โปรแกรมช่วยสอนการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำที่สุด

ECONOMIC DISPATCH SIMULATOR PROGRAM

นาย เจษฎา สร้อยแหยม

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขา วิศวกรรมไฟฟ้า ภาควิชา วิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีปทุม

ปีการศึกษา 2552

51EE202

หัวข้อโครงงาน	โปรแกรมช่วยสอนการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำที่สุด
โดย	นาย เจษฎา สร้อยแหยม
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	ผศ.ดร.กีรติ ชยะกุลคีรี

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีปทุม อนุมัติให้นับโครงงานวิศวกรรมฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

.....หัวหน้าภาควิชาวิศวกรรมไฟฟ้า

(ดร.นิมิต บุญภิรมย์)

..... อาจารย์ที่ปรึกษา

(ผศ.ดร.กีรติ ชยะกุลคีรี)

(วันที่.....เดือน.....พ.ศ.2552)

รหัสโครงงาน 51EE202

โปรแกรมช่วยสอนการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำที่สุด

ECONOMIC DISPATCH SIMULATOR PROGRAM

บทคัดย่อ (Abstract)

โครงงานนี้เป็นการพัฒนาโปรแกรมช่วยสอนการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำที่สุด เพื่อให้เกิดความสะดวกในการเรียนการสอนในเรื่องการหาคำตอบการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตแบบควอดราติค ทั้งนี้ในการคำนวณในวิชาเรียนค่อนข้างที่จะใช้เวลาในการ คำนวณในแต่ละครั้งจึงได้จัดทำโปรแกรมนี้ขึ้นเพื่อสะดวกและเข้าใจต่อนักศึกษาในการเรียนทั้งโปรแกรมนี้ยังมีรูปแบบในการใส่ค่า แสดงค่าเป็นแบบกราฟและตาราง

กิตติกรรมประกาศ

การทำโครงงานนี้สามารถสำเร็จผลลุล่วงไปด้วยดี ต้องขอขอบคุณคณะอาจารย์ที่ให้คำปรึกษา อันได้แก่ ผศ.ดร.กีรติ ชยะกุลคีรี ซึ่งได้ช่วยให้คำปรึกษาและแนะนำข้อมูลอันเป็นประโยชน์รวมทั้งแนวทางและข้อคิดต่างๆ ได้ด้วยดีตลอดมา

ขอขอบคุณคณะกรรมการทุกท่านที่ให้โอกาสผู้จัดทำไดมีโอกาสศึกษาและทำโครงงานนี้ขึ้นมาจนสำเร็จลุล่วงได้ด้วยดี รวมถึงครอบครัว พ่อแม่ ที่ให้ความสนับสนุนตลอดมา สิ่งใดที่โครงงานนี้มีความผิดพลาด ผู้จัดทำจะขอรับผิดชอบแต่เพียงผู้เดียว ส่วนความดีความชอบทั้งหลายผู้จัดทำขอมอบให้กับผู้สนับสนุนโครงงานนี้ทุกๆท่าน

เจษฎา สร้อยแหยม

พ.ศ. 2552

สารบัญ

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

สารบัญภาพ(ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและสำคัญของปัญหา

การสั่งเดินเครื่องโรงไฟฟ้าให้สอดคล้องกับภาระในระบบโดยที่มีต้นทุนการผลิตต่ำถือว่ามีความสำคัญอย่างมากในการบริหารการจัดการในระบบไฟฟ้ากำลัง ทั้งนี้เนื้อเกี่ยวข้องมีค่อนข้างมากจึงควรมีเครื่องมือที่ช่วยในการเรียนการสอนให้นักศึกษาทำความเข้าใจได้ชัดเจนและรวดเร็ว

1.2 วัตถุประสงค์ของโครงงาน

- เพื่อพัฒนาโปรแกรมสำเร็จรูปในการสอนเรื่องการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำสุด โดยแสดงผลเป็นกราฟ
- เพื่อให้นักศึกษามีความเข้าใจการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำสุดได้ง่ายขึ้น
- เพื่อเป็นแนวทางการพัฒนาโปรแกรมช่วยสอนในเรื่องอื่นๆต่อไป

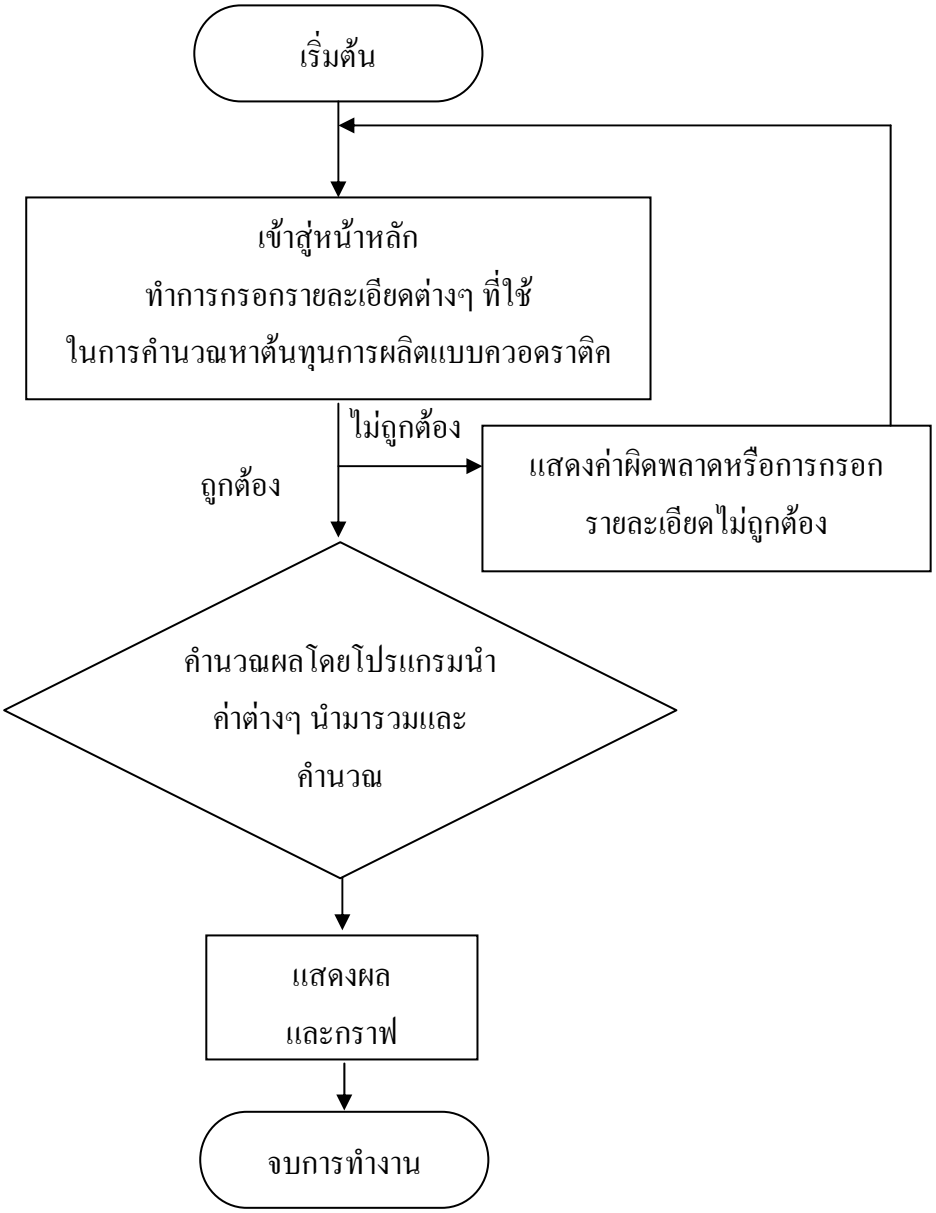
1.3 ขอบเขตของโครงงาน

- พัฒนาโปรแกรมสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำสุดโดยมีการแสดงผล แบบกราฟ และมีขั้นตอนการอธิบายโดยละเอียด
- โปรแกรมสามารถหาคำตอบการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตแบบควอดราติค (Quadratic)
- โปรแกรมมีการติดต่อกับผู้ใช้สามารถใส่ข้อมูล บันทึกข้อมูลรวมทั้งมีข้อมูลตัวอย่าง

1.4 ประโยชน์ของโครงงาน

- ได้โปรแกรมสำเร็จรูปในการสอนเรื่องการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำสุดโดยแสดงผลเป็นกราฟและตาราง
- ช่วยให้นักศึกษามีความเข้าใจการสั่งเดินเครื่องโรงไฟฟ้าที่มีต้นทุนการผลิตต่ำสุดได้ง่าย
- เป็นแนวทางการพัฒนาโปรแกรมช่วยสอนในเรื่องอื่นๆต่อไป

1.5 โครงสร้างของโครงงาน



ภาพที่ 1.1 โครงสร้างของโครงงาน

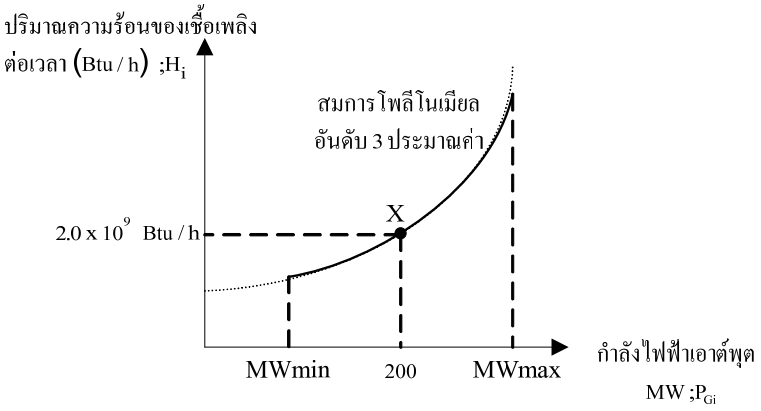
บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 การทำงานเชิงเศรษฐศาสตร์ของระบบไฟฟ้ากำลัง [1] [2] [3] [4]

การทำงานเชิงเศรษฐศาสตร์ของระบบไฟฟ้ากำลังจะเกี่ยวข้องกับการผลิตไฟฟ้าของโรงจักรที่ต้องใช้ เชื้อเพลิง และการส่งจ่ายกำลังไฟฟ้าผ่านทางสายส่ง สามารถแบ่งออกเป็น 2 ส่วนคือ การส่งจ่ายกำลังไฟฟ้าไปยังโหลดผู้ใช้ไฟที่เกี่ยวข้องกับการลดค่าใช้จ่ายในการผลิตไฟฟ้ารวมให้น้อยที่สุดที่เรียกว่า การจ่ายโหลดอย่างประหยัด (Economic dispatch) และ การส่งจ่ายกำลังไฟฟ้าไปยังโหลดผู้ใช้ไฟที่เกี่ยวข้องกับการลดค่าความสูญเสียให้น้อยที่สุดที่เรียกว่า ความสูญเสียต่ำสุด (Minimum loss)

ในการวิเคราะห์การจ่ายโหลดอย่างประหยัดยังสามารถแบ่งการวิเคราะห์ออกเป็นสองส่วน คือ การจ่ายโหลดอย่างประหยัดภายในโรงจักรที่ไม่คิดค่าความสูญเสียของสายส่ง (Economic dispatch neglecting losses) และการจ่ายโหลดอย่างประหยัดระหว่างโรงจักรที่คิดค่าความสูญเสียของสายส่งที่เชื่อมต่อระหว่างโรงจักร (Economic dispatch including losses)

การจ่ายโหลดอย่างประหยัดของเครื่องกำเนิดไฟฟ้าภายในโรงจักรที่ต้องใช้เชื้อเพลิง ต้องทราบความสัมพันธ์ระหว่างเชื้อเพลิงและกำลังไฟฟ้าเอาต์พุตที่ได้ ซึ่งมีหลายรูปแบบ เช่นอัตราความร้อน อัตราการสิ้นเปลืองเชื้อเพลิง อัตราค่าใช้จ่ายเชื้อเพลิง อัตราค่าใช้จ่ายในการผลิต เริ่มจากการพิจารณากราฟอินพุต- เอาต์พุต ดังแสดงในรูปที่ 2.1 ซึ่งเป็นความสัมพันธ์ระหว่างปริมาณความร้อนของเชื้อเพลิงต่อเวลากับกำลังไฟฟ้าที่ผลิตได้



ภาพที่ 2.1 กราฟอินพุต-เอาต์พุต

ความสัมพันธ์ระหว่างกราฟอินพุต-เอาต์พุต โดยทั่วไปสามารถประมาณค่าด้วยสมการโพลีโนเมียลอันดับสามในรูป $H_i = aP_{Gi}^3 + bP_{Gi}^2 + cP_{Gi} + d$

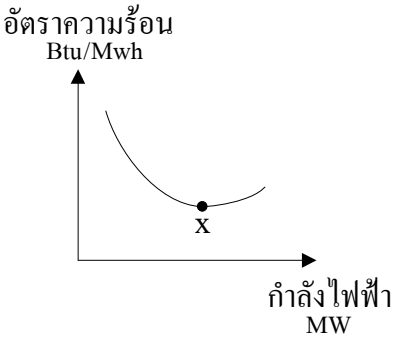
อัตราส่วนของอินพุตที่เป็นปริมาณความร้อนของเชื้อเพลิงต่อเวลา (Btu/h) ต่อกำลังไฟฟ้าเอาต์พุตที่ผลิตได้ (MW) เรียกว่า อัตราความร้อน (Heat rate) และส่วนกลับของอัตราความร้อนก็คือค่าประสิทธิภาพของเชื้อเพลิง (Fuel efficient) ดังนั้นถ้าอัตราความร้อนมีค่าต่ำ ประสิทธิภาพของเชื้อเพลิงก็จะมีค่าสูง ถ้าลากเส้นจากจุดศูนย์มาสัมผัสกราฟอินพุต-เอาต์พุตที่จุด X เป็นจุดที่มีค่าความชันต่ำสุด จะได้ค่าอัตราความร้อนที่ต่ำสุด จากรูปที่2.1 ความชันของเส้นกราฟจากจุดศูนย์ไปยังจุดบนกราฟต่ำสุดที่จุด X ดังนั้นที่จุด X จะมีประสิทธิภาพของเชื้อเพลิงสูงสุดที่กำลังไฟฟ้าเอาต์พุต 200MW ซึ่งต้องการปริมาณความร้อนของเชื้อเพลิง 2.0×10^9 Btu/h จะได้อัตราความร้อนเท่ากับ 10,000 Btu/kWh แต่ 1KWh = 3413 Btu

$$\therefore \text{ประสิทธิภาพ} = 34.13\ \%$$

$$\text{อัตราความร้อน (Heat rate)} = \frac{H_i}{P_{Gi}}\left(\frac{\text{Btu}}{\text{MWh}}\right)$$

$$\text{ประสิทธิภาพของเชื้อเพลิง (Fuel efficient)} = \frac{1}{\text{อัตราความร้อน}} = \frac{P_{Gi}}{H_i}\left(\frac{\text{MWh}}{\text{Btu}}\right)$$

เมื่อพล็อตกราฟของอัตราความร้อนเทียบกับค่ากำลังไฟฟ้าจะได้ดังรูปที่ 2.2



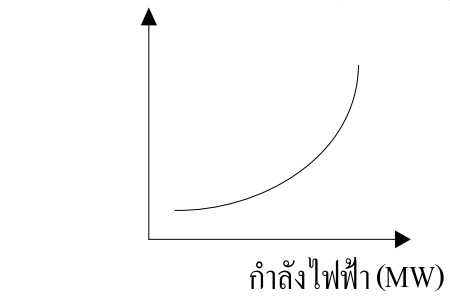
ภาพที่ 2.2 กราฟของอัตราความร้อนเทียบกับกำลังไฟฟ้า

จากกราฟรูปที่ 2.2 เป็นกราฟความสัมพันธ์ระหว่างอัตราความร้อน $\left(\frac{H_i}{P_{Gi}}\right)$ กับกำลังไฟฟ้าที่ผลิตได้ $\left(P_{Gi}\right)$ ซึ่งจุดต่ำสุดของกราฟก็คือจุด X ในรูปที่ 2.1 นั่นเองที่ให้ค่าอัตราความร้อนต่ำที่สุด

อัตราความสิ้นเปลืองเชื้อเพลิง (Incremental fuel rate)

หมายถึง อัตราส่วนของปริมาณความร้อนของเชื้อเพลิงต่อเวลา ที่เปลี่ยนแปลง ต่อกำลังไฟฟ้าที่เปลี่ยนแปลงไปเล็กน้อย สามารถแสดงด้วยกราฟดังรูปที่ 2.3

$$\text{อัตราความสิ้นเปลืองเชื้อเพลิง} = \lim_{\Delta P_{Gi} \to 0} \frac{\Delta H_i}{\Delta P_{Gi}} = \frac{dH_i}{dP_{Gi}} \left(\text{ค่า slope ที่} P_{Gi} \text{ ใดๆ}\right) \left(\frac{\text{Btu}}{\text{MWh}}\right)$$



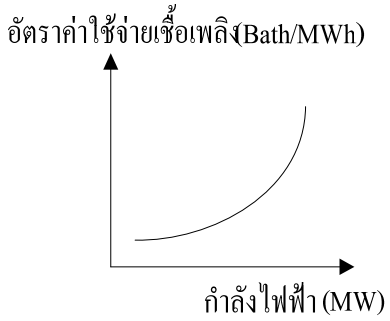
ภาพที่ 2.3 กราฟอัตราการสิ้นเปลืองเชื้อเพลิงเทียบกับค่ากำลังไฟฟ้า

อัตราค่าใช้จ่ายเชื้อเพลิง (Incremental fuel cost)

หมายถึง อัตราส่วนของค่าใช้จ่ายเชื้อเพลิงต่อเวลา ที่เปลี่ยนแปลง ต่อกำลังไฟฟ้าที่เปลี่ยนไปเล็กน้อย สามารถแสดงด้วยกราฟดังรูปที่ 2.4 ซึ่งก็คือ อัตราความสิ้นเปลืองเชื้อเพลิงคูณกับราคาเชื้อเพลิงนั่นเอง

ถ้ากำหนดให้ค่าใช้จ่ายเชื้อเพลิง F_i มีหน่วยเป็นบาทต่อชั่วโมง

$$\text{อัตราค่าใช้จ่ายเชื้อเพลิง} = \frac{dH_i}{dP_{Gi}} \times \text{ราคาเชื้อเพลิง}(\text{Bath}/\text{Btu}) = \frac{dF_i}{dP_{Gi}}\left(\frac{\text{Bath}}{\text{MWh}}\right)$$



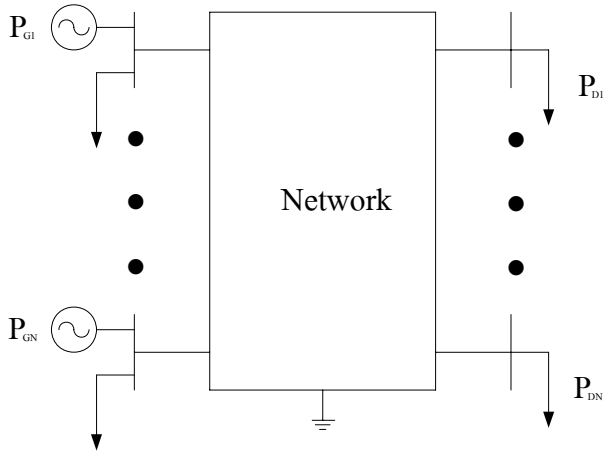
ภาพที่ 2.4 กราฟอัตราค่าใช้จ่ายเชื้อเพลิงเทียบกับค่ากำลังไฟฟ้า

อัตราค่าใช้จ่ายในการผลิต (Incremental production Cost)

หมายถึง อัตราส่วนของค่าใช้จ่ายเชื้อเพลิง + ค่าใช้จ่ายในการบำรุงรักษา + ค่าแรงและอื่นๆ ที่เปลี่ยนแปลงไป ต่อกำลังไฟฟ้าที่เปลี่ยนไปเล็กน้อย

2.2 การจ่ายโหลดอย่างประหยัดภายในโรงจักรโดยไม่คิดค่าความสูญเสียในระบบ

สมมุติว่าในโรงจักรมีเครื่องกำเนิดไฟฟ้า N หน่วยร่วมกันจ่ายโหลด ดังรูปที่ 2.5



ภาพที่ 2.5 เครื่องกำเนิดไฟฟ้า N หน่วยร่วมกันจ่ายโหลด

และกำหนดให้

P_{Gi} - กำลังไฟฟ้าในเครื่องกำเนิดไฟฟ้าหน่วยที่ i โดยมีขอบเขตในการจ่ายกำลังไฟฟ้าเอาต์พุตเป็น $P_{Gi(min)} \le P_{Gi} \le P_{Gi(max)}$

F_i - ค่าใช้จ่ายเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าหน่วยที่ i ต่อชั่วโมง

F_t - ค่าใช้จ่ายรวมของระบบต่อชั่วโมง

ดังนั้น
$$F_t = \sum_{i=1}^{N} F_i = F_1 + F_2 + ... + F_N \qquad (2.1)$$

การจ่ายโหลดอย่างประหยัดจะต้องทำให้ F_t มีค่าน้อยที่สุดและสอดคล้องกับการจ่ายโหลดของโรงจักร

$$\sum_{i=1}^{N}\left(P_{Gi} - P_{Di}\right) = 0 \Longrightarrow \sum_{i=1}^{N} P_{Gi} = \sum_{i=1}^{N} P_{Di} \qquad (2.2)$$

และขอบเขตในการจ่ายกำลังไฟฟ้าเป็น $P_{Gi(min)} \le P_{Gi} \le P_{Gi(max)}$

เนื่องจากค่าใช้จ่ายรวมของระบบ F_t เป็นฟังก์ชันของกำลังไฟฟ้า P_{Gi} ดังนั้นการจ่ายโหลดอย่างประหยัดจะได้ $dF_t = 0$ สำหรับโหลดคงที่ P_{Di}

$$dF_t = \frac{\partial F_1}{\partial P_{G1}} dP_{G1} + \frac{\partial F_2}{\partial P_{G2}} dP_{G2} + ... + \frac{\partial F_N}{\partial P_{GN}} dP_{GN} = 0 \quad (2.3)$$

จากสมการที่(2.2) เนื่องจาก P_{Di} คงที่ ดังนั้น

$$dP_{G1} + dP_{G2} + ... + dP_{GN} = 0 \quad (2.4)$$

คูณสมการที่ (2.4) ด้วย λ แล้วลบจากสมการ (2.3) (Method of LaGrange multiplies) จะได้

$$(\frac{\partial F_1}{\partial P_{G1}} - \lambda)dP_{G1} + (\frac{\partial F_2}{\partial P_{G2}} - \lambda)dP_{G2} + ... + (\frac{\partial F_N}{\partial P_{GN}} - \lambda)dP_{GN} = 0 \quad (2.5)$$

โดย λ เป็นตัวคูณลากรัง (LaGrange multiplies) จากสมการที่ (2.5) จะได้

$$\frac{\partial F_1}{\partial P_{G1}} = \lambda \ , \frac{\partial F_2}{\partial P_{G2}} = \lambda \ ,..., \ \frac{\partial F_N}{\partial P_{GN}} = \lambda$$

สรุปได้ว่าเครื่องกำเนิดไฟฟ้าแต่ละหน่วยที่ร่วมกันจ่ายโหลดได้อย่างประหยัดที่สุดสำหรับระบบที่ไม่คิด Loss $\Rightarrow \sum_{i=1}^{N} P_{Gi} = \sum_{i=1}^{N} P_{Di}$ จะต้องมีอัตราค่าใช้จ่ายเชื้อเพลิงเท่ากันทุกตัว $\frac{\partial F_i}{\partial P_{Gi}} = \lambda$ โดย i = 1,2,3,…,N

และเป็นไปตามขอบเขตการจ่ายกำลังไฟฟ้าของเครื่องกำเนิดไฟฟ้า $P_{Gi(min)} \le P_{Gi} \le P_{Gi(max)}$

2.3 เศรษฐศาสตร์การจ่ายกำลังไฟฟ้า (Economic Dispatch)

- การผลิตและส่งจ่ายพลังงานไฟฟ้าให้กับโหลด สิ่งสำคัญที่ต้องพิจารณาคือ ต้นทุน (Cost)
- หาวิธีที่ทำให้การจ่ายโหลดมีประสิทธิภาพที่สุด โดยที่เสียค่าใช้จ่ายน้อยที่สุด

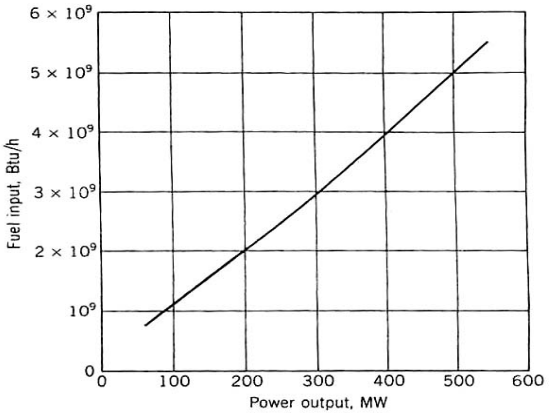
แยกศึกษาเป็น 2 ส่วน คือ

1. เศรษฐศาสตร์ระหว่างเครื่องกำเนิดไฟฟ้าในโรงไฟฟ้า (Plant)
2. เศรษฐศาสตร์ระหว่างโรงไฟฟ้า (plant) กับระบบ (system)

2.4 เศรษฐศาสตร์ระหว่างเครื่องกำเนิดไฟฟ้าในโรงไฟฟ้า

- เครื่องกำเนิดไฟฟ้าแต่ละชนิด มีการใช้เชื้อเพลิงที่แตกต่างกัน
- เชื้อเพลิงที่ใช้ในการผลิตต่างกัน ต้นทุนก็จะไม่เท่ากัน

- ค่าต้นทุนของแต่ละหน่วยผลิต จะแสดงในรูปฟังก์ชันของกำลังไฟฟ้าจ่ายออก
- ความสัมพันธ์ระหว่างค่าใช้จ่ายเชื้อเพลิงและกำลังไฟฟ้าที่ได้ออกมาจะแตกต่างกัน ตามชนิดของ เครื่องกำเนิดไฟฟ้าและเชื้อเพลิงที่ใช้

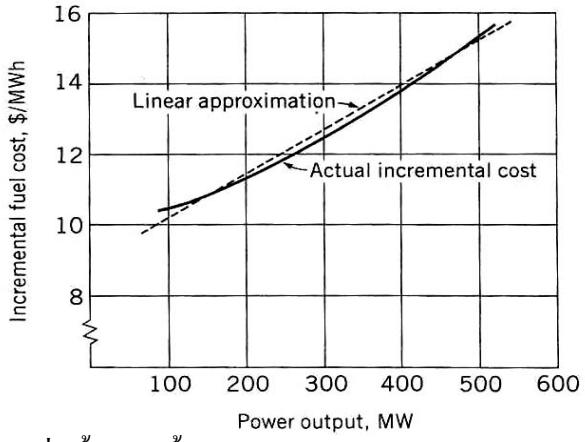


ภาพที่ 2.6 กราฟอินพุต – เอาต์พุต ของเครื่องกำเนิดไฟฟ้ากังหันไอน้ำ (Steam Turbine)

- เครื่องกำเนิดไฟฟ้าขนาดต่างกัน ประสิทธิภาพของการผลิตไฟฟ้าจะไม่เท่ากัน
- พิจารณาประสิทธิภาพจาก อัตราส่วนของปริมาณความร้อนต่อเวลาจากเชื้อเพลิงต่อกำลังไฟฟ้าที่ผลิตได้ → อัตราส่วนยิ่งต่ำ ประสิทธิภาพยิ่งสูง

กำหนดให้ Fn คือ ต้นทุนเชื้อเพลิงของหน่วยผลิตที่ n (Baht per hour)

Pn คือ กำลังผลิตของหน่วยผลิตที่ n (MW)



ภาพที่ 2.7 ต้นทุนของการเพิ่มขึ้นของเชื้อเพลิง เท่ากับ dF_n / dP_n (baht / MWh)

ตารางที่ 2.1 ต้นทุนการผลิตไฟฟ้าจากเชื้อเพลิงชนิดต่างๆ

ประเภทเชื้อเพลิง	บาท / หน่วย
โซลาร์เซล	20.20
พลังงานลม	6.90
น้ำมันดีเซล	6.75
น้ำมันเตา	4.00
ก๊าซธรรมชาติ	1.50
ถ่านหินนำเข้า	1.00
ลิกไนต์	0.53

2.5 เศรษฐศาสตร์โรงไฟฟ้านิวเคลียร์

นิวเคลียร์ เป็นพลังงานรูปแบบหนึ่งที่ถูกนำมาใช้ผลิตไฟฟ้าหลายแห่งในโลกกว่า 40 ปี แม้ว่าจะต้องใช้เงินลงทุนสูงกว่า อีกทั้งต้องมีภาระจัดการกากกัมมันตรังสี และการรื้อถอนโรงไฟฟ้าเมื่อหมดอายุการใช้งานแล้ว แต่ยังมีข้อได้เปรียบใน เรื่องต้นทุนการผลิตไฟฟ้าที่ถูกกว่าไฟฟ้าจากเชื้อเพลิงฟอสซิล

โรงไฟฟ้าเชื้อเพลิงฟอสซิล ต้องเพิ่มค่าใช้จ่ายเพื่อควบคุมผลกระทบต่อสิ่งแวดล้อมและสังคม ซึ่งได้แก่ มลภาวะต่างๆ เช่น ก๊าซซัลเฟอร์ไดออกโซด์ ก๊าซไนโตรเจนออกไซด์ ก๊าซเรือนกระจก รวมถึงฝุ่นละออง จากการเผาไหม้ถ่านหิน อีกทั้งเชื้อเพลิงฟอสซิล เริ่มมีข้อจำกัด คือ ถ่านหินมีพอใช้ได้ประมาณ 200 ปี น้ำมัน 45 ปี ก๊าซธรรมชาติ 67 ปี และหากยังมีการเร่งรัดนำมาใช้เช่นปัจจุบัน ก็หมดไปอย่างรวดเร็วกว่าที่คาดไว้

ตารางที่ 2.2 ต้นทุนการผลิตไฟฟ้าจากเชื้อเพลิงชนิดต่างๆ ของโรงไฟฟ้านิวเคลียร์

ค่าความร้อนของเชื้อเพลิงแต่ละประเภท	
เชื้อเพลิง	เมกะจูลย์ / กิโลกรัม
ไม้ฟืน	16
ถ่านหิน	24-30
ก๊าซธรรมชาติ	39
น้ำมัน	45-46
ยูเรเนียม(ชนิดระบายความร้อนด้วยน้ำ)	440,000

ปฏิกิริยาแตกตัว (fission) จากยูเรเนียม ให้กำเนิดพลังงานอย่างมหาศาลและใช้ปริมาณน้อยกว่ากล่าวคือในเชื้อเพลิงปริมาณ 1 กิโลกรัมเท่ากัน ยูเรเนียมธรรมชาติจะให้พลังงานความร้อนมากกว่าถ่านหินประมาณ 2 หมื่นเท่า นอกจากนี้ยังมีแหล่งแร่ยูเรเนียมกระจายอยู่ทั่วโลก มีผู้ผลิตและจำหน่ายมากราย การขนส่งสะดวก ราคาถูกกว่าและไม่เปลี่ยนแปลงบ่อยนัก

ภาพที่ 2.8 เปรียบเทียบค่าใช้จ่ายการจัดการกากจากโรงไฟฟ้า

นิวเคลียร์ 1 - ฝังเก็บเชื้อเพลิงใช้แล้วโดยตรง นิวเคลียร์ 2 - สกัดเชื้อเพลิงฯ ก่อนฝังเก็บกาก

ต้นทุนการผลิตเชื้อเพลิงนิวเคลียร์ต่ำกว่าถ่านหินประมาณ 1 ใน 3 ส่วนและต่ำกว่าก๊าซธรรมชาติ ประมาณ 1ใน 4 ถึง 5 ส่วน ต้นทุนการผลิตเชื้อเพลิงนิวเคลียร์ รวมถึงขบวนการผลิตและการจัดการกำจัดกากกัมมันตรังสี จากเชื้อเพลิงใช้แล้วโดยการฝังเก็บโดยตรงหรือนำมาสกัดเอายูเรเนียมที่เหลืออยู่ และพลูโตเนียมที่เกิดขึ้นจากปฏิกิริยาฟิชชัน ออกมาประกอบเป็นเชื้อเพลิงใหม่ โดยแยกส่วนที่ไม่ต้องการหรือกาก นำไปฝังเก็บ เช่นเดียวกับเชื้อเพลิงใช้แล้ว

ถึงแม้ว่า ต้นทุนการผลิตเชื้อเพลิงนิวเคลียร์จะต่ำกว่าเชื้อเพลิงฟอสซิล แต่เงินลงทุนในการก่อสร้างจะสูงกว่า เนื่องจาก ต้องมาตรการความปลอดภัยในส่วนของการออกแบบ การติดตั้งเครื่องมือ และอุปกรณ์ทางด้านความปลอดภัยต่างๆ OECD ได้ประมาณค่าพิสัยเฉลี่ย ของเงินลงทุนของโรงไฟฟ้าต่างๆในแต่ละประเทศไว้ดังนี้ โรงไฟฟ้านิวเคลียร์อยู่ระหว่าง 1,277-2,566 ดอลลาร์สหรัฐ/กิโลวัตต์ โรงไฟฟ้าถ่านหิน 772-2,678 ดอลลาร์สหรัฐ / กิโลวัตต์ โรงไฟฟ้าพลังความร้อนร่วม 402-1,514 ดอลลาร์สหรัฐ /กิโลวัตต์ และได้วิเคราะห์ความเหมาะสมเชิงเศรษฐศาสตร์ ของโรงไฟฟ้าต่างๆในกลุ่มสมาชิก โดยใช้อัตราส่วนลดของมูลค่าสินทรัพย์ (Discount rate) ที่ร้อยละ 5 ซึ่งพบว่าใน 7 ประเทศจาก 13 ประเทศเห็นควรสร้างโรงไฟฟ้านิวเคลียร์เพิ่มขึ้น เพื่อรองรับความต้องการพลังงานไฟฟ้าใน พ.ศ. 2553

ตารางที่ 2.3 เปรียบเทียบต้นทุนการผลิตไฟฟ้า

เปรียบเทียบการคาดการต้นทุนการผลิตไฟฟ้าระหว่าง พ.ศ. 2548-2553

	นิวเคลียร์	ถ่านหิน	ก๊าซธรรมชาติ
ฝรั่งเศส	3.22	4.64	4.74
รัสเซีย	2.69	4.63	3.54
ญี่ปุ่น	5.75	5.58	7.91
เกาหลี	3.07	3.44	4.25
สเปน	4.10	4.22	4.79
สหรัฐอเมริกา	3.33	2.48	2.33-2.71
แคนาดา	2.47-2.96	2.92	3.00
จีน	2.54-3.08	3.18	-

เทียบจากต้นทุนการผลิตไฟฟ้าฐาน พ.ศ.2540 หน่วย : เซนต์/กิโลวัตต์-ชั่วโมง

อย่างไรก็ตาม ต้นทุนการผลิตไฟฟ้าต่อหน่วย โดยเฉลี่ยจะถูกกว่าโรงไฟฟ้าพลังความร้อนแบบอื่นๆเนื่องจากต้นทุนการผลิตเชื้อเพลิงที่ต่ำกว่า และใช้ปริมาณน้อยกว่า หากมีการเปลี่ยนแปลงราคายูเรเนียมสูงขึ้น หนึ่งเท่าตัว จะทำให้ต้นทุนการผลิตไฟฟ้า ต่อหน่วยเพิ่มขึ้น ไม่เกินร้อยละ 10

ต้นทุนการผลิตไฟฟ้า คือ ค่าใช้จ่ายต่างๆเช่น ค่าก่อสร้างโรงไฟฟ้า การพัฒนาโครงสร้างพื้นฐาน พัฒนาบุคลากร ค่าเชื้อเพลิงนิวเคลียร์ ค่าดำเนินการเดินเครื่องและบำรุงรักษา รวมทั้งค่าใช้จ่ายในการจัดการกากกัมมันตรังสีและการรื้อถอนโรงไฟฟ้าเมื่อหมดอายุใช้งานแล้ว

2.6 สมการวัตถุประสงค์ (Objective Function)

Minimize
$$F_T(P) = F_1(P_1) + F_2(P_2) + ... + F_{NG}(P_{NG}) = \sum_{i=1}^{NG} F_i(P_i) \quad (2.4.1)$$

เงื่อนไข (Contraint)

Subject to
$$P_{G1} + P_{G2} + ... + P_{NG} = P_{demand} \quad (2.4.2)$$

หรือ
$$P_{demand} - \sum_{i=1}^{NG} P_{Gi} = 0 \quad (2.4.3)$$

หรือ

Min

$$F_T = (a_1 + b_1 P_{G1} + c_1 P_{G1}^{\ 2}) + (a_2 + b_2 P_{G2} + c_2 P_{G2}^{\ 2}) + ... + (a_N + b_N P_{GN} + c_N P_{GN}^{\ 2}) \quad (2.4.4)$$

Subject to
$$P_{demand} - P_{G1} - P_{G2} - ..... - P_{GN} = 0 \quad (2.4.5)$$

F_T = generation cost parameters

P_{Gi} = power output

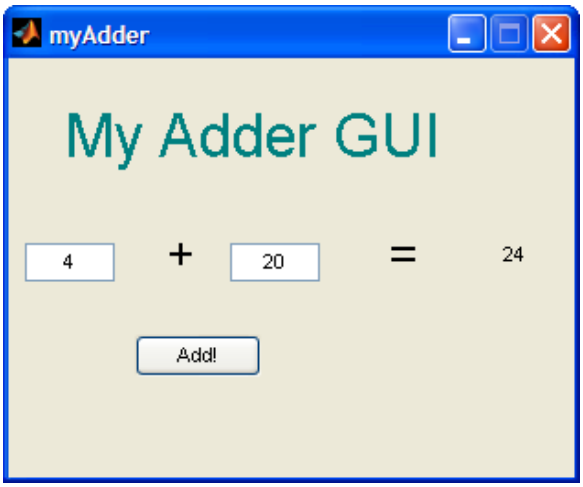
N_G = number of generating

a_i, b_i, c_i = generation cost parameters

$P_{Gi}^{\min}, P_{Gi}^{\max}$ = minimum and maximum power output

2.7 โปรแกรม MATLAB R2009a

ทำไมถึงต้องใช้ GUI ใน MATLAB ประเด็นสำคัญที่ต้องใช้ GUI ก็คือมันเป็นสิ่งที่ง่ายสำหรับผู้ที่ใช้งานโปรแกรมถ้าเราไม่ใช้ GUI ผู้คนอาจจะต้องทำงานจากคำสั่งซึ่งมันจะยุ่งยากเป็นอย่างมาก สมมุติถ้าคุณต้องป้อนคำสั่งเพื่อทำเว็บไซต์ของคุณ (แน่นอนว่าเว็บไซต์ของคุณก็ต้องเป็น GUI เช่นกัน) มันดูไม่น่าจะฝึกฝนอะไรมากใช่ไหม? โปรแกรมการสอนนี้เราจะสร้าง GUI แบบง่ายซึ่งจะเพิ่มจำนวนสองจำนวนเข้าด้วยกัน โดยจะแสดงคำตอบในช่องข้อความ



ภาพที่ 2.9 หน้าตาของโปรแกรม GUI

โปรแกรมการสอนนี้ถูกเขียนขึ้นมาเพื่อผู้ที่ไม่มีประสบการณ์ในการสร้าง MATLAB GUI ความรู้เบื้องต้นของ MATLAB มันไม่สำคัญแต่ก็เป็นสิ่งที่ต้องรู้ MATLAB เวอร์ชั่น 2007a ถูกใช้เขียนลงในโปรแกรมช่วยสอนนี้ด้วย ทั้งสองเวอร์ชั่นนั้นง่ายและใหม่กว่าน่าจะเข้ากันได้เป็นอย่างดี

หัวข้อต่างๆ มีดังนี้

- แนวทางการเริ่มต้น (สำหรับผู้สร้าง GUI)

- การสร้างรูปแบบจำลองของ GUI บทที่ 1

- การสร้างรูปแบบจำลองของ GUI บทที่ 2

- การเขียนรหัสสำหรับ GUI

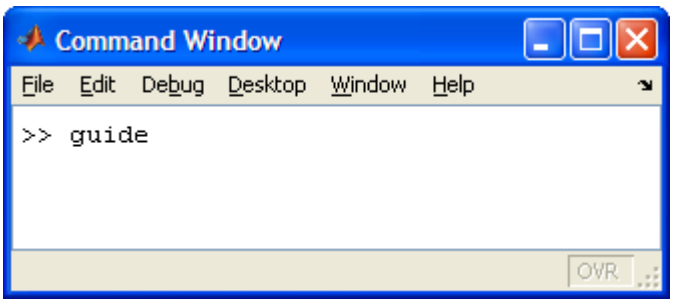
- การค้นหา GUI

- ปัญหาและการแก้ไข

- ความสัมพันธ์และการเชื่อมต่ออื่นๆ

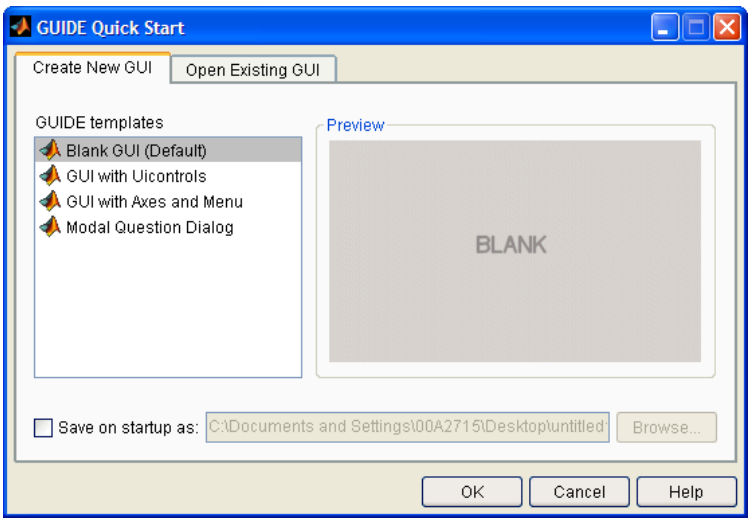
2.7.1 แนวทางการเริ่มต้น (สำหรับผู้สร้าง GUI)

1. สิ่งแรกที่ต้องทำคือ เปิดหน้าต่าง MATLAB ขึ้นมาแล้วไปที่ Command Window และพิมพ์ลงไป



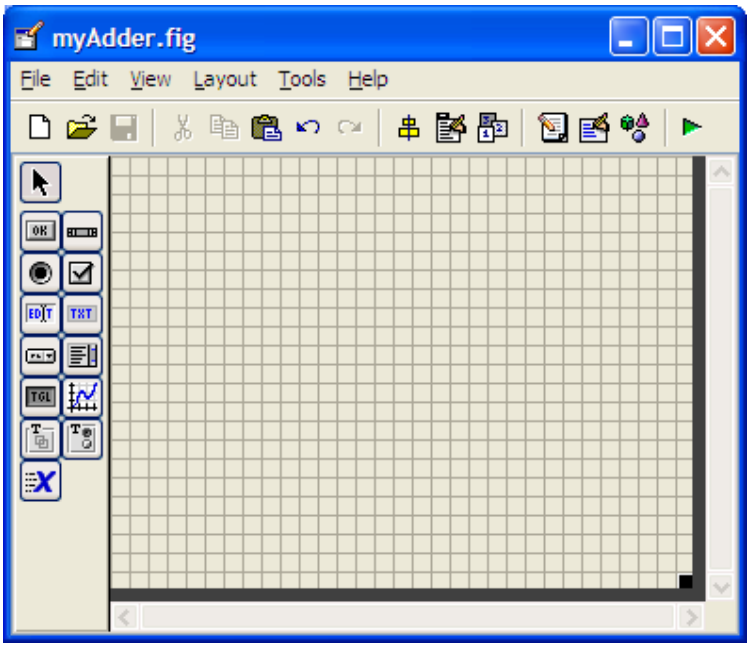
ภาพที่ 2.10 หน้าต่าง Command Window

2. คุณจะเห็นหน้าจอปรากฏขึ้น คลิกที่คำสั่งแรก Blank GUI (default)



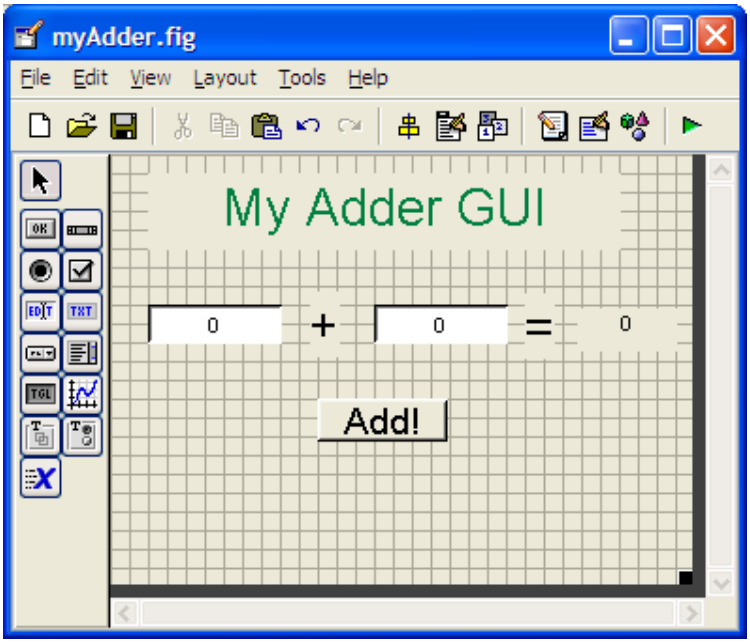
ภาพที่ 2.11 หน้าต่างของ Blank GUI

3. คุณก็จะเห็นมันขึ้นมาบนจอ หรือเห็นบางสิ่งคล้ายกันซึ่งขึ้นอยู่กับว่ามันเป็น MATLAB เวอร์ชั่นไหน ที่คุณกำลังใช้อยู่



ภาพที่ 2.12 หน้าต่างกราฟฟิกของ GUI

4. ก่อนที่จะเพิ่มส่วนประกอบลงไป มันน่าจะดีถ้าหากมีไอเดียคร่าวๆก่อนเกี่ยวกับกราฟฟิกของ GUI
5. ด้านล่างนี้เป็นการแสดงตัวอย่างการเสร็จสิ้นของ GUI จะมีลักษณะดังนี้



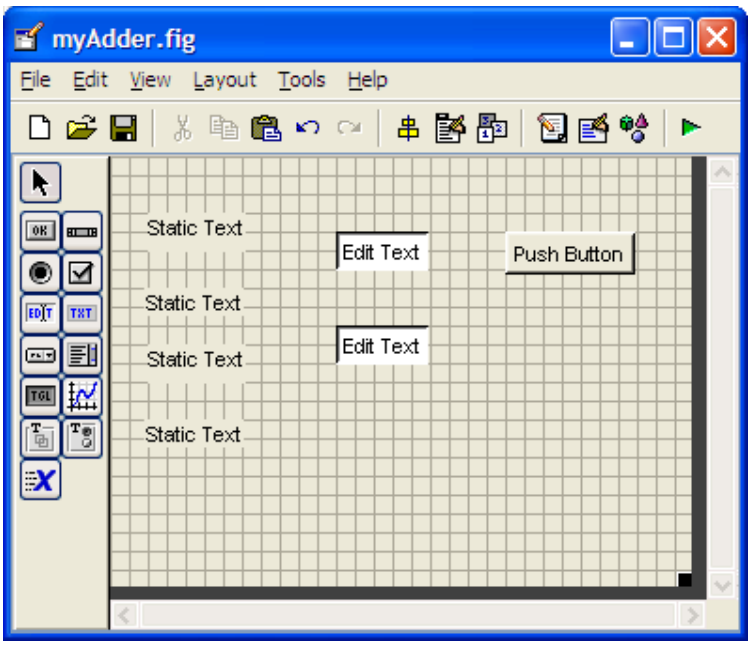
ภาพที่ 2.13 การแสดงตัวอย่างที่เสร็จสมบูรณ์

2.7.2 การสร้างรูปแบบจำลองของ GUI บทที่ 1

1. สำหรับการเพิ่มเติม GUI เราจึงต้องการดังต่อไปนี้

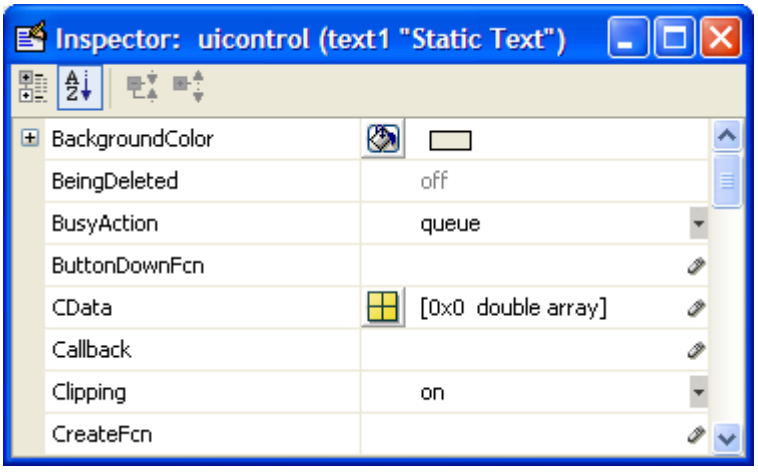
   - ส่วนในการแก้ไขข้อความ
   - static
   - 1 ปุ่มกด Push Button

เพิ่มส่วนประกอบทั้งสามส่วนลงใน GUI โดยการคลิกที่ไอคอนและวางส่วนประกอบทั้งสามลงในช่องตาราง และในส่วนนี้ GUI ของคุณก็จะมีลักษณะดังภาพต่อไปนี้



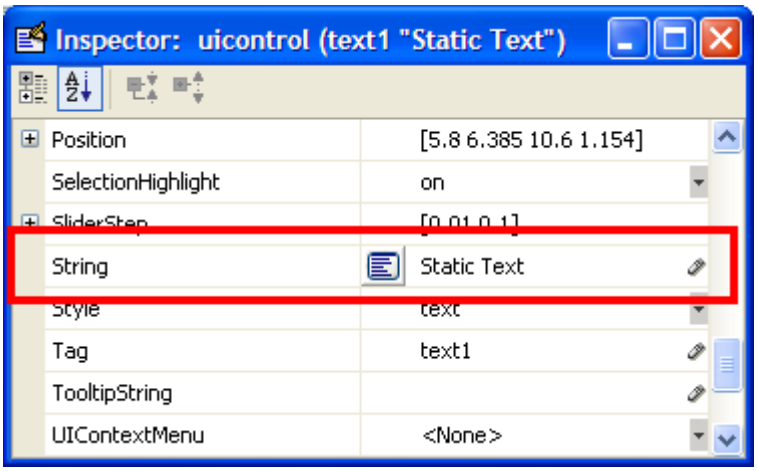
ภาพที่ 2.14 การจัดวางส่วนประกอบ

2. ลำดับต่อไปได้เวลาที่เราจะแก้ไขคุณสมบัติของส่วนประกอบต่างๆเหล่านี้ เริ่มที่ Static Text ดับเบิ้ลคลิกที่ Static Text อันใดอันหนึ่งก็ได้ คุณจะเห็นตารางปรากฏขึ้น เรียกว่า Property Inspector และคุณสามารถแก้ไขคุณสมบัติของส่วนประกอบได้



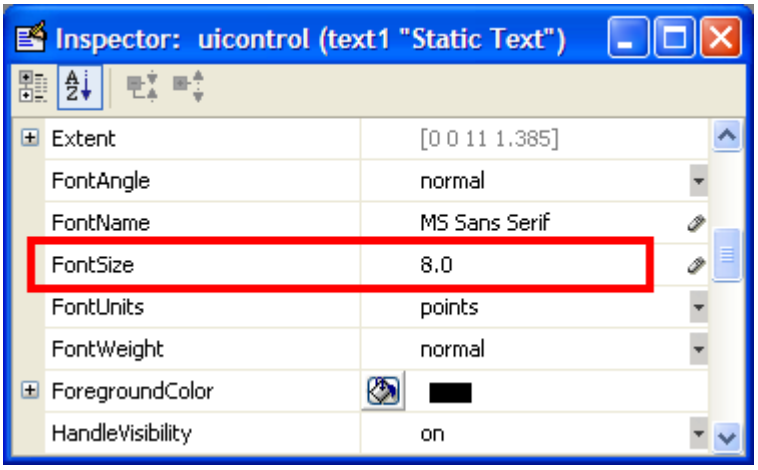
ภาพที่ 2.15 การแก้ไขคุณสมบัติของส่วนประกอบ

3. เราจะรู้สึกสนใจในการเปลี่ยน String Parameter (ข้อความของตัวแปร) เลือกและแก้ไขข้อความนี้ให้เป็น +



ภาพที่ 2.16 การแก้ไขข้อความต่างๆ

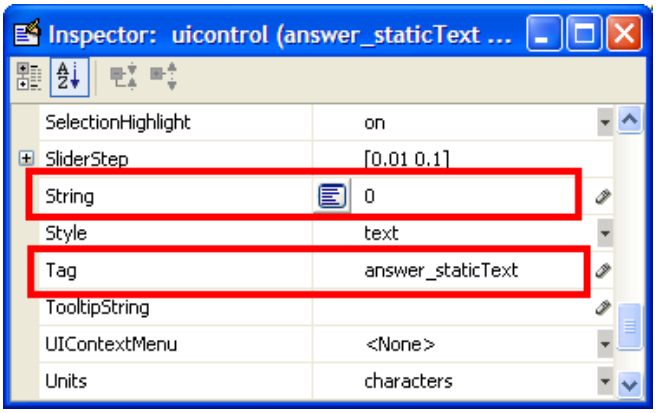
เช่นเดียวกันจะต้องเปลี่ยนขนาดตัวอักษรจาก 8 เป็น 20



ภาพที่ 2.17 การแก้ไขขนาดตัวอักษร

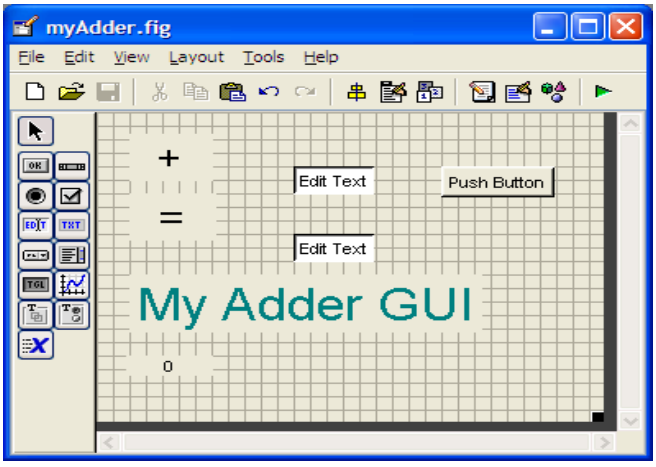
หลังจากแก้ไขคุณสมบัติต่างๆแล้ว ส่วนประกอบนั้นยังอาจจะยังมองเห็นได้ไม่ชัดนักบนตัว GUI มันแก้ไขได้ถ้าคุณลดขนาดของส่วนประกอบลง ใช้เคอเซอร์เพื่อยืดส่วนประกอบให้มีขนาดใหญ่ขึ้น

4. ทำเหมือนเดิมสำหรับ Static Text อันต่อไป แต่เปลี่ยนข้อความตัวแปรจาก 0 เป็น = แทน
5. สำหรับ Static Text อันที่สามนั้น ให้คุณเปลี่ยนข้อความตัวแปรเป็นอะไรก็ได้ที่คุณอยากจะใช้เป็นชื่อหัวข้อของ GUI ของคุณ
6. คุณสามารถทดลองเปลี่ยนรูปแบบตัวอักษรในแบบต่างๆได้เช่นเดียวกัน
7. สำหรับ Static Text อันสุดท้าย เราต้องการตั้งค่าข้อความตัวแปรเป็น 0 นอกจากนี้ เราต้องการแก้ไข Tagparameter สำหรับส่วนประกอบนี้ Tagparameter คือ ชื่อตัวแปรอย่างง่ายของส่วนประกอบนี้ อาจจะเรียกได้ว่ามันคือคำตอบของ Static Text ในส่วนประกอบนี้จะถูกใช่เพื่อแสดงคำตอบของเรา ในบางครั้งคุณอาจจะเดาไว้แล้ว



ภาพที่ 2.18 การแก้ไขชื่อตัวแปร

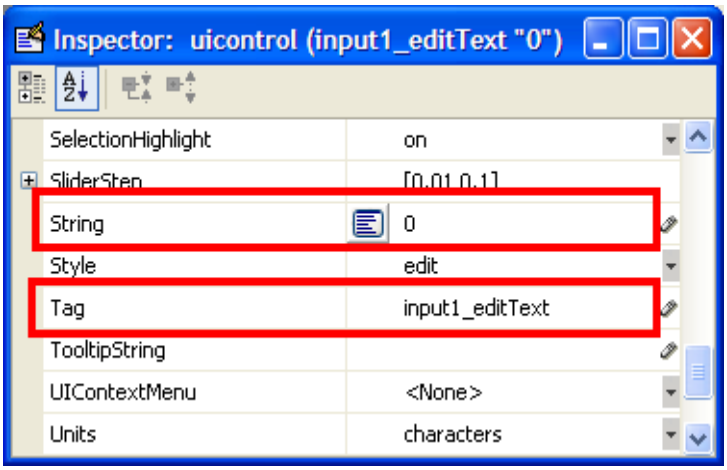
8. ซึ่งในตอนนี้น่าจะมีลักษณะดังภาพด้านล่างนี้



ภาพที่ 2.19 ลักษณะการจัดวาง

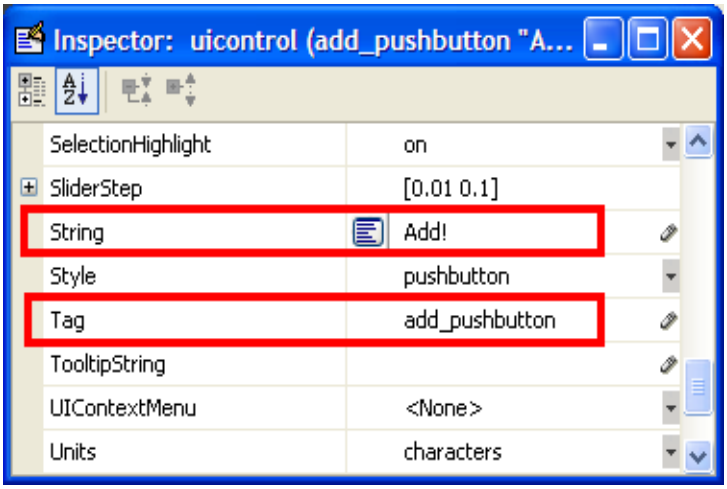
2.7.3 การสร้างรูปแบบจำลองของ GUI บทที่ 2

1. ต่อไปคือการแก้ไขส่วนประกอบ Edit Text ดับเบิ้ลคลิกที่ส่วนประกอบ Edit Text อันแรก เราต้องการตั้งค่าข้อความตัวแปรเป็น 0 และเรายังต้องการเปลี่ยน Tagparameter เป็น input1_edittext ดังภาพข้างล่างนี้ ส่วนประกอบนี้จะอยู่ในส่วนแรกของทั้งหมดสองส่วนที่จะเพิ่มเข้าไปด้วยกัน



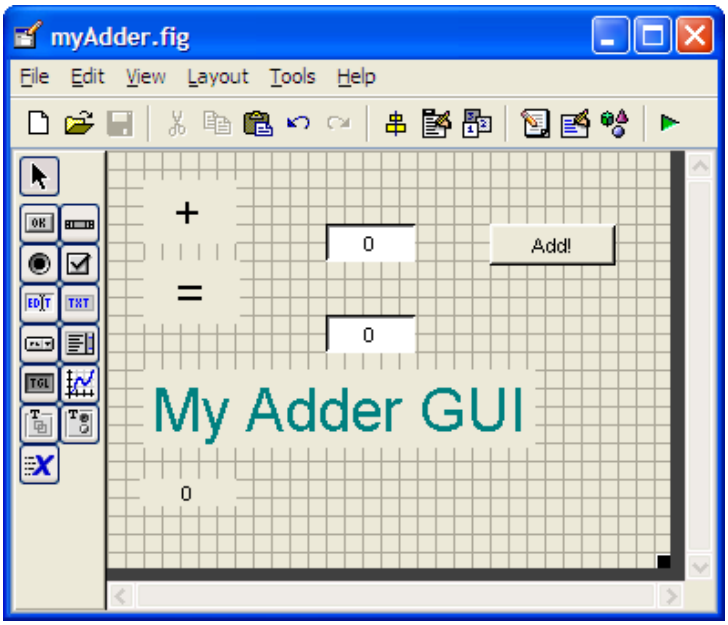
ภาพที่ 2.20 การแก้ไขตัวแปร

2. สำหรับ Edit Text อันที่สองนั้น ตั้งค่าข้อความตัวแปรเป็น 0 แต่ตั้งค่า Tagparameter เป็น input2_editText ส่วนประกอบนี้จะอยู่ในส่วนที่สองของทั้งสองจำนวนที่จะเพิ่มเข้าด้วยกัน
3. ขั้นสุดท้าย เราอยากจะแก้ไข Pushbutton โดยเปลี่ยนข้อความตัวแปรเป็น add และเปลี่ยน Tagparameter เป็น add pushbutton ปุ่มดังกล่าวจะแสดงผลรวมของทั้งสองจำนวนเข้าด้วยกัน



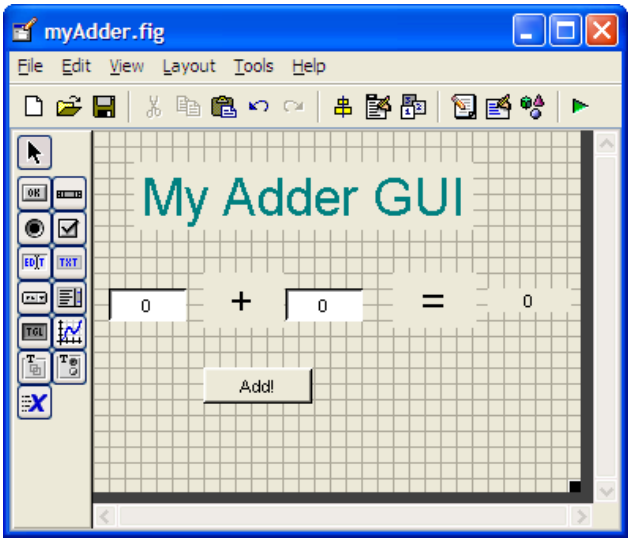
ภาพที่ 2.21 การแก้ไขปุ่มกด

4. และหน้าต่างที่ได้จะมีลักษณะดังนี้



ภาพที่ 2.22 การแก้ไขเสร็จสมบูรณ์

จัดเรียงส่วนประกอบต่างๆ อีกครั้ง เสร็จแล้วหน้าต่างของคุณจะปืนแบบที่เห็นข้างล่างนี้

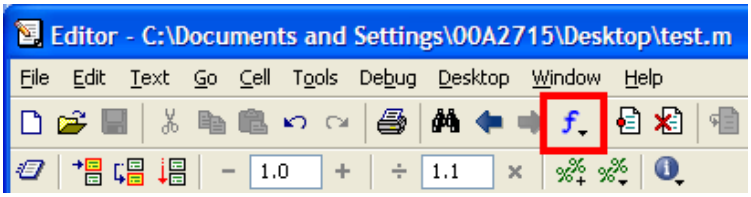


ภาพที่ 2.23 การจัดวางเสร็จสมบูรณ์

5. ตอนนี้ให้คุณจัดการเซฟ GUI โดยใช้ชื่อที่คุณตั้งขึ้นเอง สมมุติใช้ชื่อว่า myAdder เมื่อคุณทำการเซฟแล้ว MATLAB จะเป็นสองไฟล์โดยอัตโนมัติ myAdder.fig และ myAdder.m. .fig ไฟล์ ประกอบไปด้วยกราฟฟิกของคุณส่วน .m ไฟล์ ประกอบด้วยรหัสทั้งหมดของ GUI

MATLAB จะเป็นไฟล์ .m โดยอัตโนมัติตามตัวเลขที่ใส่เข้าด้วยกัน .m ไฟล์นั้นเป็นที่ซึ่งเราใช้แนบรหัสเพื่อเรียกกลับมาใช้ในแต่ละส่วนประกอบ เป้าหมายของโปรแกรมการสอนนี้คือเราต้องคำนึงถึง Callback functions เป็นสิ่งแรก คุณไม่ต้องไปกังวลกับฟังก์ชั่นอื่นๆเลย

1. เมื่อเปิดไฟล์ขึ้นมา .m ไฟล์ นั้นมันจะขึ้นมาโดยอัตโนมัติเมื่อคุณเซฟ GUI ในโปรแกรมการแก้ไข MATLAB, คลิกที่  ไอคอน, ซึ่งมันจะขึ้นรายการฟังก์ชั่นต่างๆ ภายใน .m ไฟล์. เลือกที่ input1_editText_Callback



ภาพที่ 2.24 ฟังก์ชั่นของ GUI

2. เคอเซอร์จะอยู่ที่กล่องรหัสตามข้างล่างนี้

```
function input1_editText_Callback(hObject, eventdata, handles)
% hObject    handle to input1_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'String') returns contents of input1_editText as text
%       str2double(get(hObject,'String')) returns contents of
%       input1_editText as a double
```

ใส่รหัสที่อยู่ข่างล่างนี้ลงในกล่องรหัส

```
%store the contents of input1_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));
%checks to see if input is empty. if so, default input1_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
```

รหัสแต่ละอันนี้ทำให้แน่ใจได้ง่ายว่ามันเข้ากันได้ดี เราไม่ต้องการให้ผู้ใช้ป้อนสิ่งที่ไม่ใช่ตัวเลข บรรทัดสุดท้ายของรหัสบอกว่า GUI อัพเดทโครงสร้างหลังจาก Callback นั้นเสร็จสมบูรณ์แล้ว The handles stores ทั้งหมดจะเกี่ยวข้องโดยตรงกับ GUI ในหัวข้อนี้จะถูกทำความเข้าใจอย่างลึกซึ้งในการสอนที่แตกต่าง ในตอนนี้คุณน่าจะทำที่ Face Value นั่นเป็นความคิดที่ดีที่จะเสร็จสิ้นในแต่ละ Callback function ด้วย GUI data ดังนั้น the handles จะอัพเดทอยู่เสมอหลังจาก Callback นี่จะช่วยไม่ให้คุณปวดหัวในภายหลังได้

11. ใส่รหัสลงในกล่องเดิมเป็น input2_eitText_Callback
12. ตอนนี้เราอยากจะให้คุณแก้ไขที่ add_pushbutton_Callback คลิกที่  ไอคอน และเลือก add_pushbutton_Callback ตามด้วยกล่องรหัสที่คุณน่าจะพบได้ในไฟล์ .m

13. % --- Executes on button press in add_pushbutton.
14. function add_pushbutton_Callback(hObject, eventdata, handles)
15. % hObject   handle to add_pushbutton (see GCBO)
16. % eventdata reserved - to be defined in a future version of MATLAB
17. % handles   structure with handles and user data (see GUIDATA)

นี่เป็นรหัสที่จะต้องเพิ่มเข้าไปใน Callback นี้

```
a = get(handles.input1_editText,'String');
b = get(handles.input2_editText,'String');
% a and b are variables of Strings type, and need to be converted
% to variables of Number type before they can be added together

total = str2num(a) + str2num(b);
c = num2str(total);
% need to convert the answer back into String type to display it
set(handles.answer_staticText,'String',c);
guidata(hObject, handles);
```

18. มาดูว่ารหัสเป็นอย่างไรหลังจากที่เราได้เพิ่มเข้าไปในงาน
19. a = get(handles.input1_editText,'String');
20. b = get(handles.input2_editText,'String');

รหัสสองบรรทัดด้านบนนี้เอามาจากข้อความภายในส่วนประกอบของ Edit Text และ stores รหัสสองบรรทัดนั้นเป็นตัวแปร a และ b รหัสเป็นตัวแปรของ String type และไม่เป็น number type เราไม่สามารถเพิ่มรหัสเข้าด้วยกันอย่างง่ายดาย ด้วยเหตุนี้ เราต้องแปลง a และ b เป็น Number type เสียก่อน MATLAB จึงจะสามารถรวมเข้าด้วยกันได้

21. เราสามารถแปลงตัวแปรของ String type เป็น Number type โดยใช้คำสั่ง MATLAB str2num(String argument) กล่าวง่ายๆ คือ เราสามารถทำให้มันตรงกันข้ามได้โดยการใช้ num2str(Number argument) รหัสข้างล่างนี้เป็นรหัสที่จะต้องใช้เพิ่มเข้าด้วยกัน
22. total= (str2num(a) + str2num(b));

รหัสที่แปลงต่อไปในบรรทัดด้านล่างนี้ได้รวมตัวแปรเป็น String type และ stores มันเข้าด้วยกันกับตัวแปร C

```
c = num2str(total);
```

เหตุผลที่เราแปลงในส่วนของคำตอบสุดท้ายกลับเข้าไปใน String type เป็นเพราะว่าส่วนประกอบ the Static text ไม่ได้แสดงตัวแปรของ Number type ถ้าหากคุณไม่แปลงมันจะกลับเข้าไปใน String type GUI น่าจะ error เมื่อมันพยามที่จะแสดงผลคำตอบ

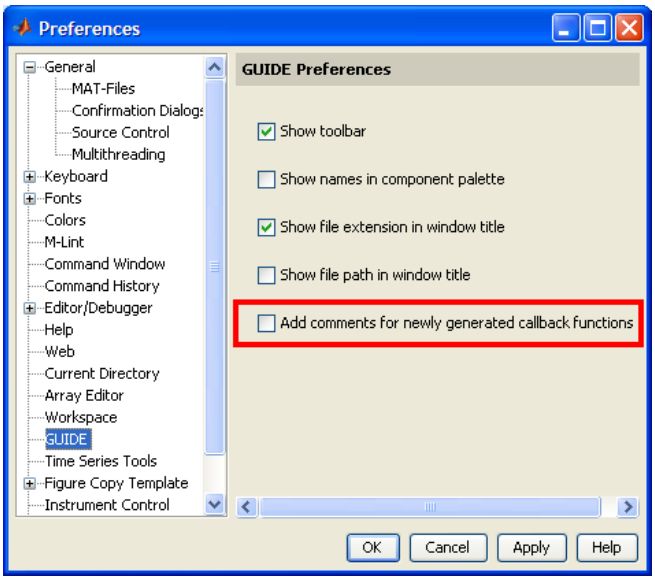
23. ตอนนี้เราแค่ต้องการส่งผลรวมของสองสิ่งเพื่อเข้าไปในกล่องคำตอบ (answer box) นั้นที่เราสร้างขึ้นไว้นี้เป็นขั้นตอนการเสร็จสิ้นของการใช้รหัสแล้ว รหัสบรรทัดด้านล่างนี้ populates the Static text ด้วนตัวแปร C
24. set(handles.answer_staticText,'String',c);

รหัสบรรทัดสุดท้ายที่อยู่ด้านล่างนี้ได้อัพเดทโครงสร้างที่พูดถึงก่อนหน้านี้

```
guidata(hObject, handles);
```

เราเสร็จการใส่รหัส GUI แล้ว อย่าลืมsave .m ไฟล์ของคุณด้วย ต่อไปเป็นการ Launch GUI

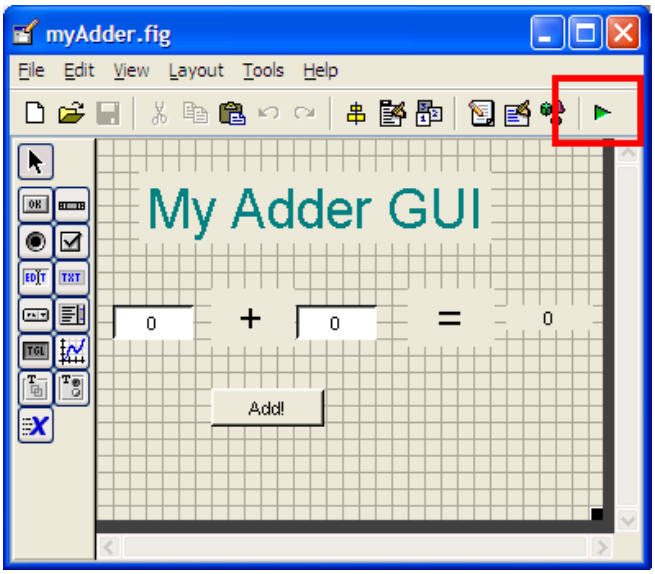
25. ถ้าคุณไม่ต้องการให้ MATLAB มันขึ้นมาโดยอัตโนมัตินี่เป็นวิธีแก้ไขโดยใช้โปรแกรม GUI Editor ไปที่ File เลือก Preferences



ภาพที่ 2.25 การแก้ไขโปรแกรม MATLAB

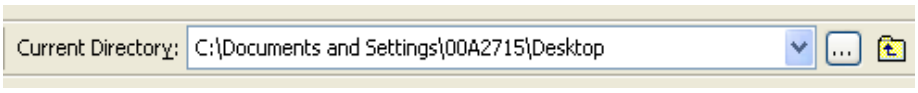
- มี 2 วิธีที่จะ ค้นหา GUI

  - วิธีแรกคือใช้ GUIDE Editor แค่กด  ไอคอนบน GUIDE Editor เหมือนภาพข้างล่างนี้



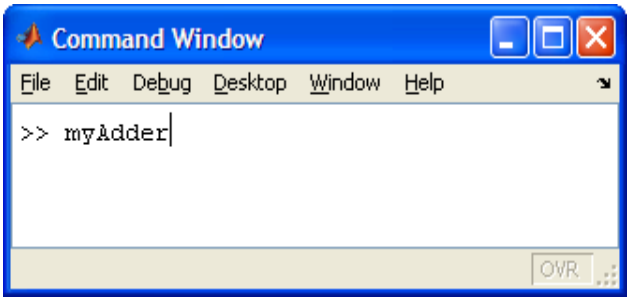
ภาพที่ 2.26 การค้นหาไฟล์ วิธีที่ 1

  - วิธีที่สอง คือ Launch the GUI จากคำสั่ง the MATLAB ได้เลย อย่างแรกตั้งค่า the MATLAB ที่เป็นอยู่ให้เป็นอะไรก็ได้ที่คุณเซฟไว้ในไฟล์ .fig และ .m



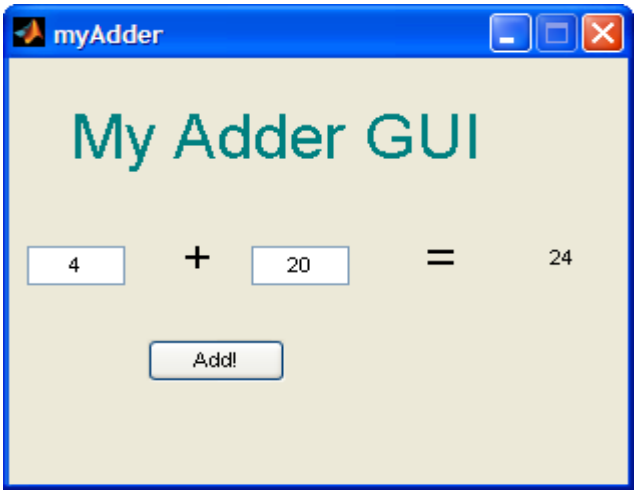
ภาพที่ 2.27 การค้นหาไฟล์ วิธีที่ 2

ขั้นต่อไปพิมพ์ลงในชื่อของ GUI ที่ตัวเลือกคำสั่ง (คุณไม่ต้องพิมพ์ .fig หรือ .m ไฟล์



ภาพที่ 2.28 การค้นหาไฟล์ (ต่อ)

GUI จะเริ่ม running แล้วในขณะนี้



ภาพที่ 2.29 การ RUN โปรแกรมที่เสร็จสมบูรณ์

ใส่หมายเลขลงไปในช่องทดสอบบน GUI ยินดีด้วยกับการสร้าง GUI ครั้งแรกของคุณ

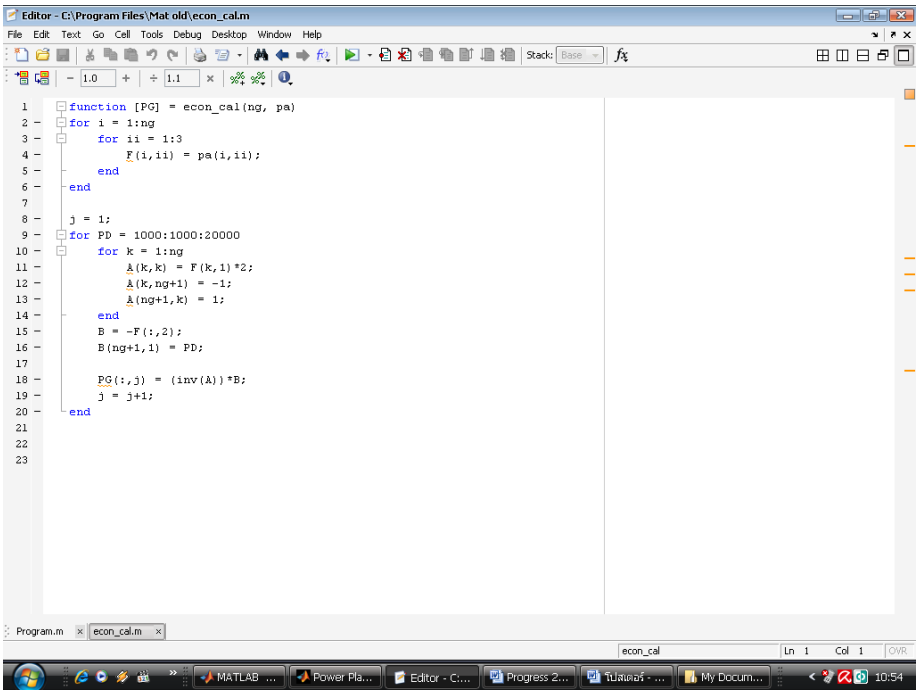
บทที่ 3
การออกแบบโครงงาน

3.1 รูปแบบของโปรแกรม M file

การออกแบบของโปรแกรมคอมพิวเตอร์ให้ใช้งานนั้น แบ่งออกเป็นหน้าโปรแกรมหลักๆ ทั้งหมด 2 หน้าโปรแกรม คือ

3.1.1 ส่วนของโปรแกรมหน้าแรก

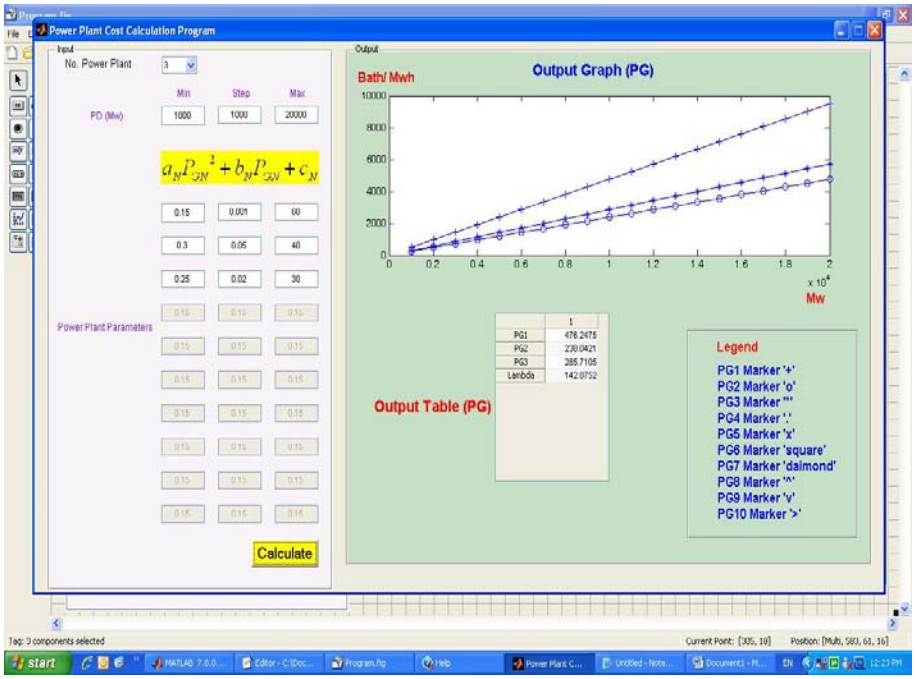
- เป็นหน้าที่ให้ใช้ใส่คำสั่งของการคำนวณสมการ ควอดราติค ดังรูปที่ 3.1



ภาพที่ 3.1 ตัวอย่างหน้าแรกของโปรแกรม

3.1.2 ส่วนของโปรแกรมหน้าที่สอง

- เป็นหน้าที่แสดงค่าของผลการคำนวณทั้งหมด ซึ่งได้มาจากสมการควอดราติคที่ใช้คำนวณหาค่าของโรงไฟฟ้าทั้งหมด ดังรูปที่3.2



ภาพที่ 3.2 ตัวอย่างหน้าที่สองของโปรแกรม

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดสอบโปรแกรม [3]

ได้ทำการทดสอบโปรแกรมกับกรณีศึกษาต่างๆดังต่อไปนี้

กรณีศึกษาที่1 อัตราค่าใช้จ่ายเชื้อเพลิง (Bath/MW-h) สำหรับโรงจักรที่ประกอบด้วยเครื่อง กำเนิด 2 หน่วย กำหนดให้เป็น

$$\frac{dF_1}{dP_{G1}} = 0.008P_{G1} + 8.0 \quad \text{และ} \quad \frac{dF_2}{dP_{G2}} = 0.0096P_{G2} + 6.4$$

สมมุติว่าเครื่องกำเนิดทั้งสองตัวทำงานตลอดเวลาเพื่อร่วมกันจ่ายโหลดที่เปลี่ยนไปตั้งแต่ 250 ถึง 1250 MW และโหลดต่ำสุดสูงสุดของเครื่องกำเนิดไฟฟ้าแต่ละหน่วยเป็น 100 MW และ 625 MW ตามลำดับ จงหาอัตราค่าใช้จ่ายเชื้อเพลิง และค่ากำลังไฟฟ้าที่จ่ายโหลดอย่างประหยัดจากเครื่องกำเนิดแต่ละหน่วย

ที่โหลดขนาด 250 MW

$$\frac{dF_1}{dP_{G1}} = \frac{dF_2}{dP_{G2}}$$

$$0.008P_{G1} + 8.0 = 0.0096P_{G2} + 6.4$$

$$P_{G1} + P_{G2} = 250$$

$$0.008P_{G1} + 8.0 = 0.0096(250 - P_{G1}) + 6.4$$

$$0.0176P_{G1} = 0.8$$

$$P_{G1} = 45.45$$

$$P_{G2} = 204.55$$

เครื่องกำเนิดไฟฟ้าหน่วยที่1มีอัตราค่าใช้จ่ายเชื้อเพลิงสูงกว่า เครื่องกำเนิดไฟฟ้าหน่วยที่2 $(0.008P_{G1} + 8.0 > 0.0096P_{G2} + 6.4)$ เมื่อแทน $P_{G1} = P_{G2}$ ดังนั้นเครื่องที่ 1 จะต้องจ่ายกำลังไฟฟ้าต่ำสุด 100 MW เครื่องที่ 2 จ่าย 150 MW

อัตราการค่าใช้จ่ายเชื้อเพลิง

$$\frac{dF_1}{dP_{G1}} = 8.8 \text{ Bath/MW-h}$$

และ $$\frac{dF_2}{dP_{G2}} = (0.0096 \text{ x } 150) + 6.4 = 7.84 \text{ Bath/MW-h}$$

ที่โหลดขนาด 500 MW

$$P_{G1} + P_{G2} = 500$$
$$0.008P_{G1} + 8.0 = 0.0096P_{G2} + 6.4$$
$$0.008P_{G1} + 8.0 = 0.0096(500 - P_{G1}) + 6.4$$
$$P_{G1} = 181.818$$
$$P_{G2} = 318.182$$
$$\lambda = 9.45 \text{ Bath/MW-h}$$

ที่โหลดขนาด 1175 MW

$$P_{G1} + P_{G2} = 1175$$
$$0.008P_{G1} + 8.0 = 0.0096P_{G2} + 6.4$$
$$P_{G1} = 550$$
$$P_{G2} = 625$$
$$\lambda = 12.40 \text{ Bath/MW-h}$$

ที่โหลดขนาด 1250 MW

$$P_{G1} + P_{G2} = 1250$$
$$0.008P_{G1} + 8.0 = 0.0096P_{G2} + 6.4$$
$$P_{G1} = 590.9$$
$$P_{G2} = 659.1$$

เนื่องจากเครื่องกำเนิดแต่ละหน่วยจ่ายกำลังไฟฟ้าสูงสุดได้ 625 MW เท่านั้น ดังนั้นเครื่องกำเนิดที่ 2 จ่ายได้เพียง 625 MW ส่วนที่เหลือจ่ายจากเครื่องที่ 1 = $1250\text{MW} - 625\text{MW} = 625\text{MW}$ ดังนั้นอัตราการค่าใช้จ่ายเชื้อเพลิง

$$\frac{dF_1}{dP_{G1}} = 13 \text{ Bath/MW-h}$$

และ $$\frac{dF_2}{dP_{G2}} = 12.4 \text{ Bath/MW-h}$$

กรณีศึกษาที่2 จากกรณีศึกษาที่1 จงหาว่าถ้าเครื่องกำเนิดไฟฟ้าร่วมกันจ่ายโหลดอย่างประหยัดที่โหลดรวมทั้งหมด 900 MW จะประหยัดค่าใช้จ่ายเชื้อเพลิงมากกว่า การจ่ายโหลดจากเครื่องกำเนิดที่จ่ายกำลังไฟฟ้าจากแต่ละหน่วยเท่าๆ กัน เท่าไร

เงื่อนไขการจ่ายโหลดอย่างประหยัด

$$\frac{dF_1}{dP_{G1}} = \frac{dF_2}{dP_{G2}}$$

$$0.008P_{G1} + 8.0 = 0.0096P_{G2} + 6.4$$

$$P_{G1} + P_{G2} = 900$$

จะได้

$$P_{G1} = 400\text{MW}$$

$$P_{G2} = 500\text{MW}$$

สำหรับการจ่ายโหลดแบบเท่ากัน

$$P_{G1} = 450\text{MW}$$

$$P_{G2} = 450\text{MW}$$

อัตราค่าใช้จ่ายเชื้อเพลิงสำหรับเครื่องกำเนิดไฟฟ้าหน่วยที่ 1

$$\frac{dF_1}{dP_{G1}} = 0.008P_{G1} + 8.0$$

ดังนั้น ค่าใช้จ่ายเชื้อเพลิงสำหรับเครื่องกำเนิดไฟฟ้าหน่วยที่ 1

$$F_1 = 0.004\ P_{G1}^{\ 2} + 8.0\ P_{G1} + C_1$$

อัตราค่าใช้จ่ายเชื้อเพลิงสำหรับเครื่องเครื่องกำเนิดไฟฟ้าหน่วยที่ 2

$$\frac{dF_2}{dP_{G2}} = 0.0096P_{G2} + 6.4$$

และค่าใช้จ่ายเชื้อเพลิงสำหรับเครื่องกำเนิดไฟฟ้าหน่วยที่ 2

$$F_2 = 0.0048P_{G2}^{\ 2} + 6.4\ P_{G2} + C_2$$

$\therefore$ ค่าใช้จ่ายเชื้อเพลิงรวม $F_t = F_1 + F_2$

$$F_t = 0.004\ P_{G1}^{\ 2} + 8.0\ P_{G1} + C_1 + 0.0048P_{G2}^{\ 2} + 6.4\ P_{G2} + C_2$$

ที่ $P_{G1} = 450 \quad MW$

$P_{G2} = 450 \quad MW$

$F_t = 8262 + C_1 + C_2 \quad Bath/MW\text{-}h$

ที่ $P_{G1} = 400 \quad MW$

$P_{G2} = 500 \quad MW$

$F_t = 8240 + C_1 + C_2 \quad Bath/MW\text{-}h$

เพราะฉะนั้น การจ่ายโหลดอย่างประหยัดจะประหยัดเงินได้มากกว่าการจ่ายโหลดแบบเท่าๆกัน ได้เท่ากับ $8240 - 8262 = -22$ Bath/MW-h (เครื่องหมายลบหมายถึงประหยัดเงิน)

กรณีศึกษาที่3 อัตราค่าใช้จ่ายเชื้อเพลิงสำหรับโรงจักรไฟฟ้าแห่งหนึ่งประกอบด้วย เครื่องกำเนิดไฟฟ้า 2 หน่วย ช่วยกันจ่ายโหลดดังนี้

$$\frac{dF_1}{dP_{G1}} = 0.011\, P_{G1} + 10$$

$$\frac{dF_2}{dP_{G2}} = 0.014\, P_{G2} + 7$$

ตลอดช่วงเวลา 24 ชั่วโมง เครื่องกำเนิดไฟฟ้าแต่ละหน่วยจะจ่ายโหลดอยู่ระหว่าง 200 MW และ 800 MW ในขณะที่โรงจักรไฟฟ้าต้องรับโหลดที่เปลี่ยนจาก 300 ถึง 1400 MW ถ้าโรงจักรไฟฟ้าต้องจ่ายโหลดสูงสุด จะต้องเดินเครื่องกำเนิดไฟฟ้าทั้ง 2 หน่วย เพื่อช่วยกันจ่ายโหลดอย่างไร ถึงทำให้ค่าใช้จ่ายเชื้อเพลิงต่ำสุด

$$\frac{dF_1}{dP_{G1}} = 0.011\, P_{G1} + 10$$

$$\frac{dF_2}{dP_{G2}} = 0.014\, P_{G2} + 7$$

$$P_{G1} + P_{G2} = P_D$$

ที่โหลดขนาด 1400 MW $P_{G1} + P_{G2} = 1400$

$$0.011P_{G1} + 10 = 0.014P_{G2} + 7$$

$$P_{G2} = 1400 - P_{G1}$$

$$0.011P_{G1}+10 = 0.014\,(1400 - P_{G1}) + 7$$

$$0.011P_{G1}+ 0.014\,P_{G1} = 19.6 + 7 - 10$$

$$0.025\,P_{G1} = 16.6$$

$$P_{G1} = 664 \text{ MW}$$

$$P_{G2} = 736 \text{ MW}$$

กรณีศึกษาที่4 โรงกำเนิดไฟฟ้าทั้ง 3 ทำงานตลอดเพื่อร่วมกันจ่ายโหลด จงหาค่ากำลังไฟฟ้าที่จ่ายโหลดอย่างประหยัดจากโรงไฟฟ้าทั้งหมด

$$F_1 = 0.15P_{G1}^{\,2} + 0.001P_{G1} + 60$$
$$F_2 = 0.30P_{G2}^{\,2} + 0.05P_{G2} + 40$$
$$F_3 = 0.25P_{G3}^{\,2} + 0.02P_{G3} + 30$$

$$P_D = 1000\ \ MW.$$

$$\frac{dF_1}{dP_{G1}} = 0.3P_{G1} + 0.001 = \lambda$$
$$\frac{dF_2}{dP_{G2}} = 0.6P_{G2} + 0.05 = \lambda$$
$$\frac{dF_3}{dP_{G3}} = 0.5P_{G3} + 0.02 = \lambda$$

$$\begin{aligned} 0.3P_{G1} + \quad & \quad -\lambda = \ -0.001 \\ +0.6P_{G2} \quad & \quad -\lambda = \ -0.05 \\ +0.5P_{G3} \quad & -\lambda = \ -0.02 \end{aligned}$$

$$\begin{pmatrix} 0.3 & 0 & 0 & -1 \\ 0 & 0.6 & 0 & -1 \\ 0 & 0 & 0.5 & -1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{G1} \\ P_{G2} \\ P_{G3} \\ \lambda \end{pmatrix} = \begin{pmatrix} -0.001 \\ -0.05 \\ -0.02 \\ 1000 \end{pmatrix}$$

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปและข้อเสนอแนะ

การดำเนินงานที่ผ่านมาได้ทำการเขียนโปรแกรม ทำการรับค่าอินพุตเข้ามาเพื่อใช้ในการคำนวณต่อไป โดยแสดงช่วงของค่าข้อมูลเป็นกราฟเพื่อง่ายต่อการพิจารณา โปรแกรมสามารถรับค่าพารามิเตอร์เหล่านี้มาใช้ในการคำนวณ และแสดงผลที่ได้จากการคำนวณในรูปแบบที่เหมาะสมต่อไป หลังจากคำนวณผลแล้ว จะได้ผลข้อมูลที่เป็นเอาท์พุตที่สนใจออกมา สามรถดูค่าข้อมูลได้จากส่วนแสดงผลข้อมูล ในส่วนการแสดงผลข้อมูล สามารถแสดงผลเป็นค่าข้อมูลกราฟได้

ทดลอง RUN โปรแกรมที่ทำขึ้น หาจุดบกพร่องและข้อผิดพลาดต่างๆทดสอบการทำงานของโปรแกรม ตรวจสอบ ปรับปรุง แก้ไข ข้อบกพร่อง จนได้ผลงานดังที่ได้เสนอในรายงานนี้ แต่อาจมีข้อผิดพลาดบางอย่างที่ยังไม่พบ หรือการแสดงผลที่ยังไม่ชัดเจน การใส่ค่าข้อมูลอินพุตที่ยังยุ่งยากอยู่สำหรับตัวแปรบางตัว

- **ปัญหาและอุปสรรค์ที่พบในโครงงาน**

1. ถ้าใช้โปรแกรม MATLAB บางเวอร์ชั่น จะทำให้ผิดพลาดบ้างเล็กน้อย
2. การเขียนโปรแกรม GUI ของ MATLAB ค่อนข้างยุ่งยาก ในเวอร์ชั่นเก่า
3. การแสดงผลบางอย่างยังไม่ค่อยเหมาะสมมากนัก

- **แนวทางการปรับปรุงแก้ไข**

1. ควรตรวจสอบเวอร์ชั่นของ MATLAB ก่อนการใช้งานโปรแกรม แก้ไข code โปรแกรมให้สามารถใช้งานได้ดีที่สุด ผิดพลาดน้อยที่สุดสำหรับทุกๆเวอร์ชั่น
2. โปรแกรม MATLAB ใหม่ๆจะมีความสามารถในการเขียนโปรแกรมได้สะดวกขึ้นมากจากเดิม ทำให้ไม่เสียเวลามากนัก
3. ค่าบางอย่างควรแสดงผลในรูปแบบตาราง จะทำให้พิจารณาได้ง่ายขึ้น

เอกสารอ้างอิง

[1] Piyadanai Pachanapan, 303427 Power System Analysis, EE&CPE, NU

[2] Patick Marchand, Graphics and GUIs with MATLAB. 2nd edition, London, CRC Press, 1999

[3] ผศ.ดร.กีรติ ชยะกุลคีรี หนังสือเรื่อง Power System Analysis มหาวิทยาลัยศรีปทุม

[4] กรมพัฒนาพลังงานทดแทนและอนุรักษ์พลังงาน (พพ.) กระทรวงพลังงาน

ภาคผนวก

โปรแกรมคำสั่ง M-file

```
%        a        b        c
F = [ 0.15     0.001    60
      0.3      0.05     40
      0.25     0.02     30 ]
PD =  1000
NG =  3
%----------------
    for  k = 1:NG
      A(k,k) = F(k,1)*2
      A(k,NG+1) = -1
      A(NG+1,k) =  1
    end
      B = -F( :,2 )
      B(NG+1,1) = PD
%----------------
    PG = ( inv(A) )*B
%            a          b          c
    H =  [0.15×2     0          0
          0          0.3×2      0
          0          0          0]

    f =  [0.001
          0.05
          0.02]

    A =  [1 1 1]

  Pmax =  [1000
           1000
           1000]

  Pmin =  [0
           0
           0]

[X, FVAL, EXITFLAG, OUTPUT] = quadprog (H, f, A, b, PD, [ ], [ ], Pmax, Pmin);
```

```
function varargout = Program(varargin)
% PROGRAM M-file for Program.fig
%      PROGRAM, by itself, creates a new PROGRAM or raises the
existing
%      singleton*.
%
%      H = PROGRAM returns the handle to a new PROGRAM or the handle
to
%      the existing singleton*.
%
%      PROGRAM('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in PROGRAM.M with the given input
arguments.
%
%      PROGRAM('Property','Value',...) creates a new PROGRAM or
raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before Program_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to Program_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

% Edit the above text to modify the response to help Program

% Last Modified by GUIDE v2.5 18-Jan-2010 23:24:45

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @Program_OpeningFcn, ...
    'gui_OutputFcn',  @Program_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 Program is made visible.
function Program_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 Program (see VARARGIN)

eq = imread('eq.jpg');
axes(handles.axes2)
imshow(eq)

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

handles.ng = 1;
% Update handles structure
guidata(hObject, handles)

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



% --- Outputs from this function are returned to the command line.
function varargout = Program_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



% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
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


% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
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


% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
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



% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit5_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



% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
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


% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
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


% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
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


% --- 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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (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 edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9
as a double


% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (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 edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10
as a double


% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (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 edit11 as text
%        str2double(get(hObject,'String')) returns contents of edit11
as a double


% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (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 edit12 as text
%        str2double(get(hObject,'String')) returns contents of edit12
as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (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 edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13
as a double


% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (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 edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14
as a double


% --- Executes during object creation, after setting all properties.
```

```
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (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 edit15 as text
%        str2double(get(hObject,'String')) returns contents of edit15
as a double


% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (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 edit16 as text
%        str2double(get(hObject,'String')) returns contents of edit16
as a double


% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (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 edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17
as a double


% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (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 edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18
as a double


% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (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 edit19 as text
%        str2double(get(hObject,'String')) returns contents of edit19
as a double


% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit20_Callback(hObject, eventdata, handles)
% hObject    handle to edit20 (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 edit20 as text
%        str2double(get(hObject,'String')) returns contents of edit20
as a double


% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit21_Callback(hObject, eventdata, handles)
% hObject    handle to edit21 (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 edit21 as text
%        str2double(get(hObject,'String')) returns contents of edit21
as a double


% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit21 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit22_Callback(hObject, eventdata, handles)
% hObject    handle to edit22 (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 edit22 as text
%        str2double(get(hObject,'String')) returns contents of edit22
as a double


% --- Executes during object creation, after setting all properties.
function edit22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit22 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end



function edit23_Callback(hObject, eventdata, handles)
% hObject    handle to edit23 (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 edit23 as text
%        str2double(get(hObject,'String')) returns contents of edit23
as a double



% --- Executes during object creation, after setting all properties.
function edit23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit23 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit24_Callback(hObject, eventdata, handles)
% hObject    handle to edit24 (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 edit24 as text
%        str2double(get(hObject,'String')) returns contents of edit24
as a double



% --- Executes during object creation, after setting all properties.
function edit24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit24 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit25_Callback(hObject, eventdata, handles)
% hObject    handle to edit25 (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 edit25 as text
%        str2double(get(hObject,'String')) returns contents of edit25
as a double


% --- Executes during object creation, after setting all properties.
function edit25_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit25 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit26_Callback(hObject, eventdata, handles)
% hObject    handle to edit26 (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 edit26 as text
%        str2double(get(hObject,'String')) returns contents of edit26
as a double


% --- Executes during object creation, after setting all properties.
function edit26_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit26 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit27_Callback(hObject, eventdata, handles)
% hObject    handle to edit27 (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 edit27 as text
%        str2double(get(hObject,'String')) returns contents of edit27
as a double


% --- Executes during object creation, after setting all properties.
function edit27_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit27 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit28_Callback(hObject, eventdata, handles)
% hObject    handle to edit28 (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 edit28 as text
%        str2double(get(hObject,'String')) returns contents of edit28
as a double


% --- Executes during object creation, after setting all properties.
function edit28_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit28 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit29_Callback(hObject, eventdata, handles)
% hObject    handle to edit29 (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 edit29 as text
%        str2double(get(hObject,'String')) returns contents of edit29
as a double


% --- Executes during object creation, after setting all properties.
function edit29_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit29 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit30_Callback(hObject, eventdata, handles)
% hObject    handle to edit30 (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 edit30 as text
%        str2double(get(hObject,'String')) returns contents of edit30
as a double


% --- Executes during object creation, after setting all properties.
function edit30_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit30 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents
as cell array
```

```
%        contents{get(hObject,'Value')} returns selected item from
popupmenu1
ng = get(hObject,'value');

set(handles.edit10, 'Enable','off');
set(handles.edit11, 'Enable','off');
set(handles.edit12, 'Enable','off');
set(handles.edit13, 'Enable','off');
set(handles.edit14, 'Enable','off');
set(handles.edit15, 'Enable','off');
set(handles.edit16, 'Enable','off');
set(handles.edit17, 'Enable','off');
set(handles.edit18, 'Enable','off');
set(handles.edit19, 'Enable','off');
set(handles.edit20, 'Enable','off');
set(handles.edit21, 'Enable','off');
set(handles.edit22, 'Enable','off');
set(handles.edit23, 'Enable','off');
set(handles.edit24, 'Enable','off');
set(handles.edit25, 'Enable','off');
set(handles.edit26, 'Enable','off');
set(handles.edit27, 'Enable','off');
set(handles.edit28, 'Enable','off');
set(handles.edit29, 'Enable','off');
set(handles.edit30, 'Enable','off');

if ng == 1

elseif ng == 2

    set(handles.edit10, 'Enable','on');
    set(handles.edit11, 'Enable','on');
    set(handles.edit12, 'Enable','on');
elseif ng == 3

    set(handles.edit10, 'Enable','on');
    set(handles.edit11, 'Enable','on');
    set(handles.edit12, 'Enable','on');
    set(handles.edit13, 'Enable','on');
    set(handles.edit14, 'Enable','on');
    set(handles.edit15, 'Enable','on');
elseif ng == 4

    set(handles.edit10, 'Enable','on');
    set(handles.edit11, 'Enable','on');
    set(handles.edit12, 'Enable','on');
    set(handles.edit13, 'Enable','on');
    set(handles.edit14, 'Enable','on');
    set(handles.edit15, 'Enable','on');
    set(handles.edit16, 'Enable','on');
    set(handles.edit17, 'Enable','on');
    set(handles.edit18, 'Enable','on');
elseif ng == 5
    set(handles.edit10, 'Enable','on');
    set(handles.edit11, 'Enable','on');
    set(handles.edit12, 'Enable','on');
    set(handles.edit13, 'Enable','on');
```

```
    set(handles.edit14, 'Enable','on');
    set(handles.edit15, 'Enable','on');
    set(handles.edit16, 'Enable','on');
    set(handles.edit17, 'Enable','on');
    set(handles.edit18, 'Enable','on');
    set(handles.edit19, 'Enable','on');
    set(handles.edit20, 'Enable','on');
    set(handles.edit21, 'Enable','on');
elseif ng == 6
    set(handles.edit10, 'Enable','on');
    set(handles.edit11, 'Enable','on');
    set(handles.edit12, 'Enable','on');
    set(handles.edit13, 'Enable','on');
    set(handles.edit14, 'Enable','on');
    set(handles.edit15, 'Enable','on');
    set(handles.edit16, 'Enable','on');
    set(handles.edit17, 'Enable','on');
    set(handles.edit18, 'Enable','on');
    set(handles.edit19, 'Enable','on');
    set(handles.edit20, 'Enable','on');
    set(handles.edit21, 'Enable','on');
    set(handles.edit22, 'Enable','on');
    set(handles.edit23, 'Enable','on');
    set(handles.edit24, 'Enable','on');
elseif ng == 7
    set(handles.edit10, 'Enable','on');
    set(handles.edit11, 'Enable','on');
    set(handles.edit12, 'Enable','on');
    set(handles.edit13, 'Enable','on');
    set(handles.edit14, 'Enable','on');
    set(handles.edit15, 'Enable','on');
    set(handles.edit16, 'Enable','on');
    set(handles.edit17, 'Enable','on');
    set(handles.edit18, 'Enable','on');
    set(handles.edit19, 'Enable','on');
    set(handles.edit20, 'Enable','on');
    set(handles.edit21, 'Enable','on');
    set(handles.edit22, 'Enable','on');
    set(handles.edit23, 'Enable','on');
    set(handles.edit24, 'Enable','on');
    set(handles.edit25, 'Enable','on');
    set(handles.edit26, 'Enable','on');
    set(handles.edit27, 'Enable','on');
else
    set(handles.edit10, 'Enable','on');
    set(handles.edit11, 'Enable','on');
    set(handles.edit12, 'Enable','on');
    set(handles.edit13, 'Enable','on');
    set(handles.edit14, 'Enable','on');
    set(handles.edit15, 'Enable','on');
    set(handles.edit16, 'Enable','on');
    set(handles.edit17, 'Enable','on');
    set(handles.edit18, 'Enable','on');
    set(handles.edit19, 'Enable','on');
    set(handles.edit20, 'Enable','on');
    set(handles.edit21, 'Enable','on');
    set(handles.edit22, 'Enable','on');
    set(handles.edit23, 'Enable','on');
```

```
    set(handles.edit24, 'Enable','on');
    set(handles.edit25, 'Enable','on');
    set(handles.edit26, 'Enable','on');
    set(handles.edit27, 'Enable','on');
    set(handles.edit28, 'Enable','on');
    set(handles.edit29, 'Enable','on');
    set(handles.edit30, 'Enable','on');
end



handles.ng = ng;
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

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



% --- 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)
ng = handles.ng;
ng = ng+2;

pa(1,1) = str2double(get(handles.edit1,'String'));
pa(1,2) = str2double(get(handles.edit2,'String'));
pa(1,3) = str2double(get(handles.edit3,'String'));
pa(2,1) = str2double(get(handles.edit4,'String'));
pa(2,2) = str2double(get(handles.edit5,'String'));
pa(2,3) = str2double(get(handles.edit6,'String'));
pa(3,1) = str2double(get(handles.edit7,'String'));
pa(3,2) = str2double(get(handles.edit8,'String'));
pa(3,3) = str2double(get(handles.edit9,'String'));
pa(4,1) = str2double(get(handles.edit10,'String'));
pa(4,2) = str2double(get(handles.edit11,'String'));
pa(4,3) = str2double(get(handles.edit12,'String'));
pa(5,1) = str2double(get(handles.edit13,'String'));
pa(5,2) = str2double(get(handles.edit14,'String'));
pa(5,3) = str2double(get(handles.edit15,'String'));
pa(6,1) = str2double(get(handles.edit16,'String'));
pa(6,2) = str2double(get(handles.edit17,'String'));
pa(6,3) = str2double(get(handles.edit18,'String'));
pa(7,1) = str2double(get(handles.edit19,'String'));
pa(7,2) = str2double(get(handles.edit20,'String'));
```

```
pa(7,3) = str2double(get(handles.edit21,'String'));
pa(8,1) = str2double(get(handles.edit22,'String'));
pa(8,2) = str2double(get(handles.edit23,'String'));
pa(8,3) = str2double(get(handles.edit24,'String'));
pa(9,1) = str2double(get(handles.edit25,'String'));
pa(9,2) = str2double(get(handles.edit26,'String'));
pa(9,3) = str2double(get(handles.edit27,'String'));
pa(10,1) = str2double(get(handles.edit28,'String'));
pa(10,2) = str2double(get(handles.edit29,'String'));
pa(10,3) = str2double(get(handles.edit30,'String'));

pdstart = str2double(get(handles.edit31,'String'));
pdstep = str2double(get(handles.edit32,'String'));
pdstop = str2double(get(handles.edit33,'String'));

[PG] = econ_cal(ng,pa,pdstart,pdstep,pdstop);


PD = [pdstart:pdstep:pdstop];

hold off
axes(handles.axes1)
for i =1:ng
    if i == 1
        plot(PD,PG(i,:),'Marker','+')
    elseif i == 2
        plot(PD,PG(i,:),'Marker','o')
    elseif i == 3
        plot(PD,PG(i,:),'Marker','*')
    elseif i == 4
        plot(PD,PG(i,:),'Marker','.')
    elseif i == 5
        plot(PD,PG(i,:),'Marker','x')
    elseif i == 6
        plot(PD,PG(i,:),'Marker','s')
    elseif i == 7
        plot(PD,PG(i,:),'Marker','d')
    elseif i == 8
        plot(PD,PG(i,:),'Marker','^')
    elseif i == 9
        plot(PD,PG(i,:),'Marker','v')
    elseif i == 10
        plot(PD,PG(i,:),'Marker','>')
    end
    hold on
end

% title('Output Graph (PG)','FontSize',16,'Color','r')
% xlabel('PD (MW)')
% ylabel('Bath/Mwh')


set(handles.uitable1, 'Data', PG(1:ng+1,1));

if ng == 3
    rnames = {'PG1','PG2','PG3','Lambda'};
```

```
elseif ng == 4
    rnames = {'PG1','PG2','PG3','PG4','Lambda'};
elseif ng == 5
    rnames = {'PG1','PG2','PG3','PG4','PG5','Lambda'};
elseif ng == 6
    rnames = {'PG1','PG2','PG3','PG4','PG5','PG6','Lambda'};
elseif ng == 7
    rnames = {'PG1','PG2','PG3','PG4','PG5','PG6','PG7','Lambda'};
elseif ng == 8
    rnames =
{'PG1','PG2','PG3','PG4','PG5','PG6','PG7','PG8','Lambda'};
elseif ng == 9
    rnames =
{'PG1','PG2','PG3','PG4','PG5','PG6','PG7','PG8','PG9','Lambda'};
elseif ng == 10
    rnames =
{'PG1','PG2','PG3','PG4','PG5','PG6','PG7','PG8','PG9','PG10','Lambda
'};
end


set(handles.uitable1,'RowName',rnames);


function edit31_Callback(hObject, eventdata, handles)
% hObject    handle to edit31 (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 edit31 as text
%        str2double(get(hObject,'String')) returns contents of edit31
as a double


% --- Executes during object creation, after setting all properties.
function edit31_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit31 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit32_Callback(hObject, eventdata, handles)
% hObject    handle to edit32 (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 edit32 as text
```

```
%        str2double(get(hObject,'String')) returns contents of edit32
as a double


% --- Executes during object creation, after setting all properties.
function edit32_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit32 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit33_Callback(hObject, eventdata, handles)
% hObject    handle to edit33 (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 edit33 as text
%        str2double(get(hObject,'String')) returns contents of edit33
as a double


% --- Executes during object creation, after setting all properties.
function edit33_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit33 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```