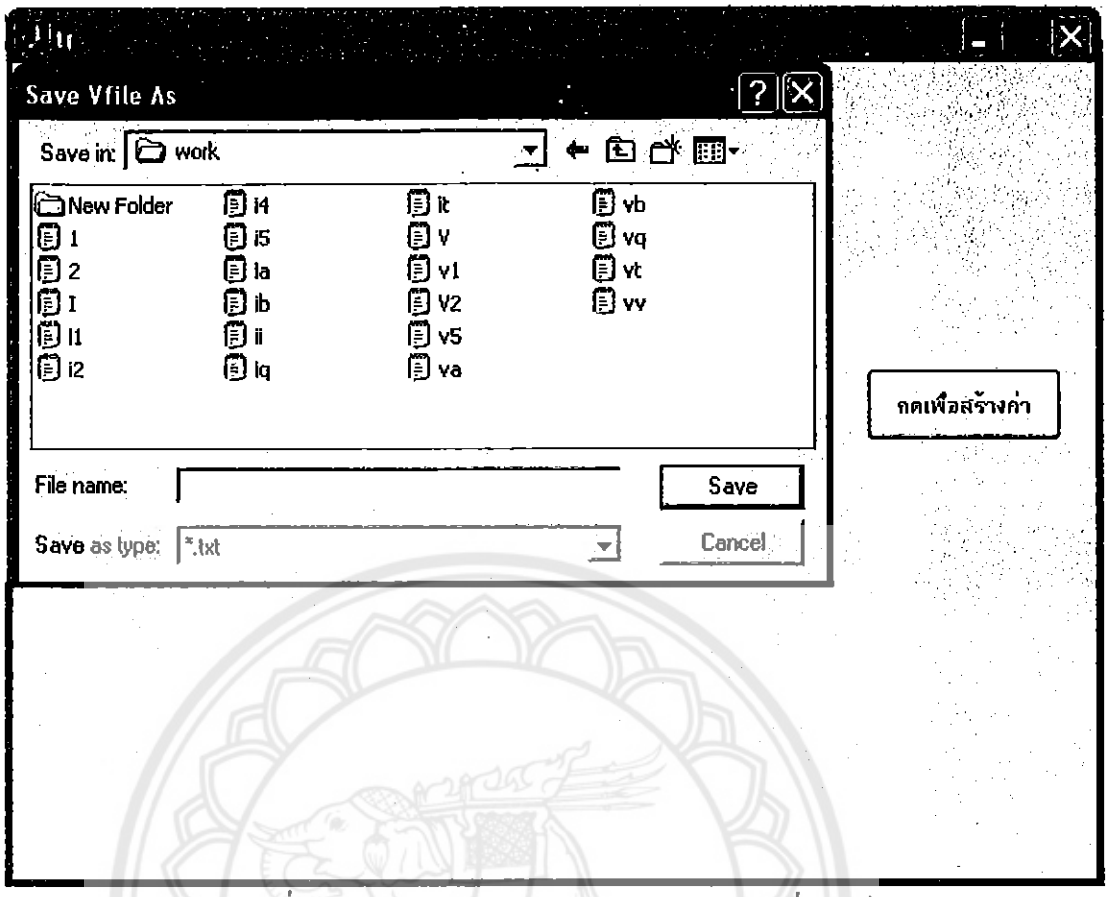
รูปที่ 4.1 การทำงานของโปรแกรมในการสุ่มค่าตัวอย่าง

จากโปรแกรมที่ทำการเปิดขึ้นมาจะปรากฏหน้าต่างการใส่ค่า ดังรูปที่ 4.1 เพื่อให้ผู้ใช้ได้ทำการกรอก ค่าความต่างศักย์ไฟฟ้า ค่ากระแสไฟฟ้า ค่ามุมเฟสของความต่างศักย์ไฟฟ้า และค่ามุมเฟสของกระแสไฟฟ้า เมื่อได้ทำการใส่ค่าครบทุกช่องแล้ว ให้ผู้ใช้กดที่ ปุ่ม "Push Button" เพื่อที่ทำการบันทึกลงไฟล์ต่อไป ดังรูปที่ 4.2



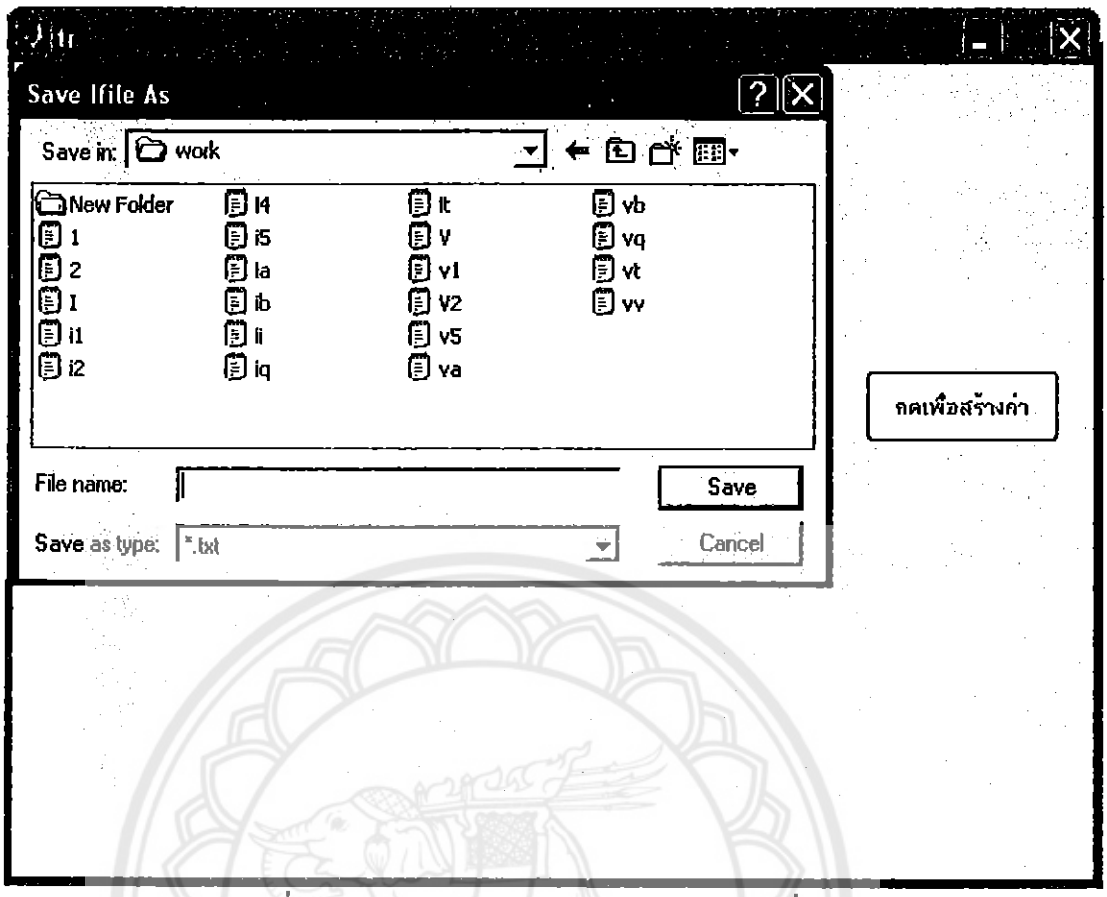
รูปที่ 4.2 การทำงานของโปรแกรมเมื่อผู้ใช้ใส่ค่าแล้ว

เมื่อทำการกดปุ่ม "Push Button" โปรแกรมก็จะทำการบันทึกข้อมูลลงในไฟล์ โปรแกรมจะให้ผู้ใช้ได้ทำการกำหนดชื่อไฟล์ที่ต้องการบันทึก โดยที่ชื่อไฟล์ที่ผู้ใช้จะต้องบันทึกนั้น โปรแกรมจะให้ผู้ใช้กำหนดชื่อไฟล์ ที่เป็นค่าความต่างศักย์ไฟฟ้าจากนั้นให้ผู้ใช้ กดปุ่ม "save" เพื่อที่โปรแกรมจะได้ทำการบันทึกไฟล์ต่อไป ดังรูปที่ 4.3



รูปที่ 4.3 การทำงานของโปรแกรมให้ผู้ใช้กำหนดชื่อไฟล์

เมื่อผู้ใช้ได้ทำการกดปุ่ม "save" แล้วโปรแกรมจะให้ผู้ใช้ได้ทำการกำหนดชื่อไฟล์ในส่วนของค่ากระแสไฟฟ้า หลังจากที่ผู้ใช้กำหนดชื่อไฟล์แล้วให้ผู้ใช้ กดปุ่ม "save" อีกครั้งหนึ่ง ดังรูป 4.4

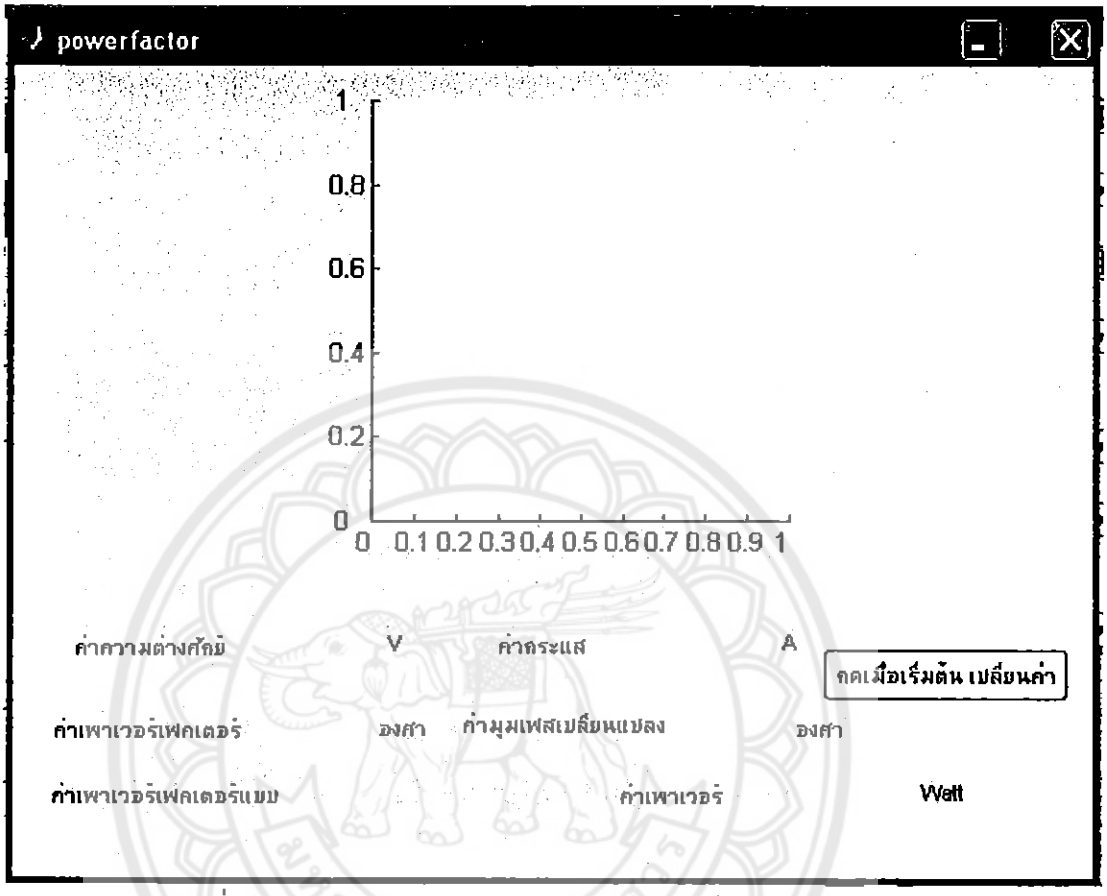


รูปที่ 4.4 การทำงานของโปรแกรมให้ผู้ใช้กำหนดชื่อไฟล์

หลังจากที่ได้ทำการสร้างไฟล์ข้อมูลการสุ่มค่าแล้วไฟล์ข้อมูลก็จะถูกเก็บอยู่ในคอมพิวเตอร์ต่อไป เพื่อให้ผู้ใช้ได้ทำการเลือกใช้ต่อไปในโปรแกรม คำนวณค่าองค์ประกอบทางไฟฟ้า

4.1.2 การทดลองในการใช้โปรแกรมคำนวณค่าองค์ประกอบทางไฟฟ้า

โปรแกรมการสุ่มค่าตัวอย่าง โดยเปิดได้จากไฟล์ tr.fig ดังรูปที่ 4.5

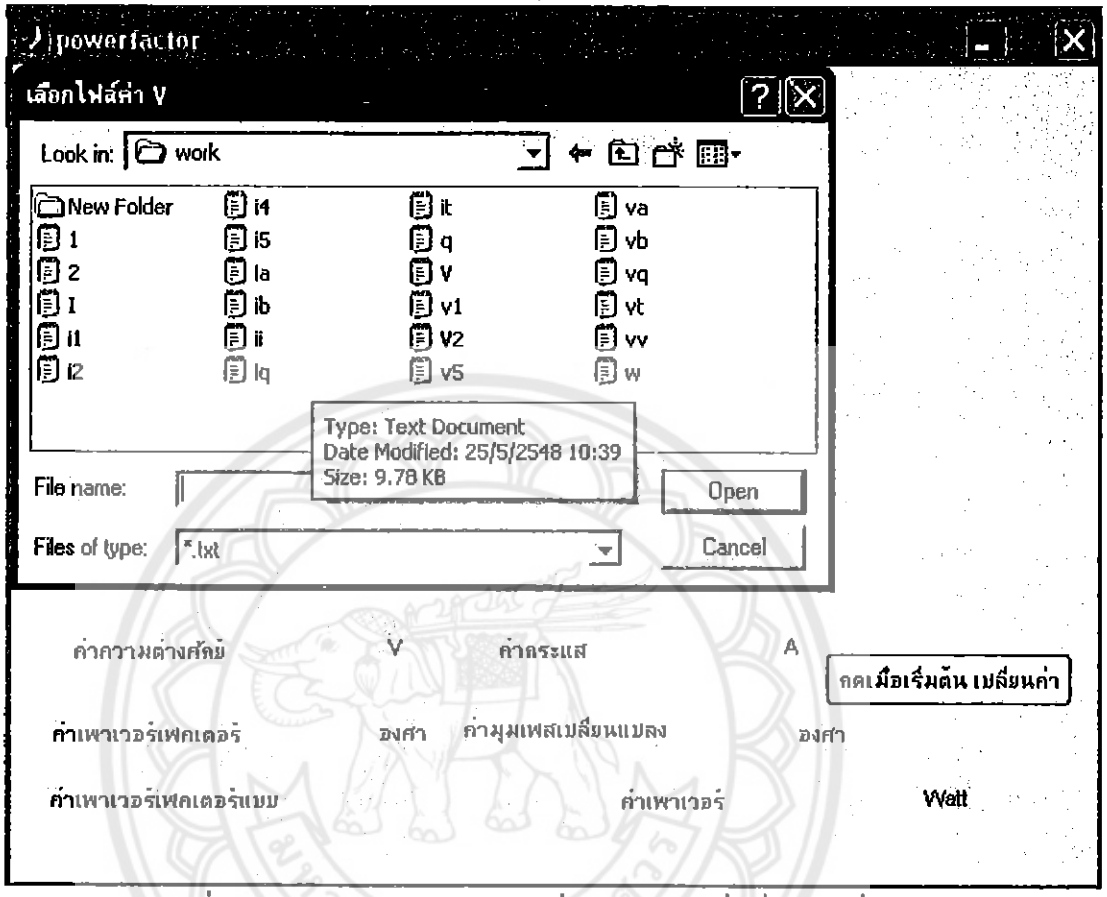


รูปที่ 4.5 การทำงานของโปรแกรมคำนวณค่าองค์ประกอบทางไฟฟ้า

เมื่อผู้ใช้ได้ทำการเปิดโปรแกรมคำนวณค่าองค์ประกอบทางไฟฟ้า ขึ้นมา ในส่วนของหน้าต่างโปรแกรม จะประกอบไปด้วย

- รูปกราฟ แสดงผลของโปรแกรม ที่ผู้ใช้ได้ทำการเปิดไฟล์ค่าความต่างศักย์ และไฟล์ค่ากระแสไฟฟ้า จากนั้นนำค่าในไฟล์ มาแสดงผลของค่าความต่างศักย์และค่ากระแสไฟฟ้า
- ค่าความต่างศักย์ไฟฟ้าที่โปรแกรมได้ทำการคำนวณ
- ค่ากระแสไฟฟ้าที่โปรแกรมได้ทำการคำนวณ
- ค่าเพาเวอร์เฟคเตอร์ที่โปรแกรมได้ทำการคำนวณ
- ค่ามุมเฟสระหว่างค่าความต่างศักย์ไฟฟ้ากับกระแสไฟฟ้า
- ค่าเพาเวอร์เฟคเตอร์แบบ Lagging หรือ ค่าเพาเวอร์เฟคเตอร์แบบ Leading
- ค่ากำลังไฟฟ้า

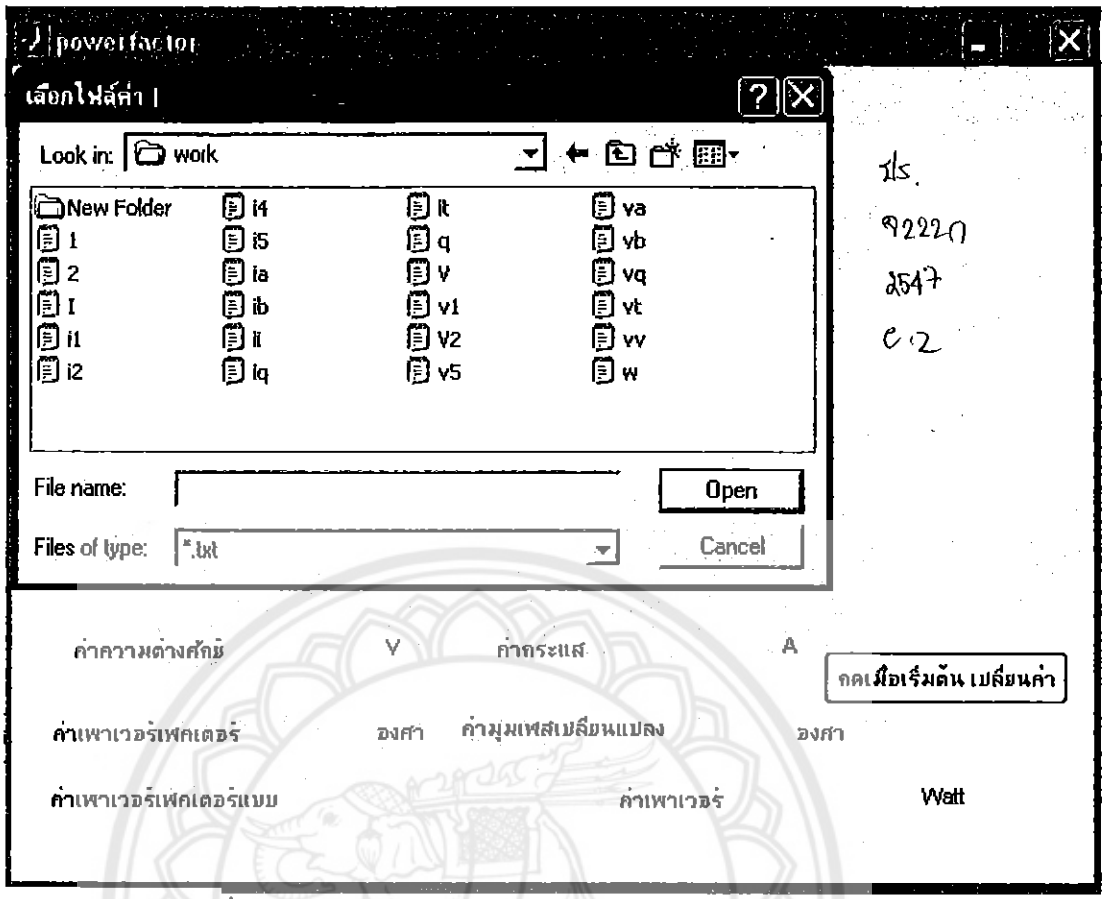
เมื่อผู้ใช้ได้ทำการกดปุ่ม "กดเมื่อเริ่มต้นเปลี่ยนค่า" โปรแกรมจะให้ผู้ใช้ได้ทำการใส่ไฟล์ข้อมูล ของค่าความต่างศักย์ไฟฟ้าเพื่อที่โปรแกรมจะได้นำมาคำนวณ ดังรูป 4.6



รูปที่ 4.6 การทำงานของโปรแกรมเมื่อกดปุ่ม "กดเมื่อเริ่มต้นเปลี่ยนค่า"

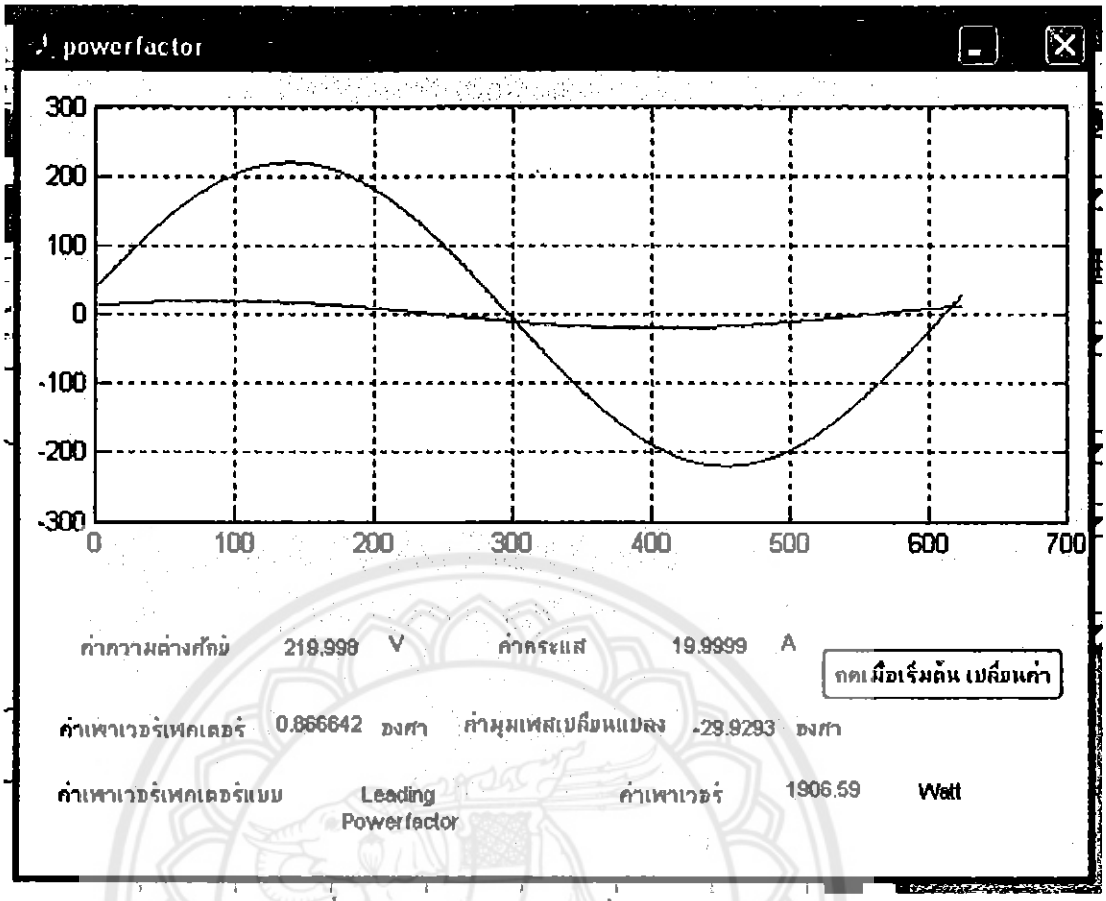
เมื่อผู้ใช้ได้ทำการใส่ชื่อไฟล์ค่าความต่างศักย์แล้วจากนั้นให้ผู้ใช้ กดปุ่ม "Open" โปรแกรมจะให้ผู้ใช้กำหนดไฟล์ค่ากระแสไฟฟ้า ดังรูป 4.7

ห้องสมุดคณะวิศวกรรมศาสตร์
4800034
I5067145



รูปที่ 4.7 การทำงานของโปรแกรมเพื่อให้กำหนดไฟล์กระแสไฟฟ้า

เมื่อผู้ใช้ได้ทำการใส่ชื่อไฟล์ค่ากระแสแล้วจากนั้นให้ผู้ใช้ กดปุ่ม "Open" โปรแกรมจะเริ่มการคำนวณ โดยการคำนวณจะนำค่าความต่างศักย์ไฟฟ้าและค่ากระแสไฟฟ้า จากไฟล์ที่ผู้ใช้ได้ทำการเลือก จากนั้น โปรแกรมจะนำค่าความต่างศักย์และค่ากระแสไฟฟ้ามาคำนวณมุมเฟสที่เปลี่ยนแปลง คำนวณค่าความต่างศักย์ คำนวณค่ากระแสไฟฟ้า คำนวนมุมเฟสที่เปลี่ยนไป คำนวณค่าเพาเวอร์เฟคเตอร์ คำนวนค่าเพาเวอร์เฟคเตอร์แบบ Leading หรือ Lagging คำนวณค่ากำลังไฟฟ้า ดังรูปที่ 4.8



รูปที่ 4.8 หน้าต่างโปรแกรมที่ทำการคำนวณแล้ว

จากการทดลอง ค่าที่ใช้ในการสุ่มตัวอย่างนั้นได้ทำการ สุ่มค่าตัวอย่างจากหลายค่าจากนั้นนำค่าที่ได้มาคำนวณแล้วทำการเปรียบเทียบระหว่างค่าการคำนวณ กับค่าที่ทำการสุ่มตัวอย่างข้อมูลซึ่งค่าที่คำนวณได้ ดังในตารางที่ 4.1

ตารางที่ 4.1 ค่าที่ได้จากการคำนวณในโปรแกรมเปรียบเทียบกับค่าการสุ่มตัวอย่าง

V_{Gen}	I_{Gen}	V_{Pro}	I_{Pro}	Phase V_{Gen}	Phase I_{Gen}	θ	Power factor	Power	Pf
220	20	219.9	20	50	90	-39.9	0.766	3370.51	Lead
220	20	220	19.9	0	50	-50	0.642	2828.2	Lead
220	50	220	49.9	0	50	-50.03	0.642	7070.4	Lead
220	50	219.9	49.9	50	10	40	0.765	8425.4	Lag
220	220	220	220	0	0	0	1	48400	Lag
220	220	219.9	220	10	0	10	0.984	47600	Lag
220	220	219.9	220	10	90	-79.9	0.173	8403.76	Lead

4.2 ผลการวิเคราะห์

จากผลการทดลองเราจะเห็นว่าค่าในตารางที่ 4.1 นั้น จากการทดลองโปรแกรม ได้ทำการทดลองจาก ไฟล์ค่าความต่างศักย์ และไฟล์ค่ากระแสไฟฟ้า จำนวน อย่างละ 7 ไฟล์ ในการสุ่มค่าการทดลอง เมื่อทำการคำนวณ ในโปรแกรม คำนวณหาค่าองค์ประกอบทางไฟฟ้า แล้วผลที่ได้ออกมาจากโปรแกรมการคำนวณค่าองค์ประกอบทางไฟฟ้า เป็นไปตามตาราง ที่ 4.1 นี้

ค่าความต่างศักย์ไฟฟ้าที่ได้จากการคำนวณ มีค่าใกล้เคียงกับค่าที่ได้ทำการสุ่มไว้ในไฟล์ของค่าความต่างศักย์ ส่วนค่ากระแสไฟฟ้าที่คำนวณได้นั้นมีค่าใกล้เคียง กับค่าที่ได้ทำการสุ่มไว้ในไฟล์ของค่ากระแสไฟฟ้า ค่ามุมเฟสที่เปลี่ยนไปที่เกิดขึ้นระหว่างค่าความต่างศักย์ไฟฟ้ากับค่ากระแสไฟฟ้า มีค่าใกล้เคียงกับค่าที่ทำการสุ่ม ค่าเพาเวอร์เฟคเตอร์ ที่ได้มีความใกล้เคียงกับการคำนวณเด้วยเครื่องคำนวณ ค่ากำลังไฟฟ้าที่ได้ มีค่าใกล้เคียงกับการคำนวณด้วยเครื่องคำนวณ ส่วนค่าเพาเวอร์เฟคเตอร์ว่าเป็นค่า Leading หรือ Lagging นั้นโปรแกรมก็สามารถที่จะแสดงได้ถูกต้อง

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

จากบทที่ 3 และ บทที่ 4 ทำให้ทราบว่าการวิเคราะห์องค์ประกอบกำลังไฟฟ้านั้นสามารถหาค่าได้จากหลายวิธีการคำนวณด้วยกัน และต่างก็มีขั้นตอนที่แตกต่างกัน แต่ผลสุดท้ายก็มีผลที่ใกล้เคียงกันมาก ส่วนจะมีผลที่เที่ยงตรงละเอียดแค่ไหนนั้นจะต่างกันที่การกำหนดการสุ่มวัดค่าต่อวินาที ว่ามีค่าที่ละเอียดแค่ไหนซึ่งจะถูกจำกัดด้วยฮาร์ดแวร์ในการทดลองนี้ มีการกำหนดค่าการสุ่มวัดค่าต่อวินาที เป็น 500ค่า ที่ความถี่ของกระแสไฟฟ้าเท่ากับ 50ค่าต่อวินาที และตัวโปรแกรมก็สามารถหาค่าองค์ประกอบกำลังไฟฟ้าได้อย่างถูกต้อง และสามารถหาค่าแอมพิจูดและค่ามุมเฟสแตกต่างของ ความต่างศักย์และกระแส จากสัญญาณที่เข้ามาได้อย่างถูกต้องได้ โปรแกรมที่เหมาะสมที่สุดกับการนำไปใช้งานจริง เนื่องจากเป็นโปรแกรมที่เขียนง่ายและใช้ทรัพยากรน้อย

ตารางที่ 5.1 ตารางแสดงผลการทดลองโปรแกรม

				ผลการทดลองจากโปรแกรม	
V	I	เฟส V	เฟส I	ค่าความแตกต่างมุมเฟส	ค่าองค์ประกอบกำลังไฟฟ้า
220	20	50	90	-39.9	0.766
220	20	0	50	-50	0.642
220	50	0	50	-50.03	0.642
220	220	50	10	40	0.765
220	220	0	0	0	1
220	220	10	0	10	0.984
220	220	10	90	-79.9	0.173

5.2 ปัญหาในการทดลองและแนวทางแก้ปัญหา

ปัญหา	แนวทางแก้ปัญหา
การกำหนดอัตตาการสุ่มค่าข้อมูล	จะต้องหาอุปกรณ์ที่มีค่าอัตตาการสุ่มค่าข้อมูลสูงพอที่จะให้ความละเอียดของข้อมูลได้มากพอ
การวิเคราะห์แบบค่าเวลาจริง	จะต้องพัฒนาให้สามารถทำงานแบบหมุนรอบและต้องลดการใช้ทรัพยากรและเวลาให้น้อยลง ในที่นี้ใช้การเก็บค่าเป็นรอบๆละ500ใน1วินาทีและประมวลผล

5.3 แนวทางในการพัฒนาโปรแกรมต่อไป

1. ควรเขียนโปรแกรมให้ใช้ทรัพยากรและเวลาการประมวลผลให้น้อยลง
2. ควรเขียนโปรแกรมให้สามารถให้สามารถต่อเชื่อมอุปกรณ์ได้จริงโดยใช้ฟังชั่น ซีเรียล โดยมีตัวอย่างดั้งนั้น

```
S1 = serial ('com2', BaudRate', 9600,'DataBit', 7)
```

หรือใช้ Simulink

เอกสารอ้างอิง

[1] ผศ.ดร.สุชาติ แย้มเม่น. เอกสารประกอบการสอนวิชาการวิเคราะห์วงจรไฟฟ้า .ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

[2] รศ.ดร.มนัส สังวรศิลป์. **คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์**. กรุงเทพมหานคร: อินโฟเพรส, 2543

โปรแกรมการสุ่มค่าตัวอย่าง

```
%Powerfactor Analysis Via Computer GenarateData Program
%นาย ณัฏฐ์ ภูวสันติ          รหัส   44370146
%นาย จักรพงศ์ สงเคราะห์      รหัส   44370088
%สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
%มหาวิทยาลัยนเรศวร ภาคการเรียนปลาย ปีการศึกษา2547
%โปรแกรมนี้เป็นส่วนหนึ่งของปริญญานิพนธ์เรื่อง powerfactor analysis via computer
function varargout = tr(varargin)
% TR M-file for tr.fig
%      TR, by itself, creates a new TR or raises the existing
%      singleton*.
%
%      H = TR returns the handle to a new TR or the handle to
%      the existing singleton*.
%
%      TR('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in TR.M with the given input arguments.
%
%      TR('Property','Value',...) creates a new TR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before tr_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to tr_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.
```

```
% Edit the above text to modify the response to help tr

% Last Modified by GUIDE v2.5 25-May-2005 10:13:58

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @tr_OpeningFcn, ...
                   'gui_OutputFcn',  @tr_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before tr is made visible.
function tr_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to tr (see VARARGIN)
```

```
% Choose default command line output for tr
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes tr wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = tr_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function edit _Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double
edit5 = str2double(get(hObject, 'String'));
if isnan(edit5)
   set(hObject, 'String', 0);
```

```
    errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit5 = edit5;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double

edit6 = str2double(get(hObject, 'String'));
```

```
if isnan(edit6)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit6 = edit6;
guidata(hObject,handles)
% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a double
edit7 = str2double(get(hObject, 'String'));
```

```
if isnan(edit7)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit7 = edit7;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a double
```

```
edit8 = str2double(get(hObject, 'String'));
if isnan(edit8)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit8 = edit8;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
t=[0:0.01:6.25];                    % time sampling
v=((handles.metricdata.edit5)*sin(t+((handles.metricdata.edit6)*pi/180)));          % Volt
```

```
i=((handles.metricdata.edit7)*sin(t+((handles.metricdata.edit8)*pi/180)));          % Current


[file,path] = uiputfile('*.txt','Save Vfile As');
[file1,path1] = uiputfile('*.txt','Save Ifile As');
save(fullfile(path, file), 'v','-ASCII')
save(fullfile(path1, file1), 'i','-ASCII')
```

โปรแกรมการคำนวณหาค่าองค์ประกอบทางไฟฟ้า

```
%Powerfartor Analysis Via Computer Simulation Program
%นาย ณัฏฐ์ ภูวสันติ          รหัส 44370146
%นาย จักรพงศ์ สงเคราะห์      รหัส 44370088
%สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
%มหาวิทยาลัยนเรศวร ภาคการเรียนปลาย ปีการศึกษา2547
%โปรแกรมนี้เป็นส่วนหนึ่งของปริญญานิพนธ์เรื่อง powerfactor analysis via computer
function varargout = powerfactor(varargin)
% POWERFACTOR M-file for powerfactor.fig
%      POWERFACTOR, by itself, creates a new POWERFACTOR or raises the existing
%      singleton*.
%
%      H = POWERFACTOR returns the handle to a new POWERFACTOR or the handle to
%      the existing singleton*.
%
%      POWERFACTOR('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in POWERFACTOR.M with the given input arguments.
%
%      POWERFACTOR('Property','Value',...) creates a new POWERFACTOR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before powerfactor_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to powerfactor_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.
```

```
% Edit the above text to modify the response to help powerfactor

% Last Modified by GUIDE v2.5 23-May-2005 10:48:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @powerfactor_OpeningFcn, ...
                   'gui_OutputFcn',  @powerfactor_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before powerfactor is made visible.
function powerfactor_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to powerfactor (see VARARGIN)
```

```
% Choose default command line output for powerfactor
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes powerfactor wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = powerfactor_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
edit1 = str2double(get(hObject, 'String'));
if isnan(edit1)
   set(hObject, 'String', 0);
```

```
    errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit1 = edit1;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
edit2 = str2double(get(hObject, 'String'));
if isnan(edit2)
```

```
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit2 = edit2;
guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double
```

```
edit3 = str2double(get(hObject, 'String'));
if isnan(edit3)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit3 = edit3;
guidata(hObject,handles)
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double
```

```
edit4 = str2double(get(hObject, 'String'));
if isnan(edit4)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end

% Save the new density value
handles.metricdata.edit4 = edit4;
guidata(hObject,handles)
% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
%clear all
%close all


t=[0:0.01:6.25];                    % time sampling
%v=(handles.metricdata.edit1*sin(t+(handles.metricdata.edit2*pi/180)));          % Volt
%i=(handles.metricdata.edit3*sin(t+(handles.metricdata.edit4*pi/180)));          % Current
[filename, pathname] = uigetfile('*.txt', 'เลือกไฟล์ค่า V');
[filename1, pathname1] = uigetfile('*.txt', 'เลือกไฟล์ค่า I');
i = load(fullfile(pathname1, filename1));
v = load(fullfile(pathname, filename));
%figure(1);

axes(handles.axes1);
cla;
plot(v,'r');
hold on
plot(i,'m');


fftv=fft(v);
ffti=fft(i);
vp=rad2deg(angle(v));
ip=rad2deg(angle(i));
pf=(angle(fftv(2))-angle(ffti(2)));
grid on
A=rad2deg(pf)
pf2=cos(pf)
set(handles.text7, 'String', pf2);
set(handles.text8, 'String', A);
l=max(v);
l1=max(i);
```

```
set(handles.text11, 'String', l);
set(handles.text13, 'String', l1);
if A>=0
   set(handles.text15, 'String', 'Lagging Powerfactor');%แสดงแบบLagging Powerfactor
   else
   set(handles.text15, 'String', 'Leading Powerfactor');%แสดงแบบLeading Powerfactor
end
pw=(((l1*l)/2)*(pf2))
set(handles.text17, 'String', pw)
%f1=(angle(v))
%f2=rad2deg(angle(v))
%f11=(angle(i))
%f21=rad2deg(angle(i))
% มุมที่เปลี่ยนเป็นองศาแล้ว
```

ประวัติผู้เขียนโครงงาน

ชื่อ นามสกุล จักรพงศ์ สงเคราะห์
เกิดวันที่ 14 กุมภาพันธ์ 2525
ภูมิลำเนา 401/184 ถนน มิตรภาพ ตำบล ในเมือง อำเภอ เมือง จังหวัด พิษณุโลก

ประวัติการศึกษา

- จบการศึกษาชั้นมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม อำเภอเมือง จังหวัด พิษณุโลก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-Mail: aue_chak@hotmail.com

ชื่อ นามสกุล ณัฏฐ์ ภูวสันติ
เกิดวันที่ 20 กุมภาพันธ์ 2526
ภูมิลำเนา 156/4 หมู่ 3 ซอย ซี3 ถนน รามคำแหง สะพานสูง จังหวัด กรุงเทพมหานคร

ประวัติการศึกษา

- จบการศึกษาชั้นมัธยมศึกษาจากโรงเรียนนวมินทราชินูทิศ เตรียมอุดมศึกษาน้อมเกล้า สะพานสูง จังหวัด กรุงเทพมหานคร
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-Mail: nat_cpe@yahoo.com