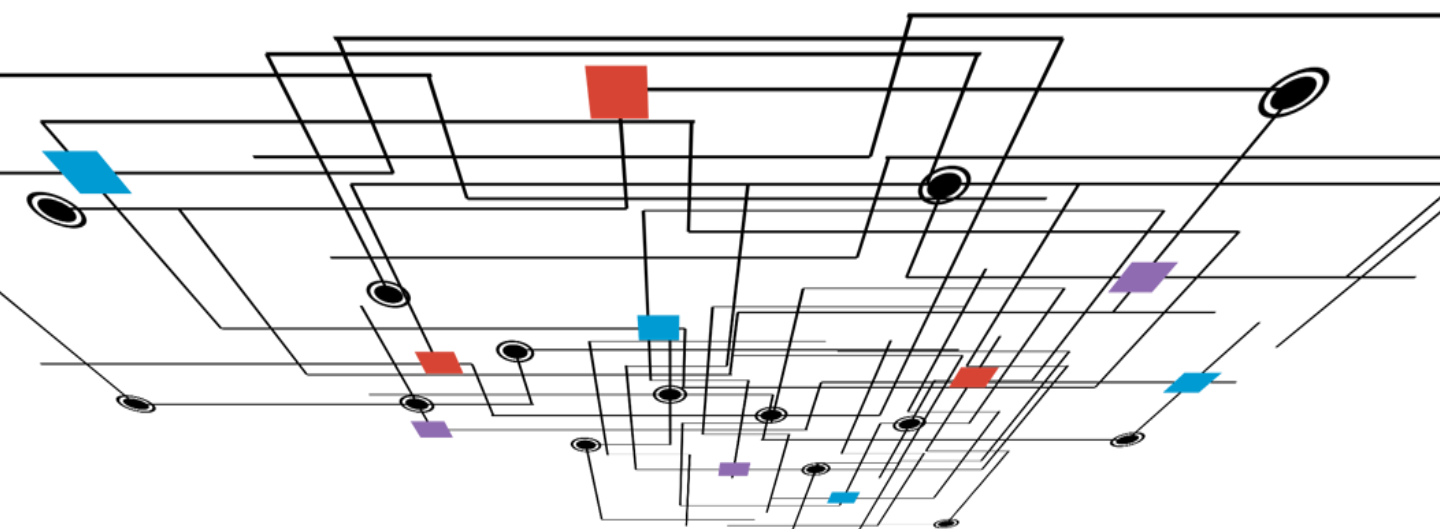
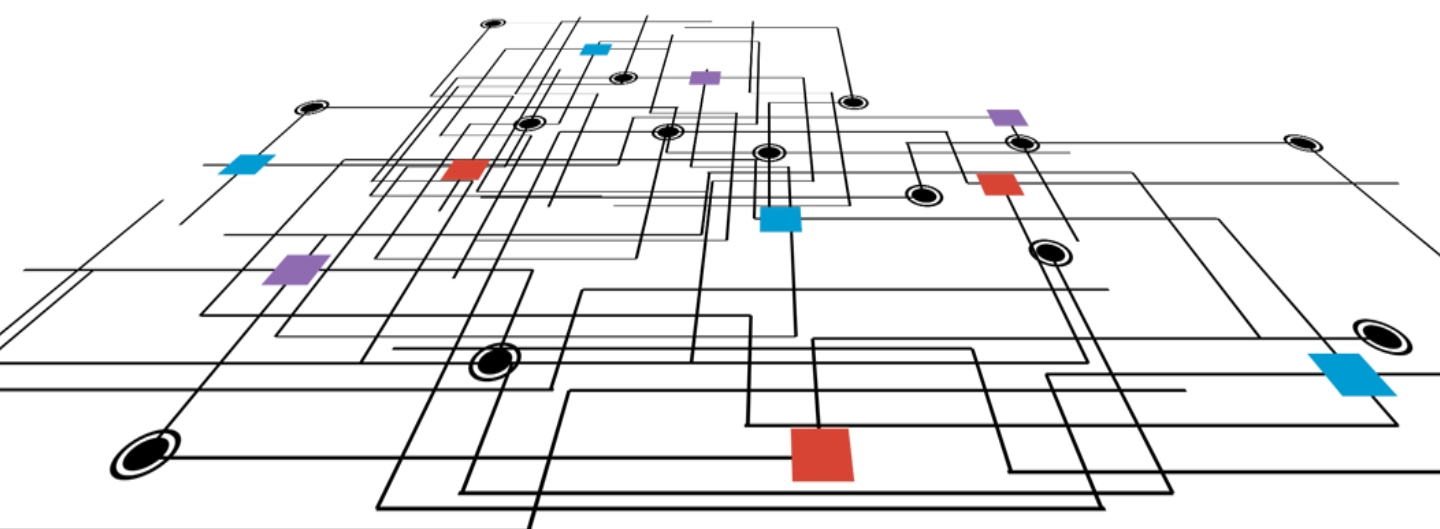




การประยุกต์ใช้โปรแกรม MATLAB

สำหรับการประมวลผลหาค่าและพื้นฐานการใช้งาน



ผู้ช่วยศาสตราจารย์ ดร.รุ่งอรุณ พรเจริญ
คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร



การประยุกต์ใช้โปรแกรม MATLAB สำหรับการประมวลผลหาค่าและพื้นฐานการใช้งาน

ผู้ช่วยศาสตราจารย์ ดร.รุ่งอรุณ พรเจริญ

คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ผู้ช่วยศาสตราจารย์ ดร.รุ่งอรุณ พรเจริญ
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ฉบับพิมพ์ครั้งที่ 1 เดือนกรกฎาคม 2564 จำนวน 200 เล่ม
ฉบับอิเล็กทรอนิกส์ (e-book) เดือนกรกฎาคม 2564

ลิขสิทธิ์ของผู้ช่วยศาสตราจารย์ ดร.รุ่งอรุณ พรเจริญ
สงวนสิทธิ์ตามพระราชบัญญัติ
ห้ามคัดลอกเนื้อหาก่อนได้รับอนุญาต

ข้อมูลทางบรรณานุกรมหนังสือ

รุ่งอรุณ พรเจริญ.

การประยุกต์ใช้โปรแกรม MATLAB สำหรับการประมวลผลค่าและพื้นฐานการใช้งาน. /
รุ่งอรุณ พรเจริญ. กรุงเทพมหานคร : คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร, 2564.

219 หน้า : ภาพประกอบ, ตาราง.

1. โปรแกรม MATLAB 2. การประมวลผลค่า 3. พื้นฐานการใช้งาน.

จัดพิมพ์โดย มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร
คณะครุศาสตร์อุตสาหกรรม 399 ถนนสามเสน แขวงวิฑูรย์พัฒนา
เขตดุสิต กรุงเทพมหานคร 10300
โทร. 02-665-3777 ต่อ 7111
E-mail : Rungaroon.s@rmutp.ac.th

คำนำ

หนังสือ “การประยุกต์ใช้งานโปรแกรม MATLAB สำหรับการประมวลผลหาค่าและพื้นฐานการใช้งาน” เล่มนี้ เขียนจากประสบการณ์การใช้โปรแกรมคอมพิวเตอร์ของผู้เขียน และได้เรียบเรียงจากเอกสาร ตำราที่เกี่ยวข้องเพิ่มเติม เพื่อให้นักเรียน นักศึกษา หรือบุคคลที่สนใจงานไฟฟ้าและอิเล็กทรอนิกส์ได้ฝึกฝนและเข้าใจการเขียนโปรแกรม MATLAB ที่ช่วยให้การคำนวณทางคณิตศาสตร์ที่ซับซ้อน สามารถการเขียนโปรแกรมและการแสดงผลรวมอยู่โปรแกรมเดียวได้อย่างมีประสิทธิภาพและง่ายต่อการใช้งาน นอกจากนี้ยังมีการตอบสนองกับผู้ใช้ Interactive สำหรับคำนวณเชิงตัวเลขและการแสดงผลข้อมูล (Data Visualization) รวมทั้งมีความสามารถในการจัดรูปคำสั่งในลักษณะการเขียนโปรแกรมเพื่อใช้งานต่าง ๆ ทั้งทางวิทยาศาสตร์ และวิศวกรรม โดยแต่ละหัวข้อได้อธิบายหลักการทำงานของโปรแกรมและตัวอย่างการใช้งานอย่างละเอียด

สำหรับโปรแกรม MATLAB ที่ใช้ในหนังสือเล่มนี้ เป็นโปรแกรม MATLAB เวอร์ชัน R2021a สำหรับ academic เท่านั้น ซึ่งมหาวิทยาลัยเทคโนโลยีราชมงคลพระนครได้ซื้อลิขสิทธิ์ซอฟต์แวร์เป็นที่เรียบร้อยแล้ว

ผู้เขียนขอขอบพระคุณครูอาจารย์และเพื่อน ๆ รวมทั้งผู้ที่เกี่ยวข้องทุกท่าน ที่ทำให้ผู้เขียนได้นำความรู้และประสบการณ์การใช้โปรแกรมมาจัดทำเป็นหนังสือเล่มนี้ได้ ซึ่งหากมีข้อผิดพลาดประการใด ผู้เขียนขอภัยและน้อมรับคำติชมจากทุกท่าน

รุ่งอรุณ พรเจริญ

rungaroon.s@rmutp.ac.th

สารบัญ

	หน้า
คำนำ	ก
สารบัญ	ค
สารบัญภาพ	ช
สารบัญตาราง	ณ
บทที่ 1 การใช้งานโปรแกรม MATLAB เบื้องต้น	1
1.1 ส่วนประกอบของโปรแกรม	1
1.2 การใช้งานโปรแกรม MATLAB เบื้องต้น	3
1.3 ตัวดำเนินการต่าง ๆ	7
1.4 สรุป	12
แบบฝึกหัดบทที่ 1 การใช้งานโปรแกรม MATLAB เบื้องต้น	13
บทที่ 2 รูปแบบคำสั่งและฟังก์ชันเบื้องต้นของโปรแกรม MATLAB	15
2.1 การแสดงค่าข้อมูลในเชิงตัวเลข	15
2.2 ฟังก์ชันการคำนวณทางพีชคณิต	17
2.3 ฟังก์ชันการหาค่าทางตรีโกณมิติ และ Logarithm	19
2.4 สรุป	23
แบบฝึกหัดบทที่ 2 รูปแบบคำสั่งและฟังก์ชันเบื้องต้นของโปรแกรม MATLAB	24
บทที่ 3 การคำนวณและเปรียบเทียบโดยใช้โปรแกรม MATLAB	27
3.1 ตัวแปรวัตถุคลาส Symbolic	27
3.2 การแก้สมการในการคำนวณทางคณิตศาสตร์เชิงตัวแปร	31
3.3 สรุป	38
แบบฝึกหัดบทที่ 3 การคำนวณและเปรียบเทียบโดยใช้โปรแกรม MATLAB	39
บทที่ 4 การเขียนโปรแกรมย่อยด้วย M-File ของโปรแกรม MATLAB	41
4.1 การเขียนโปรแกรมโดยใช้ scrip M-File	41
4.2 การจัดการไฟล์และไดเรกทอรี	45
4.3 การเขียนโปรแกรมโดยใช้ function M-File	46
4.4 สรุป	48
แบบฝึกหัดบทที่ 4 การเขียนโปรแกรมย่อยด้วย M-File ของโปรแกรม MATLAB	49

สารบัญ (ต่อ)

	หน้า
บทที่ 5	
การเขียนโปรแกรม MATLAB แบบโครงสร้าง	53
5.1 การทำงานแบบตามลำดับ	53
5.2 การเลือกกระทำตามเงื่อนไข	55
5.3 การทำซ้ำ	62
5.4 สรุป	65
แบบฝึกหัดบทที่ 5 การเขียนโปรแกรม MATLAB แบบโครงสร้าง	66
บทที่ 6	
การเขียนโปรแกรม MATLAB วาดกราฟ 2 มิติ และ 3 มิติ	69
6.1 ฟังก์ชันการวาดกราฟ 2 มิติ และ 3 มิติ	69
6.2 การโปรแกรมและวาดกราฟ 2 มิติ	73
6.3 การโปรแกรมและวาดกราฟ 3 มิติ	79
6.4 สรุป	87
แบบฝึกหัดบทที่ 6 การเขียนโปรแกรม MATLAB วาดกราฟ 2 มิติ และ 3 มิติ	88
บทที่ 7	
ส่วนต่อประสานกราฟฟิคกับผู้ใช้ของโปรแกรม MATLAB	91
7.1 การเรียกใช้ GUI ด้วย GUIDE	91
7.2 การสร้าง GUI ด้วย GUIDE	95
7.3 การเขียนโปรแกรม GUI ด้วย GUIDE	98
7.4 การเขียน GUI ด้วย GUIDE	100
7.5 การรับค่าจากแป้นพิมพ์และเมาส์	109
7.6 การจัดลำดับ Tab และแถบเมนูเครื่องมือ	114
7.7 สรุป	121
แบบฝึกหัดบทที่ 7 ส่วนต่อประสานกราฟฟิคกับผู้ใช้ของโปรแกรม MATLAB	123
บทที่ 8	
การประยุกต์ใช้ GUI ของ MATLAB สำหรับการคำนวณสมการกฎของโอห์ม	127
8.1 วิธีการดำเนินการพัฒนา GUI	128
8.2 การเขียนโปรแกรมควบคุมการทำงานของวัตถุที่หน้าต่างเมนูหลัก	132
8.3 การเขียนโปรแกรมเพื่อควบคุมวัตถุสำหรับเชื่อมโยงไปยังโปรแกรมย่อย	137
8.4 การควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมคำนวณค่าแรงดันไฟฟ้า	141
8.5 การเขียนโปรแกรมเพื่อควบคุมวัตถุคำนวณค่าแรงดันไฟฟ้า	145
8.6 สรุป	149

สารบัญ (ต่อ)

	หน้า
แบบฝึกหัดบทที่ 8 การประยุกต์ใช้ GUI ของ MATLAB สำหรับการคำนวณสมการกฎของโอห์ม	150
บทที่ 9 การประยุกต์ใช้ GUI ของ MATLAB สำหรับโปรแกรมการคำนวณวงจรไฟฟ้า	153
9.1 การควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมการคำนวณวงจรไฟฟ้า	155
9.2 การแทรกภาพในหน้าต่างโปรแกรมการคำนวณวงจรไฟฟ้า	161
9.3 สรุป	166
แบบฝึกหัดบทที่ 9 การประยุกต์ใช้ GUI ของ MATLAB สำหรับโปรแกรมการคำนวณวงจรไฟฟ้า	167
บทที่ 10 การประยุกต์ใช้ GUI ของ MATLAB สำหรับการวิเคราะห์วงจรอันดับหนึ่ง RL และ RC	169
10.1 การออกแบบโปรแกรมจำลอง	171
10.2 การพัฒนาโปรแกรมจำลอง	173
10.3 การสร้างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL	175
10.4 การสร้างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RC	177
10.5 การเขียนโปรแกรมควบคุมการทำงานของวัตถุที่หน้าต่างเมนูหลัก	179
10.6 การควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL	182
10.7 สรุป	186
แบบฝึกหัดบทที่ 10 การประยุกต์ใช้ GUI ของ MATLAB สำหรับการวิเคราะห์วงจรอันดับหนึ่ง RL และ RC	187
บทที่ 11 การประยุกต์ใช้ Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์	189
11.1 การเรียกใช้ Simulink ของ MATLAB	190
11.2 ส่วนประกอบต่าง ๆ บนหน้าต่าง Library Simulink ของ MATLAB	191
11.3 การเรียกใช้บล็อกพื้นฐานของ Simulink ของ MATLAB	194
11.4 การโปรแกรม Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์	201
11.5 การปรับค่าโปรแกรม Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์	203

สารบัญ (ต่อ)

	หน้า
11.6 สรุป	209
แบบฝึกหัดบทที่ 11 การประยุกต์ใช้ Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์	211
บรรณานุกรม	215
ดัชนี	217
ประวัติผู้เขียน	219

สารบัญภาพ

ภาพที่	หน้า
1.1 หน้าต่างของโปรแกรม MATLAB	4
1.2 การกำหนดค่าตัวแปร M และ m	5
1.1 ค่าตัวแปรที่ถูกใช้งานบนหน้าต่าง Workspace	5
1.4 ข้อมูลของตัวแปร m	6
1.5 การกำหนดค่าตัวแปรนิพจน์ที่ต้องการ	6
1.6 การตรวจสอบข้อมูลของตัวแปร	7
1.7 การใช้ตัวดำเนินการ ตัวอย่างที่ 1.3	8
1.8 การใช้ตัวดำเนินการ ตัวอย่างที่ 1.4	8
1.9 การใช้เครื่องหมายพิเศษ [] และ , ในการกำหนดค่าให้ตัวแปร	9
1.10 การใช้เครื่องหมายพิเศษ [] และ ... ในการกำหนดค่าให้ตัวแปร	10
1.11 การใช้ตัวดำเนินการเปรียบเทียบตรรกะ	11
1.12 การใช้ฟังก์ชันคณิตศาสตร์พื้นฐานและค่าตัวแปรเฉพาะ	12
2.1 การใช้คำสั่งแสดงผลตัวเลขแบบ format short	16
2.2 การใช้คำสั่งแสดงผลตัวเลขแบบ format long e	16
2.3 การใช้คำสั่งแสดงผลตัวเลขแบบ format hex	16
2.4 การใช้ sqrt, floor และ round ในการคำนวณทางพีชคณิต	18
2.5 การใช้ floor, abs และ ceil ในการคำนวณทางพีชคณิต	19
2.6 การใช้คำสั่ง sin(x) และ asin(x) ในการคำนวณค่าทางตรีโกณมิติ	20
2.7 การใช้คำสั่ง cos(x) และ atan(x) ในการคำนวณค่าทางตรีโกณมิติ	21
2.8 การใช้คำสั่ง log(x) และ log10(x) ในการหาค่า Logarithm	22
2.9 การใช้คำสั่ง log(x) และ log10(x) ในการหาค่า Logarithm	22
3.1 การประกาศตัวแปรโดยใช้คำสั่ง sym และ syms	27
3.2 การระบุรูปแบบของตัวแปร	28
3.3 การดำเนินการทางพีชคณิตโดยใช้คลาส sym	28
3.4 การดำเนินการทางพีชคณิตโดยใช้คลาส syms	29
3.5 การจัดรูปแบบคำตอบโดยใช้ฟังก์ชัน pretty	29
3.6 ฟังก์ชันที่ช่วยลดรูปแบบของสมการ	30
3.7 ฟังก์ชัน collect เขียนนิพจน์ในรูปอย่างง่าย	30
3.8 การแก้สมการเพื่อหาค่าตัวแปรโดยใช้ฟังก์ชัน solve	31
3.9 การแก้สมการเพื่อหาค่าตัวแปรโดยใช้ฟังก์ชัน solve และ real	31
3.10 ฟังก์ชันที่รองรับการคำนวณคลาส sym	32
3.11 ฟังก์ชันที่รองรับการคำนวณคลาส syms	32
3.12 การหาค่าลิมิตฟังก์ชัน f(x) โดยใช้ syms ของโปรแกรม MATLAB	33

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.13 การหาค่าลิมิต เมื่อ x มีค่าลู่เข้าหาค่าตอบทางขวาโดยใช้ syms ของโปรแกรม MATLAB	34
3.14 การหาค่าลิมิต เมื่อ x มีค่าลู่เข้าหาค่าตอบทางซ้ายโดยใช้ syms ของโปรแกรม MATLAB	34
3.15 การหาค่าอนุพันธ์อันดับหนึ่งโดยใช้ syms ของโปรแกรม MATLAB การหาอนุพันธ์อันดับ n	35
3.16 การหาค่าอนุพันธ์อันดับสองโดยใช้ syms ของโปรแกรม MATLAB	35
3.17 การหาค่าอนุพันธ์ย่อยโดยใช้ syms ของโปรแกรม MATLAB	35
3.18 การหาค่าอนุพันธ์ย่อยอันดับ n โดยใช้ syms ของโปรแกรม MATLAB	36
3.19 การหาค่าอินทิเกรตแบบไม่จำกัดเขตโดยใช้ syms ของโปรแกรม MATLAB	36
3.18 การหาค่าอินทิเกรตแบบจำกัดเขตโดยใช้ syms ของโปรแกรม MATLAB	37
3.19 การหาค่าอินทิเกรตแบบจำกัดเขตของฟังก์ชัน $f(s,t)$ โดยใช้ syms ของโปรแกรม MATLAB	37
4.1 การเปิดหน้าต่าง M-File	42
4.2 การกำหนดตัวแปรบนหน้าต่าง M-File	42
4.3 การประมวลผลบนหน้าต่างคำสั่ง Command Window	43
4.4 การเขียน Script File และแสดงผลลัพธ์ของค่าความยาวด้านตรงข้ามมุมฉาก	43
4.5 การเขียน Script File และแสดงผลลัพธ์ของค่ากระแสไฟฟ้าในวงจร	44
4.6 การเขียน Script File และแสดงผลลัพธ์ของค่าความต้านทาน	44
4.7 หน้าต่าง SetPath	45
4.8 การเรียกใช้งาน Function M-File	46
4.9 หน้าต่างสำหรับการเขียน Function	46
4.10 การสร้าง Function M-File	47
4.11 การเรียกใช้ M-File ฟังก์ชัน	47
4.12 การเขียน Function File ของค่า Sine	48
4.13 การเขียน Function File คำนวณหาพื้นที่ของรูปสามเหลี่ยมและพื้นที่ของรูปสี่เหลี่ยมผืนผ้า	48
5.1 การเขียน Script File และแสดงผลลัพธ์ของค่า y ออกหน้าจอ Command Window	54
5.2 การเขียน Script File ของค่าพื้นที่วงกลม	54
5.1 การเขียน Script File ของค่าพื้นที่สี่เหลี่ยมคางหมู	55
5.4 การใช้คำสั่ง if	56
5.5 การแสดงผลลัพธ์บนหน้าจอ Command Window	56

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
5.6 การเขียนโปรแกรมตรวจสอบคะแนน	57
5.7 การเขียนโปรแกรมตรวจสอบคำตอบ	57
5.8 การใช้คำสั่ง if...else	58
5.9 การแสดงผลลัพธ์บนหน้าจอ Command Window คำสั่ง if...else	59
5.10 การใช้คำสั่ง if-elseif-else	59
5.11 การแสดงผลลัพธ์บนหน้าจอ Command Window คำสั่ง if-elseif-else	60
5.12 การใช้คำสั่ง switch-case-otherwise	60
5.13 การแสดงผลลัพธ์บนหน้าจอ Command Window คำสั่ง switch-case-otherwise	61
5.14 การเขียนโปรแกรมอ่านตัวเลข 1-7 เป็นวันในภาษาอังกฤษ	61
5.15 การใช้คำสั่ง for	62
5.16 การแสดงผลลัพธ์บนหน้าจอ Command Window คำสั่ง for	63
5.17 การเขียนโปรแกรมการหารากที่สองของตัวเลข 1 ถึง 5	63
5.18 การใช้คำสั่ง while	64
5.19 การแสดงผลลัพธ์บนหน้าจอ Command Window คำสั่ง while	64
6.1 กราฟ 2 มิติลักษณะต่าง ๆ	70
6.2 หน้าต่างความช่วยเหลือในการค้นหาคำสั่งการวาดกราฟ	71
6.3 ผลการค้นหาข้อมูลในหน้าต่าง Help ของการวาดกราฟ 2 มิติ	71
6.4 กราฟ 1 มิติลักษณะต่าง ๆ	72
6.5 ผลการค้นหาข้อมูลในหน้าต่าง Help ของการวาดกราฟ 3 มิติ	73
6.6 การเขียน Script File สำหรับวาดกราฟฟังก์ชัน sine	74
6.7 ผลลัพธ์ของการวาดกราฟฟังก์ชัน sin	74
6.8 การเขียน Script File สำหรับวาดกราฟฟังก์ชัน sine ให้เป็นเส้นประสีแดง และมีเครื่องหมายสี่เหลี่ยม	75
6.9 ผลลัพธ์ของการวาดกราฟฟังก์ชัน sin จากการแก้ไขโปรแกรม	75
6.10 วงจร RC อนุกรม	76
6.11 การเขียน Script File สำหรับวาดกราฟกระแสของวงจร RC อนุกรม	76
6.12 ผลลัพธ์ของการวาดกราฟกระแสของวงจร RC อนุกรม	77
6.13 การเขียน Script File สำหรับวาดกราฟฟังก์ชัน sine เทียบกับฟังก์ชัน cosine	78
6.14 ผลลัพธ์ของการวาดกราฟฟังก์ชัน sine เทียบกับฟังก์ชัน cosine	78
6.15 การเขียน Script File สำหรับวาดกราฟ 3 มิติ แบบเกลียว โดยใช้ฟังก์ชัน plot3	79
6.16 ผลลัพธ์ของการวาดเกลียวโดยใช้ฟังก์ชัน plot3	80

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
6.17 การเขียน Script File สำหรับวาดกราฟ 1 มิติ โดยใช้คำสั่ง mesh, meshc และ meshz	81
6.18 ผลลัพธ์ของการใช้ฟังก์ชัน mesh	81
6.19 ผลลัพธ์ของการใช้ฟังก์ชัน meshc	82
6.20 ผลลัพธ์ของการใช้ฟังก์ชัน meshz	82
6.21 การเขียน Script File สำหรับวาดกราฟ 3 มิติ โดยใช้คำสั่ง ezmesh	83
6.22 ผลลัพธ์ของการใช้ฟังก์ชัน ezmesh	83
6.23 การเขียน Script File สำหรับวาดกราฟ 3 มิติ โดยใช้คำสั่ง surf, surf1 และ surfc	84
6.24 ผลลัพธ์ของการใช้ฟังก์ชัน surf	84
6.25 ผลลัพธ์ของการใช้ฟังก์ชัน surf1	85
6.26 ผลลัพธ์ของการใช้ฟังก์ชัน surfc	85
6.27 การเขียน Script File สำหรับวาดกราฟ 3 มิติ โดยใช้คำสั่ง ezsurf และ ezsurfc	86
6.28 ผลลัพธ์ของการใช้ฟังก์ชัน ezsurf	86
6.29 ผลลัพธ์ของการใช้ฟังก์ชัน ezsurfc	87
7.1 หน้าต่าง GUIDE	93
7.2 หน้าต่างสำหรับเลือกสร้าง GUI ใหม่	93
7.3 หน้าต่าง Preference	94
7.4 หน้าต่างการสร้าง GUI ที่แสดงรายชื่อของวัตถุต่าง ๆ	94
7.5 หน้าต่าง GUIDE เลือกวัตถุในเมนู	96
7.6 หน้าต่างการบันทึกของ GUI	96
7.7 หน้าต่าง File ex_gui.m	97
7.8 หน้าต่าง File ex_gui.fig	97
7.9 การเปลี่ยนแปลงค่าคุณสมบัติของวัตถุ	98
7.10 การสร้างฟังก์ชันเรียกกลับด้วย GUIDE	99
7.11 เมนูย่อยเป็นรายการฟังก์ชันเรียกกลับทั้งหมดของปุ่ม Push Button	100
7.12 เมนูย่อยเป็นรายการฟังก์ชันเรียกกลับทั้งหมดของปุ่ม Edit Text	100
7.13 การเขียนโปรแกรมฟังก์ชันเรียกกลับโดยใช้ปุ่ม Push Button	101
7.14 การเขียน GUI ด้วย Push Button และ Axes	101
7.15 การเขียนโปรแกรมฟังก์ชันเรียกกลับโดยใช้ปุ่ม Button Group	102
7.16 การเขียน GUI โดยใช้ Radio Button และ Button Group	103
7.17 การเพิ่ม Static Text ในหน้าต่าง GUI	103

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
7.18 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Edit Text	104
7.19 การใช้ Edit Text และ Static Text	104
7.20 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Slider Bar	105
7.21 การใช้ Slider Bar และ Static Text	106
7.22 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ List Box	106
7.23 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Pop-up Menu	107
7.24 การแสดงการแก้ไขคุณสมบัติ String	107
7.25 หน้าต่าง String ขึ้นมาเพื่อแก้ไขตัวเลือก	108
7.26 การใช้ List Box และ Pop-up Menu	108
7.27 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Check Box	109
7.28 การทดสอบการทำงานของ Check Box	110
7.29 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Figure	110
7.30 ผลการรับค่าจากคีย์บอร์ด Alt จะแสดงที่ Static Text เป็น alt	110
7.31 การเขียนใช้คำสั่ง WindowButtonDownFcn	111
7.32 ผลการรัน GUI ของการใช้คำสั่ง WindowButtonDownFcn	112
7.33 การเขียนใช้คำสั่ง WindowButtonMotionFcn	112
7.34 ผลการรัน GUI ของการใช้คำสั่ง WindowButtonMotionFcn	113
7.35 การเขียนใช้คำสั่ง WindowScrollWheelFcn	113
7.36 ผลการรัน GUI ของการใช้คำสั่ง WindowScrollWheelFcn	114
7.37 หน้าต่าง Tab Order Editor	114
7.38 หน้าต่าง Menu Editor ในส่วนของ Menu Bar	115
7.39 การสร้างเมนู File	116
7.40 การสร้างเมนูย่อย New	116
7.41 การเขียนคำสั่งเพื่อให้เมนูย่อย New ทำงาน	117
7.42 ผลการรัน GUI แล้วเลือก File > New ที่เมนูบาร์	117
7.43 หน้าต่าง Menu Editor ในส่วนของ Context Menu	118
7.44 การสร้างเมนู Show Load และ Inverse	118
7.45 หน้าต่าง Inspector : figure(ex_gui)	119
7.46 การเขียนคำสั่งสำหรับเมนู Load	119
7.47 การเขียนคำสั่งสำหรับเมนู Inverse	120
7.48 การทดลองคลิกขวาภายในหน้าต่าง Figure แล้วเลือก Load	120
7.49 การทดลองคลิกขวาภายในหน้าต่าง Figure แล้วเลือก Inverse	121

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
8.1 การต่อวงจรเมื่อความต้านทานคงที่	128
8.2 ความสัมพันธ์ระหว่างแรงดันและกระแส	128
8.3 การต่อวงจรเมื่อความต้านทานไม่คงที่	129
8.4 ความสัมพันธ์ระหว่างกระแสและความต้านทานที่ไม่คงที่	129
8.5 การต่อวงจรไฟฟ้า	130
8.6 กราฟระหว่างกระแสและแรงดันของลวดโลหะ	130
8.7 การคำนวณสมการกฎของโอห์ม	131
8.8 กระบวนการทำงานของโปรแกรมจำลอง	132
8.9 การสร้างหน้าต่างเมนูของโปรแกรมคำนวณสมการกฎของโอห์ม	133
8.10 การสร้างชื่อของโปรแกรม Ohm's Laws Simulation	134
8.11 การสร้างปุ่มเชื่อมโยงหน้าต่างของโปรแกรมน้อย	135
8.12 การกำหนด M-File ของหน้าต่างเมนูหลัก	135
8.13 การแสดงผลการรันหน้าต่างเมนูหลัก	136
8.14 การเขียนโปรแกรมสำหรับใส่ภาพโลโก้	136
8.15 หน้าต่างเมนูที่แสดงโลโก้	137
8.16 การเชื่อมโยงไปยังหน้าต่างโปรแกรมการคำนวณแรงดันไฟฟ้า	137
8.17 การเชื่อมโยงไปยังหน้าต่างโปรแกรมการคำนวณกระแสไฟฟ้า	138
8.18 การเชื่อมโยงไปยังหน้าต่างโปรแกรมการคำนวณความต้านทาน	138
8.19 การเชื่อมโยงไปยังหน้าต่าง Help	139
8.20 การปิดหน้าต่างโปรแกรม	139
8.21 การเปลี่ยนแปลงชื่อในหน้าต่างของโปรแกรมเมนูหลัก	140
8.22 กระบวนการทำงานของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม	141
8.23 การออกแบบลักษณะของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม	142
8.24 หน้าต่าง M-File ของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม	143
8.25 การคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม	143
8.26 การเขียนฟังก์ชันการทำงานใส่สภาพสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม	144
8.27 ผลลัพธ์การใส่ภาพโปรแกรม	144
8.28 การเขียนโปรแกรมเพื่อคำนวณแรงดันไฟฟ้า	145

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
8.29 ผลการจำลองโปรแกรมสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม	146
8.30 การเขียนโปรแกรมไปยังหน้าต่างเมนูหลัก	146
8.31 ผลการรันปุ่ม Main	147
8.32 การเขียนโปรแกรมปุ่ม Clear	147
8.33 ผลการรันปุ่ม Clear	148
8.34 การเขียนโปรแกรมปุ่ม Exit	148
9.1 การต่อความต้านทานแบบอนุกรม	153
9.2 การต่อความต้านทานแบบขนาน	154
9.3 การต่อความต้านทานแบบผสม	154
9.4 กระบวนการทำงานของโปรแกรม	155
9.5 การออกแบบลักษณะของโปรแกรมคำนวณวงจรไฟฟ้า	156
9.6 หน้าต่าง M-File ของโปรแกรมคำนวณวงจรไฟฟ้า	158
9.7 โลโก้ของคณะ	158
9.8 การเขียนฟังก์ชันการทำงานใส่ภาพโลโก้ของคณะ	159
9.9 ผลลัพธ์การใส่โลโก้ของคณะในโปรแกรม	159
9.10 การต่อวงจรแบบอนุกรมและสมการการคำนวณ	160
9.11 การเขียนโปรแกรมเพื่อเลือกการต่อวงจรแบบอนุกรม	160
9.12 ผลการจำลองโปรแกรมคำนวณวงจรไฟฟ้า	161
9.13 การเขียนโปรแกรมการต่อวงจรแบบอนุกรมที่ปุ่ม Calculate	162
9.14 ผลการจำลองโปรแกรมการต่อวงจรแบบอนุกรม	164
9.15 การเขียนโปรแกรมลบค่าต่าง ๆ	165
9.16 ผลการรันปุ่ม Clear	165
9.17 การเขียนโปรแกรมปิดหน้าต่างเมนู	166
10.1 การออกแบบโปรแกรมจำลอง	172
10.2 การสร้างปุ่มหน้าต่างเมนูหลัก	173
10.3 M-file ของหน้าต่างเมนูหลัก	174
10.4 การออกแบบหน้าจอโปรแกรมย่อยการวิเคราะห์วงจรอันดับหนึ่ง RL	175
10.5 การออกแบบหน้าจอโปรแกรมย่อยการวิเคราะห์วงจรอันดับหนึ่ง RC	177
10.6 หน้าต่างเมนูหลัก	179
10.7 การเขียนโปรแกรมการใส่ภาพโลโก้	180
10.8 หน้าต่างเมนูแสดงภาพโลโก้	180
10.9 การเชื่อมโยงไปยังหน้าต่างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL	181
10.10 กระบวนการทำงานของโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL	182

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
10.11 หน้าต่างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL เมื่อเปิดโปรแกรมเป็นครั้งแรก	183
10.12 ขั้นตอนการใส่ภาพสถานะของวงจร	183
10.13 สถานะของวงจรในโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL เมื่อเปิดโปรแกรมเป็นครั้งแรก	184
10.14 การแสดงสถานะของวงจรอันดับหนึ่ง RL แบบ Pure Resistor Circuit	186
11.1 การเรียกใช้งาน Simulink ของโปรแกรม	190
11.2 หน้าต่าง Library ของโปรแกรม	190
11.3 หน้าต่าง Simulink Library Browser	191
11.4 รายละเอียดของ Block ในหมวดหมู่ Sources	192
11.5 รายละเอียดของ Block ในหมวดหมู่ Math Operations	193
11.6 รายละเอียดของ Block ในหมวดหมู่ Sinks	193
11.7 ไอคอนสำหรับเปิดแผ่นงานใหม่เพื่อสร้างแบบจำลอง	194
11.8 แผ่นงานใหม่เพื่อสร้างแบบจำลอง	194
11.9 การนำ Block Sine Wave จาก Library กลุ่ม Source	195
11.10 ไดอะแกรมของการใช้ Sine wave และ Scope	196
11.11 พารามิเตอร์ของ Block Sine Wave	196
11.12 พารามิเตอร์ของการจำลองการทำงาน	197
11.13 การสั่งให้ Simulink จำลองการทำงานของระบบที่สร้างขึ้น	198
11.14 ผลการจำลองการทำงานของระบบที่สร้างขึ้น	198
11.15 การนำ Block ต่าง ๆ จาก Library Browser	199
11.16 ค่าพารามิเตอร์ของ Block Cosine Wave และ Sine Wave	200
11.17 ค่าพารามิเตอร์ Start time และ Stop time	200
11.18 ผลการจำลองการทำงานของระบบ	201
11.19 การเปิดโปรแกรม Simulink ของ MATLAB	202
11.20 การทดสอบโดยใช้ความถี่สุ่มตัวอย่างสัญญาณ $f_s = 100 \text{ Hz}$	202
11.21 การใส่ค่าภายใน Simulation Parameters ของ Block Sine Wave	203
11.22 การใส่ค่าภายใน Simulation Parameters ของ Block Pulse Generator	204
11.23 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 100 \text{ Hz}$ วัดจากสัญญาณ scope	204
11.24 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 100 \text{ Hz}$ วัดจากสัญญาณ scope1	205
11.25 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 100 \text{ Hz}$ วัดจากสัญญาณ scope2	205
11.26 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 20 \text{ Hz}$ วัดจากสัญญาณ scope	206
11.27 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 20 \text{ Hz}$ วัดจากสัญญาณ scope1	207
11.28 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 20 \text{ Hz}$ วัดจากสัญญาณ scope2	207

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
11.29	ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 15 \text{ Hz}$ วัดจากสัญญาณ scope	208
11.30	ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 15 \text{ Hz}$ วัดจากสัญญาณ scope1	209
11.31	ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 15 \text{ Hz}$ วัดจากสัญญาณ scope2	209
11.28	การออกแบบโมเดลการสุ่มตัวอย่างสัญญาณ (Sampling)	

สารบัญตาราง

ตารางที่	หน้า	
1.1	ตัวดำเนินการ	7
1.2	เครื่องหมายพิเศษ	9
1.3	ตัวดำเนินการเปรียบเทียบตรรกะ	10
1.4	ฟังก์ชันคณิตศาสตร์พื้นฐานและค่าตัวแปรเฉพาะ	11
2.1	การแสดงค่าข้อมูลในเชิงตัวเลขของโปรแกรม MATLAB	15
2.2	ฟังก์ชันในการคำนวณทางพีชคณิต	17
2.3	ฟังก์ชันในการคำนวณค่าทางตรีโกณมิติ	19
2.4	ฟังก์ชันที่ใช้สำหรับการหาค่า Logarithm	21
3.1	ฟังก์ชันที่ช่วยลดรูปแบบของสมการ	29
3.2	ฟังก์ชันที่รองรับการคำนวณคลาส sym หรือ syms	31
6.1	ฟังก์ชันรูปแบบการวาดกราฟ 2 มิติ	69
6.2	รูปแบบ สัญลักษณ์ และสีที่ใช้ในการวาดกราฟ	70
10.1	การวิเคราะห์วงจรอันดับหนึ่ง RL และ RC	170
11.1	การเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ในกรณีที่ 1	203
11.2	การเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ในกรณีที่ 2	206
11.3	การเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ในกรณีที่ 1	208

บทที่ 1

การใช้งานโปรแกรม MATLAB เบื้องต้น

MATLAB : Matrix Laboratory เป็นโปรแกรมคอมพิวเตอร์สำหรับช่วยให้นักคณิตศาสตร์ที่ซับซ้อน ซึ่งเทคนิคของ MATLAB ได้รวมการคำนวณ การเขียนโปรแกรมและการแสดงผลรวมอยู่โปรแกรมเดียวได้อย่างมีประสิทธิภาพและง่ายต่อการใช้งาน นอกจากนี้การทำงานของ MATLAB ทำงานลักษณะการตอบสนองกับผู้ใช้ Interactive สำหรับคำนวณเชิงตัวเลข และการแสดงผลข้อมูล (Data Visualization) รวมทั้งมีความสามารถในการจัดรูปคำสั่งในลักษณะการเขียนโปรแกรมเพื่อใช้งานต่าง ๆ ทั้งทางวิทยาศาสตร์ และวิศวกรรม

ก่อนใช้งานโปรแกรม MATLAB ผู้อ่านควรศึกษาและทำความเข้าใจเกี่ยวกับส่วนประกอบของโปรแกรมและหน้าต่างของโปรแกรม เพื่อเลือกใช้งานของโปรแกรมได้อย่างเหมาะสมกับการประยุกต์ใช้งานต่อไป

1.1 ส่วนประกอบของโปรแกรม MATLAB

ปัจจุบันคอมพิวเตอร์ได้สร้างโปรแกรมมาเพื่อใช้ในการแก้ไขปัญหาในเชิงตัวเลข โดยมีภาษาทางคอมพิวเตอร์หลายตัวที่ช่วยในการแก้ปัญหา เช่น ภาษา C, Fortran, Pascal ในการแก้ปัญหาเชิงตัวเลขและกราฟิกที่มีความซับซ้อนค่อนข้างจะยุ่งยากและเสียเวลามาก เพราะต้องใช้คำสั่งเป็นจำนวนมาก และมีรูปแบบคำสั่งที่แน่นอน บริษัท Math Works Inc. จึงได้พัฒนาโปรแกรมที่มีชื่อว่า MATLAB เพื่อใช้ในการคำนวณเชิงตัวเลขและกราฟิกที่ซับซ้อนให้ง่ายต่อการใช้งานมีความรวดเร็วและการเขียนโปรแกรมไม่ยุ่งยาก

โปรแกรม MATLAB เป็นภาษาคอมพิวเตอร์ขั้นสูง (High Level Language) สำหรับคำนวณทางเทคนิคที่ประกอบด้วยการคำนวณเชิงตัวเลข กราฟิกที่ซับซ้อนและการจำลองแบบเพื่อให้มองเห็นภาพพจน์ได้ง่ายและชัดเจนชื่อของ MATLAB เดิมโปรแกรม MATLAB ได้เขียนขึ้นเพื่อใช้ในการคำนวณทาง Matrix ที่พัฒนาจากโปรเจกต์ชื่อ LINKPACK และ EISPACK

โปรแกรม MATLAB จะมีกล่องเครื่องมือที่ใช้ในการหาคำตอบที่เรียกว่า Toolbox โดยโปรแกรม MATLAB จะมี Toolbox ในแต่ละสาขา เช่น การประมวลผลสัญญาณ การประมวลผลภาพ ระบบควบคุม โครงข่ายประสาท พืชซีลोजิก เวฟเลท การติดต่อสื่อสาร สถิติ และสาขาอื่น ๆ มากมาย ภายใน Toolbox แต่ละสาขาจะมีฟังก์ชันต่าง ๆ ที่เกี่ยวข้องกับการแก้ปัญหาในสาขานั้น ๆ ให้เลือกประยุกต์ใช้งานเป็นจำนวนมาก

การทำงานของโปรแกรม MATLAB จะสามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรง (Interactive) คือการเขียนคำสั่งเข้าไปทีละคำสั่งเพื่อให้ MATLAB ประมวลผลไปเรื่อย ๆ หรือสามารถ

ที่จะรวบรวมชุดคำสั่งนั้น เป็นโปรแกรมสามารถทำได้ ข้อสำคัญอย่างหนึ่งของ MATLAB คือ ข้อมูลทุกตัวอยู่ในลักษณะของ Array คือ ในแต่ละตัวแปรจะได้รับการแบ่งส่วนย่อยเล็ก ๆ ขึ้น (หรืออาจจะได้รับการแบ่ง Element) ซึ่งการใช้ตัวแปร Array ใน MATLAB นั้น จำเป็นต้องจอง Dimension เหมือนกับการเขียนโปรแกรมในภาษาอื่นทั่วๆไป ซึ่งทำให้สามารถที่แก้ไขปัญหาของตัวแปรที่อยู่ในลักษณะของ Matrix และ Vector โดยง่าย ทำให้สามารถลดเวลาของการทำลงไปได้อย่างมากเมื่อเทียบกับการเขียนโปรแกรมโดย C หรือ Fortran

ส่วนประกอบของโปรแกรม MATLAB ประกอบด้วย 5 ส่วน คือ

1) ภาษาโปรแกรม MATLAB

MATLAB เป็นโปรแกรมที่ใช้ควบคุม Flow Statement ฟังก์ชันโครงสร้างข้อมูลอินพุท เอาท์พุท และลักษณะโปรแกรม Object-Oriented Programming ทำให้การเขียนโปรแกรมไม่ยุ่งยากเมื่อเทียบกับการเขียนโปรแกรมด้วยภาษาอื่น ๆ เช่น C, Fortran, Basic เป็นต้น

2) สถาปัตยกรรมในการทำงานของ MATLAB

MATLAB จะมีกลุ่มของเครื่องมือที่มีเป็นประโยชน์สำหรับการทำงานของผู้ใช้โปรแกรม หรือ โปรแกรมเมอร์ประโยชน์ที่กล่าวนี้คือการจัดการตัวแปรใน Workspace การนำข้อมูลหรือการผ่านค่าตัวแปรเข้า ออกและกลุ่มของเครื่องมือต่าง ๆ นี้จะใช้สำหรับพัฒนาจัดการตรวจสอบความผิดพลาดของโปรแกรม Debugging ที่ได้เขียนขึ้น

3) ฟังก์ชันในการคำนวณทางคณิตศาสตร์

MATLAB จะมีไลบรารีทั่วๆไปที่ใช้ในการคำนวณอย่างกว้างขวาง เช่น Sine, Cosine และ ฟังก์ชันเชิงซ้อน โดยสามารถนำไปประยุกต์ใช้เป็นฟังก์ชันหรือไลบรารีเพิ่มเติมขึ้นจากไลบรารีที่ใช้กัน โดยทั่วๆไป เช่น ฟังก์ชันในการหา Eigenvalues และ Eigenvectors การแยกตัวประกอบและส่วนประกอบของเมทริกซ์ด้วยวิธีต่าง ๆ การวิเคราะห์ข้อมูล การหาความน่าจะเป็น และการแก้ปัญหา ระบบของสมการเชิงเส้นที่เป็นพื้นฐานของสาขาวิชาต่าง ๆ เป็นต้น ทำให้โปรแกรม MATLAB มีฟังก์ชันสำหรับใช้งานค่อนข้างมากและครอบคลุมในรายละเอียดของการคำนวณสาขาต่าง ๆ ได้มากขึ้น

4) Handle Graphics

ระบบกราฟิกของ MATLAB จะประกอบด้วยคำสั่งชั้นสูงสำหรับการพล็อตกราฟ โดยมีพื้นฐานอยู่บนแนวความคิดที่ว่าทุก ๆ สิ่งบนหน้าต่างรูปภาพของโปรแกรม MATLAB จะเป็นวัตถุ (Object) ซึ่งมีเอกลักษณ์เฉพาะตัว Handle Graphics ประกอบด้วยคำสั่งชั้นสูงให้ได้เลือกใช้ในการสร้าง Graphic User Interface บนพื้นฐานการประยุกต์ใช้งาน นอกจากนี้โปรแกรม MATLAB ยังมีฟังก์ชันที่ใช้ภาพสองมิติ ภาพสามมิติ และการสร้างภาพเคลื่อนไหว

5) The MATLAB Application Program Interface (API)

API จะใช้เพื่อสนับสนุนการติดต่อจากภายนอกโดยใช้โปรแกรมที่เป็น Mix ไฟล์ซึ่งเป็นไฟล์ที่เขียนขึ้นโดยใช้ Mix ฟังก์ชันใน MATLAB ซึ่งจะเรียกใช้จากโปรแกรมภาษา C และ Fortran หรืออาจกล่าวได้ว่า API เป็นไลบรารีที่เขียนด้วยโปรแกรมภาษา C และ Fortran ที่มีการเชื่อมต่อโปรแกรม MATLAB ด้วยไฟล์ที่เป็น Mix ฟังก์ชันอีกทั้ง MATLAB API นี้ยังมีความสามารถสำหรับเรียก routine จาก MATLAB

นอกจากลักษณะเด่นของโปรแกรม MATLAB ที่ได้กล่าวข้างต้นแล้ว โปรแกรม MATLAB ยังมีเครื่องมือที่ใช้สำหรับการวิเคราะห์และทดสอบระบบโดยการจำลองขึ้นมา คือ Simulink ซึ่งเป็นโปรแกรมที่ควบคู่กับ MATLAB ซึ่งเป็นระบบ Interactive สำหรับการจำลองและวิเคราะห์ระบบไดนามิกต่าง ๆ ที่เป็นระบบเชิงเส้นและระบบไม่เชิงเส้น

Simulink เป็นโปรแกรม Mouse-Driver ที่ใช้ระบบโมเดลโดยการวาดบล็อกไดอะแกรมบนจอภาพด้วยการใช้เมาส์ ทำให้โปรแกรม MATLAB สามารถทำการจำลองระบบได้หลายรูปแบบเช่น เชิงเส้น ไม่เชิงเส้น เวลาต่อเนื่อง เวลาไม่ต่อเนื่อง และระบบหลายอัตรา ซึ่งแต่ละรูปแบบที่นำมาสร้างแบบจำลองในการวิเคราะห์นี้ผู้ใช้จะต้องมีความเข้าใจพื้นฐานการทำงานของบล็อกแต่ละบล็อกได้เป็นอย่างดี ตลอดจนเข้าใจระบบโดยรวมของงานที่จะกระทำการ

Blocksets เป็นสิ่งที่เพิ่มเติมใน Simulink โดยจะเป็นไลบรารีของบล็อก สำหรับการประยุกต์เฉพาะเช่น การติดต่อสื่อสาร การประมวลผลข้อมูล และระบบไฟฟ้ากำลัง Real-time Workshop เป็นโปรแกรมที่ให้คุณสร้าง C Code จากบล็อกไดอะแกรม และสามารถกระทำกับบล็อกไดอะแกรมได้หลากหลายด้วยระบบเวลาจริง

โปรแกรม MATLAB เป็นโปรแกรมที่ใช้สำหรับการคำนวณทางคณิตศาสตร์และกราฟิกที่ซับซ้อน ดังนั้นจึงจำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่มีความเร็วสูง คอมพิวเตอร์พีซีที่เหมาะสมกับต้องมีซีพียูรุ่นเพนเทียมขึ้นไป แรมควรมีอย่างน้อย 32 เมกกะไบต์ ส่วนฮาร์ดดิสก์ควรมีเนื้อที่ว่างเกินกว่า 80 เมกกะไบต์ขึ้นไป

1.2 การใช้งานโปรแกรม MATLAB เบื้องต้น

1.2.1 รายละเอียดของ Menu Program MATLAB

หลังจากเปิดโปรแกรม MATLAB จะมีหน้าต่างส่วนต่าง ๆ ประกอบด้วย Command Window, Current Folder, Workspace และ Details ดังภาพที่ 1.1

1. หน้าต่าง Command Window

เป็นส่วนที่ใช้ในการป้อนชุดคำสั่งเพื่อให้ MATLAB ทำงานตามคำสั่งนั้น ๆ และผลที่เป็นตัวหนังสือในหน้าต่างนี้ มีลักษณะการทำงานเป็นบรรทัด โดยโปรแกรม MATLAB จะพร้อมใช้งานเมื่อมีเครื่องหมาย prompt (>>) ซึ่งเครื่องหมายนี้เป็นการถามหาพร้อมรับคำสั่งต่าง ๆ เพื่อทำการคำนวณและประมวล ซึ่งปุ่มที่สำคัญต่าง ๆ

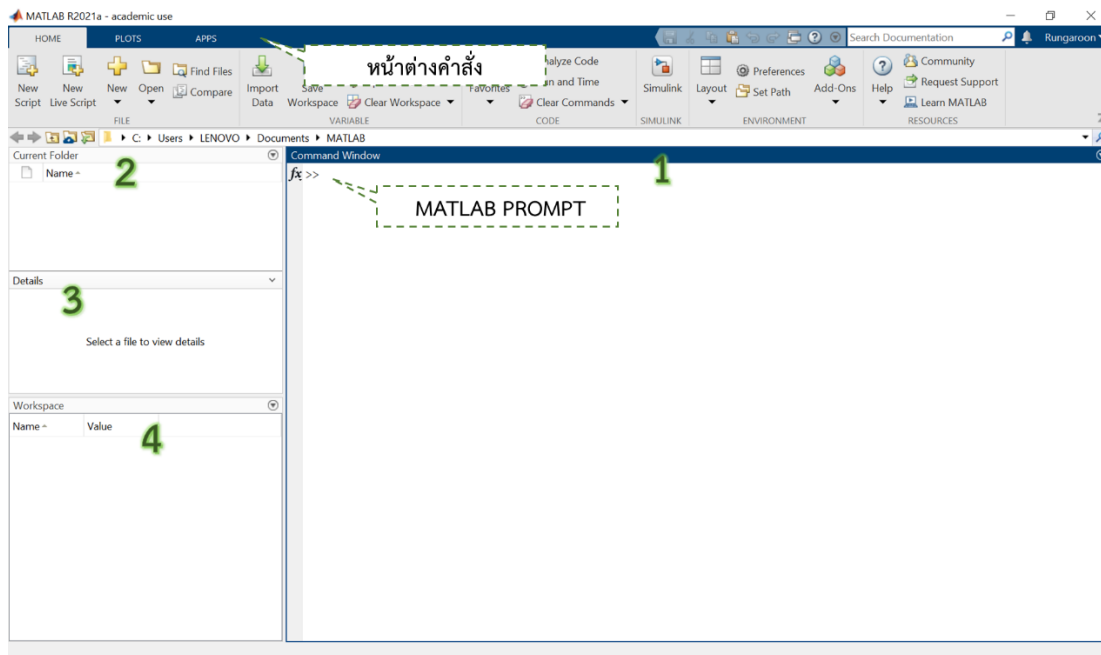
2. หน้าต่าง Current Folder

เป็นจุดเริ่มต้นของการทำงานและเป็นพื้นที่ในการค้นหาข้อมูลหรือคำสั่งต่าง ๆ ตามที่ได้รับคำสั่ง โดยการค้นหาจำกัดวงอยู่เฉพาะสองส่วนหลักเท่านั้น MATLAB จะไม่มีการค้นหา File หรือคำสั่งต่าง ๆ นอกพื้นที่ดังกล่าว ดังนั้น คำสั่งหรือ M-File ต่าง ๆ ที่ต้องการใช้งานนั้น จำเป็นต้องอยู่ใน Current Directory เสมอ

3. หน้าต่าง Details

เป็นหน้าต่างที่ Toolbox ต่าง ๆ ที่ติดตั้งไว้ในเครื่อง และสามารถเข้าเครื่องมือตัวอย่าง และเอกสารที่เกี่ยวข้องกับโปรแกรม MATLAB หรือ Toolbox ต่าง ๆ ได้ง่าย

4 การประยุกต์ใช้งานโปรแกรม MATLAB สำหรับการประมวลผลค่าและพื้นฐานการใช้งาน



ภาพที่ 1.1 หน้าต่างของโปรแกรม MATLAB

4. หน้าต่าง Workspace

เป็นหน้าต่างที่ใช้ดูว่าตัวแปรอะไรบ้างที่มีอยู่ในพื้นที่ทำงาน ซึ่งสามารถแก้ไขและลบตัวแปรต่าง ๆ ได้จากหน้าต่างนี้

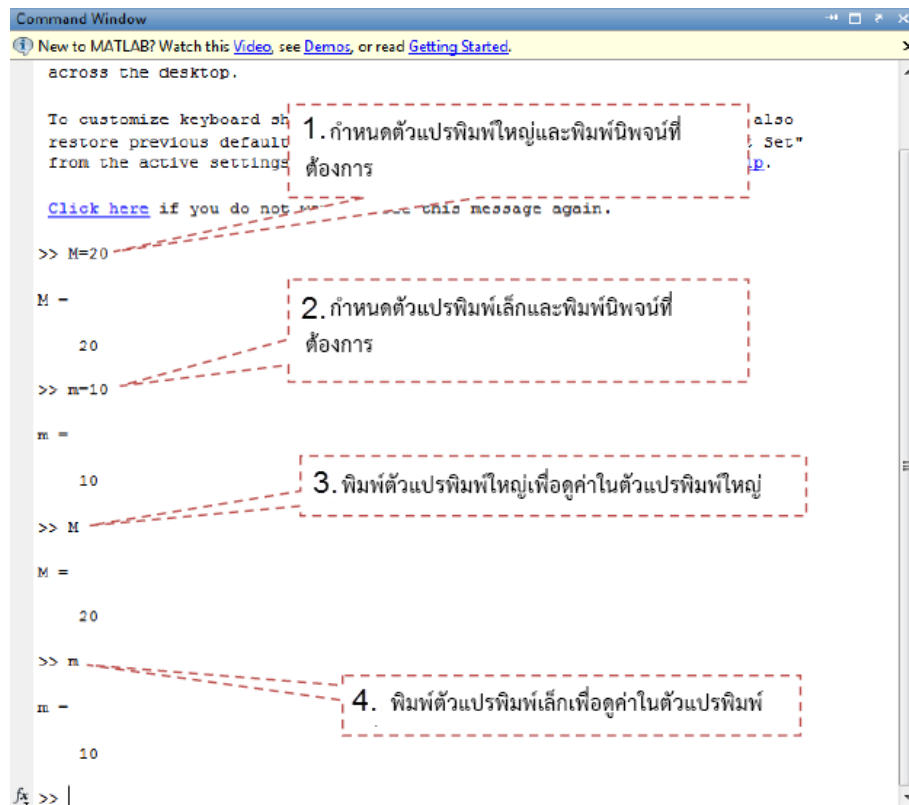
เนื่องจากโปรแกรม MATLAB มีลักษณะเป็น Interactive Mode จึงทำให้เมื่อพิมพ์คำสั่งในการคำนวณและประมวลผลโดยตรงที่เครื่องหมาย >> บนหน้าต่างคำสั่งแล้ว กดปุ่ม Enter บนคีย์บอร์ด โปรแกรม MATLAB จะทำการคำนวณแล้วให้คำตอบทันทีบนหน้าต่างคำสั่ง

1.2.2 การกำหนดค่าให้กับตัวแปร

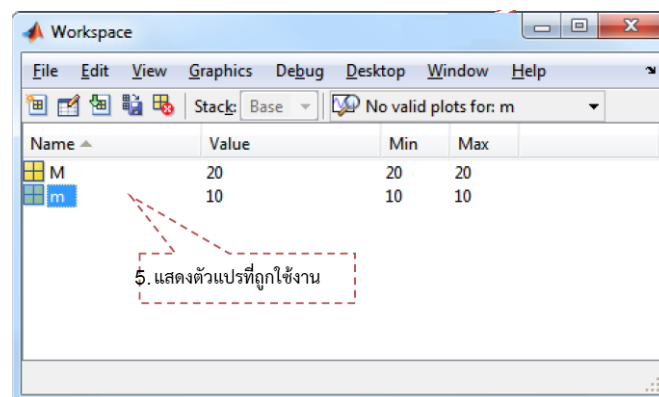
การกำหนดค่าให้กับตัวแปรในโปรแกรม MATLAB นั้นสามารถทำได้โดยการใช้เครื่องหมายเท่ากับ (=) แล้วตามด้วยค่าข้อมูลหรือนิพจน์ที่ต้องการให้กับตัวแปร โดยตัวแปรที่เป็นตัวพิมพ์ใหญ่กับตัวพิมพ์เล็กจะเป็นตัวแปรคนละตัวกัน

ตัวอย่างที่ 1.1

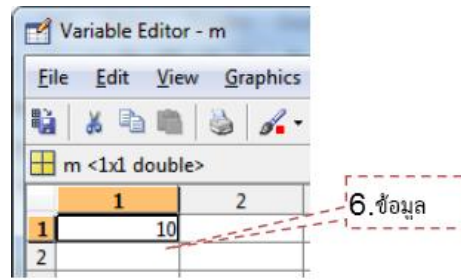
กำหนดให้ตัวแปร $M = 10$ และ $m = 20$ เมื่อทำการเปิดโปรแกรม MATLAB แล้ว ให้พิมพ์ค่าที่กำหนด และสังเกตตัวแปรที่ถูกใช้งานที่หน้าต่าง Workspace ดังภาพที่ 1.2 – ภาพที่ 1.4



ภาพที่ 1.2 การกำหนดค่าตัวแปร M และ m



ภาพที่ 1.3 การค่าตัวแปรที่ถูกใช้งานบนหน้าต่าง Workspace

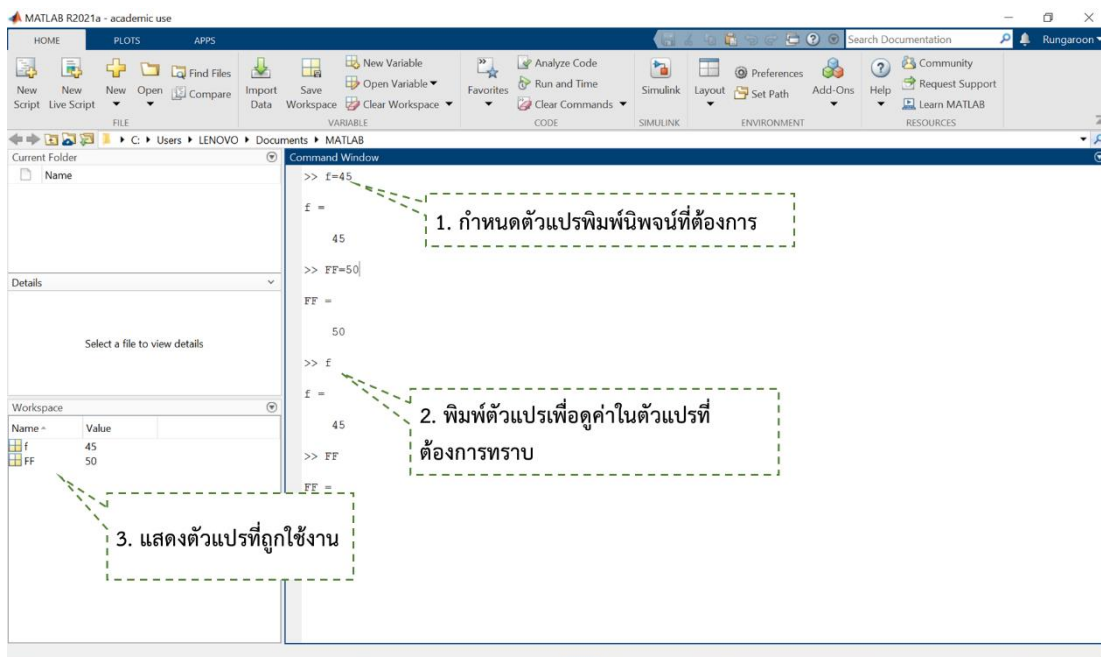


	1	2
1	10	
2		

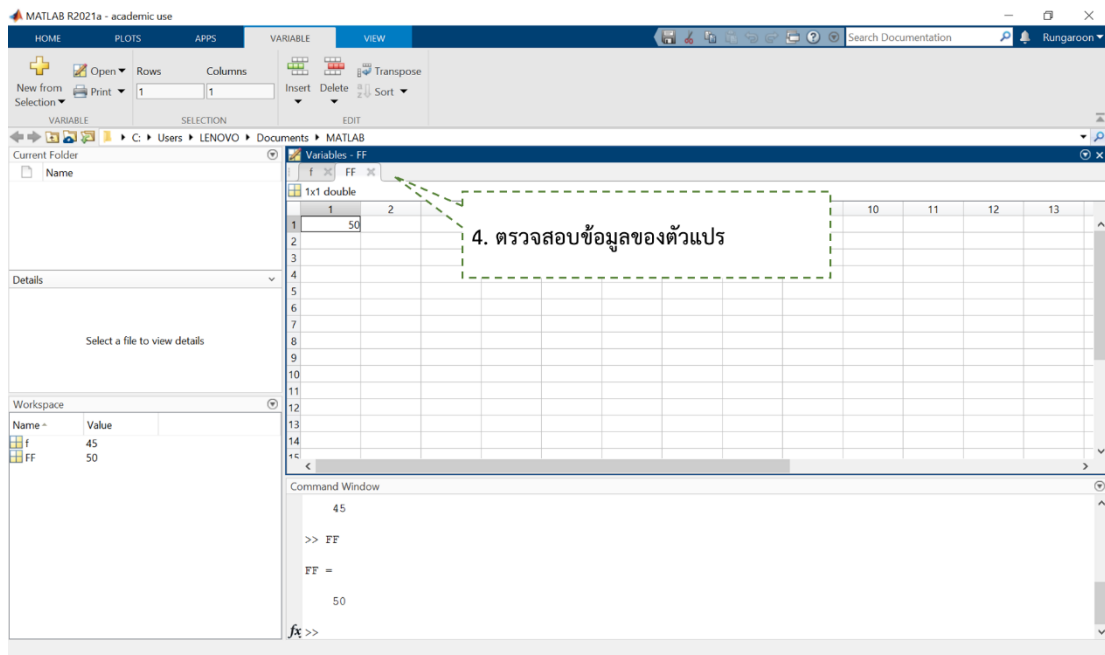
ภาพที่ 1.4 ข้อมูลของตัวแปร m

ตัวอย่างที่ 1.2

กำหนดให้ตัวแปร $f = 45$ และ $FF = 50$ เมื่อทำการเปิดโปรแกรม MATLAB แล้ว ให้พิมพ์ค่าที่กำหนด และสังเกตตัวแปรที่ถูกใช้งานที่หน้าต่าง Workspace ดังภาพที่ 1.5 – ภาพที่ 1.6



ภาพที่ 1.5 การกำหนดค่าตัวแปรนิพจน์ที่ต้องการ



ภาพที่ 1.6 การตรวจสอบข้อมูลของตัวแปร

1.3 ตัวดำเนินการ

1.3.1 ตัวดำเนินการ

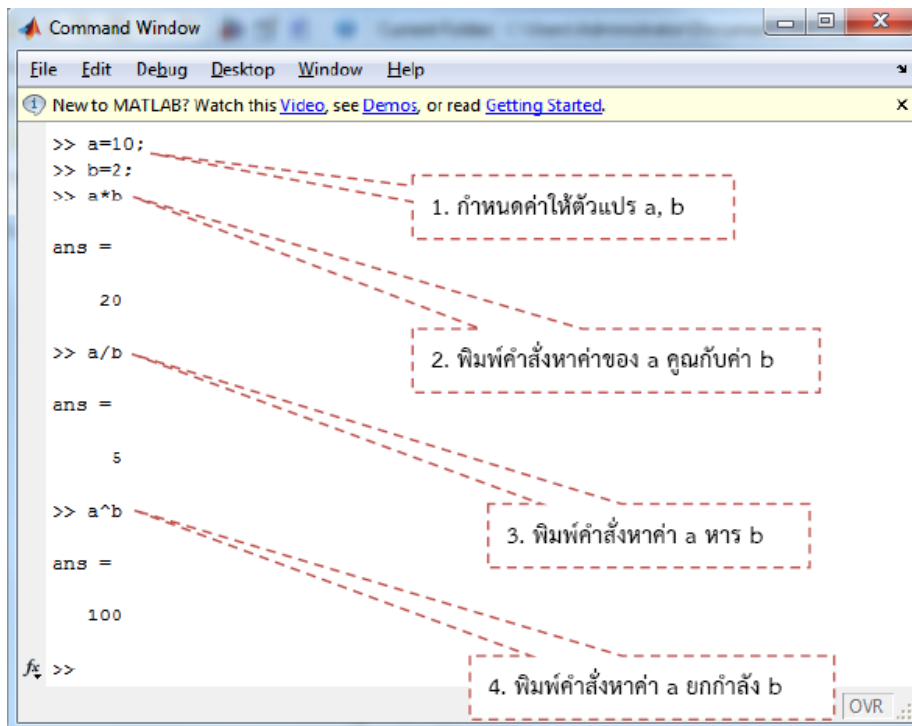
โปรแกรม MATLAB ได้กำหนดตัวดำเนินการ เพื่ออำนวยความสะดวกในการเขียนโปรแกรม ดังตารางที่ 1.1

ตารางที่ 1.1 ตัวดำเนินการ

ลักษณะดำเนินการ	ตัวดำเนินการ	รูปแบบของ MATLAB
การบวก	+	A+B
การลบ	-	A-B
การคูณ	*	A*B
การคูณเชิงสมาชิก	.*	A.*B
การหารทางขวา	/	A/B
การหารทางซ้าย	\	A\B
การหารเชิงสมาชิก	./	A./B
การยกกำลัง	^	A^B
การยกกำลังเชิงสมาชิก	.^	A.^B

ตัวอย่างที่ 1.3 กำหนดให้ตัวแปร $a = 10$ และ $b = 2$ จงหา

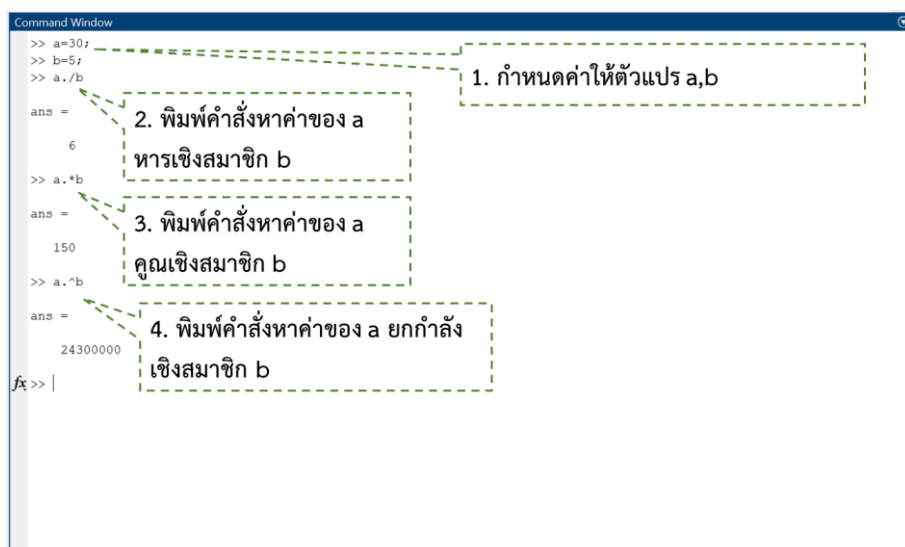
1. $a \times b$ 2. $a \div b$ 3. a^b ดังภาพที่ 1.7



ภาพที่ 1.7 การใช้ตัวดำเนินการ ตัวอย่างที่ 1.3

ตัวอย่างที่ 1.4 กำหนดให้ตัวแปร $a = 30$ และ $b = 5$ จงหา

1. $a ./ b$ 2. $a .* b$ 3. $a.^b$ ดังภาพที่ 1.8



ภาพที่ 1.8 การใช้ตัวดำเนินการ ตัวอย่างที่ 1.4

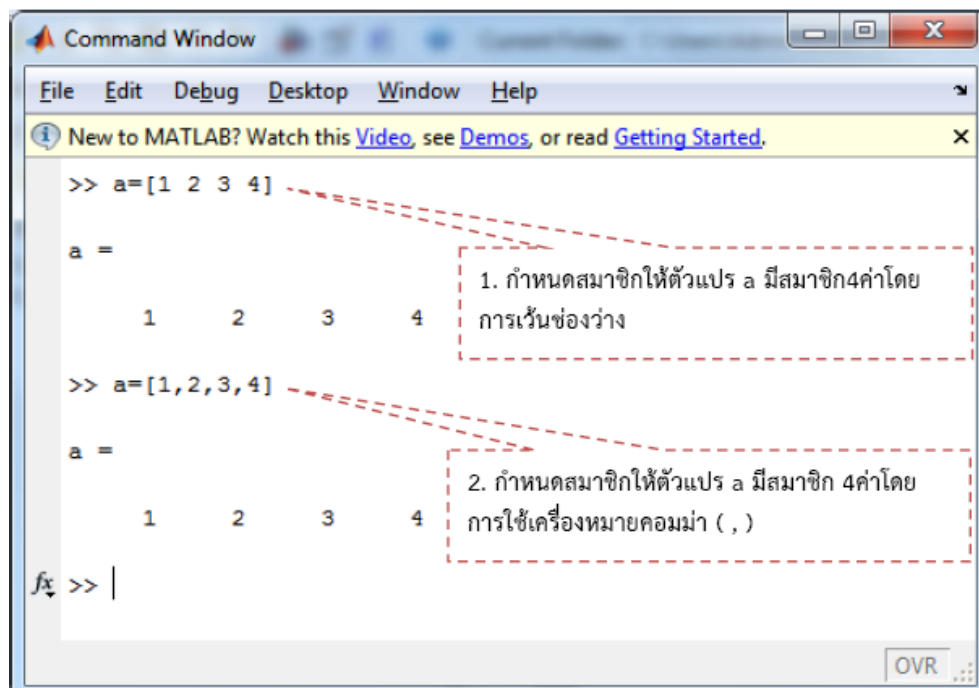
1.3.2 เครื่องหมายพิเศษ

โปรแกรม MATLAB ได้กำหนดเครื่องหมายพิเศษ เพื่ออำนวยความสะดวกในการเขียนโปรแกรม ดังตารางที่ 1.2

ตารางที่ 1.2 เครื่องหมายพิเศษ

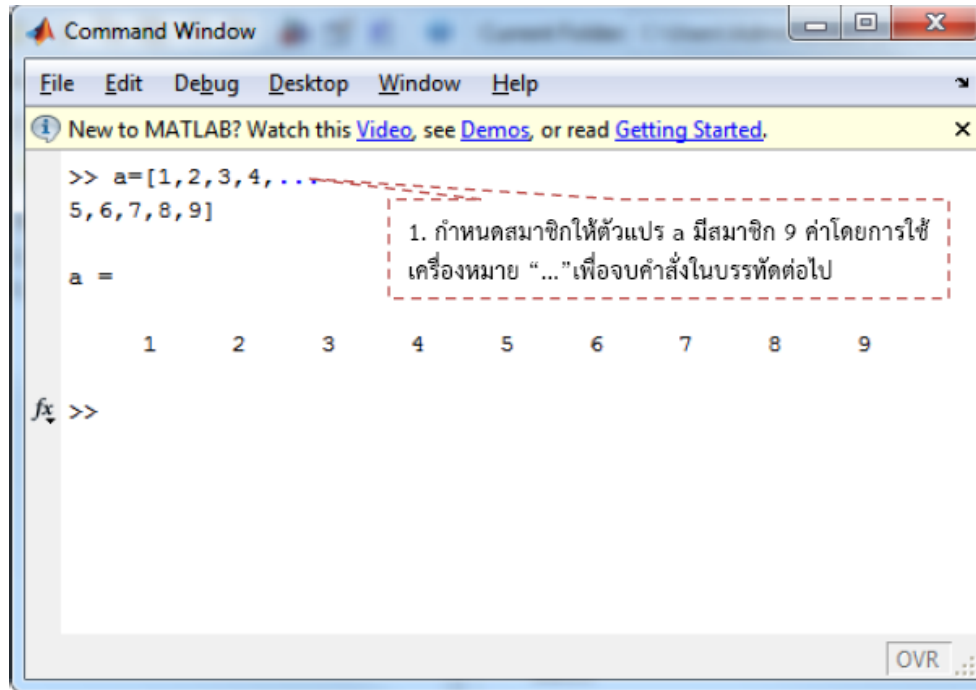
เครื่องหมาย	รายละเอียด
.	จุดทศนิยม
()	กำหนด Subscripts
=	กำหนดค่า
[]	สร้างเวกเตอร์และเมตริกซ์
:	สร้างเวกเตอร์
...	กระทำคำสั่งยังบรรทัดต่อไป
,	แยกตัวแปรภายในเมตริกซ์และตัวแปรย่อย
;	จบบรรทัด

ตัวอย่างที่ 1.5 กำหนดค่าให้ตัวแปร a มีสมาชิก 4 ค่า คือ 1,2,3 และ 4 ดังภาพที่ 1.9



ภาพที่ 1.9 การใช้เครื่องหมายพิเศษ [] และ , ในการกำหนดค่าให้กับตัวแปร

ตัวอย่างที่ 1.6 กำหนดให้ตัวแปร a มีสมาชิก 9 ตัว คือ 1,2,3,4,...,9 ดังภาพที่ 1.10



ภาพที่ 1.10 การใช้เครื่องหมายพิเศษ [] และ ... ในการกำหนดค่าให้กับตัวแปร

1.3.3 ตัวดำเนินการเปรียบเทียบตรรกะ

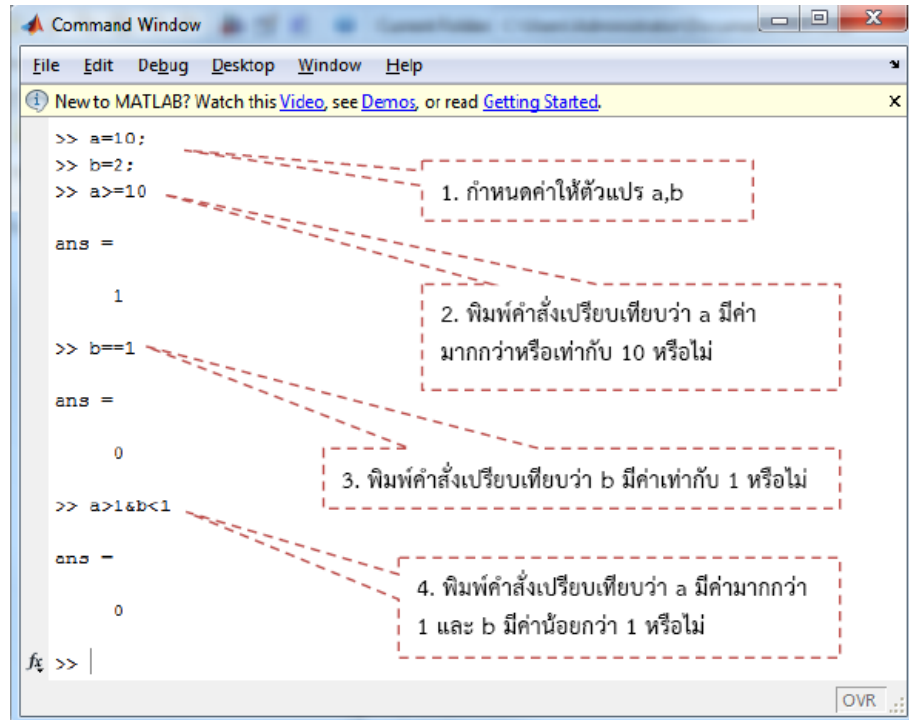
โปรแกรม MATLAB ได้กำหนดตัวดำเนินการเปรียบเทียบตรรกะ เพื่ออำนวยความสะดวกในการเขียนโปรแกรม ดังตารางที่ 1.3

ตารางที่ 1.3 ตัวดำเนินการเปรียบเทียบตรรกะ

ลักษณะดำเนินการ	ตัวดำเนินการ	รูปแบบของ MATLAB
น้อยกว่า	<	$x < 10$
น้อยกว่าหรือเท่ากับ	<=	$x \leq 10$
มากกว่า	>	$x > 10$
มากกว่าหรือเท่ากับ	>=	$x \geq 10$
เท่ากับ	==	$x == 10$
ไม่เท่ากับ	~=	$x \sim 10$
และ	&	$x > 2 \& y < 1$
หรือ		$x > 2 y < 1$
ไม่	~	$\sim x$

ตัวอย่างที่ 1.7 กำหนดให้ตัวแปร $a = 10$ และ $b = 2$ จงว่าเงื่อนไขเหล่านี้เป็นจริงหรือไม่

1. $a \geq 10$
 2. $b = 1$
 3. $a > 1 \& b > 1$
- ดังภาพที่ 1.11



ภาพที่ 1.11 การใช้ตัวดำเนินการเปรียบเทียบตรรกะ

1.3.4 ฟังก์ชันคณิตศาสตร์พื้นฐานและค่าตัวแปรเฉพาะ

โปรแกรม MATLAB ได้กำหนดฟังก์ชันคณิตศาสตร์พื้นฐานและค่าตัวแปรเฉพาะ เพื่ออำนวยความสะดวกในการเขียนโปรแกรม ดังตารางที่ 1.4

ตารางที่ 1.4 ฟังก์ชันคณิตศาสตร์พื้นฐานและค่าตัวแปรเฉพาะ

ค่าตัวแปร	รายละเอียด
pi	ค่าคงที่ pi ซึ่งมีค่าเท่ากับ 22/7
i	ค่าจำนวนเชิงซ้อนที่แทนค่าด้วย $\sqrt{-1}$
j	เหมือนกับ i
Inf	เป็นค่าคงที่ที่เกิดจากการคำนวณจากการหารด้วยค่าศูนย์ ซึ่งโปรแกรม MATLAB ได้กำหนดให้แทนค่าด้วย Infinity (∞)
realmin	Floating Point ที่มีค่าน้อยที่สุด
realmax	Floating Point ที่มีค่ามากที่สุด
NaN	ย่อมาจาก Not-a-Number คือ ค่าคงที่ที่ไม่มีการกำหนด เกิดจากการหารระหว่างค่าศูนย์กับค่าศูนย์
date	เป็นฟังก์ชันที่ใช้ออกค่าวันปัจจุบันในรูปแบบ วัน-เดือน-ปี

ค่าตัวแปร	รายละเอียด
ans	เก็บผลลัพธ์จากการคำนวณค่าปัจจุบันใด ๆ ที่ไม่ได้กำหนดชื่อตัวแปรของผลลัพธ์

ตัวอย่างที่ 1.8 กำหนดให้ตัวแปร $a = \pi$, $b = 1+j1$ และ $c = 0$ จงคำนวณค่าดังต่อไปนี้

1. $a \times b$
 2. $b \div c$
 3. $c \div c$
- ดังภาพที่ 1.12



```

Command Window
File Edit Debug Desktop Window Help
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a=pi;
>> b=1+i;
>> c=0;
>> a*b

ans =

    3.1416 + 3.1416i

>> b/c

ans =

    Inf +    Inf i
>> |
  
```

ภาพที่ 1.12 การใช้ฟังก์ชันคณิตศาสตร์พื้นฐานและค่าตัวแปรเฉพาะ

1.4 สรุป

MATLAB เป็นโปรแกรมที่ช่วยให้การคำนวณทางคณิตศาสตร์ที่ซับซ้อนง่ายขึ้น สามารถเขียนโปรแกรมทำงานในลักษณะของการติดต่อโดยตรง (Interactive) ซึ่งข้อมูลทุกตัวอยู่ในลักษณะของ Array ส่วนประกอบของโปรแกรมแบ่งเป็น 5 ส่วน ได้แก่ 1) ภาษาโปรแกรม MATLAB 2) สถาปัตยกรรมในการทำงานของ MATLAB 3) ฟังก์ชันในการคำนวณทางคณิตศาสตร์ 4) Handle Graphics และ 5) The MATLAB Application Program Interface (API)

โปรแกรม MATLAB มีหน้าต่างส่วนต่าง ๆ ประกอบด้วย Command Window, Current Folder, Workspace และ Details ซึ่งแต่ละส่วนทำหน้าที่ควบคุมการทำงานที่แตกต่างกัน แต่มีการประสานงานการทำงานระหว่างส่วนไปพร้อม ๆ กัน

การกำหนดค่าให้กับตัวแปรของโปรแกรม MATLAB สามารถทำได้โดยใช้เครื่องหมายเท่ากับ (=) แล้วตามด้วยค่าข้อมูลหรือนิพจน์ที่ต้องการให้กับตัวแปร โดยตัวแปรที่เป็นตัวพิมพ์ใหญ่กับตัวพิมพ์เล็กจะเป็นตัวแปรคนละตัวกัน

แบบฝึกหัด

บทที่ 1 การใช้งานโปรแกรม MATLAB เบื้องต้น

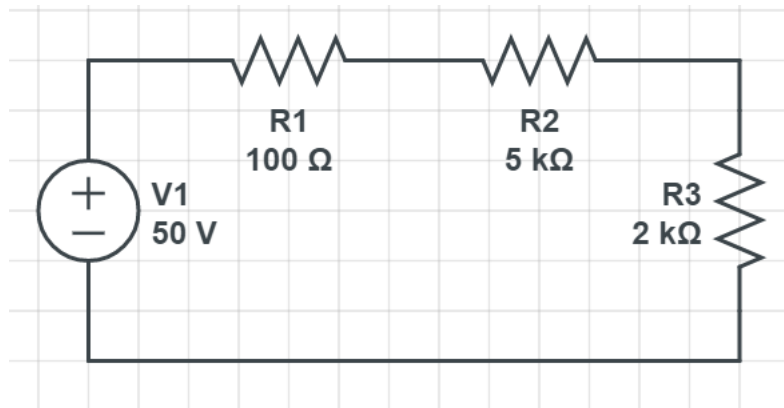
1. จงอธิบายความหมายของโปรแกรม MATLAB
2. ส่วนประกอบของโปรแกรม MATLAB ประกอบด้วยอะไรบ้าง จงอธิบายพอสังเขป
3. จงอธิบายวิธีการเข้าสู่โปรแกรม MATLAB
4. จงเรียกดูตัวแปร X และ x โดยกำหนดให้ $X = 50$ และ $x = 5$
5. จงทำการจบคำสั่งในบรรทัดต่อไป เมื่อกำหนดให้ตัวแปร x มีสมาชิก 5 ตัว คือ 1,2,...,5 โดยใช้โปรแกรม MATLAB
6. จงสร้างเวกเตอร์และเมทริกซ์ เมื่อกำหนดค่าให้ตัวแปร c มีสมาชิก 7 ค่า ดังนี้ 1,2,3,4,5,6 และ 7
7. กำหนดให้ตัวแปร $a = 2$ และ $b = 3$ จงใช้โปรแกรม MATLAB คำนวณหาค่าดังนี้
 - 7.1 $a + b$
 - 7.2 $b \times a$
 - 7.3 b^a
8. กำหนดให้ตัวแปร $X = 10$ และ $Y = 20$ จงใช้โปรแกรม MATLAB ว่าเงื่อนไขเหล่านี้เป็นจริงหรือไม่
 - 8.1 $X \leq Y$
 - 8.2 $Y > 10 \mid < 20$
 - 8.3 $X \neq Y$
9. กำหนดให้ตัวแปร $a = 1$, $b = \pi$ และ $c = 0$ จงคำนวณค่าดังนี้
 - 9.1 $b + b$
 - 9.2 $c \div c$
 - 9.3 $b \times c$
10. กำหนดให้ตัวแปร $x = [1.414, 2.687]$ จงหาจำนวนเต็มของ x ที่มีค่าใกล้เคียงอนันต์ โดยใช้โปรแกรม MATLAB
11. ถ้าลวดนิโครมมีความต้านทาน 75 โอห์ม ต่ออยู่กับเซลล์ไฟฟ้าขนาด 1.5 โวลต์ จะมีกระแสไฟฟ้าไหลผ่านลวดนิโครมเท่าใด จงใช้โปรแกรม MATLAB คำนวณหาโจทย์นี้

จากสูตร

$$I = \frac{V}{R}$$

12. ลวดทองแดงต่ออยู่กับเซลล์ไฟฟ้าขนาด 5 โวลต์ จะมีกระแสไฟฟ้าไหลผ่านลวดทองแดง 0.10 แอมแปร์ อยากทราบว่า ลวดทองแดงเส้นนี้มีความต้านทานเท่าใด จงใช้โปรแกรม MATLAB คำนวณหาโจทย์นี้

13. จงใช้โปรแกรม MATLAB คำนวณหาค่ากระแสไฟฟ้าที่ไหลในวงจร ดังรูป

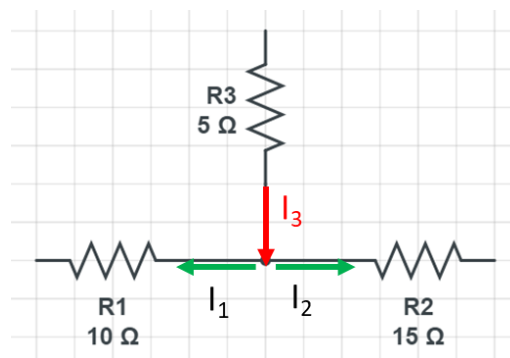


14. จากรูปข้อที่ 13 จงใช้โปรแกรม MATLAB คำนวณหาความต้านทานรวมทั้งหมดในวงจร

15. จากกฎกระแสไฟฟ้าของเคอร์ชอฟฟ์ กำหนดให้ $I_1 = 3$ แอมแปร์ $I_2 = 2$ แอมแปร์ ดังรูป อยากทราบว่ากระแสไฟฟ้าของ I_3 มีค่าเท่าใด จงใช้โปรแกรม MATLAB คำนวณหาโจทย์นี้

สูตรกฎกระแสไฟฟ้าของเคอร์ชอฟฟ์

ผลรวมของกระแสที่ไหลเข้า = ผลรวมของกระแสที่ไหลออก



รูปแบบคำสั่งและฟังก์ชันเบื้องต้นของโปรแกรม MATLAB

การคำสั่งควบคุมพื้นฐานของโปรแกรม MATLAB มีลักษณะคล้ายกับการเขียน Algorithm โดยทั่วไป สามารถแสดงค่าข้อมูลลักษณะเชิงตัวเลข โดยการกำหนดรูปแบบการแสดงผลจำนวนบน Command Window ซึ่งรูปแบบฟังก์ชันการทำงานของโปรแกรม MATLAB มีคำสั่งการเขียนโปรแกรมเหมือนกับสมการทางคณิตศาสตร์ อาทิเช่น ฟังก์ชันทางคณิตศาสตร์ เช่น การหารากที่สอง (Square Root) การหาค่าสมบูรณ การหาค่าโดยใช้ความสัมพันธ์ของวงกลมหนึ่งหน่วย เป็นต้น

สำหรับบทนี้จะนำเสนอรูปแบบคำสั่งการแสดงผลค่าข้อมูล ฟังก์ชันการคำนวณทางพีชคณิต ฟังก์ชันการหาค่าทางตรีโกณมิติ และ Logarithm ที่เป็นพื้นฐานการในคำนวณทางคณิตศาสตร์สามารถนำไปประยุกต์ใช้การคำนวณทางคณิตศาสตร์ต่อไป

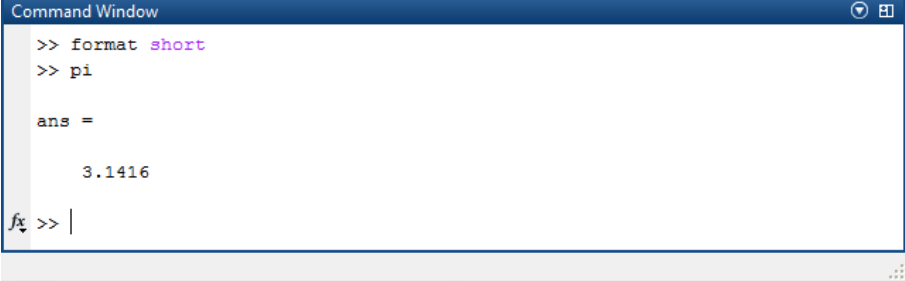
2.1 การแสดงค่าข้อมูลในเชิงตัวเลขของโปรแกรม MATLAB

การแสดงผลค่าข้อมูลในเชิงตัวเลขของโปรแกรม MATLAB สามารถควบคุมการแสดงผลค่าได้หลายรูปแบบ มีการกำหนดรูปแบบการแสดงผลจำนวนบนหน้าต่างคำสั่ง หรือการเลือกเมนูที่อยู่ในหน้าต่างคำสั่ง ในบทเรียนนี้ขอยกตัวอย่างการแสดงผลค่า pi สามารถอธิบายได้ดังตารางที่ 2.1

ตารางที่ 2.1 การแสดงผลค่าข้อมูลในเชิงตัวเลข

คำสั่งโปรแกรม MATLAB	ค่า pi	รูปแบบการแสดงผล
format short	3.1416	5 digit
format short e	3.1416e+00	5 digit plus exponent
format short g	3.1416	Best format short or format short e
format long	3.14165926535897	16 digit
format long e	3.14165926535897+00	16 digit plus exponent
format long g	3.14165926535897	Best format short or format long e
format hex	400921fb54442d18	Hexadecimal, floating point
format bank	3.14	2 decimal digit
format +	+	บวก ลบ หรือ ศูนย์
format rat	355/113	อัตราส่วนโดยประมาณ

ตัวอย่างที่ 2.1 การใช้คำสั่ง format short แสดงค่าข้อมูลในเชิงตัวเลข ดังภาพที่ 2.1

A screenshot of the MATLAB Command Window. The title bar says "Command Window". The command prompt shows the user entering the command 'format short' followed by 'pi'. The output shows 'ans =' followed by the value '3.1416' on the next line. The cursor is at the end of the command line.

```
>> format short
>> pi

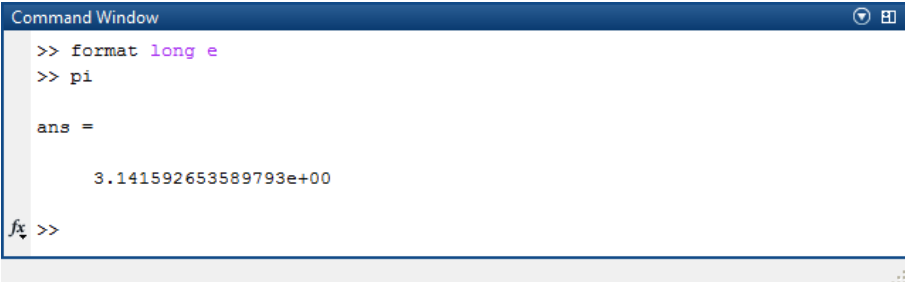
ans =

    3.1416

fx >> |
```

ภาพที่ 2.1 การใช้คำสั่งแสดงผลตัวเลขแบบ format short

ตัวอย่างที่ 2.2 การใช้คำสั่ง format long e แสดงค่าข้อมูลในเชิงตัวเลข ดังภาพที่ 2.2

A screenshot of the MATLAB Command Window. The title bar says "Command Window". The command prompt shows the user entering the command 'format long e' followed by 'pi'. The output shows 'ans =' followed by the value '3.141592653589793e+00' on the next line. The cursor is at the end of the command line.

```
>> format long e
>> pi

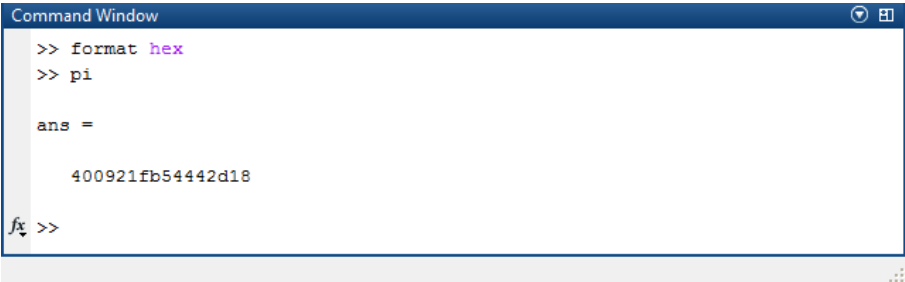
ans =

    3.141592653589793e+00

fx >>
```

ภาพที่ 2.2 การใช้คำสั่งแสดงผลตัวเลขแบบ format long e

ตัวอย่างที่ 2.3 การใช้คำสั่ง format hex แสดงค่าข้อมูลในเชิงตัวเลข ดังภาพที่ 2.3

A screenshot of the MATLAB Command Window. The title bar says "Command Window". The command prompt shows the user entering the command 'format hex' followed by 'pi'. The output shows 'ans =' followed by the hexadecimal value '400921fb54442d18' on the next line. The cursor is at the end of the command line.

```
>> format hex
>> pi

ans =

    400921fb54442d18

fx >>
```

ภาพที่ 2.3 การใช้คำสั่งแสดงผลตัวเลขแบบ format hex

2.2 ฟังก์ชันการคำนวณทางพีชคณิต

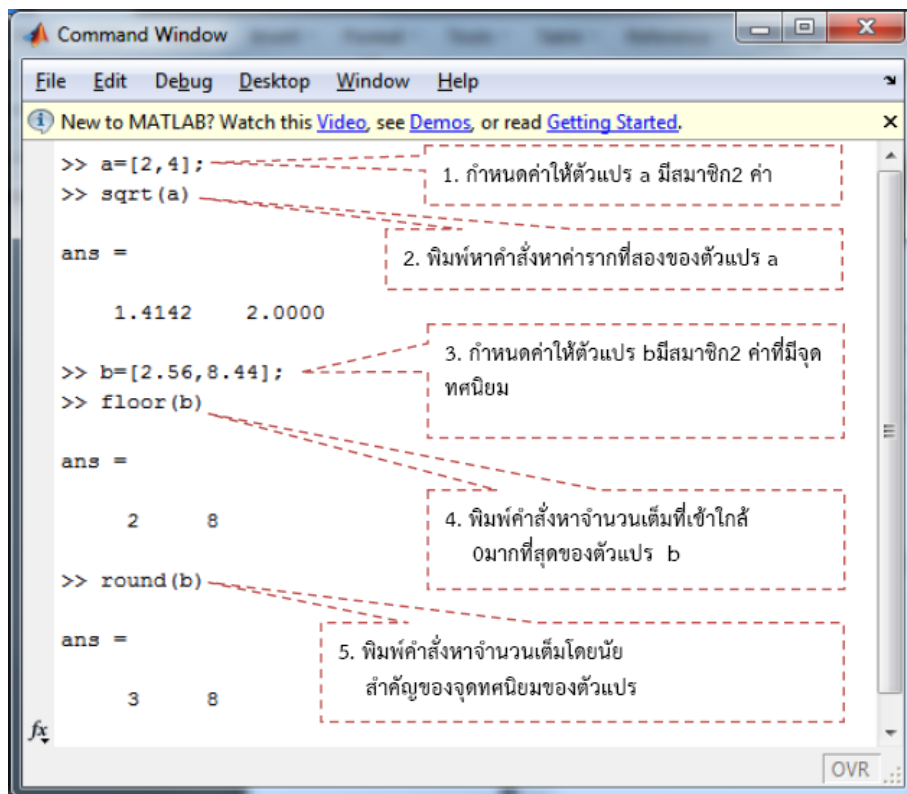
การเขียนโปรแกรม MATLAB สามารถทำการคำนวณพีชคณิตในด้านต่าง ๆ ได้ เช่น การหารากที่สอง (Square Root) การหาค่าสมบูรณ์ การปัดเศษ เป็นต้น จึงทำให้โปรแกรมที่เขียนขึ้นมีความยาวมากและเสียเวลาในการเขียนโปรแกรม ดังนั้นโปรแกรม MATLAB จึงมีฟังก์ชันที่ใช้สำหรับการคำนวณทางพีชคณิต ดังตารางที่ 2.2

ตารางที่ 2.2 ฟังก์ชันในการคำนวณทางพีชคณิต

คำสั่ง	รายละเอียด
sqrt(x)	การหารากที่สองของค่า x หรือ $\sqrt{x}$
fix(x)	การหาค่าจำนวนเต็มใด ๆ ที่ใกล้เคียงกับค่า x โดยกำหนดให้ค่านั้นเข้าใกล้ 0 มากที่สุด
floor(x)	การหาค่าจำนวนเต็มใด ๆ ที่ใกล้เคียงกับค่า x โดยกำหนดให้ค่านั้นเข้าใกล้ $-\infty$ มากที่สุด
round(x)	การหาค่าจำนวนเต็มใด ๆ ที่ใกล้เคียงกับค่า x โดยนัยสำคัญของจุดทศนิยม
ceil(x)	การหาค่าจำนวนเต็มใด ๆ ที่ใกล้เคียงกับค่า x โดยกำหนดให้ค่านั้นเข้าใกล้ $+\infty$ มากที่สุด
sign(x)	การกำหนดค่า x ใด ๆ ให้มีค่าเป็น -1, 0 หรือ 1 โดยพิจารณาถ้าค่า x มีน้อยกว่า 0 ให้เป็น -1 หรือถ้าค่า x มีค่าเท่ากับ 0 ให้เป็น 0 หรือถ้าค่า x มีค่าน้อยกว่า 0 ให้เป็น 1
rem(x,y)	การหาค่าเศษที่เหลือจากการหารกันระหว่างค่า x และ y
abs(x)	การคำนวณค่าหาสัมบูรณ์ (Absolute)

ตัวอย่างที่ 2.4 กำหนดให้ตัวแปร $a = [2, 4]$ และ $b = [2.56, 8.44]$ จงหา

1. ค่า $\sqrt{a}$
2. จำนวนเต็ม b ที่มีค่าใกล้เคียง 0
3. จำนวนเต็ม b ที่มีค่าใกล้เคียงอนันต์ ดังภาพที่ 2.4



ภาพที่ 2.4 การใช้ sqrt, floor และ round ในการคำนวณทางพีชคณิต

ตัวอย่างที่ 2.5 กำหนดให้ตัวแปร $x = [1, 2]$ และ $y = [1.53, 4.23]$ จงหา

1. จำนวนเต็ม y ที่ใกล้เคียง 0
2. ค่าสัมบูรณ์ของ x
3. จำนวนเต็ม y ที่มีค่าใกล้เคียงอนันต์ ดังภาพที่ 2.5

```

>> y=[1.53,4.23];
>> floor(y)

ans =

     1     4

>> x=[1,2];
>> abs(x)

ans =

     1     2

>> ceil(y)

ans =

     2     5
  
```

Annotations in the image:

- 1. กำหนดค่าให้ตัวแปร y มีสมาชิก 2 ค่าที่มีจุดทศนิยม
- 2. พิมพ์คำสั่งหาจำนวนเต็มทีเข้าใกล้ 0 มากที่สุดของตัวแปร y
- 3. กำหนดค่าให้ตัวแปร x มีสมาชิก 2 ค่า
- 4. พิมพ์คำสั่งหาค่าสัมบูรณ์
- 5. พิมพ์คำสั่งหาจำนวนเต็มใด ๆ ที่ใกล้เคียงกับค่า x โดยกำหนดให้ค่าเข้าใกล้ $+\infty$ มากที่สุด

ภาพที่ 2.5 การใช้ floor, abs และ ceil ในการคำนวณทางพีชคณิต

2.3 ฟังก์ชันการหาค่าทางตรีโกณมิติ และ Logarithm

2.3.1 การหาค่าทางตรีโกณมิติ

การหาค่าทางตรีโกณมิติ เป็นการหาค่าโดยใช้ความสัมพันธ์ของวงกลมหนึ่งหน่วยที่มีการวัดมุมเป็นหน่วยองศาและเรเดียน แต่ในการคำนวณของโปรแกรม MATLAB จะใช้คำนวณในหน่วยของเรเดียน ดังนั้น ก่อนการคำนวณควรพิจารณาหน่วยให้เป็นเรเดียน ดังตารางที่ 2.3

เมื่อ 180 องศา เท่ากับ π หรือ pi เรเดียน
 ดังนั้น x องศา เท่ากับ $x * \pi / 180$ เรเดียน

ตารางที่ 2.3 ฟังก์ชันในการคำนวณค่าทางตรีโกณมิติ

คำสั่ง	รายละเอียด
sin(x)	การคำนวณหาค่า sine ของค่า x
cos(x)	การคำนวณหาค่า cosine ของค่า x
tan(x)	การคำนวณหาค่า tangent ของค่า x

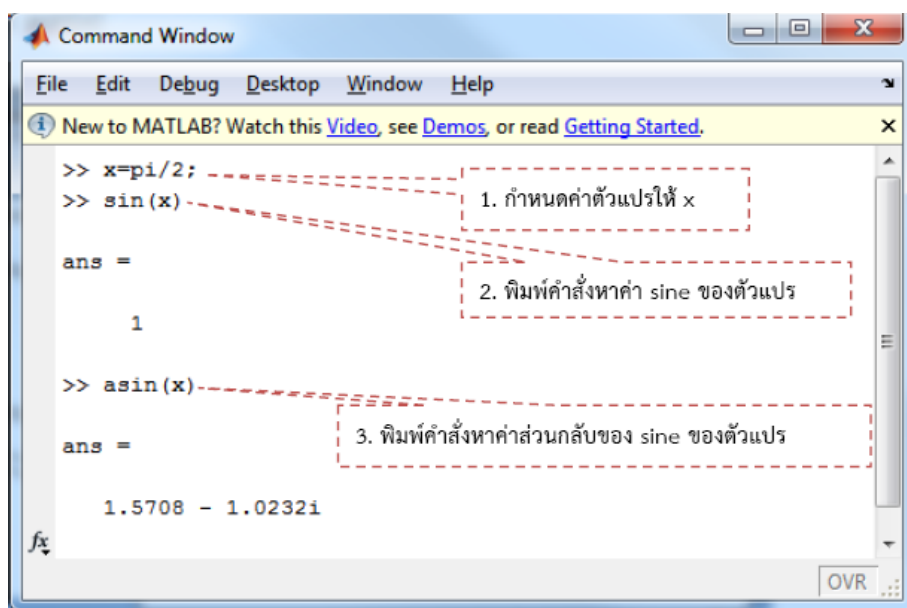
ตารางที่ 2.3 ฟังก์ชันในการคำนวณค่าทางตรีโกณมิติ (ต่อ)

คำสั่ง	รายละเอียด
asin(x)	การคำนวณค่าอินเวอร์สของ sine ซึ่งผลที่ได้จากการคำนวณอยู่ระหว่าง $\pi/2$ ถึง $\pi/2$ โดยที่ค่า x อยู่ในช่วง -1 ถึง 1
acos(x)	การคำนวณค่าอินเวอร์สของ cosine ซึ่งผลที่ได้จากการคำนวณอยู่ระหว่าง 0 ถึง π โดยที่ค่า x อยู่ในช่วง -1 ถึง 1
atan(x)	การคำนวณค่าอินเวอร์สของ tangent ซึ่งผลที่ได้จากการคำนวณอยู่ระหว่าง $\pi/2$ ถึง $\pi/2$ โดยที่ค่า x อยู่ในช่วง -1 ถึง 1

ตัวอย่างที่ 2.6 กำหนดให้ $x = 90$ องศา จงคำนวณหาค่า $\sin(x)$ และ $\text{asin}(x)$ ดังภาพที่ 2.6

วิธีทำ

$x = 90$ องศา ให้ทำการแปลงจากองศาเป็นเรเดียน ดังนี้
 90 องศา เท่ากับ $90 \cdot \pi / 180$ เรเดียน
 เท่ากับ $\pi/2$ เรเดียน



ภาพที่ 2.6 การใช้คำสั่ง $\sin(x)$ และ $\text{asin}(x)$ ในการคำนวณค่าทางตรีโกณมิติ

ตัวอย่างที่ 2.7 กำหนดให้ $x = 60$ องศา จงคำนวณหาค่า $\cos(x)$ และ $\tan(x)$ ดังภาพที่ 2.7
วิธีทำ

$x = 60$ องศา ให้ทำการแปลงจากองศาเป็นเรเดียน ดังนี้
 60 องศา เท่ากับ $60 \cdot \pi / 180$ เรเดียน
 เท่ากับ $\pi / 3$ เรเดียน

```

Command Window
>> x=pi/20;
>> cos(x)
ans =
    0.9877
>> atan(x)
ans =
    0.1558
fx >> |
    
```

1. กำหนดค่าให้ตัวแปร x

2. พิมพ์คำสั่งหาค่า cosine ของตัวแปร

3. พิมพ์คำสั่งหาค่าส่วนกลับของ tangent ของตัวแปร

ภาพที่ 2.7 การใช้คำสั่ง $\cos(x)$ และ $\tan(x)$ ในการคำนวณค่าทางตรีโกณมิติ

2.3.2 การหาค่า Logarithm

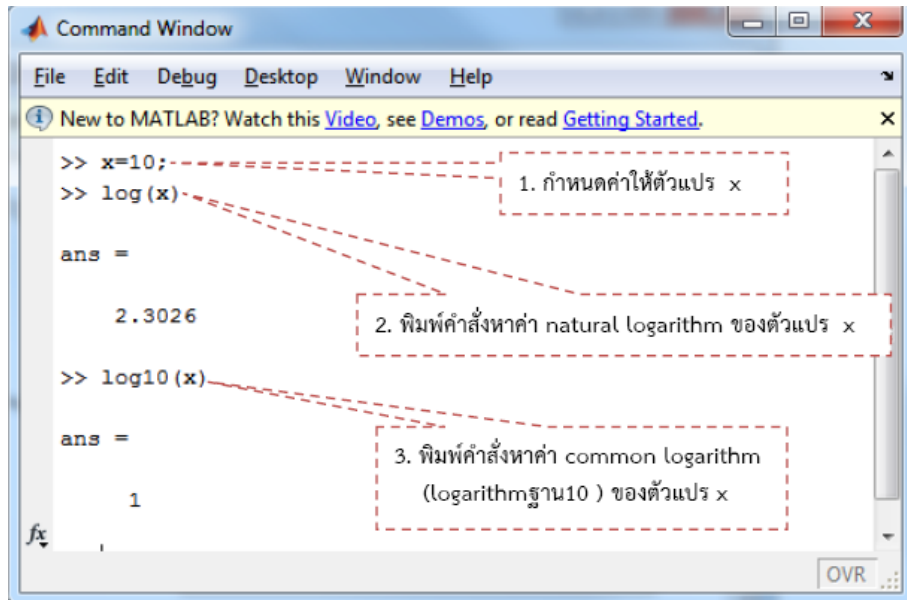
การคำนวณหาค่า Logarithm มีความยุ่งยากในการคำนวณมาก เพราะต้องมีการเปิดตารางเพื่อเทียบค่าในการผลลัพธ์ ดังนั้น เพื่อเป็นการลดความยุ่งยากดังกล่าว โปรแกรม MATLAB จึงมีฟังก์ชันที่ใช้สำหรับการหาค่า Logarithm ดังตารางที่ 2.2

ตารางที่ 2.4 ฟังก์ชันที่ใช้สำหรับการหาค่า Logarithm

คำสั่ง	รายละเอียด
$\log(x)$	การหาค่า Natural Logarithm (ln) ของค่า x
$\log_{10}(x)$	การหาค่า Common Logarithm (Logarithm ฐาน 10) ของค่า x

ตัวอย่างที่ 2.8 กำหนดให้ $a = 10$ จงหา

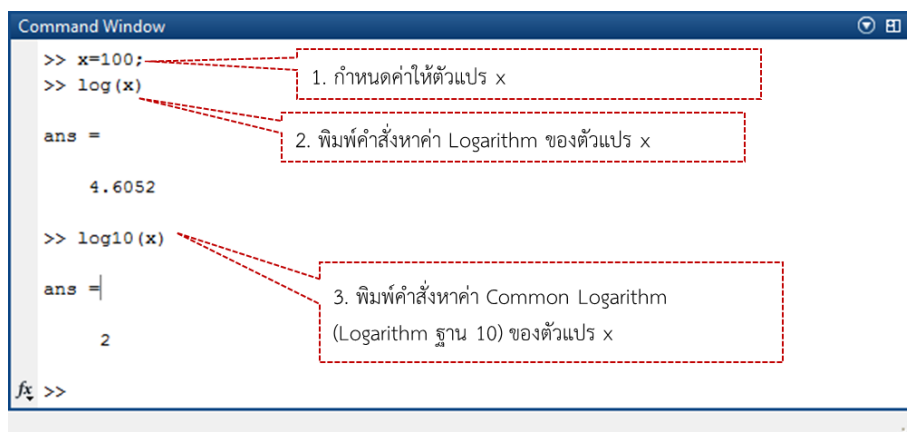
1. Natural Logarithm ของ a
2. Logarithm ฐาน 10 ของ a ดังภาพที่ 2.8



ภาพที่ 2.8 การใช้คำสั่ง $\log(x)$ และ $\log_{10}(x)$ ในการหาค่า Logarithm

ตัวอย่างที่ 2.9 กำหนดให้ $x = 100$ จงหา

1. ค่า Logarithm ของ x
2. ค่า Common Logarithm ของ x ดังภาพที่ 2.9



ภาพที่ 2.9 การใช้คำสั่ง $\log(x)$ และ $\log_{10}(x)$ ในการหาค่า Logarithm

2.4 สรุป

โปรแกรม MATLAB มีรูปแบบคำสั่งลักษณะเหมือนการเขียน Algorithm แสดงค่าข้อมูลในเชิงตัวเลขและสามารถควบคุมการแสดงผลค่าได้หลายรูปแบบ มีการกำหนดรูปแบบการแสดงผลจำนวนบนหน้าต่างคำสั่ง สำหรับฟังก์ชันการเขียนโปรแกรม MATLAB สามารถทำการคำนวณพีชคณิตในด้านต่าง ๆ ได้ เช่น การหารากที่สอง (Square Root) การหาค่าสมบูรณ์ การปัดเศษ เป็นต้น จึงทำให้โปรแกรมที่เขียนขึ้นมีความยาวมากและเสียเวลาในการเขียนโปรแกรม

การใช้รูปแบบคำสั่งสำหรับเขียนโปรแกรม MATLAB มีพื้นฐานการใช้งาน ดังนี้

1. การแสดงค่าข้อมูลในเชิงตัวเลขของโปรแกรม MATLAB
มีรูปแบบคำสั่งที่นิยมใช้ ได้แก่ format
2. ฟังก์ชันการคำนวณทางพีชคณิต
มีรูปแบบคำสั่งที่นิยมใช้ ได้แก่ sqrt(x) sign(x) abs(x)
3. ฟังก์ชันการหาค่าทางตรีโกณมิติ และ Logarithm
มีรูปแบบคำสั่งที่นิยมใช้ ได้แก่ sin(x) cos(x) tan(x) log(x) log10(x)

แบบฝึกหัด

บทที่ 2 รูปแบบคำสั่งและฟังก์ชันเบื้องต้นของโปรแกรม MATLAB

1. จงแสดงการใช้คำสั่งแสดงค่าข้อมูลในเชิงตัวเลข ดังต่อไปนี้
 - 1.1 format short, format short e, format short g
 - 1.2 format long, format long e, format long g
 - 1.3 format hex
 - 1.4 format bank
 - 1.5 format +
 - 1.6 format rat
2. จงใช้คำสั่งสำหรับฟังก์ชัน Sine หาค่าของ $\sin \pi/2$ หน่วยมมเรเดียน โดยใช้โปรแกรม MATLAB
3. จงใช้คำสั่งสำหรับฟังก์ชัน Consine หาค่าของ $\cos 47^\circ$ หน่วยมมองศา โดยใช้โปรแกรม MATLAB
4. จงใช้คำสั่งสำหรับฟังก์ชัน Secant หาค่าของ $\sec \pi$ หน่วยมมเรเดียน โดยใช้โปรแกรม MATLAB
5. จงใช้คำสั่งสำหรับฟังก์ชัน Cosecant หาค่าของ $\operatorname{cosec} \pi/4$ หน่วยมมเรเดียน โดยใช้โปรแกรม MATLAB
6. จงหาค่า $\sqrt{e}$ โดยใช้โปรแกรม MATLAB
7. จงใช้คำสั่งสำหรับฟังก์ชัน Logarithm ฐาน 10 หาค่าของ $\log 100$ โดยใช้โปรแกรม MATLAB
8. จงใช้คำสั่งสำหรับฟังก์ชัน Logarithm ฐาน e หาค่าของ $\ln 100$ โดยใช้โปรแกรม MATLAB
9. กำหนดให้ตัวแปร $x = [10,20]$ และ $y = [4.2532,6.2734]$ จงหาค่าดังต่อไปนี้ โดยใช้โปรแกรม MATLAB
 - 9.1 จำนวนเต็ม y ที่ใกล้เคียง 0
 - 9.2 ค่าสัมบูรณ์ของ x
 - 9.3 จำนวนเต็ม y ที่มีค่าใกล้เคียงอนันต์
 - 9.4 ค่า $\sqrt{x}$

10. กำหนดให้ $x = 180$ องศา จงคำนวณหาค่าดังต่อไปนี้ โดยใช้โปรแกรม MATLAB
- 10.1 ค่า $\sin(x)$ และ $\text{asin}(x)$
 - 10.2 ค่า $\cos(x)$ และ $\text{acos}(x)$
 - 10.3 ค่า $\tan(x)$ และ $\text{atan}(x)$
11. กำหนดให้ $y = 4.71$ เรเดียน จงคำนวณหาค่าดังต่อไปนี้ โดยใช้โปรแกรม MATLAB
- 11.1 ค่า $\sin(y)$ และ $\text{asin}(y)$
 - 11.2 ค่า $\cos(y)$ และ $\text{acos}(y)$
 - 11.3 ค่า $\tan(y)$ และ $\text{atan}(y)$
12. กำหนดให้ตัวแปร $z = 20$ จงคำนวณหาค่าดังต่อไปนี้ โดยใช้โปรแกรม MATLAB
- 12.1 ค่า Logarithm ของ x
 - 12.2 ค่า Common Logarithm ของ x
13. จงใช้โปรแกรม MATLAB คำนวณหาผลลัพธ์ของ $y = \ln \frac{1}{x} - \log(x) + \sin 40^\circ$ โดยกำหนดให้ $x = 1.5$
14. จงใช้โปรแกรม MATLAB คำนวณหาผลลัพธ์ของ $y = \frac{1}{\sqrt{x} + \frac{1}{\sqrt{x}}}$ โดยกำหนดให้ $x = 6$
15. จงใช้โปรแกรม MATLAB คำนวณหาผลลัพธ์ของ $x = y^2 + 2y + 10$ โดยกำหนดให้ $y = 3$

บทที่ 3

การคำนวณทางคณิตศาสตร์เชิงตัวแปร โดยใช้โปรแกรม MATLAB

การคำนวณทางคณิตศาสตร์เชิงตัวแปรของโปรแกรม MATLAB ต้องมีการประกาศตัวแปรทุกครั้ง เพื่อให้สามารถดำเนินการคำนวณทางคณิตศาสตร์ได้เหมือนกับการบวก การลบ การคูณ และการหาร ซึ่งการประกาศค่าตัวแปรของโปรแกรม MATLAB สามารถดำเนินการได้ 2 แบบ คือ ตัวแปรวัตถุคลาส `sym` หรือ `syms` ทั้ง 2 แบบนี้แตกต่างกันตรงที่วิธีการเขียนใช้งานเท่านั้น และยังสามารถสร้างอาร์เรย์ของตัวแปรใช้งานแทนปริมาณเมทริกซ์ที่ติดตัวแปร เป็นต้น

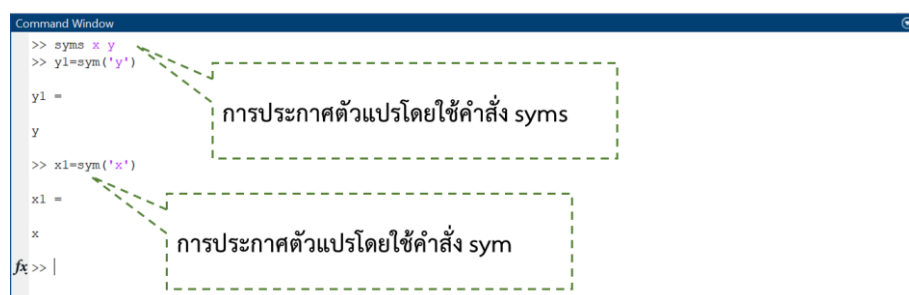
เมื่อทำการประกาศตัวแปรวัตถุคลาสเรียบร้อยแล้ว สามารถทำการระบุรูปแบบการกระทำทางดำเนินการทางพีชคณิต หรือจัดรูปแบบนิพจน์ และทำการแก้สมการเพื่อหาค่าตัวแปรต่าง ๆ ได้แก่ การแก้สมการพีชคณิตเชิงเส้น การหาลิมิตของฟังก์ชัน การหาอนุพันธ์ของฟังก์ชัน และการหาอนุพันธ์ย่อย เป็นต้น

3.1 ตัวแปรวัตถุคลาส Symbolic

ตัวแปรวัตถุคลาส Symbolic หรือเรียกย่อว่า `sym` หมายถึง คลาสที่สามารถนำตัวแปรต่าง ๆ มาดำเนินการทางคณิตศาสตร์ได้เหมือนกับการบวก การลบ การคูณ และการหาร อีกทั้งยังสามารถสร้างอาร์เรย์ของตัวแปร เพื่อใช้งานแทนปริมาณเมทริกซ์ที่ติดตัวแปร เป็นต้น

3.1.1 การสร้างตัวแปรวัตถุคลาส `sym`

การประกาศตัวแปรโดยใช้คำสั่ง `sym` หรือ `syms` นั้นมีผลการใช้งานเหมือนกันต่างจากการใช้งาน `sym` ต้องใส่วงเล็บและสร้างตัวแปรได้ที่ละตัว ส่วน `syms` ไม่ต้องใส่วงเล็บและสามารถสร้างวัตถุได้พร้อมกันหลายตัว ดังภาพที่ 3.1



ภาพที่ 3.1 การประกาศตัวแปรโดยใช้คำสั่ง `sym` และ `syms`

โดยปกติรูปแบบที่กำหนดมาในโปรแกรม MATLAB มีรูปแบบ Complex การระบุรูปแบบของตัวแปรสามารถทำได้โดยใช้คำสั่ง ดังนี้

`s=sym('x','format')` หรือ `syms x format`

โดยที่ format คือ รูปแบบที่ระบุได้ดังนี้

Complex	คือ	จำนวนเชิงซ้อน
Real	คือ	จำนวนจริง
Positive	คือ	จำนวนบวก
Clear	คือ	สามารถใช้การลบรูปแบบกลับมาเป็นค่าปกติ คือ complex

ดังภาพที่ 3.2

```

Command Window
>> syms x y real
>> s=sqrt(x)

s =
x^(1/2)

>> m=sqrt(x)

m =
x^(1/2)

>> syms x y positive
>> s=sqrt(-x)

s =
x^(1/2)*1i

>> m=sqrt(x)

m =
x^(1/2)
  
```

การระบุรูปแบบคำสั่ง real

การระบุรูปแบบคำสั่ง positive

ภาพที่ 3.2 การระบุรูปแบบของตัวแปร

3.1.2 การดำเนินการทางพีชคณิตกับคลาส sym และ syms

ตัวอย่างการดำเนินการทางพีชคณิตกับคลาส sym ดังภาพที่ 3.3

```

Command Window
>> x=sym('x','real')

x =
x

>> x*x*x+x+x

ans =
x^3 + 2*x

fx >>
  
```

ภาพที่ 3.3 การดำเนินการทางพีชคณิตโดยใช้คลาส sym

ตัวอย่างการดำเนินการทางพีชคณิตกับคลาส syms ดังภาพที่ 3.4

```
Command Window
>> syms x y
>> y+y

ans =

2*y
>> x+x

ans =

2*x
>> x*x*x+x+x-y

ans =

x^3 + 2*x - y
fx>>
```

ภาพที่ 3.4 การดำเนินการทางพีชคณิตโดยใช้คลาส syms

3.1.3 การจัดรูปนิพจน์

การจัดรูปแบบคำตอบให้สวยงามสามารถทำได้โดยใช้ฟังก์ชัน pretty ดังภาพที่ 3.5

```
Command Window
>> syms x
>> pretty((x^3+x^2+2*x+1)/(x^2+2*x))
      3      2
x  + x  + 3 x + 1
-----
      2
x  + 2 x
fx>> |
```

ภาพที่ 3.5 การจัดรูปแบบคำตอบโดยใช้ฟังก์ชัน pretty

สำหรับฟังก์ชันที่ช่วยลดรูปแบบของสมการ ดังตารางที่ 3.1

ตารางที่ 3.1 ฟังก์ชันที่ช่วยลดรูปแบบของสมการ

ฟังก์ชัน	รายละเอียด
collect	ใช้ในการรวมพจน์
expand	ใช้ในการกระจายพจน์
horner	ใช้ในการดึงตัวประกอบร่วม
factor	ใช้ในการแยกตัวประกอบ
simplify	ใช้ในการทำให้อยู่ในรูปแบบอย่างง่าย
simple	ใช้ในการค้นหารูปแบบอย่างง่ายที่สั้นที่สุดจากทุกรูปแบบที่รองรับ

ตัวอย่างที่ 3.1 การใช้ฟังก์ชันที่ช่วยลดรูปแบบของสมการ ดังภาพที่ 3.6

```
Command Window
>> syms x y
>> collect((x+2)*(x+2))

ans =

x^2 + 4*x + 4

>> expand(sin(x+y))

ans =

cos(x)*sin(y) + cos(y)*sin(x)

>> horner(x^3+x^2+x+2)

ans =

x*(x*(x + 1) + 1) + 2

>> factor(x^2+4*x+4)

ans =

[x + 2, x + 2]
```

ภาพที่ 3.6 ฟังก์ชันที่ช่วยลดรูปแบบของสมการ

ตัวอย่างที่ 3.2 จงใช้ฟังก์ชัน collect เขียนนิพจน์ต่อไปนี้ให้อยู่ในรูปอย่างง่าย ดังภาพที่ 3.7

1. $3x + 4x$

2. $2a - a$

3. $10r + 6r - r$

```
Command Window
>> syms x a r
>> collect((3*x)+(4*x))

ans =

7*x

>> collect((2*a)-a)

ans =

a

>> collect((10*r)+(6*r)-r)

ans =

15*r

fx>>
```

1. นิพจน์ $3x + 4x$

2. นิพจน์ $2a - a$

3. นิพจน์ $10r + 6r - r$

ภาพที่ 3.7 ฟังก์ชัน collect เขียนนิพจน์ในรูปอย่างง่าย

3.2 การแก้สมการในการคำนวณทางคณิตศาสตร์เชิงตัวแปร

การแก้สมการเพื่อหาค่าตัวแปร สามารถทำได้โดยใช้ฟังก์ชัน solve ดังภาพที่ 3.8

```
Command Window
>> syms a b c x
>> eqn = a*x^2 + b*x + c == 0

eqn =

a*x^2 + b*x + c == 0

>> S = solve(eqn)
S =

-(b + (b^2 - 4*a*c)^(1/2))/(2*a)
-(b - (b^2 - 4*a*c)^(1/2))/(2*a)

>> Sa = solve(eqn,a)
Sa =

-(c + b*x)/x^2

fx >>
```

ภาพที่ 3.8 การแก้สมการเพื่อหาค่าตัวแปรโดยใช้ฟังก์ชัน solve

ตัวอย่างที่ 3.3 จงแก้สมการหาค่าจำนวนจริงของ x เมื่อกำหนดให้ $x^5 = 3,125$ ดังภาพที่ 3.9

```
Command Window
>> syms x
>> eqn=x^5==3125;
>> S=solve(eqn,x,'real',true)

S =

5

fx >> |
```

ภาพที่ 3.9 การแก้สมการเพื่อหาค่าตัวแปรโดยใช้ฟังก์ชัน solve และ real

3.2.1 ฟังก์ชันเชิงเส้น

เป็นการสร้างตัวแปรที่เป็นเมทริกซ์เป็นหลัก ฟังก์ชันที่รองรับการคำนวณคลาส sym หรือ syms ในรูปแบบเมทริกซ์ ดังตารางที่ 3.2

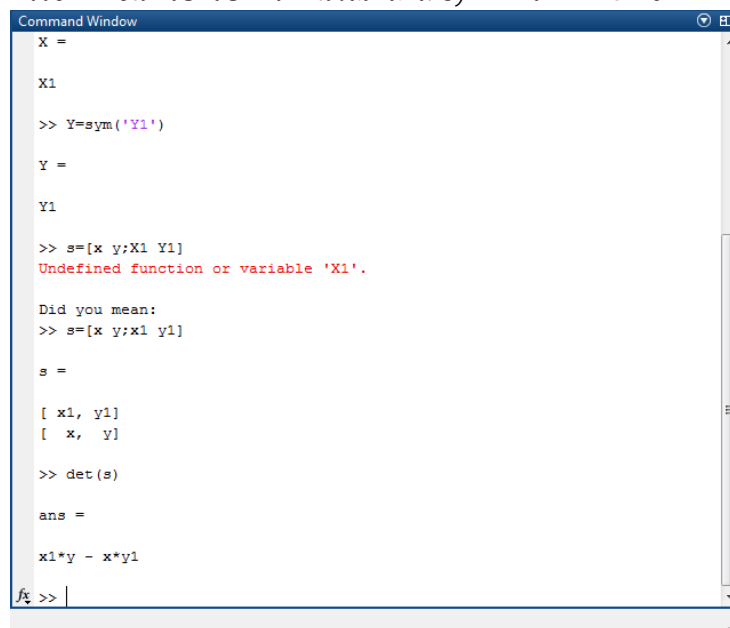
ตารางที่ 3.2 ฟังก์ชันที่รองรับการคำนวณคลาส sym หรือ syms

ฟังก์ชัน	รายละเอียด
colpase	เป็นฟังก์ชันที่คืนค่ามูลฐานสำหรับปริภูมิในแนวหลักของเมทริกซ์ป้อนเข้า
det	เป็นฟังก์ชันสำหรับหาตัวกำหนด หรือดีเทอร์มิแนนท์
diag	เป็นฟังก์ชันสำหรับคืนค่าสมาชิกในเส้นทแยงมุมหลักทั้งหมดมาเป็นเวกเตอร์
eig	เป็นฟังก์ชันสำหรับคำนวณหาค่าเจาะจง (Eigenvalue) และเวกเตอร์เจาะจง (Eigenvector)
expm	เป็นฟังก์ชันสำหรับคำนวณค่า e^A เมื่อ A เป็นเมทริกซ์จัตุรัส
inv	เป็นฟังก์ชันสำหรับการคำนวณเมทริกซ์ผกผัน

ตารางที่ 3.2 ฟังก์ชันที่รองรับการคำนวณคลาส sym หรือ syms (ต่อ)

ฟังก์ชัน	รายละเอียด
nll	เป็นฟังก์ชันสำหรับสร้างเวกเตอร์มูลฐานสำหรับปริภูมิศูนย์ของเมทริกซ์ป้อนเข้า
ply	เป็นฟังก์ชันการคำนวณสัมประสิทธิ์โพลีโนเมียลของเมทริกซ์ป้อนเข้า
rank	เป็นฟังก์ชันการคำนวณหาจำนวนชั้นของเมทริกซ์
rref	เป็นฟังก์ชันการคำนวณเมทริกซ์ให้อยู่ในรูปเมทริกซ์ลดรูปเป็นขั้น

ตัวอย่างที่ 3.4 การใช้ฟังก์ชันที่รองรับการคำนวณคลาส sym ดังภาพที่ 3.10



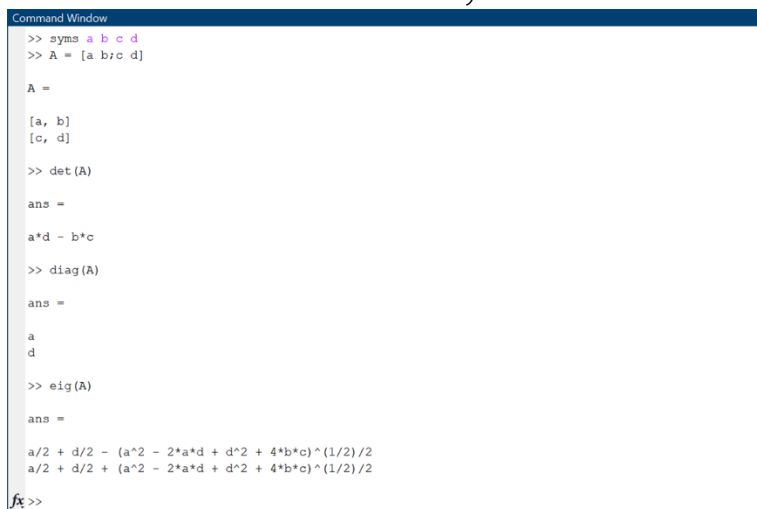
```

Command Window
X =
X1
>> Y=sym('Y1')
Y =
Y1
>> s=[x y:X1 Y1]
Undefined function or variable 'X1'.
Did you mean:
>> s=[x y:x1 y1]
s =
[ x1, y1]
[ x, y]
>> det(s)
ans =
x1*y - x*y1
fx >>

```

ภาพที่ 3.10 ฟังก์ชันที่รองรับการคำนวณคลาส sym

ตัวอย่างที่ 3.5 การใช้ฟังก์ชันที่รองรับการคำนวณคลาส syms ดังภาพที่ 3.11



```

Command Window
>> syms a b c d
>> A = [a b;c d]
A =
[a, b]
[c, d]
>> det(A)
ans =
a*d - b*c
>> diag(A)
ans =
a
d
>> eig(A)
ans =
a/2 + d/2 - (a^2 - 2*a*d + d^2 + 4*b*c)^(1/2)/2
a/2 + d/2 + (a^2 - 2*a*d + d^2 + 4*b*c)^(1/2)/2
fx >>

```

ภาพที่ 3.11 ฟังก์ชันที่รองรับการคำนวณคลาส syms

3.2.2 การหาแคลคูลัส

3.2.2.1 การหาค่าลิมิตของฟังก์ชัน $f(x)$ เมื่อ x เข้าใกล้ α สามารถเขียนเป็นสัญลักษณ์ทางคณิตศาสตร์ได้ดังนี้

$$y = \lim_{x \rightarrow \alpha} f(x)$$

สำหรับโปรแกรม MATLAB สามารถใช้ฟังก์ชัน `limit` ในการแก้ปัญหาสมการคำนวณ

ตัวอย่างที่ 3.6 จงหาค่าลิมิตของฟังก์ชันต่อไปนี้

1. $f(x) = \frac{1}{x}$ เมื่อ x เข้าสู่อินฟินิตี้

2. $f(x) = \frac{1}{x}$ เมื่อ x เข้าสู่ค่า 1

3. $f(x) = \frac{x-1}{x^2-1}$ เมื่อ x เข้าสู่ค่า 1

ผลการหาค่าลิมิตโดยใช้โปรแกรม MATLAB ดังภาพที่ 3.12

```

Command Window
>> syms x
>> f=1/x;
>> limit(f,x,inf)
ans =
0
>> limit(f,x,1)
ans =
1
>> f=(x-1)/(x^2-1)
f =
(x - 1)/(x^2 - 1)
>> limit(f,x,1)
Unrecognized function or variable 'limit'.
Did you mean:
>> limit(f,x,1)
ans =
1/2

```

Annotations in the image:

- Box 1: ข้อ 1 เมื่อ x เข้าสู่อินฟินิตี้ (Linked to `limit(f,x,inf)`)
- Box 2: ข้อ 2 เมื่อ x เข้าสู่ค่า 1 (Linked to `limit(f,x,1)`)
- Box 3: ข้อ 3 เมื่อ x เข้าสู่ค่า 1 (Linked to `limit(f,x,1)`)

ภาพที่ 3.12 การหาค่าลิมิตฟังก์ชัน $f(x)$ โดยใช้ `syms` ของโปรแกรม MATLAB

กรณีหาค่าลิมิต เมื่อ x มีค่าลู่เข้าหาค่าตอบทางขวา

จากสมการ $y = \lim_{x \rightarrow \infty} f(x)$ สามารถใช้ฟังก์ชัน limit ในการแก้ปัญหาคำนวณ

คำนวณ

ตัวอย่างที่ 3.7 จงหาค่าลิมิตของ $f(x) = \frac{1}{x}$ เมื่อ x ลู่เข้าสู่ค่า 0 ทางขวา ดังภาพที่ 3.13

```
Command Window
>> syms x
>> f=1/x;
>> limit(f,x,0,'right')
Unrecognized function or variable 'o'.

>> limit(f,x,0,'right')

ans =

Inf

fx>>
```

ภาพที่ 3.13 การหาค่าลิมิต เมื่อ x มีค่าลู่เข้าหาค่าตอบทางขวาโดยใช้ syms ของโปรแกรม MATLAB

กรณีหาค่าลิมิตเมื่อ x มีค่าลู่เข้าหาค่าตอบทางซ้าย

จากสมการ $y = \lim_{x \rightarrow \infty} f(x)$ สามารถใช้ฟังก์ชัน limit ในการแก้ปัญหาคำนวณ

คำนวณ

ตัวอย่างที่ 3.8 จงหาค่าลิมิตของ $f(x) = \frac{1}{x}$ เมื่อ x ลู่เข้าสู่ค่า 0 ทางซ้าย ดังภาพที่ 3.14

```
Command Window
>> syms x
>> f=1/x;
>> limit(f,x,1,'left')

ans =

1|

fx>>
```

ภาพที่ 3.14 การหาค่าลิมิต เมื่อ x มีค่าลู่เข้าหาค่าตอบทางซ้ายโดยใช้ syms ของโปรแกรม MATLAB

3.2.2.2 การหาอนุพันธ์

การหาอนุพันธ์ของฟังก์ชัน $f(x)$ เทียบกับ x สามารถเขียนเป็นสัญลักษณ์ทางคณิตศาสตร์ได้ดังนี้

$$y = \frac{d}{dx} f(x)$$

สำหรับโปรแกรม MATLAB สามารถใช้ฟังก์ชัน diff ในการแก้ปัญหาคำนวณ

ตัวอย่างที่ 3.9 จงหาค่าอนุพันธ์อันดับหนึ่งของ $f(x) = 2x^2 + 2x + 1$ ดังภาพที่ 3.15

```
Command Window
>> syms x
>> f=2*x^2+2*x+1;
>> diff(f)

ans =

4*x + 2

fx >>
```

ภาพที่ 3.15 การหาค่าอนุพันธ์อันดับหนึ่งโดยใช้ syms ของโปรแกรม MATLAB
การหาอนุพันธ์อันดับ n

จากสมการทางคณิตศาสตร์ $y = \frac{d}{dx} f(x, n)$ สามารถใช้ฟังก์ชัน diff ในการแก้
ปัญหาการคำนวณ

ตัวอย่างที่ 3.10 จงหาค่า $\frac{d^2}{dx^2} (2x^2 + 2x + 1)$ ดังภาพที่ 3.16

```
Command Window
>> syms x
>> f=2*x^2+2*x+1;
>> diff(f,2)

ans =

4

fx >>
```

ภาพที่ 3.16 การหาค่าอนุพันธ์อันดับสองโดยใช้ syms ของโปรแกรม MATLAB

การหาอนุพันธ์ย่อย

จากสมการทางคณิตศาสตร์ $y = \frac{\partial}{\partial s} f(s, t)$ สามารถใช้ฟังก์ชัน diff ในการแก้
ปัญหาการคำนวณ

ตัวอย่างที่ 3.11 จงหาค่า $\frac{\partial}{\partial s} (2s^2 + 2st + t^2)$ ดังภาพที่ 3.17

```
Command Window
>> syms s t
>> f=2*s^2+2*s*t+t^2;
>> diff(f,s)

ans =

4*s + 2*t

fx >>
```

ภาพที่ 3.17 การหาค่าอนุพันธ์ย่อยโดยใช้ syms ของโปรแกรม MATLAB

การหาอนุพันธ์ย่อยอันดับ n

จากสมการทางคณิตศาสตร์ $y = \frac{\partial^n}{\partial s^n} f(s, t)$ สามารถใช้ฟังก์ชัน `diff` ในการแก้
ปัญหาการคำนวณ

ตัวอย่างที่ 3.12 จงหาค่า $\frac{\partial^2}{\partial s^2} (2s^2 + 2st + t^2)$ ดังภาพที่ 3.18

```
Command Window
>> syms s t
>> f=2*s^2+2*s*t+t^2;
>> diff(f,s,2)

ans =

4

fx>>
```

ภาพที่ 3.18 การหาอนุพันธ์ย่อยโดยใช้ `syms` ของโปรแกรม MATLAB

3.2.2.3 การหาอินทิเกรต

การหาอินทิเกรตแบบไม่จำกัดเขต

การหาอินทิเกรตแบบไม่จำกัดเขตของฟังก์ชัน $f(x)$ เทียบกับ x สามารถเขียนเป็น
สมการทางคณิตศาสตร์ได้ดังนี้

$$y = \int f(x) dx$$

สำหรับโปรแกรม MATLAB สามารถใช้ฟังก์ชัน `int` ในการแก้ปัญหาการคำนวณ

ตัวอย่างที่ 3.13 จงหาค่า $\int x^3 dx$ ดังภาพที่ 3.19

```
Command Window
>> syms x
>> f=x^3;
>> int(f)

ans =

x^4/4

fx>> |
```

ภาพที่ 3.19 การหาอินทิเกรตแบบไม่จำกัดเขตโดยใช้ `syms` ของโปรแกรม MATLAB

การหาอินทิเกรตแบบจำกัดเขต

การหาอินทิเกรตแบบจำกัดเขตของฟังก์ชัน $f(x)$ เทียบกับ x สามารถเขียนเป็น
สมการทางคณิตศาสตร์ได้ดังนี้

$$y = \int_a^b f(x) dx$$

สำหรับโปรแกรม MATLAB สามารถใช้ฟังก์ชัน `int` ในการแก้ปัญหาคำนวณ

ตัวอย่างที่ 3.14 จงหาค่า $\int_1^4 x^3 dx$ ดังภาพที่ 3.20



```

Command Window
>> syms x
>> f=x^3;
>> int(f,1,4)

ans =

255/4

fx >>
    
```

ภาพที่ 3.20 การหาค่าอินทิเกรตแบบจำกัดเขตโดยใช้ `syms` ของโปรแกรม MATLAB

การหาอินทิเกรตแบบจำกัดเขตของฟังก์ชัน $f(s,t)$

การหาอินทิเกรตแบบจำกัดเขตของฟังก์ชัน $f(s,t)$ เทียบกับ s สามารถเขียนเป็นสมการทางคณิตศาสตร์ได้ดังนี้

$$y = \int_a^b f(s,t) ds$$

สำหรับโปรแกรม MATLAB สามารถใช้ฟังก์ชัน `int` ในการแก้ปัญหาคำนวณ

ตัวอย่างที่ 3.15 จงหาค่า $\int_1^2 (2s^2t + 3st^2) ds$ ดังภาพที่ 3.21



```

Command Window
>> syms s t
>> f=2*s^2*t+3*s*t^2;
>> int(f,s,1,2)

ans =

(t*(27*t + 28))/6

fx >> |
    
```

ภาพที่ 3.21 การหาค่าอินทิเกรตแบบจำกัดเขตของฟังก์ชัน $f(s,t)$ โดยใช้ `syms` ของโปรแกรม MATLAB

3.3 สรุป

การคำนวณทางคณิตศาสตร์เชิงตัวแปรของโปรแกรม MATLAB มีการประกาศตัวแปรเชิงวัตถุคลาส ซึ่งใช้คำสั่ง `sym` และ `syms` โดยปกติรูปแบบที่กำหนดมาในโปรแกรม MATLAB มีรูปแบบ Complex การระบุรูปแบบของตัวแปรสามารถทำได้โดยใช้คำสั่ง ดังนี้

`s=sym('x','format')` หรือ `syms x format`

สำหรับการจัดรูปนิพจน์ เป็นการจัดรูปแบบคำตอบให้สวยงามสามารถทำได้โดยใช้ฟังก์ชัน `pretty` อีกทั้งเมื่อทำการประกาศตัวแปรแล้ว สามารถทำการแก้สมการในการคำนวณทางคณิตศาสตร์เชิงตัวแปร พหุนามเชิงเส้น การแก้สมการแคลคูลัส การหาอนุพันธ์ การหาอนุพันธ์ย่อย การหาอินทิเกรต และการหาอินทิเกรตแบบจำกัดเขต

แบบฝึกหัด

บทที่ 3 การคำนวณและเปรียบเทียบโดยใช้โปรแกรม MATLAB

1. จงอธิบายการสร้างตัวแปรวัตถุคลาส sym ของโปรแกรม MATLAB
2. จงอธิบายการสร้างตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB
3. จงสร้างตัวแปรวัตถุคลาส sym และ syms ของโปรแกรม MATLAB จากสมการ $x^2 + 2x + 10$
4. จงอธิบายการดำเนินการทางพีชคณิตโดยใช้คลาส sym ของโปรแกรม MATLAB
5. กำหนดตัวแปร $x + x + x - y \times y + y^2$ จงสร้างตัวแปรวัตถุการดำเนินการทางพีชคณิตโดยใช้คลาส syms ของโปรแกรม MATLAB
6. กำหนดตัวแปร $\frac{x^4+4x^3+5x^2-6x+8}{x^3+x^2-x+2}$ จงจัดรูปนิพจน์ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB โดยใช้ฟังก์ชัน pretty
7. กำหนดตัวแปร $6x + 8$ จงแก้สมการเพื่อหาค่าตัวแปรวัตถุคลาส sym ของโปรแกรม MATLAB โดยใช้ฟังก์ชัน solve
8. กำหนดตัวแปร $(2xy + 3x^2) + x^2$ จงแก้สมการเพื่อหาค่าตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB โดยใช้ฟังก์ชัน solve
9. จงหาค่าลิมิต $f(x) = \frac{x-1}{x^2-1}$ เมื่อ x เข้าสู่อค่า 2 โดยใช้ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB
10. จงหาค่าลิมิต $f(x) = 7x$ เมื่อ x เข้าสู่อค่า 4 โดยใช้ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB
11. จงหาค่าลิมิต $f(x) = \frac{x^2+2x-3}{(x-1)}$ เมื่อ x เข้าสู่อค่า 1 โดยใช้ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB

12. จงหาค่าอนุพันธ์ของ $f(x) = x^4 + 4x^3 + 5x^2 - 6x + 8$ โดยใช้ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB

13. จงหาค่าอนุพันธ์ของ $f(x) = 2x^6 + 3x^2 + 3x$ โดยใช้ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB

14. จงหาค่าอินทิเกรตของ $f(x) = \frac{xdx}{\sqrt{a^2-x^2}}$ โดยใช้ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB

15. จงหาค่าอินทิเกรตของ $\int_1^3 (2x^3t + 6xt^2) dx$ โดยใช้ตัวแปรวัตถุคลาส syms ของโปรแกรม MATLAB

บทที่ 4

การเขียนโปรแกรมย่อยด้วย M-File ของโปรแกรม MATLAB

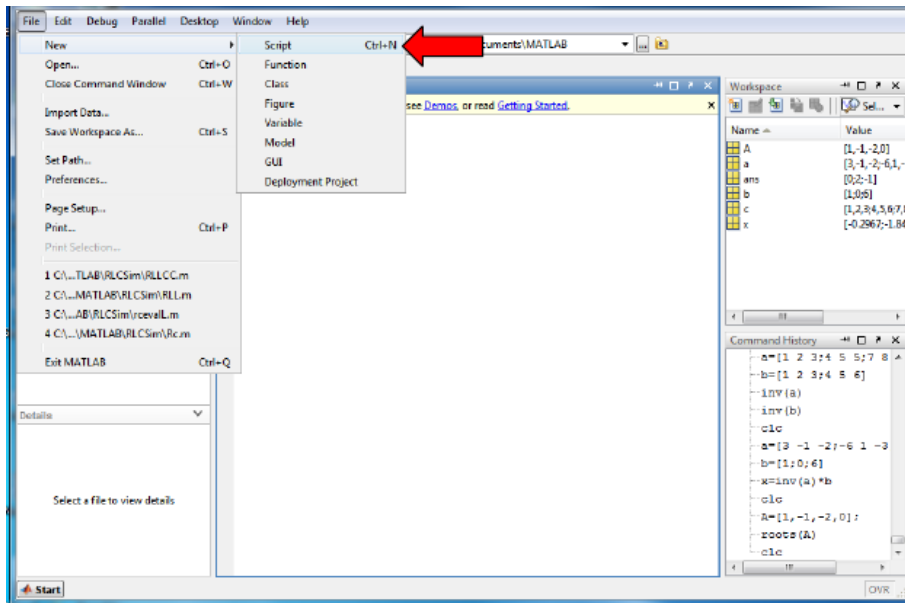
หลังจากที่ได้กล่าวถึงรูปแบบ คำสั่ง และฟังก์ชันพื้นฐานสำหรับการเขียนโปรแกรม MATLAB บนหน้าต่าง Command Window ไปแล้วนั้น จะเห็นได้ว่า การเขียน Script ของโปรแกรม MATLAB มีลักษณะการทำงานเป็นแบบบรรทัด แต่การเขียนโค้ดสามารถเขียนได้หลายวิธี เช่น การเขียนโปรแกรมโดยใช้ Script M-File การเขียนโปรแกรมโดยใช้ Function M-File เป็นต้น ซึ่งการเขียนโปรแกรมแต่ละแบบมีหลักการทำงานที่แตกต่างกัน ทั้งนี้ การเขียน Script มีข้อจำกัดคือ ไม่สามารถส่งค่าจากโปรแกรมหนึ่งไปอีกโปรแกรมหนึ่งได้ จึงทำให้ต้องมีการเขียนโปรแกรมย่อยด้วยวิธีต่าง ๆ

การเขียนโปรแกรมโดยใช้ Script M-File ข้อดีคือ ผู้เขียนสามารถเขียน Script เป็นโปรแกรมย่อย ๆ ขึ้นมาและแก้ไขชุดคำสั่งได้ โดยไม่จำเป็นต้องป้อนคำสั่งใหม่ที่ละคำสั่งเหมือนกับบนหน้าต่าง Command Window เมื่อผู้เขียนแก้ไขตัวแปรหรือคำสั่งแล้วสามารถทำการทดสอบการทำงานของโปรแกรมได้ทันที สำหรับการเขียนโปรแกรมโดยใช้ Function M-File ข้อดีคือ ผู้เขียนสามารถสร้างฟังก์ชันการคำนวณได้ด้วยตนเอง และนำมาใช้กับการเขียนโปรแกรมโดยใช้ Script M-File ได้

4.1 การเขียนโปรแกรมโดยใช้ Script M-File

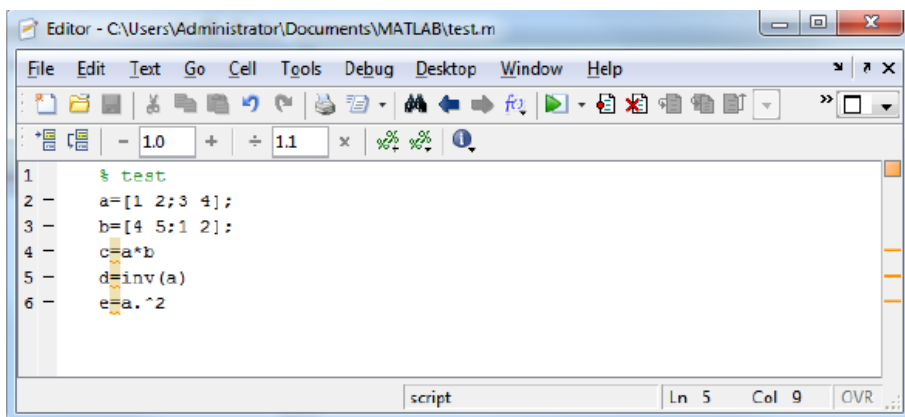
การเขียน Script File เป็นการรวมเอาคำสั่งต่าง ๆ ของโปรแกรม MATLAB ที่เกี่ยวข้องกับการคำนวณนั้น ๆ ไว้ในไฟล์ที่มีนามสกุล .m หรือ M-File ซึ่งมีโครงสร้างภายในเป็น ASCII text file สามารถเรียก M-File ขึ้นมาแก้ไขชุดคำสั่งที่ได้บันทึกไว้ โดยผู้ใช้ไม่จำเป็นต้องป้อนคำสั่งใหม่ที่ละคำสั่งอีกต่อไป เมื่อผู้ใช้แก้ไขตัวแปรหรือคำสั่งต่าง ๆ ใน Editor แล้ว สามารถสั่งให้โปรแกรมทำงานใหม่เพื่อทดสอบการทำงานของโปรแกรมได้ทันที ในการกำหนดค่าตัวแปรและเขียนคำสั่งหรือฟังก์ชันต่าง ๆ ที่ต้องการประมวลผลที่ละคำสั่งลงใน Text File ที่เป็นไฟล์นามสกุล .m แล้วสามารถทำการประมวลผล (run) ได้ด้วยการพิมพ์ชื่อไฟล์โดยไม่ต้องตามด้วยนามสกุล .m ลงในหน้าต่างคำสั่ง Command Window ซึ่งไฟล์เหล่านี้เรียกว่า Script File หรือ M-File อย่างง่าย โดยการสร้าง Script File สามารถทำได้โดยการเลือกเมนูบนหน้าต่างคำสั่ง ดังดังนี้

1. เลือกที่เมนู File >> New >> Script โดยเมื่อเลือกแล้ว จะปรากฏหน้าต่างของโปรแกรม MATLAB Editor/Debugger ขึ้นมา ดังภาพที่ 4.1



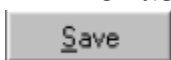
ภาพที่ 4.1 การเปิดหน้าต่าง M-File

2. ทำการกำหนดค่าตัวแปรและฟังก์ชันที่ต้องการหาค่าลงในหน้าต่าง MATLAB Editor/Debugger ดังภาพที่ 4.2

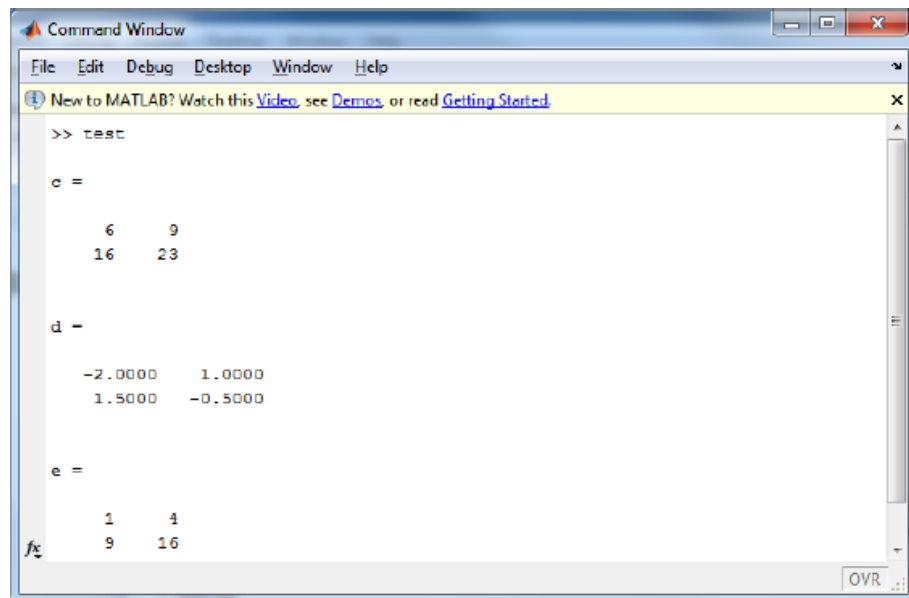


ภาพที่ 4.2 การกำหนดตัวแปรบนหน้าต่าง M-File

3. ทำการเลือกเมนู File >> Save as เพื่อทำการบันทึกข้อมูลโปรแกรมที่เขียนไว้เป็นครั้งแรก
4. ทำการกำหนดชื่อไฟล์เป็น test.m
5. เมื่อมีการเพิ่มเติมหรือเปลี่ยนแปลงข้อมูลต้องทำการ save ข้อมูลทุกครั้ง โดยการกดปุ่ม



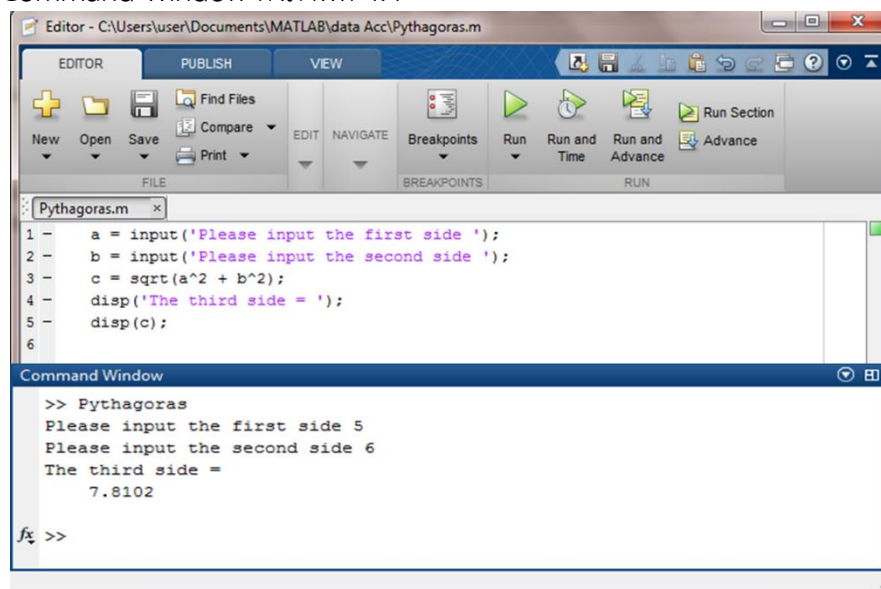
เมื่อต้องการประมวลผลของโปรแกรมที่เขียนขึ้นก็สามารถทำได้โดยพิมพ์ชื่อไฟล์โดยไม่ต้องตามด้วยนามสกุล .m ลงบนหน้าต่างคำสั่ง Command Window ดังภาพที่ 4.3



ภาพที่ 4.3 การประมวลผลบนหน้าต่างคำสั่ง Command Window

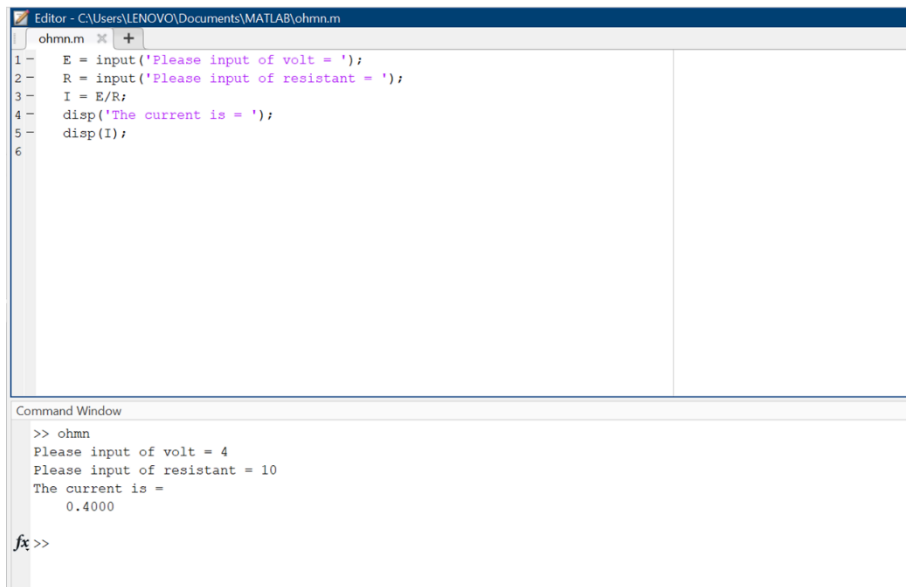
การประมวลผล (run) โปรแกรม Script File พบว่า ค่าตัวแปรทุกตัวในโปรแกรมจะปรากฏบน Work Space

ตัวอย่างที่ 4.1 จงเขียนโปรแกรมคำนวณความยาวด้านตรงข้ามมุมฉาก เมื่อป้อนข้อมูลของด้านประกอบมุมฉากทั้งสอง ตามกฎของ Pythagoras โดยเขียนโปรแกรมบน Script File แล้วแสดงออกหน้าต่าง Command Window ดังภาพที่ 4.4



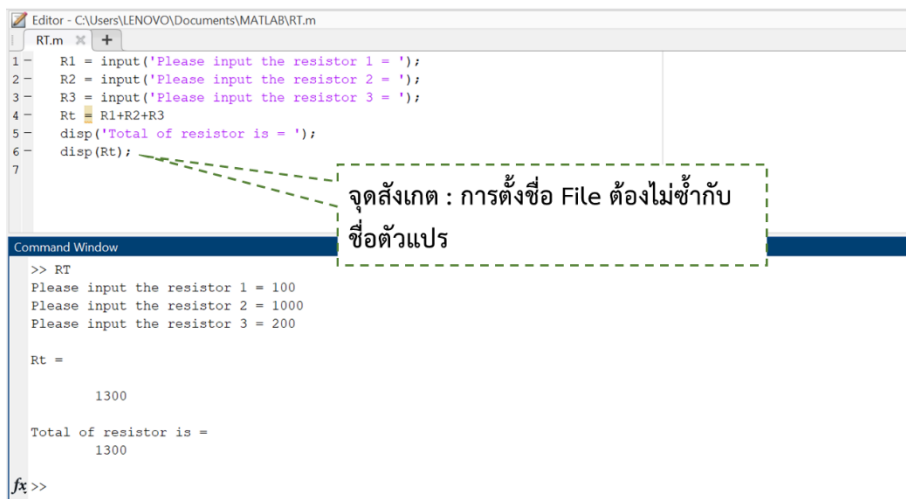
ภาพที่ 4.4 การเขียน Script File และแสดงผลลัพธ์ของค่าความยาวด้านตรงข้ามมุมฉาก

ตัวอย่างที่ 4.2 จงเขียนโปรแกรมคำนวณหาค่ากระแสไฟฟ้าในวงจร เมื่อป้อนแรงดันไฟฟ้าและตัวต้านทาน ตามกฎของโอห์ม โดยเขียนโปรแกรมบน Script File แล้วแสดงออกหน้าต่าง Command Window ดังภาพที่ 4.5



ภาพที่ 4.5 การเขียน Script File และแสดงผลลัพธ์ของค่ากระแสไฟฟ้าในวงจร

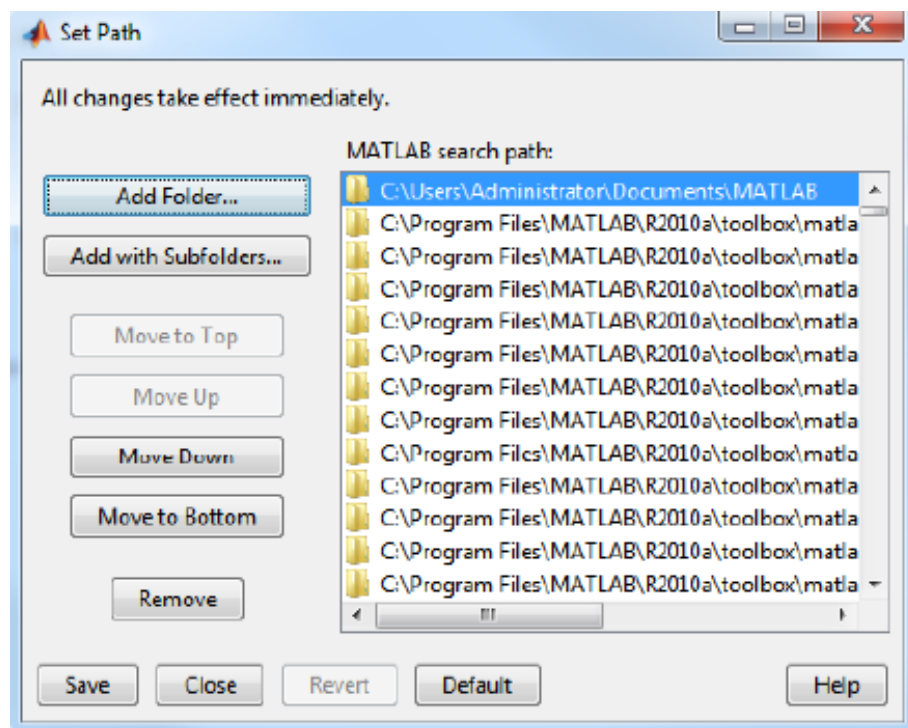
ตัวอย่างที่ 4.3 จงเขียนโปรแกรมคำนวณหาค่าความต้านทานรวม เมื่อป้อนค่าความต้านทานทั้ง 3 ตัว โดยเขียนโปรแกรมบน Script File แล้วแสดงออกหน้าต่าง Command Window ดังภาพที่ 4.6



ภาพที่ 4.6 การเขียน Script File และแสดงผลลัพธ์ของค่าความต้านทานรวม

4.2 การจัดการไฟล์และไดเรกทอรี

โปรแกรม MATLAB จะใช้ Search Path เพื่อใช้ในการหาสคริปต์และฟังก์ชัน M-File ที่ได้สร้างไว้ในโฟลเดอร์และไดเรกทอรีย่อย รวมทั้งรายการของไดเรกทอรีต่าง ๆ ทั้งหมดที่มี M-File อยู่ ที่เรียกว่า MATLAB Search Path หรือ MATLABpath การทำงานของคำสั่งหรือฟังก์ชัน M-File ต่าง ๆ ตาม Path ที่ตั้งไว้ในไดเรกทอรีปัจจุบันเท่านั้น การตรวจสอบไดเรกทอรีปัจจุบัน และไดเรกทอรีที่ได้กำหนดไว้ใน Search Path ที่มีไฟล์ต่าง ๆ อยู่ ตลอดจนทำการเปลี่ยนไดเรกทอรีปัจจุบัน จาก Path Browser สามารถกระทำได้โดยพิมพ์คำสั่ง Editpath บนหน้าต่างคำสั่งหรือเลือกที่เมนู File >> SetPath ปรากฏหน้าต่างดังภาพที่ 4.7



ภาพที่ 4.7 หน้าต่าง SetPath

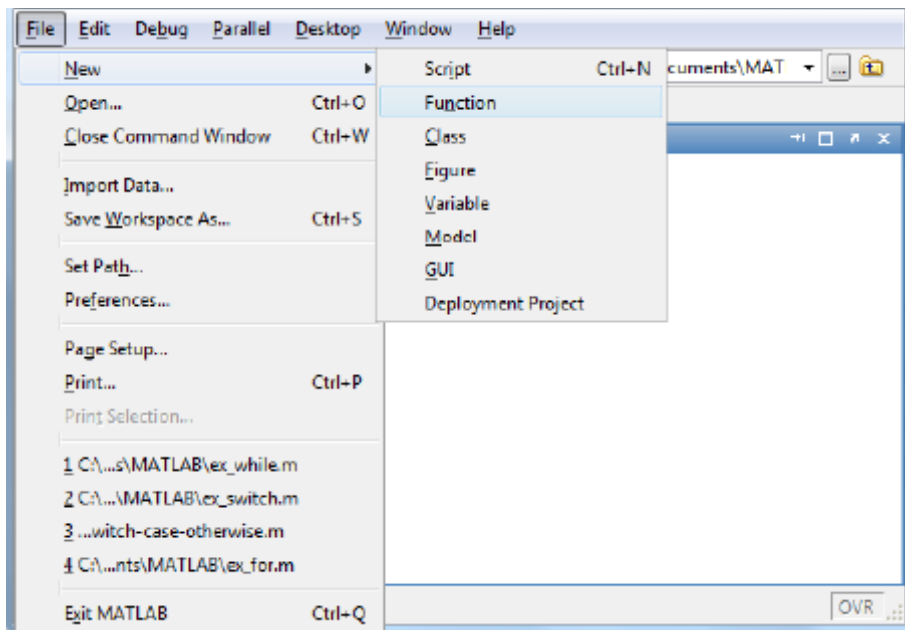
จากภาพที่ 4.7 หน้าต่าง Path Browser สามารถเลือกไดเรกทอรีปัจจุบัน โดยที่หน้าต่าง คำสั่งแสดงไดเรกทอรีปัจจุบันที่ Current Directory สามารถตั้งค่าได้ดังนี้

1. การเปลี่ยนไดเรกทอรีสามารถทำได้โดยคลิกที่ปุ่ม Addfolder... จะทำการเพิ่มโฟลเดอร์ เข้าไป เพื่อให้โปรแกรม MATLAB สามารถค้นหาฟังก์ชันต่าง ๆ ที่จะกระทำคำสั่งหรือฟังก์ชัน M-File ต่าง ๆ ที่จะกระทำคำสั่งหรือฟังก์ชัน M-File ต่าง ๆ ที่ต้องการได้
2. การลบได้โดยการเลือกโฟลเดอร์แล้วคลิกที่ปุ่ม Remove ซึ่งจะทำการลบโฟลเดอร์ที่ไม่ ต้องการให้โปรแกรมค้นหาออกไป
3. การตั้งโฟลเดอร์ให้ใช้โฟลเดอร์ เริ่มต้นโดยคลิกที่ปุ่ม Default

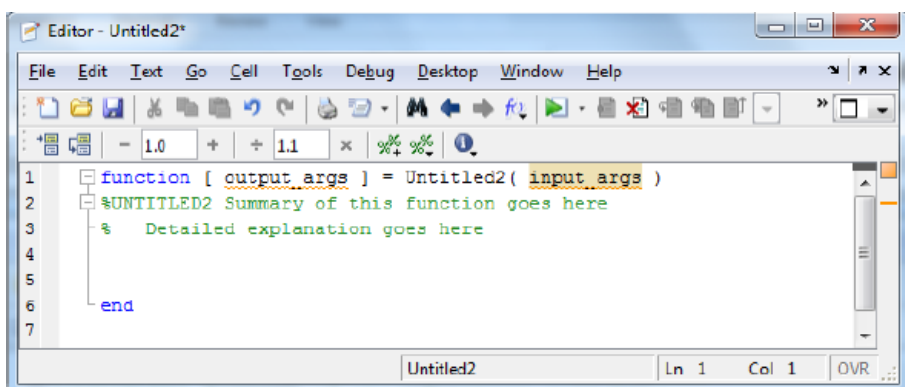
4.3 การเขียนโปรแกรมโดยใช้ Function M-File

4.3.1 การเรียกใช้ Function M-File

โปรแกรม MATLAB ได้สร้าง Function M-File ที่เป็นเครื่องมือในการคำนวณด้านคณิตศาสตร์และสาขาอื่น ๆ ทั้งนี้เพื่อความสะดวกในการเขียนโปรแกรม เช่น `inv`, `abs` เป็นต้น และฟังก์ชันที่มีอยู่ไม่สามารถแก้ปัญหาได้โดยตรงแต่สามารถนำฟังก์ชันต่าง ๆ เหล่านี้มาประกอบเป็น M-File ซึ่งฟังก์ชันเหล่านั้นสามารถสร้างขึ้นเองได้โดยใช้ M-File ฟังก์ชันการเรียกใช้งานไปที่เมนู File >> New >> Function ดังภาพที่ 4.8 และปรากฏหน้าต่างการเขียน Function ในภาพดังภาพที่ 4.9



ภาพที่ 4.8 การเรียกใช้งาน Function M-File

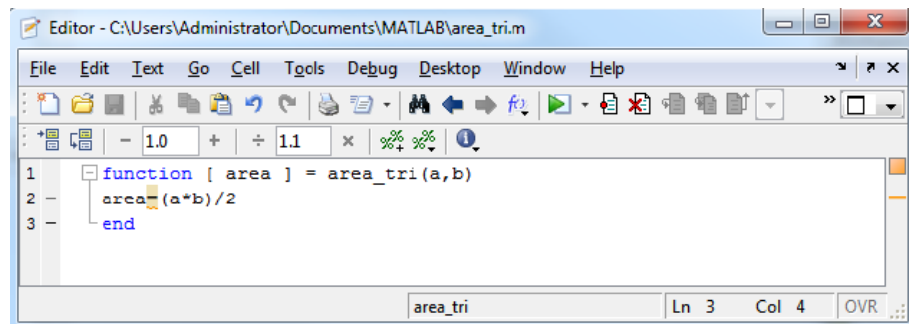


ภาพที่ 4.9 หน้าต่างสำหรับการเขียน Function

จากภาพที่ 4.9 โปรแกรม MATLAB กำหนดโครงสร้างของ Function มาให้ ประกอบด้วยตัวแปรเอาต์พุต (output_args) ชื่อฟังก์ชัน (untitled2) และตัวแปรอินพุต (input_args)

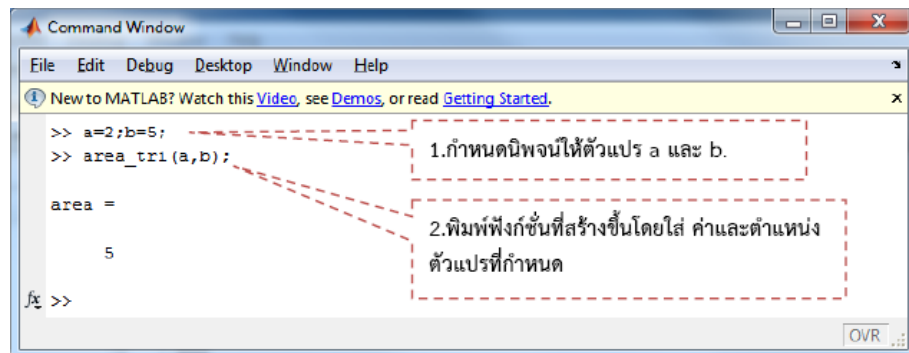
4.3.2 การสร้าง Function M-File

การสร้างฟังก์ชันสามารถทำได้โดยการประกาศให้เป็นฟังก์ชันด้วยคำสั่ง Function ตามด้วยตัวแปรเข้าออก โดยมีโครงสร้างดังภาพที่ 4.10



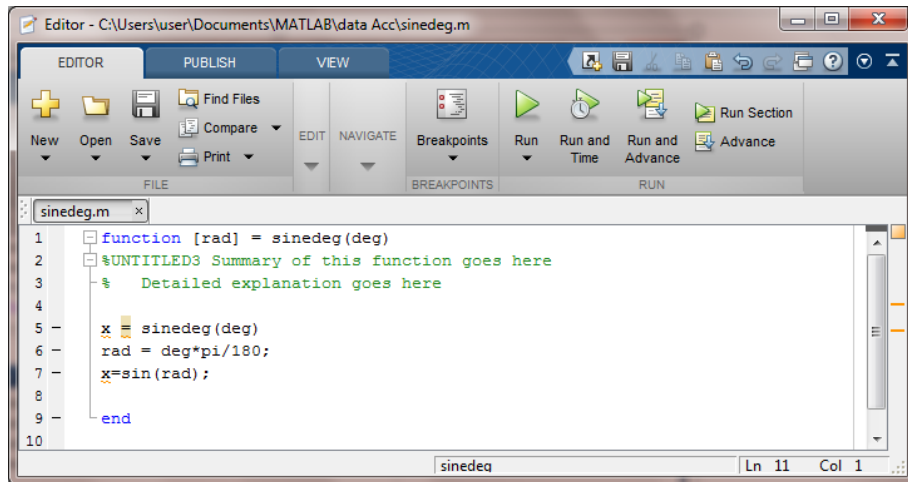
ภาพที่ 4.10 การสร้าง Function M-File

จากภาพที่ 4.10 สามารถทดสอบฟังก์ชันที่เขียนขึ้น โดยการกำหนดค่าให้กับตัวแปร a และ b บน Command Window หลังจากนั้นก็เรียกชื่อไฟล์ที่บันทึกไว้ ดังภาพที่ 4.11



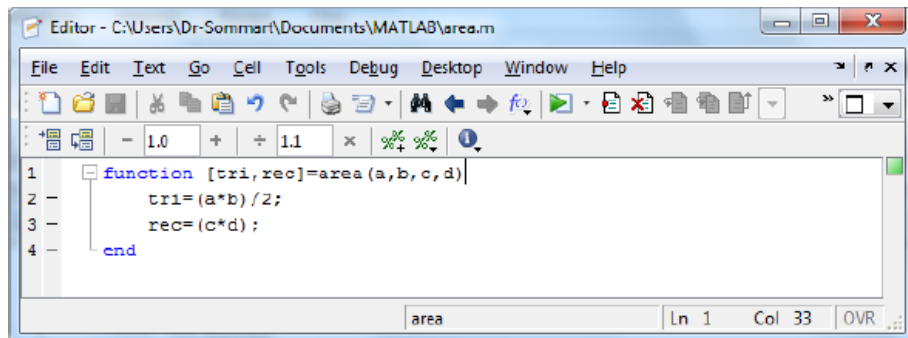
ภาพที่ 4.11 การเรียกใช้ M-File ฟังก์ชัน

ตัวอย่างที่ 4.4 จงคำนวณหาค่า Sine ของมุมในหน่วยองศา โดยให้มีชื่อว่า sinedeg แล้วทำการบันทึกชื่อไฟล์ว่า sinedeg.m โดยเขียนโปรแกรมบน Function File แล้วแสดงออกหน้าต่าง Command Window ดังภาพที่ 4.12



ภาพที่ 4.12 การเขียน Function File ของค่า Sine

ตัวอย่างที่ 4.5 จงคำนวณหาพื้นที่ของรูปสามเหลี่ยมและพื้นที่ของรูปสี่เหลี่ยมผืนผ้า โดยเขียนโปรแกรมบน Function File แล้วแสดงออกหน้าต่าง Command Window ดังภาพที่ 4.13



ภาพที่ 4.13 การเขียน Function File คำนวณหาพื้นที่ของรูปสามเหลี่ยมและพื้นที่ของรูปสี่เหลี่ยมผืนผ้า

4.4 สรุป

การเขียนโปรแกรมย่อย M-File แบบ script เหมาะสำหรับการฝึกหัดเขียนโปรแกรมที่มีขนาดเล็ก เพราะหากต้องการให้ script หลายไฟล์สามารถทำงานร่วมกันได้ ต้องอาศัยเก็บข้อมูลตัวแปรไว้ใน workspace แล้วค่อยให้โปรแกรมถัดไป มาอ่านข้อมูลตัวแปรที่อยู่ใน workspace ไปคำนวณต่อ ซึ่งเมื่อตั้งชื่อตัวแปรซ้ำกันอาจทำให้ script ประมวลการทำงานผิดพลาด

สำหรับการเขียนโปรแกรมแบบ function เหมาะสำหรับการเขียนโปรแกรมขนาดใหญ่ที่มีความซับซ้อน เพราะสามารถเขียนโปรแกรมรับค่าและส่งค่าภายใน function เดียวกันได้ และเขียนหลายโปรแกรมไว้ในไฟล์เดียวกันได้ โดยการเขียนคำว่า "function" ไว้บรรทัดบนสุด และเขียนคำว่า "end" ไว้บรรทัดล่างสุดแบบนี้ก็ได้ไฟล์ประเภทฟังก์ชันแล้วครับ

แบบฝึกหัด

บทที่ 4 การเขียนโปรแกรมย่อยด้วย M-File ของโปรแกรม MATLAB

1. การเขียนโปรแกรมโดยใช้ Script M-File กับ Function M-File แตกต่างกันอย่างไร
2. จงอธิบายการจัดการไฟล์และไดเรกทอรีของโปรแกรม MATLAB
3. จงบอกข้อจำกัดในการตั้งชื่อ File ของโปรแกรม MATLAB พร้อมอธิบาย
4. จงเขียนโปรแกรมคำนวณหาค่าความต้านทานรวม (R_{total}) ของวงจรขนานที่มีตัวต้านทานต่อขนานกัน 4 ตัว โดยเขียนโปรแกรมบน Script File แล้วแสดงออกหน้าต่าง Command Window
5. จงเขียนโปรแกรมคำนวณหาค่าความต้านทานในวงจร จากกฎของโอห์ม เมื่อกำหนดให้ป้อนค่าวงจรไฟฟ้า และกระแสไฟฟ้าในวงจร โดยเขียนโปรแกรมบน Script File แล้วแสดงออกหน้าต่าง Command Window
6. จงเขียนโปรแกรมคำนวณหาปริมาณการใช้ไฟฟ้าภายในบ้าน เมื่อกำหนดให้ป้อนค่าความต้านทานค่าแรงดันไฟฟ้า และเวลาการใช้งาน โดยเขียนโปรแกรมบน Script File แล้วแสดงออกหน้าต่าง Command Window มีสูตรดังนี้

$$W (kWh) = P (kW) \times t (h)$$

7. จงเขียนโปรแกรมคำนวณค่ากระแส โดยกำหนดให้ค่าความต้านทานรวม เท่ากับ 10 โอห์ม และค่าแรงดัน เท่ากับ 20 โวลต์ โดยเขียนโปรแกรมบน Function File แล้วแสดงออกหน้าต่าง Command Window
8. จงคำนวณหาค่า Cosine ของมุมในหน่วยองศา โดยให้มีชื่อว่า cosine แล้วทำการบันทึกชื่อไฟล์ว่า cosinedeg.m โดยเขียนโปรแกรมบน Function File แล้วแสดงออกหน้าต่าง Command Window ดังภาพที่ 4.12
9. จงเขียนโปรแกรมคำนวณค่าความต้านทานของวงจร โดยกำหนดให้ในวงจรไฟฟ้ามีแรงดันไฟฟ้า 15 โวลต์ ต่ออนุกรมกับตัวต้านทานที่มีกระแสไฟฟ้า 3 แอมแปร์ ไหลผ่านวงจร โดยเขียนโปรแกรมบน Function File แล้วแสดงออกหน้าต่าง Command Window

10. จงเขียนโปรแกรมคำนวณหาความเร่งที่เกิดจากแรงกระทำที่ 50 นิวตัน โดยกำหนดให้วัตถุหนึ่งถูกแรง 100 นิวตัน กระทำแล้วเกิดความเร่ง 10 เมตร/วินาที² ตามกฎของนิวตัน โดยเขียนโปรแกรมบน Function File แล้วแสดงออกหน้าต่าง Command Window

$$F = ma$$

11. จงเขียนโปรแกรมคำนวณหาน้ำหนักที่เครื่องชั่งสามารถอ่านได้ โดยกำหนดให้ชายคนหนึ่งมีน้ำหนัก 50 กิโลกรัม ยืนบนเครื่องชั่งที่กำลังเคลื่อนที่ขึ้นด้วยความเร่ง 2m/s² ตามกฎของนิวตัน โดยเขียนโปรแกรมบน Function File แล้วแสดงออกหน้าต่าง Command Window

12. จงเขียนโปรแกรมคำนวณค่าความต้านทานของวงจรอนุกรมที่มีตัวต้านทานต่อกัน 5 ตัว โดยผู้ใช้งานสามารถป้อนค่าตัวต้านทานทีละตัวได้ โดยกำหนดรายละเอียดดังนี้

12.1 เขียนโปรแกรมหลัก (Main Program) เพื่อใช้ในการกำหนดค่าตัวต้านทานแต่ละตัวบน Script File

12.2 เขียนโปรแกรมคำนวณ (Calculation) เพื่อใช้ในการคำนวณค่าตัวต้านทานรวมบน Function File

12.3 ให้แสดงผลลัพธ์ออกทางหน้าต่าง Command Window

13. จงเขียนโปรแกรมคำนวณหา กำลังไฟฟ้า โดยผู้ใช้งานสามารถป้อนค่าต่าง ๆ ได้ โดยกำหนดรายละเอียดดังนี้

$$\text{พลังงานไฟฟ้า (จูล)} = \text{กำลังไฟฟ้า (วัตต์)} \times \text{เวลา (วินาที)}$$

13.1 เขียนโปรแกรมหลัก (Main Program) เพื่อใช้ในการกำหนดค่าพลังงานไฟฟ้า และเวลาที่ใช้งาน

13.2 เขียนโปรแกรมคำนวณ (Calculation) เพื่อใช้ในการคำนวณค่ากำลังไฟฟ้า

13.3 ให้แสดงผลลัพธ์ออกทางหน้าต่าง Command Window

14. จงเขียนโปรแกรมคำนวณค่าความต้านทานของวงจรขนานที่มีตัวต้านทานต่อกัน 5 ตัว โดยผู้ใช้งานสามารถป้อนค่าตัวต้านทานทีละตัวได้ โดยกำหนดรายละเอียดดังนี้

14.1 เขียนโปรแกรมหลัก (Main Program) เพื่อใช้ในการกำหนดค่าตัวต้านทานแต่ละตัวบน Script File

14.2 เขียนโปรแกรมคำนวณ (Calculation) เพื่อใช้ในการคำนวณค่าตัวต้านทานรวมบน Function File

14.3 ให้แสดงผลลัพธ์ออกทางหน้าต่าง Command Window

15. จงเขียนโปรแกรมคำนวณหาพลังงานไฟฟ้า โดยผู้ใช้งานสามารถป้อนค่าต่าง ๆ ได้ โดยกำหนดรายละเอียดดังนี้

$$\text{พลังงานไฟฟ้า (จูล)} = \text{กำลังไฟฟ้า (วัตต์)} \times \text{เวลา (วินาที)}$$

15.1 เขียนโปรแกรมหลัก (Main Program) เพื่อใช้ในการกำหนดค่ากำลังไฟฟ้า และเวลาที่ใช้งาน

15.2 เขียนโปรแกรมคำนวณ (Calculation) เพื่อใช้ในการคำนวณค่าพลังงานไฟฟ้า

15.3 ให้แสดงผลลัพธ์ออกทางหน้าต่าง Command Window

การเขียนโปรแกรม MATLAB แบบโครงสร้าง

การเขียนโปรแกรมเป็นกระบวนการใช้คอมพิวเตอร์เพื่อกำหนดโครงสร้างของข้อมูลและกำหนดขั้นตอนวิธี เพื่อใช้แก้ปัญหาที่ได้ออกแบบไว้ โดยอาศัยหลักเกณฑ์การเขียนโปรแกรมคอมพิวเตอร์แต่ละภาษา การเขียนโปรแกรมมี 3 แบบ ได้แก่ การเขียนโปรแกรมโครงสร้าง (Structure Programming) การเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming : OOP) และการเขียนโปรแกรมเชิงจินตภาพ (Virus Programming) ซึ่งการเขียนโปรแกรมแต่ละแบบมีแนวคิด และวิธีการเขียนโปรแกรมงานอย่างง่ายแตกต่างกันไป

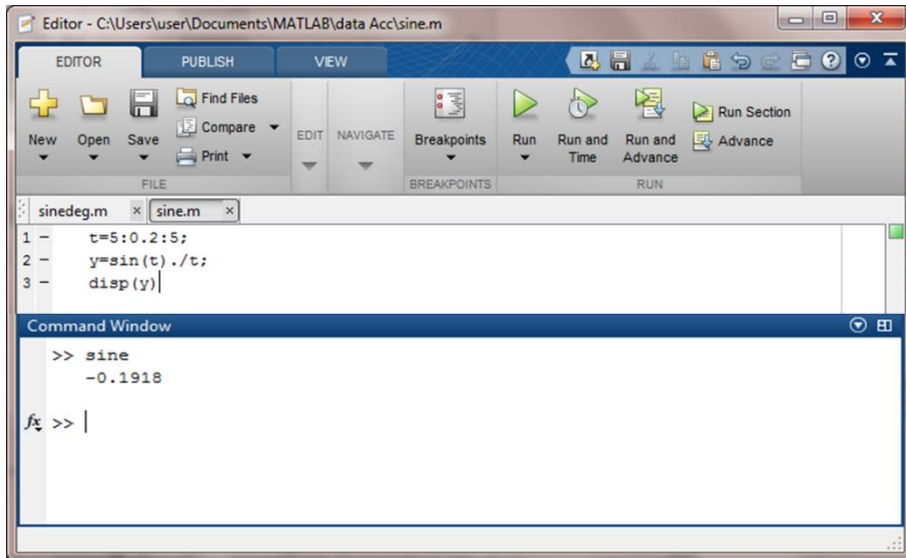
สำหรับบทนี้ขอแนะนำการเขียนโปรแกรมแบบโครงสร้าง (Structure Programming) ซึ่งเป็นการเขียนโปรแกรมที่ผู้เขียนสามารถนำมาใช้งานตามลักษณะของปัญหาความสามารถและความถนัดของตนเอง โปรแกรม MATLAB มีรูปแบบคำสั่งและหลักการในการเขียนโปรแกรม 3 แบบ ได้แก่ โครงสร้างแบบตามลำดับ (Sequential Structure) โครงสร้างแบบมีทางเลือกกระทำตามเงื่อนไข (Selection Structure) และโครงสร้างแบบการทำซ้ำ (Repetition Structure) ถึงแม้ว่าปัจจุบันแนวคิดเกี่ยวกับการเขียนโปรแกรมเปลี่ยนไป แต่แนวคิดเกี่ยวกับวิธีการเขียนโปรแกรมแบบโครงสร้างก็ยังเป็นความรู้พื้นฐานในการเขียนโปรแกรมแบบสมัยใหม่

5.1 การทำงานแบบตามลำดับ

โครงสร้างที่ทำงานแบบตามลำดับเป็นขั้นตอนเพื่อแก้ปัญหาจะทำงานตามคำสั่งที่เขียนไว้ตามลำดับ ตั้งแต่คำสั่งแรกไปจนถึงคำสั่งสุดท้าย โดยที่คำสั่งในที่นี้อาจเป็นคำสั่งเดี่ยว ๆ หรือเป็นคำสั่งเชิงซ้อนที่มีหลายคำสั่งย่อยประกอบกันในลักษณะเป็นโครงสร้างแบบทางเลือกหรือแบบวนซ้ำการทำงานแบบตามลำดับนั้น เมื่อเขียนเป็นผังงาน มีกระบวนการทำงานพื้นฐานอยู่ 3 ชนิด ได้แก่

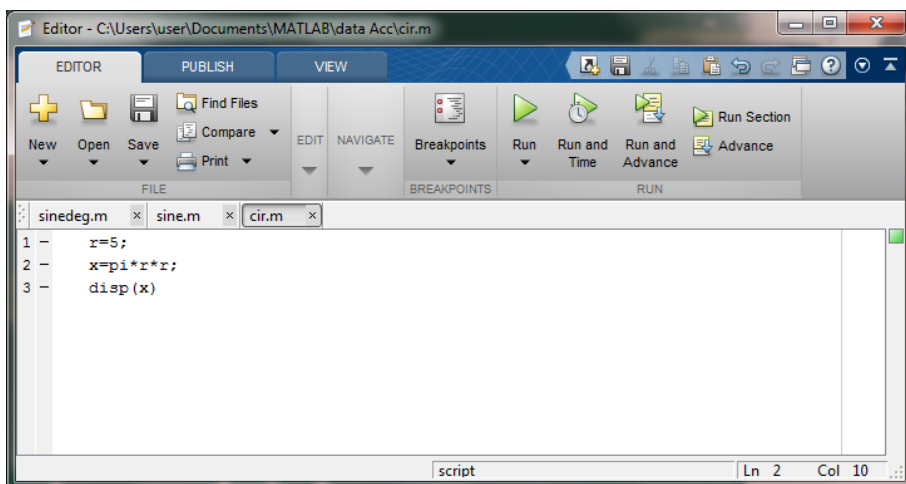
1. การรับข้อมูลเข้า เป็นกระบวนการรับข้อมูลจากอุปกรณ์ของหน่วยรับเข้า เช่น คีย์บอร์ด เพื่อนำค่าไปกำหนดให้กับตัวแปรและเก็บไว้ในหน่วยความจำ
 2. การคำนวณ เป็นกระบวนการที่คอมพิวเตอร์ทำการคำนวณ ประมวลผล ซึ่งรวมไปถึงการกำหนดค่าให้กับตัวแปร เพื่อให้สามารถนำค่าของตัวแปรนั้นมาใช้ในภายหลังได้
 3. การส่งข้อมูลออก เป็นกระบวนการนำค่าของข้อมูลไปแสดงผลยังอุปกรณ์ของหน่วยส่งออก เช่น จอภาพหรือเครื่องพิมพ์ ข้อมูลที่จะส่งออกโดยทั่วไปจะเป็นค่าคงที่ หรือค่าของตัวแปร
- การดำเนินการเพื่อแก้ปัญหาด้วยคอมพิวเตอร์ กระบวนการเหล่านี้ต้องถูกเปลี่ยนให้อยู่รูปของคำสั่งหลายคำสั่งประกอบกันเพื่อให้ทำงานตามขั้นตอนที่ได้ออกแบบไว้

ตัวอย่างที่ 5.1 กำหนดให้ตัวแปร $t = 5:0.2:5$ จงเขียน Script File เพื่อคำนวณค่า $y = \frac{\sin(t)}{t}$ โดยให้แสดงผลลัพธ์ของค่า y ออกหน้าจอ ดังภาพที่ 5.1



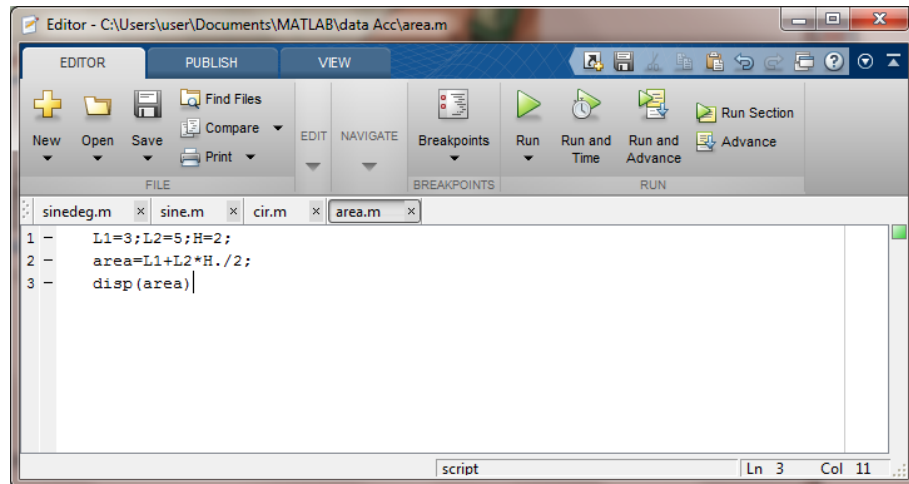
ภาพที่ 5.1 การเขียน Script File และแสดงผลลัพธ์ของค่า y ออกหน้าจอ Command Window

ตัวอย่างที่ 5.2 กำหนดให้ตัวแปร $r = 5$ จงเขียน Script File เพื่อคำนวณหาค่าพื้นที่วงกลม ดังภาพที่ 5.2



ภาพที่ 5.2 การเขียน Script File ของค่าพื้นที่วงกลม

ตัวอย่างที่ 5.3 กำหนดให้ตัวแปร $L1 = 3$ และ $L2 = 5$ เป็นความยาวของด้านคู่ขนาน และ $H = 2$ เป็นความสูง เพื่อคำนวณหาพื้นที่สี่เหลี่ยมคางหมู ดังภาพที่ 5.3



ภาพที่ 5.3 การเขียน Script File ของค่าพื้นที่สี่เหลี่ยมคางหมู

5.2 การเลือกกระทำตามเงื่อนไข

เป็นการตัดสินใจเพื่อเลือกที่จะใช้วิธีการใด โดยต้องมีการตรวจสอบว่าเงื่อนไขที่ใช้ในการตัดสินใจว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงจะไปเลือกทำคำสั่งชุดหนึ่ง แต่ถ้าเป็นเท็จจะไปเลือกทำคำสั่งอีกชุดหนึ่ง ซึ่งชุดคำสั่งเหล่านี้จะประกอบด้วยโครงสร้างแบบลำดับนั่นเอง

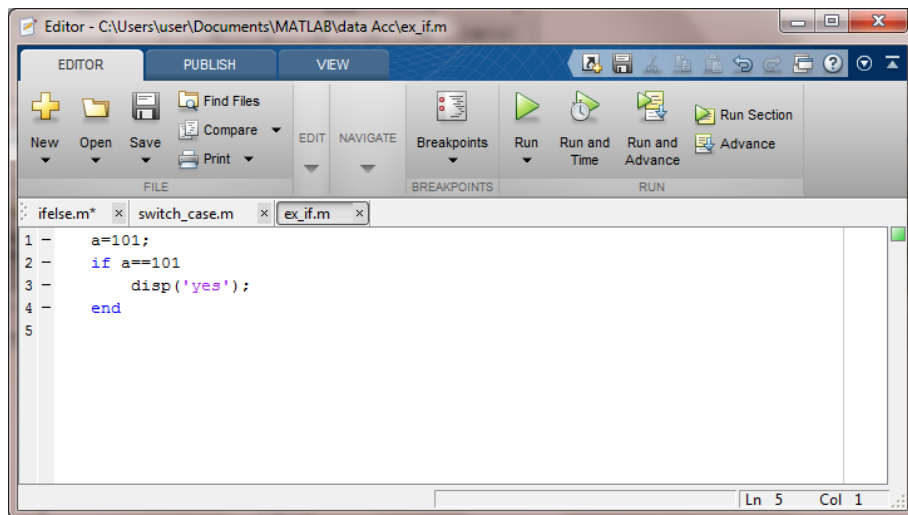
โครงสร้างการทำงานแบบทางเลือกกระทำตามเงื่อนไข อาจเรียกว่า มีโครงสร้างการทำงานแบบ if...then...else... ซึ่งเป็นการเลือกทำแบบทางใดทางหนึ่ง เนื่องจากต้องมีการเลือกทำชุดคำสั่งใดชุดคำสั่งหนึ่งตามผลของเงื่อนไข แต่ในการเขียนโปรแกรมหรือผังงาน เพื่อแก้ปัญหา เมื่อผลของเงื่อนไขเป็นเท็จก็ได้ ซึ่งเรียกว่ามีโครงสร้างของการทำงานแบบเลือกทำเพียงทางเดียว หรือแบบ if...then...

5.2.1 รูปแบบคำสั่งโดยใช้ if

สำหรับการเขียนโปรแกรม MATLAB รูปแบบนี้ใช้คำสั่ง if เป็นการตรวจสอบเงื่อนไขพจน์ (Condition) ที่กำหนดหลัง if หากเงื่อนไขเป็นจริงจะกระทำคำสั่งที่ตามมา (Module 1)

```
if      Condition
      Module 1
end
```

ตัวอย่างที่ 5.4 จงเขียนโปรแกรมบน Script File โดยใช้คำสั่ง if มีเงื่อนไขดังนี้ ถ้า $a = 101$ ให้แสดงผลว่า yes ดังภาพที่ 5.4



ภาพที่ 5.4 การใช้คำสั่ง if

เมื่อกดปุ่ม run เพื่อทำการประมวลผล ผลลัพธ์จะแสดงบนหน้าจอ Command Window คือ yes ดังภาพที่ 5.5



ภาพที่ 5.5 การแสดงผลลัพธ์บนหน้าจอ Command Window

ตัวอย่างที่ 5.5 จงเขียนโปรแกรมตรวจสอบคะแนนบน Script File โดยใช้คำสั่ง if กำหนดให้ป้อนคะแนนสอบ และหากคะแนนมากกว่า 5 คะแนน ให้แสดงข้อความว่า “Pass” ดังภาพที่ 5.6



The image shows the MATLAB Editor and Command Window. The Editor displays a script named 'score.m' with the following code:

```
1- Sc = input('Please input Score = ');
2- if Sc>=5
3-     disp('Pass');
4- end
5- disp(Sc)
```

The Command Window shows the execution of the script for three different inputs:

```
>> score
Please input Score = 6
Pass
    6

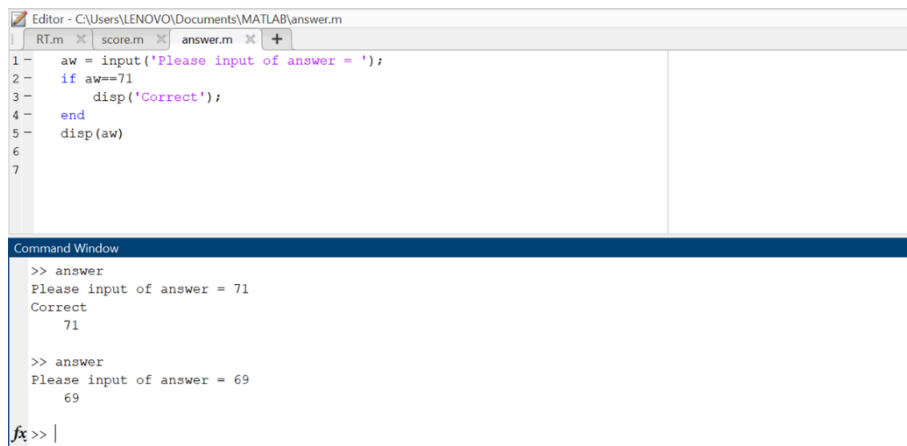
>> score
Please input Score = 3
    3

>> score
Please input Score = 9
Pass
    9

fx >>
```

ภาพที่ 5.6 การเขียนโปรแกรมตรวจสอบคะแนน

ตัวอย่างที่ 5.6 จงเขียนโปรแกรมตรวจสอบคำตอบบน Script File โดยใช้คำสั่ง if กำหนดให้ป้อนคำตอบ หากถูกให้แสดงข้อความว่า “Correct” จากคำถามต่อไปนี้ $29 + 42$ ดังภาพที่ 5.7



The image shows the MATLAB Editor and Command Window. The Editor displays a script named 'answer.m' with the following code:

```
1- aw = input('Please input of answer = ');
2- if aw==71
3-     disp('Correct');
4- end
5- disp(aw)
```

The Command Window shows the execution of the script for two different inputs:

```
>> answer
Please input of answer = 71
Correct
    71

>> answer
Please input of answer = 69
    69

fx >> |
```

ภาพที่ 5.7 การเขียนโปรแกรมตรวจสอบคำตอบ

5.2.2 รูปแบบคำสั่งโดยใช้ if-elseif-else

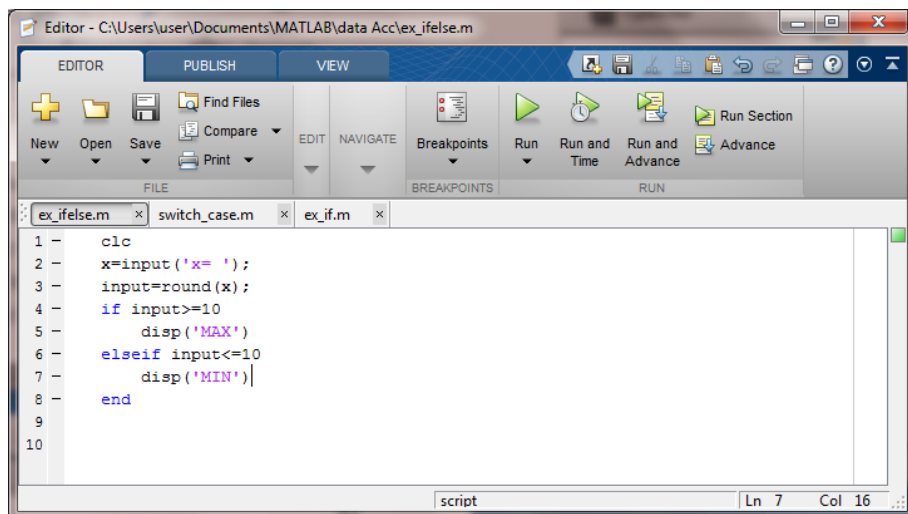
รูปแบบนี้เป็นการตรวจสอบเงื่อนไขในนิพจน์ (Condition) ที่กำหนดหลัง if หากเงื่อนไขเป็นจริงจะกระทำสิ่งที่ตามมา (Module 1) แต่ถ้าเงื่อนไขเป็นเท็จจะกระทำคำสั่งหลัง else (Module 2) โดยมีโครงสร้างของคำสั่ง ดังนี้

```
if    Condition
      Module 1

else
      Module 2

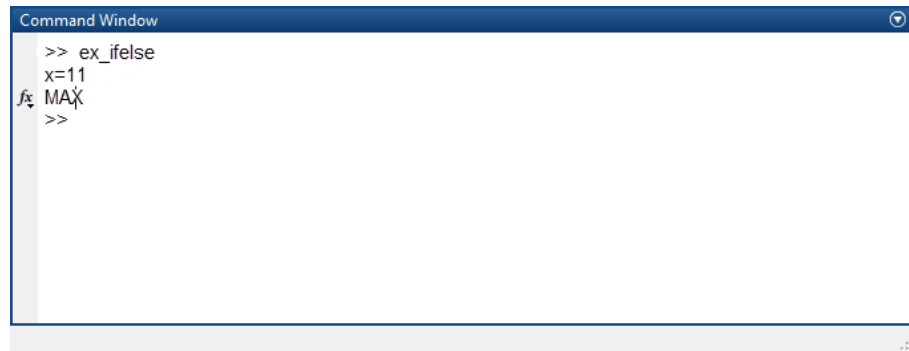
end
```

ตัวอย่างที่ 5.7 จงเขียนโปรแกรมบน Script File โดยใช้คำสั่ง if...else มีเงื่อนไขดังนี้ กำหนดให้ทำการรับค่าตัวเลขแล้วตรวจสอบตัวเลขที่ป้อนเข้าไปว่า มากกว่าหรือเท่ากับ 10 ให้แสดงข้อความทางจอภาพว่า MAX หากน้อยกว่าหรือเท่ากับ 10 ให้แสดงข้อความทางจอภาพว่า MIN ดังภาพที่ 5.8



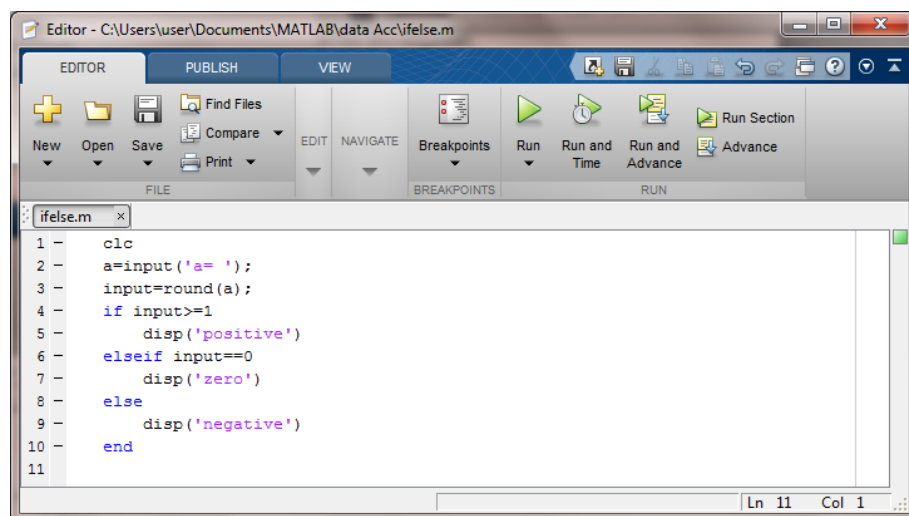
ภาพที่ 5.8 การใช้คำสั่ง if...else

เมื่อกดปุ่ม run เพื่อทำการประมวลผล ผลลัพธ์จะแสดงบนหน้าจอ Command Window ดังภาพที่ 5.9



ภาพที่ 5.9 การแสดงผลรันบนหน้าจอ Command Window คำสั่ง if...else

ตัวอย่างที่ 5.8 จงเขียนโปรแกรมบน Script File โดยใช้คำสั่ง if-elseif-else ดังนี้
รับค่าที่จะตรวจสอบทาง Command Window
ถ้าเป็นบวกแสดงผลที่หน้า Command Window ว่า Positive
ถ้าเป็นลบแสดงผลที่หน้า Command Window ว่า Negative
ถ้าเป็นศูนย์แสดงผลที่หน้า Command Window ว่า Zero ดังภาพที่ 5.10



ภาพที่ 5.10 การใช้คำสั่ง if-elseif-else

เมื่อกดปุ่ม run เพื่อทำการประมวลผล ผลลัพธ์จะแสดงบนหน้าจอ Command Window

คือ

- ถ้ากำหนดค่า a มีค่ามากกว่าหรือเท่ากับ 1 จะแสดงข้อความว่า Positive
- ถ้ากำหนดค่า a มีค่าเท่ากับ 0 จะแสดงข้อความว่า Zero
- ถ้ากำหนดค่า a มีค่าอื่น ๆ จะแสดงข้อความว่า Negative ดังภาพที่ 5.11



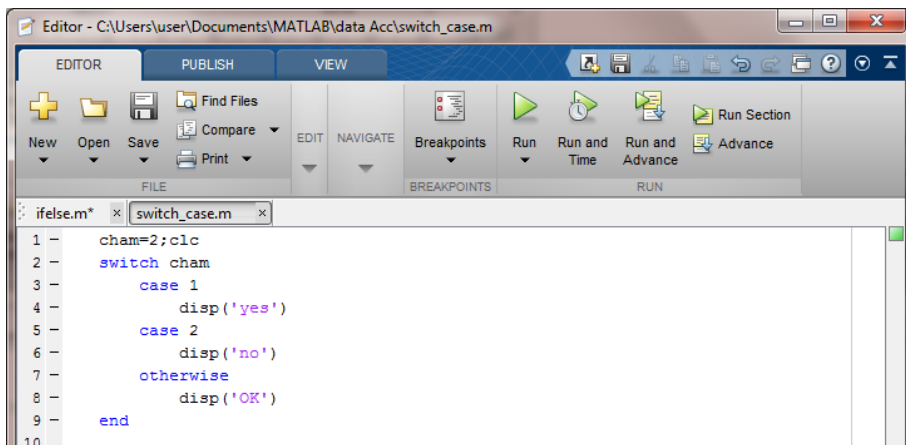
ภาพที่ 5.11 การแสดงผลรันหน้าจอ Command Window คำสั่ง if-elseif-else

5.2.3 รูปแบบคำสั่งโดยใช้ switch-case-otherwise

รูปแบบนี้ทำงานเหมือนกับคำสั่ง if-elseif แต่มีข้อจำกัดที่ไม่สามารถกำหนดกรณีของตัวแปรอื่น นอกจากนั้นกำหนดไว้หลัง switch สามารถเขียนโครงสร้างของคำสั่งได้ ดังนี้

```
switch      variable
case      Condition 1
           Module 1
case      Condition 2
           Module 2
case      Condition 3
           Module 3
end
```

ตัวอย่างที่ 5.9 จงเขียนโปรแกรมบน Script File เพื่อแสดงเงื่อนไขในกรณีถ้า cham=2 ให้แสดงผลว่า no โดยใช้คำสั่ง switch-case-otherwise ดังภาพที่ 5.12



ภาพที่ 5.12 การใช้คำสั่ง switch-case-otherwise

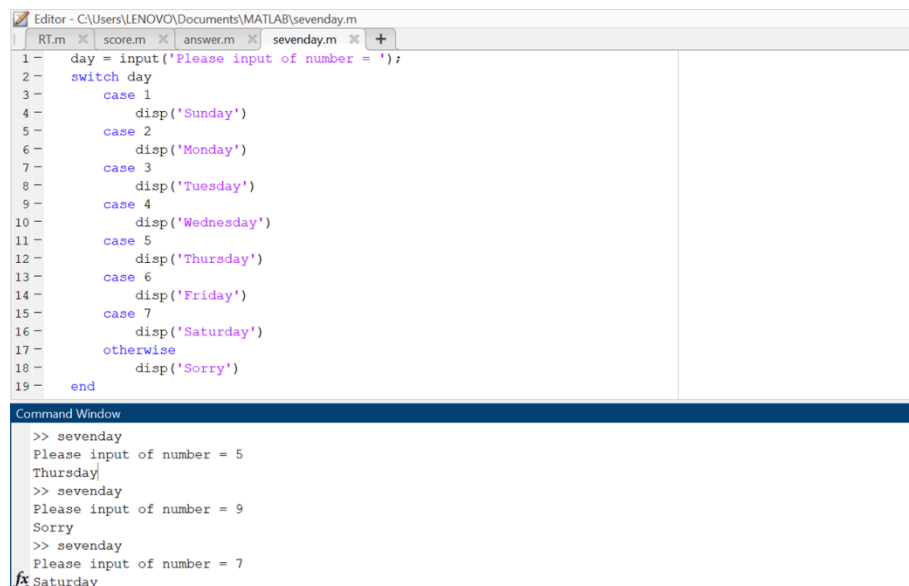
เมื่อกดปุ่ม run เพื่อทำการประมวลผล ผลลัพธ์จะแสดงบนหน้าจอ Command Window ดังภาพที่ 5.13



ภาพที่ 5.13 การแสดงผลลัพธ์บนหน้าจอ Command Window คำสั่ง switch-case-otherwise

ตัวอย่างที่ 5.10 จงเขียนโปรแกรมอ่านตัวเลข 1-7 เป็นวันในภาษาอังกฤษ บน Script File โดยใช้คำสั่ง switch-case-otherwise ซึ่งมีเงื่อนไขดังนี้ (ดังภาพที่ 5.14)

- ถ้าพิมพ์เลข 1 ให้แสดงข้อความว่า "Sunday"
- ถ้าพิมพ์เลข 2 ให้แสดงข้อความว่า "Monday"
- ถ้าพิมพ์เลข 3 ให้แสดงข้อความว่า "Tuesday"
- ถ้าพิมพ์เลข 4 ให้แสดงข้อความว่า "Wednesday"
- ถ้าพิมพ์เลข 5 ให้แสดงข้อความว่า "Thursday"
- ถ้าพิมพ์เลข 6 ให้แสดงข้อความว่า "Friday"
- ถ้าพิมพ์เลข 7 ให้แสดงข้อความว่า "Saturday"
- ถ้าพิมพ์เลขอื่นใดให้แสดงข้อความว่า "Sorry"



ภาพที่ 5.14 การเขียนโปรแกรมอ่านตัวเลข 1-7 เป็นวันในภาษาอังกฤษ

5.3 การทำซ้ำ

การแก้ปัญหาบางอย่างอาจต้องมีการทำงานในบางคำสั่งหรือบางชุดของคำสั่งซ้ำกันมากกว่าหนึ่งรอบขึ้นไป โครงสร้างแบบมีการวนซ้ำนี้จะต้องมีการตัดสินใจร่วมกันอยู่ด้วยเสมอ เพื่อเป็นเงื่อนไขที่จะตัดสินใจว่าเมื่อใดจะวนซ้ำ หรือเมื่อใดจะถึงเวลาหยุดการวนซ้ำ การวนซ้ำแบบที่ต้องตรวจสอบเงื่อนไขที่จะให้วนซ้ำก่อนที่จะทำงานตามชุดคำสั่งในโครงสร้างแบบวนซ้ำ เรียกว่า การวนซ้ำแบบ while ซึ่งสังเกตได้ว่าถ้าเงื่อนไขไม่เป็นจริงตั้งแต่แรก คำสั่งในโครงสร้างแบบวนซ้ำจะไม่ถูกเรียกให้ทำงานเลย แต่การวนซ้ำแบบที่มีการตรวจสอบเงื่อนไขที่จะให้วนซ้ำหลังจากที่ได้ทำงานตามชุดคำสั่งในโครงสร้างแบบวนซ้ำไปรอบหนึ่งแล้ว เรียกว่า การวนซ้ำแบบ until

5.3.1 รูปแบบคำสั่งโดยใช้ for

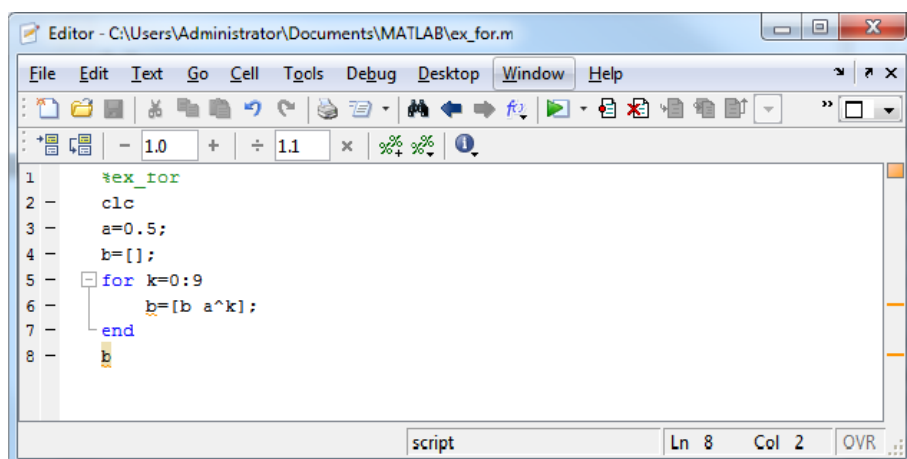
รูปแบบนี้เป็นการตรวจสอบเงื่อนไขในการทำซ้ำตามเวกเตอร์ดัชนีที่กำหนดขึ้น ดังนี้

For k = เริ่มต้น : ค่าที่เพิ่มขึ้น : ค่าสิ้นสุด
m : s : n

การทำซ้ำเป็นการทำตามเงื่อนไขของค่าเริ่มต้นจนถึงค่าสิ้นสุด (m:s:n) ถ้าเงื่อนไขเป็นจริงจะกระทำคำสั่งที่ตามมา (Module) จนกว่าเงื่อนไขจะเป็นเท็จ สามารถเขียนโครงสร้างของคำสั่งได้ ดังนี้

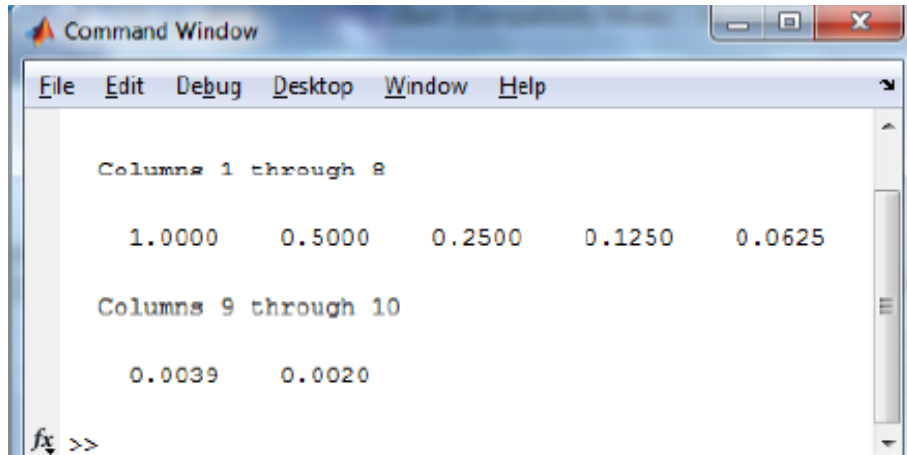
for k=m:s:n
Module
end

ตัวอย่างที่ 5.11 จงเขียนโปรแกรมบน Script File เพื่อหาค่า 0.5^{10} โดยใช้คำสั่ง for ดังภาพที่ 5.15



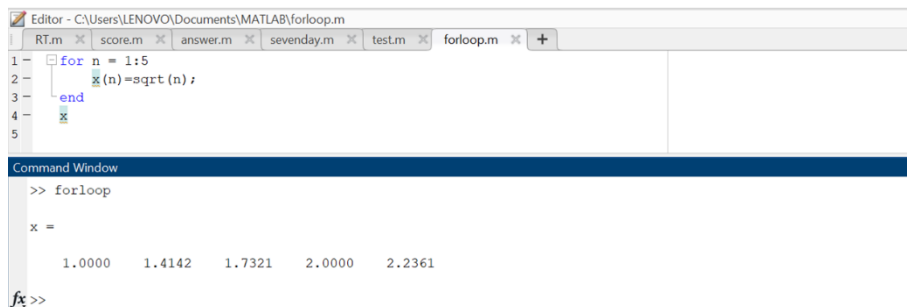
ภาพที่ 5.15 การใช้คำสั่ง for

เมื่อกดปุ่ม run เพื่อทำการประมวลผล ผลลัพธ์จะแสดงบนหน้าจอ Command Window ดังภาพที่ 5.16



ภาพที่ 5.16 การแสดงผลลัพธ์บนหน้าจอ Command Window คำสั่ง for

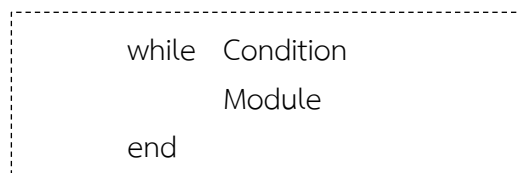
ตัวอย่างที่ 5.12 จงเขียนโปรแกรมโดยใช้คำสั่ง for ในการหารากที่สองของตัวเลข 1 ถึง 5 ดังภาพที่ 5.17



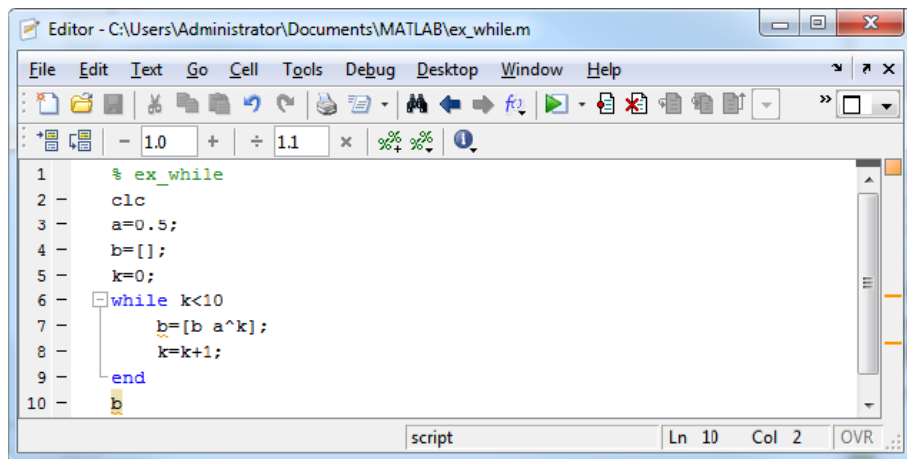
ภาพที่ 5.17 การเขียนโปรแกรมการหารากที่สองของตัวเลข 1 ถึง 5

5.3.2 รูปแบบคำสั่งโดยใช้ while

รูปแบบนี้เป็นการทำซ้ำเพื่อทำชุดคำสั่งภายใน while ถ้าเงื่อนไขเป็นจริงจะกระทำการคำสั่งที่ตามมา (Module) จนกว่าเงื่อนไขจะเป็นเท็จ สามารถเขียนโครงสร้างของคำสั่งได้ ดังนี้

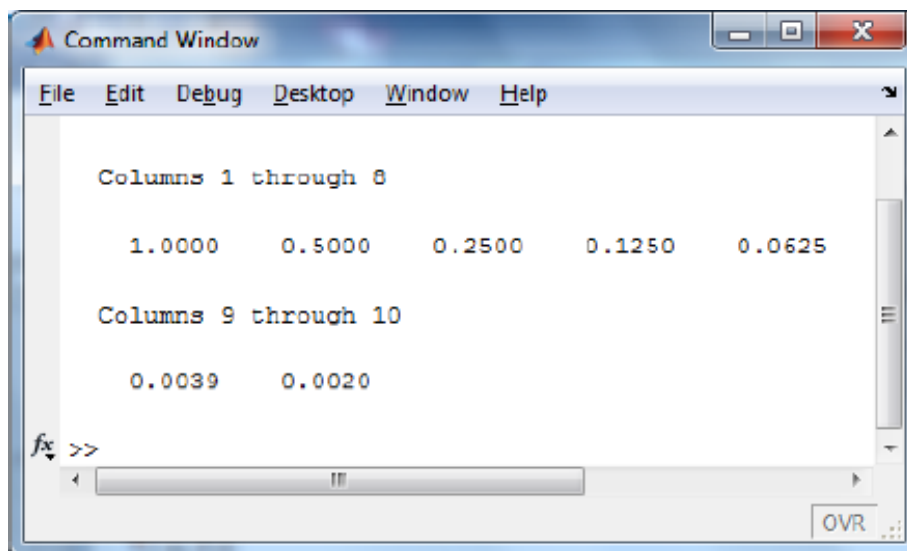


ตัวอย่างที่ 5.13 จงเขียนโปรแกรมบน Script File เพื่อหาค่า 0.5^{10} โดยใช้คำสั่ง while ดังภาพที่ 5.18



ภาพที่ 5.18 การใช้คำสั่ง while

เมื่อกดปุ่ม run เพื่อทำการประมวลผล ผลลัพธ์จะแสดงบนหน้าจอ Command Window ดังภาพที่ 5.19



ภาพที่ 5.19 การแสดงผลบนหน้าจอ Command Window คำสั่ง while

5.4 สรุป

การเขียนโปรแกรม MATLAB แบบมีโครงสร้าง เป็นการกำหนดขั้นตอนการทำงานของโปรแกรมมีโครงสร้างการควบคุมพื้นฐาน 3 โครงสร้าง ได้แก่ การทำงานแบบลำดับ การเลือกกระทำตามเงื่อนไข และการทำซ้ำ

1. การทำงานแบบตามลำดับ คือ การเขียนให้ทำงานจากบนลงล่าง เขียนคำสั่งเป็นบรรทัด และทำทีละบรรทัดจากบรรทัดบนสุดลงไปจนถึงบรรทัดล่างสุด สมมติให้มีการทำงาน 3 กระบวนการ คือ การรับข้อมูลเข้า การคำนวณ และการส่งข้อมูลออก โดยกระบวนการถูกเปลี่ยนให้อยู่รูปของคำสั่งหลายคำสั่งประกอบกันเพื่อให้ทำงานตามขั้นตอนที่ได้ออกแบบไว้

2. การเลือกกระทำตามเงื่อนไข คือ การเขียนโปรแกรมเพื่อนำค่าไปเลือกกระทำ โดยปกติจะมีเหตุการณ์ให้ทำ 2 กระบวนการ คือ เงื่อนไขเป็นจริงจะกระทำกระบวนการหนึ่ง และเป็นเท็จจะกระทำอีกกระบวนการหนึ่ง แต่ถ้าซับซ้อนมากขึ้น จะต้องใช้เงื่อนไขหลายชั้น เช่น การตัดเกรดนักศึกษา เป็นต้น ซึ่งโครงสร้างการทำงานอยู่ในรูปแบบคำสั่งของ if, if-elseif-else และ switch-case-otherwise

3. การทำซ้ำ คือ การทำกระบวนการหนึ่งหลายครั้ง โดยมีเงื่อนไขในการควบคุม หมายถึงการทำซ้ำเป็นหลักที่ทำให้ความเข้าใจได้ยากกว่า 2 รูปแบบแรก เพราะการเขียนโปรแกรมแต่ละภาษา จะไม่แสดงภาพอย่างชัดเจนเหมือนการเขียนผังงาน ผู้เขียนโปรแกรมต้องจินตนาการ ถึงรูปแบบการทำงาน ซึ่งโครงสร้างการทำงานอยู่ในรูปแบบคำสั่งของ for, while

แบบฝึกหัด

บทที่ 5 การเขียนโปรแกรม MATLAB แบบโครงสร้าง

- จงอธิบายการทำงานแบบตามลำดับในการเขียนโปรแกรมแบบมีโครงสร้าง
- กำหนดให้ตัวแปร $t = 5:0.2:5$ จงเขียน Script File เพื่อคำนวณค่า $z = \frac{\cos(t)}{t}$ โดยให้แสดงผลลัพธ์ของค่า z ออกหน้าจอ
- จงอธิบายการเลือกกระทำตามเงื่อนไขในการเขียนโปรแกรมแบบมีโครงสร้าง
- จงเขียนโปรแกรมบน Script File โดยใช้คำสั่ง if มีเงื่อนไขดังนี้ ถ้า $x = 5$ ให้แสดงผลว่า yes
- จงเขียนโปรแกรมบน Script File โดยใช้คำสั่ง if...else มีเงื่อนไขดังนี้ การตัดสินผลการเรียนโรงเรียนแห่งหนึ่ง มีการตัดสินผลการเรียนเป็นผ่าน (Pass) และไม่ผ่าน (Fail) โดยมีเงื่อนไขตัดสินดังนี้ หากได้คะแนนสอบมากกว่าหรือเท่ากับ 50 คะแนน ให้แสดงข้อความทางจอภาพว่า PASS แต่ถ้าไม่อยู่ในเงื่อนไขให้แสดงข้อความ FAIL
- จงเขียนโปรแกรมตรวจสอบเลขคู่เลขคี่ โดยใช้คำสั่ง if-else โดยกำหนดเงื่อนไขดังนี้
 - มีการรับข้อมูลเข้าทางแป้นพิมพ์
 - แสดงผลลัพธ์บนหน้าจอ Command Window
- จงเขียนโปรแกรมบน Script File โดยใช้คำสั่ง if-elseif-else ดังนี้
 - มีการรับข้อมูลเข้าทางแป้นพิมพ์
 - ถ้าค่ามากกว่าหรือเท่ากับ 1 แสดงผลที่หน้าจอ Command Window ว่า Positive
 - ถ้าค่าน้อยกว่า 0 แสดงผลที่หน้าจอ Command Window ว่า Negative
 - ถ้าเป็นศูนย์แสดงผลที่หน้าจอ Command Window ว่า Zero
- จงเขียนโปรแกรมตัดเกรด 5 ระดับ ตามช่วงคะแนน ดังนี้

ช่วงคะแนน	เกรด
≥ 80	4.00
≥ 71 และ < 80	3.00
≥ 61 และ < 70	2.00
≥ 51 และ < 60	1.00
≤ 50	0.00

โดยกำหนดเงื่อนไขดังนี้

1. มีการรับข้อมูลเข้าทางแป้นพิมพ์
 2. แสดงผลลัพธ์บนหน้าจอ Command Window ตามช่วงคะแนน
9. จงเขียนโปรแกรมบน Script File เพื่อแสดงเงื่อนไขในกรณีถ้า $x=1$ ให้แสดงผลว่า True โดยใช้คำสั่ง switch-case-otherwise
10. จงเขียนโปรแกรมแปลงตัวเลขโดด 0 - 3 ที่ผู้ใช้ป้อนเข้ามา ให้เป็นตัวหนังสือ Number to word โดยใช้คำสั่ง switch-case-otherwise
11. จงอธิบายการทำซ้ำในการเขียนโปรแกรมแบบมีโครงสร้าง
12. จงเขียนโปรแกรมแสดงเลขจาก 1 ถึง 10 ทางจอภาพ โดยใช้คำสั่ง for
13. จงเขียนโปรแกรมบน Script File เพื่อหาค่า 2^{10} โดยใช้คำสั่ง for
14. จงเขียนโปรแกรมแสดงผลรวมของจำนวนจาก 1 ถึง N ทางจอภาพ โดยใช้คำสั่ง while
15. จงเขียนโปรแกรมบน Script File เพื่อหาค่า 2^{10} โดยใช้คำสั่ง while

บทที่ 6

การเขียนโปรแกรม MATLAB วาดกราฟ 2 และ 3 มิติ

โปรแกรม MATLAB เป็นโปรแกรมที่มีความสามารถในการวาดกราฟลักษณะต่าง ๆ ได้อย่าง
อย่างดี ภายใต้การเขียนฟังก์ชันการทำงานที่ตอบสนองความต้องการของผู้เขียนโปรแกรมได้ครบ
สมบูรณ์มากที่สุด ซึ่งการวาดกราฟที่สร้างจากโปรแกรม MATLAB มีการใช้งานง่ายและไม่ซับซ้อน
เหมือนกันเขียนโปรแกรมโดยทั่ว ๆ ไป การวาดกราฟของโปรแกรมแบ่งเป็น 4 ประเภท ได้แก่ 1) งาน
กราฟฟิกแบบ 2 มิติ 2) งานกราฟฟิกแบบ 3 มิติ 3) งาน GUI และ 4) งานเคลื่อนไหว แต่ละประเภทมี
การทำงานที่แตกต่างกันออกไป ซึ่งในบทนี้ขออธิบายงานกราฟฟิกทางคณิตศาสตร์ ซึ่งสามารถวาด
กราฟได้สองแบบ คือ แบบ 2 มิติ และแบบ 3 มิติ โดยการกำหนดค่าพิกัดหรือการกำหนดสมการ
มีรูปแบบกราฟให้เลือกมากมาย เช่น กราฟแท่ง กราฟเส้น กราฟวงกลม หรืออื่น ๆ อีกมากมาย

6.1 ฟังก์ชันการวาดกราฟ 2 และ 3 มิติ

6.1.1 ฟังก์ชันการวาดกราฟ 2 มิติ

การวาดกราฟเส้น 2 มิติ ของโปรแกรม MATLAB เป็นการสร้างจากเส้นแกน 2 เส้น คือ แกน
นอน (แกน y) และแกนตั้ง (แกน x) โดยโปรแกรม MATLAB มีฟังก์ชันที่ใช้สำหรับพล็อตกราฟเส้นใน
แกน x-y ได้หลายวิธีด้วยการกำหนดลักษณะของเส้น สัญลักษณ์และสี ในการแสดงผลที่ต่าง
กัน ซึ่งมีโครงสร้างดังตารางที่ 6.1 และตารางที่ 6.2

ตารางที่ 6.1 ฟังก์ชันรูปแบบการวาดกราฟ 2 มิติ

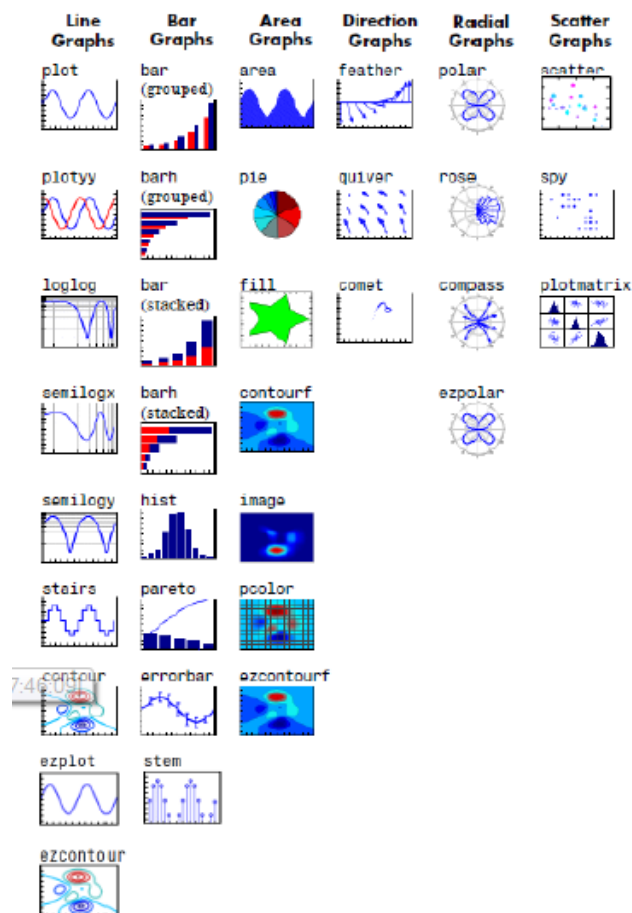
ฟังก์ชัน	รายละเอียด
plot(x,y)	พล็อตกราฟของ x-y โดยโปรแกรม MATLAB จะพล็อตเป็นเส้นทึบ สีน้ำเงิน เมื่อไม่ได้กำหนดสี รูปแบบ และสัญลักษณ์ที่จะทำการพล็อต
plot(x,y,x,z)	พล็อตกราฟของ x-y ก่อนแล้วจึงพล็อตกราฟของ x,z โดยที่กราฟทั้งสอง เส้นอยู่บนแกนเดียวกันแต่แทนด้วยสีที่ต่างกัน เพื่อแยกความแตกต่าง
subplot(m,n,p)	แบ่งหน้าต่างรูปภาพเป็นหน้าต่างย่อยด้วย m แถว n หลัก และลำดับ การพล็อตคำสั่งด้วยหน้าต่างย่อยที่ p
figure(n)	สร้างหน้าต่างรูปภาพหมายเลข n
clf	ลบรูปภาพต่าง ๆ บนหน้าต่างรูปภาพ
close(n)	ลบหน้าต่างรูปภาพหมายเลข n
close(all)	ลบหน้าต่างรูปภาพทุกหน้าต่าง

ตารางที่ 6.2 รูปแบบ สัญลักษณ์ และสีที่ใช้ในการวาดกราฟ

สัญลักษณ์	สี	สัญลักษณ์	เครื่องหมาย	สัญลักษณ์	รูปแบบเส้น
r	red	.	จุด	:	เส้นจุด
g	Green	o	วงกลม	-	เส้นทึบ
b	Blue	+	บวก	-.	เส้นประและจุด
c	Cyan	x	กากบาท	- -	เส้นประ
m	Magenta	*	ดอกจัน		
y	Yellow	s	สี่เหลี่ยมจัตุรัส		
k	Black	d	รูปข้ามหลามตัด		
w	white	^	สามเหลี่ยมบน		
		>	สามเหลี่ยมขวา		

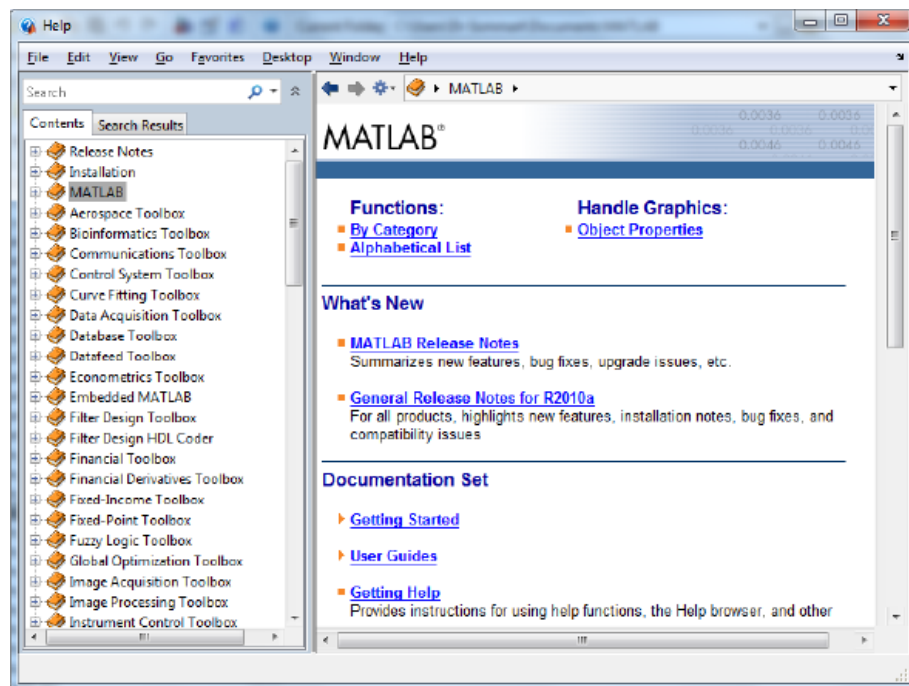
6.1.2 การวาดกราฟ 2 มิติ ลักษณะต่าง ๆ

นอกจากกราฟเส้นแล้ว โปรแกรม MATLAB สามารถสร้างกราฟ 2 มิติในลักษณะต่าง ๆ ได้ ดังภาพที่ 6.1

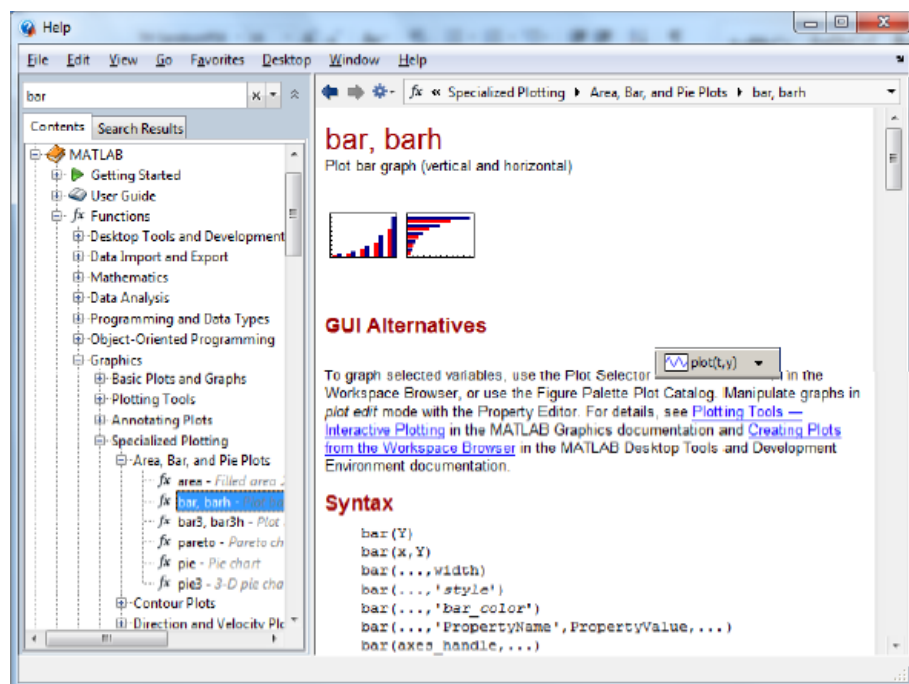


ภาพที่ 6.1 กราฟ 2 มิติลักษณะต่าง ๆ

จากภาพที่ 6.1 การใช้งานของกราฟ 2 มิติ สามารถดูได้โดยการคลิกปุ่ม Help>>Product Help ปรากฏหน้าต่าง Help ดังภาพที่ 6.2 หลังจากนั้นป้อนชื่อกราฟที่ต้องการสร้างลงในช่อง Search แล้วทำการค้นหา เช่น ป้อนชื่อกราฟว่า bar ลงในช่อง Search แล้วกด Enter ปรากฏหน้าต่างดังภาพที่ 6.3 ซึ่งสามารถศึกษารายละเอียดวิธีการสร้างกราฟลักษณะต่าง ๆ ได้



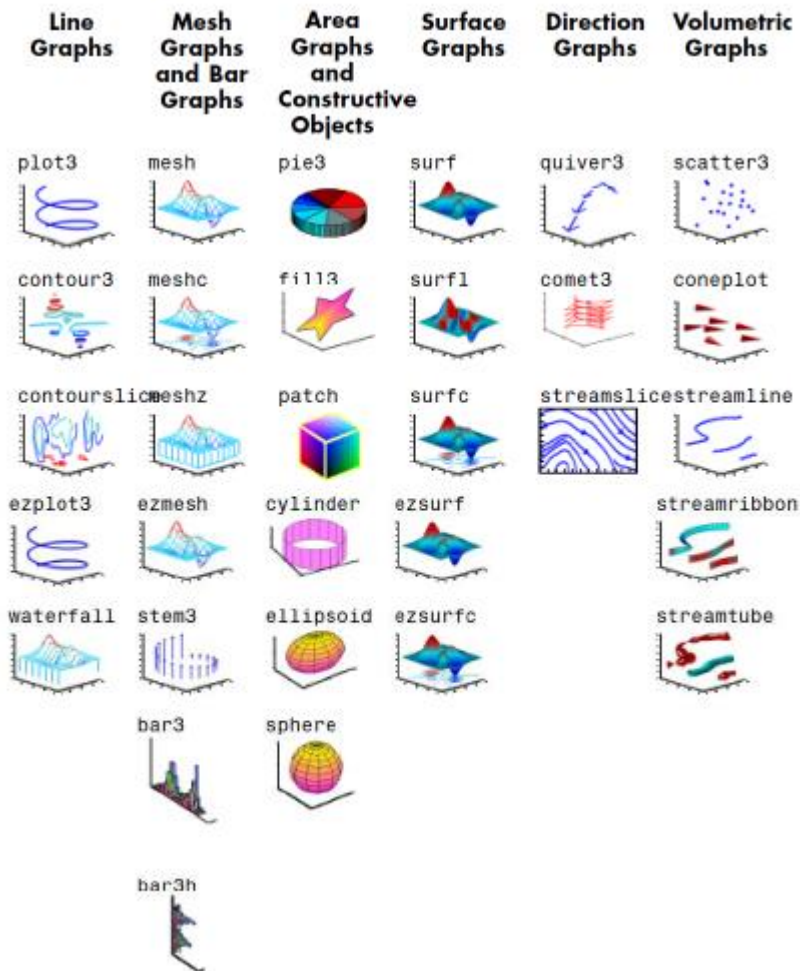
ภาพที่ 6.2 หน้าต่างความช่วยเหลือในการค้นหาคำสั่งการวาดกราฟ



ภาพที่ 6.3 ผลการค้นหาข้อมูลในหน้าต่าง Help ของการวาดกราฟ 2 มิติ

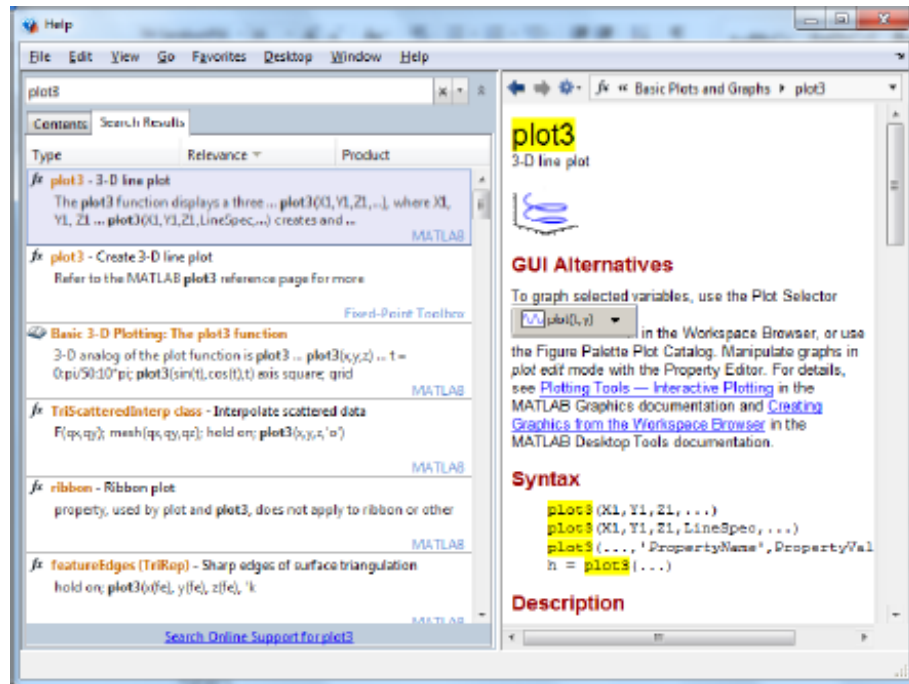
6.1.3 การวาดกราฟ 3 มิติ ลักษณะต่าง ๆ

การวาดกราฟ 3 มิติ โดยใช้โปรแกรม MATLAB สามารถใช้คำสั่งต่าง ๆ สำหรับการวาดกราฟ ดังภาพที่ 6.4



ภาพที่ 6.4 กราฟ 3 มิติลักษณะต่าง ๆ

จากภาพที่ 6.4 การใช้งานของกราฟ 3 มิติ สามารถดูได้โดยการคลิกปุ่ม Help>>Product Help ปรากฏหน้าต่าง Help ดังภาพที่ 6.2 หลังจากนั้นป้อนชื่อกราฟที่ต้องการสร้างลงในช่อง Search แล้วทำการค้นหา เช่น ป้อนชื่อกราฟว่า plot3 ลงในช่อง Search แล้วกด Enter ปรากฏหน้าต่างดังภาพที่ 6.5 ซึ่งสามารถศึกษารายละเอียดวิธีการสร้างกราฟลักษณะต่าง ๆ ได้



ภาพที่ 6.5 ผลการค้นหาข้อมูลในหน้าต่าง Help ของการวาดกราฟ 3 มิติ

6.2 การเขียนโปรแกรมและวาดกราฟ 2 มิติ

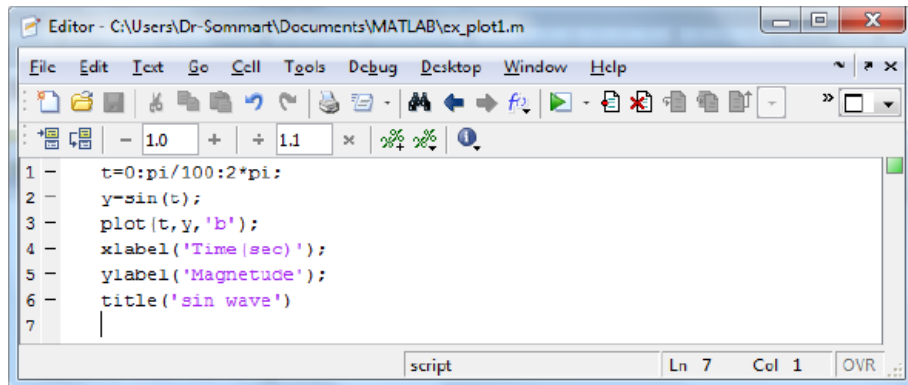
6.2.1 การวาดกราฟเส้นโดยใช้คำสั่ง plot

จากคำสั่ง Plot(x,y) สามารถกำหนดรูปแบบเส้น สี สัญลักษณ์ ชื่อภาพ แกน ที่กระทำวาดกราฟได้

ตัวอย่างที่ 6.1 จงวาดกราฟของฟังก์ชัน sine โดยกำหนดให้กราฟมีสีน้ำเงินที่เวลา (t) 0 ถึง 2π
วิธีทำ

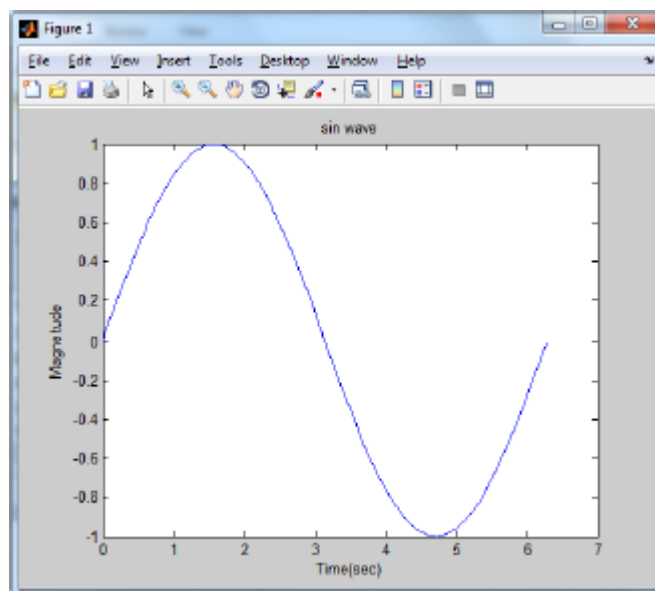
1. เปิดหน้าต่าง Editor ของ Script File แล้วกำหนดค่าเวลา $t = 0 : 2\pi$
2. สร้างสมการของฟังก์ชัน sine ดังนี้ $y = \sin(t)$
3. วาดกราฟด้วยฟังก์ชัน plot พร้อมทั้งกำหนดสีของเส้นกราฟเป็นสีน้ำเงิน
4. กำหนดชื่อแกน x เป็น Time (sec)
5. กำหนดชื่อแกน y เป็น Magnitude
6. กำหนดชื่อกราฟ เป็น sine wave

ดังภาพที่ 6.6



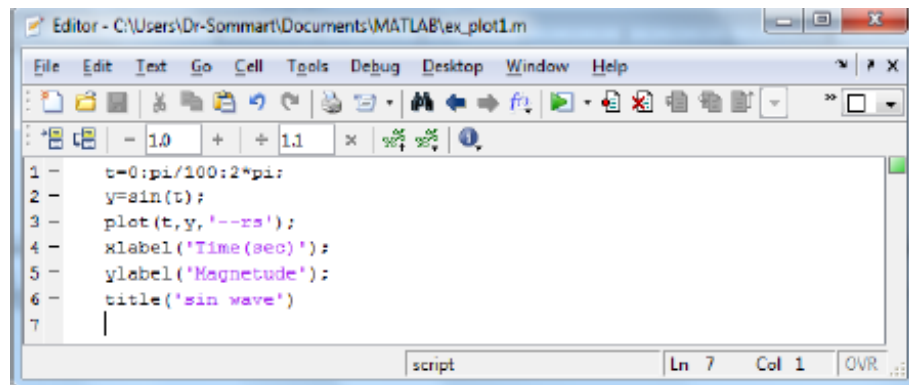
ภาพที่ 6.6 การเขียน Script File สำหรับวาดกราฟฟังก์ชัน sine

จากนั้นทำการทดสอบโดยการคลิกปุ่ม run ปรากฏหน้าต่าง Figure 1 ดังภาพที่ 6.7



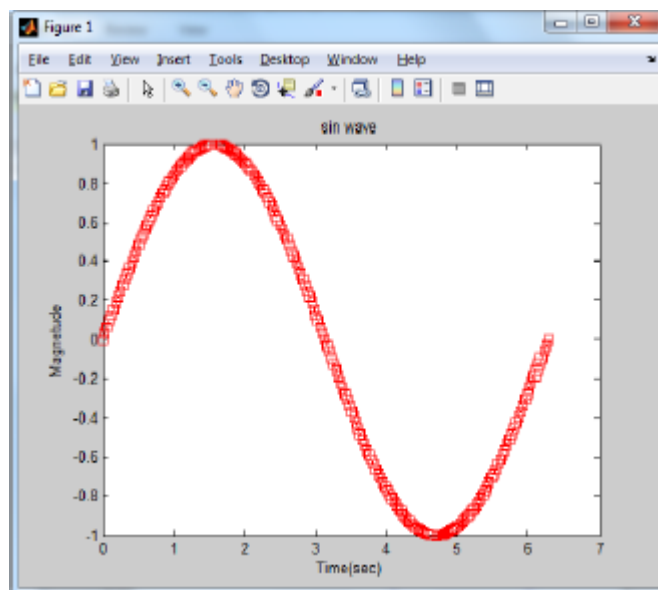
ภาพที่ 6.7 ผลลัพธ์ของการวาดกราฟฟังก์ชัน sine

จากภาพที่ 6.7 หากต้องการเปลี่ยนแปลงรูปแบบเส้นได้ เช่น ให้เป็นเส้นประสีแดง มีเครื่องหมายสี่เหลี่ยมเพื่อแสดงตำแหน่งข้อมูลแต่ละจุด (ซึ่งให้แก้ไขคำสั่งเฉพาะในบรรทัดที่ 3) ดังภาพที่ 6.8



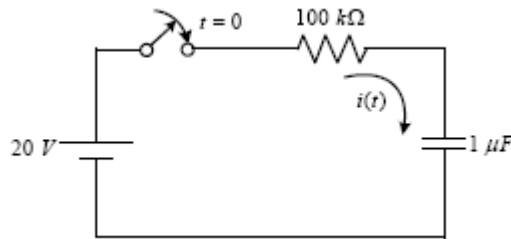
ภาพที่ 6.8 การเขียน Script File สำหรับวาดกราฟฟังก์ชัน sine ให้เป็นเส้นประสีแดง และมีเครื่องหมายสี่เหลี่ยม

จากนั้นทำการทดสอบโดยการคลิกปุ่ม run ปรากฏหน้าต่าง Figure 1 ดังภาพที่ 6.9



ภาพที่ 6.9 ผลลัพธ์ของการวาดกราฟฟังก์ชัน sine จากการแก้ไขโปรแกรม

ตัวอย่างที่ 6.2 จากวงจร RC อนุกรม ดังภาพที่ 6.10 จงวาดกราฟของฟังก์ชันกระแส $i(t)$ sine โดยที่กราฟจะมีสีน้ำเงินที่เวลา (t) 0 ถึง 0.5 วินาที

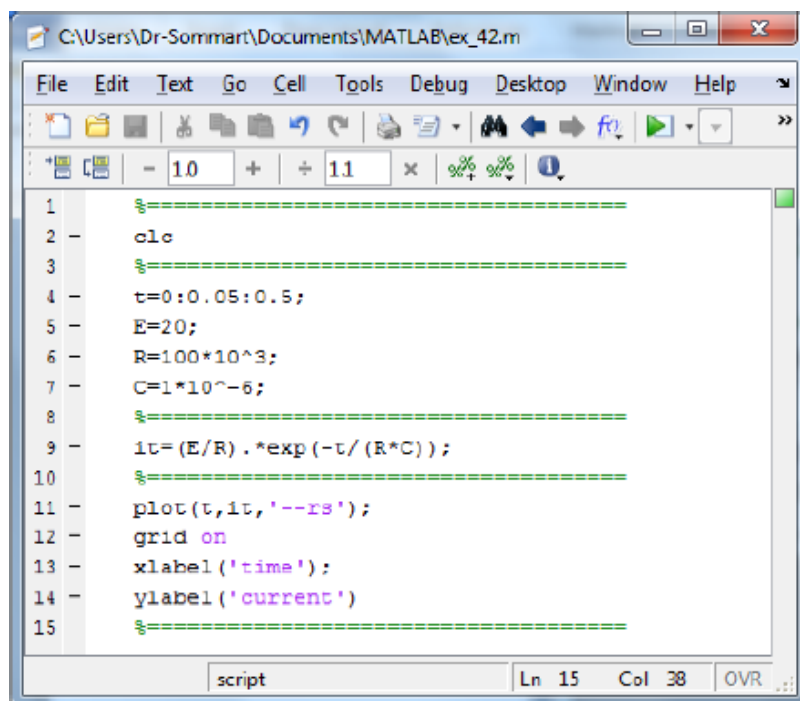


ภาพที่ 6.10 วงจร RC อนุกรม

วิธีทำ

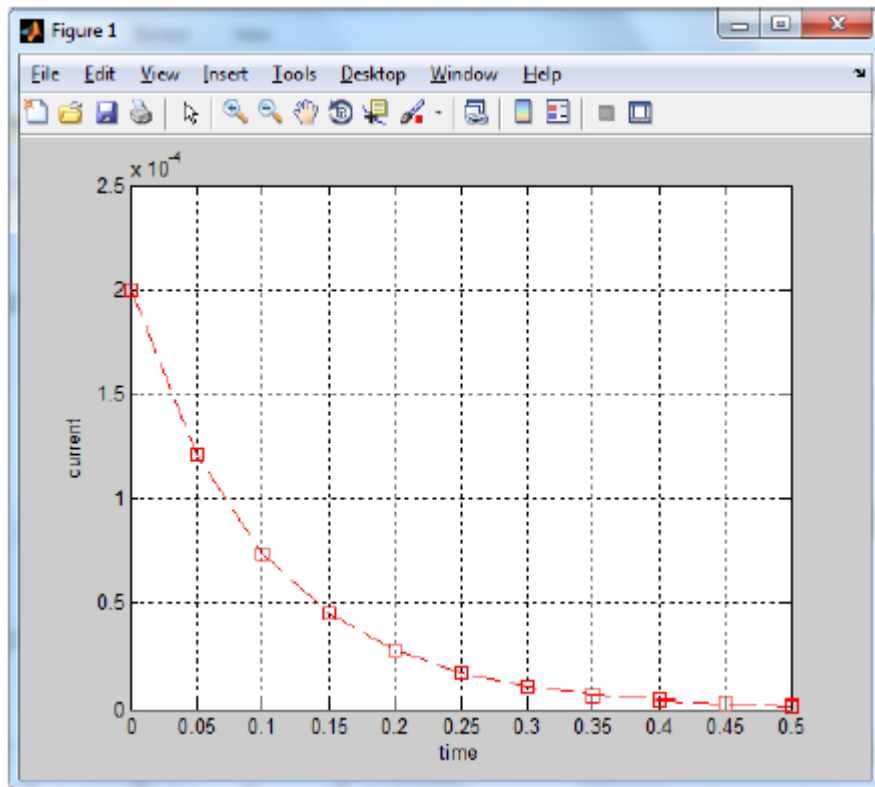
1. เปิดหน้าต่าง Editor ของ Script File แล้วกำหนดค่าเวลา $t = 0:0.05:0.5$
2. กำหนดค่า $E = 20 \text{ V}$, $R = 100 \text{ k}\Omega$, $C = 1 \text{ }\mu\text{F}$
3. สร้างสมการของฟังก์ชัน $i(t) = \frac{E}{R} e^{-t/RC}$
4. วาดกราฟด้วยฟังก์ชัน plot พร้อมทั้งกำหนดสีของเส้นกราฟเป็นสีน้ำเงิน และกำหนดกริด
5. กำหนดชื่อแกน x เป็น Time
6. กำหนดชื่อแกน y เป็น Amp

ดังภาพที่ 6.11



ภาพที่ 6.11 การเขียน Script File สำหรับวาดกราฟกระแสของวงจร RC อนุกรม

จากนั้นทำการทดสอบโดยการคลิกปุ่ม run ปรากฏหน้าต่าง Figure 1 ดังภาพที่ 6.12



ภาพที่ 6.12 ผลลัพธ์ของการวาดกราฟกระแสของวงจร RC อนุกรม

6.1.2 การวาดกราฟเส้นโดยใช้คำสั่ง subplot

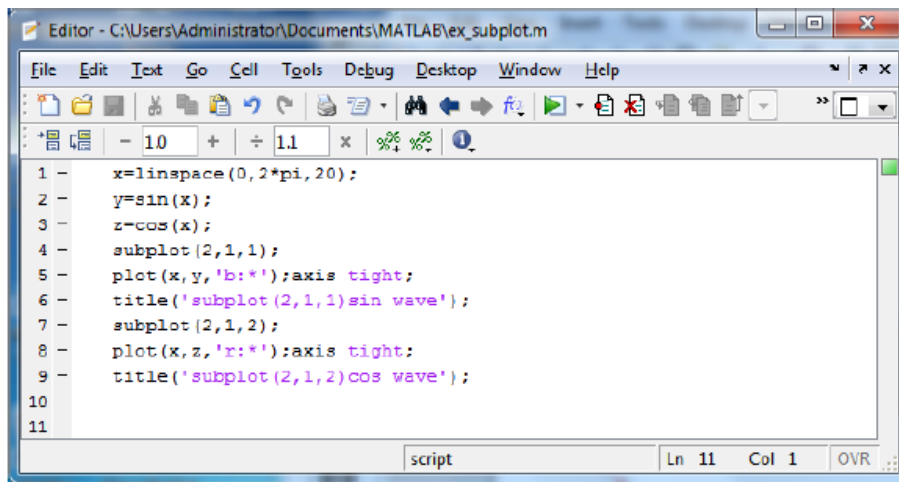
คำสั่ง subplot เป็นคำสั่งที่สามารถแบ่งหน้าต่างภาพเป็นหน้าต่างย่อย ๆ

ตัวอย่างที่ 6.3 จงวาดกราฟของฟังก์ชัน sine เทียบกับฟังก์ชัน cosine โดยให้แสดงรูป 2 รูป และเส้นเป็นเส้นประมี * บนเส้น ฟังก์ชัน sine เส้นสีน้ำเงิน และฟังก์ชัน cosine เส้นสีแดง

วิธีทำ

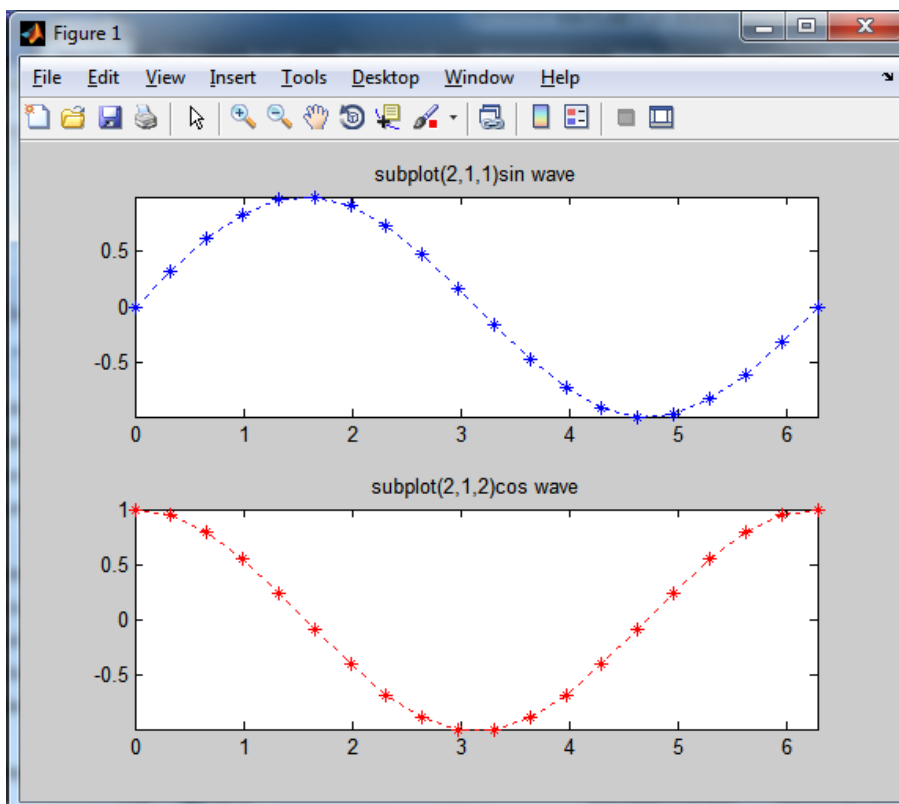
1. เปิดหน้าต่าง Editor ของ Script File แล้วกำหนดค่า x
2. สร้างสมการของฟังก์ชัน sine
3. สร้างสมการของฟังก์ชัน cosine
4. วาดกราฟด้วยคำสั่ง subplot พร้อมทั้งกำหนดฟังก์ชัน sine เป็นกราฟเส้นสีน้ำเงิน และฟังก์ชัน cosine เป็นกราฟเส้นสีแดง
5. กำหนดชื่อกราฟแต่ละฟังก์ชันดังนี้
ฟังก์ชัน sine กำหนดชื่อเป็น subplot(2,1,1)sin wave
ฟังก์ชัน cosine กำหนดชื่อเป็น subplot(2,1,)cos wave

ดังภาพที่ 6.13



ภาพที่ 6.13 การเขียน Script File สำหรับวาดกราฟฟังก์ชัน sine เทียบกับฟังก์ชัน cosine

จากนั้นทำการทดสอบโดยการคลิกปุ่ม run ปรากฏหน้าต่าง Figure 1 ดังภาพที่ 6.14



ภาพที่ 6.14 ผลลัพธ์ของการวาดกราฟฟังก์ชัน sine เทียบกับฟังก์ชัน cosine

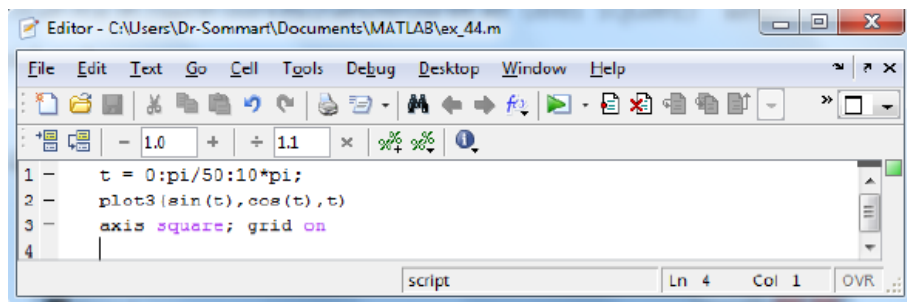
6.3 การเขียนโปรแกรมและวาดกราฟ 3 มิติ

6.3.1 การวาดกราฟเส้น 3 มิติ

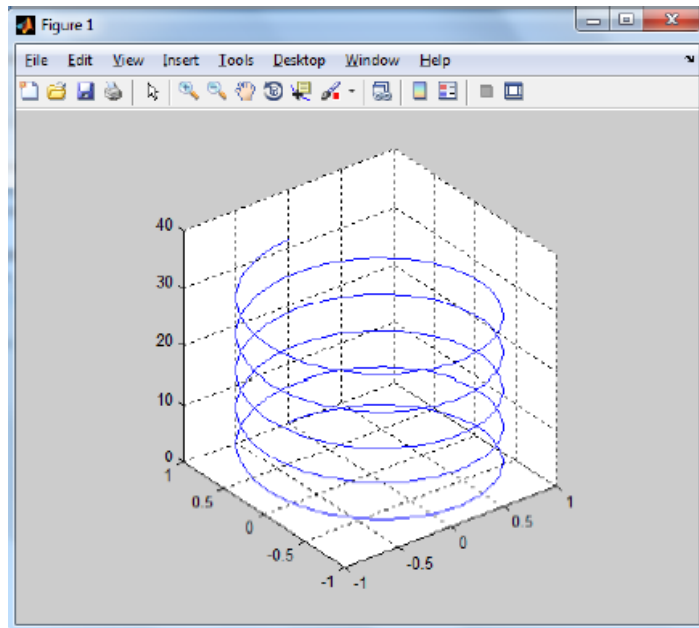
การวาดกราฟเส้น 3 มิติ สามารถทำได้เช่นเดียวกับการวาดกราฟเส้น 2 มิติ โดยที่การวาดกราฟ 3 มิติ ใช้ฟังก์ชัน `plot3`, `contour3`, `contourslice`, `ezplote3` และ `waterfall` เช่น `plot3(x,y,z)` ถ้าค่า x, y และ z คือ เวกเตอร์ที่มีความยาวเดียวกัน การสร้างเส้นในแบบ 3 มิติ ผ่านจุดที่มีพิกัดที่มีองค์ประกอบของ x, y และ z

ตัวอย่างที่ 6.4 จงวาดกราฟ 3 มิติ แบบเกลียว ที่เวลาเริ่มต้นจาก 0 ถึง 10π โดยใช้ฟังก์ชัน `plot3` วิธีทำ

1. เปิดหน้าต่าง Editor ของ Script File แล้วกำหนดค่า $t = 0:\pi/50:10*\pi$
2. วาดกราฟด้วยคำสั่ง `plot3(sin(t),cos(t),t)`
3. กำหนดหน้าต่างแสดงผลให้เป็นสี่เหลี่ยมจัตุรัส (Axis Square) และกำหนดเส้นกริดในตาราง (Grid On) ดังภาพที่ 6.15
4. เมื่อกดปุ่ม run แสดงผลลัพธ์ดังภาพที่ 6.16



ภาพที่ 6.15 การเขียน Script File สำหรับวาดกราฟ 3 มิติ แบบเกลียว โดยใช้ฟังก์ชัน `plot3`



ภาพที่ 6.16 ผลลัพธ์ของการวาดเกลียวโดยใช้ฟังก์ชัน plot3

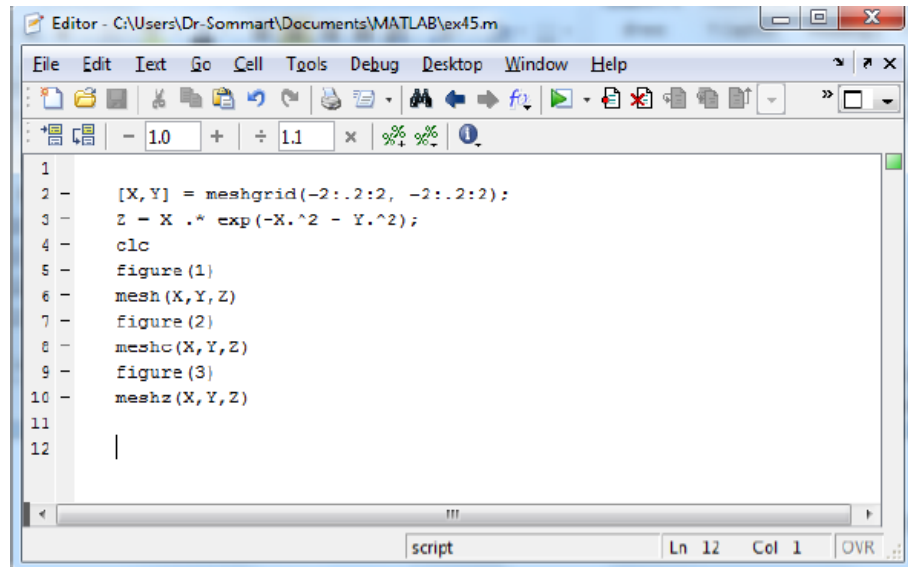
6.3.2 การวาดกราฟ 3 มิติ แบบตาข่ายและบาร์

การวาดกราฟ 3 มิติ แบบตาข่ายและบาร์จะใช้ฟังก์ชัน mesh, meshc, meshz, ezmesh, stem3, bar3 และ bar3h

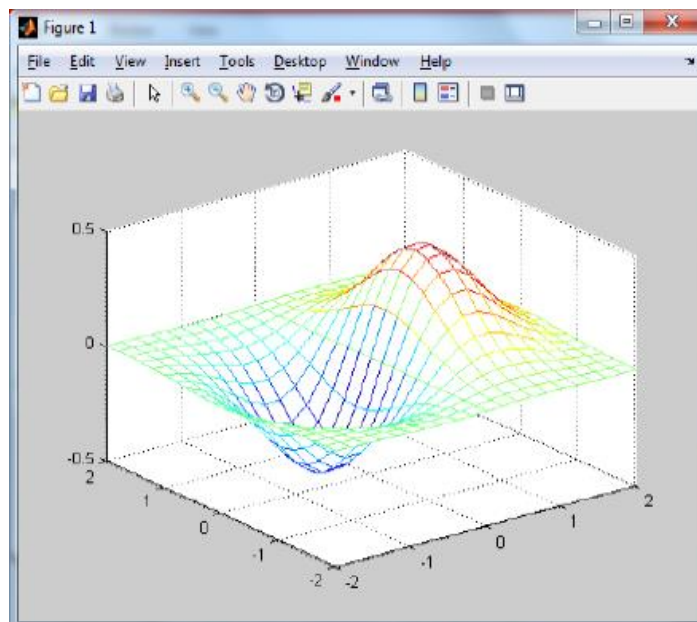
ตัวอย่างที่ 6.5 จงใช้ฟังก์ชัน mesh, meshc และ meshz สำหรับวาดกราฟ 3 มิติ โดยกำหนดให้ $[X,Y] = \text{meshgrid}(1:3,10:14)$ และ $Z = X \cdot \exp(-X.^2 - Y.^2)$

วิธีทำ

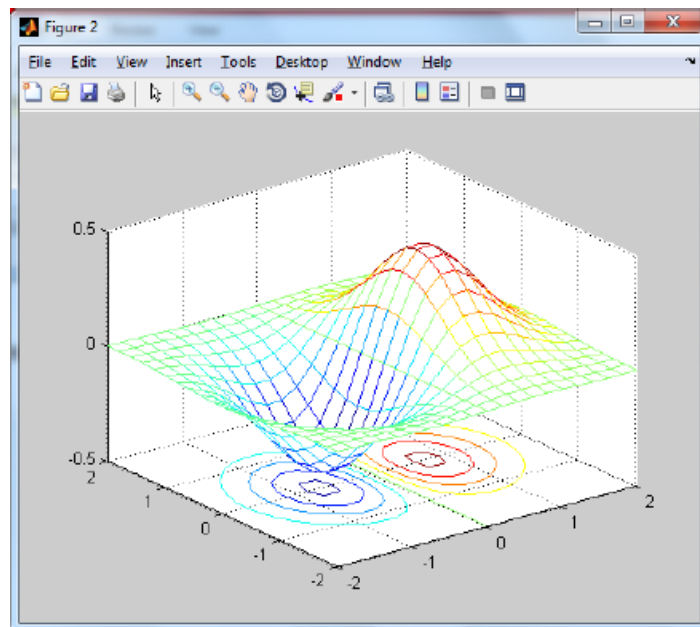
1. เปิดหน้าต่าง Editor ของ Script File แล้วกำหนดค่า $[X,Y] = \text{meshgrid}(1:3,10:14)$
2. กำหนดค่า $Z = X \cdot \exp(-X.^2 - Y.^2)$
3. วาดกราฟด้วยคำสั่ง mesh, meshc และ meshz ตามลำดับ ดังภาพที่ 6.17
4. เมื่อกดปุ่ม run แสดงผลลัพธ์ดังภาพที่ 6.18 – 6.20 ตามลำดับ



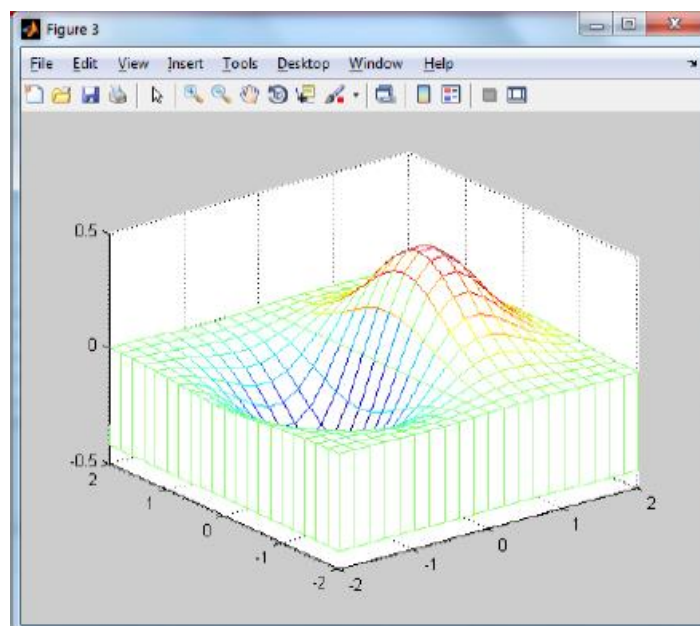
ภาพที่ 6.17 การเขียน Script File สำหรับวาดกราฟ 3 มิติ
โดยใช้คำสั่ง mesh, meshc และ meshz



ภาพที่ 6.18 ผลลัพธ์ของการใช้ฟังก์ชัน mesh



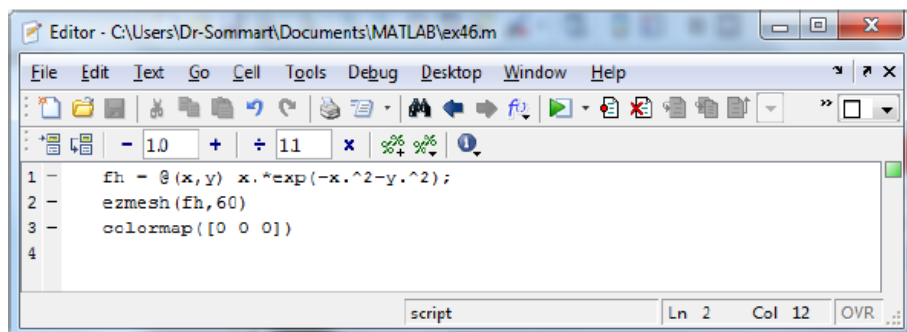
ภาพที่ 6.19 ผลลัพธ์ของการใช้ฟังก์ชัน `meshc`



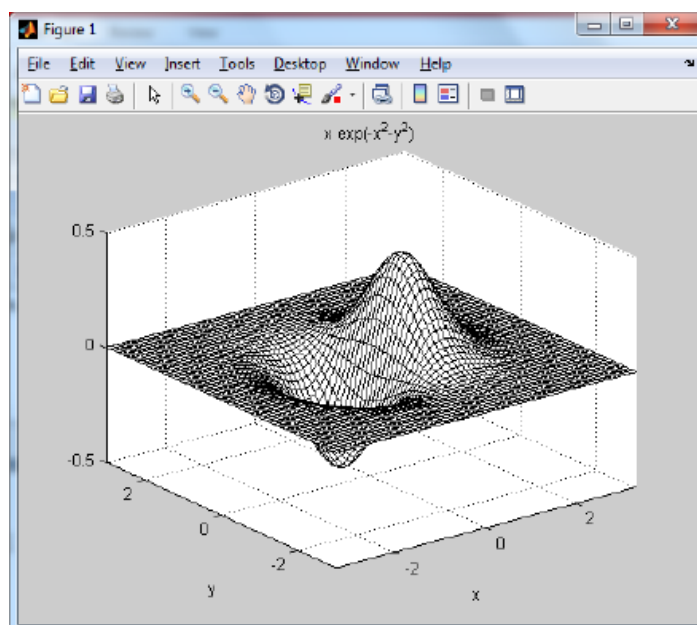
ภาพที่ 6.20 ผลลัพธ์ของการใช้ฟังก์ชัน `meshz`

ตัวอย่างที่ 6.6 จงใช้ฟังก์ชัน ezmesh สำหรับวาดกราฟ 3 มิติของ $f(x,y) = ex^{-x^2-y^2}$ วิธีทำ

1. เปิดหน้าต่าง Editor ของ Script File
2. กำหนดค่า fh = @(x,y)x.*exp(-x.^2 - y.^2)
3. ใช้ฟังก์ชัน ezmesh โดยใช้ความละเอียดเท่ากับ 60
4. กำหนดสีของกราฟให้เป็นสีดำ colormap([1 1 1]) ดังภาพที่ 6.21
5. เมื่อกดปุ่ม run แสดงผลลัพธ์ดังภาพที่ 6.22



ภาพที่ 6.21 การเขียน Script File สำหรับวาดกราฟ 3 มิติ โดยใช้คำสั่ง ezmesh



ภาพที่ 6.22 ผลลัพธ์ของการใช้ฟังก์ชัน ezmesh

6.3.3 การวาดกราฟ 3 มิติ แบบพื้นผิว

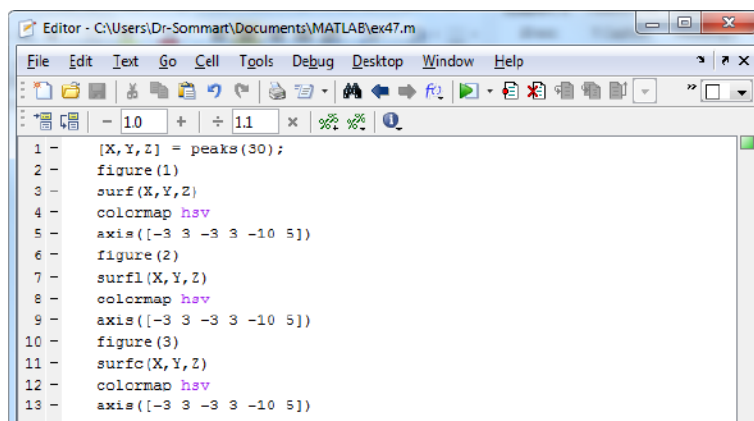
การวาดกราฟ 3 มิติ แบบพื้นผิว จะใช้ฟังก์ชัน surf, surfl, surfc, ezsurf และ ezsurf

ตัวอย่างที่ 6.7 จงใช้ฟังก์ชัน surf, surfl และ surfc สำหรับวาดกราฟ 3 มิติ โดยกำหนดให้

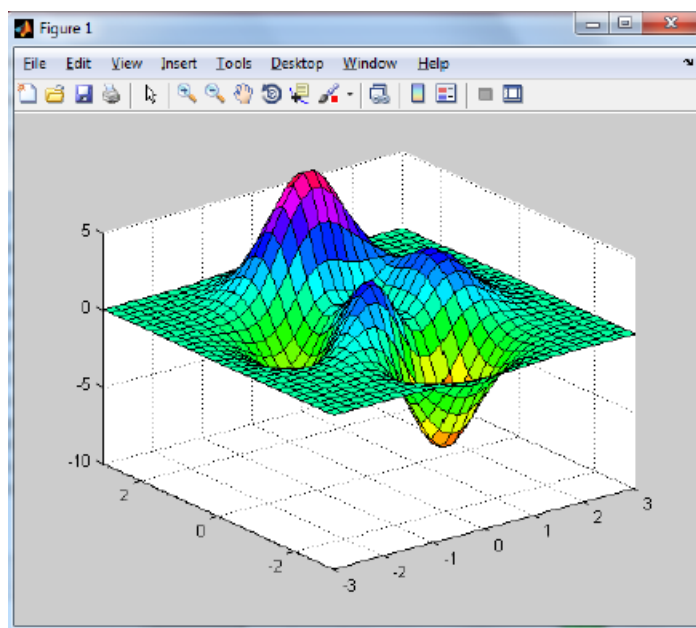
$[X,Y,Z] = \text{peaks}(30)$

วิธีทำ

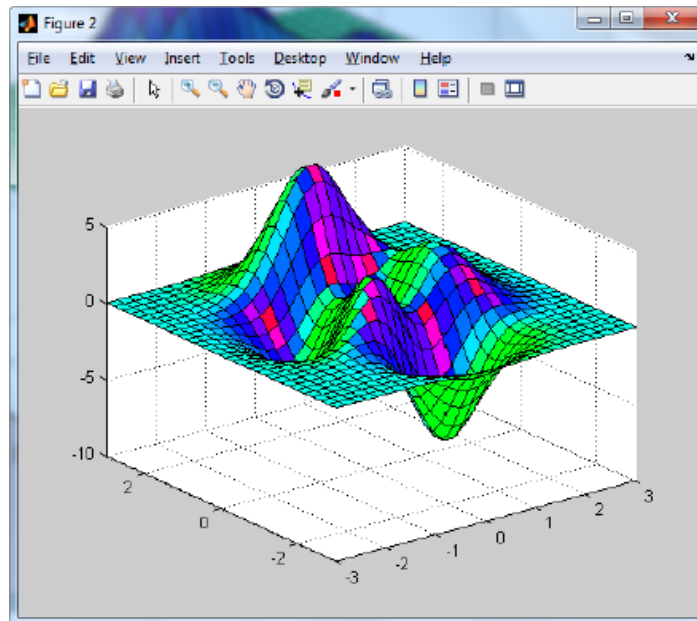
1. เปิดหน้าต่าง Editor ของ Script File แล้วกำหนดค่า $[X,Y,Z] = \text{peaks}(30)$
2. ทำการวาดกราฟโดยใช้ฟังก์ชัน surf, surfl และ surfc
3. กำหนดขนาดของกราฟเป็นดังนี้ $\text{axis}([-3 \ 3 \ -3 \ 3 \ -10 \ 5])$ ดังภาพที่ 6.23
4. เมื่อกดปุ่ม run แสดงผลลัพธ์ดังภาพที่ 6.24 – 6.26 ตามลำดับ



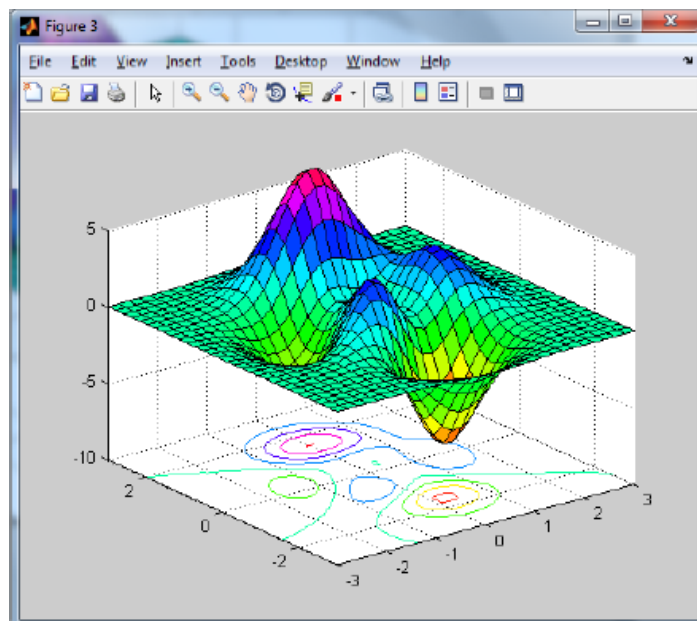
ภาพที่ 6.23 การเขียน Script File สำหรับวาดกราฟ 3 มิติ โดยใช้คำสั่ง surf, surfl และ surfc



ภาพที่ 6.24 ผลลัพธ์ของการใช้ฟังก์ชัน surf



ภาพที่ 6.25 ผลลัพธ์ของการใช้ฟังก์ชัน `surf`

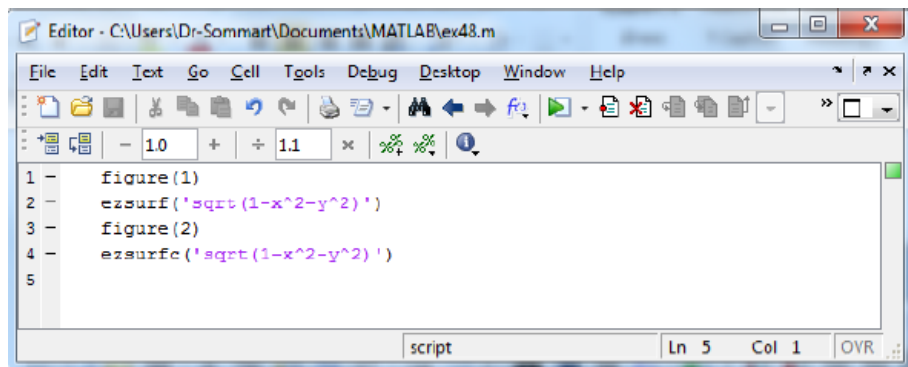


ภาพที่ 6.26 ผลลัพธ์ของการใช้ฟังก์ชัน `surf`

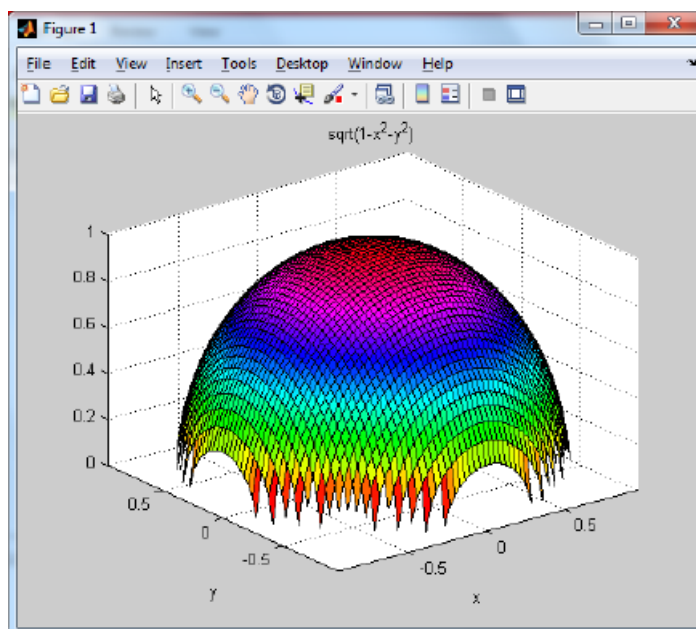
ตัวอย่างที่ 6.8 จงใช้ฟังก์ชัน `ezsurf` และ `ezsurf` สำหรับวาดกราฟ 3 มิติของ $\sqrt{1-x^2-y^2}$ เมื่อกำหนดให้ $x^2 - y^2 \leq 1$

วิธีทำ

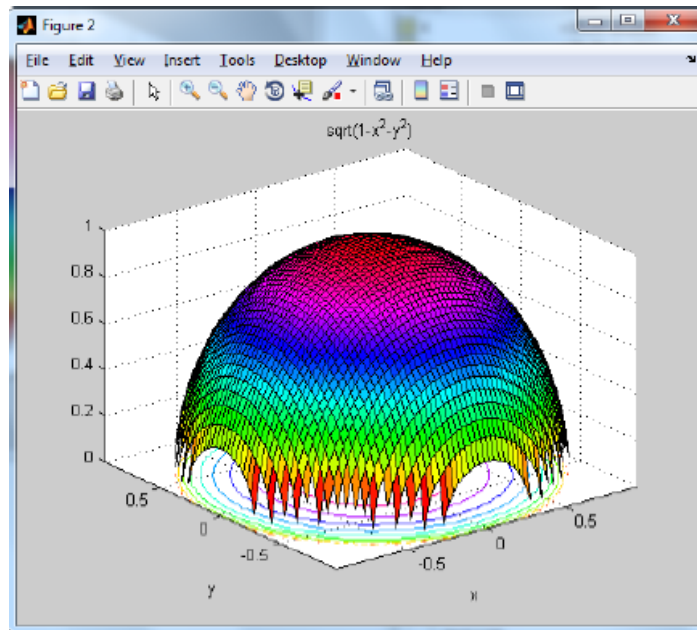
1. เปิดหน้าต่าง Editor ของ Script File
2. ทำการวาดกราฟโดยใช้ฟังก์ชัน `ezsurf` และ `ezsurf` ดังภาพที่ 6.27
3. เมื่อกดปุ่ม run แสดงผลลัพธ์ดังภาพที่ 6.28 – 6.29



ภาพที่ 6.27 การเขียน Script File สำหรับวาดกราฟ 3 มิติ
โดยใช้คำสั่ง `ezsurf` และ `ezsurf`



ภาพที่ 6.28 ผลลัพธ์ของการใช้ฟังก์ชัน `ezsurf`



ภาพที่ 6.29 ผลลัพธ์ของการใช้ฟังก์ชัน ezsurf

6.4 สรุป

การวาดกราฟของโปรแกรม MATLAB สามารถใส่ค่าข้อมูลได้ 2 ประเภท ได้แก่ 1) การกำหนดเป็นพิกัด และ 2) การกำหนดเป็นสมการโดยใน MATLAB มี function ซึ่งการวาดกราฟเส้น 2 มิติ ของโปรแกรม MATLAB เป็นการสร้างจากเส้นแกน 2 เส้น คือ แกนนอน (แกน y) และแกนตั้ง (แกน x) โดยโปรแกรม MATLAB มีฟังก์ชันที่ใช้สำหรับพล็อตกราฟเส้นในแกน x-y ได้หลายวิธีด้วยการกำหนดลักษณะของเส้น รูปแบบฟังก์ชันการทำงานของกราฟ plot ซึ่งสามารถสร้างกราฟในลักษณะเส้นหรือจุด โดยมีวิธีการใช้งานง่าย ๆ ดังนี้

`plot(x,y);`

การวาดกราฟเส้น 3 มิติ สามารถทำได้เช่นเดียวกับการวาดกราฟ 2 มิติ คือ สามารถป้อนค่าเข้าเป็นพิกัด หรือสมการ โดยที่การวาดกราฟ 3 มิติ ใช้ฟังก์ชัน plot3, contour3, contourslice, ezplote3 และ waterfall เช่น `plot3(x,y,z)` ถ้าค่า x,y และ z คือ เวกเตอร์ที่มีความยาวเดียวกัน การสร้างเส้นในแบบ 3 มิติ ผ่านจุดที่มีพิกัดที่มีองค์ประกอบของ x,y และ z

`plot3(x,y,z);`

แบบฝึกหัด

บทที่ 6 การเขียนโปรแกรม MATLAB วาดกราฟ 2 และ 3 มิติ

1. จงอธิบายรายละเอียดของฟังก์ชันการวาดกราฟ 2 มิติ และ 3 มิติของโปรแกรม MATLAB
2. จงวาดกราฟของฟังก์ชัน cosine โดยกำหนดให้กราฟมีสีน้ำเงินที่เวลา (t) 0 ถึง 2π
3. จงวาดกราฟในระบบพิกัดมุมฉาก โดยกำหนดให้ $f(x) = 2x + 3$ บนช่วง $[-5, 5]$
4. จงวาดกราฟหลายเส้น โดยกำหนดให้ $f(x) = 2x + 3$ กับ $g(x) = x^2 - 4$ บนช่วง $[-5, 5]$

5. จงวาดกราฟแท่งของข้อมูลทางสถิติ โดยกำหนดให้กราฟแท่งมีสีแดง ดังนี้

คนที่	คะแนนรวม
1	27
2	15
3	20
4	18
5	10

6. จงวาดกราฟในระบบพิกัดเชิงขั้ว โดยกำหนดให้กราฟเป็นรูปหัวใจ $r = 3 + 2\sin\theta$
7. จงวาดกราฟในระบบพิกัดเชิงขั้ว โดยกำหนดให้กราฟเป็นรูปกลีบกุหลาบ $r = 2\cos 2\theta$
8. จงวาดกราฟของฟังก์ชัน $\tan$ เทียบกับฟังก์ชัน atan โดยให้แสดงรูป 2 รูป และเส้นเป็นเส้นประมี + บนเส้น ฟังก์ชัน $\tan$ เส้นสีเขียว และฟังก์ชัน atan เส้นสีเหลือง
9. จงใช้ฟังก์ชัน `pie3` สำหรับวาดกราฟ 3 มิติ โดยกำหนดให้ $y = \sin(x)$
10. จงวาดกราฟ 3 มิติ แบบพื้นผิวไฮเพอร์โบลิกพาราโบลอยด์หรือพื้นผิวรูปอานม้า โดยกำหนดให้ $f(x, y) = x^2 - y^2$
11. จงวาดกราฟพารามетริก 3 มิติ ของเส้นโค้งที่มีสมการพารามетริก $r(t) = \cos t, y(t) = \sin t$ และ $z(t) = \frac{t}{4}$ บนช่วง $0 < t < 2\pi$

12. จงใช้ฟังก์ชัน mesh สำหรับวาดกราฟ 3 มิติ ของสมการสองตัวแปร ดังนี้

$$M(x,y) = x^2y^4 + x^4y^2 + 1 - 3x^2y^2$$

13. จงเขียนกราฟพื้นผิว $f(x,y) = x^2 - y^2$ บนช่วง $[-2,2] \times [-2,2]$ และกราฟ contour

14. จงเขียนคำสั่ง MATLAB เพื่อวาดกราฟ 3 มิติ โดยกำหนดสมการ $z_1 = x^2 + y^2$ และ $z_2 = 25 - x^2 - y^2$ มีเงื่อนไขดังนี้ $x, y \in [-5,5]$ และ $\Delta x = 0.1, \Delta y = 0.1$ (เปรียบเทียบกราฟทั้งสอง)

15. จงเขียนคำสั่ง MATLAB เพื่อสร้างกราฟของสมการต่อไปนี้

$$v(x) = \sqrt{\frac{1 + ax^2}{(1 - x^2)^2 + ax^2}}$$

โดยกำหนดให้ตัวแปรต้นมีค่าขอบเขตระหว่าง 0 ถึง 4 และมีค่าความละเอียดกราฟ 0.01

- ก. กำหนดให้ $a = 0.160$ สร้างกราฟของสมการโดยให้เส้นกราฟเป็นสีแดง
- ข. กำหนดให้ $a = 0.025$ สร้างกราฟของสมการโดยให้เส้นกราฟเป็นสีน้ำเงินแบบเส้นประ
- ค. แสดงสองกราฟเปรียบเทียบโดยการแสดงในรูปเดียวกันพร้อมระบุชื่อกราฟและแกน

ส่วนต่อประสานกราฟฟิคกับผู้ใช้ของโปรแกรม MATLAB

การออกแบบส่วนส่วนติดต่อกับผู้ใช้งานเป็นการออกแบบจอภาพออกแบบจอภาพ เพื่อให้ผู้ใช้งานสามารถโต้ตอบกับระบบได้ตามต้องการอย่างมีประสิทธิภาพ ปัจจุบันการออกแบบส่วนติดต่อกับผู้ใช้งานแบบกราฟิกหรือ GUI (Graphic User interface) มุ่งเน้นการปฏิสัมพันธ์ระหว่างมนุษย์กับคอมพิวเตอร์ จึงควรมีการออกแบบเพื่อดึงดูดความสนใจ และเลือกใช้สื่ออุปกรณ์ที่เหมาะสมกับปฏิสัมพันธ์ เพราะสามารถสื่อสารกับผู้ใช้งานในรูปแบบข้อและรูปภาพต่าง ๆ ได้ ทำให้ใช้งานสะดวกและเรียนรู้ได้รวดเร็ว การออกแบบส่วนติดต่อกับผู้ใช้งานมีหลายรูปแบบ ซึ่งแต่ละแบบอาจนำมาทำงานร่วมกันได้

ส่วนต่อประสานกราฟฟิคกับผู้ใช้ของโปรแกรม MATLAB จะเรียกว่า GUI ย่อมาจาก Graphic User Interface เป็นเชื่อมประสานระหว่างผู้ใช้งานกับคอมพิวเตอร์ผ่านสัญลักษณ์หรือภาพ เพื่ออำนวยความสะดวกให้ผู้ใช้งานสามารถติดต่อกับโปรแกรมได้ง่ายขึ้น ดังนั้น เมื่อผู้เขียนโปรแกรมเสร็จเรียบร้อยแล้วใช้งานแล้วจะสร้างหน้า GUI ขึ้นมา แล้วเขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนต่างลงใน GUI แทน ซึ่งจะทำให้โปรแกรมดูสวยงามและสะดวกต่อการใช้งานจริง

7.1 การเรียกใช้งาน GUI ด้วย GUIDE

โปรแกรม MATLAB จะสร้างส่วนต่อประสานกราฟฟิคกับผู้ใช้ (Graphical User Interface : GUI) GUI อยู่บนหน้าต่างรูปภาพ (Figure Window) ซึ่งภายในหน้าต่างนี้จะมีส่วนประกอบต่าง ๆ อยู่ได้ไม่ว่าจะเป็น Axes, Uicontrol หรือวัตถุอื่น ๆ เราสามารถสร้าง Uicontrol, Uimenu แบบต่าง ๆ ลงในหน้าต่างรูปภาพได้แต่เป็นไปด้วยความลำบากเพราะการสร้างเป็น Text Base ต่อมาจนกระทั่ง ปัจจุบัน MATLAB ได้สร้าง Graphical User Interface Development Environment หรือ GUIDE ขึ้นเพื่อช่วยให้สร้างบันทึก และแก้ไข GUI ได้สะดวกขึ้น

การสร้าง GUI จะประกอบด้วย 2 ขั้นตอนดังนี้

- 1) กำหนดและวางส่วนประกอบต่าง ๆ ลงบน GUI
- 2) เขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนต่าง ๆ ใน GUI

GUIDE โดยหลักใหญ่จะมีหน้าที่ในการวางส่วนประกอบที่ต้องการให้มีลงใน GUI จากนั้น GUIDE จะสร้าง M-file ที่บรรจุ Handle ของวัตถุหรือ Object ทั้งหมดที่สร้างขึ้นรวมทั้งคำสั่ง GUI ทำงาน นอกจากนั้น M-file จะให้แนวทางในการเขียนฟังก์ชัน ที่ทำงานหลังจากผู้ใช้กดเมาส์ปุ่มซ้ายหรือปรับเปลี่ยนค่าของวัตถุนั้น เรียกว่า Callback ของวัตถุนั้น

การสร้าง GUI ขึ้นมาได้โดยการเขียนเป็น M-File ขึ้นมา แต่การใช้ GUIDE ทำให้การทำงานง่ายขึ้น เพราะจะช่วยให้กำหนดตำแหน่งของวัตถุต่าง ๆ ได้โดยง่าย หลังจากนั้น GUIDE จะสร้างไฟล์ขึ้นมา 2 ไฟล์ เพื่อเก็บและนำ GUI มาใช้ต่อไป ประกอบด้วย

1) FLG-File จะบรรจุรายละเอียดของวัตถุต่าง ๆ ที่องค์ประกอบอยู่ในหน้าต่างรูปภาพที่เป็น GUI

2) M-File ที่บรรจุฟังก์ชันที่กำหนดการทำงานของ GUI รวมถึง Callback ทั้งหมด ซึ่ง Callback เหล่านี้จะบรรจุเป็น Sub Function อยู่ใน M-File และเรียก M-File ที่ควบคุมการทำงานของ GUI นี้ Application M-File

ดังนั้น Application M-File จะไม่มีข้อมูล จะไม่มีข้อมูลเกี่ยวกับรูปแบบของส่วนประกอบที่บรรจุอยู่ใน GUI เช่น สี ขนาด ตำแหน่ง อื่น ๆ เลย เพราะข้อมูลเหล่านั้นจะบรรจุอยู่ใน Fig-File

GUIDE จะรวบรวมองค์ประกอบต่าง ๆ ภายใน GUI แล้วสร้าง Application M-File โดยอัตโนมัติโดยรูปแบบของการสร้างที่ชัดเจน เพื่อให้ได้โครงสร้างของ Application M-File จากนั้นสามารถนำโครงสร้างโดยอัตโนมัตินั้นมาปรับแก้ เพื่อให้เกิดการควบคุม GUI ตามต้องการ การกระทำดังกล่าวทำให้ได้ข้อได้เปรียบหลายประการ

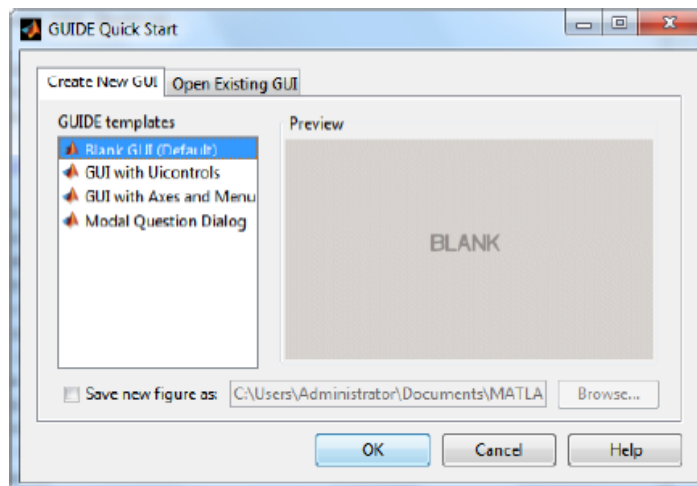
- M-File จะประกอบด้วยคำสั่งที่จำเป็นในการควบคุม GUI ครบถ้วน
- M-File จะทำให้เราส่งข้อมูลไปที่ส่วนต่าง ๆ ได้ง่าย สะดวก รวดเร็ว
- M-File จะทำให้เราส่งข้อมูลไปที่ส่วนต่าง ๆ ภายใน MATLAB ได้
- Application M-File จะสร้าง Sub Function สำหรับ Uicontrols ทุกแบบที่มีใน GUI เพื่อทำให้เราเขียน Callback ต่าง ๆ ได้สะดวกขึ้น

3. การเลือก GUIDE Application Options

เมื่อต้องการจะใช้งาน GUIDE ครั้งแรกบน MATLAB Command Window ที่ Prompt ดังนี้

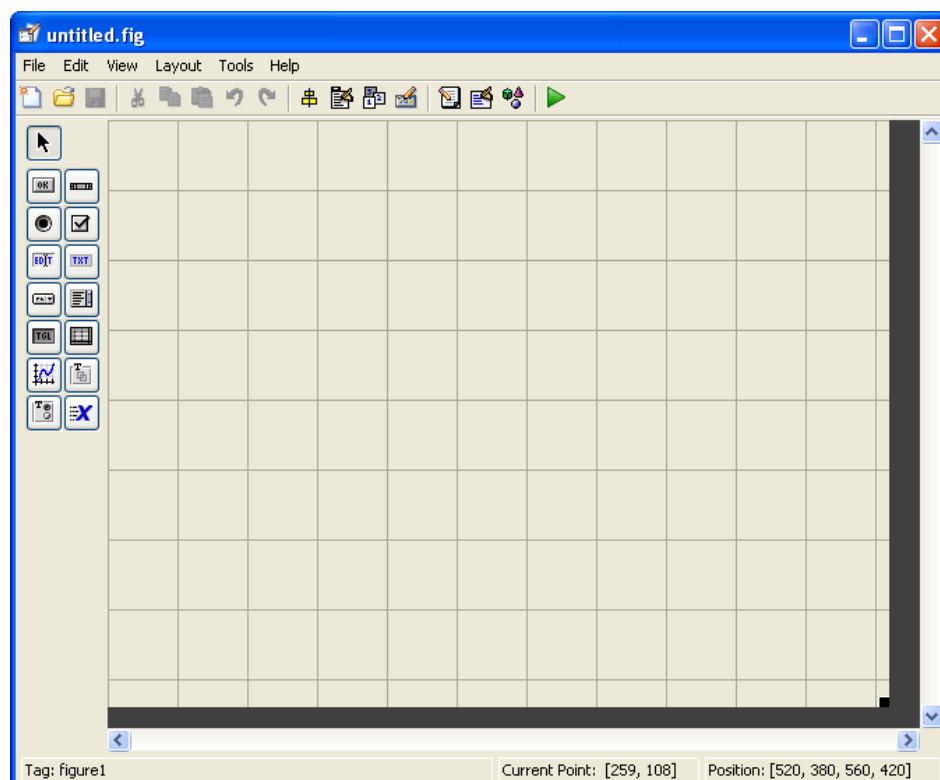
```
>> guide
```

เมื่อพิมพ์คำสั่ง guide ที่ Command Window หน้าต่าง GUIDE ปรากฏขึ้นมา ดังภาพที่ 7.1



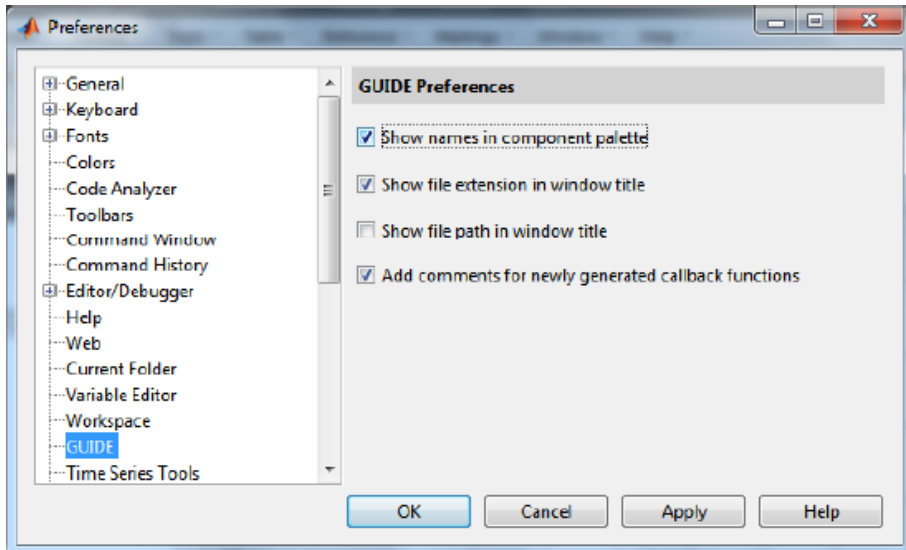
ภาพที่ 7.1 หน้าต่าง GUIDE

หน้าต่าง GUIDE Quick Start เป็นหน้าต่างสำหรับเลือกสร้าง GUI ใหม่ หรือเปิด GUI เดิมที่เคยสร้างไว้แล้ว หากต้องการสร้าง GUI แบบว่างเปล่าให้เลือก Blank GUI (Default) แล้วคลิกปุ่ม OK จะปรากฏหน้าต่างใหม่พร้อมเครื่องมือในการสร้าง GUI ดังภาพที่ 7.2

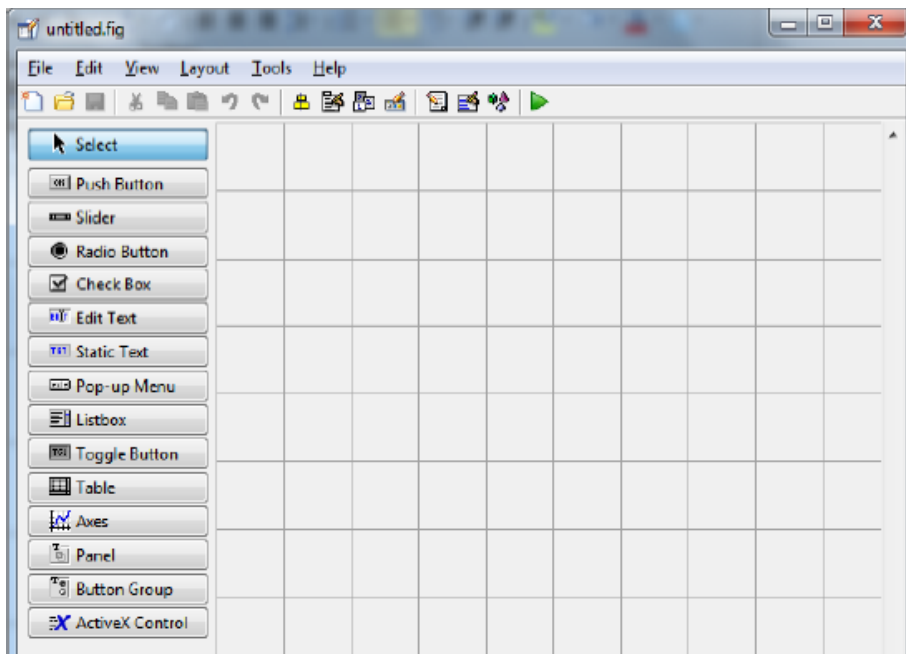


ภาพที่ 7.2 หน้าต่างสำหรับเลือกสร้าง GUI ใหม่

จากภาพที่ 7.2 หากต้องการปรับเปลี่ยนรูปแบบการแสดงลักษณะของหน้าต่าง สามารถทำได้ โดยการคลิกเลือก File >> Preference ปรากฏหน้าต่าง Preference ดังภาพที่ 7.3 และทำการคลิก ✓ ที่ Show names in component palette แล้วคลิก OK ทำให้วัตถุที่จะนำมาสร้างเป็นส่วนติดต่อกับผู้ใช้งานแสดงรายชื่อ ดังภาพที่ 7.4



ภาพที่ 7.3 หน้าต่าง Preference



ภาพที่ 7.4 หน้าต่างการสร้าง GUI ที่แสดงรายชื่อของวัตถุต่าง ๆ

จากภาพที่ 7.4 วัตถุต่าง ๆ ที่สร้างขึ้นใน GUI เป็นวัตถุส่วนติดต่อผู้ใช้ (User Interface Object) เดียวกัน แต่วัตถุที่คุณสมบัติ Style ที่สร้าง ประกอบด้วย

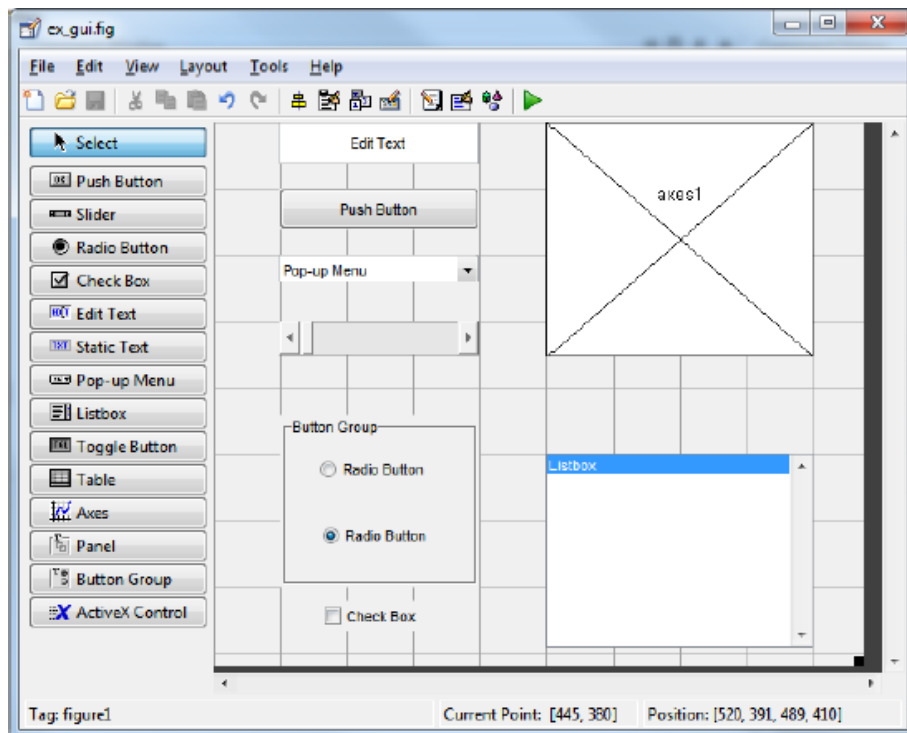
- Push Button เหมาะสำหรับให้ผู้ใช้กดเพื่อเป็นการตอบตกลง ยกเลิก หรือใช้ในการบอกให้เริ่มต้นหรือสิ้นสุดเหตุการณ์ใด ๆ
- Slider Bar เหมาะสำหรับให้ผู้ใช้ปรับค่าแบบเพิ่มหรือลด ขึ้นหรือลง ซ้ายหรือขวา เป็นต้น
- Radio Button เหมาะสำหรับเป็นรายการให้ผู้ใช้เป็นตัวเลือกเดียว ซึ่งผู้ใช้สามารถเห็นตัวเลือกได้ทุกตัว โดยบรรจุ Radio Button ที่ต้องการให้เลือกข้อใดข้อหนึ่งไว้ใน Button Group (Panel) เดียวกัน
- Check Box เหมาะสำหรับให้ผู้ใช้เลือกเพื่อเปิดหรือปิด เอาหรือไม่เอา ใช่หรือไม่ใช่ เป็นต้น
- Edit Text เหมาะสำหรับให้ผู้ใช้ป้อนข้อมูล
- Static Text เหมาะสำหรับการแสดงข้อความ คำอธิบาย ชื่อ เป็นต้น ในลักษณะถาวร โดยผู้ใช้ไม่มีสิทธิ์เปลี่ยนแปลงได้ (แต่ผู้พัฒนาโปรแกรมสามารถเปลี่ยนแปลงข้อความได้โดยการเขียนโปรแกรม)
- Pop-up Menu เหมาะสำหรับเป็นรายการให้ผู้ใช้เป็นตัวเลือก แบบเลือกได้ตัวเลือกเดียว
- List Box เหมาะสำหรับเป็นรายการให้ผู้ใช้เป็นตัวเลือกแบบเลือกได้ตัวเลือกเดียวหรือหลายตัวเลือกพร้อมกัน
- Axes เหมาะสำหรับการแสดงกราฟฟิก เช่น กราฟ ภาพ หรือ วิดีโอ เป็นต้น

7.2 การสร้าง GUI ด้วย GUIDE

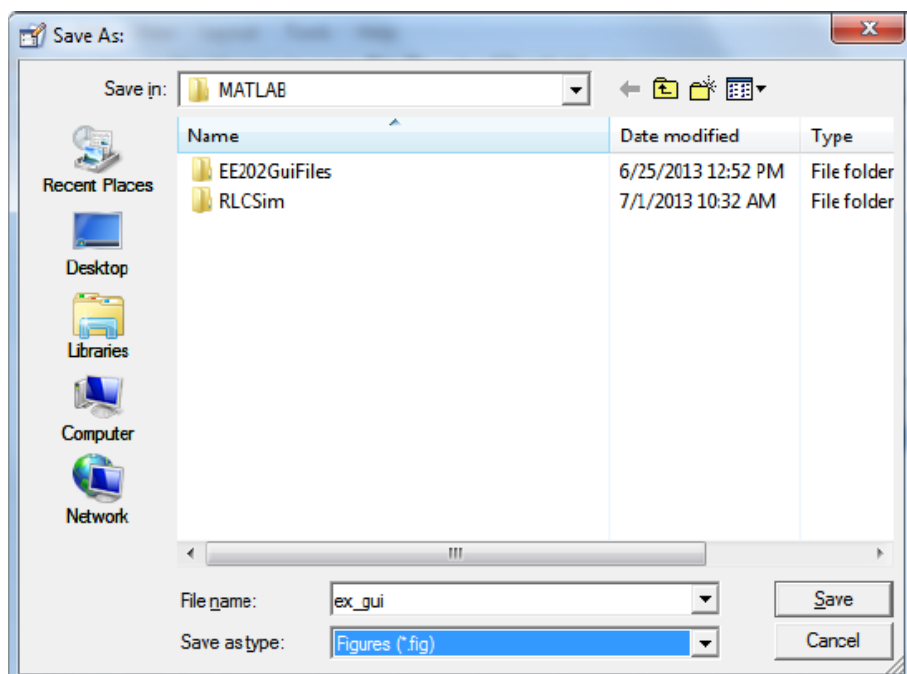
การสร้าง GUI ทำการเลือกวัตถุในเมนูด้านซ้าย โดยให้สร้าง GUI ให้มีลักษณะดังภาพที่ 7.5 จากนั้นคลิกปุ่ม Save ที่เมนูบาร์ ปรากฏหน้าต่างสำหรับการบันทึก ดังภาพที่ 7.6 พร้อมทั้งตั้งชื่อเป็น ex_gui

ช่อง File Name ให้ตั้งชื่อ ซึ่งชื่อนี้มีความสำคัญ ดังนี้

1. เป็นชื่อหน้าต่าง Figure ของโปรแกรม
2. เป็นชื่อฟังก์ชันที่ใช้ในการเรียกเพื่อเปิดใช้งาน GUI ที่ได้สร้างขึ้น

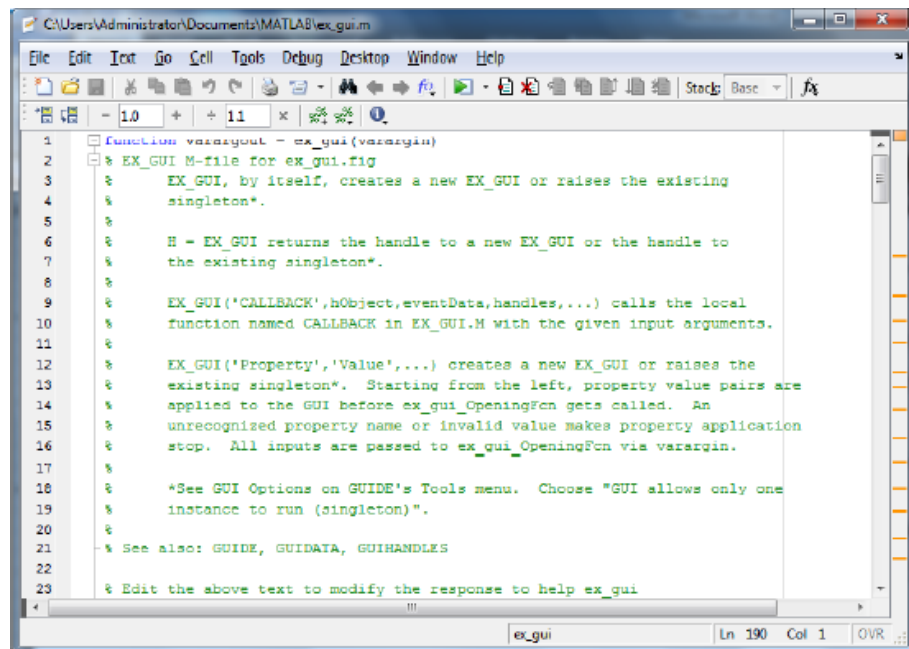


ภาพที่ 7.5 หน้าต่าง GUIE เลือกวัตถุในเมนู



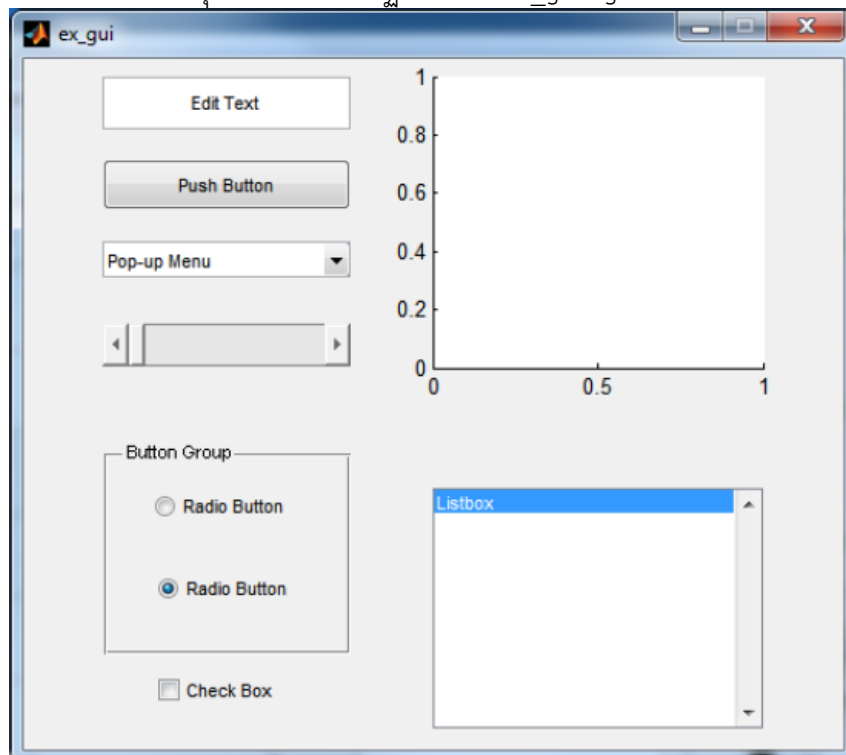
ภาพที่ 7.6 หน้าต่างการบันทึกของ GUI

เมื่อทำการตั้งชื่อ ex_gui แล้วคลิกปุ่ม Save เพื่อทำการบันทึก จะปรากฏแฟ้มข้อมูลเกิดขึ้นจำนวน 2 File คือ File นามสกุล .fig และ File นามสกุล .m ดังภาพที่ 7.7



ภาพที่ 7.7 หน้าต่าง File ex_gui.m

และเมื่อทำการคลิกปุ่ม run จะปรากฏหน้าต่าง ex_gui.fig ดังภาพที่ 7.8



ภาพที่ 7.8 หน้าต่าง File ex_gui.fig

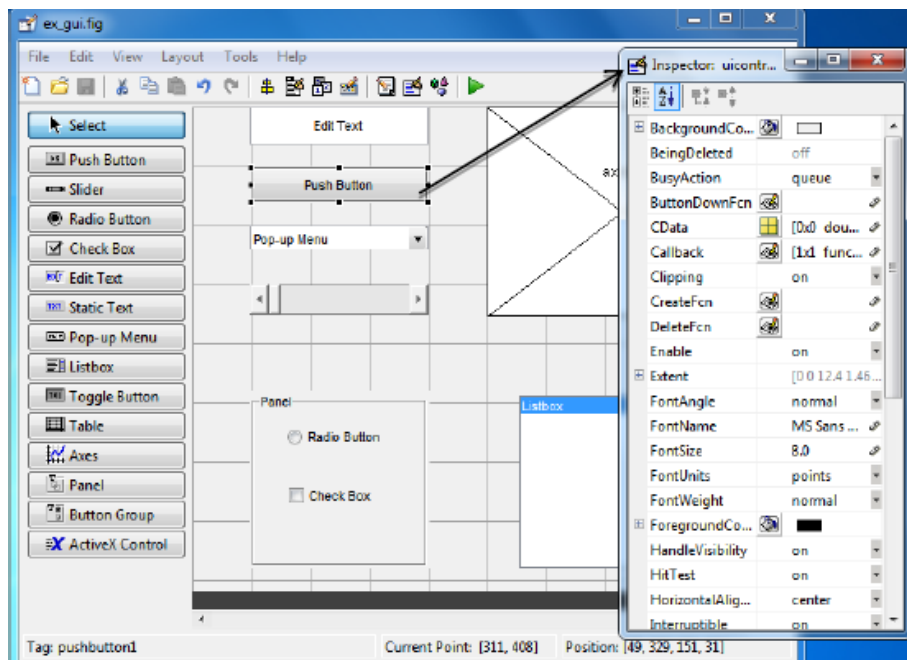
7.3 การเขียนโปรแกรม GUI ด้วย GUIDE

การเขียน GUI ในโปรแกรม MATLAB มีหลักการสำคัญอยู่ที่ตัวแปร **handles** เพื่อใช้เป็นตัวอ้างอิงหรือชี้ไปที่วัตถุแต่ละวัตถุ ค่าคุณสมบัติ (Property) ต่าง ๆ ของแต่ละวัตถุ และฟังก์ชันเรียกกลับ (Callback Function) ซึ่งมีหลักการเดียวกับการแก้ไขหน้าต่าง figure

ตัวแปร handles เป็นตัวแปรที่ใช้เก็บเลขจำนวนหนึ่งที่ใช้เป็นตัวอ้างอิงถึงวัตถุแต่ละวัตถุ คล้ายกับ Binder ในภาษา C โดยหากสร้าง GUI ด้วย GUIDE ตัวแปร handles จะถูกสร้างให้เองสำหรับทุก ๆ วัตถุที่สร้างขึ้น และบรรจุไว้ในตัวแปรชื่อ handles แล้วใช้ Tag ในการเรียกใช้

7.3.1 การเปลี่ยนแปลงค่าคุณสมบัติของวัตถุ

การเปลี่ยนแปลงค่าคุณสมบัติของวัตถุที่สร้าง GUI ทำได้โดยการดับเบิลคลิกที่วัตถุแต่ละชิ้น เช่น ดับเบิลคลิกที่ Push Button ปรากฏหน้าต่าง Inspector ซึ่งสามารถแก้ไขค่าในหน้าต่างได้ ดังภาพที่ 7.9



ภาพที่ 7.9 การเปลี่ยนแปลงค่าคุณสมบัติของวัตถุ

กรณีต้องการเปลี่ยนแปลงค่าคุณสมบัติขณะที่โปรแกรมกำลังใช้งานอยู่ทำได้ คือ เรียกดูค่าและกำหนดค่าคุณสมบัติได้ที่ฟังก์ชัน get และ set ตามลำดับ ซึ่ง handles ถูกส่งมาให้วัตถุเรียกกลับทุกฟังก์ชันภายในโปรแกรม กรณีใช้ GUIDE ในการสร้างวัตถุส่วนติดต่อผู้ใช้แต่ละแบบ (Style) มีคุณสมบัติที่สำคัญต่าง ๆ กัน เช่น static Text มีคุณสมบัติ คือ เป็น String สามารถเรียกดูได้โดย

```
A = get(handles.text1.string);
```

หากต้องการเปลี่ยนค่าตัวแปร A ที่เก็บไว้ในสตริง สามารถทำได้ ดังนี้
Set(handles.text1, string, Hello World)

ตอนนี้ static text ถูกชี้ด้วย handles.text1 ทำให้มีข้อความเปลี่ยนเป็น Hello World แทน หรือสามารถทำการเปลี่ยนแปลงค่าคุณสมบัติของวัตถุส่วนติดต่อผู้ใช้งานได้โดยผ่านฟังก์ชัน guidata ดังนี้

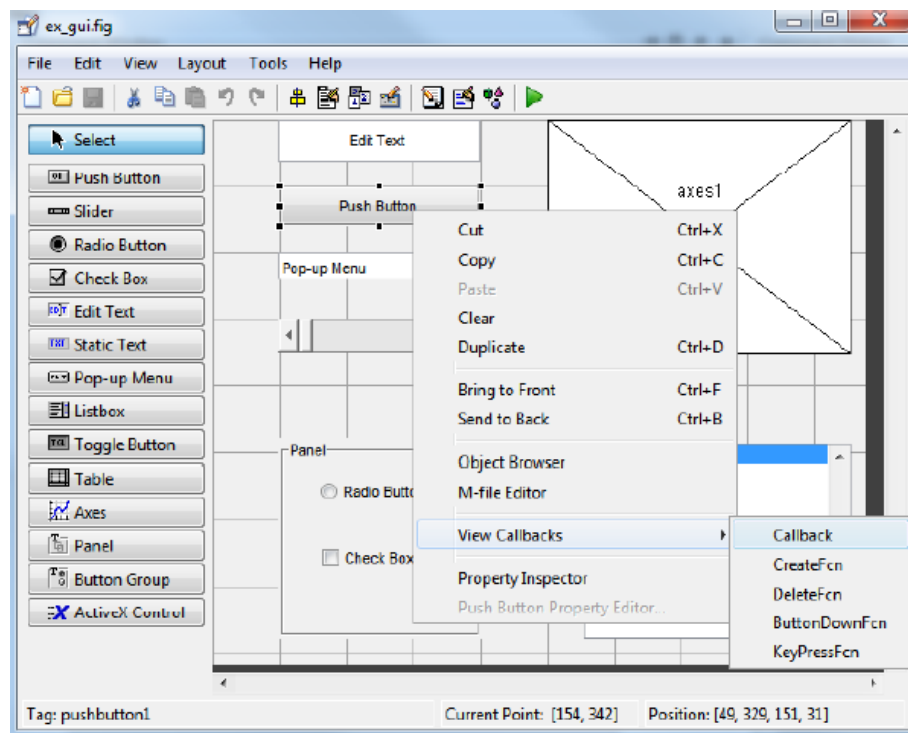
Guidata(hObject,handles)

ตัวแปร hobject หมายถึง handles ของวัตถุที่เป็นตัวเรียกใช้ฟังก์ชันเรียกกลับนั้น ถูกส่งมาให้ในฟังก์ชันเรียกกลับภายในโปรแกรมที่สร้างด้วย GUIDE อยู่แล้ว

7.3.2 ฟังก์ชันเรียกกลับ (Callback Function)

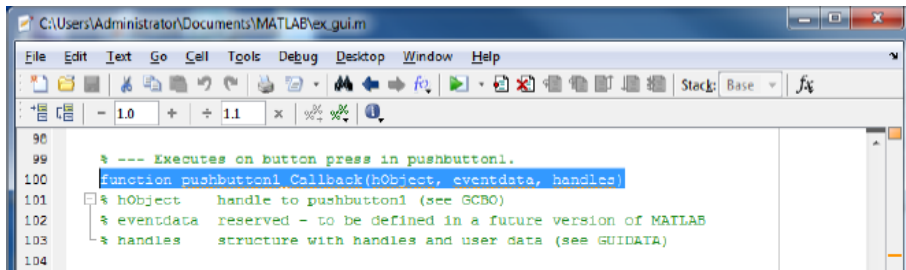
ฟังก์ชันเรียกกลับ เป็นฟังก์ชันที่ถูกเรียกใช้งาน เพื่อตอบสนองเหตุการณ์ เช่น การกดปุ่ม การลากเมาส์ผ่าน การคลิก เป็นต้น

การสร้างฟังก์ชันเรียกกลับด้วย GUIDE โดยคลิกขวาวบนวัตถุที่ต้องการ แล้วเลือก View Callbacks ในฟังก์ชันที่ต้องการ เช่น การสร้างฟังก์ชันเรียกกลับของปุ่ม Push Button ให้คลิกขวาที่บริเวณ Push Button เลือก View Callbacks ดังภาพที่ 7.10

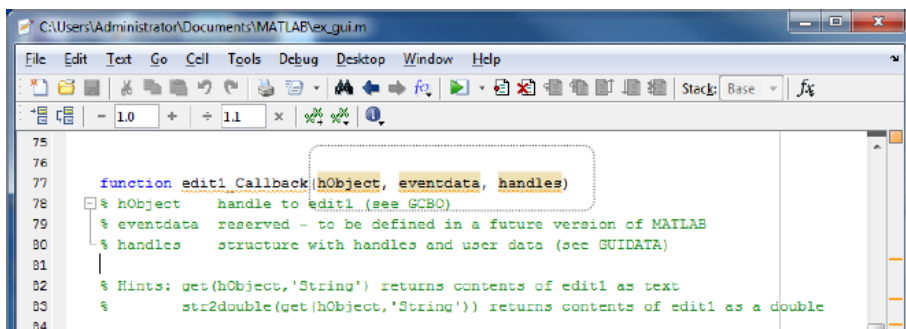


ภาพที่ 7.10 การสร้างฟังก์ชันเรียกกลับด้วย GUIDE

จากภาพที่ 7.10 พบว่าเมนูย่อยเป็นรายการฟังก์ชันเรียกกลับทั้งหมดของปุ่ม Push Button ในบทรนี้เลือกฟังก์ชันเรียกกลับชื่อ Callback โปรแกรมจะสร้างโค้ดของฟังก์ชันนี้ขึ้นให้อัตโนมัติใน ex_gui.m ดังภาพที่ 7.11



ภาพที่ 7.11 เมนูย่อยเป็นรายการฟังก์ชันเรียกกลับทั้งหมดของปุ่ม Push Button โดยฟังก์ชันเรียกกลับนี้ ถูกสร้างขึ้นในลักษณะเดียวกัน ภาพที่ 7.12



ภาพที่ 7.12 เมนูย่อยเป็นรายการฟังก์ชันเรียกกลับทั้งหมดของปุ่ม Edit Text

จากภาพที่ 7.12 ตัวแปรทั้ง 3 มีรายละเอียด ดังนี้

1. hObject เป็นตัวแปร handle หรือตัวชี้ของวัตถุที่ใช้เป็นตัวเรียกฟังก์ชันเรียกกลับ โดยที่มีค่าเดียวกับ handles.pushbutton1
2. eventdata ตัวนี้ถูกใช้ในบางฟังก์ชันเรียกกลับบางฟังก์ชันเท่านั้น เพื่อเก็บข้อมูลบางอย่าง เช่น แป้นพิมพ์ที่ถูกกด หรือตัวเลือกที่ถูกเลือก เป็นต้น
3. handles เป็นตัวแปรที่สำคัญที่สุดในโปรแกรม GUI ซึ่งใช้ในการเก็บ handles ของวัตถุต่าง ๆ รวมกันทั้งหมด

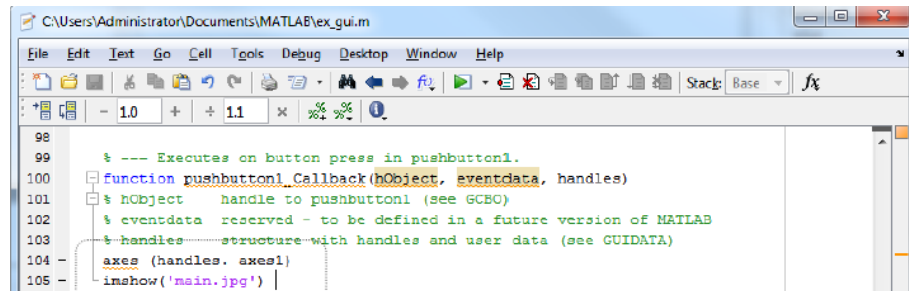
7.4 การเขียน GUI ด้วย GUIDE

การเขียน GUI ด้วย GUIDE มีเป้าหมายเพื่อการทำงานของวัตถุส่วนติดต่อผู้ใช้ทั้งหมด โดยการสร้างให้แต่ละวัตถุมีปฏิสัมพันธ์ติดต่อกัน

7.4.1 การใช้ Push Button และ Axes

หากต้องการการเขียนโปรแกรมเมื่อกดปุ่ม (pushbutton1) แล้วปรากฏภาพขึ้นในแกน (axes1) ดังนั้นวัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับคือ pushbutton1

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Push Button ที่มี Tag เป็น pushbutton1 ใน GUIDE แล้วเลือก View Callbacks > Callback ดังภาพที่ 7.13
2. ทำการแก้ไขโปรแกรม ดังนี้



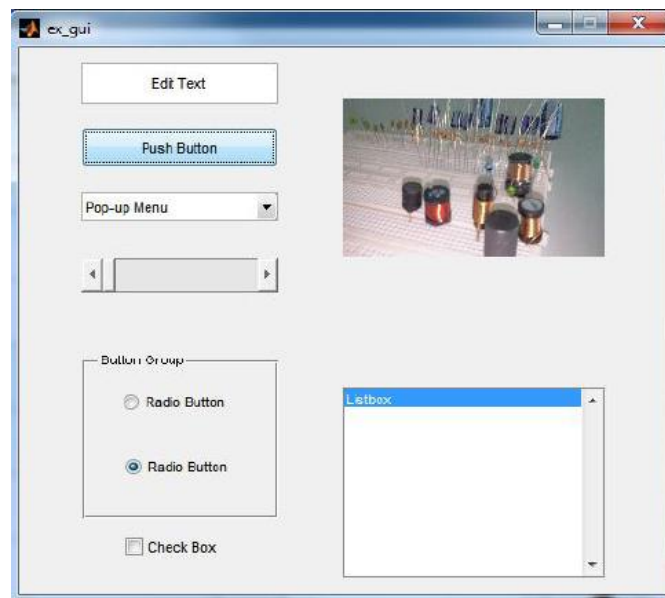
ภาพที่ 7.13 การเขียนโปรแกรมฟังก์ชันเรียกกลับโดยใช้ปุ่ม Push Button

Main เป็นข้อมูลภาพ (main.jpg) ที่มีอยู่แล้วในโปรแกรม MATLAB

Axes(handles.axes1)

ฟังก์ชัน axes นี้ไว้ใช้เลือกแกนที่ต้องการทำงานต่อไป ในกรณีที่ GUI มีหลายแกน หากมีเพียงแกนเดียว คำสั่งนี้ไม่จำเป็น และสามารถลบออกได้

3. คลิกปุ่ม run หรือปุ่ม <F5> แล้วทดลองคลิกปุ่ม Push Button จะปรากฏภาพในวงจรรแกน axes1 ดังภาพที่ 7.14



ภาพที่ 7.14 การเขียน GUI ด้วย Push Button และ Axes

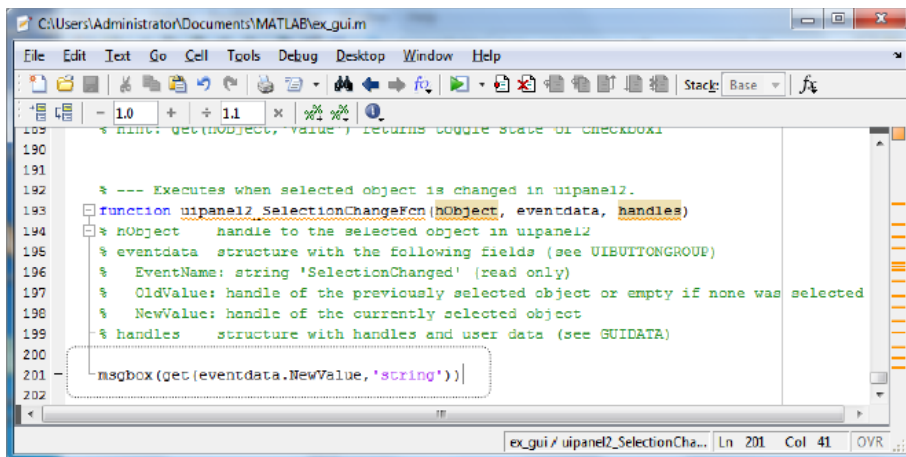
หมายเหตุ : ควรใส่ภาพที่ต้องการใช้งานไว้ใน Folder ของโปรแกรม MATLAB ก่อน

7.4.2 การใช้ Radio Button และ Button Group

การเขียนโปรแกรมนี้ เมื่อเลือกตัวใดตัวหนึ่งแล้ว จะมีกล่องข้อความของตัวเลือกนั้นปรากฏขึ้น ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับ คือ Button Group (uipanel1) ที่ใช้บรรจุ Radio Button (radio button1 และ radio button2) ทั้ง 2 ตัวเลือก

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Button Group ที่มี Tag เป็น uipanel1 ใน GUIDE แล้วเลือก View Callbacks > Callback เหมือนกับข้อความที่แล้ว

2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.15

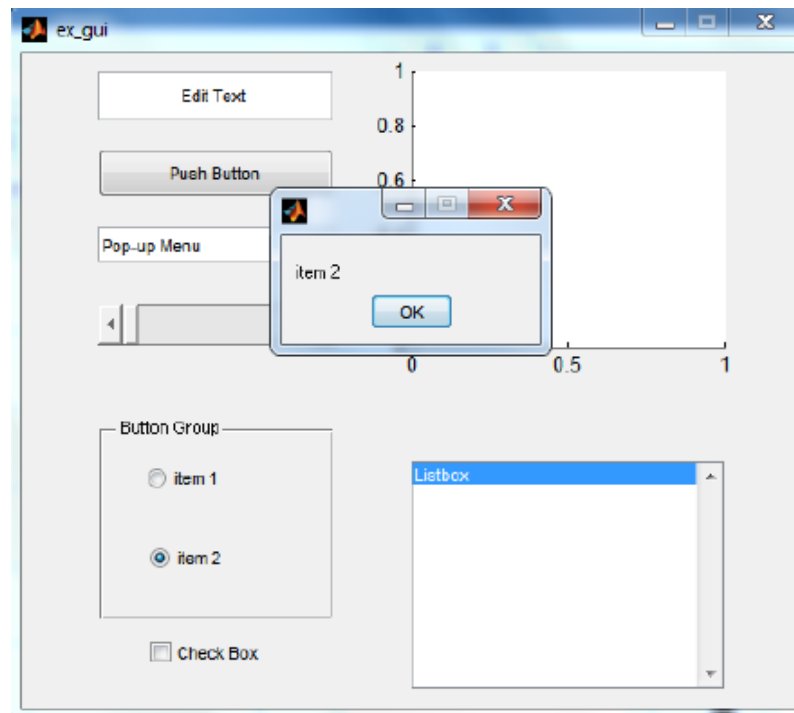


ภาพที่ 7.15 การเขียนโปรแกรมฟังก์ชันเรียกกลับโดยใช้ปุ่ม Button Group

ฟังก์ชันเรียกกลับนี้ใช้ตัวแปร eventdata ในการเก็บตัวแปร handle ของ Radio Button ที่ถูกเลือกในฟิลด์ NewValue และตัวแปร handle ของ Radio Button ที่ถูกเลือกก่อนหน้าในฟิลด์ oldValue

3. ทำการแก้ไขคุณสมบัติ String ของ radio button1 และ radio button2 ในหน้าต่าง Inspector เป็น item1 และ item2 ตามลำดับ

4. ทำการทดลองเลือก item2 ปรากฏกล่องข้อความที่แสดงตัวเลือก ดังภาพที่ 7.16

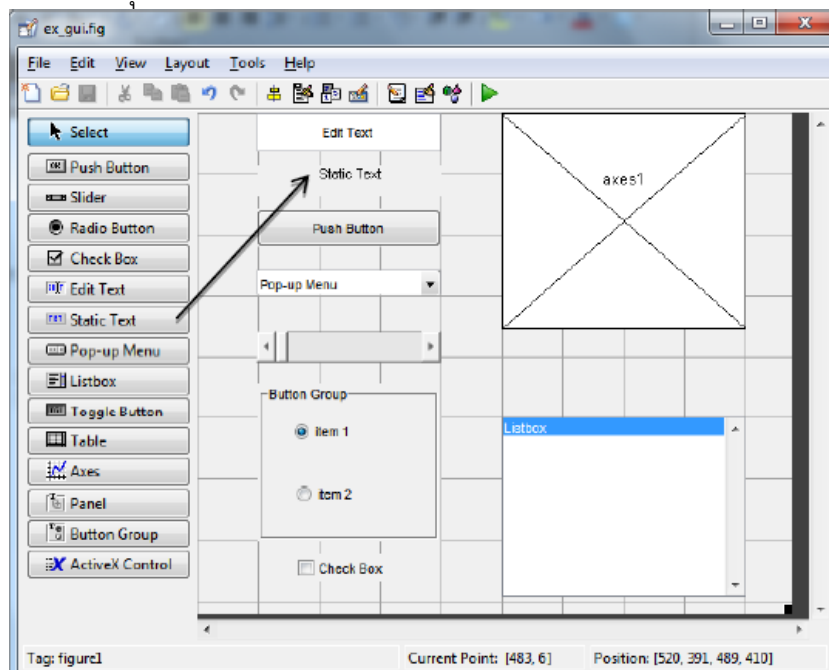


ภาพที่ 7.16 การเขียน GUI โดยใช้ Radio Button และ Button Group

7.4.3 การใช้ Edit Text และ Static Text

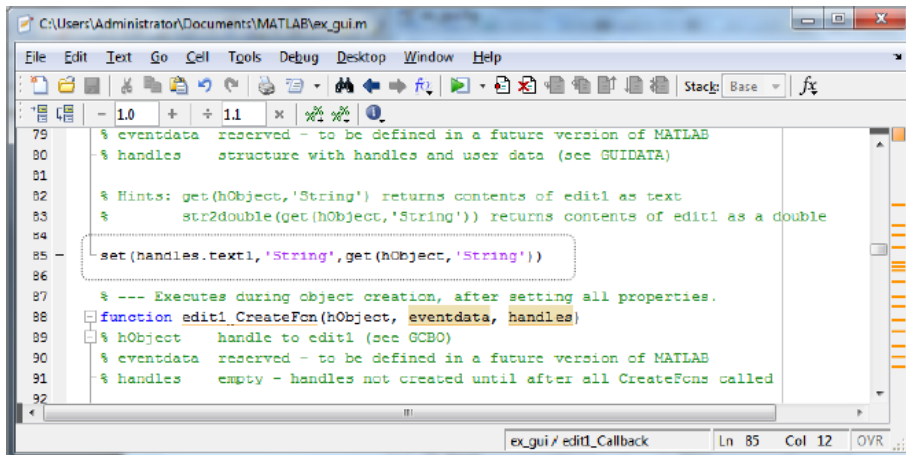
การเขียนโปรแกรมนี้ เมื่อพิมพ์ข้อความใด ๆ ใน Edit Text แล้วกดปุ่ม <Enter> ปรากฏข้อความนั้นขึ้นใน Static Text ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับ คือ Edit Text (edit1)

1. ให้เพิ่มวัตถุ Static Text ในหน้าต่าง GUI ดังภาพที่ 7.17



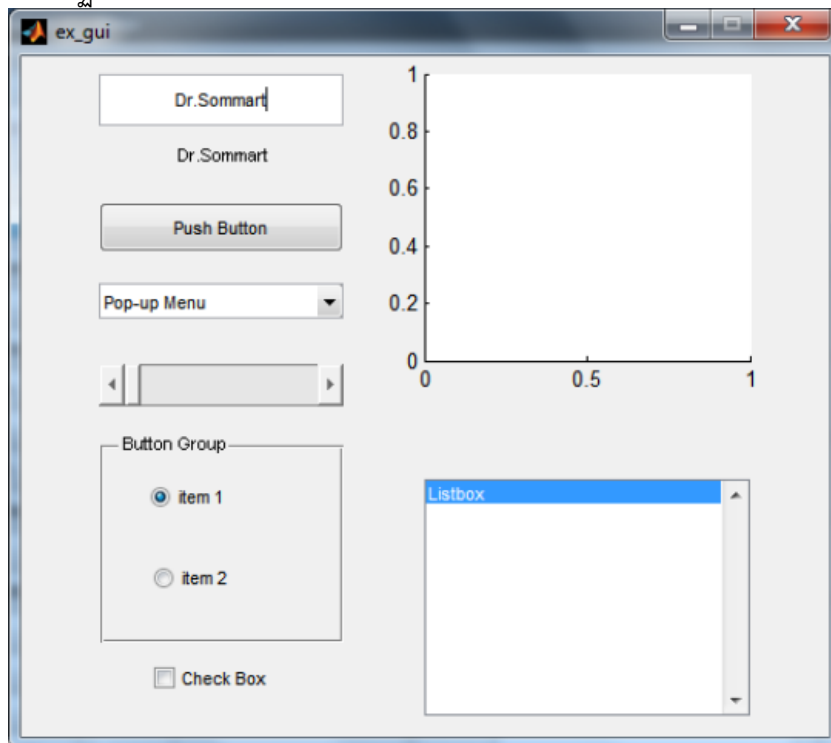
ภาพที่ 7.17 การเพิ่ม Static Text ในหน้าต่าง GUI

2. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Edit Text ที่มี Tag เป็น edit1 ใน GUIDE แล้วเลือก View Callbacks > Callback เหมือนกับข้อความที่แล้ว
3. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.18



ภาพที่ 7.18 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Edit Text

4. คลิกปุ่ม run หรือปุ่ม <F5> แล้วทดลองพิมพ์ข้อความในช่อง Edit Text แล้วกดปุ่ม <Enter> จะปรากฏข้อความใน Static Text ดังภาพที่ 7.19

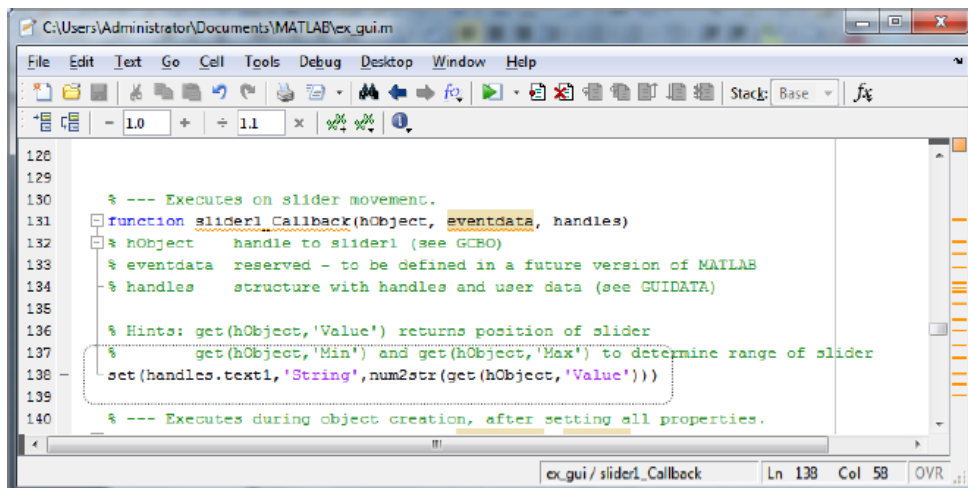


ภาพที่ 7.19 การใช้ Edit Text และ Static Text

7.4.4 การใช้ Slider Bar และ Static Text

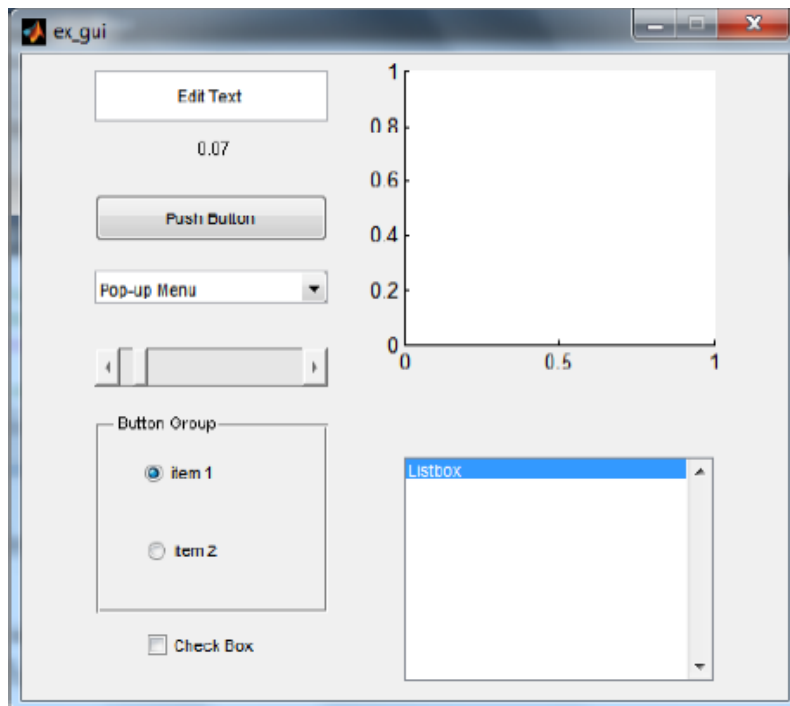
การเขียนโปรแกรมนี้ เมื่อเลื่อน Slider Bar แล้วจะปรากฏตัวเลขสเกลที่เลือกขึ้นใน Static Text ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับคือ Slider Bar (Slider1)

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Slider Bar ที่มี Tag เป็น Slider1 ใน GUIDE แล้วเลือก View Callbacks > Callback เหมือนกับข้อความที่แล้ว
2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.20



ภาพที่ 7.20 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Slider Bar

3. คลิกปุ่ม run หรือปุ่ม <F5> แล้วทดลองเลื่อน Slider Bar จะปรากฏข้อความใน Static Text เป็นตัวเลขตั้งแต่ค่า 0 ถึง 1 ขึ้นอยู่กับตำแหน่งของ Slider Bar ดังภาพที่ 7.21



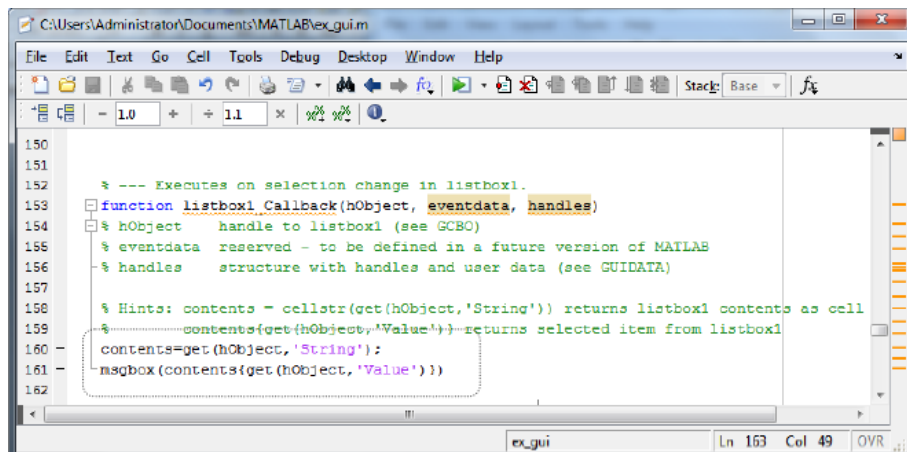
ภาพที่ 7.21 การใช้ Slider Bar และ Static Text

7.4.5 การใช้ List Box และ Pop-up Menu

การเขียนโปรแกรมนี้ เมื่อเลือกตัวเลือกภายใน List Box หรือ Pop-up Menu จะมีกล่องข้อความของตัวเลือกนั้นปรากฏขึ้นมา ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับ คือ List Box (listbox1) หรือ Pop-up Menu (popupmenu1) ซึ่งมีลักษณะการใช้งานและการเขียนโปรแกรมของทั้ง List Box หรือ Pop-up Menu มีลักษณะคล้ายกัน

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ List Box ที่มี Tag เป็น listbox1 ใน GUIDE แล้วเลือก View Callbacks > Callback เหมือนกับข้อความที่แล้ว

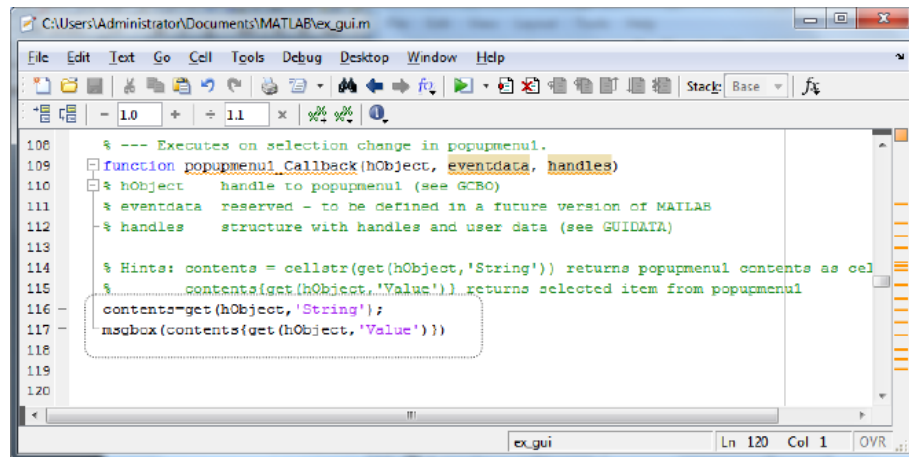
2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.22



ภาพที่ 7.22 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ List Box

3. ให้สร้างฟังก์ชันเรียกกลับ Pop-up Menu แบบเดียวกับ List Box ด้วยการคลิกขวาที่ Pop-up Menu ที่มี Tag เป็น popupmenu1 ใน GUIDE แล้วเลือก View Callbacks > Callback เหมือนกับข้อความที่แล้ว

4. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.23



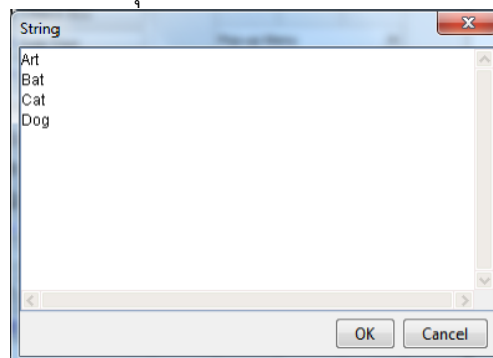
ภาพที่ 7.23 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Pop-up Menu

5. การเลือกข้อความเพิ่มเติมใน List Box และ Pop-up Menu โดยการดับเบิลคลิกที่วัตถุ นั้นในหน้าต่าง GUIDE จะปรากฏหน้าต่าง Inspector ให้ไปแก้ไขคุณสมบัติ String โดยการคลิกปุ่ม หลังชื่อคุณสมบัติแสดง ดังภาพที่ 7.24



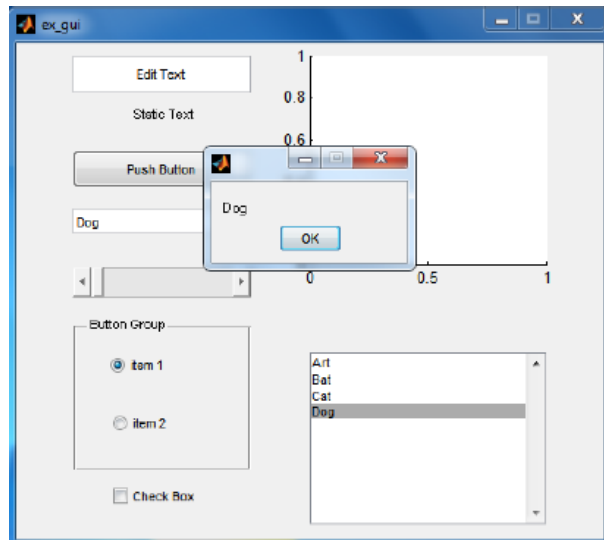
ภาพที่ 7.24 การแสดงการแก้ไขคุณสมบัติ String

6. ปรากฏหน้าต่าง String ขึ้นมาเพื่อแก้ไขตัวเลือก ให้แก้ไขโดยป้อนข้อความแต่ละตัวเลือก ซึ่งคั่นด้วยการขึ้นบรรทัดใหม่ แล้วคลิกปุ่ม OK ดังภาพที่ 7.25



ภาพที่ 7.25 หน้าต่าง String ขึ้นมาเพื่อแก้ไขตัวเลือก

7. ทำเช่นนี้กับ Pop-up Menu ด้วย จากนั้นทดลอง run โปรแกรมจากปุ่ม run ในหน้าต่าง GUIDE แล้วทดลองเลือกตัวเลือกใน List Box หรือ Pop-up Menu ดังภาพที่ 7.26

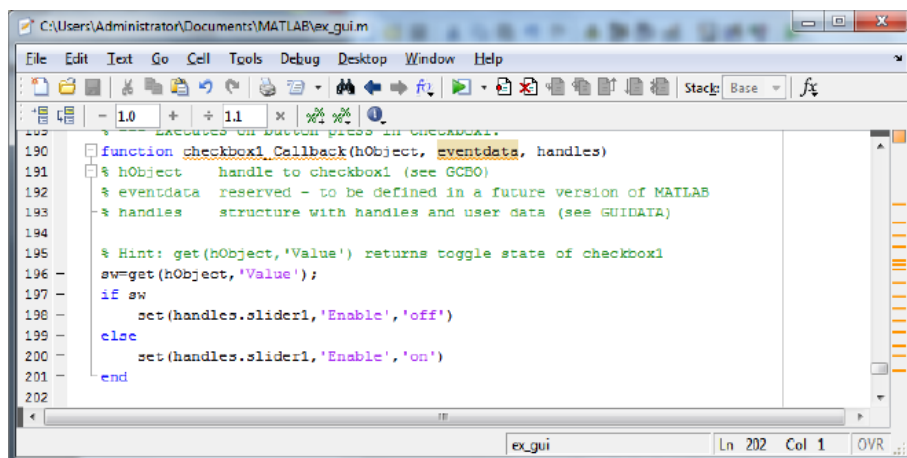


ภาพที่ 7.26 การใช้ List Box และ Pop-up Menu

7.4.6 การใช้ Check Box และ Slider Bar

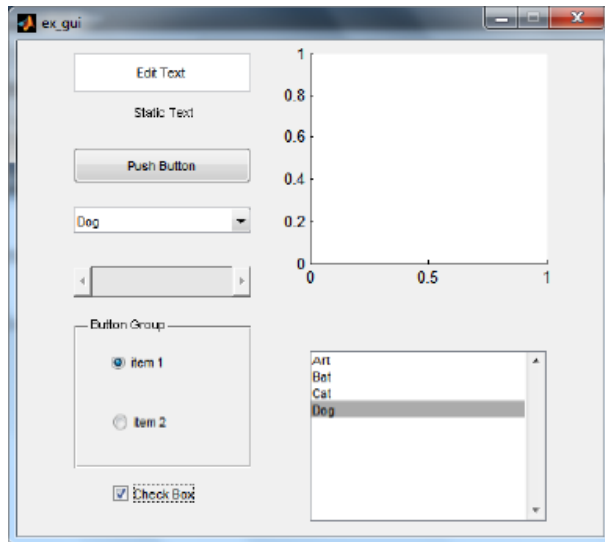
การเขียนโปรแกรมนี้ เมื่อคลิกที่ Check Box จะเป็นการเปิดหรือปิดการทำงานของ Slider Bar ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับที่ Check Box (checkbox1)

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Check Box ที่มี Tag เป็น checkbox1 ใน GUIDE แล้วเลือก View Callbacks > Callback เหมือนกับข้อความที่แล้ว
2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.27



ภาพที่ 7.27 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Check Box

3. คลิกปุ่ม run หรือปุ่ม <F5> แล้วทดลองเลื่อน Check Box จะถูกทำให้ไม่สามารถใช้งานได้ดังภาพที่ 7.28



ภาพที่ 7.28 การทดสอบการทำงานของ Check Box

7.5 การรับค่าจากแป้นพิมพ์ (Keyboard) และเมาส์ (Mouse)

7.5.1 การรับค่าจากแป้นพิมพ์

การรับค่าจากแป้นพิมพ์ สามารถแบ่งเป็นเหตุการณ์ตามฟังก์ชันเรียกกลับ คือ

KeyPressFcn ซึ่งจะถูกรับใช้เมื่อปุ่มบนแป้นพิมพ์ถูกกดลง และโฟกัสอยู่ที่หน้าต่าง Figure

WindowKeyPressFcn ซึ่งจะถูกรับใช้เมื่อปุ่มแป้นพิมพ์ถูกกดลง และโฟกัสอยู่ที่หน้าต่าง Figure หรือวัตถุใด ๆ ในหน้าต่าง Figure

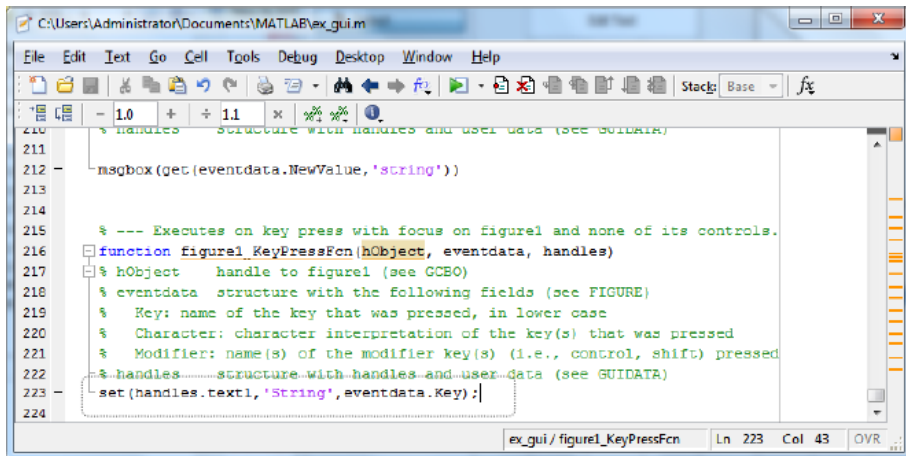
KeyReleaseFcn ซึ่งจะถูกรับใช้เมื่อปุ่มบนแป้นพิมพ์ถูกปล่อยขึ้น และโฟกัสอยู่ที่หน้าต่าง Figure หรือวัตถุใด ๆ ในหน้าต่าง Figure

WindowKeyReleaseFcn ซึ่งจะถูกรับใช้เมื่อปุ่มบนแป้นพิมพ์ถูกปล่อยขึ้น และโฟกัสอยู่ที่หน้าต่าง Figure หรือวัตถุใด ๆ ในหน้าต่าง Figure

การเขียนโปรแกรมนี้ เมื่อกดปุ่มบนแป้นพิมพ์จะปรากฏชื่อของปุ่มบนแป้นพิมพ์นั้นในช่อง Static Text ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับ คือ Figure (figure1)

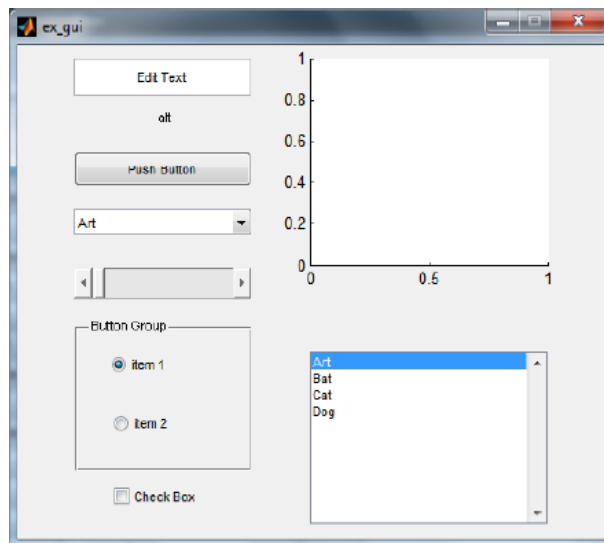
1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Figure ที่มี Tag เป็น figure1 ใน GUIDE แล้วเลือก View Callbacks > KeyPressFcn

2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.29



ภาพที่ 7.29 การเขียนคำสั่งเพื่อควบคุมฟังก์ชันเรียกกลับในวัตถุ Figure

3. คลิกปุ่ม run หรือปุ่ม <F5> แล้วทดลองกดปุ่มใด ๆ บนแป้นพิมพ์ เช่น ปุ่ม Alt จะพบคำว่า alt (ชื่อแป้นพิมพ์เป็นตัวพิมพ์เล็กทั้งหมด) ในช่อง Static Text ดังภาพที่ 7.30



ภาพที่ 7.30 ผลการรับค่าจากคีย์บอร์ด Alt จะแสดงที่ Static Text เป็น alt

7.5.2 การรับค่าจากเมาส์ (Mouse)

การรับค่าจากเมาส์ สามารถแบ่งเป็นเหตุการณ์ตามฟังก์ชันเรียกกลับคือ

WindowButtonDownFcn ซึ่งจะถูกรู้จักใช้เมื่อปุ่มใด ๆ บนเมาส์ถูกกดลง และโฟกัสอยู่ที่หน้าต่าง Figure หรือวัตถุใด ๆ ในหน้าต่าง Figure

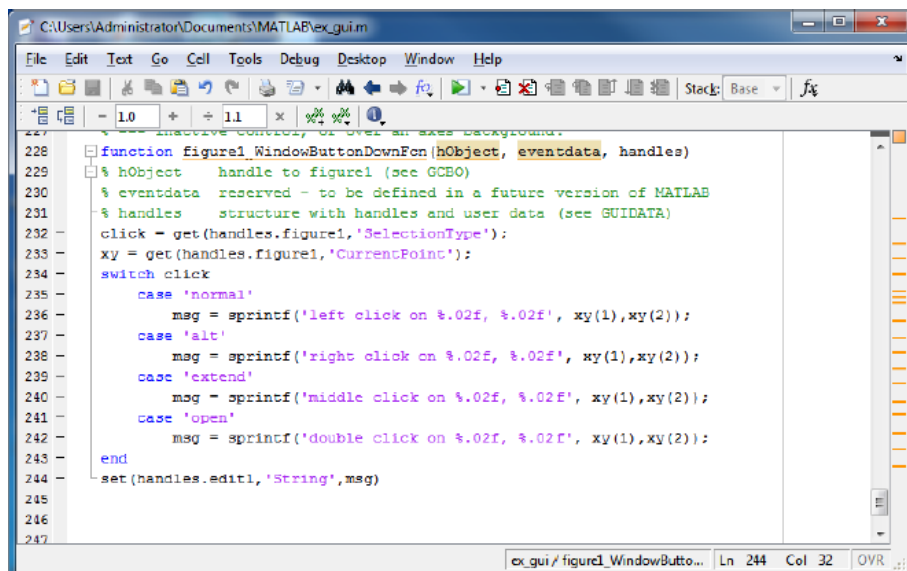
WindowButtonUpFcn ซึ่งจะถูกรู้จักใช้เมื่อปุ่มใด ๆ บนเมาส์ถูกปล่อยขึ้น และโฟกัสอยู่ที่หน้าต่าง Figure หรือวัตถุใด ๆ ในหน้าต่าง Figure

WindowButtonMotionFcn ซึ่งจะถูกรู้จักใช้เมื่อตัวชี้เมาส์เคลื่อนที่ผ่านบนหน้าต่าง Figure หรือวัตถุใด ๆ ในหน้าต่าง Figure

WindowScrollWheelFcn ซึ่งจะถูกรู้จักใช้เมื่อลูกกลิ้งของเมาส์ถูกหมุน และโฟกัสอยู่ที่หน้าต่าง Figure หรือวัตถุใด ๆ ในหน้าต่าง Figure

การเขียนโปรแกรมรับค่าจากเมาส์นี้ เมื่อคลิกเมาส์ปุ่มซ้าย ขวา หรือดับเบิลคลิก จะปรากฏข้อความบอกปุ่มที่ผู้ใช้กด และตำแหน่งที่กดปุ่มนั้นใน Edit Text ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับคือ Figure (figure1)

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Figure (พื้นหลังที่ว่าง ๆ) ที่มี Tag เป็น figure1 ใน GUIDE แล้วเลือก View Callbacks > WindowButtonDownFcn
2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.31

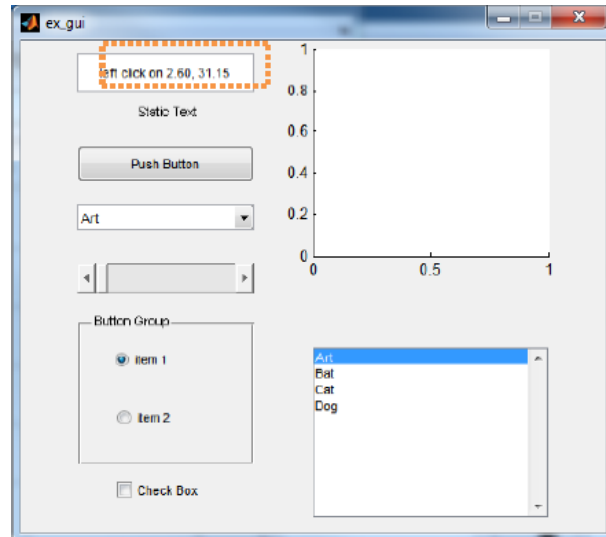


ภาพที่ 7.31 การเขียนใช้คำสั่ง WindowButtonDownFcn

คุณสมบัติ CurrentPoint ของ Figure หมายถึง คู่อันดับพิกัด (x,y) ของตัวชี้เมาส์

คุณสมบัติ SelectionType ของ Figure หมายถึง ปุ่มหรือวิธีการคลิก Normal หมายถึง คลิกซ้าย (กรณี Mouse แบบปกติ) Alt หมายถึง คลิกขวา (กรณี Mouse แบบปกติ) Extend หมายถึง คลิกปุ่มกลาง และ Open หมายถึง ดับเบิลคลิกปุ่มใดก็ได้

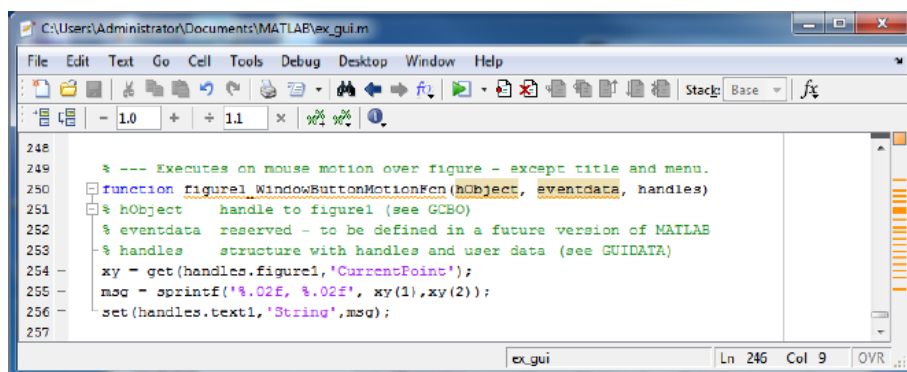
3. คลิกปุ่ม run หรือกดปุ่มเมาส์ เช่น คลิกปุ่มซ้าย ดังภาพที่ 7.32



ภาพที่ 7.32 ผลการรัน GUI ของการใช้คำสั่ง WindowButtonDownFcn

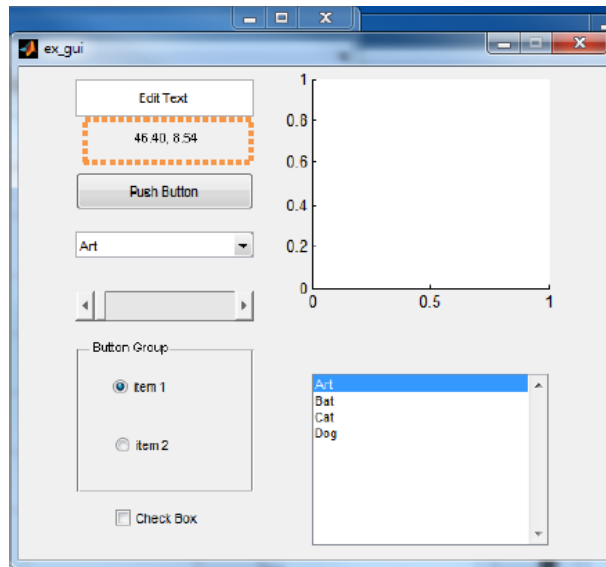
การเขียนโปรแกรมรับค่าจากเมาส์ โดยการบอกพิกัดเมาส์ใน Static Text เมื่อตัวชี้เมาส์เคลื่อนที่บนหน้าต่าง GUI ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับคือ Figure (figure1)

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Figure (พื้นหลังที่ว่าง ๆ) ที่มี Tag เป็น figure1 ใน GUIDE แล้วเลือก View Callbacks > WindowButtonMotionFcn
2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.33



ภาพที่ 7.33 การเขียนใช้คำสั่ง WindowButtonMotionFcn

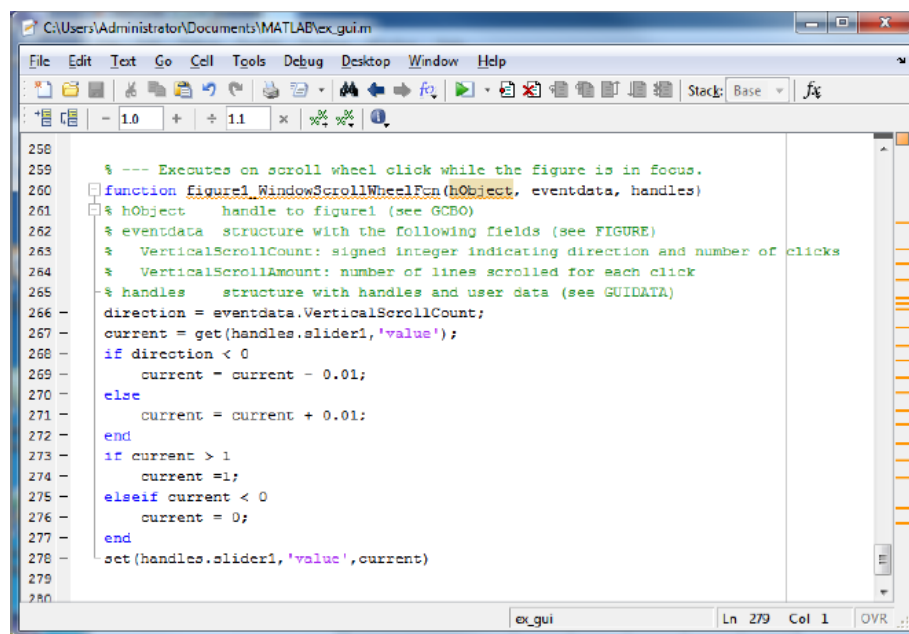
2. เมื่อกดปุ่ม run แล้ว ให้เลื่อนตัวชี้เมาส์ผ่านหน้าต่าง Figure จะปรากฏคู่อันดับ (x,y) ใน Static Text ดังภาพที่ 7.34



ภาพที่ 7.34 ผลการรัน GUI การใช้คำสั่ง WindowButtonMotionFcn

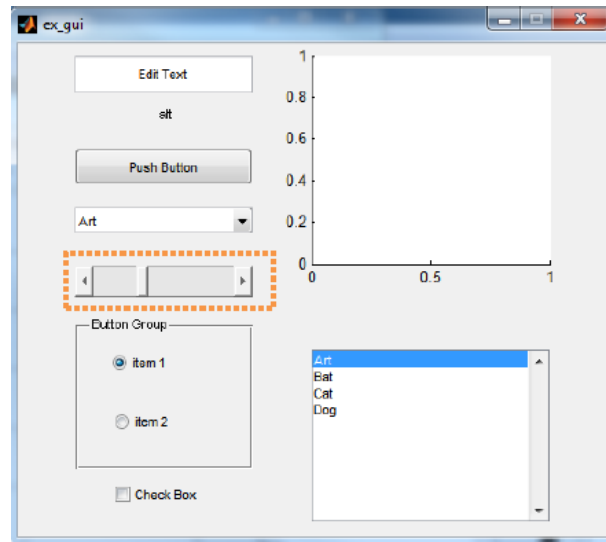
การเขียนโปรแกรมรับค่าจากเมาส์ โดยเมื่อหมุนลูกกลิ้งเมาส์จะทำการไปเลื่อน Slider Bar ไปทางซ้ายหรือขวาตามการหมุน ดังนั้น วัตถุที่ใช้ในการเรียกฟังก์ชันเรียกกลับคือ Figure (figure1)

1. ให้สร้างฟังก์ชันเรียกกลับด้วยการคลิกขวาที่ Figure (พื้นหลังที่ว่าง ๆ) ที่มี Tag เป็น figure1 ใน GUIDE แล้วเลือก View Callbacks > WindowScrollWheelFcn
2. ทำการแก้ไขโปรแกรม ดังภาพที่ 7.35



ภาพที่ 7.35 การเขียนใช้คำสั่ง WindowScrollWheelFcn

3. เมื่อกดปุ่ม run แล้ว ให้ทดลองหมุนลูกกลิ้งเมาส์ จะพบว่า Slider Bar เลื่อนตามที่หมุน ดังภาพที่ 7.36

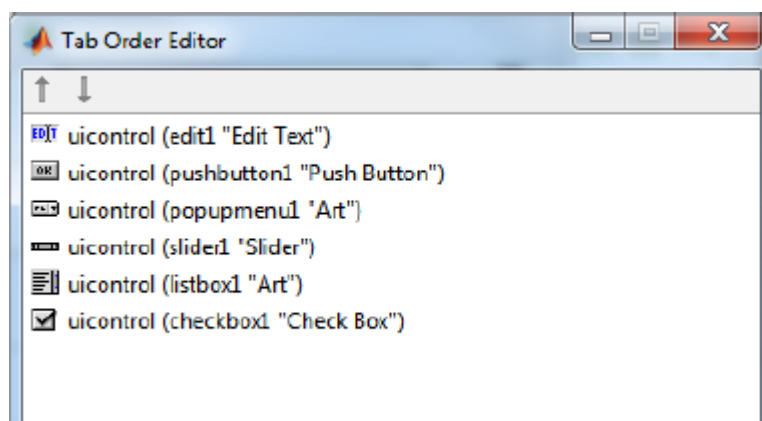


ภาพที่ 7.36 ผลการรัน GUI การใช้คำสั่ง WindowScrollWheelFcn

7.6 การจัดลำดับ Tab และแถบเมนูเครื่องมือ

7.6.1 การจัดลำดับ Tab

หากต้องการใช้แป้นพิมพ์ (ปุ่ม Tab) ในการเลือกวัตถุแต่ละปุ่มโดยการกดปุ่ม <Tab> แต่ละครั้งจะมีลำดับ ซึ่งสามารถแก้ไขได้โดยคลิกปุ่ม  บริเวณแถบเครื่องมือด้านบนในหน้าต่าง GUIDE และเมื่อคลิกแล้วจะปรากฏหน้าต่าง Tab Order Editor ดังภาพที่ 7.37



ภาพที่ 7.37 หน้าต่าง Tab Order Editor

หน้าต่างนี้สามารถเลื่อนลำดับได้ด้วยปุ่มลูกศรขึ้นหรือลง โดยวัตถุที่อยู่บนสุดจะเป็นอันดับแรกและเรียงถัดลงมาเรื่อย ๆ

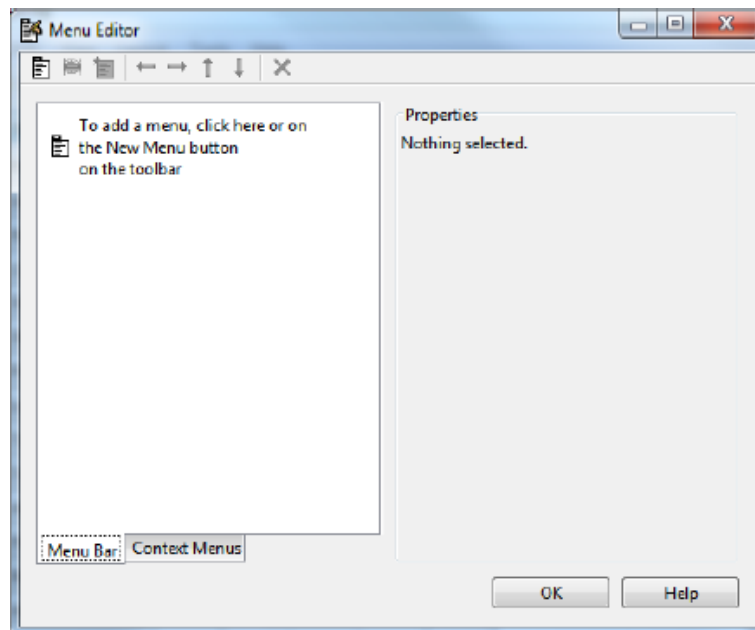
7.6.2 เมนูแถบเครื่องมือ

เมนูแบบข้อความแบ่งออกเป็น 2 ประเภท คือ แบบที่ปรากฏอยู่ในแถบเมนูในหน้าต่าง Figure (Menu Bar) และแบบที่จะปรากฏเมื่อมีการคลิกขวาในแต่ละวัตถุ (Context Menu) และเมนูประเภทที่เป็นรูปไอคอนหรือแถบเครื่องมือ (Toolbar)

- แถบเมนู (Menu Bar)

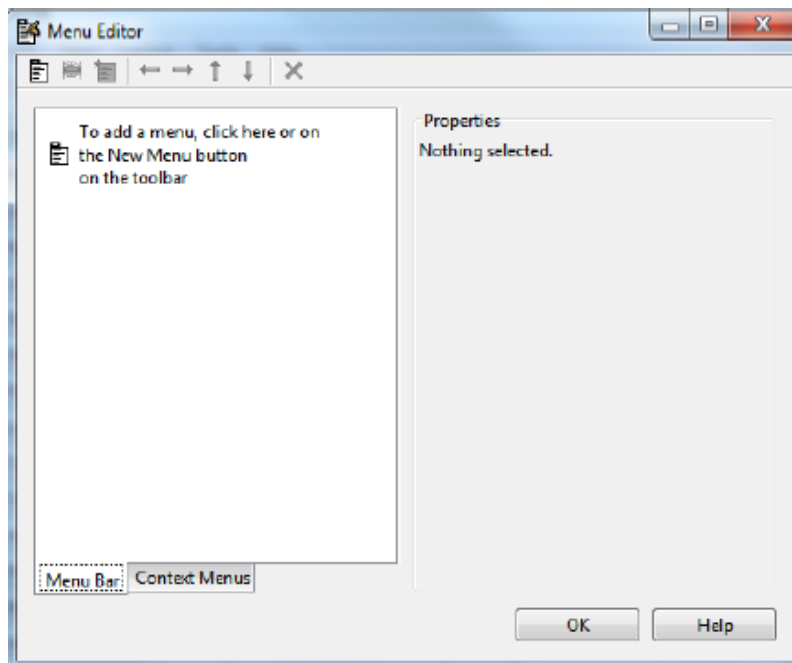
ก่อนอื่นใช้ GUIDE สร้าง GUI แบบว่างเปล่าขึ้นมาก่อน โดยเลือก Black GUI (Default) ในหน้าต่าง GUIDE Quick Start

1. การสร้างเมนู GUIDE ทำได้โดยการคลิกปุ่ม  บริเวณแถบเครื่องมือด้านบนในหน้าต่าง GUIDE เมื่อคลิกแล้วจะปรากฏหน้าต่าง Menu Editor ดังภาพที่ 7.38



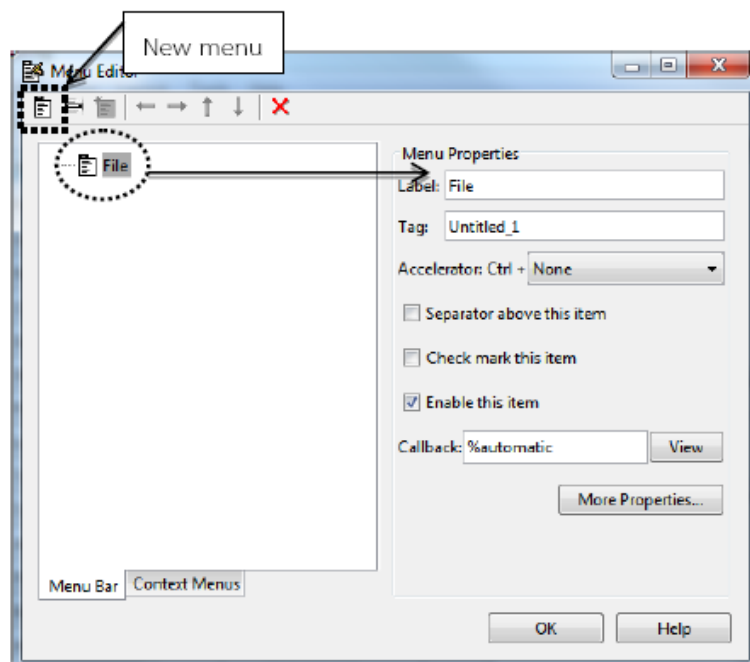
ภาพที่ 7.38 หน้าต่าง Menu Editor ในส่วนของ Menu Bar

2. เมื่อต้องการสร้างเมนูใหม่ให้คลิกปุ่ม New Menu ในกรอบสี่เหลี่ยมเส้นประจะปรากฏเมนูใหม่ชื่อ Untitled1 ในกรอบด้านล่างให้คลิกเมนูใหม่ที่สร้างขึ้น เพื่อแก้ไขค่าคุณสมบัติ Label เป็น File ดังภาพที่ 7.39



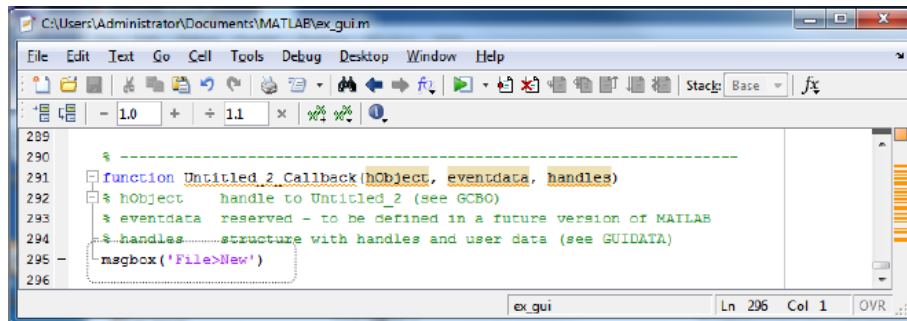
ภาพที่ 7.39 การสร้างเมนู File

3. เมื่อต้องการสร้างเมนูย่อยให้กดปุ่ม New Menu Item ในวงกลมเส้นประจะปรากฏเมนูย่อยใหม่ ในกรอบด้านล่างให้คลิกเมนูใหม่ที่สร้างขึ้น เพื่อแก้ไขค่าคุณสมบัติ Label เป็น File ดังภาพที่ 7.40



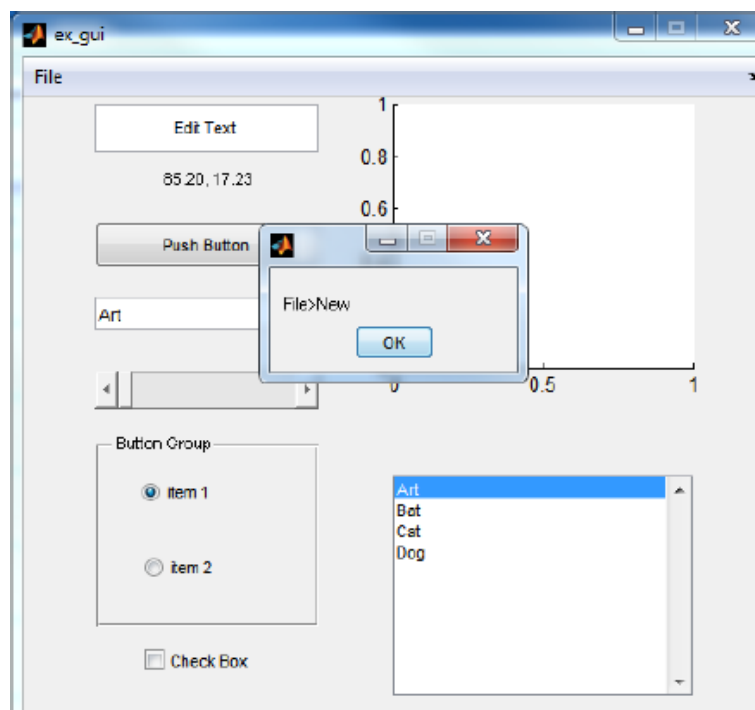
ภาพที่ 7.40 การสร้างเมนูย่อย New

4. เมื่อต้องการเขียนโปรแกรมให้ทำงาน เมื่อมีการกดเมื่อนั้น ๆ ทำได้โดยกดปุ่ม View ใน วงกลมเส้นทึบจะปรากฏฟังก์ชันนี้ แล้วให้เขียนคำสั่ง ดังภาพที่ 7.41



ภาพที่ 7.41 การเขียนคำสั่งเพื่อให้เมนูย่อย New ทำงาน

เมื่อสั่งรันและเลือกเมนู New จะปรากฏข้อความว่า File > New ดังภาพที่ 7.42

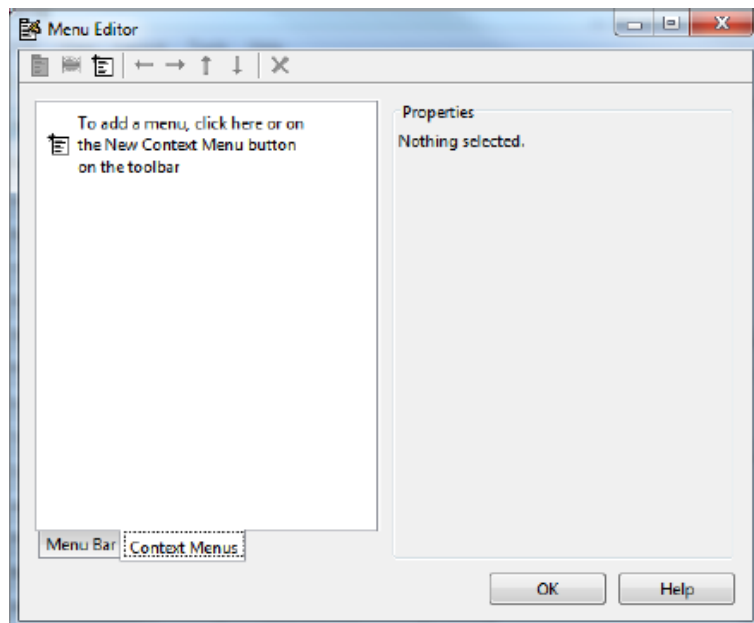


ภาพที่ 7.42 ผลการรัน GUI แล้วเลือก File > New ที่เมนูบาร์

- Context Menus

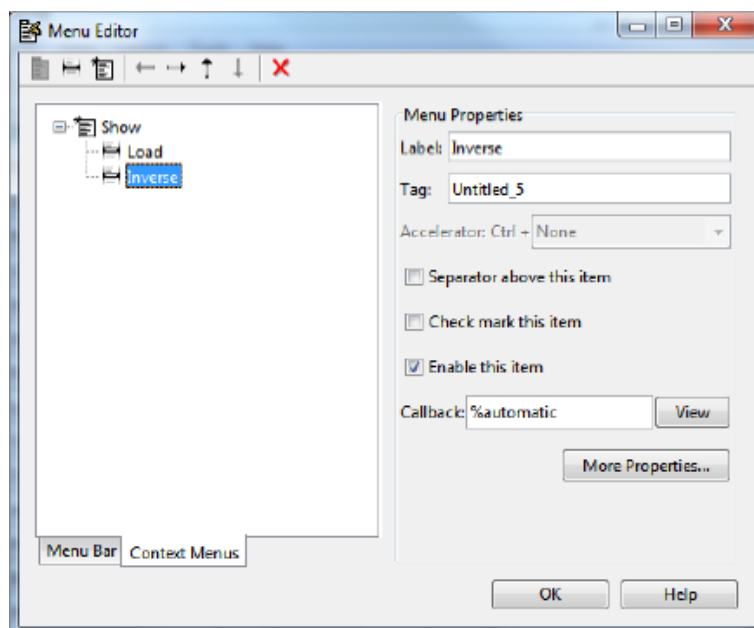
การสร้าง Context Menus ทำได้ในหน้าต่าง Menu Editor เช่นเดียวกับการสร้างแถบเมนู

1. ให้คลิกที่ Context Menus ดังภาพที่ 7.43



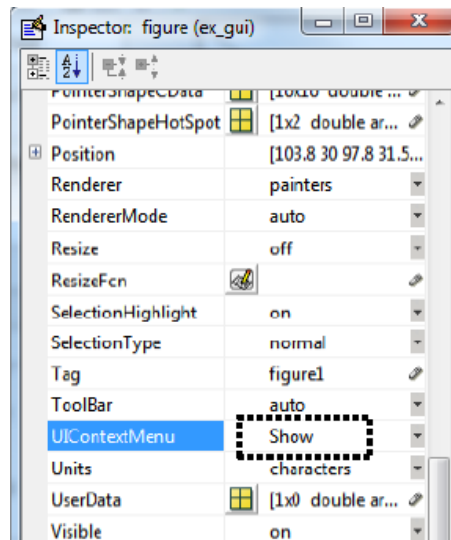
ภาพที่ 7.43 หน้าต่าง Menu Editor ในส่วนของ Context Menus

2. สามารถสร้างเมนูและแก้ไขคุณสมบัติ Label ให้มีลักษณะต่าง ๆ ดังภาพที่ 7.44



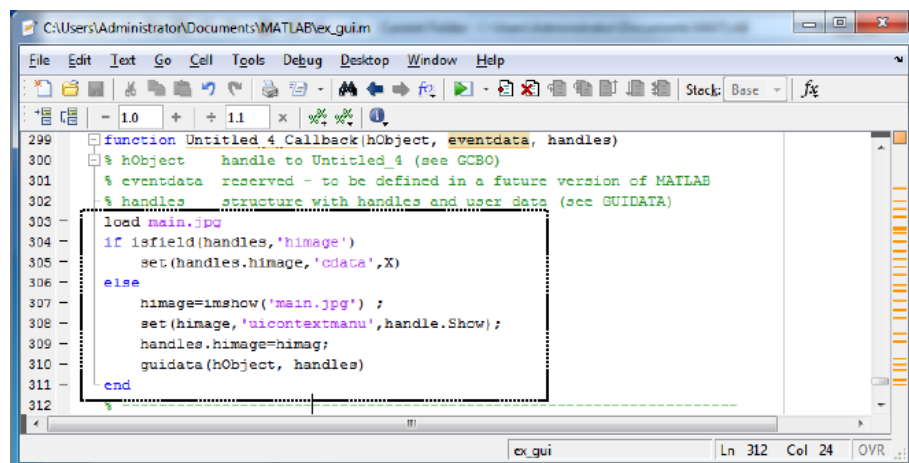
ภาพที่ 7.44 การสร้างเมนู Show, Load และ Inverse

3. การเพิ่ม Context Menus เข้าไปในหน้าต่าง Figure โดยการแก้ไขคุณสมบัติ UI Context Menus ในหน้าต่าง Inspector (ดับเบิลคลิกที่ Figure1 ใน GUIDE เพื่อเปิด) เป็น Show ชื่อ Tag ของ Context Menus ที่สร้างขึ้น ดังภาพที่ 7.45



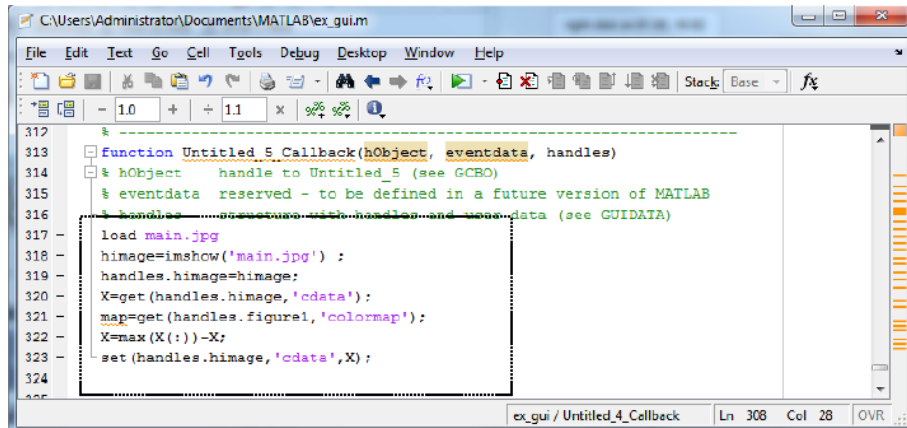
ภาพที่ 7.45 หน้าต่าง Inspector : figure(ex_gui)

4. เมนู Load ให้คลิกปุ่ม View บริเวณ Callback เพื่อแก้ไขโปรแกรม ดังภาพที่ 7.46



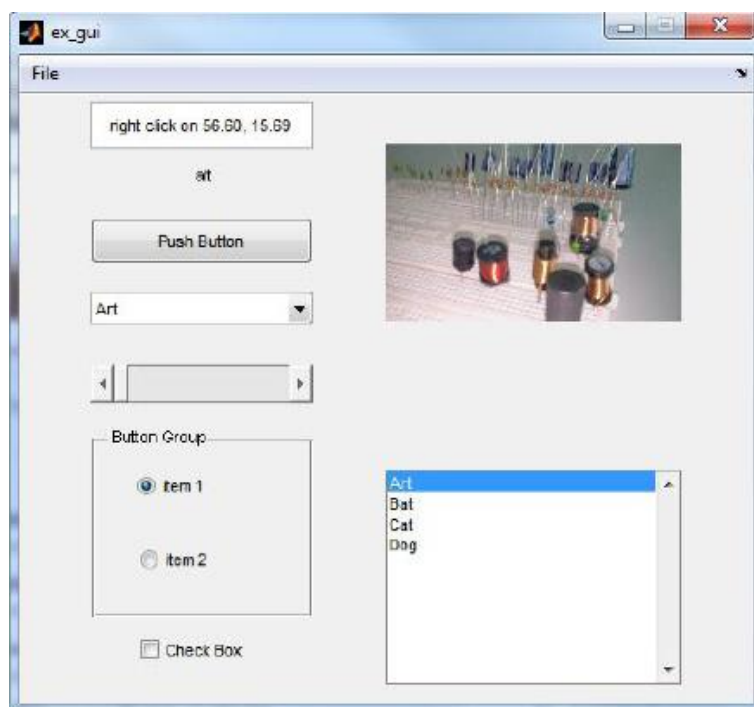
ภาพที่ 7.46 การเขียนคำสั่งสำหรับเมนู Load

5. เมนู Inverse ให้คลิกปุ่ม View บริเวณ Callback เพื่อแก้ไขโปรแกรม ดังภาพที่ 7.47



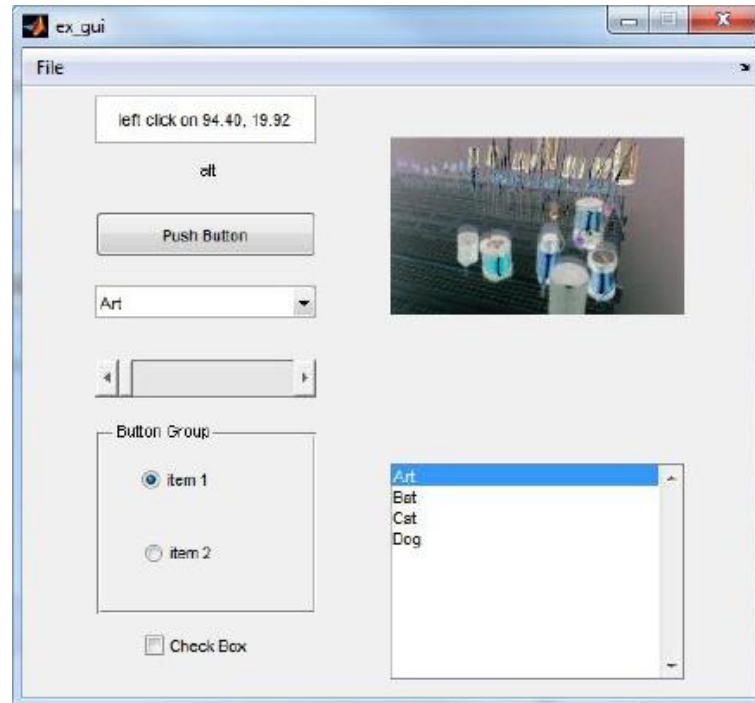
ภาพที่ 7.47 การเขียนคำสั่งสำหรับเมนู Inverse

6. เมื่อสั่งรันแล้ว ทดลองคลิกขวาภายในหน้าต่าง Figure แล้วเลือก Load จะได้ผล ดังภาพที่ 7.48



ภาพที่ 7.48 การทดลองคลิกขวาภายในหน้าต่าง Figure แล้วเลือก Load

เมื่อทดลองคลิกขวาแล้วเลือก Inverse จะได้ภาพแบบลบ (Negative) ดังภาพที่ 7.49



ภาพที่ 7.49 การทดลองคลิกขวาภายในหน้าต่าง Figure แล้วเลือก Inverse

7.7 สรุป

โปรแกรม MATLAB จะสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface : GUI) GUI อยู่บนหน้าต่างรูปภาพ (Figure Window) ซึ่งภายใต้หน้าต่างนี้จะมีส่วนประกอบต่าง ๆ อยู่ได้ไม่ว่าจะเป็น Axes, Uicontrol หรือวัตถุอื่น ๆ เราสามารถสร้าง Uicontrol, Uimenu แบบต่าง ๆ ลงในหน้าต่างรูปภาพได้แต่เป็นไปด้วยความลำบากเพราะการสร้างเป็น Text Base ต่อมาจนกระทั่ง ปัจจุบัน MATLAB ได้สร้าง Graphical User Interface Development Environment หรือ GUIDE ขึ้นเพื่อช่วยให้สร้างบันทึก และแก้ไข GUI ได้สะดวกขึ้น การสร้าง GUI จะประกอบด้วย 2 ขั้นตอนดังนี้

- 1) กำหนดและวางส่วนประกอบต่าง ๆ ลงบน GUI
- 2) เขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนต่าง ๆ ใน GUI

การเขียน GUI ในโปรแกรม MATLAB มีหลักการสำคัญอยู่ที่ตัวแปร **handles** เพื่อใช้เป็นตัวอ้างอิงหรือชี้ไปที่วัตถุแต่ละวัตถุ ค่าคุณสมบัติ (Property) ต่าง ๆ ของแต่ละวัตถุ และฟังก์ชันเรียกกลับ (Callback Function) ซึ่งมีหลักการเดียวกับการแก้ไขหน้าต่าง figure

ตัวแปร handles เป็นตัวแปรที่ใช้เก็บเลขจำนวนหนึ่งที่ใช้เป็นตัวอ้างอิงถึงวัตถุแต่ละวัตถุ คล้ายกับ Binder ในภาษา C โดยหากสร้าง GUI ด้วย GUIDE ตัวแปร handles จะถูกสร้างให้เองสำหรับทุก ๆ วัตถุที่สร้างขึ้น และบรรจุไว้ในตัวแปรชื่อ handles แล้วใช้ Tag ในการเรียกใช้ ซึ่งการ

เขียน GUI ด้วย GUIDE มีเป้าหมายเพื่อการทำงานของวัตถุส่วนติดต่อผู้ใช้ทั้งหมด โดยการสร้างให้แต่ละวัตถุมีปฏิสัมพันธ์ติดต่อกัน เช่น การใช้ Push Button และ Axes, การใช้ Radio Button และ Button Group, การใช้ Edit Text และ Static Text, การใช้ Slider Bar และ Static Text, การใช้ List Box และ Pop-up Menu, การใช้ Check Box และ Slider Bar เป็นต้น

แบบฝึกหัด

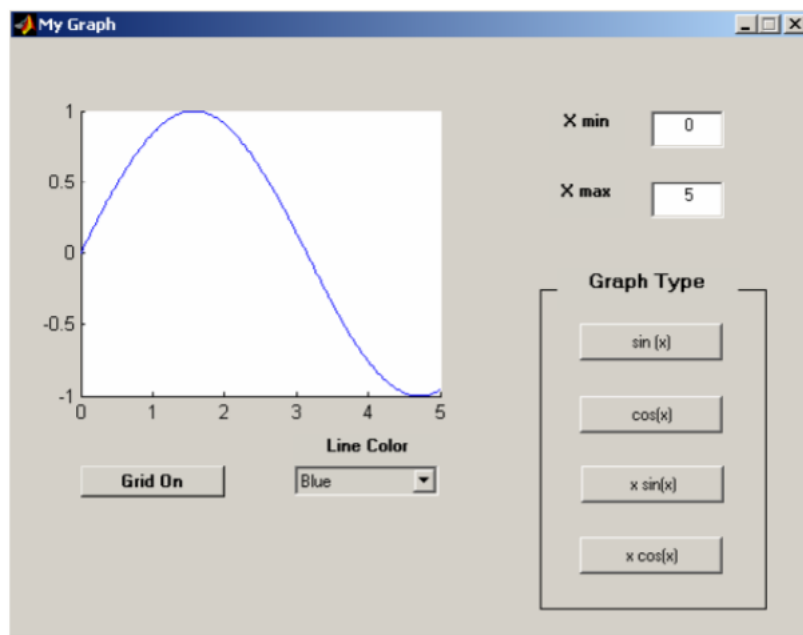
บทที่ 7 ส่วนต่อประสานกราฟฟิคกับผู้ใช้ของโปรแกรม MATLAB

1. จงอธิบายการเรียกใช้งาน GUI ของโปรแกรม MATLAB
2. จงอธิบายหลักการเขียนโปรแกรมและ GUI ด้วย GUIDE ของโปรแกรม MATLAB มาพอสังเขป
3. จงอธิบายการใช้งานวัตถุต่าง ๆ ที่นำมาสร้าง GUI ของโปรแกรม MATLAB
4. จงอธิบายการรับค่าจากแป้นพิมพ์และเมาส์ของโปรแกรม MATLAB
5. จงอธิบายคุณสมบัติการทำงานของ Check Box, List Box และ Pop-up Menu มาพอสังเขป
6. จงออกแบบส่วนต่อประสานกราฟฟิคของการใช้งาน Pop-up Menu โดยกำหนดให้มี 3 เมนู ได้แก่ Black, Gray, White
7. จงออกแบบส่วนต่อประสานกราฟฟิคของการใช้งาน Edit Text และ Static Text โดยกำหนดให้เมื่อใส่ข้อความลงใน Edit Text แล้ว ให้แสดงข้อความใน Static Text เหมือนกัน
8. จงออกแบบโปรแกรมแสดงสีตามที่ user เลือก ซึ่งมีทั้งหมด 3 สีคือ Red, Green, Blue โดยจะใช้ Radio button เพื่อเลือกทีละสี และใช้ Check box เพื่อเลือกหลายสี และแสดงค่าแบบผสมกัน (ผสมสี)
9. จงออกแบบโปรแกรมแปลงภาพสีเป็นภาพสีเทา โดยกำหนดเงื่อนไข ดังนี้
 - 9.1 ให้ user เลือกรูปภาพ หรือ เลือกไฟล์เดอร์ที่เก็บภาพ ในกรณีที่ user เลือกภาพเดียว โปรแกรมก็จะแปลงและ save ภาพเดียว แต่ถ้า user เลือกไฟล์เดอร์ โปรแกรมก็จะแปลงภาพทั้งหมดในไฟล์เดอร์นั้น และ save ภาพทั้งหมด
 - 9.2 แสดงภาพตัวอย่างที่ user เลือกในหน้า GUI
 - 9.3 สร้างปุ่ม convert ภาพ เพื่อให้ user ดูตัวอย่างภาพก่อน save ได้
 - 9.4 สร้างปุ่ม save ภาพ ถ้ามีภาพเดียวจะ save ตามชื่อที่ user ตั้ง แต่ถ้ามีหลายภาพจะเพิ่มตัวเลขลำดับต่อท้ายจากชื่อที่ user ตั้งเอาไว้
10. จงออกแบบโปรแกรมโดยการใช้ slider ไปปรับค่ารัศมีของวงกลม หลักการเหมือนเดิมครับ ค่าที่รับมาจาก slider หรือ edit text ถูกนำไปให้ตัวแปรที่ใช้กำหนดรัศมีของวงกลม

11. จงออกแบบโปรแกรม โดยมีเงื่อนไข ดังนี้

- 11.1 ผู้ใช้สามารถเลือกรายงาน Detail อย่างน้อย 2 อย่าง จาก Pop-up Menu
- 11.2 ผู้ใช้สามารถเลือกจะใช้ผลิตภัณฑ์นั้นที่ใด คือ at work , at home, at bus
- 11.3 เมื่อเลือกรายการอะไรและผลิตภัณฑ์ใดแล้ว ให้แสดงผลด้วย Static Text ข้างล่าง

12. จงสร้าง GUI แสดงผลการทำงานของคำสั่งตรีโกณมิติ ดังนี้



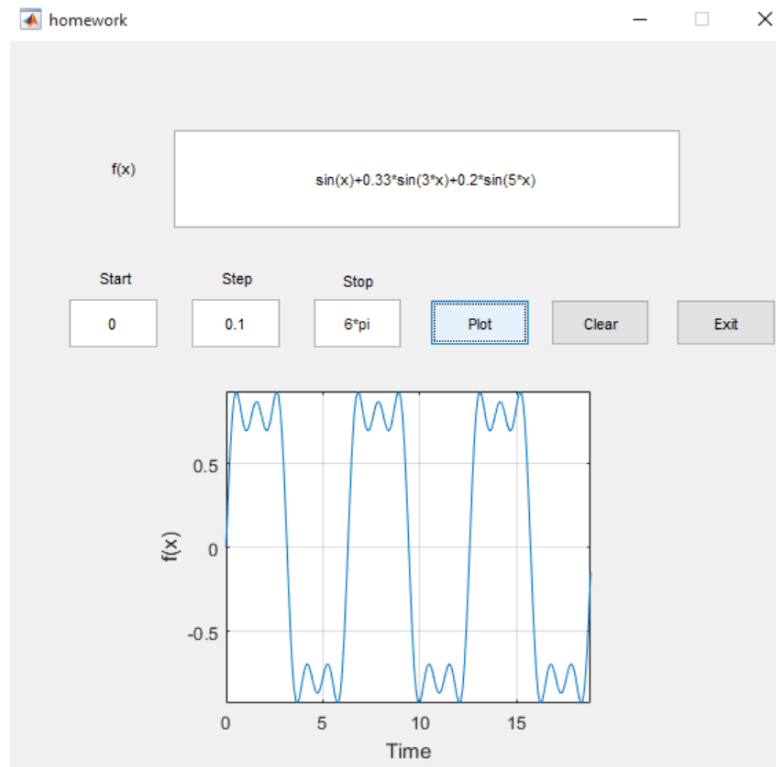
13. จงสร้าง GUI ของโปรแกรม MATLAB สำหรับคำนวณหาค่าวงจรไฟฟ้า โดยมีเงื่อนไขดังต่อไปนี้

- 13.1 กำหนดหน้าต่างของโปรแกรม จะต้องรับค่าความต้านทาน 5 ตัว
- 13.2 สามารถคำนวณหาค่าวงจรไฟฟ้าแบบอนุกรม วงจรไฟฟ้าแบบขนาน และวงจรไฟฟ้าแบบผสม
- 13.3 มีการแสดงภาพสูตรการคำนวณวงจรไฟฟ้า
- 13.4 มีปุ่มกดเพื่อคำนวณ และมีการแสดงผลลัพธ์ที่เป็นค่าคงที่

14. จงสร้าง GUI ของโปรแกรม MATLAB สำหรับคำนวณค่ากำลังงานไฟฟ้า โดยมีเงื่อนไขดังต่อไปนี้

- 14.1 กำหนดหน้าต่างของโปรแกรม จะต้องรับค่าแรงดัน (E) และค่ากระแส (I)
- 14.2 มีการแสดงภาพสูตรกำลังไฟฟ้า
- 14.3 มีปุ่มกดเพื่อคำนวณ และมีการแสดงผลลัพธ์ที่เป็นค่าคงที่

15. จงเขียนโปรแกรมแสดงผลดังนี้ จากสมการ $f(x) = \sin(x) + 0.33 \sin(3x) + 0.2 \sin(5x)$



บทที่ 8

การประยุกต์ใช้ GUI ของ MATLAB สำหรับการคำนวณสมการกฎของโอห์ม

โปรแกรม MATLAB จะสร้าง GUI อยู่บนหน้าต่างรูปภาพ (Figure Window) ซึ่งภายใต้หน้าต่างนี้จะมีส่วนประกอบต่าง ๆ อยู่ได้ไม่ว่าจะเป็น Axes, Uicontrol หรือวัตถุอื่น ๆ เราสามารถสร้าง Uicontrol, Uimenu แบบต่าง ๆ ลงในหน้าต่างรูปภาพได้แต่เป็นไปด้วยความลำบากเพราะการสร้างเป็น Text Base ต่อมาจนกระทั่ง ปัจจุบัน MATLAB ได้สร้าง Graphical User Interface Development Environment หรือ GUIDE ขึ้นเพื่อช่วยให้สร้างบันทึก และแก้ไข GUI ได้สะดวกขึ้น การสร้าง GUI จะประกอบด้วย 2 ขั้นตอนดังนี้

- 3) กำหนดและวางส่วนประกอบต่าง ๆ ลงบน GUI
- 4) เขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนต่าง ๆ ใน GUI

ซึ่ง GUIDE โดยหลักใหญ่จะมีหน้าที่ในการวางส่วนประกอบที่ต้องการให้มีลงใน GUI จากนั้น GUIDE จะสร้าง M-file ที่บรรจุ Handle ของวัตถุหรือ Object ทั้งหมดที่สร้างขึ้นรวมทั้งคำสั่ง GUI ทำงาน นอกจากนั้น M-file จะให้แนวทางในการเขียนฟังก์ชัน ที่ทำงานหลังจากผู้ใช้กดเมาส์ปุ่มซ้ายหรือปรับเปลี่ยนค่าของวัตถุนั้น เรียกว่า Callback ของวัตถุนั้น ปัจจุบันได้มีการนำ GUI ของโปรแกรม MATLAB มาประยุกต์ในการพัฒนาโปรแกรมคำนวณทางคณิตศาสตร์มากมาย อาทิเช่น โปรแกรมคำนวณหาค่าไฟฟ้า โปรแกรมจำลองการวิเคราะห์ค่าพารามิเตอร์ของสายอากาศ โปรแกรมจำลองค่าพารามิเตอร์สายส่งความถี่สูงโดยใช้ Matlab/Simulink โปรแกรมจำลองสำหรับศึกษาและวิเคราะห์วงจรกรองความถี่ภายในท่อนำคลื่นด้วยวิธีการวนรอบของคลื่น เป็นต้น

สำหรับบทนี้เป็นการพัฒนาโปรแกรมการคำนวณสมการกฎของโอห์ม โดยการประยุกต์ใช้ GUI ของโปรแกรม MATLAB ซึ่งเป็นสมการนี้เป็นสมการพื้นฐานที่ใช้แสดงความสัมพันธ์ระหว่างแรงดันไฟฟ้า กระแสไฟฟ้าและความต้านทาน โดยเริ่มตั้งแต่วิธีการดำเนินการพัฒนา GUI การเขียนโปรแกรมควบคุมการทำงานของวัตถุที่หน้าต่างเมนูหลัก และการเขียนโปรแกรมเพื่อควบคุมวัตถุเพื่อเชื่อมโยงไปยังโปรแกรมย่อย

8.1 วิธีการดำเนินการพัฒนา GUI

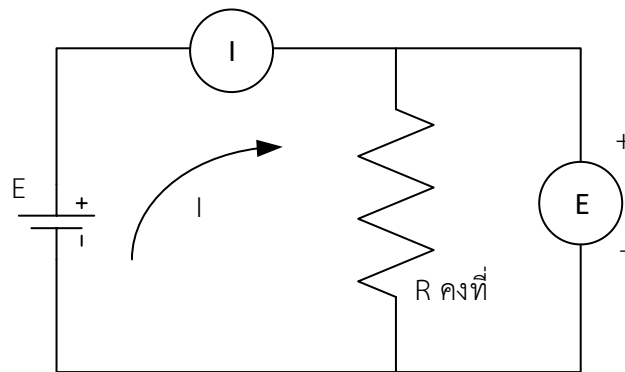
8.1 การวิเคราะห์งาน

การวิเคราะห์งานเกี่ยวกับกฎของโอห์ม ใช้อธิบายความสัมพันธ์ระหว่างกระแสไฟฟ้า (I) แรงดันไฟฟ้า (E) และความต้านทาน (R) ซึ่งเป็นเนื้อหาที่พื้นฐานที่ช่างไฟฟ้าและอิเล็กทรอนิกส์ใช้ในการคำนวณหาค่ากระแสไฟฟ้า ค่าแรงดันไฟฟ้า และค่าความต้านทานภายในวงจร ที่สามารถนำมาพัฒนาเป็นโปรแกรมจำลองสำหรับใช้ในการศึกษาเพื่อเชื่อมโยงไปสู่เนื้อหาในส่วนของปฏิบัติ ได้แก่ การคำนวณหาค่ากระแสไฟฟ้า (I) การคำนวณหาค่าแรงดันไฟฟ้า (E) และการคำนวณหาค่าความต้านทาน (R) ซึ่งมีรายละเอียดดังต่อไปนี้

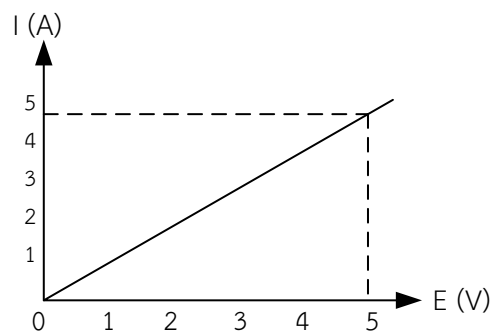
1. การคำนวณหาค่ากระแสไฟฟ้า (I)

เป็นความสัมพันธ์ระหว่างแรงดันไฟฟ้า (E) และความต้านทาน (R) ตามการศึกษากฎของโอห์ม สามารถทำได้ 2 กรณี คือ

กรณีที่ 1 เมื่อความต้านทาน (R) คงที่ และเพิ่มค่าแรงดันไฟฟ้าให้สูงขึ้น เป็นสองเท่า ส่งผลให้กระแสไฟฟ้า (I) เพิ่มขึ้นเป็นสองเท่าด้วย แต่ถ้าแรงดันไฟฟ้าถูกลดลงครึ่งหนึ่ง กระแสไฟฟ้าจะลดลงครึ่งหนึ่งเช่นกัน ดังภาพที่ 8.1 และภาพที่ 8.2



ภาพที่ 8.1 การต่อวงจรเมื่อความต้านทานคงที่

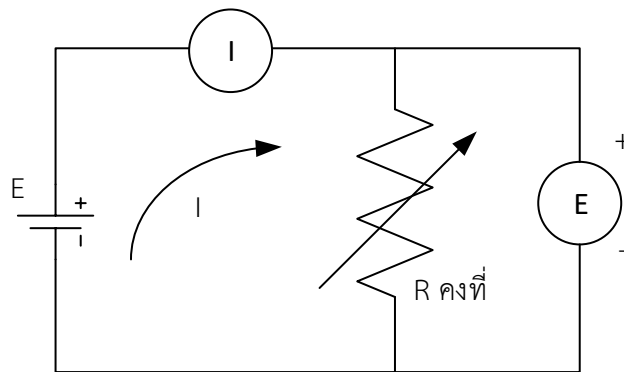


ภาพที่ 8.2 ความสัมพันธ์ระหว่างแรงดันไฟฟ้าและกระแสไฟฟ้า

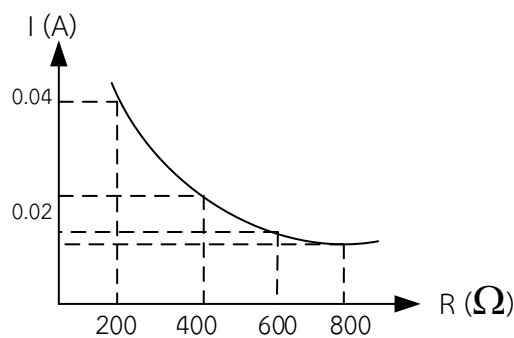
จากภาพที่ 8.2 พบว่า เมื่อแรงดันไฟฟ้า (E) เพิ่มขึ้นจะทำให้กระแสไฟฟ้าเพิ่มขึ้นด้วย เมื่อความต้านทานคงที่ (R) จึงกล่าวได้ว่ากระแสไฟฟ้าจะแปรผันเป็นสัดส่วนโดยตรงกับแรงดันไฟฟ้า ดังสมการที่ (1)

$$I \sim E \quad (1)$$

กรณีที่ 2 ทำนองเดียวกัน ถ้านำความต้านทานที่ปรับค่าได้มาต่อเข้ากับแหล่งจ่ายแรงดันไฟฟ้าที่มีค่าแรงดันไฟฟ้าคงที่ แล้ววัดค่ากระแสไฟฟ้าที่ไหลในวงจร พบว่า เมื่อค่าความต้านทานมีค่าสูงขึ้น กระแสไฟฟ้าจะไหลผ่านได้น้อยลง แต่ถ้าความต้านทานมีค่าน้อย กระแสไฟฟ้าจะไหลผ่านได้มากขึ้น ดังภาพที่ 8.3



ภาพที่ 8.3 การต่อวงจรเมื่อความต้านทานไม่คงที่



ภาพที่ 8.4 ความสัมพันธ์ระหว่างกระแสไฟฟ้าและความต้านทานที่ไม่คงที่

จากภาพที่ 8.4 พบว่า เมื่อความต้านทาน (R) เพิ่มขึ้นทำให้กระแสไฟฟ้า (I) ลดลง เมื่อแรงดันไฟฟ้า (E) คงที่ ดังนั้นจึงกล่าวได้ว่า “กระแสไฟฟ้าจะแปรผันเป็นส่วนกลับหรือแปรผกผันกับความต้านทาน” ดังสมการที่ (2)

$$I \sim \frac{1}{R} \quad (2)$$

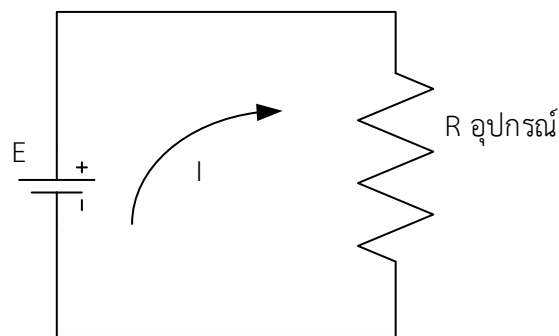
จากความสัมพันธ์ของกระแสไฟฟ้า แรงดันไฟฟ้า และความต้านทานทั้ง 2 กรณี ทำให้ทราบว่ากระแสไฟฟ้า (I) เปลี่ยนแปลงเป็นสัดส่วนโดยตรงกับแรงดันไฟฟ้าและกระแสไฟฟ้าจะเป็นสัดส่วนกลับกับความต้านทาน (R) ดังนั้น จึงสามารถสรุปเป็นกฎของโอห์ม คือ กระแสไฟฟ้าจะแปรผันตรงกับแรงดันไฟฟ้า และจะแปรผกผันกับความต้านทาน ดังสมการที่ (3)

$$I = \frac{E}{R} \quad (3)$$

เมื่อ	I	คือ	กระแสไฟฟ้า	หน่วย	แอมแปร์ (A)
	E	คือ	แรงดันไฟฟ้า	หน่วย	โวลต์ (V)
	R	คือ	ความต้านทาน	หน่วย	โอห์ม (Ω)

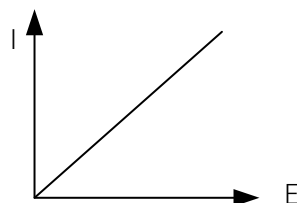
2. การคำนวณหาค่าแรงดันไฟฟ้า (E)

การศึกษากฎของโอห์ม แรงดันไฟฟ้า (E) เป็นแหล่งจ่ายพลังงานภายนอกที่ทำให้เกิดกระแสไฟฟ้า กล่าวคือ ปริมาณการไหลของกระแสไฟฟ้าเมื่อผ่านค่าความต้านทานไฟฟ้าย่อมทำให้เกิดค่าของแรงดันไฟฟ้า ดังภาพที่ 8.5



ภาพที่ 8.5 การต่อวงจรไฟฟ้า

จากภาพที่ 8.5 เมื่อทำให้ปลายทั้งสองของลวดโลหะมีความต่างศักย์ไฟฟ้า จะมีกระแสไฟฟ้าผ่านลวดโลหะนี้ ทำให้เกิดความสัมพันธ์ระหว่างกระแสไฟฟ้าและแรงดันไฟฟ้า ดังภาพที่ 8.6



ภาพที่ 8.6 กราฟระหว่างกระแสไฟฟ้าและแรงดันไฟฟ้าของลวดโลหะ

จากภาพที่ 8.6 พบว่า กระแสไฟฟ้าที่ผ่านลวดโลหะมีค่าแปรผันตรงกับความต่างศักย์ไฟฟ้าระหว่างปลายทั้งสองของลวดโลหะ ดังสมการที่ (4)

$$I \propto E \quad (4)$$

ดังนั้น

$$I = kE \quad \text{เมื่อ } k \text{ เป็นค่าคงตัวของการแปรผัน}$$

หรือ

$$\frac{E}{I} = \frac{1}{k}$$

ถ้าให้

$$\frac{1}{k} = R$$

จะได้ว่ากฎของโอห์ม ดังสมการที่ (5)

$$\frac{E}{I} = R \quad (5)$$

หรือ

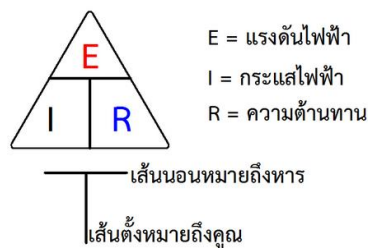
$$E = IR$$

3. การคำนวณหาความต้านทาน (R)

การศึกษากฎของโอห์ม ความต้านทาน (R) ของวัตถุจะถูกกำหนดให้เป็นอัตราส่วนของแรงดันไฟฟ้า (E) ต่อกำลังไฟฟ้า (I) ที่ไหลผ่านขณะที่แรงดันไฟฟ้า (E) เป็นตรงกันข้าม ดังสมการที่ (6)

$$R = \frac{E}{I} \quad (6)$$

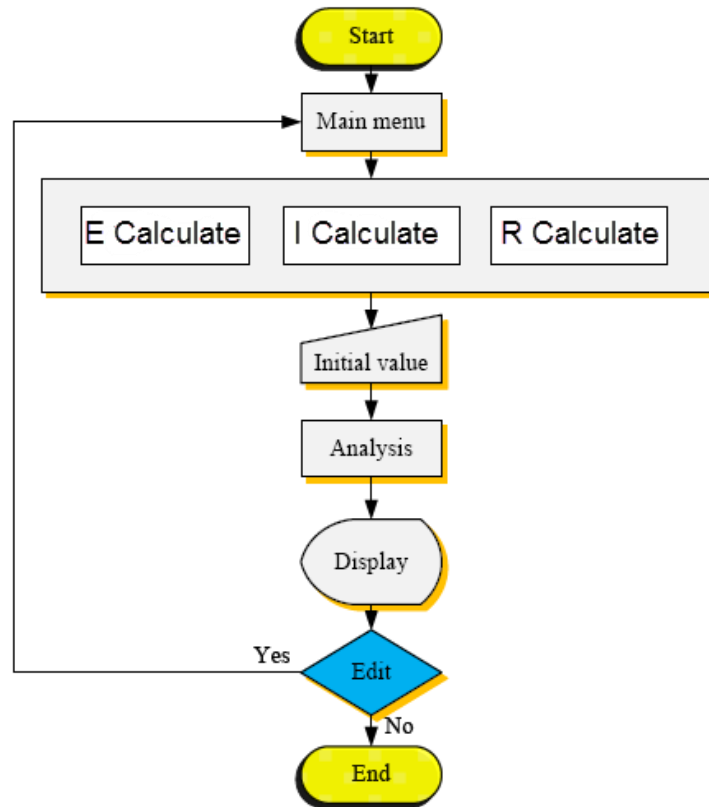
สามารถสรุปสมการกฎของโอห์ม ได้ดังภาพที่ 8.7



ภาพที่ 8.7 การคำนวณสมการกฎของโอห์ม

8.1.2 การออกแบบโปรแกรมจำลอง

การออกแบบโปรแกรมจำลองจะทำการวิเคราะห์อินพุต (Input) เอาต์พุต (Output) และกระบวนการทำงาน (Process Analysis) ของเนื้อหาแต่ละเรื่อง ซึ่งสามารถแสดงกระบวนการทำงานของโปรแกรมจำลอง ดังภาพที่ 8.8



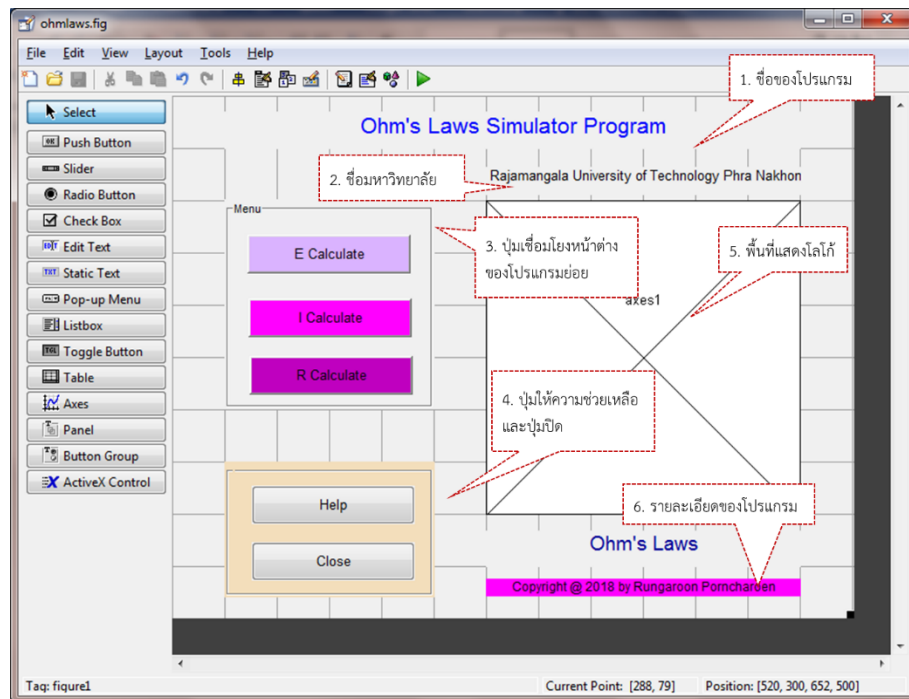
ภาพที่ 8.8 กระบวนการทำงานของโปรแกรมจำลอง

จากภาพที่ 8.8 เริ่มจากการเข้าสู่หน้าต่างหลัก (Main Menu) ของโปรแกรมจำลอง หลังจากนั้นเลือกเนื้อหาที่ทำการจำลอง กำหนดค่าเริ่มต้น (Initial Value) เข้าสู่กระบวนการวิเคราะห์ (Analysis) และแสดงผลการจำลอง (Display) ทั้งนี้ผู้ใช้สามารถปรับเปลี่ยนค่ากระแสไฟฟ้า แรงดันไฟฟ้า และความต้านทานได้ตามต้องการ

8.2 การเขียนโปรแกรมควบคุมการทำงานของวัตถุที่หน้าต่างเมนูหลัก

การพัฒนาโปรแกรมจำลองการคำนวณสมการกฎของโอห์ม จากที่ได้ออกแบบในหัวข้อ 8.1.2 โดยทำการสร้างส่วนติดต่อกับผู้ใช้งาน (Graphic User Interface : GUI) ในส่วนของอินพุต เอาต์พุต และส่วนของการประมวลผลจะเขียนอัลกอริทึมโดยใช้ M-File ของโปรแกรม MATLAB จากสมการคณิตศาสตร์ที่แสดงไว้ในหัวข้อที่ 8.1.1 พร้อมคู่มือการใช้งาน ซึ่งมีรายละเอียดดังนี้

การสร้างหน้าต่างเมนูหลัก (Main Menu) เป็นส่วนแรกของโปรแกรม ดังภาพที่ 8.9



ภาพที่ 8.9 การสร้างหน้าต่างเมนูของโปรแกรมคำนวณสมการกฎของโอห์ม

1. การกำหนดชื่อของโปรแกรม

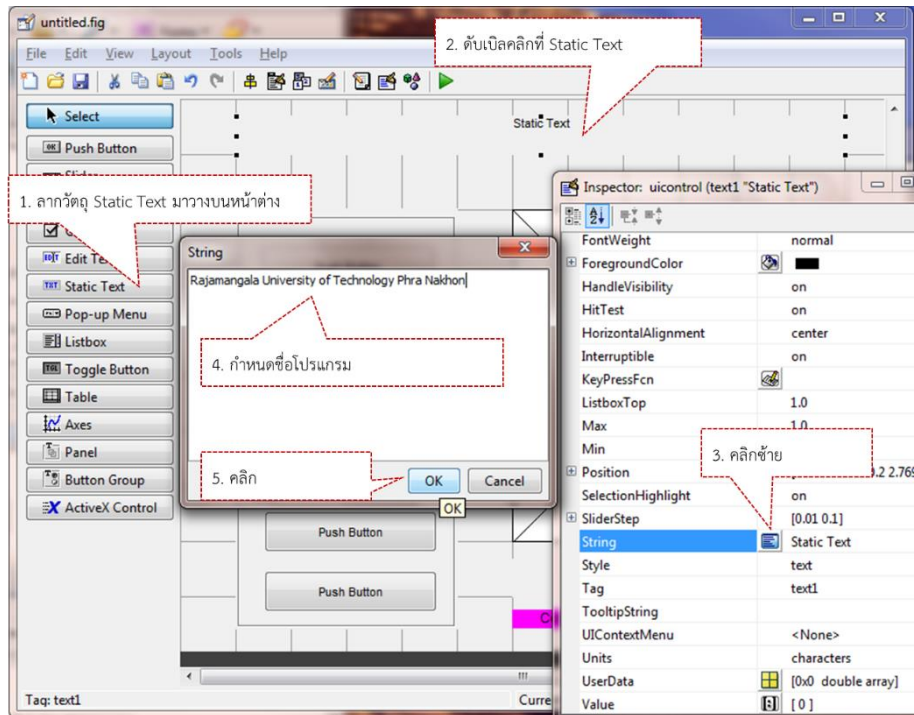
สามารถสร้างโดยใช้วัตถุชนิด Static Text โดยการลากวัตถุดังกล่าวมาวางไว้ที่หน้าต่างการออกแบบ แล้วทำการดับเบิลคลิกที่วัตถุชนิด Static Text จะปรากฏหน้าต่างการกำหนดคุณลักษณะ (Inspector) หลังจากนั้นกำหนด String เป็น Ohm's Laws Simulation

2. การกำหนดชื่อของมหาวิทยาลัย

สามารถสร้างได้เหมือนกับการกำหนดชื่อของโปรแกรม แต่ให้กำหนด String เป็น ชื่อมหาวิทยาลัย (Rajamangala University of Technology Phra Nakhon)

3. การสร้างปุ่มเชื่อมโยงหน้าต่างของโปรแกรมน้อย

สามารถสร้างโดยใช้วัตถุชนิด Push Button 3 วัตถุ และใช้วัตถุชนิด Panel เพื่อรวมกลุ่มของ Push Button ทั้ง 3 เป็นเมนูให้ผู้ใช้โปรแกรมได้เลือก



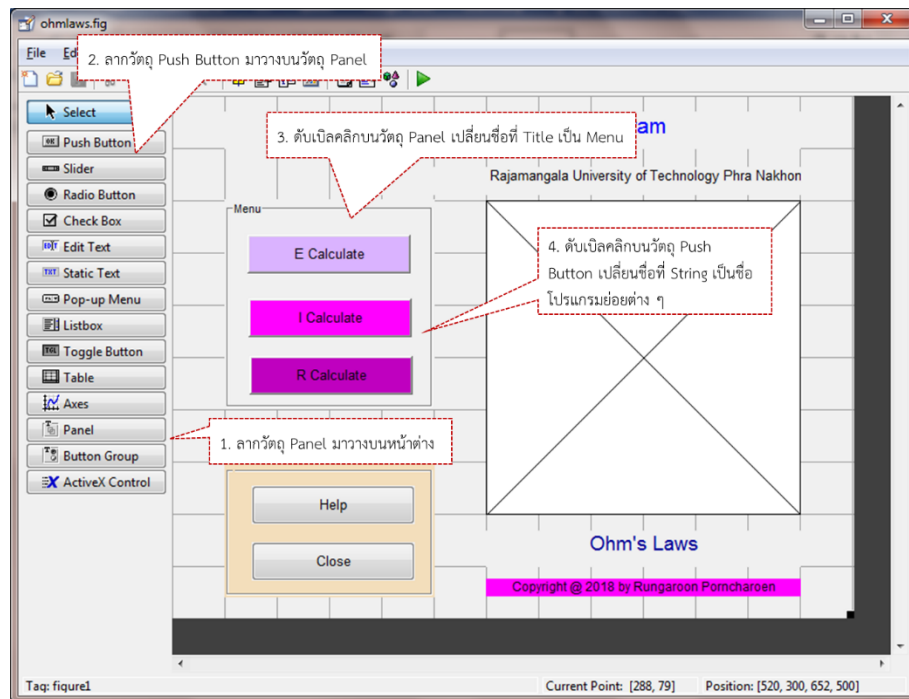
ภาพที่ 8.10 การสร้างชื่อของโปรแกรม Ohm's Laws Simulation

4. การสร้างปุ่มให้ความช่วยเหลือและปุ่มปิด

สามารถสร้างโดยใช้วัตถุชนิด Push Button 2 วัตถุ และใช้วัตถุชนิด Panel เพื่อรวมกลุ่มของ Push Button ทั้ง 2 เป็นเมนูให้ผู้ใช้โปรแกรมได้เลือก โดยมีขั้นตอนการสร้างเหมือนกับหัวข้อที่ 3 แต่เปลี่ยนที่ String ของวัตถุ Push Button เป็น Help และ Close ตามลำดับ และชื่อของวัตถุชนิด Panel ให้ลบบอก (ไม่ต้องกำหนดชื่อ)

5. การแสดงภาพโลโก้

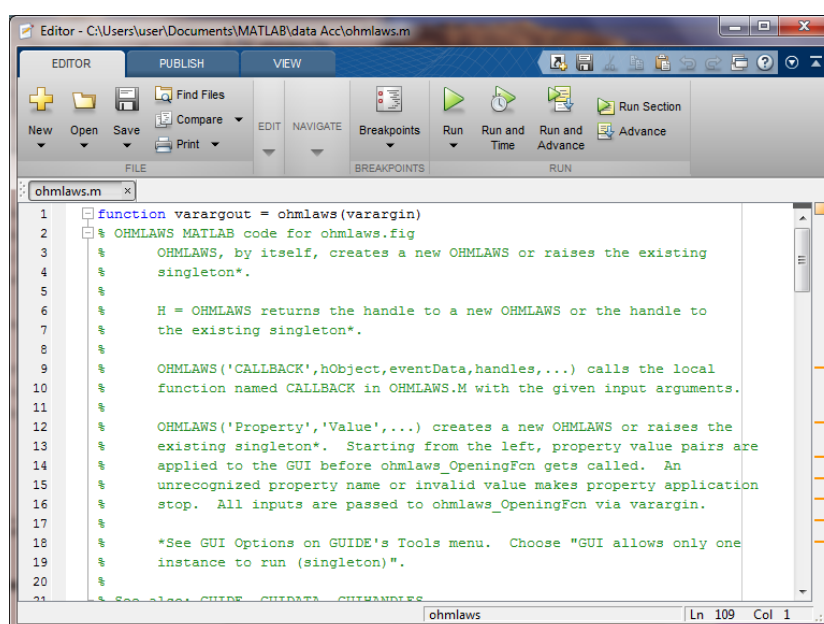
สามารถสร้างโดยใช้วัตถุชนิด Axes โดยการคลิกซ้ายที่วัตถุ Axes ค้างไว้ แล้วลากมาวางบนหน้าต่างออกแบบตามตำแหน่งที่กำหนดไว้



ภาพที่ 8.11 การสร้างปุ่มเชื่อมโยงหน้าต่างของโปรแกรมน้อย

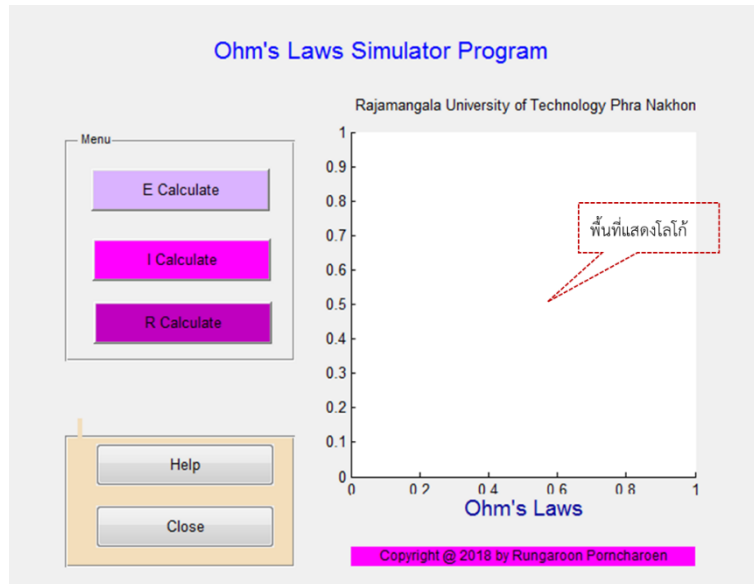
6. การกำหนดรายละเอียดอื่น ๆ ให้กับหน้าต่างเมนูหลักโดยใช้ข้อความ สามารถกระทำได้เช่นเดียวกับหัวข้อที่ 1 และ 2

เมื่อทำการสร้างหน้าต่างเมนูหลักตามที่ออกแบบไว้แล้ว ให้ทำการบันทึกโปรแกรมเป็นชื่อ ohmlaws จากนั้นโปรแกรม MATLAB จะทำการกำหนด M-File ให้อัตโนมัติ ดังภาพที่ 8.12



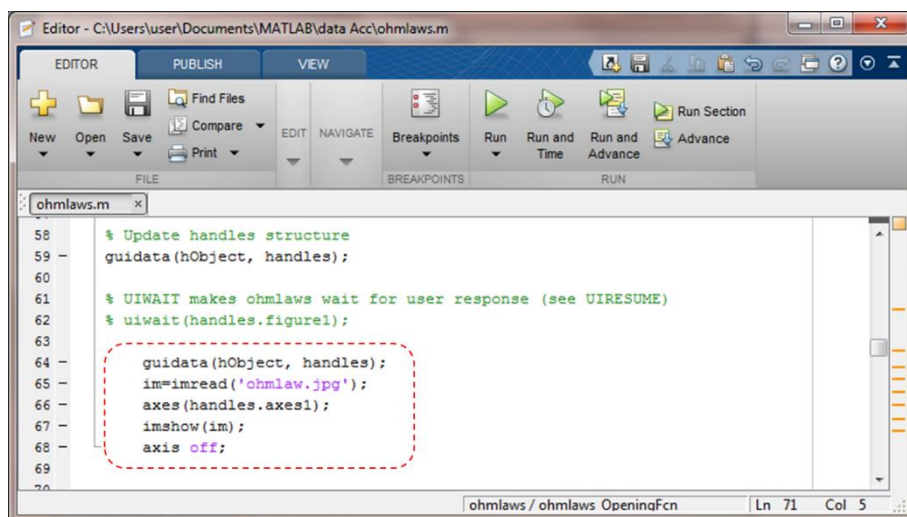
ภาพที่ 8.12 การกำหนด M-File ของหน้าต่างเมนูหลัก

เมื่อทำการรันโปรแกรมที่หน้าต่างเมนูหลัก จะแสดงโลโก้ ดังภาพที่ 8.13



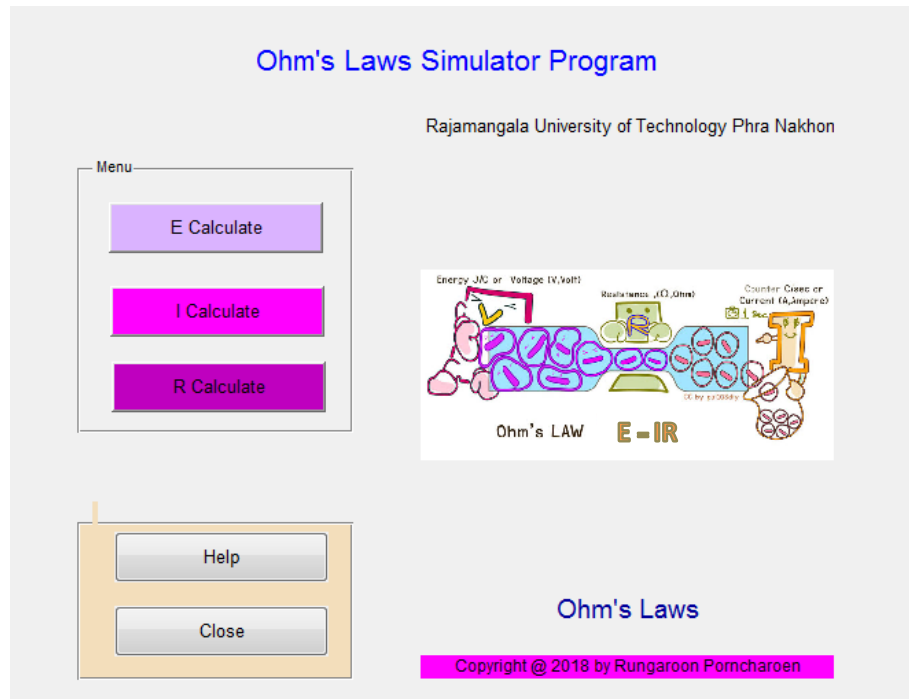
ภาพที่ 8.13 การแสดงผลการรันหน้าต่างเมนูหลัก

จากภาพที่ 8.13 จะเห็นว่าพื้นที่แสดงโลโก้ไม่มีภาพ ดังนั้นจึงต้องสร้างภาพที่นำมาแสดงเป็นโลโก้ก่อน สำหรับในที่นี้ให้ชื่อว่า ohmlaw.jpg ซึ่งเก็บไว้ใน Folder ของโปรแกรม MATLAB สำหรับการเขียนโปรแกรมสำหรับใส่ภาพโลโก้ สามารถเขียนได้ดังภาพที่ 8.14



ภาพที่ 8.14 การเขียนโปรแกรมสำหรับใส่ภาพโลโก้

จากนั้นเมื่อทำการรันโปรแกรมที่หน้าต่างเมนูหลัก สามารถแสดงโลโก้ ดังภาพที่ 8.15

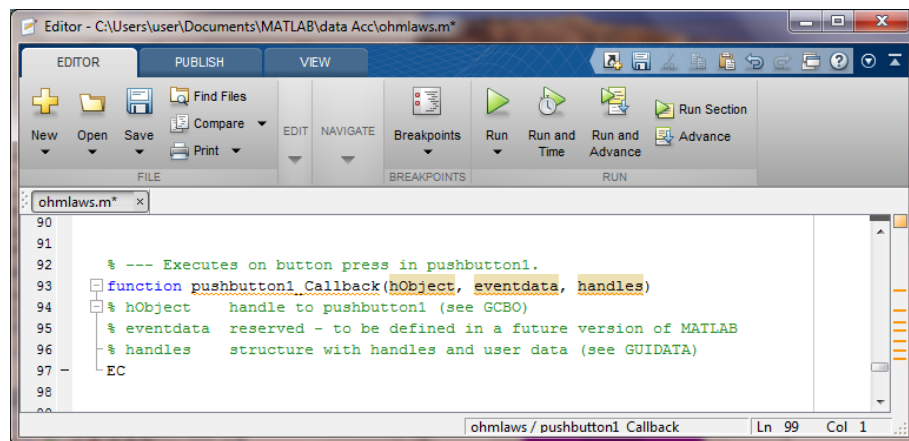


ภาพที่ 8.15 หน้าต่างเมนูที่แสดงโลโก้

8.3 การเขียนโปรแกรมเพื่อควบคุมวัตถุสำหรับเชื่อมโยงไปยังโปรแกรมย่อย

การเขียนโปรแกรมเพื่อควบคุมวัตถุสำหรับเชื่อมโยงไปยังโปรแกรมย่อย ได้แก่ โปรแกรมการคำนวณค่ากระแสไฟฟ้า โปรแกรมการคำนวณค่าแรงดันไฟฟ้า และโปรแกรมการคำนวณค่าความต้านทาน มีรายละเอียดดังนี้

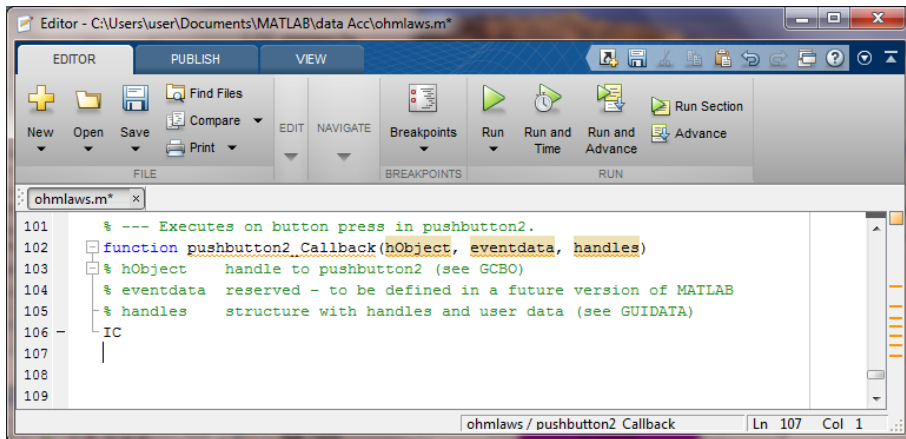
1. คลิกขวาที่ปุ่ม **E Calculate** เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callbacks จะปรากฏ M-File Editor ที่ฟังก์ชัน pushbutton1 ให้พิมพ์ชื่อ EC (EC หมายถึง โปรแกรมการคำนวณค่าแรงดันไฟฟ้า) และทำการบันทึก ดังภาพที่ 8.16



ภาพที่ 8.16 การเชื่อมโยงไปยังหน้าต่างโปรแกรมการคำนวณค่าแรงดันไฟฟ้า

I Calculate

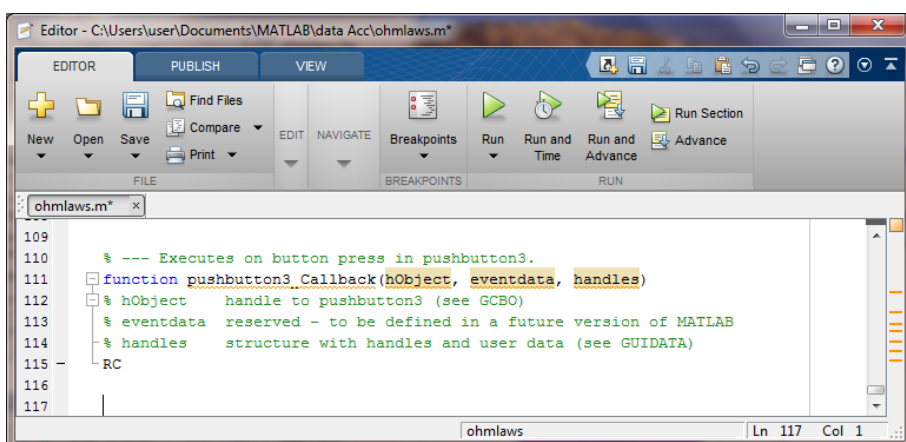
2. คลิกขวาที่ปุ่ม **I Calculate** เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callbacks จะปรากฏ M-File Editor ที่ฟังก์ชัน pushbutton2 ให้พิมพ์ชื่อ IC (IC หมายถึง โปรแกรมการคำนวณค่ากระแสไฟฟ้า) และทำการบันทึก ดังภาพที่ 8.17



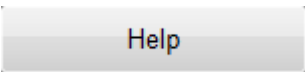
ภาพที่ 8.17 การเชื่อมโยงไปยังหน้าต่างโปรแกรมการคำนวณค่ากระแสไฟฟ้า

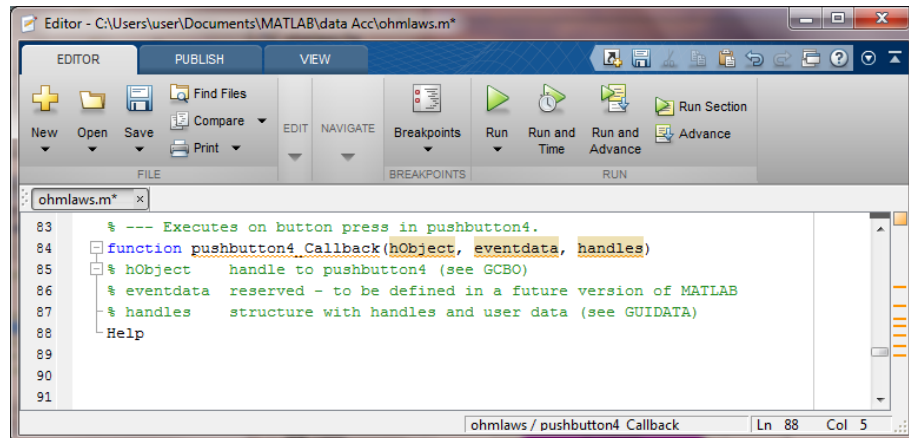
R Calculate

3. คลิกขวาที่ปุ่ม **R Calculate** เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callbacks จะปรากฏ M-File Editor ที่ฟังก์ชัน pushbutton3 ให้พิมพ์ชื่อ RC (RC หมายถึง โปรแกรมการคำนวณค่าความต้านทาน) และทำการบันทึก ดังภาพที่ 8.18

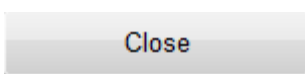


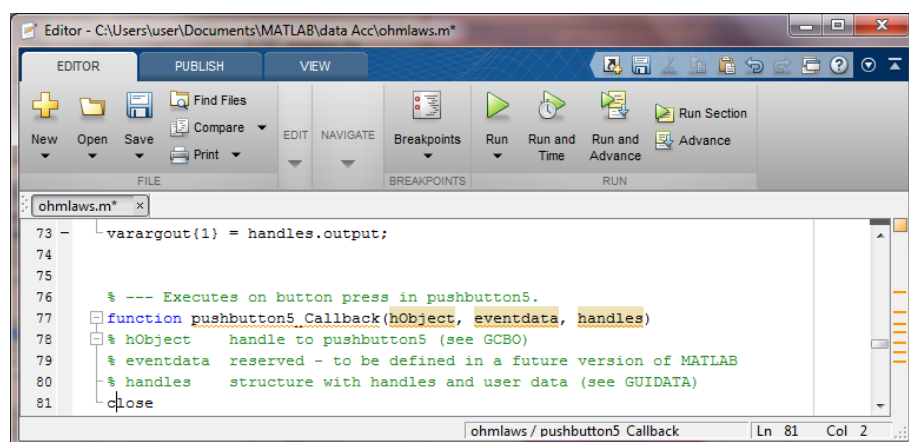
ภาพที่ 8.18 การเชื่อมโยงไปยังหน้าต่างโปรแกรมการคำนวณค่าความต้านทาน

4. คลิกขวาที่ปุ่ม  เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callbacks จะปรากฏ M-File Editor ที่ฟังก์ชัน pushbutton4 ให้พิมพ์ชื่อ Help (Help หมายถึง การแสดงคู่มือการใช้งานของโปรแกรม) และทำการบันทึก ดังภาพที่ 8.19



ภาพที่ 8.19 การเชื่อมโยงไปยังหน้าต่าง Help

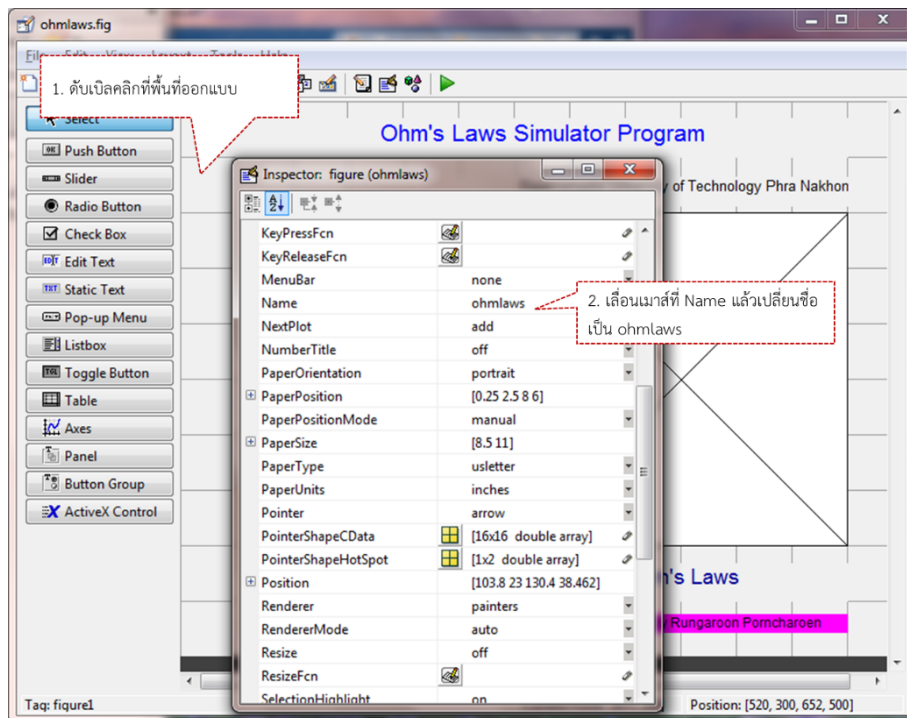
5. คลิกขวาที่ปุ่ม  เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callbacks จะปรากฏ M-File Editor ที่ฟังก์ชัน pushbutton5 ให้พิมพ์ชื่อ Close (Close หมายถึง ออกจากการทำงานของโปรแกรม) และทำการบันทึก ดังภาพที่ 8.20



ภาพที่ 8.20 การปิดหน้าต่างโปรแกรม

เมื่อทำการรันโปรแกรมหน้าต่างเมนูหลัก จะปรากฏชื่อโปรแกรมที่มุมด้านบนซ้ายว่า Main ซึ่งสามารถปรับเปลี่ยนชื่อได้ ดังภาพที่ 8.21 สามารถอธิบายได้ดังนี้

1. ดับเบิลคลิกบนพื้นที่ออกแบบวงจร จะปรากฏหน้าต่าง Inspector
2. เลื่อนเมาส์มาที่ Name แล้วเปลี่ยนชื่อเป็น Ohm Simulator
3. เมื่อทำการรันโปรแกรม ชื่อโปรแกรมที่มุมด้านบนซ้ายจะเปลี่ยนเป็น Ohm Simulator



ภาพที่ 8.21 การเปลี่ยนชื่อในหน้าต่างของโปรแกรมเมนูหลัก

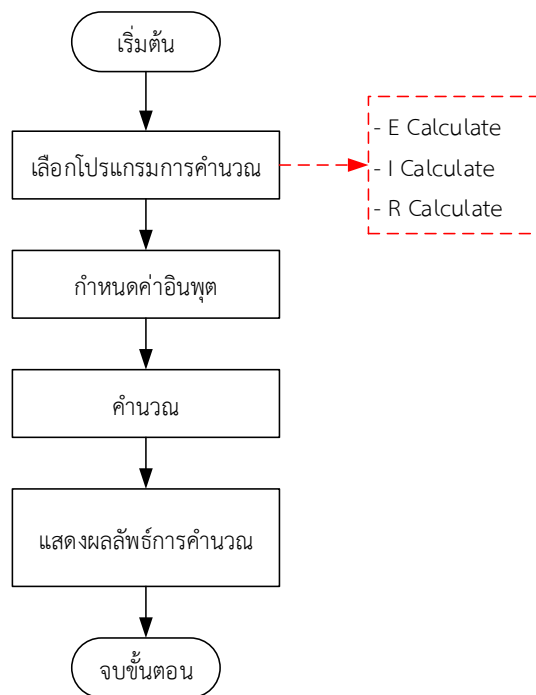
เมื่อเสร็จสิ้นการเขียนโปรแกรมควบคุมการทำงานของวัตถุเพื่อเชื่อมโยงไปยังโปรแกรมย่อยต่าง ๆ สามารถทดสอบการเชื่อมโยงว่าเป็นไปตามเงื่อนไขที่ผู้ออกแบบต้องการหรือไม่

ถ้าใช่ จบกระบวนการ

ถ้าไม่ใช่ ให้ปรับปรุงแก้ไข

8.4 การควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมคำนวณค่าแรงดันไฟฟ้า

ก่อนที่จะกล่าวถึงการควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม ควรต้องทราบรายละเอียดของกระบวนการทำงานของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม ดังภาพที่ 8.22

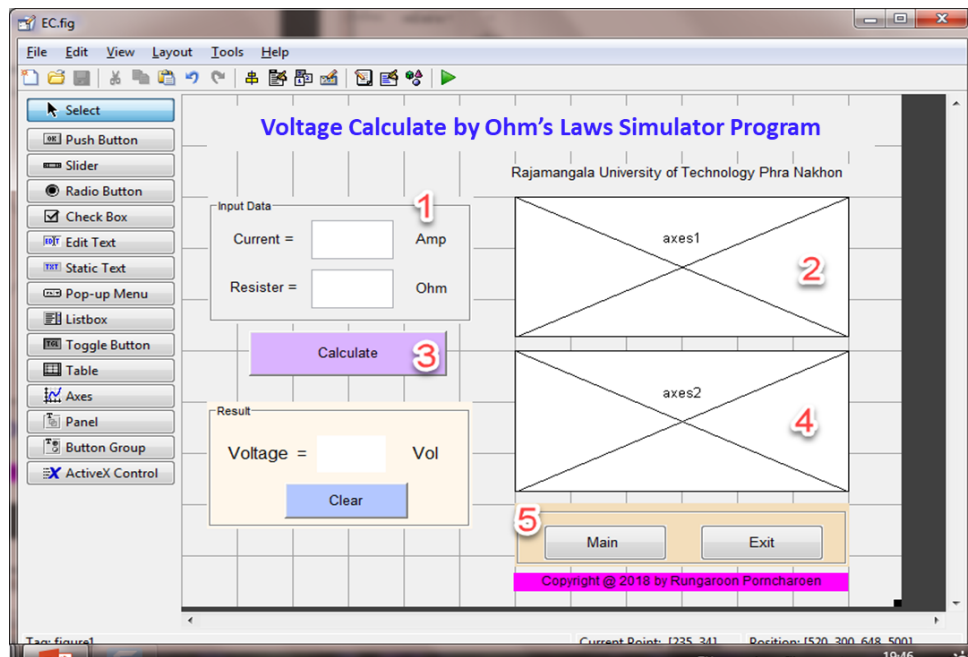


ภาพที่ 8.22 กระบวนการทำงานของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม

การทำงานของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม สามารถอธิบายได้ดังนี้ เมื่อเปิดโปรแกรมเมนูหลักและทำการเลือก E Calculate ขึ้นมา หน้าต่างของโปรแกรมจะทำการกำหนดค่าเริ่มต้นให้กับโปรแกรมโดยอัตโนมัติ ซึ่งผู้ใช้งานสามารถปรับเปลี่ยนค่าได้ตามต้องการ โดยการกำหนดค่ากระแสไฟฟ้า และค่าความต้านทาน แล้วกด Calculate โปรแกรมจะทำการผลลัพธ์ซึ่งรายละเอียดการเขียนโปรแกรมสามารถอธิบายได้ ดังนี้

8.4.1 การสร้างโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม

โปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม เป็นโปรแกรมย่อยสำหรับคำนวณหาค่าแรงดันไฟฟ้า ซึ่งมีความสัมพันธ์ระหว่างกระแสไฟฟ้า (I) และความต้านทาน (R) (ดังรายละเอียดในบทที่ 10) โดยการกำหนดวัตถุรับค่าอินพุต ได้แก่ กระแสไฟฟ้า (I) และความต้านทาน (R) และแสดงผลของค่าแรงดันไฟฟ้า ดังภาพที่ 8.23



ภาพที่ 8.23 การออกแบบลักษณะของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม

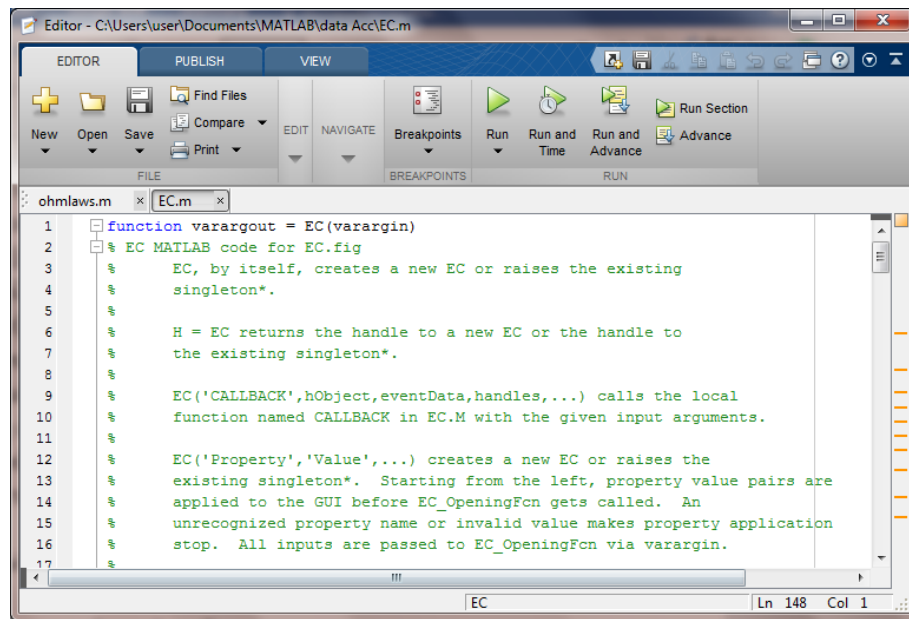
จากภาพที่ 8.23 สามารถอธิบายขั้นตอนการสร้างโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม ดังนี้

1. การสร้างวัตถุรับค่าอินพุต (Input)
วัตถุที่นำมาใช้สำหรับการสร้างวัตถุรับค่าอินพุต (Input) ประกอบด้วยวัตถุชนิดต่าง ๆ ดังนี้
 - Panel สำหรับรวมกลุ่มของอินพุตทั้งหมด โดยกำหนดชื่อเป็น Input
 - Edit Text สำหรับใช้ในการป้อนค่ากระแส (I) ค่าความต้านทาน (R) โดยทำการกำหนดค่าที่ String ของแต่ละวัตถุให้เป็นค่าเริ่มต้นของโปรแกรม
 - Static Text สำหรับใช้ในการกำหนดชื่อและหน่วยของตัวรับค่าอินพุต
2. การสร้างพื้นที่แสดงภาพสมการกฎของโอห์ม
วัตถุที่นำมาใช้สำหรับการสร้างพื้นที่แสดงภาพสมการกฎของโอห์ม ได้แก่ Axes ชื่อ axes1
3. การสร้างปุ่มกดเพื่อแสดงผลการคำนวณ
วัตถุที่นำมาใช้สำหรับการสร้างปุ่มกดเพื่อแสดงผลการคำนวณ ได้แก่ Push Button โดยทำการกำหนดค่าที่ String ของวัตถุแต่ละวัตถุให้เป็น EC
4. การสร้างพื้นที่แสดงกราฟสัญญาณที่เกิดขึ้น
วัตถุที่นำมาใช้สำหรับการสร้างพื้นที่แสดงภาพวงจรไฟฟ้า ได้แก่ Axes ชื่อ axes2
5. การสร้างปุ่มอำนวยความสะดวกอื่น ๆ
การสร้างปุ่มอำนวยความสะดวกอื่น ๆ ประกอบด้วย ปุ่มต่าง ๆ ดังต่อไปนี้
 - ปุ่ม Main เป็นปุ่มที่ใช้เรียกกลับไปยังหน้าต่างเมนูหลักของโปรแกรมที่สร้างขึ้น ได้แก่ วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Main

- ปุ่ม Clear เป็นปุ่มที่ใช้สำหรับลบข้อมูลต่าง ๆ ทั้งหมด วัตถุที่นำมาใช้สำหรับสร้าง ได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Main

- ปุ่ม Exit เป็นปุ่มที่ใช้สำหรับปิดโปรแกรม วัตถุที่นำมาใช้สำหรับสร้าง ได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Exit

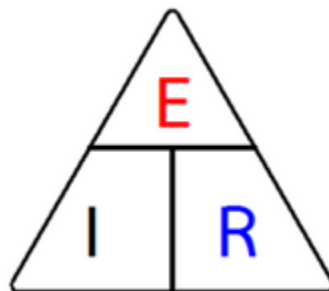
เมื่อทำการสร้างหน้าต่างโปรแกรมตามที่ออกแบบไว้เรียบร้อยแล้ว ให้ทำการบันทึกโปรแกรมเป็นชื่อ EC จากนั้นโปรแกรม MATLAB จะทำการสร้าง M-File ให้อัตโนมัติ ดังภาพที่ 8.24



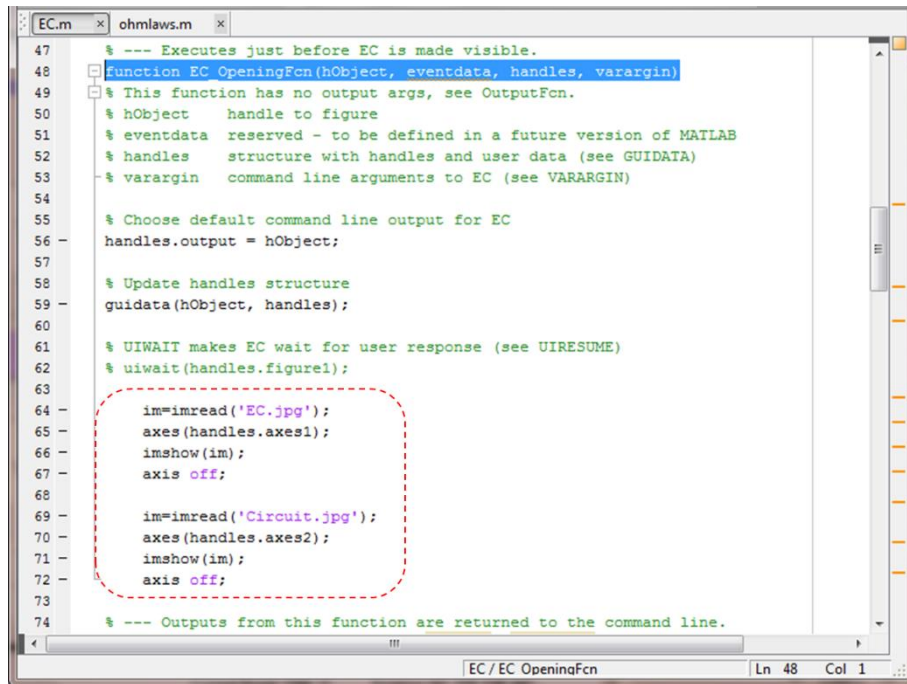
ภาพที่ 8.24 หน้าต่าง M-File ของโปรแกรมคำนวณค่าแรงดันไฟฟ้าจากสมการกฎของโอห์ม

9.1.2 การใส่ภาพสูตรการคำนวณค่าแรงดันไฟฟ้า

เมื่อเลือกโปรแกรม E Calculate จากหน้าต่างเมนูหลักครั้งแรก ปรากฏดังภาพที่ 8.25 จะพบว่า โปรแกรมยังไม่เป็นไปตามเงื่อนไขของกระบวนการทำงาน คือ โปรแกรมไม่แสดงภาพสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม ดังนั้น ต้องมีการเขียนฟังก์ชันการทำงาน เพื่อโปรแกรมจะได้เรียกภาพที่ต้องการขึ้นมาบนหน้าต่างตามกระบวนการทำงานที่วางไว้ ดังภาพที่ 8.26

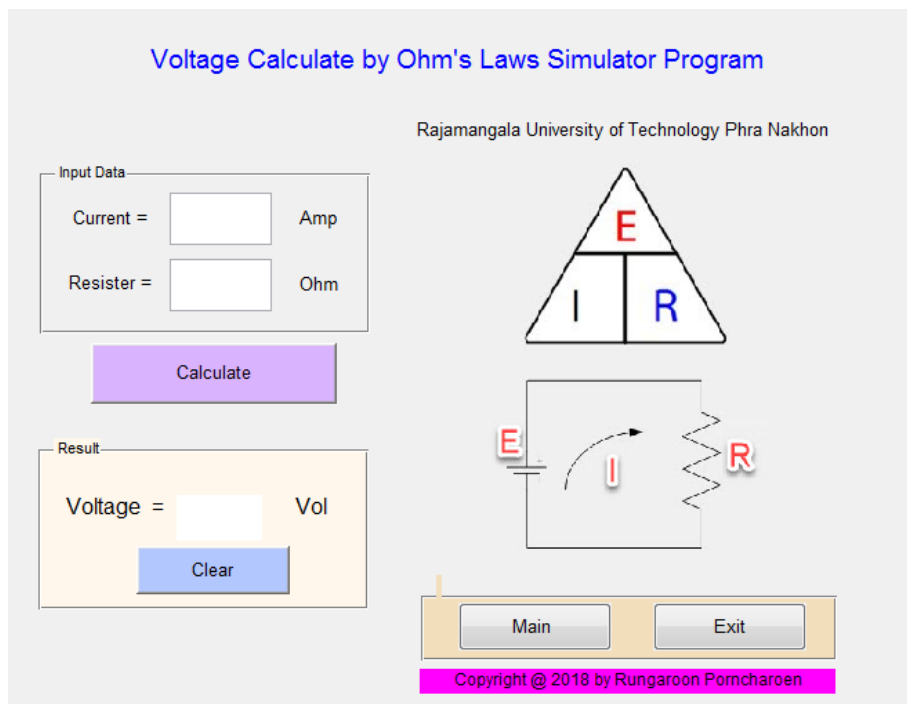


ภาพที่ 8.25 สมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม



ภาพที่ 8.26 การเขียนฟังก์ชันการทำงานใส่ภาพสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม

เมื่อทำการรันโปรแกรมสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม ปรากฏดังภาพที่ 8.27



ภาพที่ 8.27 ผลลัพธ์การใส่ภาพในโปรแกรม

8.5 การเขียนโปรแกรมเพื่อควบคุมวัตถุคำนวณค่าแรงดันไฟฟ้า

สำหรับการเขียนโปรแกรมเพื่อควบคุมวัตถุคำนวณค่าแรงดันไฟฟ้า ตามกระบวนการทำงานที่วางไว้นั้น ประกอบด้วย ปุ่ม Calculate ปุ่ม Main ปุ่ม Clear และปุ่ม Exit มีขั้นตอนการเขียนโปรแกรมดังนี้

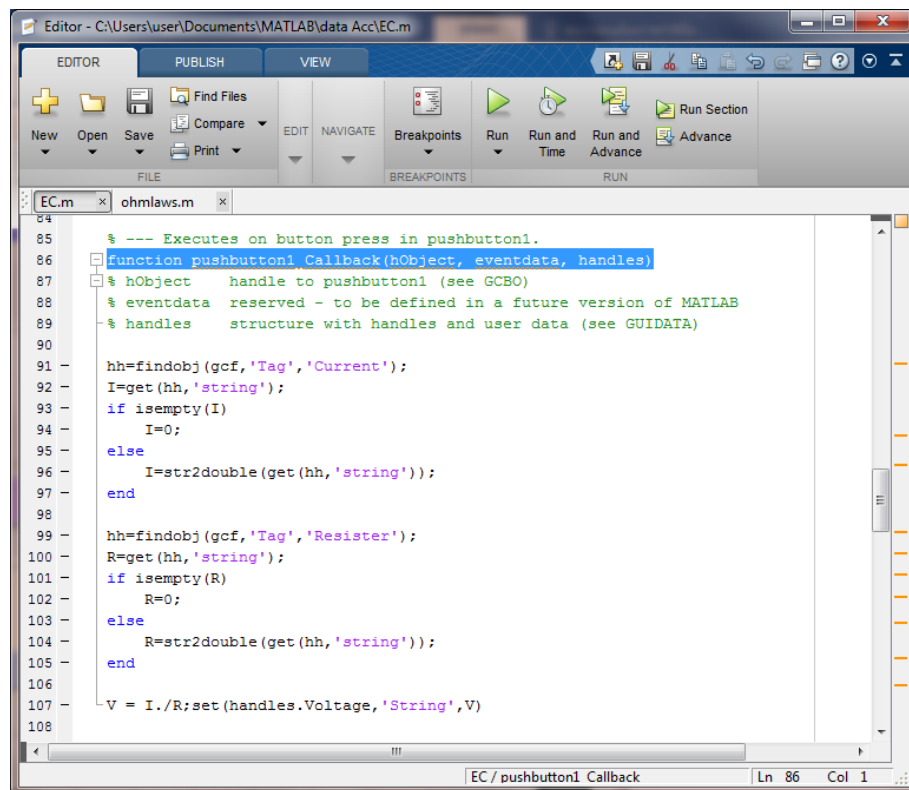
- ปุ่ม Calculate

Calculate

1. คลิกขวาที่ปุ่ม แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

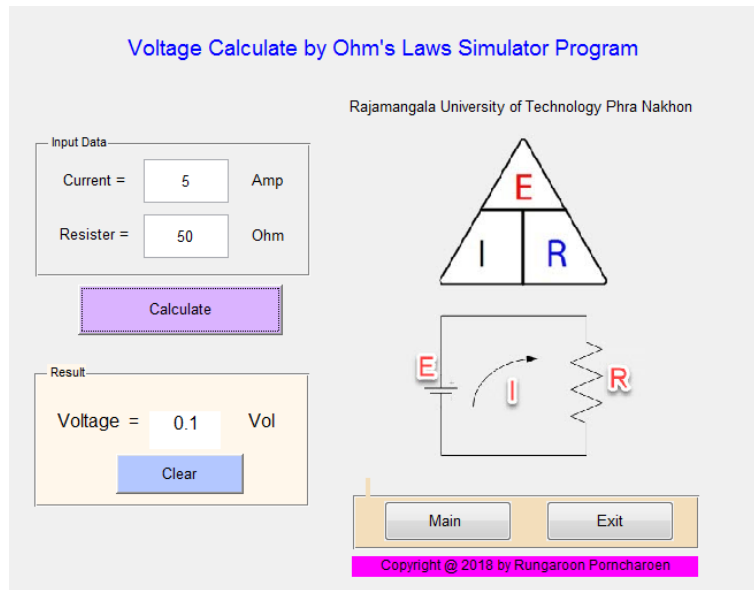
```
function pushbutton1_Callback(hObject, eventdata, handles)
```

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 8.28



ภาพที่ 8.28 การเขียนโปรแกรมเพื่อคำนวณแรงดันไฟฟ้า

2. เมื่อทำการรันโปรแกรมสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม แล้วทำการใส่ค่ากระแสไฟฟ้าและค่าความต้านทาน จากนั้นกดปุ่ม Calculate จะแสดงผลลัพธ์การคำนวณ ดังภาพที่ 8.29



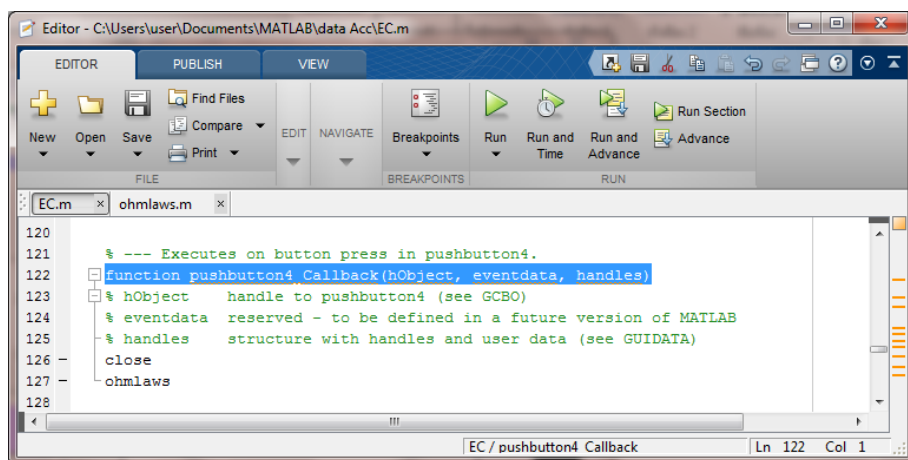
ภาพที่ 8.29 ผลการจำลองโปรแกรมสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม

- ปุ่ม Main

1. คลิกขวาที่ปุ่ม  แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

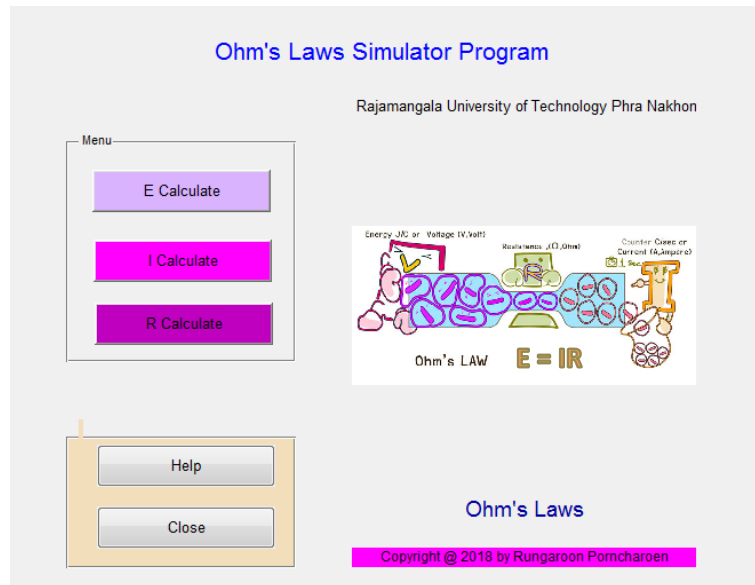
function pushbutton4_Callback(hObject, eventdata, handles)

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 8.30



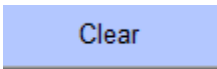
ภาพที่ 8.30 การเขียนโปรแกรมไปยังหน้าต่างเมนูหลัก

2. เมื่อทำการรันโปรแกรมสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม หากต้องการกลับไปยังเมนูหลักสามารถทำได้โดยการกดปุ่ม Main ดังภาพที่ 8.31



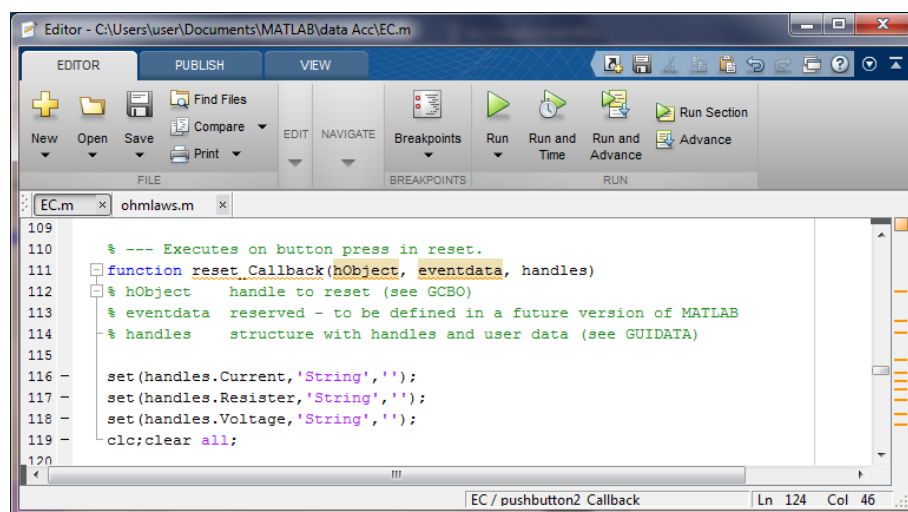
ภาพที่ 8.31 ผลการรันปุ่ม Main

- ปุ่ม Clear

1. คลิกขวาที่ปุ่ม  แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

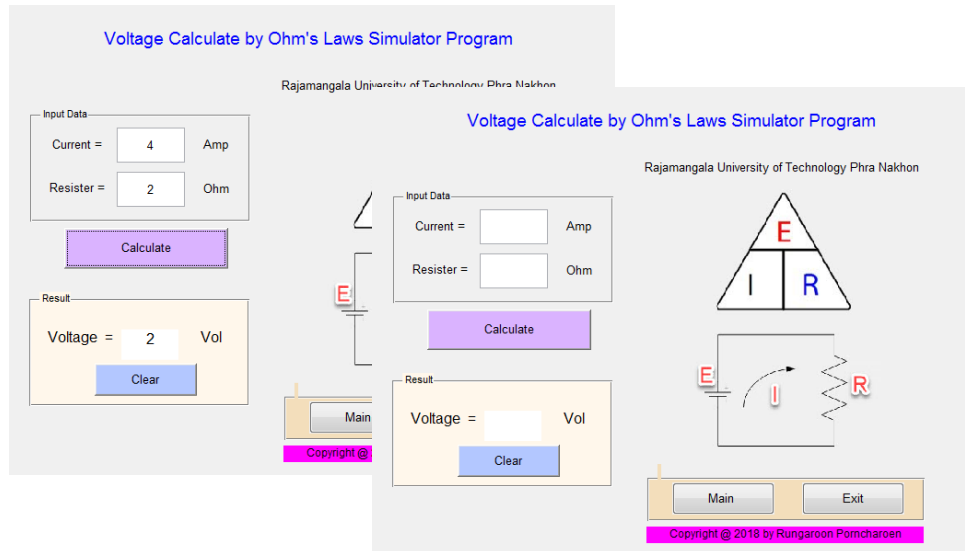
function reset_Callback(hObject, eventdata, handles)

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 8.32



ภาพที่ 8.32 การเขียนโปรแกรมปุ่ม Clear

2. เมื่อทำการรันโปรแกรมสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์มแล้ว หากต้องการลบข้อมูลที่ใส่ไป สามารถทำได้โดยการกดปุ่ม Clear ดังภาพที่ 8.33



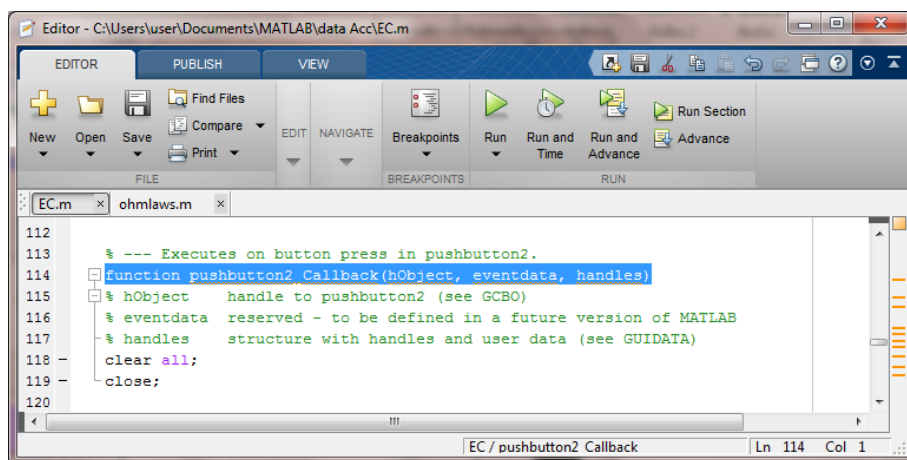
ภาพที่ 8.33 ผลการรันปุ่ม Clear

- ปุ่ม Exit

1. คลิกขวาที่ปุ่ม  แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

function pushbutton2_Callback(hObject, eventdata, handles)

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 8.34



ภาพที่ 8.34 การเขียนโปรแกรมปุ่ม Exit

2. เมื่อทำการรันโปรแกรมสมการคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์มแล้ว หากต้องการออกจากโปรแกรมทั้งหมด สามารถทำได้โดยการกดปุ่ม Exit

8.6 สรุป

การประยุกต์ใช้ GUI ของโปรแกรม MATLAB สำหรับการเขียนโปรแกรมต่าง ๆ เริ่มจากการวิเคราะห์งาน และออกแบบโปรแกรมจำลอง เพื่อกำหนดลักษณะของหน้าจอของโปรแกรม หลังจากนั้นเขียนโปรแกรมควบคุมการทำงานของวัตถุที่หน้าต่างเมนูหลัก โดยทำการสร้างส่วนติดต่อกับผู้ใช้งาน (Graphic User Interface : GUI) ในส่วนของอินพุต เอาต์พุต และส่วนของการประมวลผลจะเขียนอัลกอริทึมโดยใช้ M-File ของโปรแกรม MATLAB รวมถึงการเขียนโปรแกรมเพื่อควบคุมวัตถุสำหรับเชื่อมโยงไปยังโปรแกรมย่อย เพื่อควบคุมการทำงานของเมนูหลักและเมนูย่อย

โปรแกรมการคำนวณค่าแรงดันไฟฟ้าของสมการกฎของโอห์ม เป็นโปรแกรมเพื่อควบคุมวัตถุสำหรับเชื่อมโยงไปยังโปรแกรมสมการกฎของโอห์ม มีกระบวนการดำเนินงานดังนี้ เมื่อเปิดโปรแกรมเมนูหลักและทำการเลือก E Calculate ขึ้นมา หน้าต่างของโปรแกรมจะทำการกำหนดค่าเริ่มต้นให้กับโปรแกรมโดยอัตโนมัติ ซึ่งผู้ใช้งานสามารถปรับเปลี่ยนค่าได้ตามต้องการ โดยการกำหนดค่ากระแสไฟฟ้า และค่าความต้านทาน แล้วกด Calculate โปรแกรมจะทำการผลลัพธ์

ข้อควรระวังสำหรับการใส่ภาพลงใน GUI ต้องเก็บ file ภาพลงในโปรแกรม MATLAB จะทำให้สามารถเรียกมาใช้งานได้อย่างสะดวก และการตั้งชื่อ Function ต้องไม่ซ้ำกับชื่อ File เพื่อลดการซ้ำซ้อนของโปรแกรม

แบบฝึกหัด

บทที่ 8 การประยุกต์ใช้ GUI ของ MATLAB สำหรับการคำนวณสมการกฎของโอห์ม

1. จงบอกวัตถุประสงค์ที่ใช้ในการสร้างหน้าต่างหลัก (Main Menu) ของโปรแกรมการคำนวณสมการกฎของโอห์ม พร้อมอธิบายขั้นตอนการสร้าง
2. จงอธิบายการสร้างปุ่มให้ความช่วยเหลือและปุ่มปิดว่าควรใช้วัตถุชนิดใด และเพราะเหตุใด
3. หากต้องการแสดงภาพโลโก้โปรแกรมจำลอง ควรดำเนินการและเขียนโปรแกรมอย่างไรให้ภาพแสดงบนโปรแกรม
4. จงอธิบายขั้นตอนการสร้างโปรแกรมคำนวณแรงดันไฟฟ้าจากสมการกฎของโอห์ม
5. จงเขียนโปรแกรมคำนวณหาค่าแรงดันไฟฟ้า ลงในหน้าต่าง M-File และเมื่อกดปุ่ม Calculate สามารถแสดงคำตอบลงบนหน้าต่างโปรแกรม
6. จงเขียนปุ่ม Main เพื่อควบคุมการทำงานโดยการกลับสู่หน้าจอหลัก
7. จงอธิบายการควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมคำนวณค่ากระแสไฟฟ้าจากสมการกฎของโอห์ม
8. จงอธิบายการแทรกภาพในหน้าต่างโปรแกรมคำนวณค่ากระแสไฟฟ้าจากสมการกฎของโอห์ม
9. จงประยุกต์ใช้ GUI ของ MATLAB ตามตัวอย่างในบทที่ 8 ในการสร้างโปรแกรมการคำนวณเกี่ยวกับช่างไฟฟ้าและอิเล็กทรอนิกส์ โดยใช้หลักการออกแบบตามบทที่ 2 พร้อมทั้งอธิบายรายละเอียดของการสร้าง โดยให้นำเสนอในรูปแบบรายงานประกอบโปรแกรมที่สร้างขึ้น
10. จงออกแบบและเขียนโปรแกรมคำนวณค่าความต้านทานจากสมการกฎของโอห์ม พร้อมทั้งอธิบายรายละเอียดของการสร้าง
11. จงออกแบบและเขียนโปรแกรมคำนวณค่าอนุพันธ์พื้นฐาน พร้อมทั้งอธิบายรายละเอียดของการสร้าง
12. จงออกแบบและเขียนโปรแกรมคำนวณกฎการเคลื่อนที่ของนิวตัน พร้อมทั้งอธิบายรายละเอียดของการสร้าง

13. จงออกแบบและเขียนโปรแกรมจำลองการวิเคราะห์วงจรไฟฟ้าจากกฎของเคอร์ชอฟฟ์ พร้อมทั้งอธิบายรายละเอียดของการสร้าง โดยมีเงื่อนไขดังนี้

- 13.1 มีหน้าต่างเมนูหลัก
- 13.2 มีการแสดงภาพในแต่ละเมนู
- 13.3 มีปุ่มคำนวณ แสดงผลบนโปรแกรม และปุ่มปิดโปรแกรม

14. จงออกแบบและเขียนโปรแกรมวิเคราะห์วงจรไบอัสทรานซิสเตอร์ 4 แบบ ได้แก่ วงจรไบอัสคงที่ วงจรไบอัสอีมิเตอร์สเตบิไลซ์ วงจรไบอัสแบบแบ่งแรงดันไฟฟ้า และวงจรไบอัสป้อนกลับคอลเลคเตอร์ พร้อมทั้งอธิบายรายละเอียดของการสร้าง โดยมีเงื่อนไขดังนี้

- 14.1 มีหน้าต่างเมนูหลัก
- 14.2 มีการแสดงภาพในแต่ละเมนู
- 14.3 มีปุ่มคำนวณ แสดงผลบนโปรแกรม และปุ่มปิดโปรแกรม

15. จงออกแบบและเขียนโปรแกรมคำนวณการเดินทางของแสงแบบเมอริเดียน โดยโปรแกรมสามารถคำนวณหาค่าต่าง ๆ ได้ดังนี้

- 15.1 มุมวิกฤตที่รอยต่อระหว่างแก้วแกนในและแคลดดิ้ง
- 15.2 ค่าความสามารถรับแสง
- 15.3 มุมรับแสง

บทที่ 9

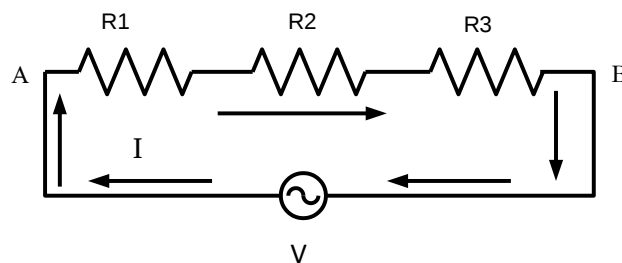
การประยุกต์ใช้ GUI ของ MATLAB สำหรับโปรแกรมการคำนวณวงจรไฟฟ้า

บทนี้เป็นการอธิบายขั้นตอนการพัฒนาโปรแกรมการคำนวณวงจรไฟฟ้าที่ประยุกต์ใช้ GUI ของโปรแกรม MATLAB โดยจะทำการควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมอย่างละเอียดทีละขั้นตอน เพื่อให้เกิดความเข้าใจและสามารถนำไปประยุกต์ใช้งานได้ดียิ่งขึ้น

ก่อนเริ่มเขียนโปรแกรมการคำนวณวงจรไฟฟ้า มาทำศึกษารายละเอียดการต่อวงจรความต้านทานในรูปแบบต่าง ๆ ก่อน ดังนี้

การต่อความต้านทาน ความต้านทานใช้สัญลักษณ์ Ω เมื่อนำมาต่อในรูปแบบต่าง ๆ จะมีคุณสมบัติดังนี้

1. การต่อแบบอนุกรม



ภาพที่ 9.1 การต่อความต้านทานแบบอนุกรม

เมื่อนำความต้านทาน R_1 , R_2 และ R_3 มาต่อแบบอนุกรมกันดังภาพที่ 9.1 แล้วนำปลาย A และ B ไปต่อกับความต่างศักย์ V จะได้ว่า

$$V = V_1 + V_2 + V_3$$

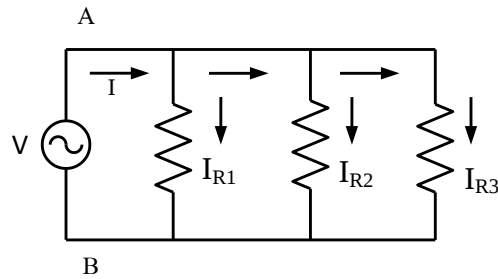
$$I = I_1 = I_2 = I_3$$

และ

$$R = R_1 + R_2 + R_3$$

เมื่อ V_1 , V_2 และ V_3 เป็นความต่างศักย์ระหว่างปลายของความต้านทาน R_1 , R_2 และ R_3 ตามลำดับ I_1 , I_2 และ I_3 เป็นกระแสที่ผ่าน R_1 , R_2 และ R_3 ตามลำดับ I เป็นกระแสที่ไหลเข้าที่จุด A ออกที่จุด B และ R เป็นความต้านทานรวม

2. การต่อแบบขนาน



ภาพที่ 9.2 การต่อความต้านทานแบบขนาน

เมื่อนำความต้านทาน R_1 , R_2 และ R_3 มาต่อแบบขนานดังภาพที่ 9.2 แล้วนำปลาย A และ B ไปต่อกับความต่างศักย์ V จะได้ว่า

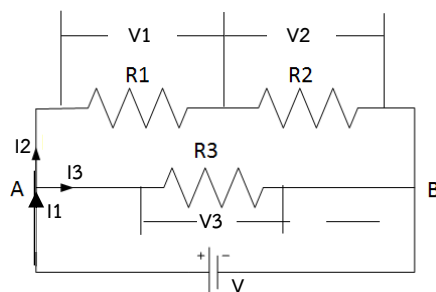
$$V = V_1 = V_2 = V_3$$

$$I = I_1 + I_2 + I_3$$

และ

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

3. การต่อแบบผสม



ภาพที่ 9.3 การต่อความต้านทานแบบผสมอย่างหนึ่ง

เมื่อนำความต้านทาน R_1 , R_2 และ R_3 มาต่อแบบผสมดังภาพที่ 9.3 แล้วนำปลาย A และ B ไปต่อกับความต่างศักย์ V จากคุณสมบัติการต่อความต้านทานแบบอนุกรมและขนานจะได้ว่า

$$V_2 = V_3$$

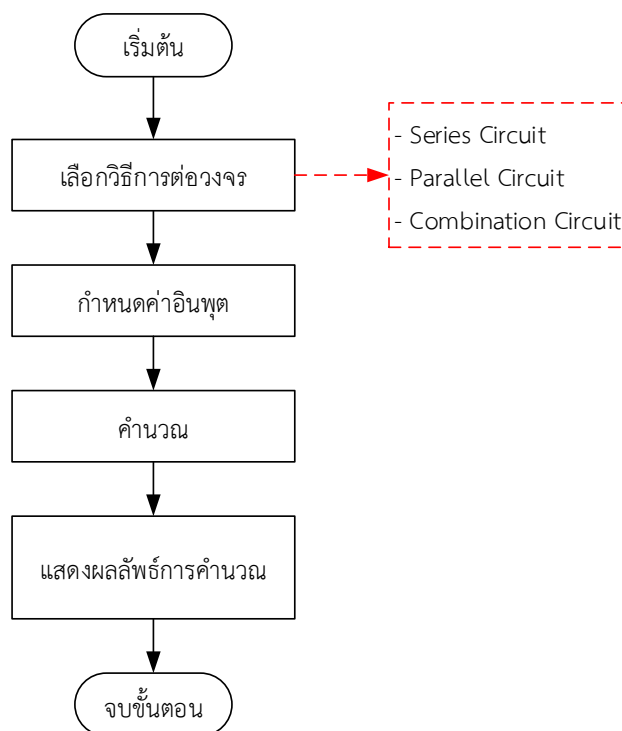
$$V = V_1 + V_2 \text{ หรือ } = V_3$$

$$I = I_1 = I_2 + I_3$$

และ
$$R = \frac{(R_1 + R_2)R_3}{R_1 + R_2 + R_3}$$

9.1 การควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมการคำนวณวงจรไฟฟ้า

ก่อนที่จะกล่าวถึงการควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมการคำนวณวงจรไฟฟ้า ควรต้องทราบรายละเอียดของกระบวนการทำงานของโปรแกรมการคำนวณวงจรไฟฟ้า ดังภาพที่ 9.4



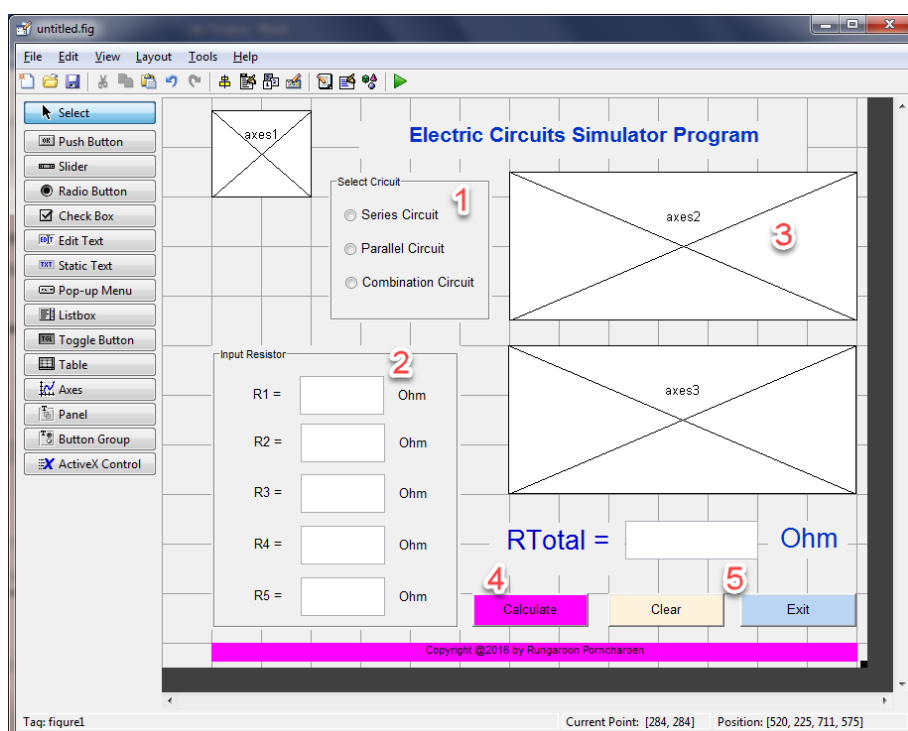
ภาพที่ 9.4 กระบวนการทำงานของโปรแกรม

การทำงานของโปรแกรมการคำนวณวงจรไฟฟ้า สามารถอธิบายได้ ดังนี้ เมื่อเปิดโปรแกรมจะทำการกำหนดค่าเริ่มต้นให้กับโปรแกรมโดยอัตโนมัติ ซึ่งผู้ใช้งานสามารถปรับเปลี่ยนค่าได้ตามต้องการ

โดยการกำหนดค่าความต้านทานแต่ละตัว สำหรับโปรแกรมนี้นี้ มีขอบเขตการใส่ค่าความต้านทานได้ 5 ตัว จากนั้นกดปุ่ม Calculate โดยโปรแกรมจะทำการผลลัพธ์ ซึ่งรายละเอียดการเขียนโปรแกรมสามารถอธิบายได้ ดังนี้

9.1.1 การสร้างโปรแกรมการคำนวณวงจรไฟฟ้า

โปรแกรมการคำนวณวงจรไฟฟ้า เป็นโปรแกรมสำหรับการคำนวณการต่อวงจรความต้านทานแต่ละแบบ ประกอบด้วย การต่อแบบอนุกรม การต่อแบบขนาน และการต่อแบบผสม ซึ่งเป็นพื้นฐานของการต่อวงจรไฟฟ้า โปรแกรมมีการกำหนดตัวรับค่าความต้านทาน จำนวน 5 ตัว และก่อนใช้งานต้องเลือกการต่อวงจรแต่ละแบบ ดังภาพที่ 9.5



ภาพที่ 9.5 การออกแบบลักษณะของโปรแกรมการคำนวณวงจรไฟฟ้า

จากภาพที่ 9.5 สามารถอธิบายขั้นตอนการสร้างโปรแกรมการคำนวณวงจรไฟฟ้าดังนี้

1. การสร้างวัตถุเลือกการต่อวงจร (Select Circuit) ประกอบด้วยวัตถุชนิดต่าง ๆ ดังนี้
 - Panel สำหรับรวมกลุ่มของการต่อวงจร โดยกำหนดชื่อเป็น Select Circuit
 - Radio Button สำหรับใช้เลือกการต่อวงจรแบบต่าง ๆ โดยทำการกำหนดค่าที่ String ของแต่ละวัตถุให้เป็นดังนี้
 - Radio Button1 กำหนดค่าที่ String เป็น SC
 - Radio Button2 กำหนดค่าที่ String เป็น PC
 - Radio Button3 กำหนดค่าที่ String เป็น CC

2. การสร้างวัตถุรับค่าอินพุต (Input Resistor)

วัตถุที่นำมาใช้สำหรับการสร้างวัตถุรับค่าอินพุต (Input) ประกอบด้วยวัตถุชนิดต่าง ๆ ดังนี้

- Panel สำหรับรวมกลุ่มของอินพุตทั้งหมด โดยกำหนดชื่อเป็น Input
- Edit Text สำหรับใช้ในการป้อนค่ากระแส (I) ค่าความต้านทาน (R) โดยทำการกำหนดค่าที่

String ของแต่ละวัตถุให้เป็นค่าเริ่มต้นของโปรแกรม

- Static Text สำหรับใช้ในการกำหนดชื่อและหน่วยของตัวรับค่าอินพุต

3. การสร้างพื้นที่แสดงภาพ ๆ

วัตถุที่นำมาใช้สำหรับการสร้างพื้นที่แสดงภาพต่าง ๆ ดังนี้

- Axes1 ชื่อ axes1 ใช้แสดงภาพโลโก้ของคณะ
- Axes2 ชื่อ axes2 ใช้แสดงภาพภาพวงจร ซึ่งจะเปลี่ยนไปตามการเลือกการต่อวงจร
- Axes3 ชื่อ axes3 ใช้แสดงสมการคำนวณการต่อวงจร ซึ่งจะเปลี่ยนไปตามการเลือกการต่อ

วงจร

4. การสร้างปุ่มกดเพื่อแสดงผลการคำนวณ

วัตถุที่นำมาใช้สำหรับการสร้างปุ่มกดเพื่อแสดงผลการคำนวณ ได้แก่ Push Button โดยทำการกำหนดค่าที่ String ของวัตถุแต่ละวัตถุให้เป็น EE

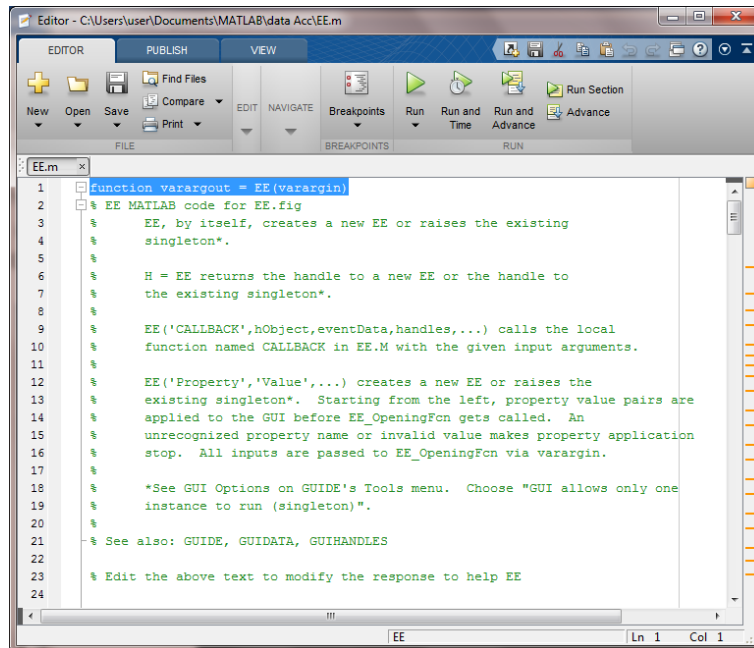
5. การสร้างปุ่มอำนวยความสะดวกอื่น ๆ

การสร้างปุ่มอำนวยความสะดวกอื่น ๆ ประกอบด้วย ปุ่มต่าง ๆ ดังต่อไปนี้

- ปุ่ม Clear เป็นปุ่มที่ใช้สำหรับลบข้อมูลต่าง ๆ ทั้งหมด วัตถุที่นำมาใช้สำหรับสร้าง ได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Clear

- ปุ่ม Exit เป็นปุ่มที่ใช้สำหรับปิดโปรแกรม วัตถุที่นำมาใช้สำหรับสร้าง ได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Exit

เมื่อทำการสร้างหน้าต่างโปรแกรมตามทีออกแบบไว้เรียบร้อยแล้ว ให้ทำการบันทึกโปรแกรมเป็นชื่อ EE จากนั้นโปรแกรม MATLAB จะทำการสร้าง M-File ให้อัตโนมัติ ดังภาพที่ 9.6



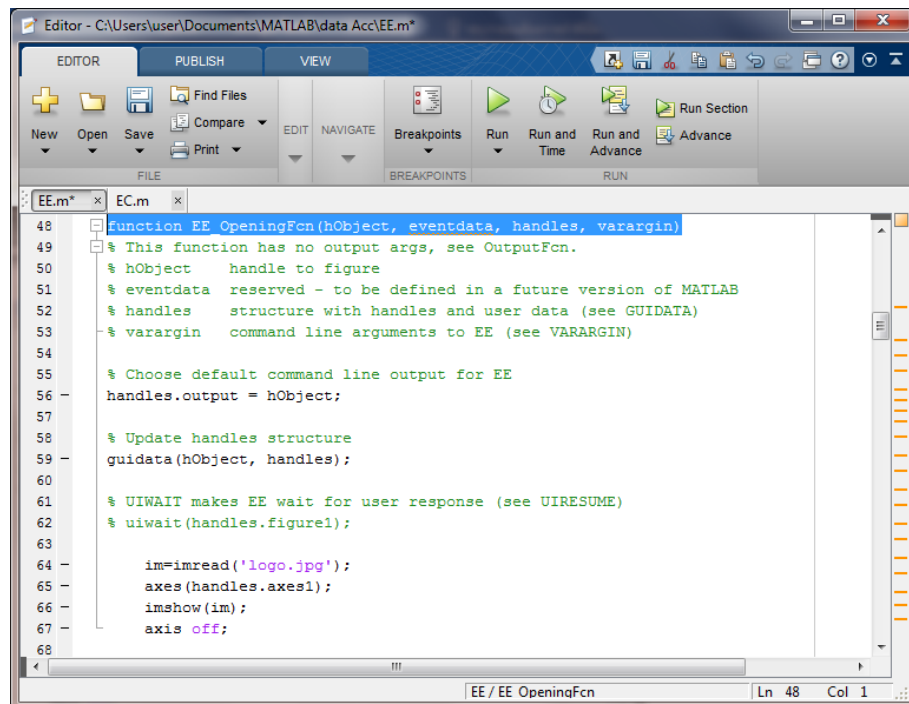
ภาพที่ 9.6 หน้าต่าง M-File ของโปรแกรมการคำนวณวงจรไฟฟ้า

9.1.2 การใส่ภาพโลโก้

เมื่อเลือกการต่อวงจรแบบอนุกรม (Series Circuit) จากหน้าต่างเมนูครั้งแรก ปรากฏดังภาพที่ 9.7 จะพบว่า โปรแกรมยังไม่เป็นไปตามเงื่อนไขของกระบวนการทำงาน คือ โปรแกรมไม่แสดงภาพโลโก้ของคณะ ดังนั้น ต้องมีการเขียนฟังก์ชันการทำงาน เพื่อโปรแกรมจะได้เรียกภาพที่ต้องการ ขึ้นมาบนหน้าต่างตามกระบวนการทำงานที่วางไว้ ดังภาพที่ 9.8

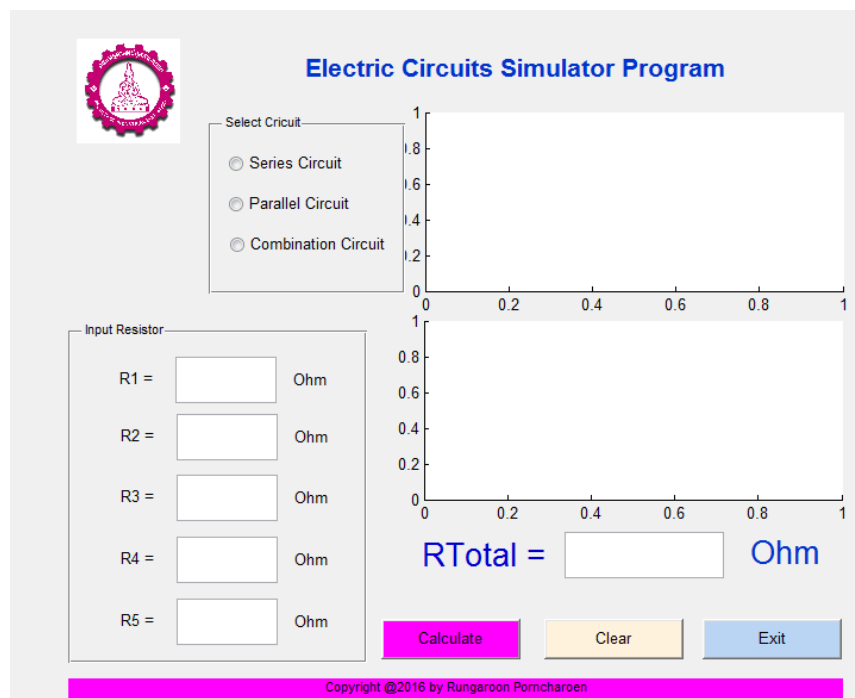


ภาพที่ 9.7 โลโก้ของคณะ



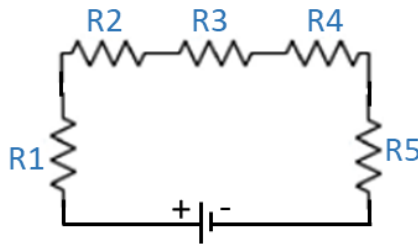
ภาพที่ 9.8 การเขียนฟังก์ชันการทำงานใส่ภาพโลโก้ของคณะ

เมื่อทำการรันโปรแกรมการคำนวณวงจรไฟฟ้า ปรากฏดังภาพที่ 9.9



ภาพที่ 9.9 ผลลัพธ์การใส่โลโก้ของคณะในโปรแกรม

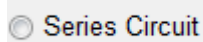
9.1.3 การเลือกการต่อวงจร (Select Circuit) สามารถทำการเลือกได้ 3 แบบ ได้แก่ การต่อวงจรแบบอนุกรม การต่อวงจรแบบขนาน และการต่อวงจรแบบผสม ซึ่งเมื่อเลือกการต่อวงจรแล้ว โปรแกรมจะแสดงภาพการต่อวงจรแบบอนุกรมและสมการการคำนวณ สำหรับในบทที่จะกล่าวถึงการต่อวงจรแบบอนุกรม ดังภาพที่ 9.10



$$R_T = R_1 + R_2 + R_3 + \cdots + R_n$$

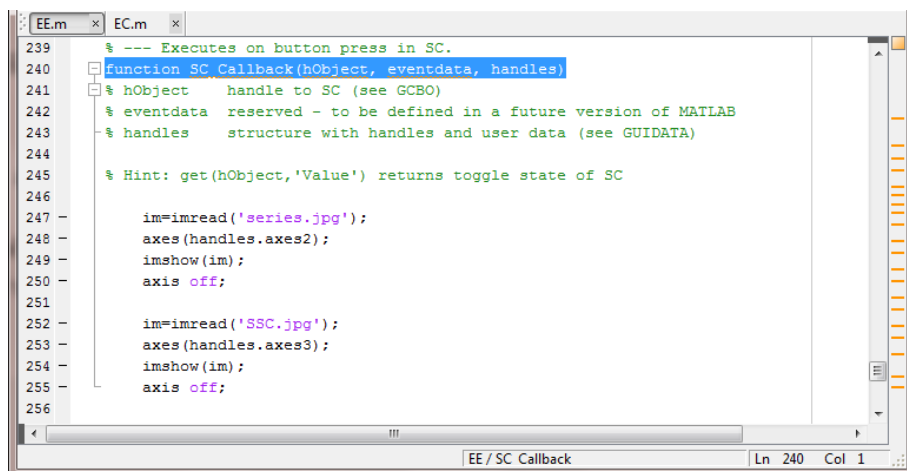
ภาพที่ 9.10 การต่อวงจรแบบอนุกรมและสมการการคำนวณ

จากนั้นทำการเขียนโปรแกรมเพื่อควบคุมวัตถุการเลือกการต่อวงจร ดังนี้

1. การต่อวงจรแบบอนุกรม (Series Circuit) ให้คลิกขวาที่  แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

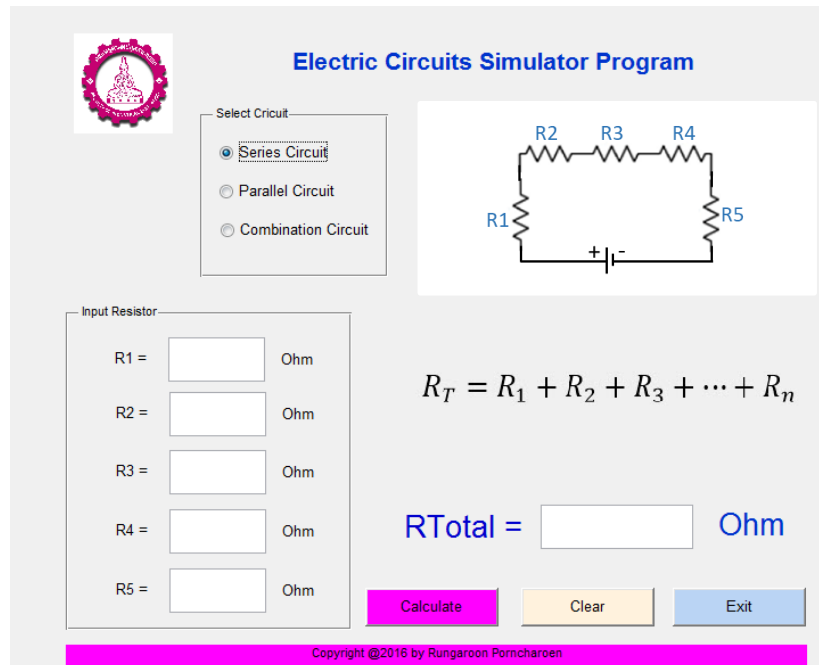
`function SC_Callback(hObject, eventdata, handles)`

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 9.11



ภาพที่ 9.11 การเขียนโปรแกรมเพื่อเลือกการต่อวงจรแบบอนุกรม

2. เมื่อทำการรันโปรแกรมการคำนวณวงจรไฟฟ้าจะปรากฏภาพการต่อวงจรแบบอนุกรมและสมการการคำนวณ ดังภาพที่ 9.12

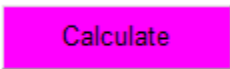


ภาพที่ 9.12 ผลการจำลองโปรแกรมการคำนวณวงจรไฟฟ้า

9.2 การแทรกภาพในหน้าต่างโปรแกรมการคำนวณวงจรไฟฟ้า

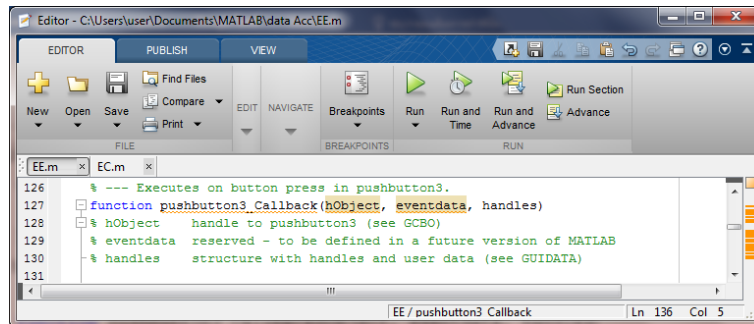
สำหรับการเขียนโปรแกรมเพื่อควบคุมวัตถุคำนวณค่ากระแส ตามกระบวนการทำงานที่วางไว้ นั้น ประกอบด้วย ปุ่ม Calculate ปุ่ม Clear และปุ่ม Exit มีขั้นตอนการเขียนโปรแกรมดังนี้

- ปุ่ม Calculate

1. คลิกขวาที่ปุ่ม  แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

`function pushbutton3_Callback(hObject, eventdata, handles)`

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 9.13



ภาพที่ 9.13 การเขียนโปรแกรมการต่อวงจรแบบอนุกรมที่ปุ่ม Calculate

การเขียนโปรแกรมการต่อวงจรแบบอนุกรมที่ปุ่ม Calculate

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
hh=findobj(gcf,'Tag','r1');
r1=get(hh,'string');
if isempty(r1)
    r1=0;
else
    r1=str2double(get(hh,'string'));
end
```

```
hh=findobj(gcf,'Tag','r2');
r2=get(hh,'string');
if isempty(r2)
    r2=0;
else
    r2=str2double(get(hh,'string'));
end
```

```
hh=findobj(gcf,'Tag','r3');
r3=get(hh,'string');
```

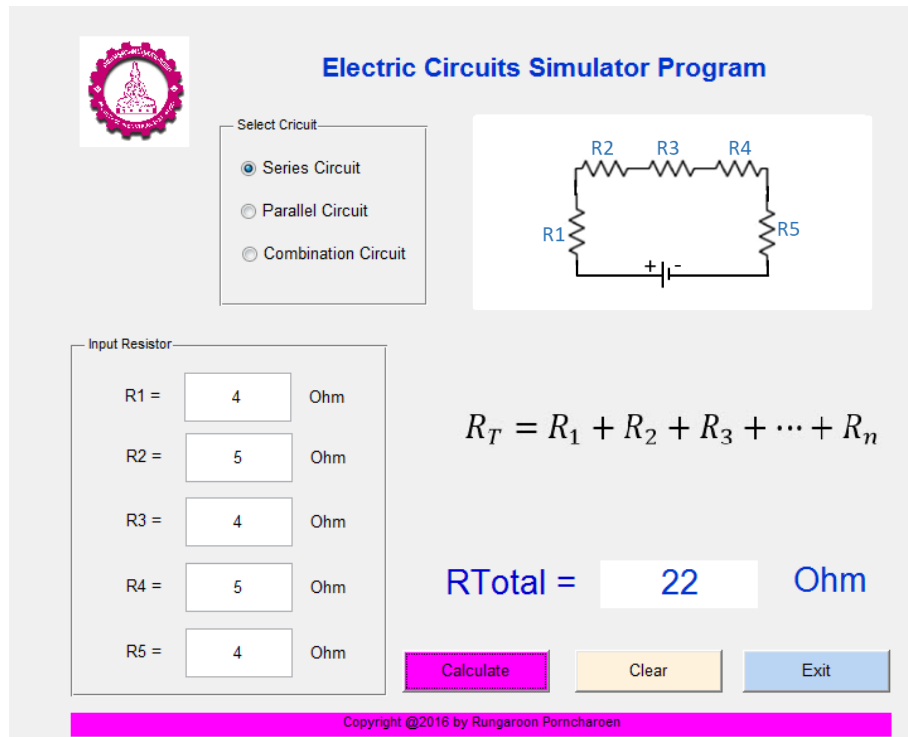
```
if isempty(r3)
    r3=0;
else
    r3=str2double(get(hh,'string'));
end

hh=findobj(gcf,'Tag','r4');
r4=get(hh,'string');
if isempty(r4)
    r4=0;
else
    r4=str2double(get(hh,'string'));
end

hh=findobj(gcf,'Tag','r5');
r5=get(hh,'string');
if isempty(r5)
    r5=0;
else
    r5=str2double(get(hh,'string'));
end

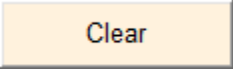
rt = r1+r2+r3+r4+r5;set(handles.rt,'String',rt)
```

2. เมื่อทำการรันโปรแกรมการคำนวณวงจรไฟฟ้า แล้วทำการใส่ค่าความต้านทาน 5 ตัว จากนั้นกดปุ่ม Calculate จะแสดงผลลัพธ์การคำนวณ ดังภาพที่ 9.14



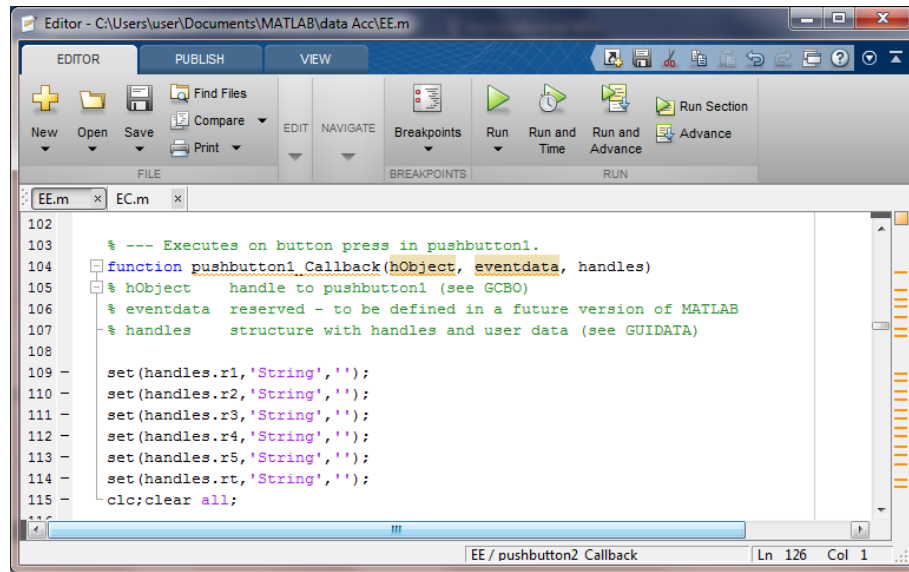
ภาพที่ 9.14 ผลการจำลองโปรแกรมการต่อวงจรแบบอนุกรม

- ปุ่ม Clear

1. คลิกขวาที่ปุ่ม  แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

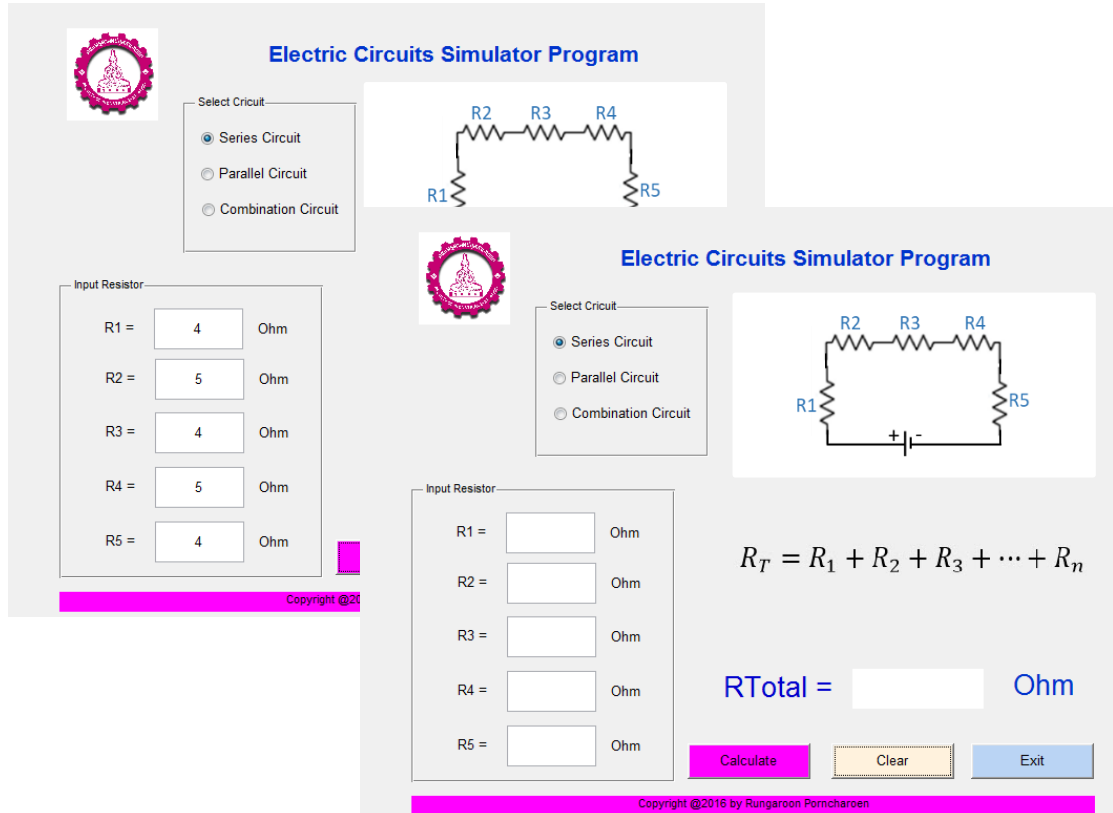
`function pushbutton1_Callback(hObject, eventdata, handles)`

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 9.15



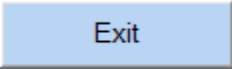
ภาพที่ 9.15 การเขียนโปรแกรมลบค่าต่าง ๆ

2. เมื่อทำการรันโปรแกรมการต่อวงจรแบบอนุกรมแล้ว หากต้องการลบข้อมูลที่ใส่ไปสามารถทำได้โดยการกดปุ่ม Clear ดังภาพที่ 9.16



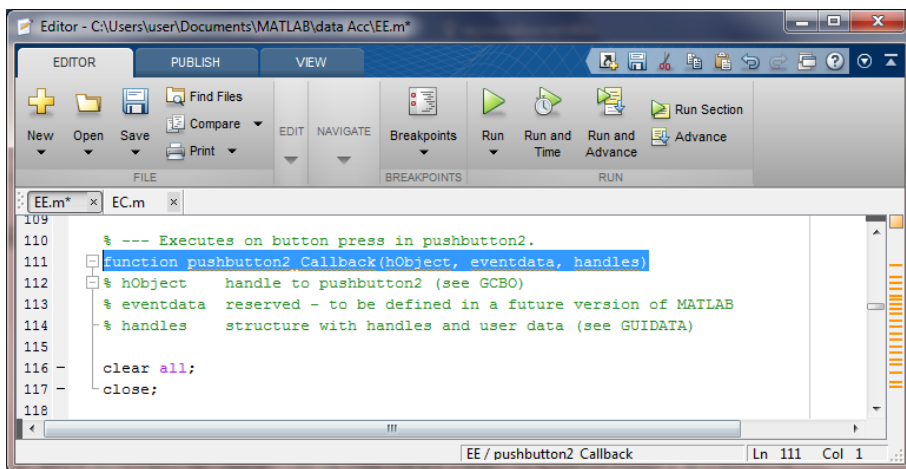
ภาพที่ 9.16 ผลการรันปุ่ม Clear

- ปุ่ม Exit

1. คลิกขวาที่ปุ่ม  แล้วเลือก View Callbacks > Callback จากนั้นจะปรากฏ M-File Editor ที่

`function pushbutton2_Callback(hObject, eventdata, handles)`

แล้วให้เขียนโปรแกรม จากนั้นทำการบันทึก ดังภาพที่ 9.17



ภาพที่ 9.17 การเขียนโปรแกรมปิดหน้าต่างเมนู

2. เมื่อทำการรันโปรแกรมการต่อวงจรแบบอนุกรมแล้ว หากต้องการออกจากโปรแกรมทั้งหมด สามารถทำได้โดยการกดปุ่ม Exit

9.3 สรุป

การพัฒนาโปรแกรมการคำนวณวงจรไฟฟ้า เป็นการประยุกต์ใช้ GUI ของโปรแกรม MATLAB มีกระบวนการดำเนินงานดังนี้ เมื่อเปิดโปรแกรมจะทำการกำหนดค่าเริ่มต้นให้กับโปรแกรมโดยอัตโนมัติ ซึ่งผู้ใช้งานสามารถปรับเปลี่ยนค่าได้ตามต้องการ โดยการกำหนดค่าความต้านทานแต่ละตัว สำหรับโปรแกรมนี้นี้ มีขอบเขตการใส่ค่าความต้านทานได้ 5 ตัว จากนั้นกดปุ่ม Calculate โดยโปรแกรมจะทำการผลลัพธ์

แบบฝึกหัด

บทที่ 9 การประยุกต์ใช้ GUI ของ MATLAB สำหรับโปรแกรมการคำนวณวงจรไฟฟ้า

1. จงอธิบายขั้นตอนการวิเคราะห์สำหรับการพัฒนา GUI ของโปรแกรมคำนวณวงจรไฟฟ้า
2. จงออกแบบโปรแกรมจำลองกระบวนการทำงานของโปรแกรมคำนวณวงจรไฟฟ้า
3. จงบอกวัตถุที่ใช้ในการออกแบบโปรแกรมคำนวณการต่อวงจรแบบอนุกรมของหน้าจอเมนูหลัก
4. จงอธิบายการทำงานและการเขียนโปรแกรมของปุ่มคำนวณของโปรแกรมคำนวณการต่อวงจรแบบอนุกรม
5. จงเพิ่มปุ่มคู่มือการใช้งานไว้ที่หน้าจอเมนูหลักของโปรแกรมคำนวณการต่อวงจรแบบอนุกรม
6. จงอธิบายหน้าที่การทำงานของปุ่ม Exit พร้อมเขียนโปรแกรม
7. จงออกแบบและเขียนโปรแกรมคำนวณการต่อวงจรแบบขนาน พร้อมทั้งอธิบายรายละเอียดของการสร้าง
8. จงออกแบบและเขียนโปรแกรมคำนวณการต่อวงจรแบบผสม พร้อมทั้งอธิบายรายละเอียดของการสร้าง โดยมีเงื่อนไขดังนี้
 - 8.1 มีหน้าต่างเมนูหลัก
 - 8.2 มีการแสดงภาพในแต่ละเมนู
 - 8.3 มีปุ่มคำนวณ แสดงผลบนโปรแกรม และปุ่มปิดโปรแกรม
9. จงออกแบบและเขียนโปรแกรมจำลองระบบการมอดูเลตเชิงขนาดและเชิงความถี่ พร้อมทั้งอธิบายรายละเอียดของการสร้าง โดยมีเงื่อนไขดังนี้
 - 9.1 มีหน้าต่างเมนูหลัก
 - 9.2 มีการแสดงภาพในแต่ละเมนู
 - 9.3 มีปุ่มคำนวณ แสดงผลบนโปรแกรม และปุ่มปิดโปรแกรม
10. จงออกแบบและเขียนโปรแกรมจำลองการวิเคราะห์ค่าพารามิเตอร์พื้นฐานของสายอากาศ พร้อมทั้งอธิบายรายละเอียดของการสร้าง โดยมีเงื่อนไขดังนี้
 - 10.1 มีหน้าต่างเมนูหลัก
 - 10.2 มีการแสดงภาพในแต่ละเมนู
 - 10.3 มีปุ่มคำนวณ แสดงผลบนโปรแกรม และปุ่มปิดโปรแกรม

บทที่ 10

การประยุกต์ใช้ GUI ของ MATLAB สำหรับ การวิเคราะห์วงจรอันดับหนึ่ง RL และ RC

บทนี้เป็นการอธิบายขั้นตอนการพัฒนาหน้าต่างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL และ RC ซึ่งเป็นเครื่องมือพื้นฐานที่ใช้ในการวิเคราะห์วงจรไฟฟ้าจากกฎของโอห์มที่กล่าวไว้ว่า กระแสที่เกิดขึ้นในวงจรไฟฟ้าจะมีค่าแปรผันตรงกับค่าแรงดันและแปรผกผันกับค่าความต้านทาน ส่วนกฎแรงดันของเคอร์ชอฟฟ์ให้หลักการไว้ว่าผลรวมของแรงดันที่ตกคร่อมอุปกรณ์ในวงจรไฟฟ้ารวมกันมีค่าเท่ากับศูนย์ และกฎกระแสของเคอร์ชอฟฟ์ให้หลักการไว้ว่าผลรวมของกระแสที่จุดใด ๆ ของวงจรไฟฟ้ารวมกันมีค่าเท่ากับศูนย์ หรือกระแสไหลเข้าจุดต่อวงจรเท่ากับกระแสไหลออกจากจุดต่อวงจร โดยที่กฎพื้นฐานเหล่านี้จะนำมาเป็นเครื่องมือในการวิเคราะห์วงจรไฟฟ้าเริ่มต้น

การวิเคราะห์วงจรไฟฟ้ากระแสตรง เป็นพื้นฐานในการเรียนรู้เกี่ยวกับผลการตอบสนองของตัวสะสมพลังงานของอุปกรณ์ ได้แก่ ตัวต้านทาน ตัวเหนี่ยวนำ และตัวเก็บประจุ พบว่า มีแต่ตัวต้านทานเท่านั้นที่ไม่สามารถสะสมพลังงานไฟฟ้าได้ ส่วนตัวเหนี่ยวนำและตัวเก็บประจุสามารถสะสมพลังงานได้ โดยตัวเหนี่ยวนำเก็บสะสมพลังงานอยู่ในรูปสนามแม่เหล็กและตัวเก็บประจุเก็บสะสมพลังงานอยู่ในรูปสนามไฟฟ้า โดยที่พลังงานที่อุปกรณ์ทั้งสองสะสมไว้จ่ายคืนให้กับวงจรภายหลังจากแหล่งกำเนิดพลังงานของวงจรมีค่าเป็นศูนย์ เมื่อพิจารณาวงจรที่ประกอบด้วยตัวเก็บประจุและตัวเหนี่ยวนำสามารถพิจารณาได้จากสมการอนุพันธ์ การสะสมพลังงานของตัวเก็บประจุขึ้นอยู่กับแรงดัน ทำให้การสะสมหรือการคายประจุไม่สามารถทำได้แบบฉับพลัน เช่นเดียวกับในกรณีของการสะสมและการคายพลังงานของตัวเหนี่ยวนำที่ขึ้นอยู่กับกระแสไม่สามารถทำได้แบบฉับพลันเช่นเดียวกัน ดังนั้นจึงต้องมีการพิจารณาค่ากระแสและแรงดันของวงจรตัวต้านทานกับตัวเก็บประจุและตัวต้านทานกับตัวเหนี่ยวนำในสถานะที่ตัวเก็บประจุ หรือตัวเหนี่ยวนำสะสม หรือคายพลังงานที่เก็บสะสมไว้ หรือการวิเคราะห์สภาวะชั่วคราวของกระแสและแรงดัน ซึ่งสามารถพิจารณาได้จากสมการอนุพันธ์อันดับหนึ่ง สำหรับวงจรที่ประกอบด้วยตัวต้านทานกับตัวเหนี่ยวนำ (RL) หรือวงจรที่ประกอบด้วยตัวต้านทานกับตัวเก็บประจุ (RC) สามารถพิจารณาได้จากสมการอนุพันธ์อันดับสอง

ก่อนเริ่มเขียนโปรแกรมการวิเคราะห์วงจรอันดับหนึ่ง RL และ RC มาทำศึกษารายละเอียดเนื้อหาทฤษฎีที่สามารถนำมาพัฒนาโปรแกรมจำลอง ดังนี้

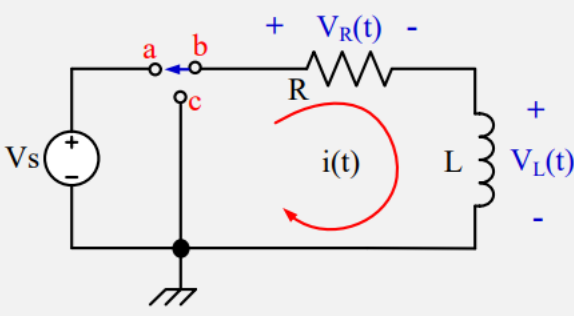
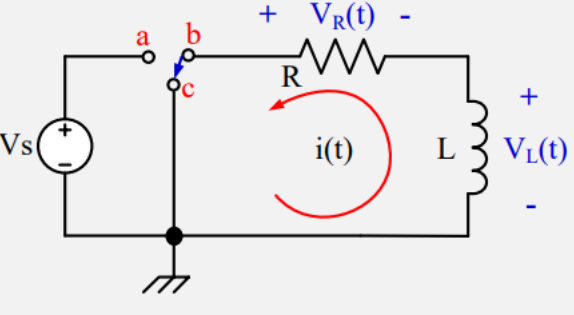
ตัวเหนี่ยวนำ (RL) และตัวเก็บประจุ (RC) เป็นอุปกรณ์ไฟฟ้าชนิดสองขั้ว ซึ่งแตกต่างจากตัวต้านทานทางไฟฟ้า ในแง่ที่อุปกรณ์ทั้งสองสามารถเก็บสะสมพลังงานที่ได้รับจากวงจรไฟฟ้าได้ และปลดปล่อยพลังงานที่สะสมดังกล่าวคืนกลับสู่วงจรไฟฟ้าได้ในสถานะที่เหมาะสม ในกรณีของตัวเหนี่ยวนำพลังงานจะสะสมอยู่ในรูปของสนามแม่เหล็ก ส่วนในกรณีของตัวเก็บประจุพลังงานจะสะสมอยู่ในรูปของสนามไฟฟ้า ซึ่งสามารถวิเคราะห์สมการอนุพันธ์ของวงจรอันดับหนึ่ง RL และ RC ที่จะนำไปใช้ในส่วนของการประมวลผลสำหรับ

การพัฒนา GUI ของ MATLAB โดยมีขั้นตอนการวิเคราะห์ดังนี้

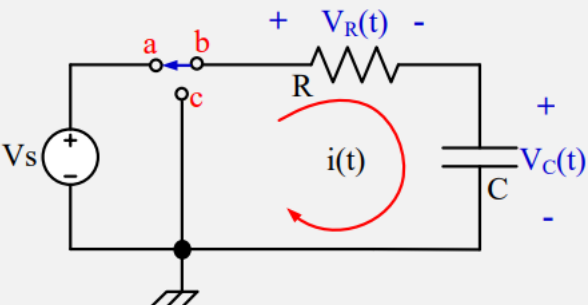
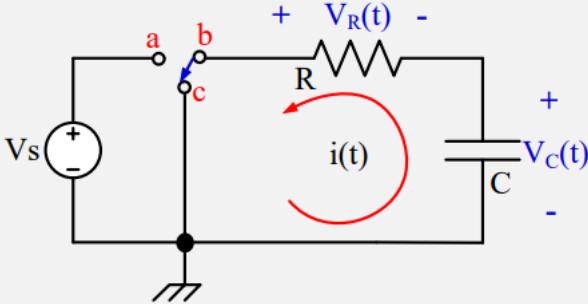
- 1) พิจารณาวงจรและใช้กฎของเคอร์ชอฟฟ์ในการเขียนสมการ
- 2) แทนค่าตัวแปรและจัดรูปสมการให้อยู่ในรูปของสมการอนุพันธ์
- 3) แก้สมการอนุพันธ์โดยใช้วิธีพื้นฐานทางคณิตศาสตร์

ผลการวิเคราะห์แสดงดังตารางที่ 10.1

ตารางที่ 10.1 การวิเคราะห์วงจรอันดับหนึ่ง RL และ RC

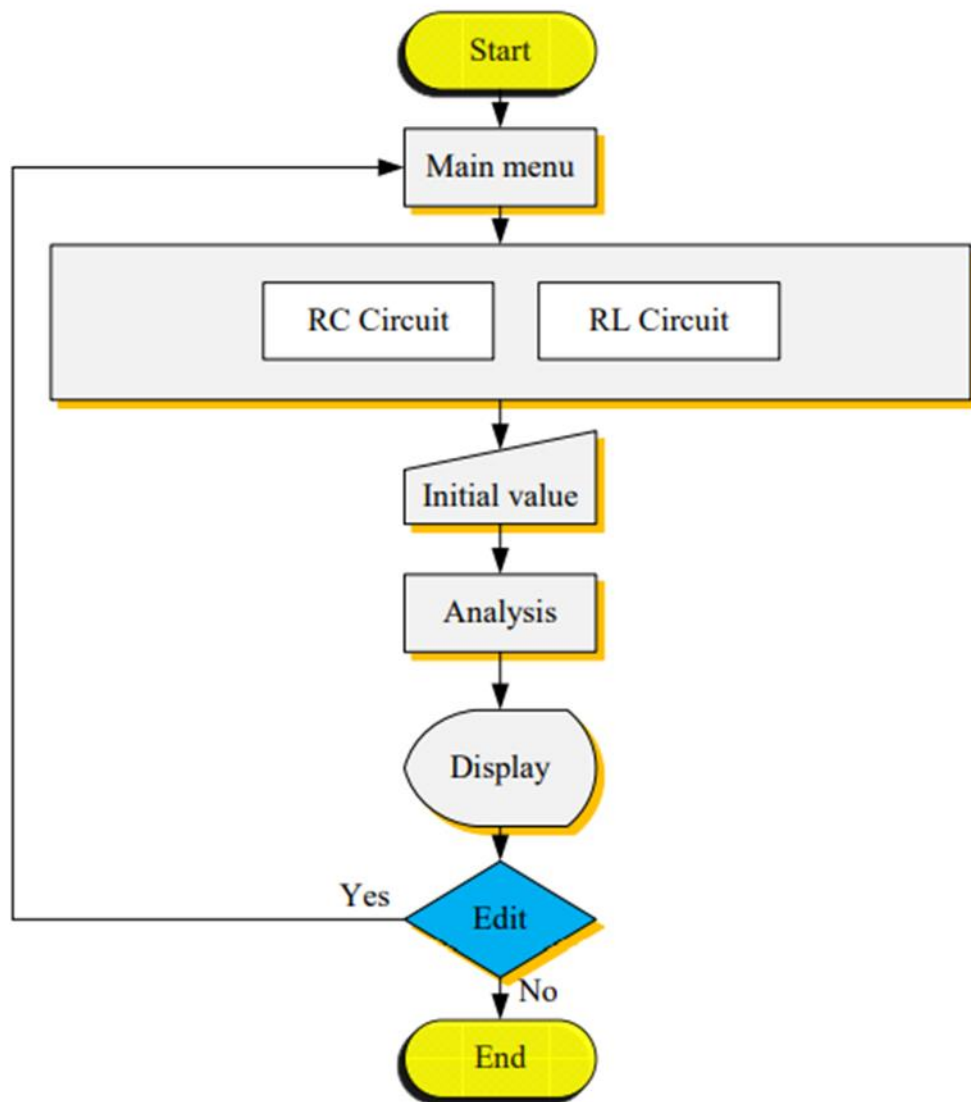
ชนิดของวงจร	สมการ
ตัวต้านทานกับตัวเหนี่ยวนำ (RL)	
- ช่วงเก็บพลังงาน	
	สมการแรงดัน $v_L(t) = E e^{-t/\tau}$ $v_R(t) = E e^{t/\tau}$ สมการกระแส $i(t) = \frac{E}{R} (1 - e^{-t/\tau})$
- ช่วงคายพลังงาน	
	สมการแรงดัน $v_L(t) = -E e^{-t/\tau}$ $v_R(t) = E e^{t/\tau}$ เมื่อ $\tau = L/R$ สมการกระแส $i(t) = \frac{E}{R} e^{-t/\tau}$ เมื่อ $\tau = L/R$

ตารางที่ 10.1 การวิเคราะห์ห้วงจรอันดับหนึ่ง RL และ RC (ต่อ)

ชนิดของวงจร	สมการ
ตัวต้านทานกับตัวเก็บประจุ (RC)	
- ช่วงเก็บพลังงาน	
	สมการแรงดัน $v_R(t) = E e^{-t/\tau}$ $v_C(t) = E(1 - e^{-t/\tau})$ สมการกระแส $i(t) = \frac{E}{R} e^{-t/\tau}$
- ช่วงคายพลังงาน	
	สมการแรงดัน $v_R(t) = -E e^{-t/RC}$ $= -E e^{-t/\tau}$ $v_C(t) = E e^{-t/RC}$ $= E e^{-t/\tau}$ เมื่อ $\tau = RC$ สมการกระแส $i(t) = -\frac{E}{R} e^{-t/\tau}$ เมื่อ $\tau = RC$

10.1 การออกแบบโปรแกรมจำลอง

การออกแบบโปรแกรมจำลองจะทำการวิเคราะห์อินพุต เอาต์พุต และกระบวนการทำงาน (Input, Output and Process analysis) ของเนื้อหาในแต่ละเรื่อง ซึ่งสามารถแสดงกระบวนการทำงานของโปรแกรมจำลองดังภาพที่ 10.1



ภาพที่ 10.1 การออกแบบโปรแกรมจำลอง

จากภาพที่ 10.1 เริ่มต้นด้วยการเข้าสู่หน้าต่างหลัก (Main menu) ของโปรแกรมจำลอง จากนั้นเลือกเนื้อหาที่จะทำการจำลอง กำหนดค่าเริ่มต้น (Initial value) เข้าสู่กระบวนการวิเคราะห์ (Analysis) แสดงผลการจำลอง (Display) และผู้ใช้สามารถปรับเปลี่ยนค่าของตัวต้านทาน ตัวเหนี่ยวนำ และตัวเก็บประจุได้ตามต้องการ

10.2 การพัฒนาโปรแกรมจำลอง

การพัฒนาโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL และ RC ที่ได้จากการออกแบบในหัวข้อที่ 10.1 โดยทำการสร้างส่วนการติดต่อกับผู้ใช้งาน (Graphic User Interface : GUI) ในส่วนของอินพุตและเอาต์พุต และส่วนของการประมวลผลจะเขียนอัลกอริทึมโดยใช้เอ็มไฟล์(M-File) ของโปรแกรม MATLAB จากสมการทางคณิตศาสตร์ที่แสดงไว้ดังกล่าวข้างต้น แสดงรายละเอียดดังนี้



ภาพที่ 10.2 การสร้างหน้าจอหลัก

จากภาพที่ 10.2 สามารถอธิบายรายละเอียดแต่ละขั้นตอนได้ ดังนี้

1. การกำหนดชื่อโปรแกรม

สามารถสร้างโดยใช้วัตถุชนิด Static Text โดยการลากวัตถุดังกล่าวมาวางไว้ที่หน้าต่างการออกแบบ แล้วทำการดับเบิลคลิกที่วัตถุชนิด Static Text จะปรากฏหน้าต่างการกำหนดคุณลักษณะ (Inspector) หลังจากนั้นให้กำหนด String เป็น Simulation Program for Response Analysis of First with RLC

2. การกำหนดชื่อของมหาวิทยาลัย

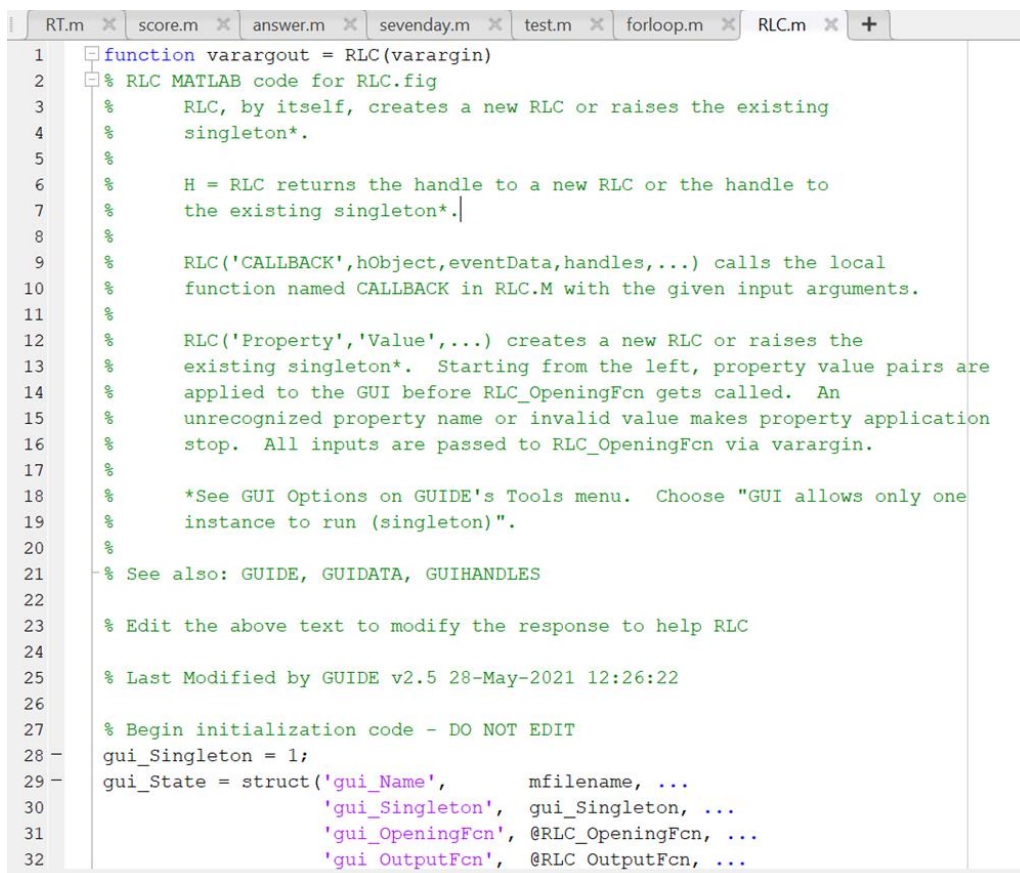
สามารถสร้างได้เหมือนกับการกำหนดชื่อของโปรแกรม แต่ให้กำหนด String เป็นชื่อมหาวิทยาลัย (Rajamangala University of Technology Phra Nakhon)

3. การสร้างปุ่มเชื่อมโยงหน้าต่างของโปรแกรมน้อย สามารถสร้างโดยใช้วัตถุชนิด Push Button 2 วัตถุ และใช้วัตถุชนิด Panel เพื่อรวมกลุ่มของ Push Button ทั้ง 2 เป็นเมนูให้ผู้ใช้โปรแกรมได้เลือก ทำการเปลี่ยนชื่อปุ่ม ดังนี้ ปุ่มที่ 1 RL Circuit และปุ่มที่ 2 RC Circuit โดยการดับเบิลคลิกบนวัตถุ Push Button เปลี่ยนชื่อที่ String เป็นชื่อโปรแกรมน้อยต่าง ๆ

4. การสร้างปุ่มให้ความช่วยเหลือและปุ่มปิด สามารถสร้างโดยใช้วัตถุชนิด Push Button 2 วัตถุ และใช้วัตถุชนิด Panel เพื่อรวมกลุ่มของ Push Button ทั้ง 2 เป็นเมนูให้ผู้ใช้โปรแกรมได้เลือก โดยมีขั้นตอนการสร้างเหมือนกับหัวข้อที่ 3 แต่เปลี่ยนที่ String ของวัตถุ Push Button เป็น Help และ close ตามลำดับ และชื่อของวัตถุชนิด Panel ให้ลบออก (ไม่ต้องกำหนดชื่อ)

5. การแสดงภาพโลโก้ สามารถสร้างโดยใช้วัตถุชนิด Axes โดยการคลิกซ้ายที่วัตถุ Axes ค้างไว้ แล้วลากมาวางบน หน้าต่างออกแบบตามตำแหน่งที่กำหนดไว้

6. การกำหนดรายละเอียดอื่น ๆ ให้กับหน้าต่างเมนูหลักโดยใช้ข้อความ สามารถกระทำเช่นเดียวกันกับหัวข้อที่ 1 และ 2 เมื่อทำการสร้างหน้าต่างเมนูหลักตามที่ออกแบบไว้เรียบร้อยแล้ว ให้ทำการบันทึกโปรแกรมเป็น ชื่อ RLC เมื่อทำการบันทึกเสร็จ โปรแกรม MATLAB จะทำการกำหนด M-file ให้อัตโนมัติ แสดงดังภาพที่ 10.3



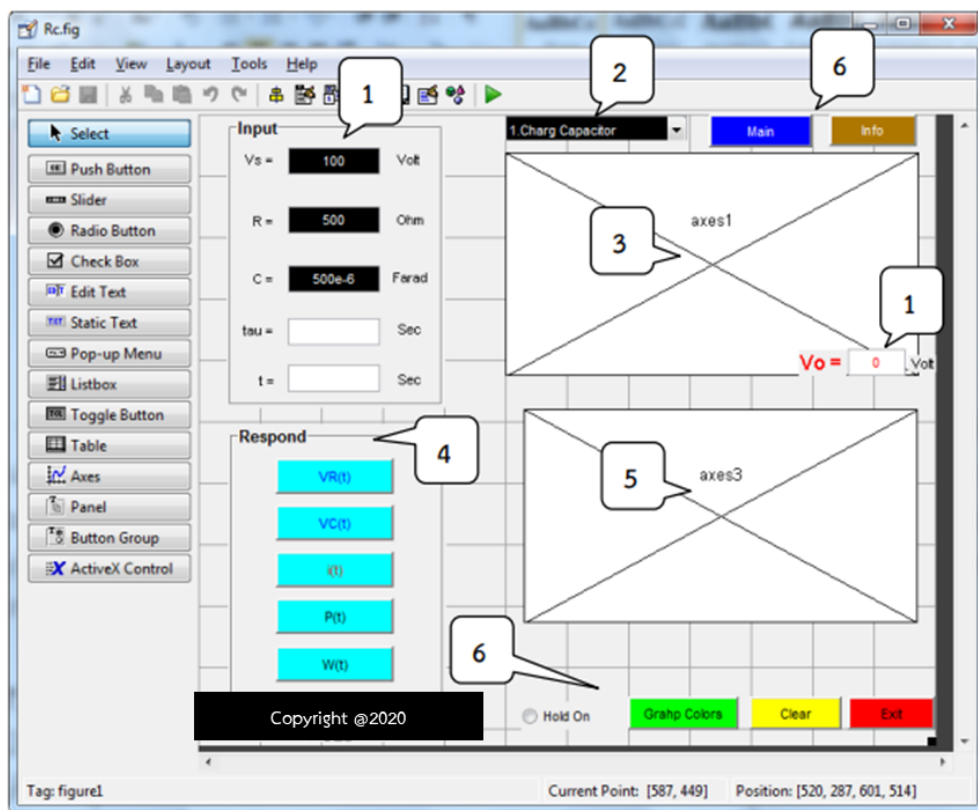
```

1 function varargout = RLC(varargin)
2 % RLC MATLAB code for RLC.fig
3 % RLC, by itself, creates a new RLC or raises the existing
4 % singleton*.
5 %
6 % H = RLC returns the handle to a new RLC or the handle to
7 % the existing singleton*.|
8 %
9 % RLC('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in RLC.M with the given input arguments.
11 %
12 % RLC('Property','Value',...) creates a new RLC or raises the
13 % existing singleton*. Starting from the left, property value pairs are
14 % applied to the GUI before RLC_OpeningFcn gets called. An
15 % unrecognized property name or invalid value makes property application
16 % stop. All inputs are passed to RLC_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help RLC
24
25 % Last Modified by GUIDE v2.5 28-May-2021 12:26:22
26
27 % Begin initialization code - DO NOT EDIT
28 gui_Singleton = 1;
29 gui_State = struct('gui_Name', mfilename, ...
30 'gui_Singleton', gui_Singleton, ...
31 'gui_OpeningFcn', @RLC_OpeningFcn, ...
32 'gui_OutputFcn', @RLC_OutputFcn, ...
  
```

ภาพที่ 10.3 M-file ของหน้าต่างเมนูหลัก

10.3 การสร้างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL

โปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL เป็นโปรแกรมย่อยสำหรับวิเคราะห์ผลตอบสนองของวงจร RL ไฟฟ้ากระแสตรง 3 สถานะ คือ ค่าตัวต้านทานบริสุทธิ์ (Pure Resistor Circuit) ค่าตัวเหนี่ยวนำบริสุทธิ์ (Pure Inductor Circuit) และค่าตัวต้านทานและค่าตัวเหนี่ยวนำบริสุทธิ์ (Pure Resistor and Inductor Circuit) โดยการกำหนดอินพุต ได้แก่ แรงดันที่แหล่งกำเนิด (V_s) ค่าความต้านทาน (R) ค่าความเหนี่ยวนำ (L) ส่วนค่าคงที่ของเวลา (τ) และค่าเวลา (t) โปรแกรมจะคำนวณให้อัตโนมติ ส่วนของผลตอบสนอง (Respond) จะประกอบด้วย แรงดัน กระแส กำลังและพลังงานที่เกิดขึ้นในวงจรแสดงลักษณะของโปรแกรกดังภาพที่ 10.4



ภาพที่ 10.4 การออกแบบหน้าจอโปรแกรมย่อยการวิเคราะห์วงจรอันดับหนึ่ง RL

จากภาพที่ 10.4 สามารถอธิบายขั้นตอนการสร้างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL ไฟฟ้ากระแสตรง ดังนี้

1. การสร้างวัตถุรับค่าอินพุต (Input) วัตถุที่นำมาใช้สำหรับการสร้างตัวรับค่าอินพุต (Input) ประกอบด้วยวัตถุชนิดต่าง ๆ ดังนี้

- Panel สำหรับรวมกลุ่มของอินพุตทั้งหมด โดยกำหนดชื่อเป็น Input

- Edit Text สำหรับใช้ในการป้อนค่าแรงดัน (Vs) ค่าความต้านทาน (R) ค่าความเหนี่ยวนำ (L) และไว้แสดงค่าคงที่ของเวลา (tau) และค่าเวลา (t) โดยทำการกำหนดค่าที่ String ของแต่ละวัตถุให้เป็นค่าเริ่มต้นของโปรแกรม

- Static Text สำหรับใช้ในการกำหนดชื่อและหน่วยของตัวรับค่าอินพุต

2. การสร้างวัตถุเลือกสถานะของวงจร วัตถุที่นำมาใช้สำหรับการสร้างตัวเลือกสถานะของวงจร ได้แก่ Pop-up Menu โดยทำการเปลี่ยนรายการที่ String เป็นดังนี้

- Pure Resistor Circuit

- Pure Inductor Circuit

- Pure R and L Circuit

3. การสร้างพื้นที่แสดงภาพของวงจรในสถานะต่าง ๆ วัตถุที่นำมาใช้สำหรับการสร้างพื้นที่แสดงภาพของวงจรในสถานะต่าง ๆ ได้แก่ Axes ชื่อ axes1

4. การสร้างปุ่มกดเพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ วัตถุที่นำมาใช้สำหรับการสร้างปุ่มกดเพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ ประกอบด้วยวัตถุชนิดต่าง ๆ ดังนี้

- Panel สำหรับรวมกลุ่มของผลตอบสนองทั้งหมด โดยกำหนดชื่อเป็น Respond

- Push Button สำหรับใช้เป็นปุ่มกดเมื่อต้องการให้แสดงผลตอบสนองของวงจร โดยทำการกำหนดค่าที่ String ของแต่ละวัตถุให้เป็นดังนี้

1. Push Button1 กำหนดค่าที่ String เป็น VR(t)

2. Push Button2 กำหนดค่าที่ String เป็น VL(t)

3. Push Button3 กำหนดค่าที่ String เป็น i(t)

4. Push Button4 กำหนดค่าที่ String เป็น P(t)

5. Push Button5 กำหนดค่าที่ String เป็น W(t)

5. การสร้างพื้นที่เพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ วัตถุที่นำมาใช้สำหรับการสร้างพื้นที่เพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ ได้แก่ Axes ชื่อ axes3

6. การสร้างปุ่มอำนวยความสะดวกอื่น ๆ การสร้างปุ่มอำนวยความสะดวกอื่น ๆ ประกอบด้วยปุ่มต่าง ๆ ดังต่อไปนี้

- ปุ่ม Main เป็นปุ่มที่ใช้เรียกกลับไปยังหน้าต่างเมนูหลักของโปรแกรม วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Main

- ปุ่ม Info เป็นปุ่มที่ใช้เรียกกลับไปยังหน้าต่างแสดงรายละเอียดของโปรแกรม วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Info

- ปุ่ม Graph Colors เป็นปุ่มที่ใช้เรียกกลับไปยังหน้าต่างการกำหนดสีของกราฟ วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Graph Colors

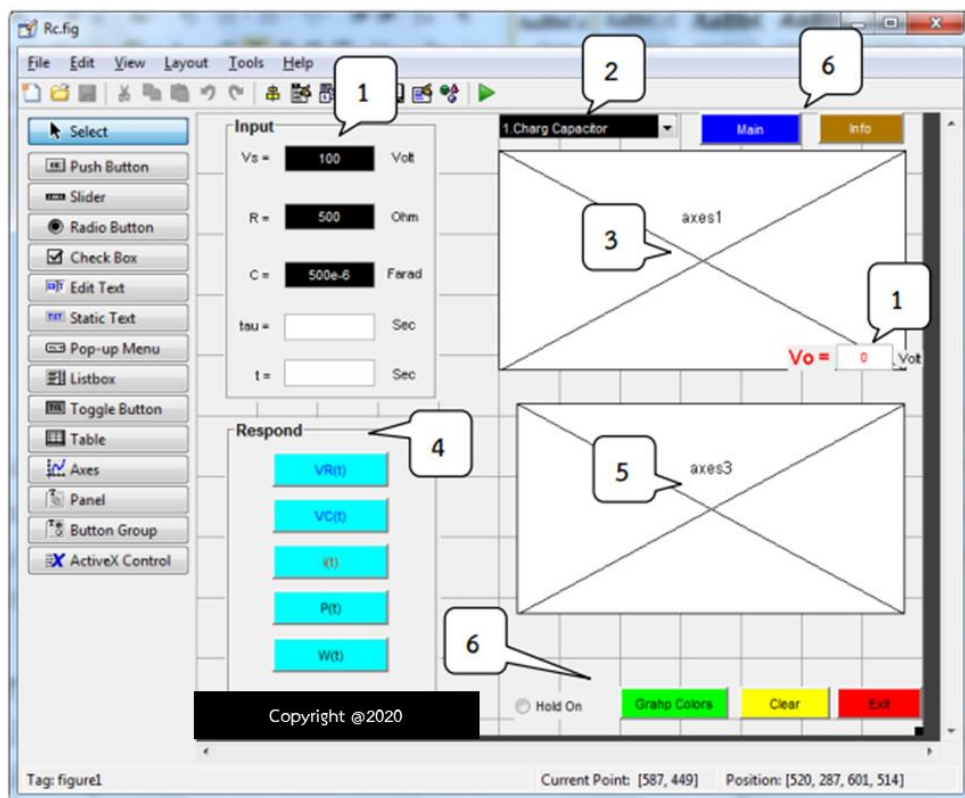
- ปุ่ม Clear เป็นปุ่มที่ใช้สำหรับลบกราฟ วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Clear

- ปุ่ม Exit เป็นปุ่มที่ใช้สำหรับปิดโปรแกรม วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Exit

- ปุ่ม Hold On เป็นปุ่มที่ใช้สำหรับการวาดกราฟมากกว่าสองกราฟ วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Radio Button โดยให้กำหนดค่าที่ String เป็น Hold On เมื่อทำการสร้างหน้าต่างโปรแกรมตามทีออกแบบไว้เรียบร้อยแล้ว ให้ทำการบันทึกโปรแกรมเป็น ชื่อ RLL เมื่อทำการบันทึกเสร็จ โปรแกรม MATLAB จะทำการกำหนด M-file ให้อัตโนมัติ

10.4 การสร้างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RC

โปรแกรมวิเคราะห์วงจร RC เป็นโปรแกรมย่อยสำหรับวิเคราะห์ผลตอบสนองของวงจร RC ไฟฟ้ากระแสตรง 3 สถานะ คือ ในช่วงการเก็บประจุ คายประจุ และทั้งเก็บและคายประจุ โดยการกำหนดอินพุต ได้แก่ แรงดันที่แหล่งกำเนิด (V_s) ค่าความต้านทาน (R) ค่าความจุ (C) ค่าแรงดันสุดท้ายที่ตกคร่อมตัวเก็บประจุ (V_o) ส่วนค่าคงที่ของเวลา (τ) และค่าเวลา (t) โปรแกรมจะคำนวณให้อัตโนมัติ ส่วนของผลตอบสนอง (Respond) ประกอบด้วย แรงดัน กระแส กำลังและพลังงาน ที่เกิดขึ้นในวงจรแสดงลักษณะของโปรแกรกดังภาพที่ 10.5



ภาพที่ 10.5 การออกแบบหน้าจอโปรแกรมย่อยการวิเคราะห์วงจรอันดับหนึ่ง RC

จากภาพที่ 10.5 สามารถอธิบายขั้นตอนการสร้างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RC ไฟฟ้ากระแสตรง ดังนี้

1. การสร้างวัตถุรับค่าอินพุต (Input) วัตถุที่นำมาใช้สำหรับการสร้างตัวรับค่าอินพุต (Input) ประกอบด้วยวัตถุชนิดต่าง ๆ ดังนี้

- Panel สำหรับรวมกลุ่มของอินพุตทั้งหมด โดยกำหนดชื่อเป็น Input
- Edit Text สำหรับใช้ในการป้อนค่าแรงดัน (Vs) ค่าความต้านทาน (R) ค่าความจุ (C) ค่าแรงดันสุดท้ายที่ตกคร่อมตัวเก็บประจุ (Vo) และค่าคงที่ของเวลา (tau) และค่าเวลา (t) โดยทำการกำหนดค่าที่ String ของแต่ละวัตถุให้เป็นค่าเริ่มต้นของโปรแกรม

- Static Text สำหรับใช้ในการกำหนดชื่อและหน่วยของตัวรับค่าอินพุต

2. การสร้างวัตถุเลือกสถานะของวงจร วัตถุที่นำมาใช้สำหรับการสร้างตัวเลือกสถานะของวงจร ได้แก่ Pop-up Menu โดยทำการเปลี่ยนรายการที่ String เป็นดังนี้

- Charge Capacitor
- Discharge Capacitor
- Charge and Discharge

3. การสร้างพื้นที่แสดงภาพของวงจรในสถานะต่าง ๆ วัตถุที่นำมาใช้สำหรับการสร้างพื้นที่แสดงภาพของวงจรในสถานะต่าง ๆ ได้แก่ Axes ชื่อ axes1

4. การสร้างปุ่มกดเพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ วัตถุที่นำมาใช้สำหรับการสร้างปุ่มกดเพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ ประกอบด้วยวัตถุชนิดต่าง ๆ ดังนี้

- Panel สำหรับรวมกลุ่มของผลตอบสนองทั้งหมด โดยกำหนดชื่อเป็น Respond
- Push Button สำหรับใช้เป็นปุ่มกดเมื่อต้องการให้แสดงผลตอบสนองของวงจร

โดยทำการกำหนดค่าที่ String ของแต่ละวัตถุให้เป็นดังนี้

1. Push Button1 กำหนดค่าที่ String เป็น VR(t)
2. Push Button2 กำหนดค่าที่ String เป็น VC(t)
3. Push Button3 กำหนดค่าที่ String เป็น i(t)
4. Push Button4 กำหนดค่าที่ String เป็น P(t)
5. Push Button5 กำหนดค่าที่ String เป็น W(t)

5. การสร้างพื้นที่เพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ วัตถุที่นำมาใช้สำหรับการสร้างพื้นที่เพื่อแสดงผลตอบสนองของวงจรในสถานะต่าง ๆ ได้แก่ Axes ชื่อ axes3

6. การสร้างปุ่มอำนวยความสะดวกอื่น ๆ การสร้างปุ่มอำนวยความสะดวกอื่น ๆ ประกอบด้วยปุ่มต่าง ๆ ดังต่อไปนี้

- ปุ่ม Main เป็นปุ่มที่ใช้เรียกกลับไปยังหน้าต่างเมนูหลักของโปรแกรม วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Main

- ปุ่ม Info เป็นปุ่มที่ใช้เรียกกลับไปยังหน้าต่างแสดงรายละเอียดของโปรแกรม วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Info

- ปุ่ม Graph Colors เป็นปุ่มที่ใช้เรียกกลับไปยังหน้าต่างการกำหนดสีของกราฟ วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Graph Colors

- ปุ่ม Clear เป็นปุ่มที่ใช้สำหรับลบกราฟ วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Clear

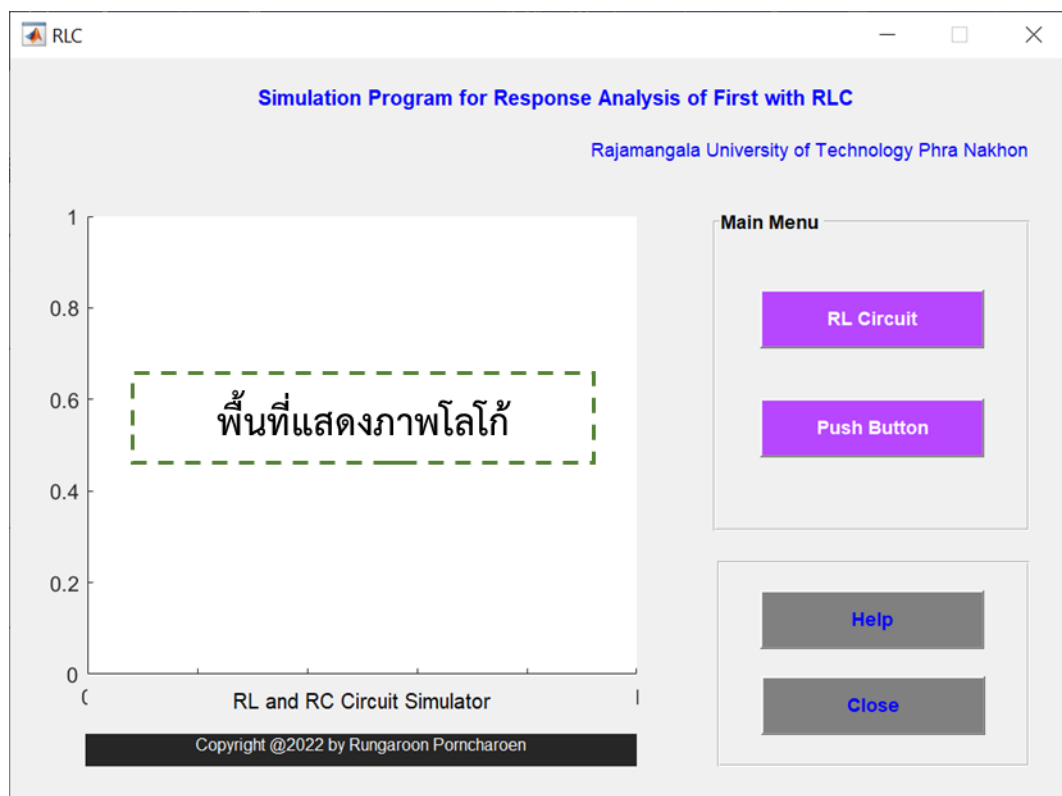
- ปุ่ม Exit เป็นปุ่มที่ใช้สำหรับปิดโปรแกรม วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Push Button โดยให้กำหนดค่าที่ String เป็น Exit

- ปุ่ม Hold On เป็นปุ่มที่ใช้สำหรับการวาดกราฟมากกว่าสองกราฟ วัตถุที่นำมาใช้สำหรับสร้างได้แก่ Radio Button โดยให้กำหนดค่าที่ String เป็น Hold On เมื่อทำการสร้างหน้าต่างโปรแกรมตามที่ยกแบบไว้เรียบร้อยแล้ว ให้ทำการบันทึกโปรแกรมเป็น ชื่อ Rc เมื่อทำการบันทึกเสร็จโปรแกรม MATLAB จะทำการกำหนด M-file ให้อัตโนมัติ

10.5 การเขียนโปรแกรมควบคุมการทำงานของวัตถุที่หน้าต่างเมนูหลัก

10.5.1 การแสดงภาพโลโก้

เมื่อรันโปรแกรมครั้งแรกจะปรากฏดังภาพที่ 10.6 ซึ่งจะเห็นว่าในพื้นที่แสดงภาพโลโก้ไม่มีภาพแสดง ดังนั้นจึงต้องสร้างภาพที่นำมาแสดงเป็นโลโก้ก่อน ในที่นี้จะใช้ภาพมีชื่อว่า main.jpg



ภาพที่ 10.6 หน้าต่างเมนูหลัก

สำหรับการเขียนโปรแกรมเพื่อควบคุมการทำงานการแสดงผลภาพโลโก้หน้าต่างของโปรแกรมหลัก มีขั้นตอนดังภาพที่ 10.7

1. คลิกที่ M-file Editor จากนั้นจะปรากฏหน้าต่างของ function การทำงาน ดังนี้

`function` RLC_OpeningFcn(hObject, eventdata, handles, varargin)

2. พิมพ์คำสั่งการทำงาน จากนั้นกดบันทึก

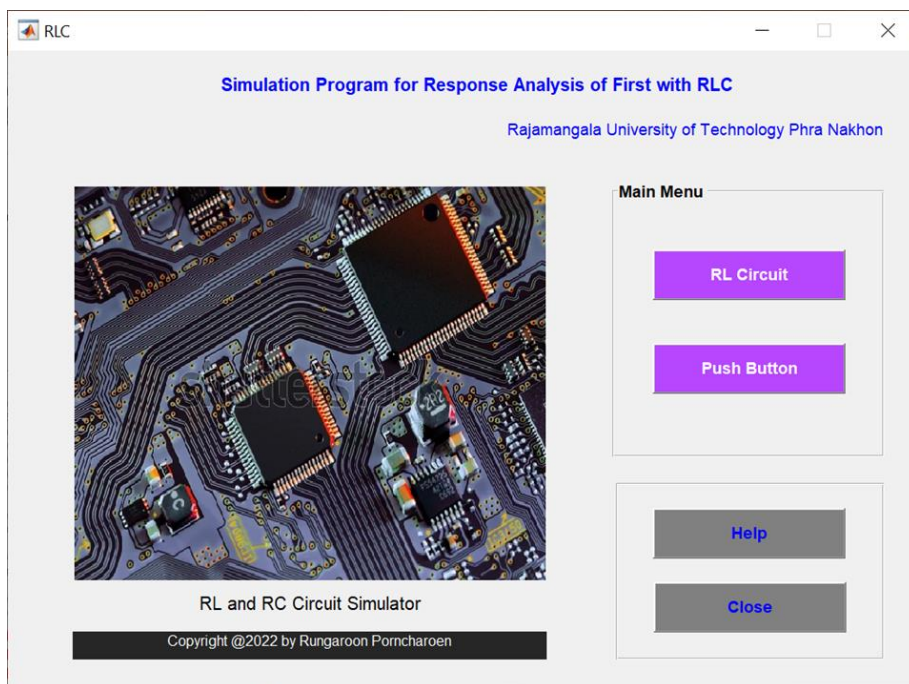
```

46
47 % --- Executes just before RLC is made visible.
48 function RLC_OpeningFcn(hObject, eventdata, handles, varargin)
49 % This function has no output args, see OutputFcn.
50 % hObject    handle to figure
51 % eventdata  reserved - to be defined in a future version of MATLAB
52 % handles    structure with handles and user data (see GUIDATA)
53 % varargin   command line arguments to RLC (see VARARGIN)
54
55 % Choose default command line output for RLC
56 handles.output = hObject;
57
58 % Update handles structure
59 guidata(hObject, handles);
60
61 % UIWAIT makes RLC wait for user response (see UIRESUME)
62 % uiwait(handles.figure1);
63
64 axes(handles.axes1)
65 handles.layer = imread('main.jpg');
66 image(handles.layer)
67 set(handles.axes1, 'Visible', 'off');
68

```

ภาพที่ 10.7 การเขียนโปรแกรมการใส่ภาพโลโก้

3. เมื่อทำการรันโปรแกรมหน้าต่างเมนูหลักจะแสดงภาพโลโก้ ดังภาพที่ 10.8

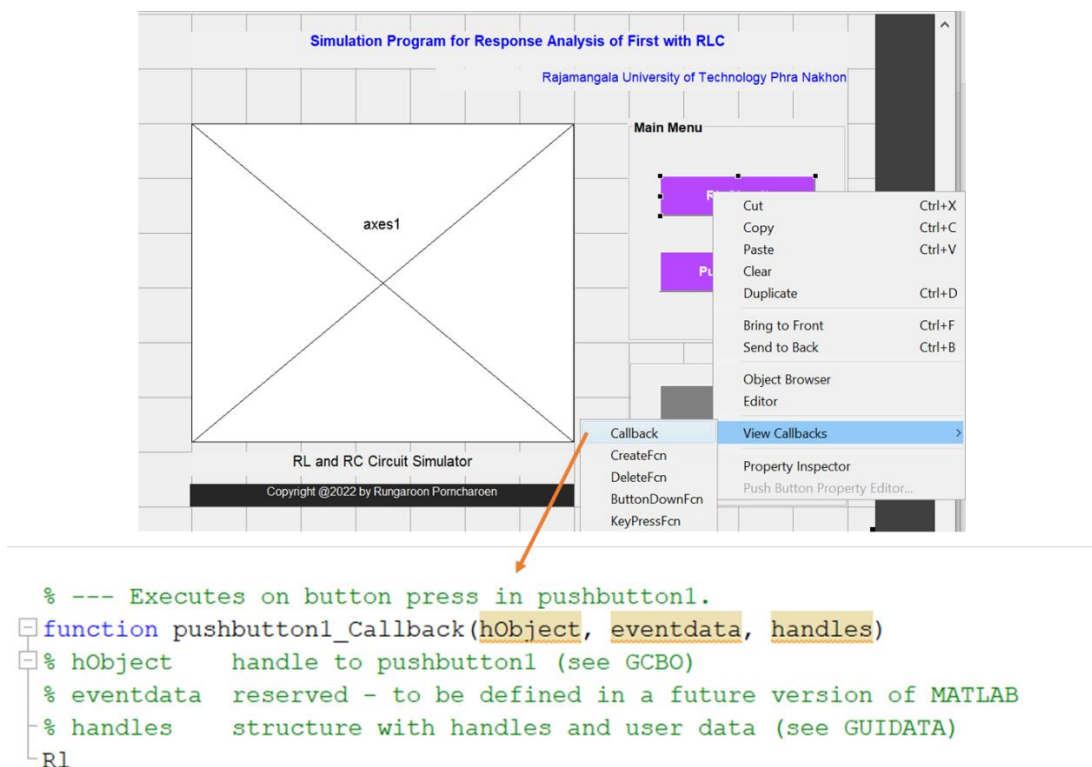


ภาพที่ 10.8 หน้าต่างเมนูแสดงภาพโลโก้

10.5.2 การเขียนโปรแกรมเพื่อควบคุมวัตถุเชื่อมโยงโปรแกรมย่อย

การเขียนโปรแกรมเพื่อควบคุมวัตถุเพื่อเชื่อมโยงไปยังโปรแกรมย่อย ได้แก่ โปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL และโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RC มีรายละเอียดดังต่อไปนี้

1. คลิกขวาที่ปุ่ม RL CIRCUIT เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callback จะปรากฏ M-file Editor ที่ฟังก์ชัน pushbutton1 ให้พิมพ์ชื่อ RL (RL คือชื่อของโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL ที่ทำการบันทึกไว้ก่อนหน้านี้) แสดงดังภาพที่ 10.9 เสร็จแล้วทำการบันทึก



ภาพที่ 10.9 การเชื่อมโยงไปยังหน้าต่างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL

2. คลิกขวาที่ปุ่ม RC CIRCUIT เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callback จะปรากฏ M-file Editor ที่ฟังก์ชัน pushbutton1 ให้พิมพ์ชื่อ Rc (Rc คือชื่อของโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RC ที่ทำการบันทึกไว้ก่อนหน้านี้) เสร็จแล้วทำการบันทึก

3. เมื่อเสร็จสิ้นการเขียนโปรแกรมควบคุมการทำงานของวัตถุเพื่อเชื่อมโยงไปยังโปรแกรมย่อยต่าง ๆ สามารถทดสอบการเชื่อมโยงว่าเป็นไปตามเงื่อนไขที่ผู้ออกแบบต้องการหรือไม่

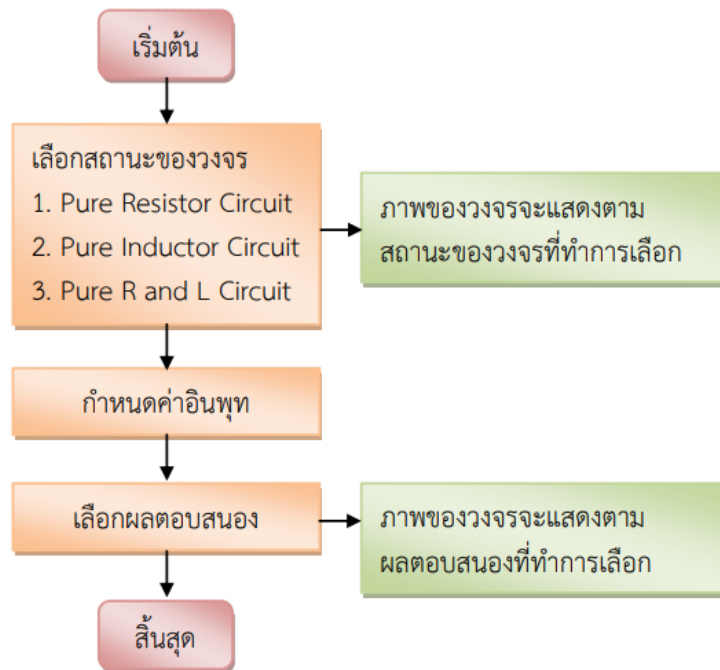
ถ้าใช่ จบกระบวนการ

ถ้าไม่ใช่ ให้ปรับปรุงแก้ไข

ข้อสังเกต : เมื่อทำการรันโปรแกรมหน้าต่างเมนูหลักจะปรากฏชื่อโปรแกรมที่มุมด้านบนซ้ายว่า RLC (จะปรากฏชื่อตามชื่อที่บันทึกไว้)

10.6 การควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL

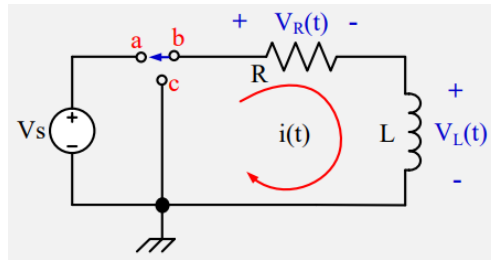
การควบคุมการทำงานของวัตถุที่หน้าต่างโปรแกรมวิเคราะห์วงจร RL ในที่นี้ขอกล่าวถึงรายละเอียดของกระบวนการทำงานของโปรแกรมวิเคราะห์วงจร RL ดังภาพที่ 10.10



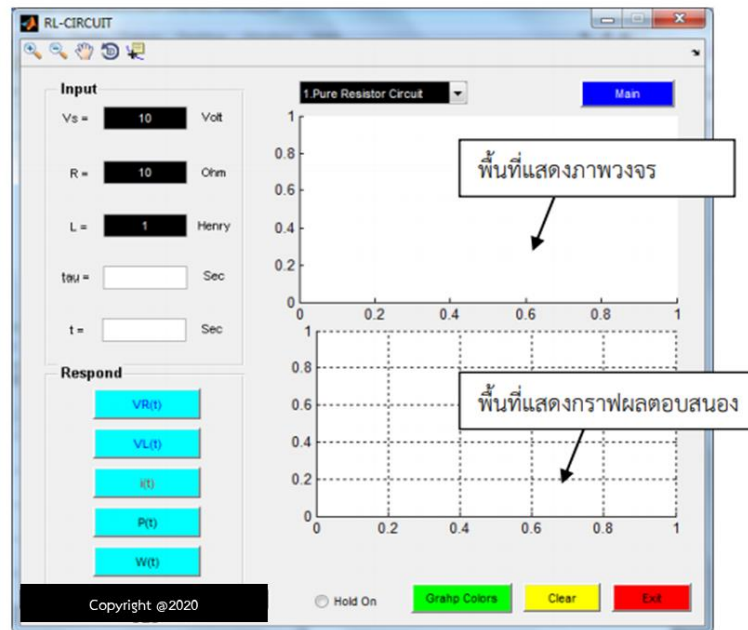
ภาพที่ 10.10 กระบวนการทำงานของโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL

จากภาพที่ 10.10 การทำงานของโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL ไฟฟ้ากระแสตรงสามารถอธิบายได้ดังนี้ เมื่อเปิด โปรแกรมขึ้นมาหน้าต่างของโปรแกรมจะทำการกำหนดค่าเริ่มต้นให้กับโปรแกรมโดยอัตโนมัติ ซึ่งผู้ใช้งานสามารถปรับเปลี่ยนค่าได้ตามต้องการ เมื่อทำการเลือกสถานะวงจรภาพของวงจรจะทำการเปลี่ยนให้ตรงกับสถานะของวงจรที่เลือก หลังจากนั้นให้กำหนดค่าอินพุตของวงจร และทำการเลือกผลตอบสนองที่ต้องการให้แสดงผล สามารถอธิบายรายละเอียดการเขียนโปรแกรมเป็นดังนี้

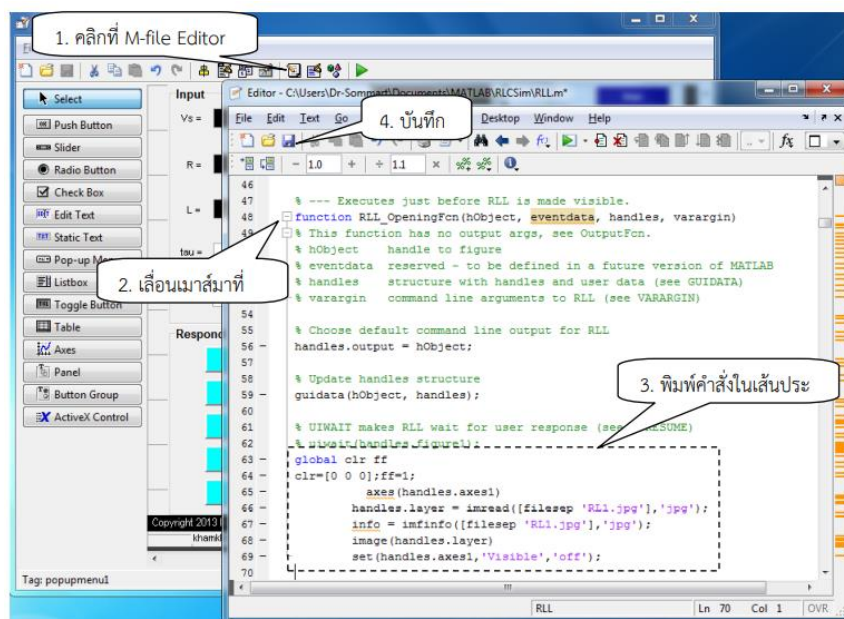
1. เมื่อรันโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL ครั้งแรก จะปรากฏดังภาพที่ 10.11 ซึ่งแสดงให้เห็นว่าโปรแกรมยังไม่เป็นไปตามเงื่อนไขของกระบวนการทำงานในภาพที่ 10.10 คือโปรแกรมไม่แสดงภาพสถานะของวงจร เมื่อเปิดโปรแกรมในครั้งแรกจะเป็นสถานะ Pure Resistor Circuit ดังนั้นภาพที่แสดงจะต้องเป็นภาพตามสถานะ โดยสามารถสร้างภาพประกอบได้จากโปรแกรมประยุกต์ต่าง ๆ ในที่นี้ใช้ภาพที่มีชื่อว่า RL1.jpg ขั้นตอนการใส่ภาพสถานะของวงจรแสดงดังภาพที่ 10.12



สถานะ Pure Resistor Circuit

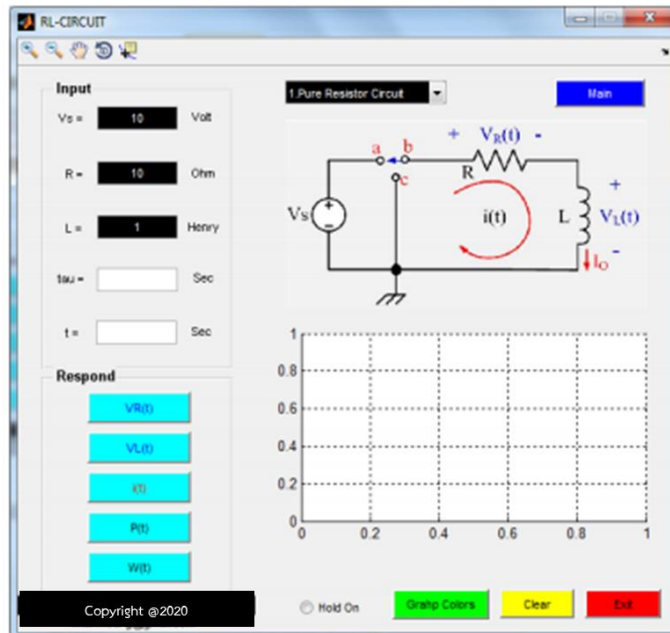


ภาพที่ 10.11 หน้าต่างโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL เมื่อเปิดโปรแกรมเป็นครั้งแรก



ภาพที่ 10.12 ขั้นตอนการใส่ภาพสถานะของวงจร

เมื่อทำการรันโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL จะแสดงภาพสถานะของวงจร ดังภาพที่ 10.13



ภาพที่ 10.13 สถานะของวงจรในโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL เมื่อเปิดโปรแกรมครั้งแรก

2. การเขียนโปรแกรมเพื่อเลือกสถานะของวงจรอันดับหนึ่ง RL

การเขียนโปรแกรมเพื่อเลือกสถานะของวงจรพร้อมทั้งแสดงภาพเมื่อเลือกสถานะนั้น ๆ สิ่งที่ต้องเตรียมคือภาพสถานะของวงจรแบบต่าง ๆ มีการตั้งชื่อภาพดังนี้

- ภาพชื่อ RL1.jpg เพื่อแสดงสถานะของวงจร Pure Resistor Circuit
- ภาพชื่อ RL2.jpg เพื่อแสดงสถานะของวงจร Pure Inductor Circuit
- ภาพชื่อ RL3.jpg เพื่อแสดงสถานะของวงจร Pure R and L Circuit

3. จากนั้นทำการเขียนโปรแกรมลงในปุ่มต่าง ๆ ดังนี้

- ปุ่ม Pure Resistor Circuit เลื่อนเมาส์ไปที่ View Callbacks แล้วเลือก Callback จะปรากฏ M-file Editor ที่

```
function popupmenu1_Callback(hObject, eventdata, handles)
```

แล้วให้เขียนโปรแกรม และทำการบันทึก

```
%=====start=====
global ff
mtt=get(hObject,'value');
switch mtt
```

case 1

```
ff=1;
axes(handles.axes1)
handles.layer = imread([filesep 'RL1.jpg'],'jpg');
info = imfinfo([filesep 'RL1.jpg'],'jpg');
image(handles.layer)
set(handles.axes1,'Visible','off');
axes(handles.axes3);title("");xlabel("");ylabel(""); cla
```

case 2

```
ff=2;
axes(handles.axes1)
handles.layer = imread([filesep 'RL2.jpg'],'jpg');
info = imfinfo([filesep 'RL1.jpg'],'jpg');
image(handles.layer)
set(handles.axes1,'Visible','off');
axes(handles.axes3);title("");xlabel("");ylabel(""); cla
```

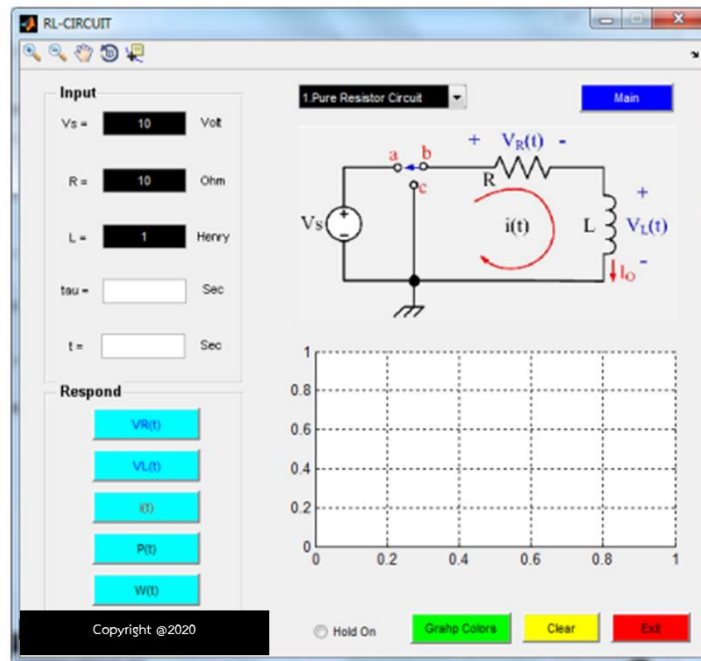
case 3

```
ff=3;
axes(handles.axes1)
handles.layer = imread([filesep 'RL3.jpg'],'jpg');
info = imfinfo([filesep 'RL1.jpg'],'jpg');
image(handles.layer)
set(handles.axes1,'Visible','off');
axes(handles.axes3);title("");xlabel("");ylabel(""); cla
```

end

%=====stop=====

4. ทำการรันโปรแกรมวิเคราะห์ห้วงจรอันดับหนึ่ง RL และทำการเลือกสถานะของวงจรเป็น Pure Resistor Circuit จะแสดงดังภาพที่ 10.14



ภาพที่ 10.14 การแสดงสถานะของวงจรอันดับหนึ่ง RL แบบ Pure Resistor Circuit

สำหรับการเขียนโปรแกรมคำสั่งปุ่มต่าง ๆ สามารถดำเนินการเขียนได้ในลักษณะเดียวกันกับที่กล่าวมาข้างต้น

10.7 สรุป

การประยุกต์ใช้ GUI ของ MATLAB สำหรับงานทางด้านไฟฟ้าที่นำเสนอในบทนี้ เป็นการสร้าง GUI สำหรับการวิเคราะห์วงจรอันดับหนึ่ง RL และ RC ของวงจรไฟฟ้ากระแสตรง โดยจะกล่าวถึงปัญหาว่าทำไมต้องทำการสร้าง GUI ทฤษฎีที่เกี่ยวข้อง ขั้นตอนในการสร้าง GUI ในหน้าต่างโปรแกรมหลัก โปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL และโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RC และการเขียนโปรแกรมควบคุมการทำงานของวัตถุ พร้อมทั้งแสดงการเขียนโปรแกรมเพื่อแสดงภาพที่หน้าต่างเมนูหลัก ซึ่งเป็นรายละเอียดขั้นต้นของการสร้าง GUI ทั้งหมด

ส่วนการพิจารณาหาค่ากระแสและแรงดันของวงจรตัวต้านทานกับตัวเหนี่ยวนำ และตัวต้านทานกับตัวเก็บประจุ ในสถานะที่ตัวเหนี่ยวนำหรือตัวเก็บประจุสะสมหรือคายพลังงานที่เก็บสะสมไว้ หรือซึ่งสามารถพิจารณาได้จากสมการอนุพันธ์อันดับหนึ่ง สำหรับวงจรที่ประกอบด้วยตัวต้านทานกับตัวเหนี่ยวนำ (RL) หรือวงจรที่ประกอบด้วยตัวต้านทานกับตัวเก็บประจุ (RC) เป็นตัวอย่างการวิเคราะห์โดยใช้โปรแกรมคอมพิวเตอร์ เป็นการหาคำตอบวิธีการหนึ่งที่นิยมนำมาใช้ เพราะสะดวกและมีความรวดเร็วในการคำนวณ ทำให้นักศึกษาสามารถเข้าใจหลักการทำงานของวงจรไฟฟ้ากระแสตรงได้เป็นอย่างดี

แบบฝึกหัด

บทที่ 10 การประยุกต์ใช้ GUI ของ MATLAB สำหรับการวิเคราะห์วงจรอันดับหนึ่ง RL และ RC

1. จงอธิบายขั้นตอนการวิเคราะห์สำหรับการพัฒนา GUI ของโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL และ RC
2. จงออกแบบโปรแกรมจำลองกระบวนการทำงานของโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL และ RC
3. จงเขียนโปรแกรมเพื่อควบคุมปุ่ม $VR(t)$, $VC(t)$, $i(t)$, $P(t)$ และ $W(t)$ ของการวิเคราะห์วงจรอันดับหนึ่ง RL
4. จงเขียนโปรแกรมเพื่อควบคุมปุ่ม $VR(t)$, $VC(t)$, $i(t)$, $P(t)$ และ $W(t)$ ของการวิเคราะห์วงจรอันดับหนึ่ง RC
5. จงเขียนโปรแกรมเพื่อควบคุมวัตถุอื่น ๆ ในโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL
6. จงเขียนโปรแกรมเพื่อควบคุมวัตถุอื่น ๆ ในโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RC
7. จงอธิบายขั้นตอนวิธีดำเนินการพัฒนา GUI ของโปรแกรมวิเคราะห์วงจรอันดับสอง RLC
8. จงออกแบบและเขียนโปรแกรมวิเคราะห์วงจรอันดับสอง RLC โดยมีเงื่อนไขดังนี้
 - 8.1 มีรูปแบบผลตอบสนองสัญญาณเข้า 3 รูปแบบ ได้แก่ หนึ่งเกิน หนึ่งขาด และหนึ่งวิกฤต
 - 8.2 แสดงสมการผลตอบสนองสัญญาณแต่ละรูปแบบ
9. จงเขียนคู่มือการใช้งานโปรแกรมวิเคราะห์วงจรอันดับหนึ่ง RL และ RC จากโปรแกรมที่พัฒนาขึ้น
10. จงเขียนคู่มือการใช้งานโปรแกรมวิเคราะห์วงจรอันดับสอง RLC จากโปรแกรมที่พัฒนาขึ้น

บทที่ 11

การประยุกต์ใช้ Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์

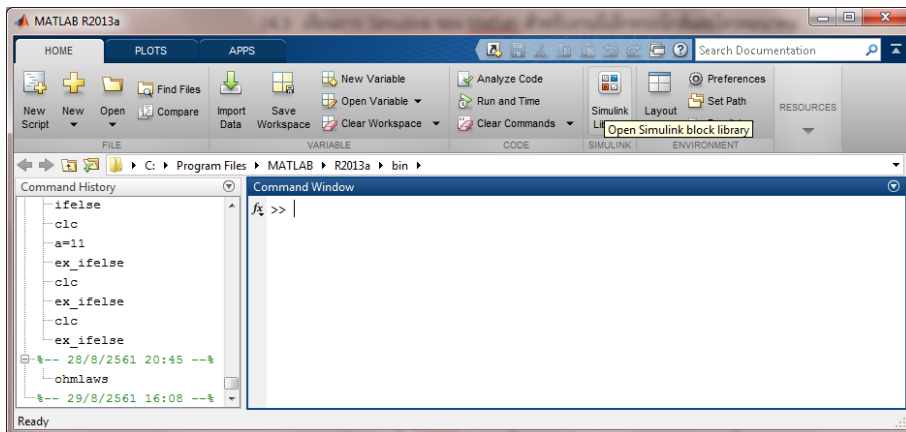
จากที่ทราบโปรแกรม MATLAB เป็นโปรแกรมสำหรับช่วยงานคำนวณทางคณิตศาสตร์ที่มีชื่อเสียงในปัจจุบัน และเป็นที่ยอมรับในทางด้านการวิจัย พัฒนาและวิเคราะห์ โดย MATLAB เป็นภาษาคอมพิวเตอร์ชั้นสูง (High-level Language) ทำให้การคำนวณผลลัพธ์ พัฒนาอัลกอริทึม สร้างแบบจำลองและแอปพลิเคชันได้ง่ายและรวดเร็ว โปรแกรม MATLAB สามารถทำงานได้ทั้งในลักษณะการเขียนคำสั่งเข้าไปทีละคำสั่ง เพื่อให้ MATLAB ประมวลผลหรือสามารถรวบรวมชุดคำสั่งให้เป็นโปรแกรมได้ โดยเก็บข้อมูลในลักษณะของเมทริกซ์ในการสร้างตัวแปรเพื่อเก็บข้อมูล ไม่จำเป็นต้องจองมิติของเมทริกซ์เหมือนกับการเขียนโปรแกรมทั่วไป ทำให้สามารถแก้ปัญหาของตัวแปรที่อยู่ในรูปแบบเมทริกซ์หรือเวกเตอร์ได้โดยง่าย

อีกทั้งความสามารถหลักของ MATLAB คือ เป็นโปรแกรมที่มีความสามารถในการคำนวณและแสดงผลได้ทั้งตัวเลขและกราฟที่มีประสิทธิภาพและมีฟังก์ชันที่เหมาะสมกับงานทางด้านวิทยาศาสตร์และวิศวกรรมศาสตร์ นอกจากนั้นผู้ใช้อย่างยังสามารถเขียนฟังก์ชันขึ้นมาใหม่โดยสามารถใช้ประโยชน์จากฟังก์ชันที่มีอยู่แล้วเพื่อให้เหมาะสมกับงาน การใช้งานโปรแกรม MATLAB มีฟังก์ชันทางคณิตศาสตร์ให้เลือกใช้มากมายนอกจากนั้นยังสามารถสร้างฟังก์ชันขึ้นมาใช้งานได้เอง ฟังก์ชันที่สร้างขึ้นจะถูกบันทึกไว้ในไฟล์ที่มีนามสกุล .m หรือ m-file โดย Algorithm สามารถพัฒนาได้ง่ายและสามารถแก้ไขปัญหาด้านคณิตศาสตร์ที่มีความซับซ้อนได้ง่ายและรวดเร็ว นอกจากนั้นยังมีแบบจำลอง (Simulink) ที่สามารถสร้างบล็อกไดอะแกรมเพื่อใช้ทดสอบ วิเคราะห์ ตรวจสอบและประเมินผลระบบต่าง ๆ ก่อนนำไปใช้งานจริง

การเขียนโปรแกรมใน MATLAB จะใกล้เคียงการเขียนสมการทางคณิตศาสตร์ จึงง่ายกว่าการเขียนโปรแกรมโดยใช้ภาษา C, Fortran หรือภาษาอื่น ๆ และยังสามารถในการสร้างกราฟหรือรูปภาพทั้งในรูปแบบ 2 มิติ และ 3 มิติ นอกจากนั้นยังสามารถทำงานแบบ Dynamic link กับโปรแกรม Word, Excel, RapidMiner, DiGSILENT หรือโปรแกรมอื่น ๆ ที่ร่วมทำงานอยู่บนระบบปฏิบัติการ (Operation System) ได้ จึงทำให้สามารถนำโปรแกรม MATLAB ไปประยุกต์ใช้งานได้หลากหลายสาขา ทั้ง Signal processing, Image and Video processing, Control system, Neural networks, Fuzzy logic, Wavelet, Communication, Statistics, Data Cleaning, Economic และอื่น ๆ อีกมากมาย สำหรับบทนี้เป็นการประยุกต์ใช้ Simulink ซึ่งเป็นเครื่องมือ Toolbox ที่อยู่ในโปรแกรม MATLAB ภายใต้การทำงานของหน้าต่างที่เป็นการเชื่อมต่อทางรูปภาพ (GUI) ของ Simulink

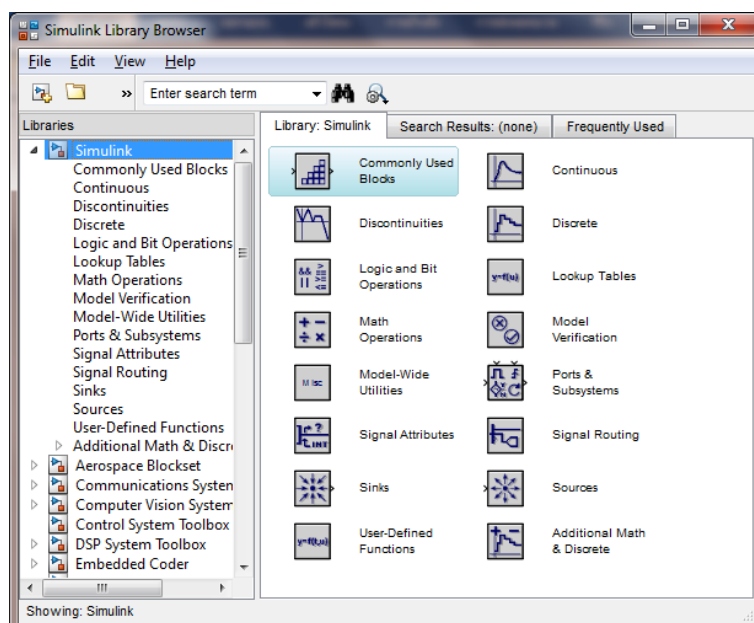
11.1 การเรียกใช้ Simulink ของ MATLAB

การใช้งาน Simulink เริ่มจากเปิดโปรแกรม MATLAB จากนั้นคลิกปุ่ม  บนแถบเครื่องมือ เพื่อเรียกใช้งาน Simulink ของโปรแกรม ดังภาพที่ 11.1



ภาพที่ 11.1 การเรียกใช้งาน Simulink ของโปรแกรม

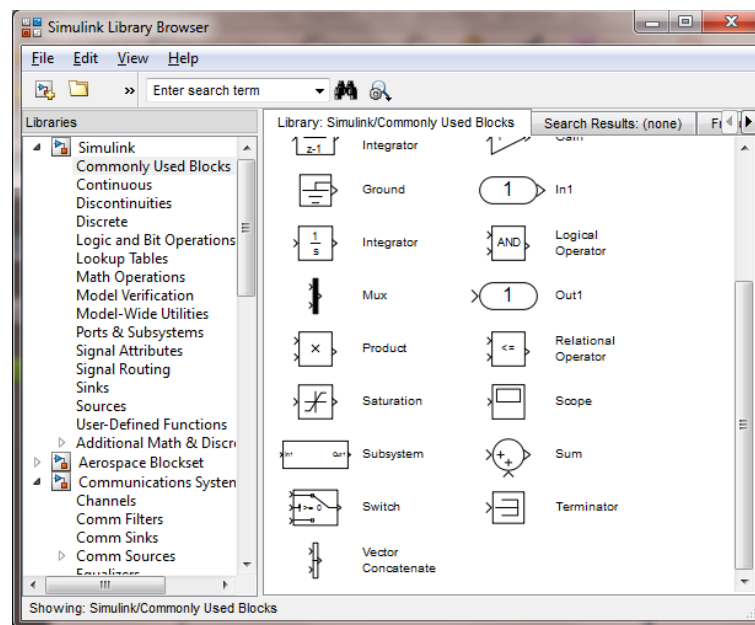
จากนั้นจะปรากฏหน้าต่าง Library ของโปรแกรม ดังภาพที่ 11.2



ภาพที่ 11.2 หน้าต่าง Library ของโปรแกรม

11.2 ส่วนประกอบต่าง ๆ บนหน้าต่าง Library Simulink ของ MATLAB

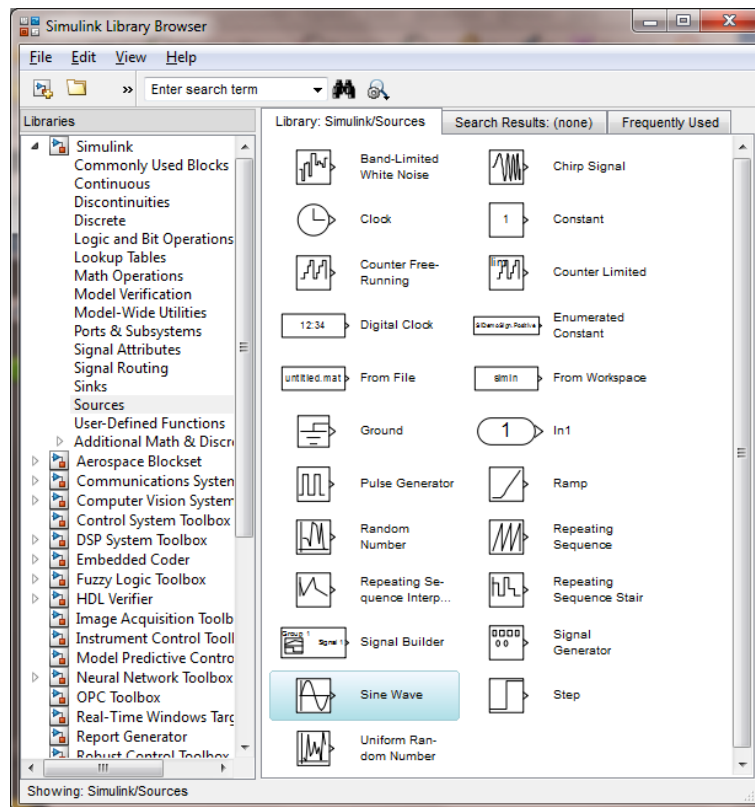
เมื่อเริ่มใช้งาน Simulink จะพบหน้าต่าง Simulink Library Browser ดังภาพที่ 11.3 ประกอบด้วย Block Diagram ที่ถูกจัดเป็นหมวดหมู่ต่าง ๆ การเลือกใช้งานในหมวดหมู่ใดในหน้าต่าง Simulink Library Browser ให้คลิกที่ไอคอนของหมวดหมู่นั้น และปรากฏหน้าต่างรายละเอียดของ Block Diagram บนหน้าต่างของ Simulink Library นั้น ๆ ในการจำลองการทำงานของระบบ อุปกรณ์เหล่านี้ไม่สามารถทำงานได้เพียงตัวเดียวโดด ๆ จะต้องเชื่อมต่อกับอุปกรณ์ตัวอื่น ๆ จึงจะทำงานได้



ภาพที่ 11.3 หน้าต่าง Simulink Library Browser

สำหรับ Simulink Library ที่เกี่ยวข้องกับงานไฟฟ้าและอิเล็กทรอนิกส์

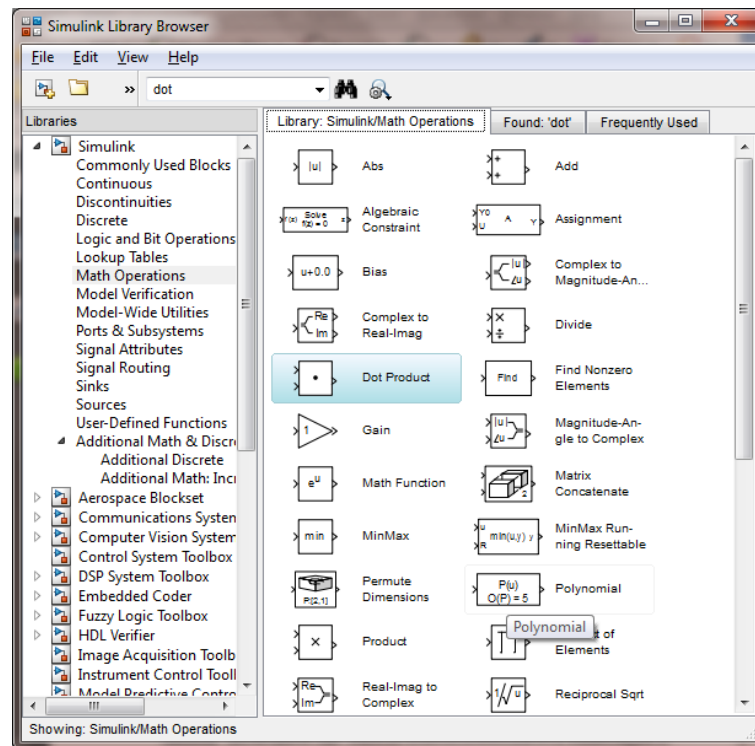
1. Source Library ประกอบด้วย Block ต่าง ๆ ที่เกี่ยวข้องกับการกำเนิดสัญญาณเพื่อใช้ในการเป็นอินพุตให้กับระบบ เช่น Sine Wave, Pulse Generator เป็นต้น ดังภาพที่ 11.4



ภาพที่ 11.4 รายละเอียดของ Block ในหมวดหมู่ Sources

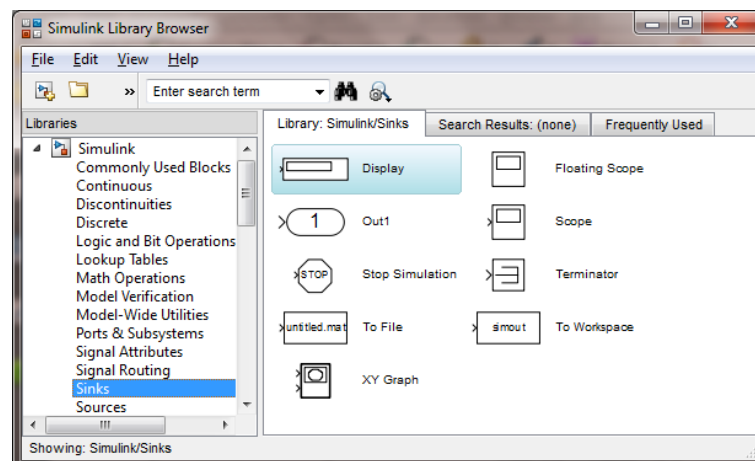
จากภาพที่ 11.4 ภายใน Block แต่ละ Block มีช่องสำหรับป้อนค่าพารามิเตอร์ ซึ่งสามารถแก้ไขค่าต่าง ๆ ให้ตรงกับความต้องการของผู้ใช้ได้

2. Math Operations ประกอบด้วย Block ที่เกี่ยวกับฟังก์ชันการดำเนินการทางคณิตศาสตร์ทั่วไป ดังภาพที่ 11.5



ภาพที่ 11.5 รายละเอียดของ Block ในหมวดหมู่ Math Operations

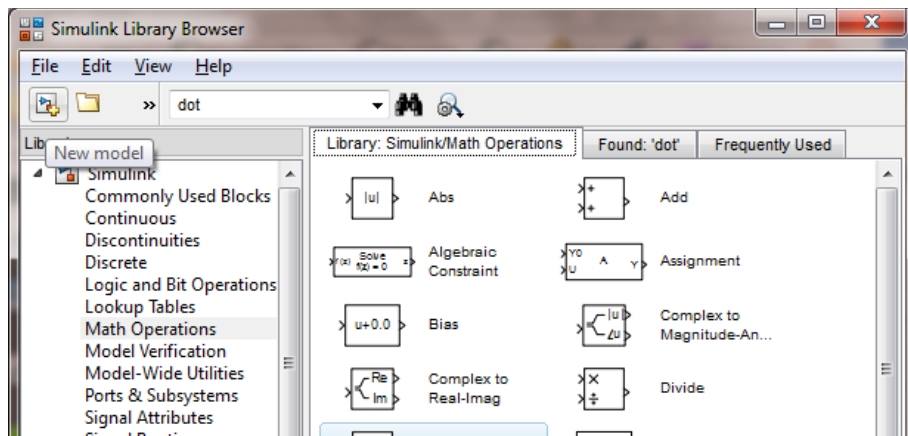
3. Sinks Library ประกอบด้วย Block ที่ใช้ในการรับค่าของสัญญาณหรือเอาต์พุตในแต่ละจุดของแบบจำลอง เพื่อนำไปแสดงผลหรือควบคุมการทำงานของ Simulink ดังภาพที่ 11.6



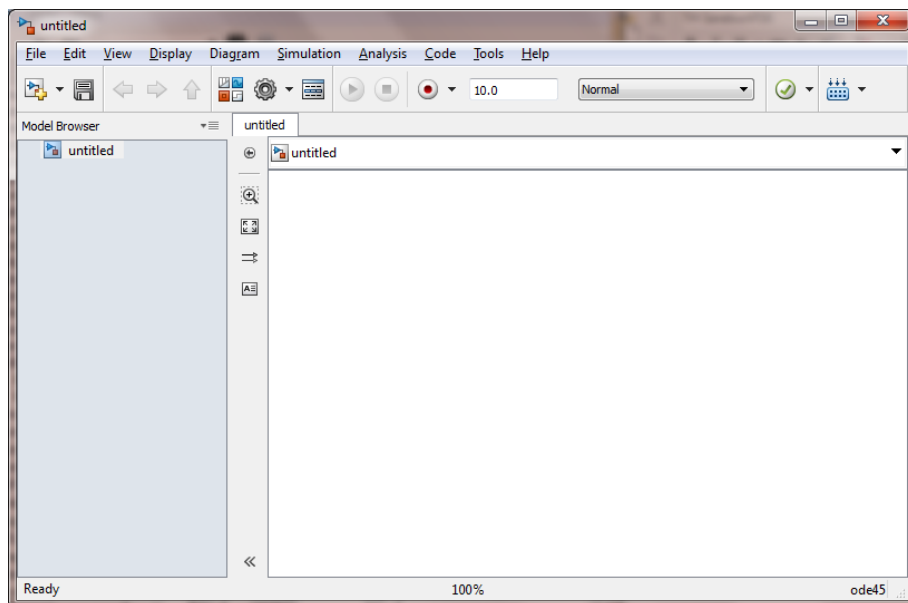
ภาพที่ 11.6 รายละเอียดของ Block ในหมวดหมู่ Sinks

11.3 การเรียกใช้ Block พื้นฐานของ Simulink ของ MATLAB

เมื่อต้องการจำลองเหตุการณ์หรือการทำงานของระบบใด ๆ สิ่งที่ต้องทำอันดับแรก คือ การสร้างแบบจำลองของระบบนั้น เมื่อต้องการสร้างแบบจำลองบน Simulink ให้ผู้ใช้คลิกที่ไอคอนรูปกระดาษเพื่อสร้างแผ่นงานใหม่ ดังภาพที่ 11.7 เมื่อคลิกที่ไอคอนดังกล่าวปรากฏแผ่นงานขึ้น ดังภาพที่ 11.8 แผ่นงานที่สร้างขึ้นจะเป็นที่เก็บแบบจำลอง ซึ่งสร้างขึ้นจากการนำ Block แต่ละตัวมาต่อกันอย่างเหมาะสมตามสมการทางคณิตศาสตร์ ซึ่งอธิบายการทำงานของระบบที่เราต้องการจำลอง เมื่อทำการบันทึกแบบจำลองที่สร้างขึ้น จะถูกเก็บไว้ในรูป File ที่มีนามสกุล .mdl



ภาพที่ 11.7 ไอคอนสำหรับเปิดแผ่นงานใหม่เพื่อสร้างแบบจำลอง



ภาพที่ 11.8 แผ่นงานใหม่เพื่อสร้างแบบจำลอง

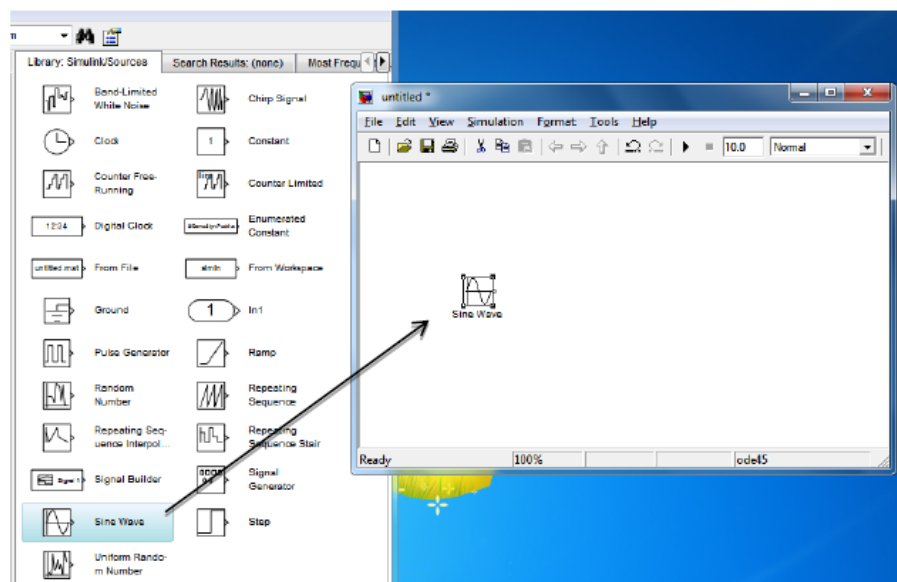
การเลือกอุปกรณ์ หรือ Block ที่อยู่ในหน้าต่าง Simulink Library Browser นั้น สามารถทำได้โดยการเลื่อนเมาส์ไปยังไอคอนของอุปกรณ์ที่ต้องการเลือก แล้วกดเมาส์ปุ่มซ้ายค้างไว้และลากรูปไอคอนนั้นมายังแผ่นงาน

การลบ Block ที่ได้เลือกไว้ สามารถใช้เมาส์คลิกที่ Block นั้นแล้วกดปุ่ม Delete บนคีย์บอร์ด

การหมุน Block สามารถทำได้โดยใช้เมาส์คลิกบน Block ที่ต้องการจะหมุนแล้วกดคีย์ Ctrl+R Block จะทำการหมุนที่ละ 90 องศา

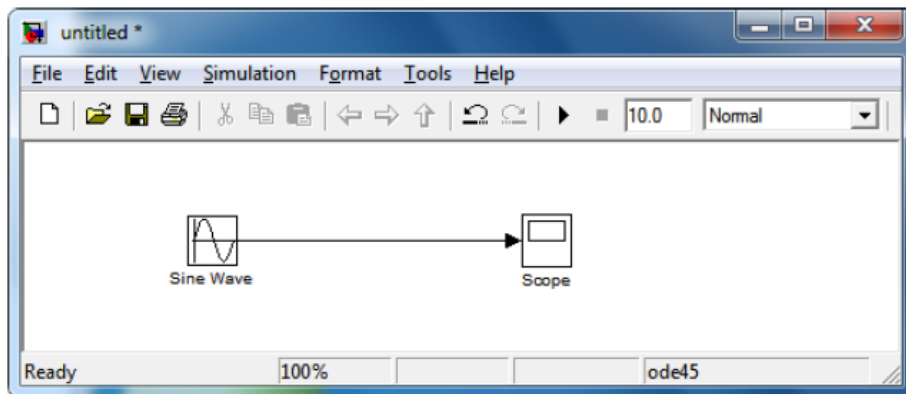
ตัวอย่างที่ 11.1 การใช้งาน Sine Wave และ Scope โดยกำหนดค่า Amplitude = 5 และค่า Frequency = $2\pi \times 50$ rad/sec

เริ่มต้นจากการนำ Block Sine Wave จาก Library กลุ่ม Source ออกมาวาง ดังภาพที่ 11.9



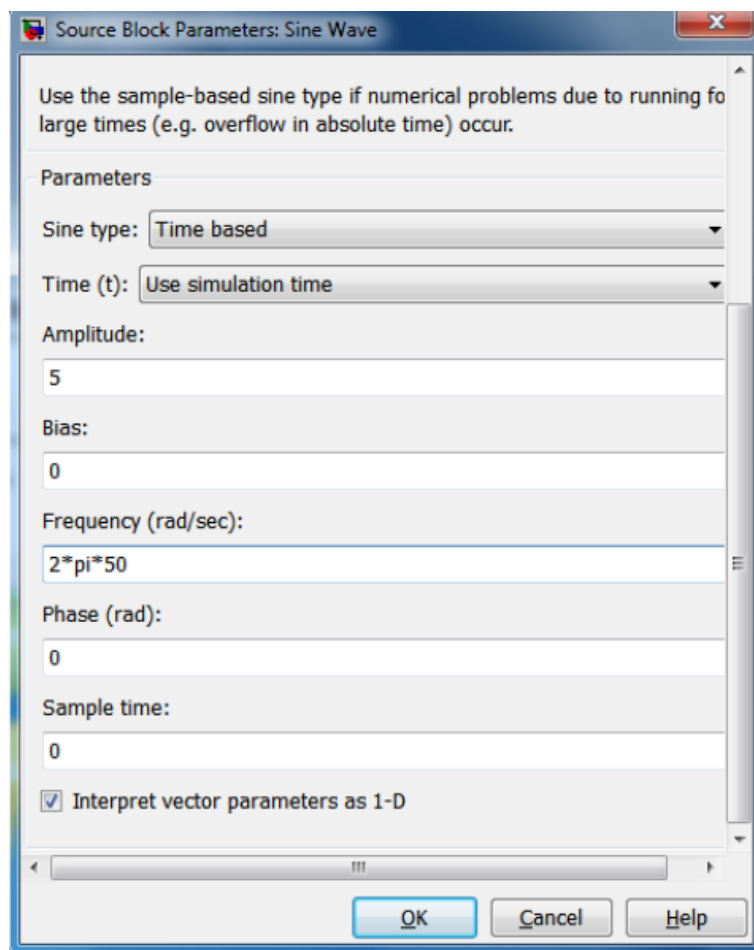
ภาพที่ 11.9 การนำ Block Sine Wave จาก Library กลุ่ม Source

จากนั้นนำ Scope จาก Library กลุ่ม Sinks ออกมาวางและทำการเชื่อมโยง Block ทั้งสองเข้าด้วยกัน ดังภาพที่ 11.10



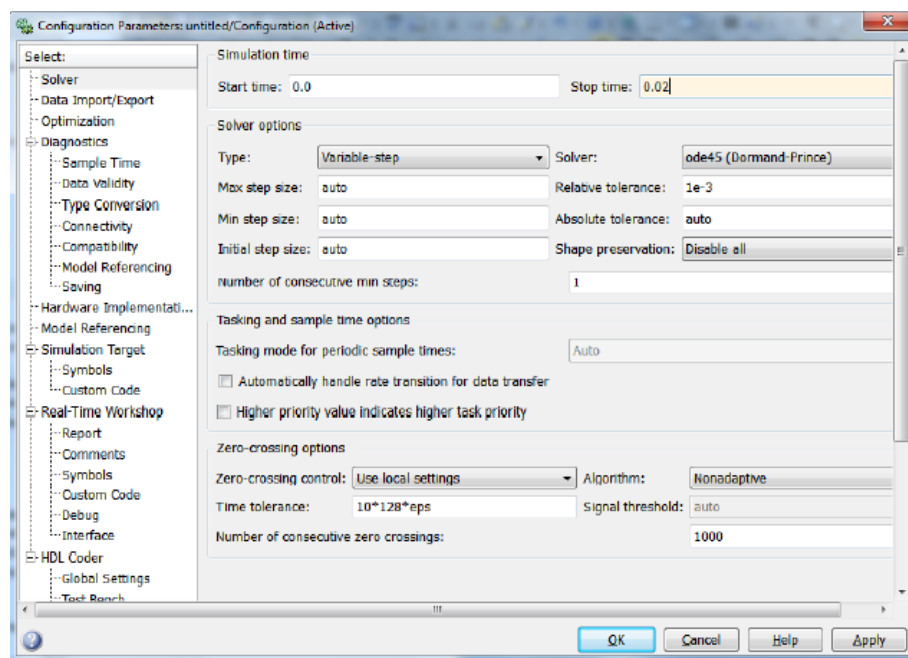
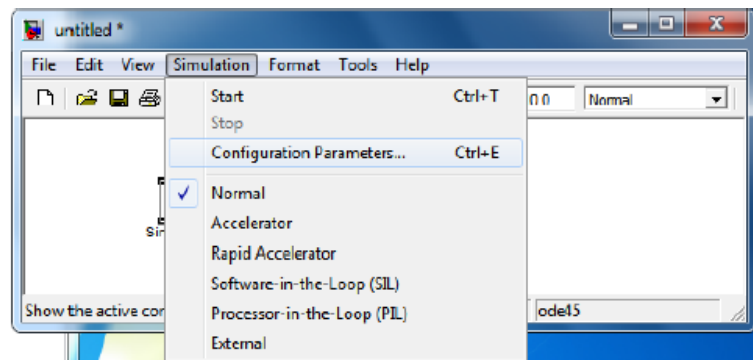
ภาพที่ 11.10 ไดอะแกรมของการใช้ Sine Wave และ Scope

ลำดับต่อไปคือ การป้อนค่าพารามิเตอร์ที่เหมาะสมให้กับ Block ที่ถูกนำมาวาง โดยใช้เมาส์วางบน Block แล้วดับเบิลคลิก จะปรากฏหน้าต่างกำหนดค่า สำหรับในตัวอย่างนี้ กำหนดให้พารามิเตอร์ของ Block Sine Wave แล้วคลิกปุ่ม OK ดังภาพที่ 11.11

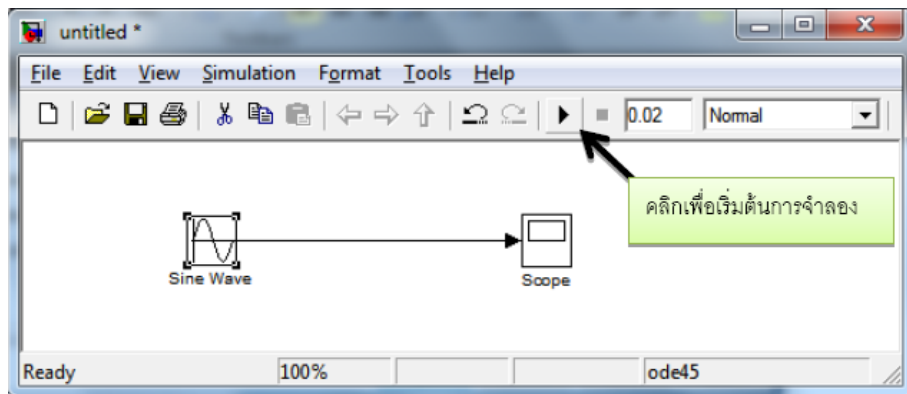


ภาพที่ 11.11 พารามิเตอร์ของ Block Sine Wave

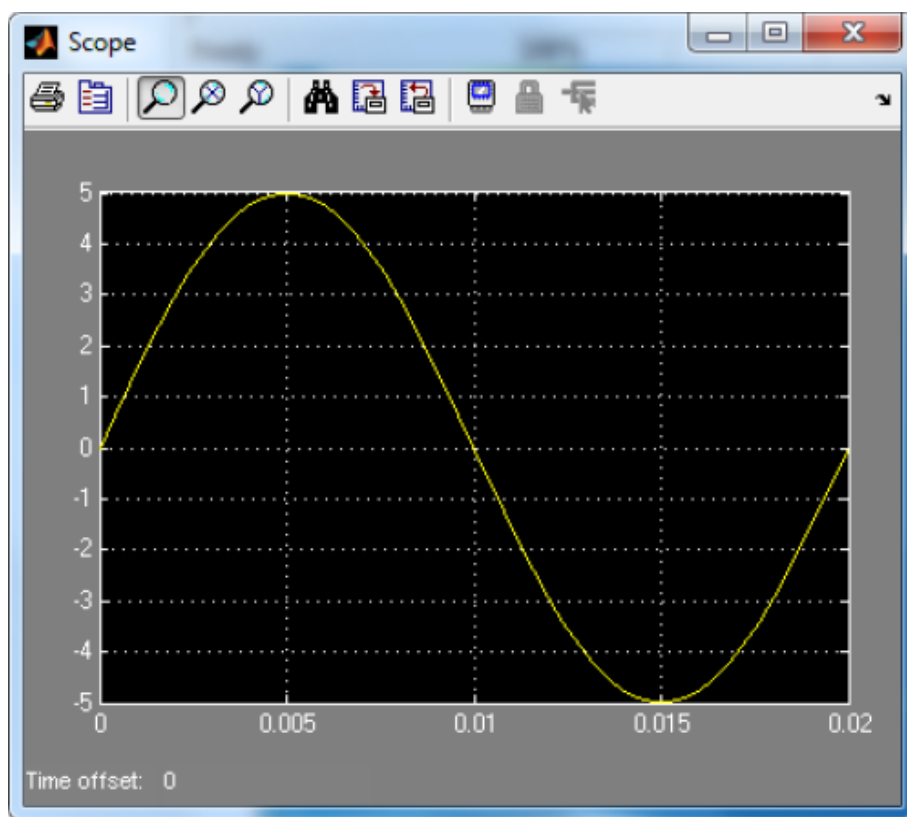
จากนั้นทำการกำหนดค่าพารามิเตอร์ของการจำลองการทำงาน โดยค่าพารามิเตอร์และวิธีการกำหนดค่า ดังภาพที่ 11.12 คือ จากหน้าต่างแผนงานให้คลิกที่ Simulation >> Configuration Parameters จะปรากฏหน้าต่างการป้อนค่าให้ผู้ใช้ป้อนค่าตามความต้องการ เมื่อเสร็จสิ้นให้กด OK เพื่อยืนยันและปิดหน้าต่าง จากนั้นทำการเริ่มต้นการจำลองการทำงานโดยคลิกปุ่ม run ดังภาพที่ 11.13 เมื่อ Simulink ประมวลผลเสร็จแล้วให้ผู้ใช้งานดับเบิลคลิกที่ Block Scope เพื่อดูผลการจำลอง ดังภาพที่ 11.14



ภาพที่ 11.12 พารามิเตอร์ของการจำลองการทำงาน



ภาพที่ 11.13 การสั่งให้ Simulink จำลองการทำงานของระบบที่สร้างขึ้น

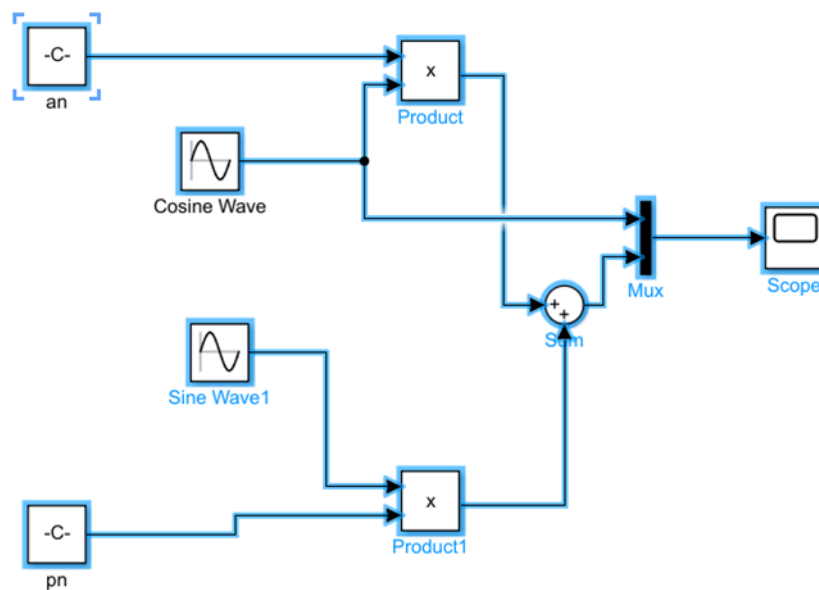
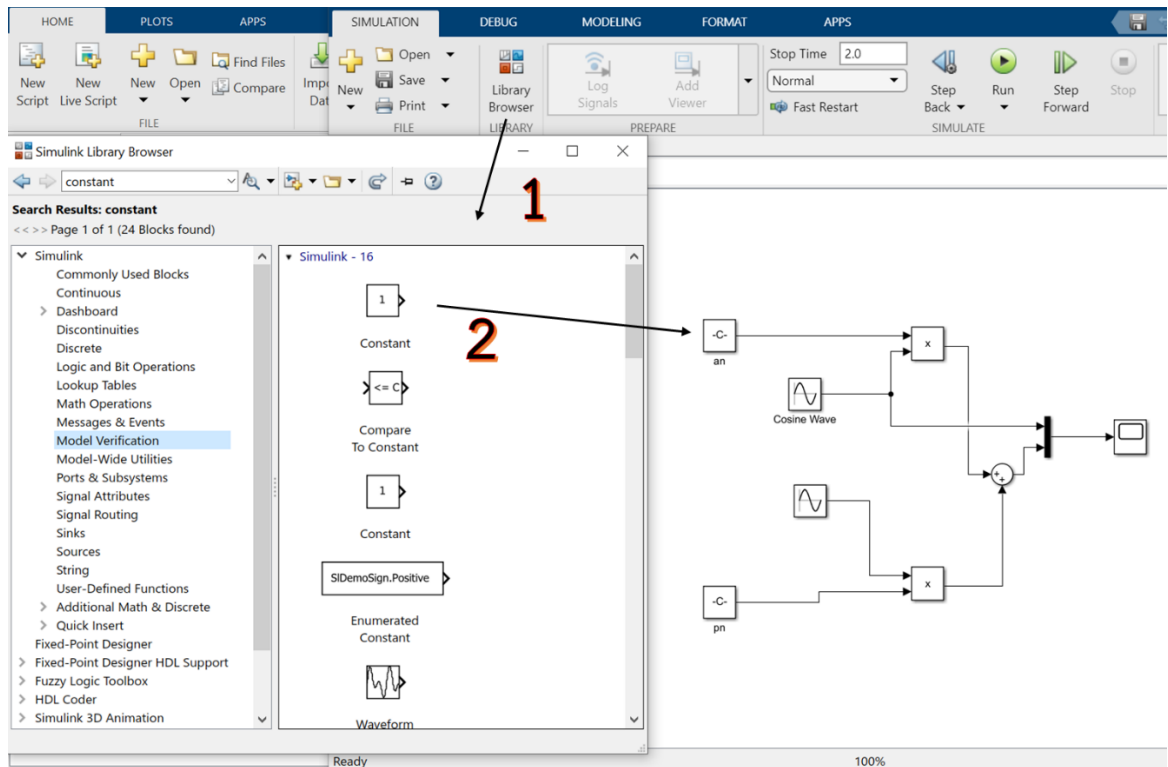


ภาพที่ 11.14 ผลการจำลองการทำงานของระบบที่สร้างขึ้น

ตัวอย่างที่ 11.2 การหาค่าความต่างเฟสของสัญญาณ

$$f(t) = a_0 + \sum_{n=1}^{\infty} ((a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)))$$

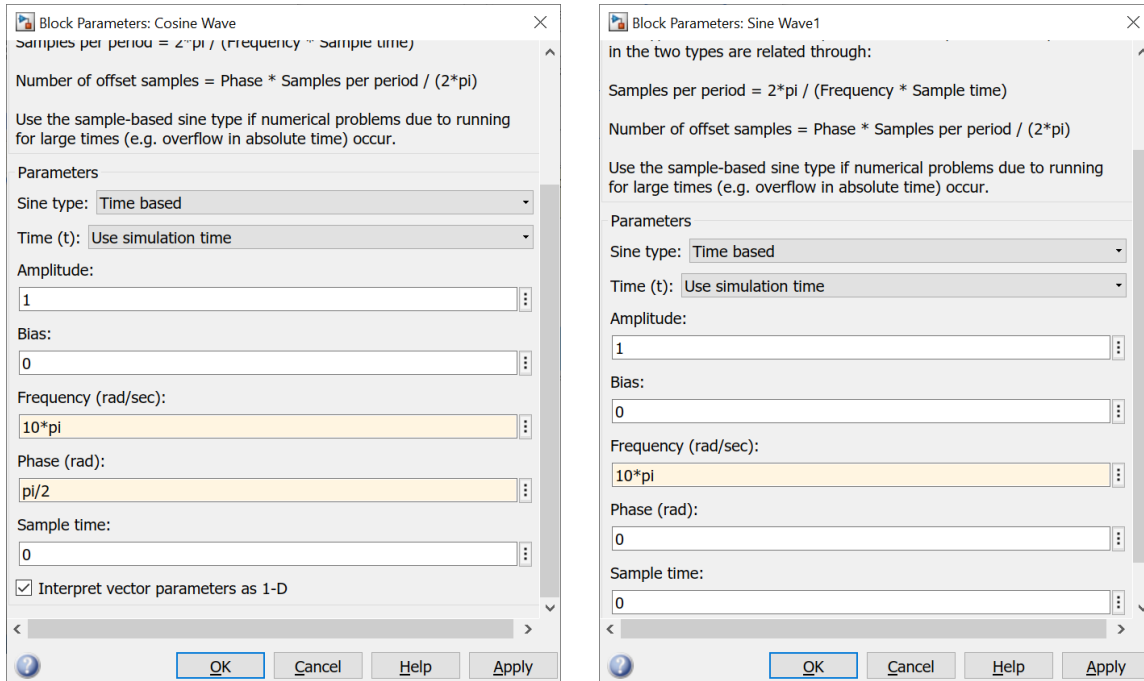
เริ่มต้นจากการนำ Block ต่าง ๆ ดังภาพที่ 11.15 จาก Library Browser มาต่อเป็นวงจร



ภาพที่ 11.15 การนำ Block ต่าง ๆ จาก Library Browser

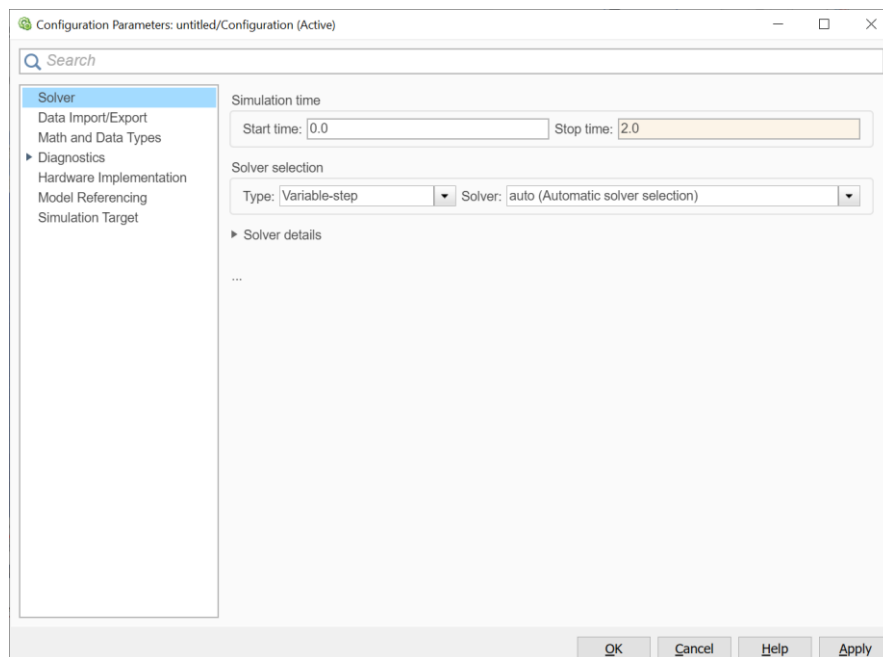
จากนั้น กำหนดค่าพารามิเตอร์ของ Block ต่าง ๆ ดังนี้

1. ค่าพารามิเตอร์ของ Block Cosine wave และ Sine wave กำหนดค่าดังภาพที่ 11.16



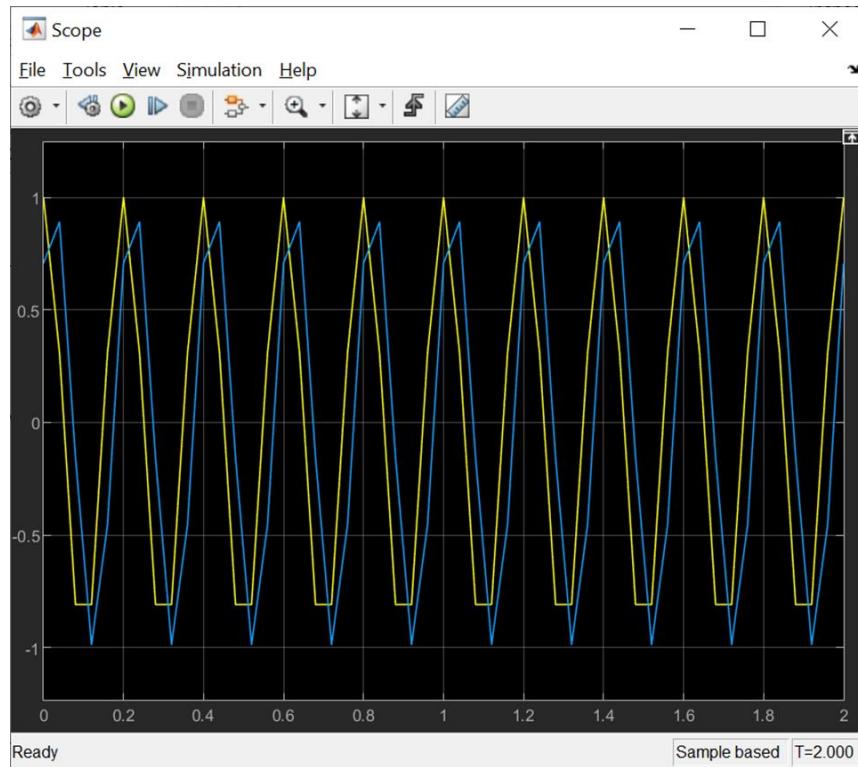
ภาพที่ 11.16 ค่าพารามิเตอร์ของ Block Cosine wave และ Sine wave

2. ตั้งค่าพารามิเตอร์ Start time ที่ 0.0 และ Stop time ที่ 2.0 ดังภาพที่ 11.17



ภาพที่ 11.17 ค่าพารามิเตอร์ Start time และ Stop time

3. ทำการรันโปรแกรม โดยการดับเบิลคลิกที่ Block Scope เพื่อดูผลการจำลอง ดังภาพที่ 11.18

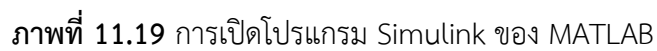


ภาพที่ 11.18 ผลการจำลองการทำงานของระบบ

11.4 การเขียนโปรแกรม Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์

การเขียนโปรแกรม Simulink ของ MATLAB เป็นการจำลองผลการทำงานของระบบต่าง ๆ ตัวอย่างที่นำมาใช้ในการจำลองผลการทำงานนี้เกี่ยวกับการทดสอบการสุ่มตัวอย่างสัญญาณ (Sampling) ที่ใช้โปรแกรม Simulink ของ MATLAB ในการดำเนินงาน ดังนี้

1. ทำการเปิดโปรแกรม Simulink ของ MATLAB ดังภาพที่ 11.19



กรณีที่ 1 คือ $f_s > 20 \text{ Hz}$ โดยกำหนดให้ความถี่ $f_s = 100 \text{ Hz}$ ดังภาพที่ 11.20



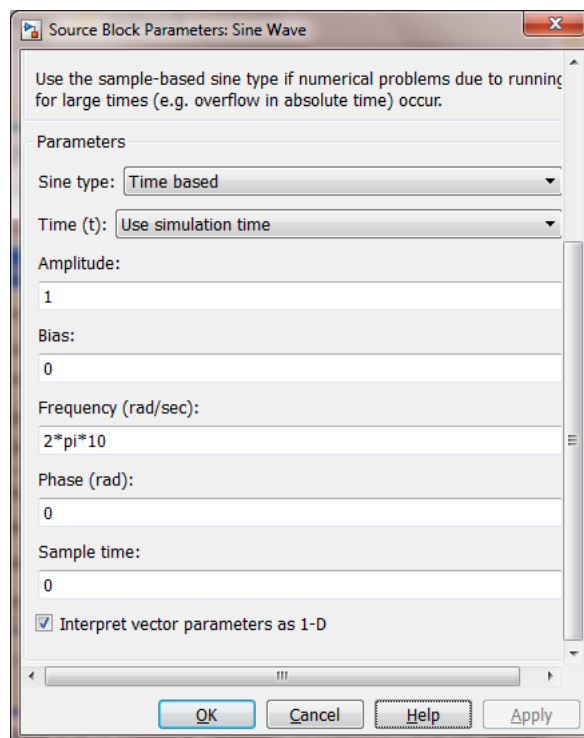
จากภาพที่ 11.20 เป็นการทดสอบโดยใช้ความถี่สัญญาณ $f_s = 100 \text{ Hz}$ โดยสามารถที่จะเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ดังตารางที่ 11.1

ตารางที่ 11.1 การเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ในกรณีที่ 1

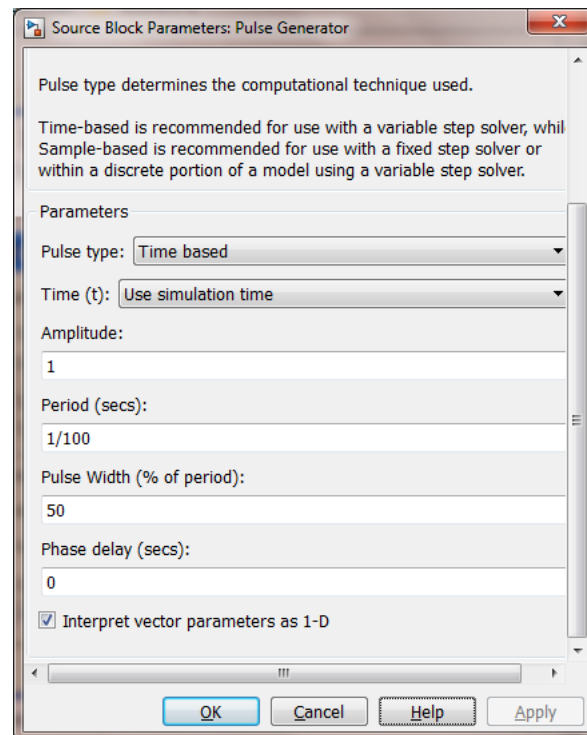
ส่วนประกอบของโมเดล	ตัวแปร	ค่า
Sine Wave	Amplitude	1
	Frequency (rad/sec)	$2\pi \cdot 10$
	Phase (rad)	0
	Sample time	0
Pulse Generator	Period (sec)	1/100
	Pulse width (% of period)	50
	Amplitude	1
	Start time	0
Dot Product	-	-
Scope(1,2,3)	-	-

11.5 การปรับค่าโปรแกรม Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์

จากตารางที่ 11.1 การเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ นั้น สามารถทำการปรับค่าในแต่ละ Block จากนั้นคลิกปุ่ม OK เพื่อทำการบันทึก ดังภาพที่ 11.21 – ภาพที่ 11.22

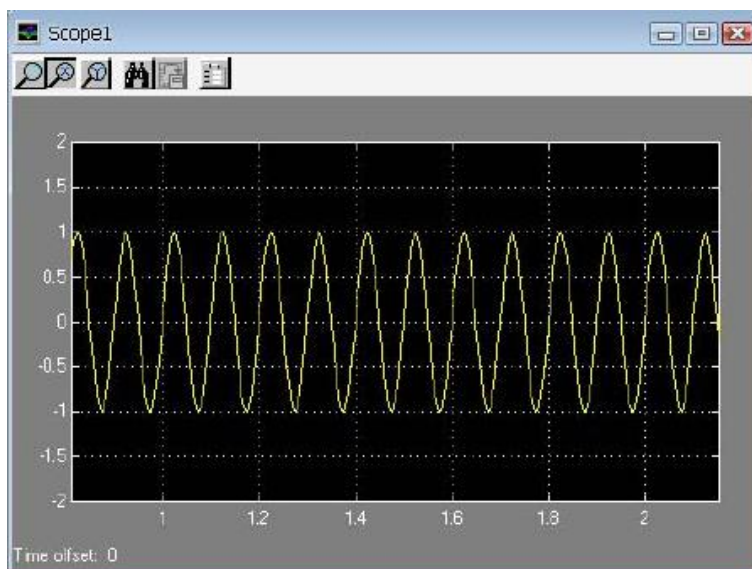


ภาพที่ 11.21 การใส่ค่าภายใน Simulation Parameters ของ Block Sine Wave



ภาพที่ 11.22 การใส่ค่าภายใน Simulation Parameters ของ Block Pulse Generator

จากนั้นทำการทดสอบโปรแกรม Simulink โดยการกดปุ่ม run เพื่อแสดงผลการจำลองสัญญาณสุ่มตัวอย่าง ดังภาพที่ 11.23 – ภาพที่ 11.25



ภาพที่ 11.23 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 100 \text{ Hz}$ วัดจากสัญญาณ scope



ภาพที่ 11.24 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 100 \text{ Hz}$ วัดจากสัญญาณ scope1



ภาพที่ 11.25 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 100 \text{ Hz}$ วัดจากสัญญาณ scope2

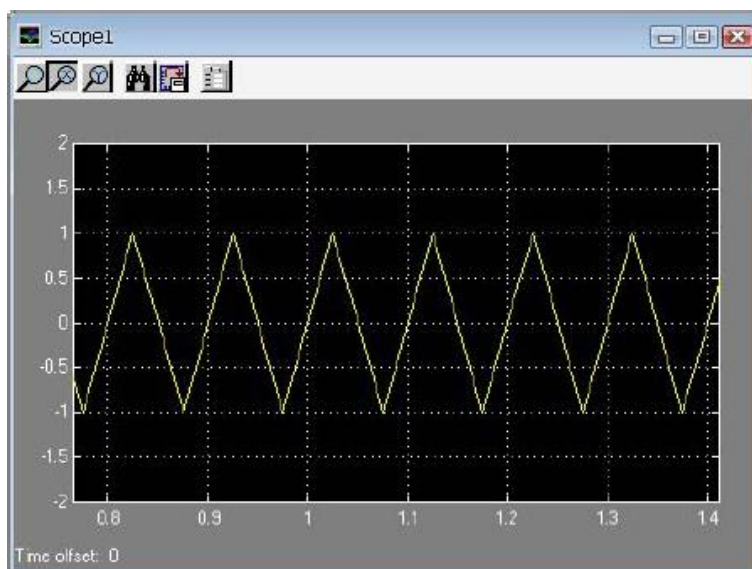
จากภาพที่ 11.23 – ภาพที่ 11.25 พบว่า ผลการทดสอบโดยใช้ความถี่สุ่มตัวอย่างสัญญาณความถี่ $f_s = 100 \text{ Hz}$ มีลักษณะที่เหมือนกับสัญญาณ Sine Wave มากที่สุด แต่จะมีสัญญาณบางส่วนที่ขาดหายไปนั่นคือ สัญญาณสุ่มตัวอย่าง สำหรับการทดสอบกรณีที่ 2 และ กรณีที่ 3 สามารถทำได้เหมือนกรณีที่ 1

กรณีที่ 2 คือ $f_s = 20 \text{ Hz}$ โดยกำหนดให้ความถี่ $f_s = 20 \text{ Hz}$ สามารถที่จะเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ดังตารางที่ 11.2

ตารางที่ 11.2 การเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ในกรณีที่ 2

ส่วนประกอบของโมเดล	ตัวแปร	ค่า
Sine Wave	Amplitude	1
	Frequency (rad/sec)	$2\pi \cdot 10$
	Phase (rad)	0
	Sample time	0
Pulse Generator	Period (sec)	1/20
	Pulse width (% of period)	50
	Amplitude	1
	Start time	0
Dot Product	-	-
Scope(1,2,3)	-	-

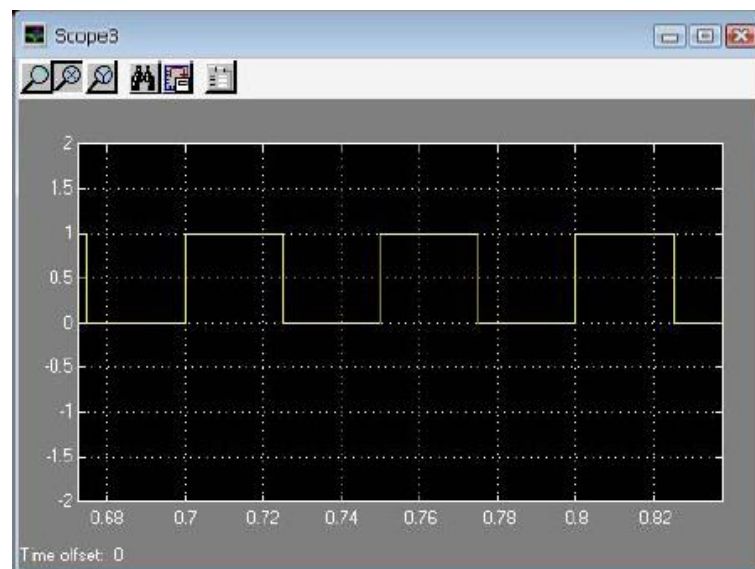
เมื่อทำการปรับค่าพารามิเตอร์ต่าง ๆ ตามตารางที่ 11.2 แล้ว จากนั้นทำการทดสอบ โปรแกรม Simulink โดยการกดปุ่ม run เพื่อแสดงผลการจำลองสัญญาณคู่ตัวอย่าง ดังภาพที่ 11.26 – ภาพที่ 11.28



ภาพที่ 11.26 ผลการจำลองสัญญาณคู่ตัวอย่าง $f_s = 20 \text{ Hz}$ วัดจากสัญญาณ scope



ภาพที่ 11.27 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 20 \text{ Hz}$ วัดจากสัญญาณ scope1



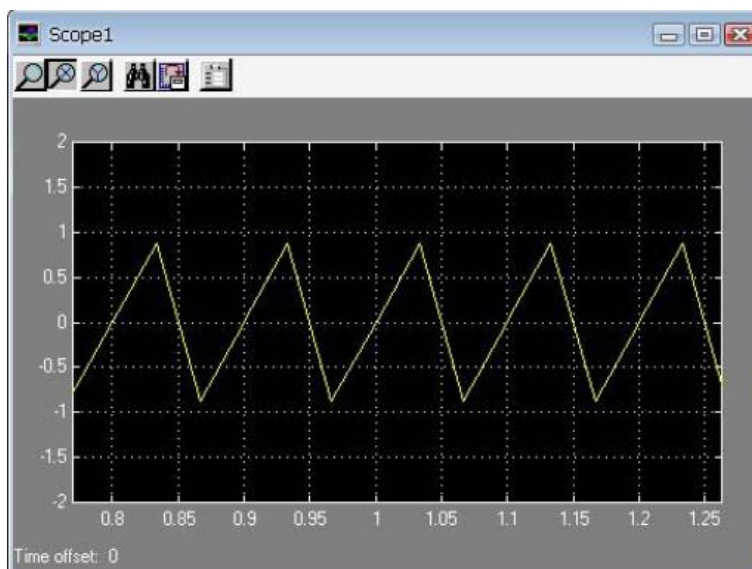
ภาพที่ 11.28 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 20 \text{ Hz}$ วัดจากสัญญาณ scope2

กรณีที่ 3 คือ $f_s < 20 \text{ Hz}$ โดยกำหนดให้ความถี่ $f_s = 15 \text{ Hz}$ สามารถที่จะเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ดังตารางที่ 11.3

ตารางที่ 11.3 การเลือกใช้ Simulink Model จาก Library และการกำหนดค่าต่าง ๆ ในกรณีที่ 3

ส่วนประกอบของโมเดล	ตัวแปร	ค่า
Sine Wave	Amplitude	1
	Frequency (rad/sec)	$2\pi \cdot 10$
	Phase (rad)	0
	Sample time	0
Pulse Generator	Period (sec)	1/15
	Pulse width (% of period)	50
	Amplitude	1
	Start time	0
Dot Product	-	-
Scope(1,2,3)	-	-

เมื่อทำการปรับค่าพารามิเตอร์ต่าง ๆ ตามตารางที่ 11.3 แล้ว จากนั้นทำการทดสอบ โปรแกรม Simulink โดยการกดปุ่ม run เพื่อแสดงผลการจำลองสัญญาณคู่ตัวอย่าง ดังภาพที่ 11.29 – ภาพที่ 11.31



ภาพที่ 11.29 ผลการจำลองสัญญาณคู่ตัวอย่าง $f_s = 15 \text{ Hz}$ วัดจากสัญญาณ scope



ภาพที่ 11.30 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 15 \text{ Hz}$ วัดจากสัญญาณ scope1



ภาพที่ 11.31 ผลการจำลองสัญญาณสุ่มตัวอย่าง $f_s = 15 \text{ Hz}$ วัดจากสัญญาณ scope2

11.6 สรุป

การประยุกต์ใช้งาน Simulink ซึ่งเป็นเครื่องมือ Toolbox ที่อยู่ในโปรแกรม MATLAB โดยจะทำงานภายใต้หน้าต่างที่เป็นการเชื่อมต่อทางรูปภาพ (GUI) ของ Simulink เท่านั้น ซึ่งเป็นการนำ Block ในหน้าต่าง Library-Simulink มาต่อกันตามที่เรต้องการ และสามารถจำลองระบบได้ทั้งระบบที่เป็นเชิงเส้นและไม่เป็นเชิงเส้น ระบบเวลาต่อเนื่องและไม่ต่อเนื่อง การจำลองระบบสามารถกระทำได้โดยป้อนอินพุตที่ป้อนเข้าไป

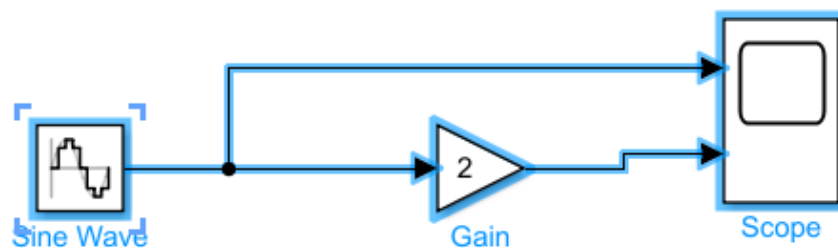
เมื่อเริ่มใช้งาน Simulink จะพบหน้าต่าง Simulink Library Browser ประกอบด้วย Block Diagram ที่ถูกจัดเป็นหมวดหมู่ต่าง ๆ การเลือกใช้งานในหมวดหมู่ใดในหน้าต่าง Simulink Library Browser ให้คลิกที่ไอคอนของหมวดหมู่นั้นจะปรากฏหน้าต่างรายละเอียดของ Block Diagram บนหน้าต่างของ Simulink Library ในการจำลองการทำงานของระบบ ซึ่งอุปกรณ์ต่าง ๆ ไม่สามารถทำงานได้เพียงตัวอุปกรณ์เดียวจะต้องมีการเชื่อมต่อกับอุปกรณ์ตัวอื่น ๆ จึงจะทำงานได้

การเรียกใช้ Block พื้นฐานของ Simulink ของ MATLAB เมื่อต้องการสร้างแบบจำลองบน Simulink ให้ผู้ใช้คลิกที่ไอคอนรูปกระดาษเพื่อสร้างแผ่นงานใหม่ จากนั้นนำ Block แต่ละตัวมาต่อกันอย่างเหมาะสมตามสมการทางคณิตศาสตร์ ซึ่งอธิบายการทำงานของระบบที่เราต้องการจำลอง เมื่อทำการบันทึกแบบจำลองที่สร้างขึ้น จะถูกเก็บไว้ในรูป File ที่มีนามสกุล .mdl

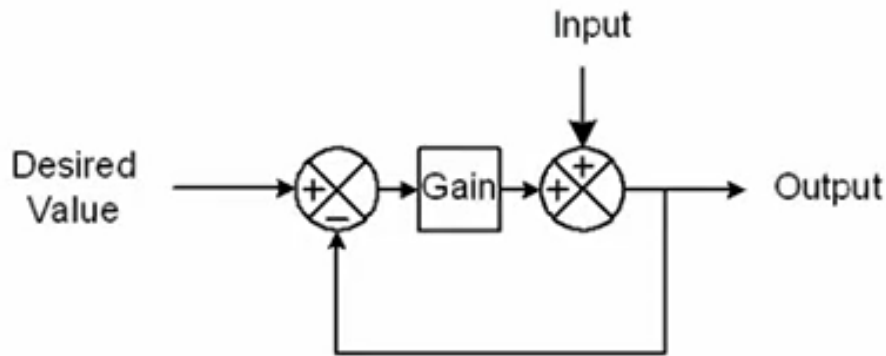
แบบฝึกหัด

บทที่ 11 การประยุกต์ใช้ Simulink ของ MATLAB สำหรับงานไฟฟ้าและอิเล็กทรอนิกส์

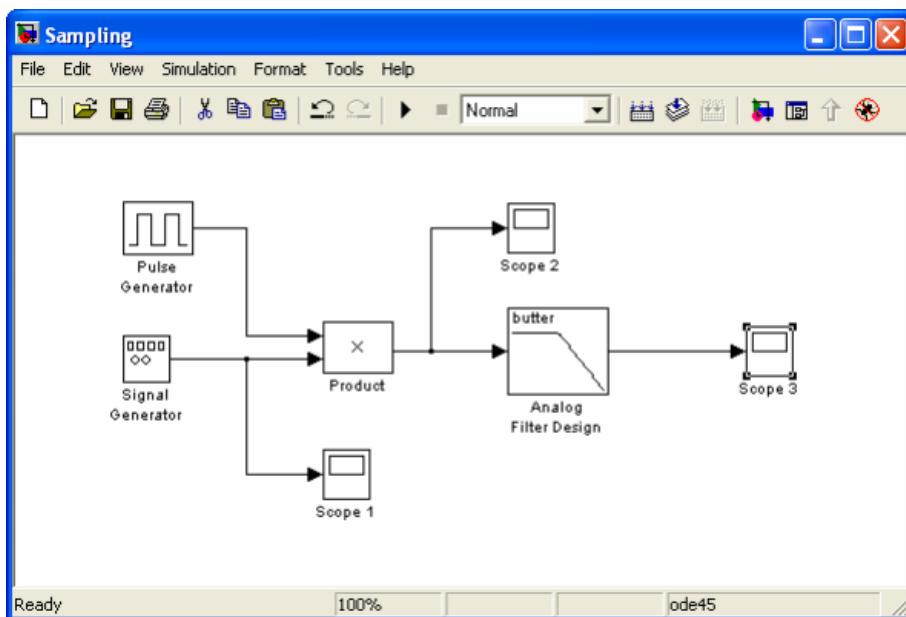
1. จงอธิบายการเรียกใช้ Simulink ของ MATLAB
2. จงอธิบายส่วนประกอบต่าง ๆ บนหน้าต่าง Library Simulink ของ MATLAB
3. จงอธิบายการเรียกใช้ Block พื้นฐาน ด้วย Simulink ของ MATLAB
4. จงอธิบายวิธีการเรียกใช้ Block พิเศษชนิดอื่น ด้วย Simulink ของ MATLAB
5. จงประยุกต์ใช้ Simulink ของ MATLAB สำหรับการใช้งาน Cosine Wave และ Scope โดยกำหนดค่า Amplitude = 10 และค่า Frequency = $2\pi \cdot 50$ rad/sec
6. จงประยุกต์ใช้ Simulink ของ MATLAB สำหรับการใช้งาน sine Wave และ Scope โดยกำหนดค่า Amplitude = 20 และค่า Frequency = $2\pi \cdot 50$ rad/sec
7. จงทำการจำลองเพื่อการออกแบบระบบควบคุมชนิดลูปปิดการวางตำแหน่ง Poles (Closed loop pole placement design) โดยกำหนดให้ $S_1 = -0.0019 + j3.28$, $S_2 = -0.0019 - j3.28$ โดยใช้การประยุกต์ใช้ Simulink ของ MATLAB
8. จงทำการจำลองระบบควบคุมชนิด Open loop โดยกำหนดให้มี Gain ac supply ; Gain = $220/110 = 2$



9. จงทำการจำลองระบบควบคุมชนิด Closed loop โดยกำหนดให้มี ac supply ระหว่าง 100 – 250 V ซึ่งกำหนดให้อาท์พุต Gain = 0.1 , Gain = 1 และ Gain = 10



10. จงประยุกต์ใช้ Simulink ของ MATLAB เกี่ยวกับการสุ่มตัวอย่างสัญญาณ (Sampling) โดยทำการออกแบบโมเดล และตั้งค่าตัวแปรภายในโมเดล จากนั้นทำการแสดงสัญญาณต่าง ๆ



การออกแบบโมเดลการสุ่มตัวอย่างสัญญาณ (Sampling)

ตาราง การตั้งค่าตัวแปรภายในโมเดลการสุ่มตัวอย่างสัญญาณ (Sampling)

ส่วนประกอบของโมเดล	ตัวแปร	ค่าที่กำหนด
Pulse Generator	Pulse type Amplitude Period (secs) Pulse Width (% of period) Phase delays (secs) Interpret vector parameters as 1-D	Timebased 1 0.01 1 0 Yes
Product	Wave form Amplitude Frequency Units Interpret vector parameters as 1-D	Sine 1 5 Hertz Yes
Analog Filter Design	Design method Filter type Filter order Passband edge frequency (rads/sec)	Butterworth Lowpass 8 $5 \times 2 \times \pi$

บรรณานุกรม

- เจษฎา ชินรุ่งเรือง. 2557. **ทฤษฎีวงจรไฟฟ้าเบื้องต้น**. กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.
- ธันต์ ศรีสกุล, ภูมิพร ศิลาพันธ์ และมนตรี ศิริปรัชญานันท์. 2551. การทดสอบเทคนิคการเชื่อมต่อสัญญาณด้วยโปรแกรม MATLAB แบบ Simulink Model. นิตยสาร What Electronics. กรกฎาคม. 48-54.
- ธีรวัฒน์ ประกอบผล. 2558. **การออกแบบและพัฒนาโปรแกรม**. กรุงเทพมหานคร : ชัคเชส มีเดีย.
- ธีรวัฒน์ ประกอบผล และจันทนา ผ่องเพ็ญศรี. 2551) **สถาปัตยกรรมคอมพิวเตอร์**. กรุงเทพมหานคร : ส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).
- ปราโมทย์ เดชะอำไพ และคณะ. 2560. **พื้นฐานแมทแลบ**. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย.
- ปริญญญา สงวนสัตย์. 2556. **คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์**. กรุงเทพมหานคร : ไอทีซี พรีเมียร์.
- เพ็ญศรี ศิวิโรจน์ และคณะ. 2557. **คอมพิวเตอร์และระบบปฏิบัติการ**. เอกสารการสอน สาขาวิชา วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช.
- พิสิฐ สอนละ และคณะ. 2560. **การพัฒนาโปรแกรมจำลองค่าพารามิเตอร์สายส่งความถี่สูงโดยใช้ Matlab/Simulink**. รายงานการวิจัย มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร.
- รัชลิดา ลิปิกรณ์. 2558. **MATLAB การประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า**. กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.
- รุ่งอรุณ พรเจริญ. 2563. **เอกสารประกอบการสอน รายวิชา IE2152212 ปฏิบัติการเขียนโปรแกรมคอมพิวเตอร์เพื่อการประยุกต์ทางวิศวกรรมอิเล็กทรอนิกส์**. คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร.
- _____. 2561. **เอกสารคำสอน รายวิชา 52133201 การโปรแกรมคอมพิวเตอร์สำหรับงานอิเล็กทรอนิกส์และโทรคมนาคม**. คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร.
- _____. 2559. **การพัฒนาชุดสื่อประสมเรื่องสายอากาศไมโครสตริปสำหรับการเรียนการสอนทางด้านวิศวกรรมโทรคมนาคม**. รายงานการวิจัย มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร.
- _____. 2559. **การพัฒนาโปรแกรมจำลองการวิเคราะห์ค่าพารามิเตอร์ที่สำคัญของสายอากาศสำหรับประยุกต์ใช้ในการเรียนการสอนด้านวิศวกรรมโทรคมนาคม**. รายงานการวิจัย มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร.
- ลัญฉกร วุฒิสถิธิกุลกิจ และคณะ. 2553. **MATLAB การประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า**. พิมพ์ครั้งที่ 4. กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

ศรัณย์ ชูคติ และสมศักดิ์ อรรถทิมากุล. การพัฒนาโปรแกรมจำลองสำหรับศึกษาและวิเคราะห์วงจร
กรองความถี่ภายในท่อนำคลื่นด้วยวิธีการวนรอบของคลื่น. วารสารวิชาการพระจอมเกล้า
พระนครเหนือ. ปีที่ 22 ฉบับที่ 3 กันยายน – ธันวาคม. 560-571.

โศรฎา แฉียงการ และกนต์ธร ชำนิประศาสน์. การใช้ MATLAB สำหรับงานวิศวกรรม. สืบค้นเมื่อ
วันที่ 4 เมษายน 2561 จาก <https://www.tt-torrent.com/details.php?id=805515>

สมมาตร ขำเกลี้ยง. 2558. เอกสารประกอบการสอน รายวิชา 6514620 การประยุกต์ใช้
คอมพิวเตอร์ในงานไฟฟ้า. โปรแกรมวิชาอุตสาหกรรมและเทคโนโลยี คณะเทคโนโลยี
มหาวิทยาลัยราชภัฏสงขลา.

สมศักดิ์ อรรถทิมากุล และพินิจ เนื่องภิรมย์. 2557. การใช้งานฟังก์ชัน GUI โปรแกรม MATLAB.
กรุงเทพมหานคร : ศูนย์ผลิตตำราเรียน มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนคร
เหนือ.

สันติ ตันตระกูล. 2558. แมทแลปและการประยุกต์ใช้งานทางการศึกษา. วารสารครุศาสตร์
อุตสาหกรรม. ปีที่ 14 ฉบับที่ 3 กันยายน – ธันวาคม. 1-4.

โอภาส เอี่ยมสิริวงศ์ และสมโชค ชื่นเอี่ยม. 2561. การออกแบบและพัฒนาโปรแกรม.
กรุงเทพมหานคร : ซีเอ็ดดูเคชั่น.

H. Moore. 2015. MATLAB for Engineers (Global Edition). 4rd Edition. London : Pearson
Education

M.J. Roberts. 2018. Signals and Systems: Analysis Using Transform Methods &
MATLAB. 3rd edition Edition. NewYork : McGraw-Hill.

R. F. Harrington. 2001. Time-Harmonic Electromagnetic Fields. IEEE Press. New York
: Reissued original 1961.

R.Porncharoen. 2018. Graphican user interface at Matlab in teaching
telecommunication engineer. Journal of Thai Interdisciplinary Research.
Vol 13 No 5. 44-47.

W.J. Palm. 2000. Introduction to MATLAB 6 for Engineers. NewYork : McGraw-Hill.

ดัชนี

ก	การหาอนุพันธ์, 35 การหาอินทิเกรต, 36
การเขียน Script, 41 การเขียนโปรแกรม GUI ด้วย GUIDE, 99 การเขียนโปรแกรมแบบโครงสร้าง, 54 การเขียนโปรแกรมโดยใช้ Function M-File, 46 การเขียนโปรแกรมโดยใช้ Script M-File, 42 การเรียกใช้ Block พื้นฐานของ Simulink, 195 การเลือกกระทำตามเงื่อนไข, 56 การใช้ Check Box, 109 การใช้ Edit Text, 104 การใช้ List Box, 107 การใช้ Push Button, 101 การใช้ Radio Button, 103 การใช้ Slider Bar, 106 การคำนวณทางพีชคณิต, 17 การจัดการไฟล์, 45 การดำเนินการทางพีชคณิตกับคลาส, 28 การทำงานแบบตามลำดับ, 54 การทำซ้ำ, 63 การรับค่าจากเมาส์, 111 การรับค่าจากแป้นพิมพ์, 110 การวาดกราฟ 3 มิติ, 73 การวาดกราฟ 3 มิติ แบบตาข่ายและบาร์, 81 การวาดกราฟ 3 มิติ แบบพื้นผิว, 85 การวาดกราฟ plot, 88 การวาดกราฟเส้นโดยใช้คำสั่ง plot, 74 การวาดกราฟเส้นโดยใช้คำสั่ง subplot, 78 การสร้าง GUI ด้วย GUIDE, 96 การหาแคลคูลัส, 33 การหาค่า Logarithm, 21 การหาค่าทางตรีโกณมิติ, 19 การหาค่าลิมิตของฟังก์ชัน, 33	ข
	ข้อมูลในเชิงตัวเลข, 15, 23
	ค
	ค่าตัวแปรเฉพาะ, 11 คำสั่ง format hex, 16 คำสั่ง format long e, 16 คำสั่ง format short, 16
	ด
	ตัวแปร, 4 ตัวแปร handles, 99 ตัวแปรวัตถุคลาส, 27 ตัวดำเนินการ, 7 ตัวดำเนินการเปรียบเทียบตรรกะ, 10
	น
	นิพจน์, 4, 29, 38
	บ
	บล็อกไดอะแกรม, 3, 190
	พ
	พีชคณิตเชิงเส้น, 31
	ฟ
	ฟังก์ชัน solve, 31 ฟังก์ชันเรียกกลับ, 100 ฟังก์ชันการวาดกราฟ 2 มิติ, 70 ฟังก์ชันคณิตศาสตร์พื้นฐาน, 11 ฟังก์ชันที่ช่วยลดรูปแบบของสมการ, 30

ร

รูปแบบคำสั่งโดยใช้ if, 56
รูปแบบคำสั่งโดยใช้ if-elseif-else, 58
รูปแบบคำสั่งโดยใช้ switch-case-
 otherwise, 61
รูปแบบคำสั่งโดยใช้ while, 64

ห

หน้าต่าง Command Window, 3, 41
หน้าต่าง Current Folder, 3
หน้าต่าง Details, 3
หน้าต่าง Library, 191
หน้าต่าง Workspace, 4

ส

ส่วนต่อประสานกราฟฟิก, 92, 122

ประวัติผู้เขียน



ผู้ช่วยศาสตราจารย์ ดร.รุ่งอรุณ พรเจริญ
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

การศึกษา

- ปร.ด. สาขาวิจัยและพัฒนการสอนเทคนิคศึกษา
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
- คอ.ม. สาขาวิชาครุศาสตร์ไฟฟ้า
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
- คอ.บ. สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม
สถาบันเทคโนโลยีราชมงคล วิทยาเขตเทเวศร์
- ปวส. ช่างอิเล็กทรอนิกส์-คอมพิวเตอร์
วิทยาลัยเทคนิคนครสวรรค์

งานวิจัยที่สนใจ

การวิจัยเกี่ยวกับการพัฒนาสื่อการเรียนการสอนทางด้านอิเล็กทรอนิกส์และโทรคมนาคม การพัฒนาหลักสูตรฝึกอบรม และการวิจัยเพื่อพัฒนาการเรียนรู้ การพัฒนาโปรแกรมเกี่ยวกับไฟฟ้าและอิเล็กทรอนิกส์

งานวิจัยที่เคยได้รับรางวัล

การพัฒนาแบบจำลองคอมพิวเตอร์ช่วยสอนแบบอัจฉริยะโดยใช้การวิจัยเชิงปฏิบัติการ ได้รับรางวัลการนำเสนอผลงานวิจัยในการประชุมสัมมนาวิชาการนำเสนอผลงานวิจัยระดับชาติและนานาชาติ “ราชภัฏนครสวรรค์วิจัย” ครั้งที่ 1 (ระดับดีเด่น) ณ มหาวิทยาลัยราชภัฏนครสวรรค์ นักวิจัยดีเด่นของมหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร ประจำปี พ.ศ. 2561

