

การโปรแกรมมิ่ง Arduino ด้วย MATLAB & Simulink



อธิบายเบื้องต้นตั้งแต่พื้นฐาน

การเขียนโปรแกรมควบคุมบอร์ด Arduino Uno
การโปรแกรมมิ่งด้วยภาษา MATLAB, Simulink
และ App Designer



ประกอบด้วยตัวอย่าง

การโปรแกรมมิ่งเป็นขั้นตอนแบบง่ายๆ



อ่านเข้าใจง่าย

ตัวอย่างการทดลองพร้อมคำอธิบาย
เหมาะสำหรับผู้เริ่มต้นเขียนโปรแกรม
แบบ GPL และเขียน Coding

เหมาะสำหรับ

นักเรียน นักศึกษาระดับ ปวช. ปวส.
ปริญญาตรี และผู้ที่สนใจเขียนโปรแกรม
ภาษา MATLAB



สุวิทย์ เมษะราษี



ดาวน์โหลดตัวอย่างโปรแกรมใช้งานได้จาก
เว็บไซต์ <https://www.witscode.org>

การโปรแกรมมิ่ง Arduino ด้วย MATLAB & Simulink

โดย สุวิทย์ เมษะราษี

สงวนลิขสิทธิ์ตามกฎหมาย โดย สุวิทย์ เมษะราษี © พ.ศ. 2564

ห้ามคัดลอก ลอกเลียน ดัดแปลง ทำซ้ำ จัดพิมพ์ หรือกระทำการอื่นใด โดยวิธีการใดๆ ในรูปแบบใดๆ
ไม่ว่าส่วนหนึ่งส่วนใดของหนังสือเล่มนี้ เพื่อเผยแพร่ในสื่อทุกประเภท หรือเพื่อวัตถุประสงค์ใดๆ
นอกจากจะได้รับอนุญาต

ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ

สุวิทย์ เมษะราษี.

การโปรแกรมมิ่ง Arduino ด้วย MATLAB & Simulink. --กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2564.
352 หน้า.

1. การเขียนโปรแกรม (คอมพิวเตอร์). 2. ไมโครคอนโทรลเลอร์.

I. ชื่อเรื่อง.

629.895

Barcode (e-book) 9786160841066

ผลิตและจัดจำหน่ายโดย



บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
SE-EDUCATION PUBLIC COMPANY LIMITED

เลขที่ 1858/87-90 ถนนเทพรัตน แขวงบางนาใต้ เขตบางนา กรุงเทพฯ 10260

โทรศัพท์ 0-2826-8000

หากมีคำแนะนำหรือติชม สามารถติดต่อได้ที่ comment@se-ed.com

คำนำ

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ได้รับความนิยมมากในปัจจุบัน อาจเป็นเพราะว่ามันง่ายต่อการพัฒนาเนื่องจากมีรูปแบบคำสั่งพื้นฐานที่ไม่ซับซ้อน มีราคาไม่แพง อีกทั้งการพัฒนายังเป็นแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ จึงเข้าถึงกลุ่มผู้ใช้งานตั้งแต่ระดับผู้เริ่มต้นศึกษาใหม่ไปจนถึงนักพัฒนามืออาชีพ ทั้งนี้ ผู้ใช้งานทุกๆ กลุ่มยังสามารถแก้ไข ดัดแปลง เพิ่มเติม และพัฒนาทั้งตัวบอร์ดและโปรแกรมต่อได้อีกด้วย

ในส่วนของการพัฒนาโปรแกรมเพื่อเข้าถึง จัดการและควบคุมบอร์ด Arduino ให้ทำงานตามคำสั่งนั้น คือ โปรแกรม Arduino IDE (Sketch) ซึ่งผู้พัฒนาจะต้องศึกษาโครงสร้างการเขียนโปรแกรม C/C++ Syntax แต่อาจจะมีข้อจำกัดหรือเกิดความไม่สะดวกมากนัก หากมองในมุมมองของนักพัฒนาโปรแกรมผู้เริ่มต้นศึกษาใหม่ ซึ่งตัวโปรแกรม Arduino IDE นั้น จะไม่มีเครื่องมือช่วยในการออกแบบและสร้างโปรเจกต์ แบบมีหน้าต่างทำงานแบบ GUI (Graphical User Interface) ระบบตรวจสอบและวิเคราะห์โค้ดคำสั่งแบบ Advanced Debugging และ ระบบ Diagnostics เป็นต้น ที่มีประสิทธิภาพสูงและรวดเร็ว เพื่อสนับสนุนและส่งเสริมให้ผู้เริ่มต้นใหม่ได้เรียนรู้และฝึกทักษะการพัฒนาโปรแกรม อีกทั้งผู้ใช้งานจะต้องเรียนรู้และจดจำคำสั่ง/ฟังก์ชันต่างๆ มากมาย จึงจะสามารถพัฒนาโปรแกรม วิเคราะห์ปัญหาและแก้ไขข้อผิดพลาดที่เกิดขึ้นได้อย่างรวดเร็ว มีผลไปถึงโปรแกรมสามารถทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

ด้วยเหตุนี้ ผู้เขียนจึงได้นำเสนอโปรแกรมทางเลือกอีกโปรแกรมหนึ่ง คือ โปรแกรม MATLAB ซึ่งสามารถที่จะอินเทอร์เฟซร่วมกับบอร์ด Arduino ได้ ซึ่งบริษัท MathWorks ได้พัฒนาโมดูลและแพลตฟอร์ม “Arduino Support Package” ที่สามารถเข้าถึง จัดการและควบคุมบอร์ด Arduino ได้อย่างมีประสิทธิภาพ นอกจากนี้ โปรแกรม MATLAB ยังเป็นภาษาระดับสูง สามารถดูแลผลลัพธ์การเข้าถึงและจัดการกับบอร์ดแบบประมวลผลคำสั่งทีละคำสั่ง บนหน้าต่าง “Command

Window” สามารถแสดงผลข้อมูลที่เป็นกราฟและรูปภาพได้อย่างรวดเร็ว อีกทั้งยังสนับสนุนบอร์ดให้ทำงานได้ทั้งแบบ Standalone และ Client-Server บนหน้าต่างทำงานแบบ GUI: Figure อีกด้วย

หนังสือเล่มนี้ เป็นเนื้อหาการฝึกปฏิบัติการเขียนโปรแกรมด้วยภาษา MATLAB 3 รูปแบบ ได้แก่ (1) **แบบเขียนโค้ดคำสั่ง/ไฟล์สคริปต์** (Code Writing) (2) **แบบโอคอนรูปภาพกราฟิก** (Graphical Programming Language: GPL) และ (3) **แบบอินเตอร์เฟซกับผู้ใช้** (Users Interface: **UI Figure**) ซึ่งจะช่วยสนับสนุนและส่งเสริมให้ผู้ใช้งาน ที่ไม่เคยเขียนโปรแกรมหรือมีพื้นฐานการเขียนโปรแกรมมาก่อน ได้เปิดมุมมองการเรียนรู้ ทำความเข้าใจ และฝึกปฏิบัติการใช้งาน MATLAB และ Simulink เพื่อ**ควบคุมการทำงานของบอร์ด “Arduino”** ให้ทำงานตามเงื่อนไขที่ต้องการได้ เนื้อหาการเรียนรู้และโจทย์การฝึกปฏิบัติได้นำเสนอเป็นลำดับขั้นตอนแบบง่ายๆ ไม่ซับซ้อน โดยหวังเป็นอย่างยิ่งว่าผู้อ่านจะสามารถนำไปประยุกต์ใช้งานเพื่อสร้างสรรค์โปรเจกต์และโครงการของตนเองได้

สุดท้ายนี้ ขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต ที่เป็นแรงผลักดันและแรงบันดาลใจทำให้ผู้เขียนนำความรู้และทักษะวิชาชีพมาบูรณาการในการจัดทำหนังสือเล่มนี้ให้สำเร็จลุล่วงได้ด้วยดี คุณค่าที่ได้จากหนังสือเล่มนี้ ขอนอบน้อมบูชาคุณมารดาบิดา คุณครูอาจารย์ และผู้มีพระคุณทุกๆ ท่าน หากผิดพลาดหรือขาดตกบกพร่องประการใด ขออภัยมา ณ โอกาสนี้ด้วย

สุวิทย์ เมาะราชิ

สารบัญ

บทที่ 1	แนวคิดในการเขียนโปรแกรม.....	9
1.1	ความหมายของโปรแกรมคอมพิวเตอร์.....	9
1.2	ความสำคัญและการทำงานของโปรแกรมคอมพิวเตอร์.....	9
1.3	การเขียนโปรแกรมเชิงโครงสร้าง.....	11
1.4	การเขียนโปรแกรมเชิงวัตถุ.....	16
	แบบฝึกหัดท้ายบท.....	19
บทที่ 2	แนะนำโปรแกรม MATLAB และ MATLAB Simulink.....	21
2.1	คุณลักษณะพิเศษของโปรแกรม MATLAB.....	22
2.2	ลักษณะการโปรแกรมมิ่งและการทำงานของโปรแกรม MATLAB.....	23
2.3	จุดเริ่มต้นของโปรแกรม MATLAB.....	23
2.4	ส่วนประกอบสำคัญของ MATLAB.....	25
2.5	โปรแกรม MATLAB Simulink.....	30
	แบบฝึกหัดท้ายบท.....	38
บทที่ 3	แนะนำบอร์ดไมโครคอนโทรลเลอร์ Arduino	39
3.1	พัฒนาการของบอร์ดไมโครคอนโทรลเลอร์ Arduino.....	41
3.2	ไมโครคอนโทรลเลอร์ Arduino Uno R3.....	42
	แบบฝึกหัดท้ายบท.....	46

บทที่ 4 การติดตั้ง Simulink Supporting Blocksets สำหรับ Arduino.....47

- 4.1 การติดตั้งชุดแพ็คเกจ Simulink Supporting Blocksets.....50
- 4.2 การทดสอบความพร้อมใช้งาน Simulink Supporting Blocksets.....58
- แบบฝึกหัดท้ายบท.....61

บทที่ 5 การใช้งาน MATLAB Desktop และ MATLAB Simulink เบื้องต้น.....65

- 5.1 การใช้งาน MATLAB Desktop เบื้องต้น.....65
- 5.2 MATLAB Desktop กับการคำนวณทั่วไป.....73
- 5.3 การใช้งาน MATLAB Simulink เบื้องต้น.....77
- 5.4 MATLAB Simulink กับการคำนวณทั่วไป.....79
- แบบฝึกหัดท้ายบท.....95

บทที่ 6 เตรียมความพร้อม Hardware และ Software Support Package.....97

- 6.1 การเตรียมเครื่องคอมพิวเตอร์ และการปรับตั้งค่าโปรแกรม MATLAB.....98
- แบบฝึกหัดท้ายบท.....117

บทที่ 7 การทดลองที่ 1 Blink LED..... 119

- 7.1 การทดลองที่ 1.1 “Blink LED 1”.....119
- 7.2 การทดลองที่ 1.2 “Blink LED 2”.....128
- แบบฝึกหัดท้ายบท.....134

บทที่ 8 การทดลองที่ 2 สวิตช์ควบคุม LED 135

- 8.1 การทดลองที่ 2.1 “LED Controller 1”.....136
- 8.2 การทดลองที่ 2.2 “LED Controller 2”.....142
- แบบฝึกหัดท้ายบท.....149

บทที่ 9 การทดลองที่ 3 Analog Input 151

- 9.1 การทดลองที่ 3.1 “Analog Input 1”.....152
- 9.2 การทดลองที่ 3.2 “Analog Input 2”.....158
- 9.3 การทดลองที่ 3.3 “Analog Input 3 : Photo Cell Switch”.....164
- แบบฝึกหัดท้ายบท.....171

บทที่ 10	การทดลองที่ 4 RC Servo Motor Controller.....	173
10.1	การทดลองที่ 4.1 RC Servo Motor Control using Pulse Generator.....	174
10.2	การทดลองที่ 4.2 RC Servo Motor Control using Slider Gain.....	180
10.3	การทดลองที่ 4.3 RC Servo Motor Control using Potentiometer.....	185
	แบบฝึกหัดท้ายบท.....	191
บทที่ 11	การทดลองที่ 5 Light Meter Project	193
11.1	การทดลองที่ 5 Light Meter Project.....	194
	แบบฝึกหัดท้ายบท.....	201
บทที่ 12	การโปรแกรมมิ่ง MATLAB Desktop กับ Arduino.....	203
12.1	การทดลองที่ 6.1 ทดสอบเชื่อมต่อ Arduino กับคอมพิวเตอร์.....	204
12.2	การทดลองที่ 6.2 ควบคุม LED ด้วยคำสั่งควบคุมบน Command Line.....	207
12.3	การทดลองที่ 6.3 ควบคุมหลอด LED ด้วยการรันไฟล์สคริปต์ (.m file).....	212
12.4	การทดลองที่ 6.4 ควบคุมหลอด LED ทำงานแบบ “Real-time”.....	217
12.5	การทดลองที่ 6.5 ควบคุมหลอด LED 2 ดวง.....	220
	แบบฝึกหัดท้ายบท.....	227
บทที่ 13	การโปรแกรมมิ่ง MATLAB Desktop กับ Arduino (ต่อ) (Arduino Input/Output: IO)	229
13.1	การทดลองที่ 7.1 สวิตช์ควบคุม LED.....	230
13.2	การทดลองที่ 7.2 สวิตช์ควบคุม blink LED.....	235
13.3	การทดลองที่ 7.3 Analog Input กับความต้านทานปรับค่าได้.....	239
13.4	การทดลองที่ 7.4 Analog Input กับ Light Dependent Resistor; LDR.....	244
13.5	การทดลองที่ 7.5 Photo Cell Switch.....	249
13.6	การทดลองที่ 7.6 RC Servo Motor.....	254
13.7	การทดลองที่ 7.7 ควบคุม Servo Motor ด้วย Potentiometer.....	260
	แบบฝึกหัดท้ายบท.....	269

บทที่ 14	การโปรแกรมมิ่ง MATLAB GUI กับ Arduino.....	271
14.1	การทดลองที่ 8.1 LED “On-Off”	272
14.2	การทดลองที่ 8.2 Analog and Digital Input Monitoring.....	298
	แบบฝึกหัดท้ายบท.....	328
ภาคผนวก	การติดตั้งโปรแกรม MATLAB.....	331
	การติดตั้งโปรแกรม MATLAB.....	332
บรรณานุกรม		351



Unit
1

แนวคิดในการเขียนโปรแกรม (Concept of Programming)

1.1 ความหมายของโปรแกรมคอมพิวเตอร์

โปรแกรมคอมพิวเตอร์ หมายถึงชุดคำสั่งที่มีหน้าที่สั่งการให้เครื่องคอมพิวเตอร์หรืออุปกรณ์ฮาร์ดแวร์อื่นๆ ทำการประมวลผลข้อมูลตามที่ผู้เขียนต้องการ หรือเรียกอีกอย่างหนึ่งว่า **ซอฟต์แวร์คอมพิวเตอร์** ซึ่งคำสั่งแต่ละคำสั่งจะออกแบบให้ทำงานเฉพาะบางอย่าง เมื่อนำมารวมเป็นโปรแกรมหรือรวมเป็นชุดคำสั่งแล้ว จะสามารถสั่งงานให้คอมพิวเตอร์ทำงานได้งานที่หลากหลายและประมวลผลงานที่มีความซับซ้อนได้

1.2 ความสำคัญและการทำงานของโปรแกรมคอมพิวเตอร์

คอมพิวเตอร์ในยุคแรกๆ ใช้คนเป็นผู้ควบคุมการทำงาน จึงจะทำงานได้อย่างมีประสิทธิภาพเมื่อระบบคอมพิวเตอร์ได้รับการพัฒนา มีความเร็วและประสิทธิภาพมากขึ้น ในขณะเดียวกันคนจะมีข้อจำกัด ไม่สามารถพัฒนาความสามารถและความเร็วให้เท่าเทียมกับคอมพิวเตอร์ได้ การพัฒนาระบบคอมพิวเตอร์ในยุคต่อมาจึงเป็นการพัฒนาชุดคำสั่งเพื่อเพิ่มประสิทธิภาพของระบบคอมพิวเตอร์ให้ทำงานได้อย่างเต็มศักยภาพและสมรรถนะของมัน ชุดของคำสั่งนี้เรียกว่า **โปรแกรมคอมพิวเตอร์** หรือ **ซอฟต์แวร์คอมพิวเตอร์**

การเขียนโปรแกรมเพื่อให้คอมพิวเตอร์ทำงานตามคำสั่งและเงื่อนไขตามที่ต้องการได้นั้นขึ้นอยู่กับความสามารถและประสบการณ์ของผู้เขียนโปรแกรม ผู้เขียนโปรแกรมจะเป็นผู้เลือกใช้ภาษาใดๆ ขึ้นอยู่กับลักษณะของงานหรือตามความถนัดของตนเอง อีกทั้งในปัจจุบัน ภาษาคอมพิวเตอร์มีมากมาย ภาษาหลักๆ ได้แก่ ภาษา **Basic**, **C**, **Python** เป็นต้น ภาษาแต่ละภาษาจะมีจุดเด่น คุณลักษณะ และมีความเหมาะสมกับงานที่ต้องการพัฒนามากน้อยแตกต่างกันออกไป

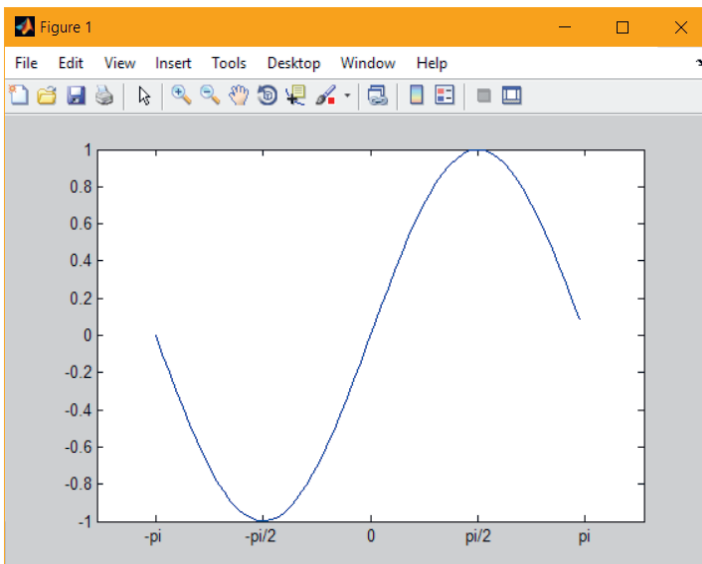
ภาษาที่ใช้เขียนโปรแกรมจะเป็น **ภาษาระดับสูง (High Level Language)** โปรแกรมหรือชุดคำสั่งที่เขียนขึ้นจึงต้องถูกแปลเป็นภาษาเครื่องก่อน แล้วจึงสั่งงานให้คอมพิวเตอร์ทำงานตามคำสั่งนั้นๆ ดังนั้นผู้เขียนโปรแกรมจึงต้องเลือกใช้ซอฟต์แวร์สำหรับการพัฒนาโปรแกรม หรือเขียนโปรแกรมรูปแบบ IDE (Integrated Development Environment) ซึ่งจะมีทูลหรือเครื่องมืออำนวยความสะดวกต่างๆ ไว้สนับสนุนการเขียน/แก้ไขโค้ดคำสั่ง (Code Editor) เครื่องมือตรวจสอบ/แก้ไขข้อผิดพลาดของโปรแกรม (Debugger) เครื่องมือที่ช่วยในการรันคำสั่ง ซึ่งจะต้องอำนวยความสะดวกได้ทั้ง 2 โหมด คือ **Immediate Mode** และ **Script Mode**

การทำงานแบบ Immediate Mode เป็นการพิมพ์คำสั่งทีละคำสั่ง แล้วตัวแปลคำสั่งจะนำคำสั่งนั้นๆ ไปประมวลผลหรือทำงานทันที แสดงดังตัวอย่างโปรแกรม MATLAB Desktop ทำงานแบบ Immediate Mode ดังแสดงในรูปที่ 1.1 ซึ่งจะเหมาะสำหรับการทดสอบฟังก์ชัน หรือชุดคำสั่งที่ต้องการให้ได้ผลตามที่ต้องการก่อนที่จะนำไปเขียนโปรแกรมแบบ Script Mode

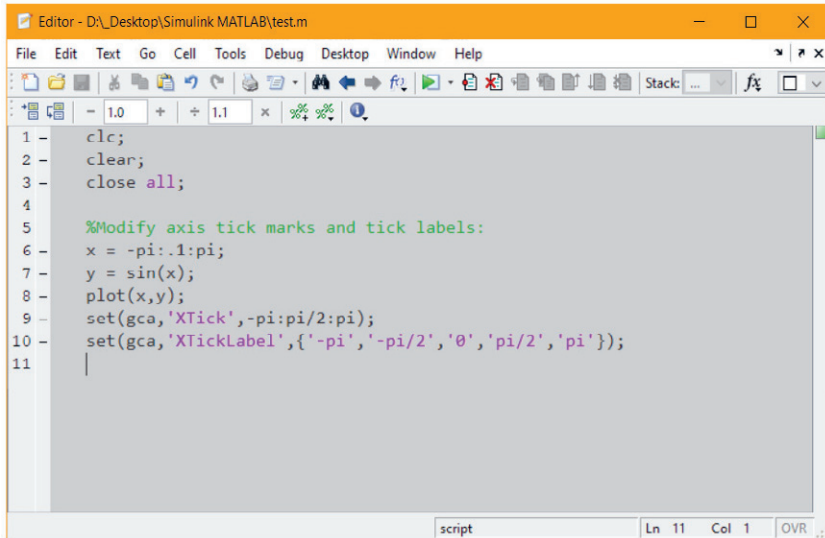


```

Command Window
File Edit Debug Desktop Window Help
>> x = -pi:1:pi;
>> y = sin(x);
>> plot(x,y)
>> set(gca,'XTick',-pi:pi/2:pi)
>> set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})
>> |
  
```



รูปที่ 1.1 การเขียนโปรแกรมแบบ Immediate Mode ด้วยภาษา MATLAB



รูปที่ 1.2 การเขียนโปรแกรมแบบ Script Mode ด้วยภาษา MATLAB

การทำงานแบบ Script Mode เป็นการเขียนโค้ดคำสั่งต่างๆ รวมกันเป็นโปรแกรมแล้วบันทึกไว้เป็นไฟล์ เรียกว่า “ไฟล์สคริปต์” หรือ “ซอร์สโค้ด” เมื่อผู้เขียนโปรแกรมรันไฟล์สคริปต์นั้น ตัวแปลภาษาจะนำโค้ดคำสั่งไปประมวลผลหรือทำงานตั้งแต่บรรทัดแรกไปจนถึงบรรทัดสุดท้าย จะเห็นว่าการรันโปรแกรมในโหมดนี้ จะต้องเขียนโปรแกรมให้เสร็จสมบูรณ์ถึงจะสามารถนำไปรันโปรแกรมได้ แสดงดังตัวอย่างการเขียนโปรแกรมภาษา MATLAB ให้ทำงานแบบ Script Mode ดังแสดงในรูปที่ 1.2 ซึ่งเมื่อรันไฟล์สคริปต์นี้แล้ว จะได้ผลลัพธ์เหมือนกัน ดังแสดงในรูปที่ 1.1

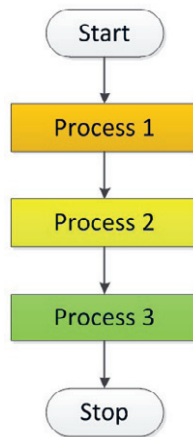
แนวคิดในการเขียนโปรแกรม รูปแบบของแนวคิดในการเขียนโปรแกรม สามารถแบ่งออกได้เป็น 2 แนวคิดใหญ่ๆ คือ การเขียนโปรแกรมเชิงโครงสร้าง และการเขียนโปรแกรมเชิงวัตถุ

1.3 การเขียนโปรแกรมเชิงโครงสร้าง (Structured Programming)

เป็นการจัดการคำสั่งต่างๆ ให้มีรูปแบบและมาตรฐานที่สามารถเขียนโปรแกรมได้ง่าย ตรวจสอบได้ง่าย และสะดวกในการปรับปรุงพัฒนาโปรแกรมต่อไปในอนาคตได้ โดยมีโครงสร้างการทำงานพื้นฐาน 3 รูปแบบ ได้แก่

1. การทำงานแบบตามลำดับ (Sequence)
2. การทำงานแบบมีทางเลือกหรือตามเงื่อนไข (Decision) และ
3. การทำงานแบบทำซ้ำๆ (Loop)

1. การทำงานแบบตามลำดับ (Sequence) ประกอบด้วยคำสั่งหรือชุดคำสั่งที่ไม่มีเงื่อนไขใด ไม่มีการตัดสินใจ และมีทางเข้าทางเดียวและมีทางออกทางเดียว ดำเนินการเรียงลำดับไปอย่างต่อเนื่อง โดยแต่ละขั้นตอนจะมีการประมวลผล (Process) เพียงครั้งเดียว แสดงผังงานดังรูปที่ 1.3 และตัวอย่างโปรแกรม MATLAB กับการทำงานแบบตามลำดับ ดังแสดงในรูปที่ 1.4

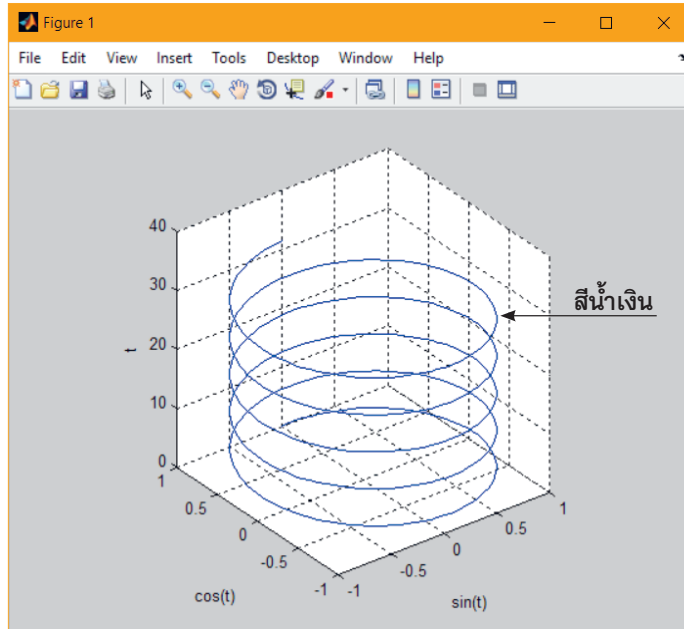


รูปที่ 1.3 ผังงานการทำงานตามลำดับ

```

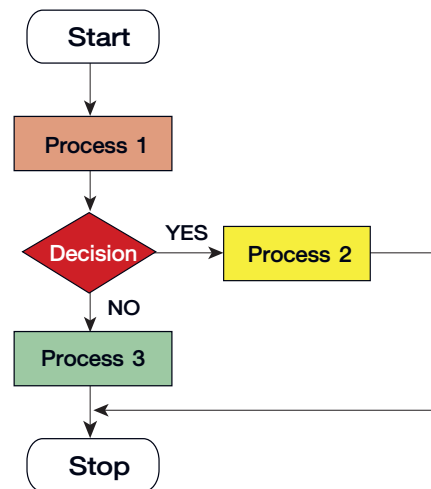
1 - clc;
2 - clear;
3 - close all;
4
5 - %Plot a three-dimensional helix.
6 - t = 0:pi/50:10*pi;
7 - plot3(sin(t),cos(t),t)
8 - xlabel('sin(t)')
9 - ylabel('cos(t)')
10 - zlabel('t')
11 - grid on
12 - axis square
13
    
```

รูปที่ 1.4 โปรแกรม MATLAB กับการทำงานแบบตามลำดับ



รูปที่ 1.4 (ต่อ) ผลลัพธ์จากการรันไฟล์สคริปต์โปรแกรม MATLAB ให้ทำงานแบบตามลำดับ

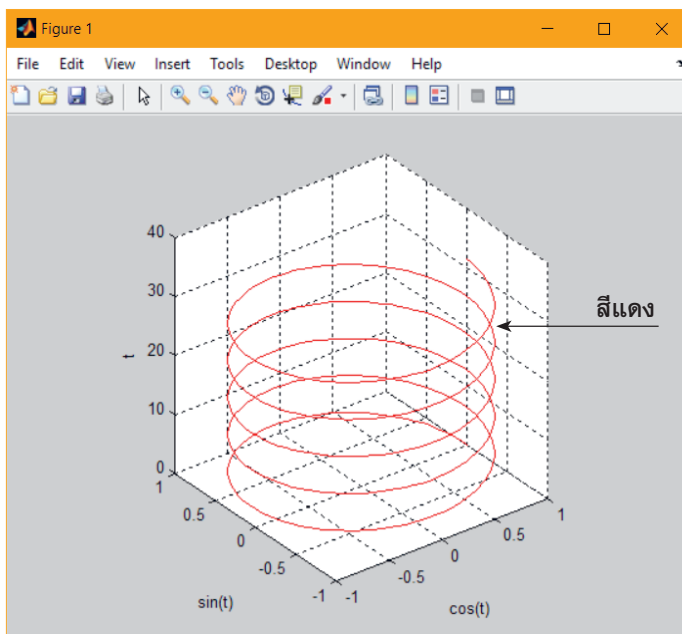
2. การทำงานแบบมีทางเลือกหรือตามเงื่อนไข (Decision) เป็นโครงสร้างการทำงานตามเงื่อนไขหรือมีเงื่อนไข โดยจะทำการตรวจสอบเงื่อนไขว่าเป็น “จริงหรือเท็จ” แล้วดำเนินการตามคำสั่งนั้นตามที่กำหนด แสดงผังงานดังในรูปที่ 1.5 และตัวอย่างโปรแกรม MATLAB กับการทำงานแบบมีทางเลือก ดังแสดงในรูปที่ 1.6 เป็นการเลือกแสดงผลเมื่อเงื่อนไขเป็นจริง คือ ประมวลผลข้อมูลแล้วแสดงผลด้วยเส้นสีแดง ดังรูป ถ้าหากเงื่อนไขเป็นเท็จแล้วจะเปลี่ยนเป็นแสดงผลด้วยเส้นสีน้ำเงิน



รูปที่ 1.5 ผังงานการทำงานแบบตามเงื่อนไข

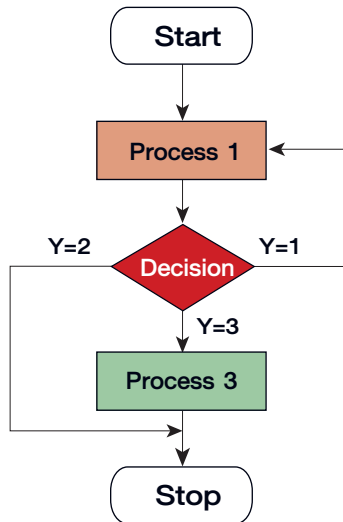
14 การโปรแกรมมิ่ง Arduino ด้วย MATLAB & Simulink

```
Editor - D:\_Desktop\Simulink MATLAB\test_02.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 - clc;
2 - clear;
3 - close all;
4
5 - %Plot a three-dimensional helix.
6 - t = 0:pi/50:10*pi;
7 - swap=true;
8 - if (swap)
9 -     plot3(cos(t),sin(t),t,'color','r')
10 -     xlabel('cos(t)')
11 -     ylabel('sin(t)')
12 - else
13 -     plot3(sin(t),cos(t),t,'color','b')
14 -     xlabel('sin(t)')
15 -     ylabel('cos(t)')
16 - end
17 - zlabel('t')
18 - grid on
19 - axis square
20
script Ln 20 Col 1 OVR
```



รูปที่ 1.6 โปรแกรม MATLAB กับการทำงานแบบตามเงื่อนไข

3. การทำงานแบบทำซ้ำๆ (Loop) เป็นการทำงานลักษณะวนซ้ำไปเรื่อยๆ จนกระทั่งเป็นไปตามเงื่อนไขที่กำหนดไว้จึงจะหยุดการทำงาน หรือไปประมวลผลคำสั่งอื่นๆ ต่อไป แสดงผังงานดังในรูปที่ 1.7 และตัวอย่างโปรแกรม MATLAB กับการทำงานแบบวนซ้ำ ดังแสดงในรูปที่ 1.8

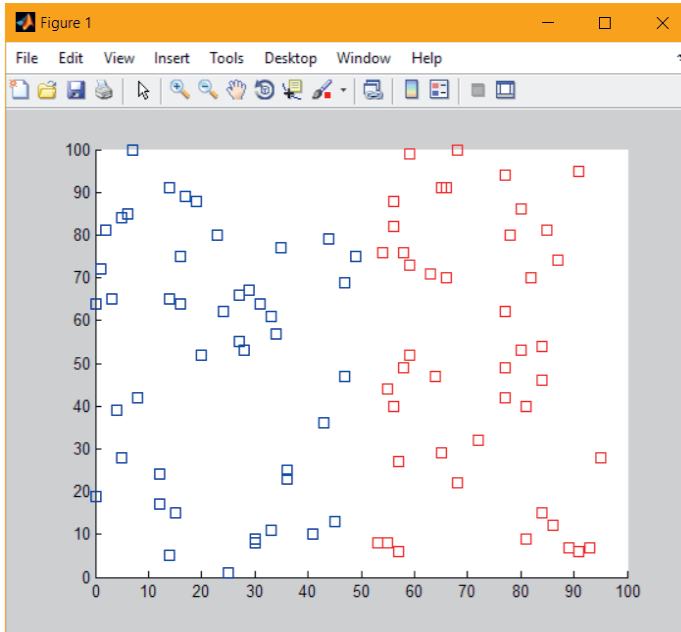


รูปที่ 1.7 ผังงานการทำงานแบบทำซ้ำๆ (Loop)

```

1 - clc;
2 - clear;
3 - close all;
4
5 - loop=0;
6 - figure;
7 - hold on;
8 - while (loop<=90)
9 -     x=round(rand(1,1)*100);
10 -    y=round(rand(1,1)*100);
11 -    color = 'r';
12 -    if (x<=50)
13 -        color='b';
14 -    end
15 -    plot(x,y,'s','color',color);
16 -    loop = loop + 1;
17 - end
  
```

รูปที่ 1.8 โปรแกรม MATLAB กับการทำงานแบบทำซ้ำๆ



รูปที่ 1.8 (ต่อ) ผลลัพธ์จากการรันไฟล์สคริปต์โปรแกรม MATLAB ให้ทำงานแบบทำซ้ำๆ

1.4

การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming; OOP)

การเขียนโปรแกรมเชิงวัตถุ เรียกอีกอย่างหนึ่งว่า **แนวคิดเชิงวัตถุ** ผู้เขียนโปรแกรมหรือผู้พัฒนาโปรแกรมจะต้องมองโปรแกรมเป็นเหมือนวัตถุ โดยจะต้องจำแนกประเภทของวัตถุต่างๆ (Class) มีความเข้าใจรายละเอียดคุณลักษณะของวัตถุ (Attribute) เมื่อมองโปรแกรมคอมพิวเตอร์เทียบเคียงหรือเป็นเสมือนวัตถุหนึ่งๆ แล้ว โปรแกรมจะประกอบด้วยหลายๆ คลาส ซึ่งแต่ละคลาสนั้น จะมีคุณสมบัติหรือคุณลักษณะเฉพาะตัว และแต่ละคลาส จะสามารถสืบทอดคุณสมบัติ (Inheritance) ที่เรียกว่า Subclass ได้ โดยสามารถนำกลับมาใช้ใหม่ (Reusable) คุณสมบัติของโปรแกรมต่างๆ เหล่านี้ จะทำให้ลดขั้นตอนการพัฒนา ลดขนาดโปรแกรม และลดความซับซ้อนลงได้

อีกนัยหนึ่ง OOP คือการเขียนโปรแกรมแบบ **ธรรมชาติของวัตถุ** จะมองโปรแกรมเป็นเหมือน **วัตถุชิ้นหนึ่ง** ซึ่งวัตถุหนึ่ง จะมีคุณลักษณะต่างๆ ได้แก่ มีสีอะไร มีสีแดงหรือสีเขียว มีขนาดเท่าไร ยาวหรือสั้น เป็นต้น ซึ่งก็คือคุณสมบัติหรือคุณลักษณะประจำของวัตถุชิ้นนั้น จึงนำมากำหนดเป็น **“ประเภท”** หรือ **“คลาส”** ให้กับวัตถุชิ้นนั้น

เมื่อ OOP ถูกมองว่าเป็นการเขียนโปรแกรมเปรียบเหมือนวัตถุชิ้นหนึ่ง “วัตถุชิ้นนั้น จะมีคุณลักษณะหรือมีวิธีการใช้งานเฉพาะตัวของมันเอง” กล่าวคือ วัตถุแต่ละชิ้น นอกจากจะมีรูปร่าง ลักษณะที่แตกต่างกันแล้ว วิธีการนำไปใช้งาน ก็จะมี ความแตกต่างกันออกไป เราเรียกคุณลักษณะของวัตถุนี้ว่า **แอตทริบิวต์ (Attribute)** และเรียกวิธีการนำวัตถุไปใช้งานว่า **เมธอด (Method)** เช่น “ปากกาเป็นวัตถุชิ้นหนึ่ง” มีลักษณะเป็นแท่ง ทรงกลม มีขนาดเล็ก และเป็นแท่งยาว ภายในจะมีไส้ซึ่งบรรจุน้ำหมึกไว้ น้ำหมึกจะมีสีแดง สีน้ำเงิน และสีดำ เป็นต้น จะนำไปใช้สำหรับการเขียน วิธีการใช้งานปากกาทำได้โดยใช้มือจับและเขียนลงบนกระดาษ เป็นต้น การนำเอาแนวคิดของ OOP มาใช้ในการเขียนโปรแกรม ผู้พัฒนาโปรแกรมสามารถกำหนดวิธีการทำงานและหน้าที่การเก็บข้อมูลที่แตกต่างกัน โดยไม่ให้มีส่วนที่เกี่ยวข้องกัน วิธีการเขียนโปรแกรมแบบ OOP โดยใช้โปรแกรม MATLAB ดังแสดงในรูปที่ 1.9

```

1 classdef BasicClass
2     properties
3         Value
4     end
5     methods
6         function r = Power(obj)
7             r = obj.Value*obj.Value;
8         end
9         function r = By2n(obj,n)
10            r = obj.Value * n;
11        end
12    end
13end

```

```

>> a = BasicClass;
>> a.Value=50;
>> Power(a)

ans =

    2500

>> By2n(a,4)

ans =

    200
fx >>

```

รูปที่ 1.9 การเขียนโปรแกรมแบบ OOP ด้วยภาษา MATLAB

สรุปท้ายบท

บทนี้ได้เรียนรู้เพื่อทำความเข้าใจเกี่ยวกับความหมาย ความสำคัญ และการทำงานของโปรแกรมคอมพิวเตอร์ โดยภาษาที่ใช้เขียนโปรแกรมจะเป็น “ภาษาระดับสูง” ซึ่งโปรแกรมที่เขียนขึ้นนี้จะถูกแปลเป็นภาษาเครื่องก่อน แล้วจึงสั่งงานให้คอมพิวเตอร์ทำงานตามคำสั่ง โดยใช้ซอฟต์แวร์สำหรับการพัฒนาโปรแกรมที่อยู่ในรูปแบบ IDE (Integrated Development Environment) ซึ่งจะประกอบด้วยเครื่องมือสำหรับอำนวยความสะดวกต่างๆ เพื่อการเขียน/การแก้ไขโค้ดคำสั่ง การตรวจสอบ/การแก้ไขข้อผิดพลาด และการรันโปรแกรม เป็นต้น ซึ่งจะต้องสนับสนุนการทำงานได้ทั้ง Immediate Mode และ Script Mode เช่น โปรแกรม MATLAB และ Simulink ซึ่งมีคุณลักษณะพิเศษต่างๆ มากมาย ได้แก่ เป็นภาษาระดับสูงเพื่อการคำนวณเชิงตัวเลขด้านวิศวกรรมศาสตร์และวิทยาศาสตร์ มีความสามารถด้านการจัดการกับเมทริกซ์และอาร์เรย์ พล็อตและกราฟิก 2 มิติและ 3 มิติ, พีชคณิตสมการเชิงเส้น, พีชคณิตสมการฟังก์ชันที่ไม่ใช่เชิงเส้น, สถิติ, การวิเคราะห์ข้อมูล, แคลคูลัสและสมการเชิงอนุพันธ์, การคำนวณตัวเลข และฟังก์ชันพิเศษอื่นๆ เป็นต้น ซึ่งจะได้แนะนำโดยละเอียดในบทถัดไป



แบบฝึกหัดท้ายบท



1. จงอธิบายความหมายของคำว่า “โปรแกรมคอมพิวเตอร์” มีความหมายเหมือนกันหรือแตกต่างกันอย่างไรกับคำว่า “ซอฟต์แวร์”
2. จงให้เหตุผลว่า เหตุใดโปรแกรมคอมพิวเตอร์จึงมีความสำคัญและมีความจำเป็นในปัจจุบัน
3. จงยกตัวอย่างภาษาคอมพิวเตอร์ที่นอกเหนือจากที่ได้ยกตัวอย่างไปแล้วในบทเรียน
4. การทำงานแบบ Immediate Mode และ Script Mode มีความแตกต่างกันอย่างไร อธิบาย
5. คุณลักษณะของการโปรแกรมมิ่งแบบ OOP มีลักษณะอย่างไร และโปรแกรม MATLAB สามารถเขียนโปรแกรมแบบ OOP ได้หรือไม่

• • •



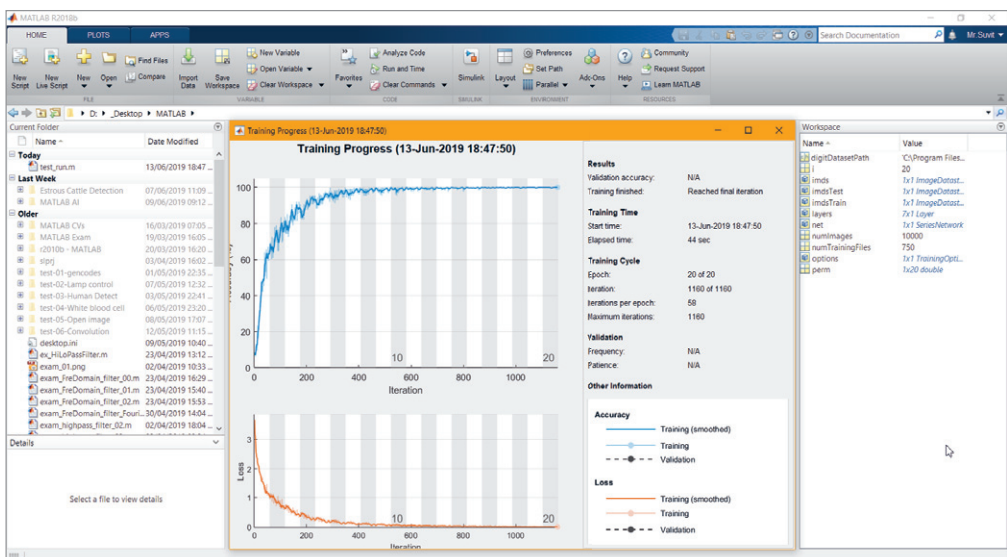
2

บทที่

แนะนำโปรแกรม MATLAB และ MATLAB Simulink

MATLAB® ย่อมาจาก Matrix Laboratory เป็นโปรแกรมคอมพิวเตอร์สมรรถนะสูง ที่ถูกสร้างขึ้นมาเพื่อช่วยในงานคำนวณด้านวิศวกรรมศาสตร์และวิทยาศาสตร์ ประกอบด้วยฟังก์ชันสำเร็จรูปต่างๆ ซึ่งครอบคลุมการคำนวณพื้นฐานทางด้านวิศวกรรมศาสตร์และวิทยาศาสตร์เกือบทั้งหมด ดังนั้นผู้เขียนโปรแกรมจึงไม่มีความจำเป็นต้องเขียนฟังก์ชันและทดสอบการทำงานด้วยตัวเองให้ยุ่งยาก ทำให้ผู้ใช้งานสามารถพัฒนาโปรแกรมได้ง่ายและมีประสิทธิภาพมากยิ่งขึ้น การออกแบบและพัฒนาโปรแกรมสามารถทำได้ง่ายกว่าภาษาอื่นๆ (เช่น ภาษา Java, Fortran และภาษา C เป็นต้น) จึงทำให้ได้รับความนิยมใช้งานกันอย่างแพร่หลายทั่วโลก รูปร่างหน้าตาโปรแกรม MATLAB ดังแสดงในรูปที่ 2.1

MATLAB Desktop



รูปที่ 2.1 ลักษณะหน้าต่างทำงานโปรแกรม MATLAB เวอร์ชัน r2018b

2.1 คุณสมบัติพิเศษของโปรแกรม MATLAB

MATLAB เป็นภาษาระดับสูงในการคำนวณเชิงตัวเลข ถูกสร้างขึ้นเพื่อช่วยในการคำนวณด้านวิศวกรรมศาสตร์และวิทยาศาสตร์ ได้แก่ การจัดการกับเมทริกซ์และอาร์เรย์, พล็อตและกราฟิก 2 มิติและ 3 มิติ, พีชคณิตสมการเชิงเส้น, พีชคณิตสมการฟังก์ชันที่ไม่ใช่เชิงเส้น, สถิติ, การวิเคราะห์ข้อมูล, แคลคูลัสและสมการเชิงอนุพันธ์, การคำนวณตัวเลข และฟังก์ชันพิเศษอื่นๆ อีกมากมาย ถูกนำไปประยุกต์ใช้งานในหลายๆ กลุ่มงาน เช่น การประมวลผลสัญญาณ (Signal Processing), การสื่อสาร (Communication), การประมวลผลภาพและวิดีโอ (Image and Video Processing), ระบบควบคุม (Control System), การวัดและควบคุม (Instruments and Control) การคำนวณทางเศรษฐศาสตร์ (Economic) และการคำนวณทางชีววิทยา (Biology) เป็นต้น คุณสมบัติพิเศษของโปรแกรม MATLAB มีรายละเอียดแสดงพอสังเขปดังนี้

1. ง่ายต่อการใช้งาน โปรแกรม MATLAB เป็นภาษาที่ใช้งานได้ง่ายคือ สามารถโปรแกรมได้ทั้งบน Command Line และบนหน้าต่างทำงาน Editor พร้อมด้วยฟังก์ชันสำเร็จรูปมีให้เกือบทุกแขนง ทั้งทางด้านวิศวกรรมศาสตร์และวิทยาศาสตร์ ดังนั้นจึงเป็นที่นิยมใช้ในหมู่นักศึกษานักวิชาการ นักวิจัย และครู/อาจารย์ อีกทั้งในสถาบันการศึกษาระดับอุดมศึกษา ได้นำโปรแกรม MATLAB เป็นวิชาหนึ่งในการเรียนการสอน รวมทั้งยังมีเครื่องมืออื่นๆ ที่ช่วยอำนวยความสะดวกในการเขียนโปรแกรมอีกมากมาย ได้แก่ Editor, Debugger, Online Documents, Manuals และ Demo Program เป็นต้น

2. รองรับหลายระบบปฏิบัติการ โปรแกรม MATLAB ถูกออกแบบมาให้สามารถทำงานบนระบบปฏิบัติการได้หลากหลาย ได้แก่ Windows (XP, Vista, 7, 8 และ 10), Linux, Unix และ Macintosh โปรแกรมที่เขียนและพัฒนาขึ้นด้วย MATLAB ไม่ว่าจะเขียนบนระบบปฏิบัติการอะไร เราจะสามารถนำโค้ดคำสั่งไปรันบนระบบปฏิบัติการอื่นได้ทั้งนั้น

3. ฟังก์ชันพิเศษ (Pre-defined Function) มีฟังก์ชันให้เลือกใช้งานได้มากมายและครอบคลุมหลายสาขา การเขียนโปรแกรมทำได้ง่ายและสะดวกขึ้น เช่น ถ้าหากผู้ใช้งานต้องการเขียนโปรแกรมที่ต้องคำนวณด้านสถิติ โปรแกรม MATLAB จะมีฟังก์ชันด้านนี้มากมายหลายฟังก์ชันให้ผู้พัฒนาได้เลือกใช้งาน

4. การแสดงผลข้อมูล MATLAB สามารถแสดงผลข้อมูลได้หลายรูปแบบ ได้แก่ แบบตัวเลข แบบรูปภาพ และแบบกราฟทั้ง 2 มิติ และ 3 มิติ ซึ่งจะไม่เหมือนกับโปรแกรมอื่น ทำให้ผู้เขียนโปรแกรมสามารถแสดงผลข้อมูลที่มีความหลากหลายง่ายขึ้น และเลือกได้ตรงกับความต้องการของผู้ใช้งาน

5. ส่วนติดต่อกับผู้ใช้งาน (Graphical User Interface; GUI) การใช้งานโปรแกรมที่เราเองเป็นผู้ออกแบบและพัฒนาขึ้น อาจจะไม่ใช่ว่าเรื่องยาก แต่ถ้าเราต้องเขียนโปรแกรม เพื่อให้ผู้อื่นได้ใช้งานในรูปแบบที่ต้องป้อนข้อมูลด้วยโค้ดคำสั่งล้วนๆ อาจจะไม่เป็นวิธีการที่ดีนักสำหรับผู้ใช้งานทั่วไป ฉะนั้น MATLAB จึงได้ออกแบบส่วนติดต่อกับผู้ใช้งานแบบ GUI เพื่อตอบสนองและบรรเทาปัญหาเหล่านี้ จะทำให้ผู้ใช้งานโปรแกรมของเรา สามารถป้อนข้อมูล หรือควบคุมค่าพารามิเตอร์ต่างๆ ส่วนของข้อมูลนำเข้า (Input) อย่างเดียว แล้วให้โปรแกรมทำการประมวลผล และแสดงผลข้อมูล (Output) เฉพาะส่วนที่ต้องการได้

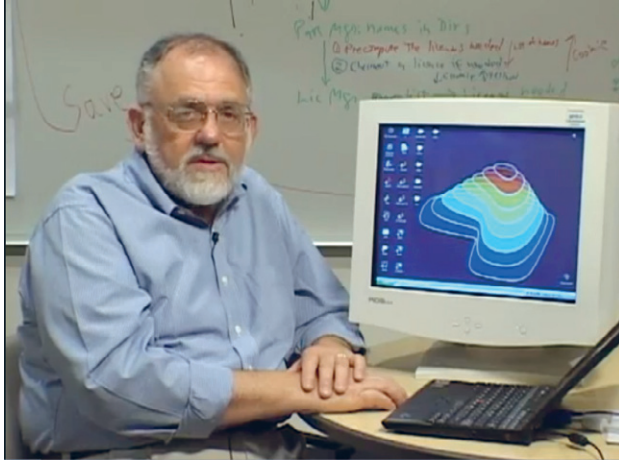
2.2 ลักษณะการโปรแกรมมิ่งและการทำงานของโปรแกรม MATLAB

การทำงานของโปรแกรม MATLAB จะออกแบบให้ทำงานได้ในลักษณะแบบประมวลผลทันที (Interactive) คือ การเขียนคำสั่งทีละคำสั่ง และให้ MATLAB ประมวลผลทีละคำสั่ง (Interpreter) หรือจะออกแบบรวบรวมชุดคำสั่งรวมเป็นโปรแกรมเดียวแล้วประมวลผลในครั้งเดียว (Compiler) ก็ได้

ข้อสำคัญอย่างหนึ่งของ MATLAB คือข้อมูลทุกตัวจะถูกเก็บอยู่ในลักษณะของอาร์เรย์หรือเมทริกซ์ กล่าวคือ แต่ละตัวแปรจะถูกแบ่งออกเป็นส่วยย่อยระดับ Element ซึ่งในการใช้งานตัวแปรแบบอาร์เรย์ของโปรแกรม MATLAB ไม่ต้องทำการจองหน่วยความจำ ไม่ต้องประกาศชนิดตัวแปร และกำหนดขนาดของข้อมูล (Dimension) เหมือนกับการเขียนโปรแกรมภาษาระดับสูงอื่นๆ ซึ่งจะทำให้สามารถแก้ปัญหาหรือจัดการกับตัวแปรที่อยู่ในรูปของเมทริกซ์และเวกเตอร์ได้ง่ายขึ้นมาก ช่วยลดเวลาและความยุ่งยากในการเขียนโปรแกรมลงไปได้มาก เมื่อเทียบกับการเขียนโปรแกรมโดยใช้ภาษาสูงอื่นๆ อย่างเช่น ภาษา C หรือ Fortran เป็นต้น

2.3 จุดเริ่มต้นของโปรแกรม MATLAB

โปรแกรม MATLAB ได้เริ่มต้นออกแบบและพัฒนาขึ้น เพื่อต้องการให้โปรแกรมสามารถคำนวณและแก้ปัญหาตัวแปรแบบเมทริกซ์ให้มีความสะดวกและง่ายขึ้น เริ่มพัฒนาครั้งแรกโดย ดร. คลีฟ โมเลอร์ (Cleve Moler) ซึ่งพัฒนาโปรแกรมขึ้นด้วยภาษา Fortran ภายใต้โครงการ LINPACK และ EISPACK



รูปที่ 2.2 คลีฟ โมเลอร์ (Cleve Moler)

ที่มา : (About MathWorks)

<https://www.mathworks.com/company/aboutus/founders/clevemoler.html>

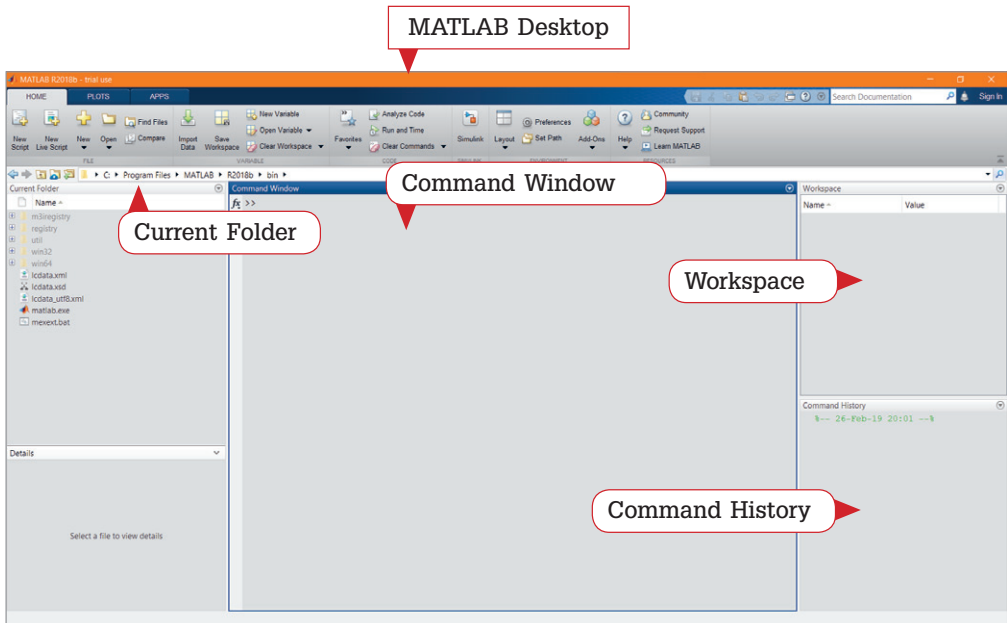
ปี ค.ศ. 1970 คลีฟ โมเลอร์ ได้พัฒนาซอฟต์แวร์สำหรับติดต่อกับไลบรารี Linpack ซึ่งเป็นไลบรารีสำเร็จรูปเพื่อคำนวณด้านพีชคณิต และ Eispack ซึ่งเป็นไลบรารีสำหรับการคำนวณด้านเมทริกซ์ โดยได้จัดวางตัวแปรต่างๆ ให้เก็บค่าข้อมูลอยู่ในรูปเมทริกซ์โดยอัตโนมัติ สำหรับใช้ประกอบการเรียนการสอนวิชาคณิตศาสตร์ ไลบรารีและฟังก์ชัน Linpack และ Eispack (Open source Library) เป็นฟังก์ชันสำเร็จรูปที่สามารถเรียกใช้ผ่านภาษาคอมพิวเตอร์ทั่วๆ ไปได้ เช่น ฟังก์ชันผลลัพธ์การอินเวอร์สเมทริกซ์ทุกขนาด ฟังก์ชันหาค่ารากของสมการ (n -root) และฟังก์ชันมาตรฐานอื่นอีกมากมาย สมัยก่อนนิยมเรียกใช้ไลบรารีเหล่านี้ ด้วยโปรแกรมที่เขียนจากภาษา C และภาษา Fortran

ต่อมาในปี ค.ศ. 1984 คลีฟ โมเลอร์ และลูกศิษย์ แจ็ค ลิตเทิล (Jack Little) ได้ต่อยอดงาน โดยการพัฒนาโปรแกรม MATLAB ขึ้นมาเพื่อการค้า โดยใช้ชื่อบริษัท MathWorks (www.mathworks.com) และ MATLAB ได้พัฒนาออกมาเป็นเวอร์ชันต่างๆ โดยใช้ภาษา C สำหรับเรียกใช้ไลบรารีฟังก์ชันขั้นสูงของตัวมันเอง ได้แก่ Lapack และ Arpack พร้อมทั้ง Toolboxes (ไลบรารีและฟังก์ชัน MATLAB) ที่ครอบคลุมงานหลายแขนง ได้กำหนดและใช้ “.m-file” เป็นไฟล์นามสกุลสำหรับโปรแกรมภาษา MATLAB โดยเฉพาะ ต่อมาได้พัฒนาโปรแกรม “Simulink” เพิ่มเติมขึ้นมาอีก เพื่อใช้สร้างแบบจำลองทางคณิตศาสตร์ ทำให้ตอบสนองความต้องการในแวดวงการศึกษาและภาคอุตสาหกรรม ซึ่งนำไปสู่การพัฒนานวัตกรรมสิ่งประดิษฐ์ใหม่ๆ แห่งโลกเทคโนโลยีทางวิทยาศาสตร์ และวิศวกรรมศาสตร์

2.4 ส่วนประกอบสำคัญของ MATLAB

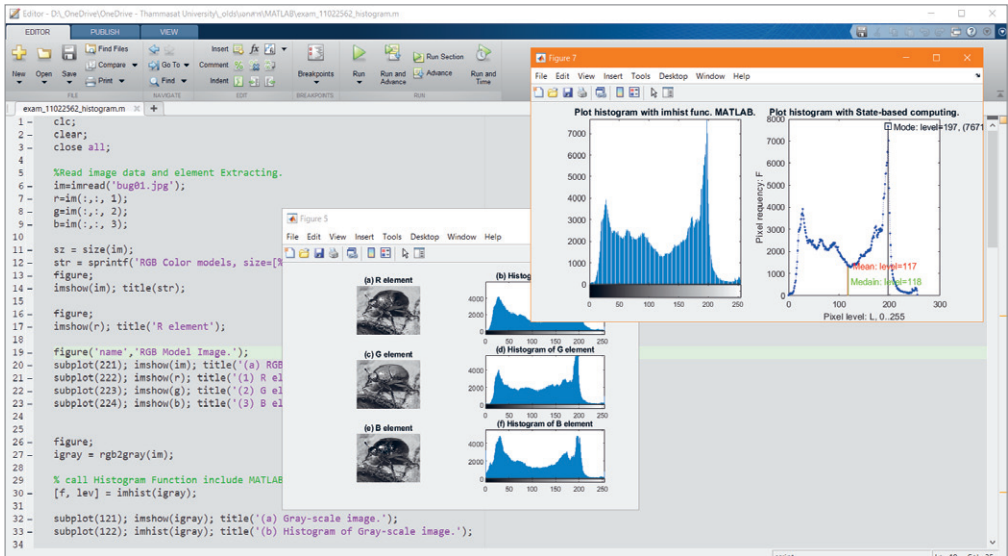
MATLAB ออกแบบมาเพื่อสนับสนุนการทำงานของผู้ใช้งานประกอบด้วย 5 ส่วนหลักๆ เพื่อช่วยในการวิเคราะห์ข้อมูล แสดงผลข้อมูล และเชื่อมต่อกับอุปกรณ์ภายนอกต่างๆ ได้อย่างรวดเร็ว และมีประสิทธิภาพ มีส่วนประกอบของโปรแกรมสำคัญๆ ดังแสดงในรูปที่ 2.3 ดังนี้

1. หน้าต่างหลักและหน้าต่างรองของ MATLAB หรือเรียกอีกอย่างหนึ่งว่า “MATLAB Desktop Environment” ช่วยให้ผู้ใช้งานทำงานได้ง่ายและรวดเร็วมากขึ้น หน้าต่างทำงานในส่วนนี้ จะประกอบด้วยชุดเครื่องมือที่ช่วยให้เราสามารถใช้งานฟังก์ชันและไฟล์ต่างๆ ด้วยเครื่องมือแบบไอคอนรูปภาพ (Graphic User Interface: GUI) ประกอบด้วยหน้าต่างย่อย ได้แก่ หน้าต่าง Current Folder, Command Window, Command History และหน้าต่าง Workspace แสดงดังรูปที่ 2.3



รูปที่ 2.3 MATLAB Desktop Environment ประกอบด้วยหน้าต่างหลักและหน้าต่างย่อย

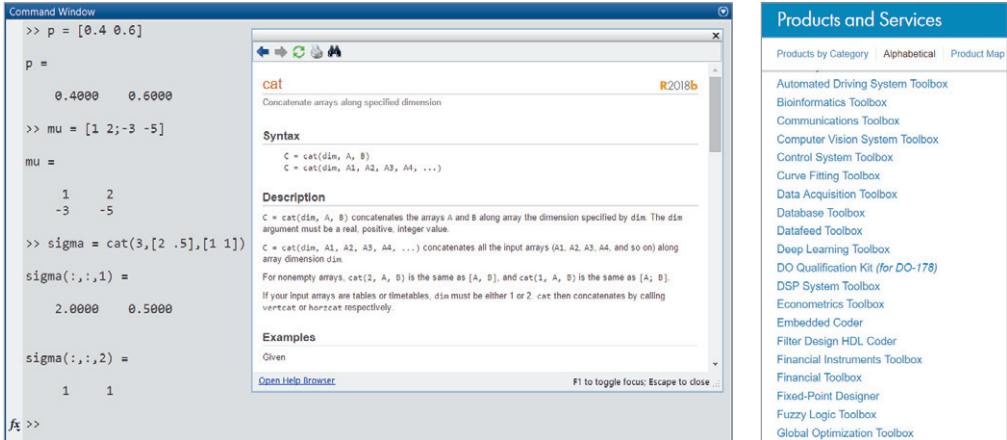
2. ภาษาของโปรแกรม MATLAB โปรแกรม MATLAB สามารถเขียนเป็นแอปพลิเคชันเพื่อให้รันบนระบบปฏิบัติการได้ หรือจะสร้างเป็นฟังก์ชันไวยากรณ์โดยเฉพาะได้ โดยจะใช้เครื่องมือชื่อว่า “MATLAB Editor” ซึ่งจะเขียนโค้ดคำสั่งอยู่ในรูปฟังก์ชันนามสกุล “.m-file” พร้อมทั้งเครื่องมือสำหรับตรวจสอบข้อผิดพลาดของโค้ดคำสั่ง โปรแกรม MATLAB ตั้งแต่เวอร์ชัน 2013a เป็นต้นมา จะมีแอปพลิเคชันเฉพาะด้านต่างๆ ให้ดาวน์โหลดเพิ่มเติม (Add-On) ได้ที่เว็บไซต์ <http://www.mathworks.com>



รูปที่ 2.4 หน้าต่างทำงาน “MATLAB Editor” สำหรับเขียนโค้ดคำสั่ง

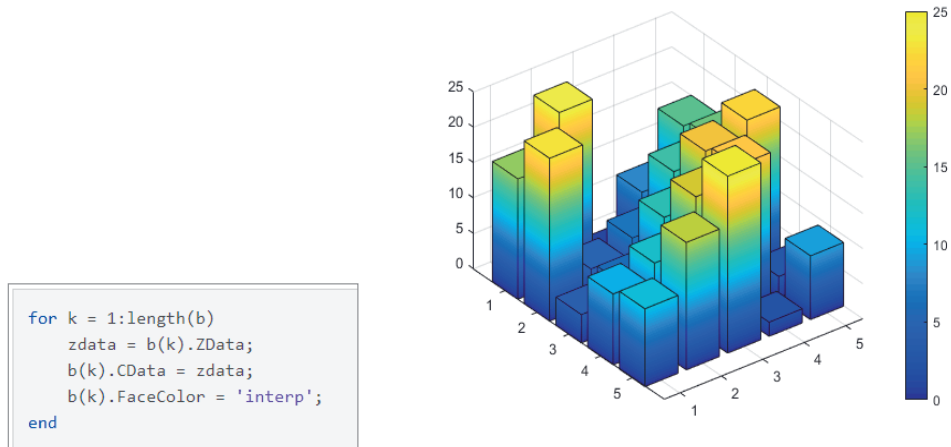
3. โลบารรีฟังก์ชัน และบล็อกไดอะแกรม MATLAB ได้ออกแบบและสร้างฟังก์ชันสำหรับใช้งานให้ครอบคลุมหลายสาขางาน โดยรวบรวมฟังก์ชันต่างๆ ซึ่งอยู่ในรูปของ **.m-file** หรือ **“Blocksets”** ต่างๆ บน MATLAB Simulink ทำให้เป็นไฟล์ย่อยๆ ซึ่งแต่ละไฟล์จะถูกสร้างขึ้นเพื่อใช้สำหรับการคำนวณและหาผลลัพธ์อย่างใดอย่างหนึ่ง หรือแก้ปัญหอย่างใดอย่างหนึ่ง เรียกว่า **“อัลกอริทึม”** (Algorithms)

เริ่มจากฟังก์ชันพื้นฐานง่ายๆ เช่น การบวก/ลบ ฟังก์ชันตรีโกณมิติพื้นฐาน sine, cos, tan และ log เป็นต้น จนถึงฟังก์ชันที่มีความซับซ้อน มีขั้นตอนวิธีในการคำนวณที่ยุ่งยาก เช่น การหาอินเวอร์สเมทริกซ์ การหาค่าสมการเชิงซ้อน การหาทูลไคลด์ส เป็นต้น รวมทั้งฟังก์ชันเฉพาะทางต่างๆ เรียกว่า **“ทูลบ็อกซ์”** (Toolboxes) ได้แก่ Control System, Bioinformatics, Signal Processing, Fuzzy Logic, Aerospace, Image Processing และ Econometrics เป็นต้น แสดงการเรียกใช้งานฟังก์ชันตัวอย่าง ดังแสดงในรูปที่ 2.5



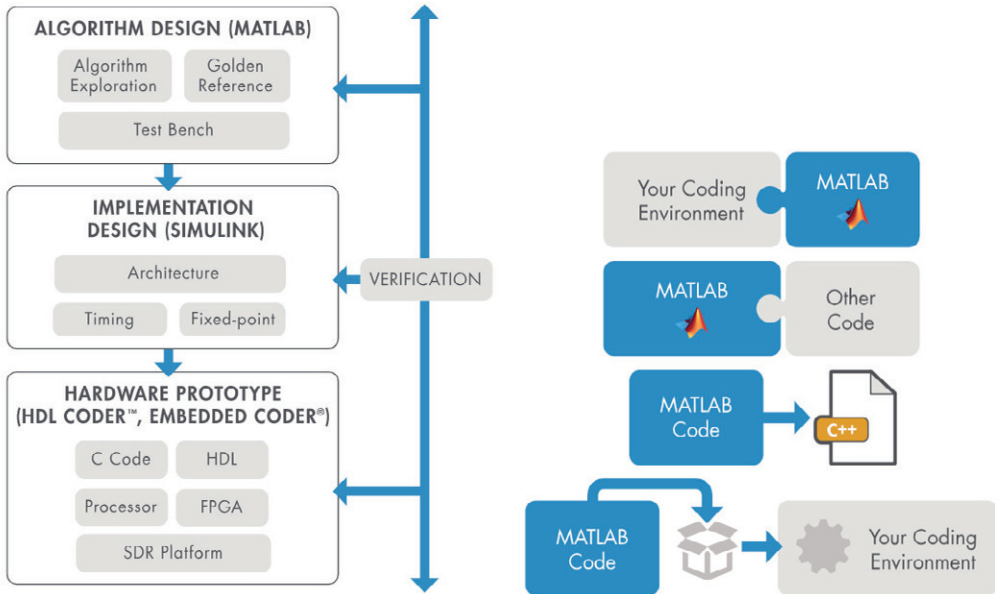
รูปที่ 2.5 ฟังก์ชันการใช้งาน และหมวดหมู่ของฟังก์ชันที่เรียกว่า “Toolboxes”

4. Handle Graphics เป็นไลบรารีฟังก์ชันของ MATLAB ใช้สำหรับแสดงผลพล็อตหรือข้อมูลต่างๆ อยู่ในรูปของตัวเลข ตัวอักษร กราฟิก รูปภาพ เสียง วิดีโอ พร้อมด้วยฟังก์ชันไลบรารี Guide เป็นเครื่องมือใช้สำหรับสร้างแอปพลิเคชันแบบ GUI (Graphic User Interface) ทำให้สามารถนำขั้นตอนวิธีหรืออัลกอริทึมที่ออกแบบไว้ไปพัฒนาเป็นแอปพลิเคชันให้ทำงานบนระบบปฏิบัติการต่างๆ ได้



รูปที่ 2.6 โค้ดคำสั่งสำหรับ Plot กราฟซึ่งเป็นเครื่องมือในการสร้างแอปพลิเคชัน GUI

5. MATLAB Application Program Interface (MATLAB API) เป็นส่วนติดต่อกับภาษาคอมพิวเตอร์อื่นๆ ได้แก่ ภาษา C++, C, Java, Fortran และ ภาษา Python หรือจากโปรแกรมภายนอกอื่น ที่เป็นไฟล์นามสกุลแบบต่างๆ หรือจะใช้ส่วนอินเตอร์เฟซนี้ทำการเชื่อมต่อเข้ากับฮาร์ดแวร์ภายนอกก็ได้

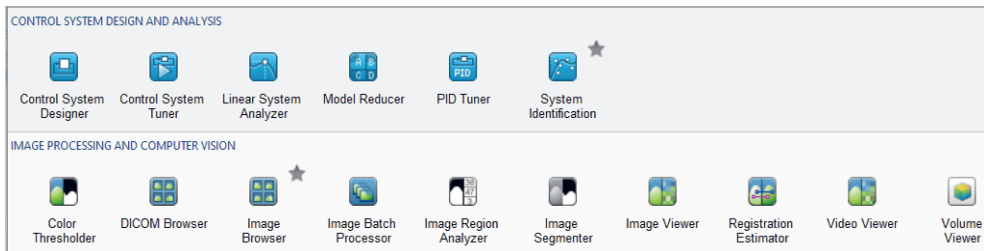


(บน) ฟังงานโครงสร้างการอินเตอร์เฟซของ MATLAB กับภาษาคอมพิวเตอร์ภาษาอื่นๆ

Language	Engine API	Matlab Supported Interfaces to Other Languages	MATLAB Compiler and MATLAB Compiler SDK	MATLAB Production Server
	Calling MATLAB from another language	Calling other languages or system components from MATLAB		
C - All versions prior to 11	✓	✓	✓	✓
C - Version 11	✓		✓	
C++ - All versions prior to 11	✓		✓	✓
C++ - Version 11	✓		✓	✓
C++ - Version 14	✓		✓	✓
Fortran - All versions	✓	✓		
Python 2.7	✓	✓	✓	✓
Python 3.5	✓	✓	✓	
Python 3.6	✓	✓	✓	
.COM components and apps - all versions			✓	
Java 8	✓	✓	✓	✓
.NET CLR version 2.0		✓		
.NET CLR version 4.0		✓	✓	✓
HTTP 1.1		✓		✓
Perl 5 26.1		✓		

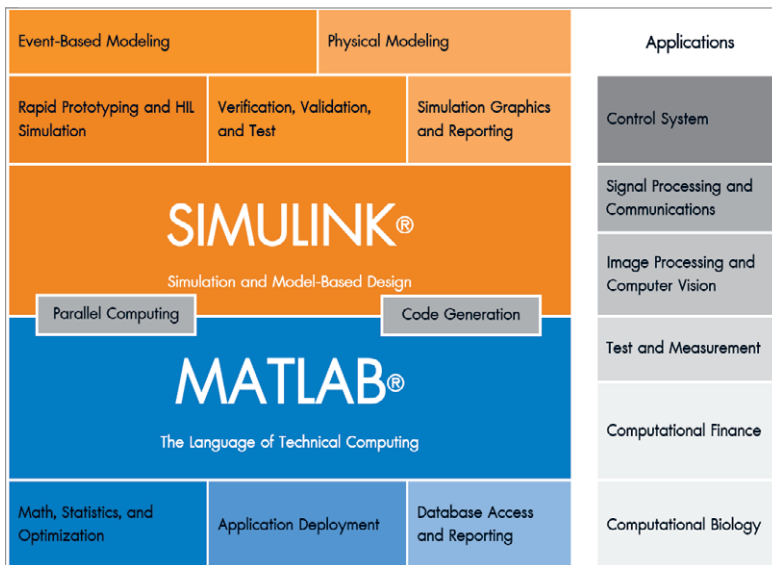
รูปที่ 2.7 (ล่าง) ภาษาคอมพิวเตอร์และแพลตฟอร์มที่โปรแกรม MATLAB สนับสนุนการอินเตอร์เฟซ

MATLAB มีทุลบ็อกซ์ (Toolboxes) และบล็อกเซต (Blocksets) มากมาย สำหรับการคำนวณเชิงเทคนิค (Technical Computing) โดยพัฒนาต่อยอดมาจาก Numerical Computing เพื่อให้ง่ายต่องานวิจัยและการสร้างนวัตกรรมเชิงวิศวกรรม เพราะเหตุนี้ MATLAB จึงสร้างเครื่องมือประเภท “One stop service” เพื่อลดเวลาและขั้นตอนวิธีที่ยุ่งยากสำหรับการสร้างงานวิจัยหรือนวัตกรรม โดย MATLAB มีไลบรารีตั้งแต่การออกแบบอัลกอริทึมจนถึงไลบรารีสำหรับสร้างผลิตภัณฑ์หรือนวัตกรรมโดยใช้ Toolboxes Supporting ได้แก่ Control System, PID Tuner, Image Segmenter, Color Thresholder เป็นต้น ดังแสดงในรูปที่ 2.8



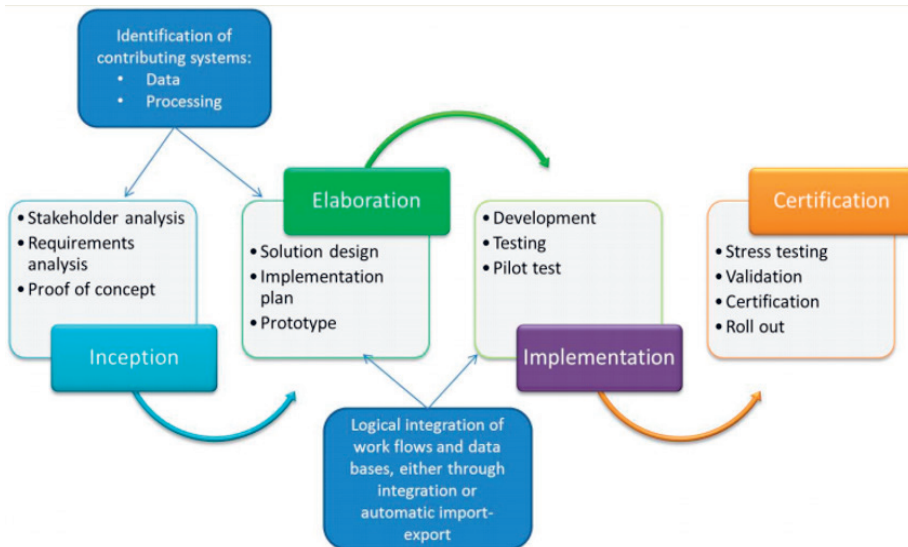
รูปที่ 2.8 Apps แบบ One stop service งานต่างๆ สำหรับงานวิจัยและนวัตกรรม

กระบวนการสร้างงานวิจัยหรือนวัตกรรมด้วย MATLAB ที่มาพร้อม Toolboxes ต่างๆ สนับสนุนการทำงานมากมาย สามารถจัดเป็นหมวดหมู่ตามลักษณะการใช้งานการคำนวณเชิงเทคนิค แสดงผังงานดังในรูปที่ 2.9



รูปที่ 2.9 ผังงานการคำนวณเชิงเทคนิคของโปรแกรม MATLAB

จากรูปที่ 2.9 คำว่า “The Language of Technical Computing” คือ โปรแกรมที่ใช้ภาษาคอมพิวเตอร์ ประกอบด้วยฟังก์ชันมากมายหลายสาขา เช่น สาขาวิศวกรรมไฟฟ้า วิศวกรรมคอมพิวเตอร์ และวิศวกรรมเครื่องกล เป็นต้น สำหรับงานวิจัยและการสร้างนวัตกรรมต้นแบบที่ต้องการทดสอบค่าตัวแปรต่างๆ ซ้ำกันหลายรอบ สามารถเปรียบเทียบกับกระบวนการพัฒนาโปรแกรมคอมพิวเตอร์และงานด้านวิศวกรรม แสดงผังงานดังในรูปที่ 2.10

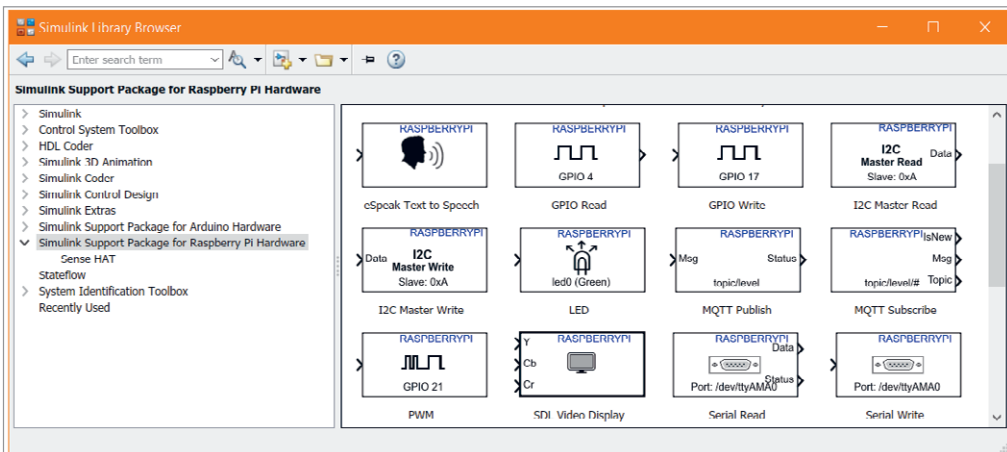
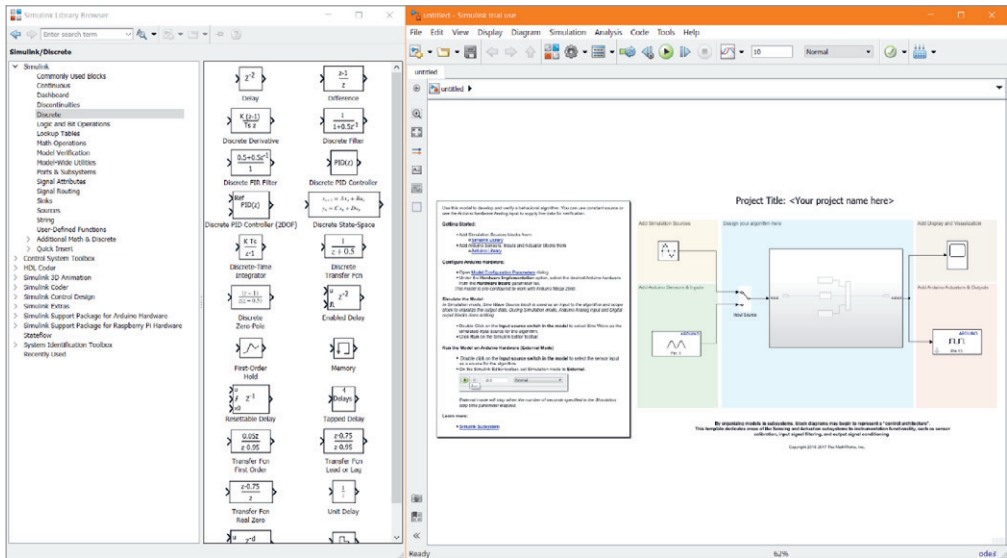


รูปที่ 2.10 ผังงานกระบวนการพัฒนางานโปรแกรมคอมพิวเตอร์และงานด้านวิศวกรรม

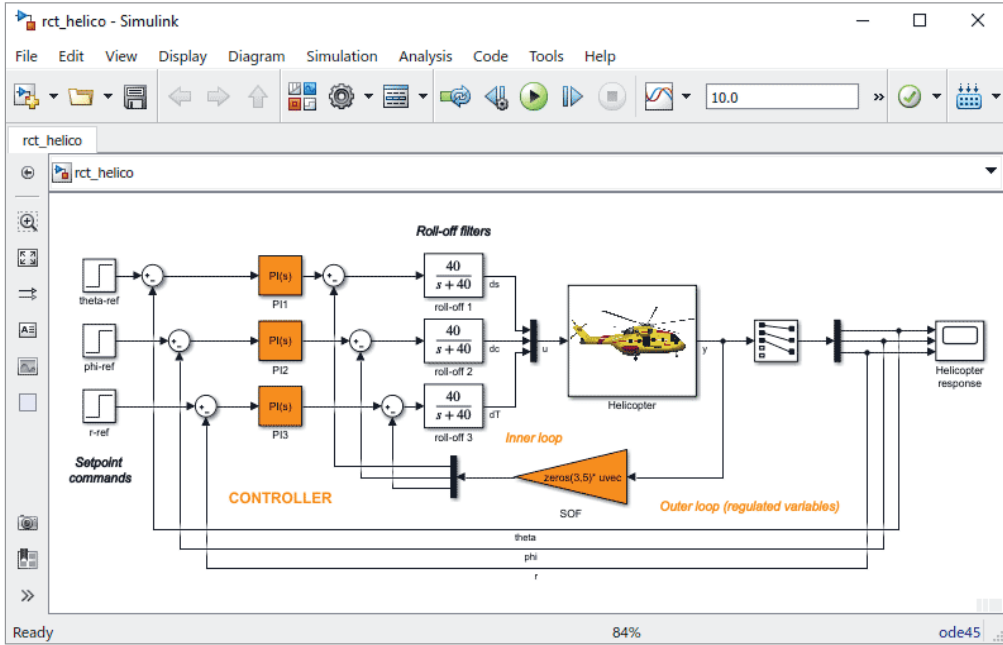
2.5 โปรแกรม MATLAB Simulink

MATLAB Simulink ออกแบบและพัฒนาโดยบริษัท “MathWorks” เป็นการเขียนโปรแกรมคอมพิวเตอร์โดยใช้รูปภาพหรือกราฟิก (Graphical Programming: GP) สำหรับการสร้างแบบจำลอง (Modeling) การจำลองการทำงาน (Simulating) และการวิเคราะห์ (Analyzing) ที่ใช้สำหรับระบบฝังตัวแบบไดนามิกและระบบพลวัตหลายโดเมน (Multi-domain Dynamical Systems) โครงสร้างของโปรแกรมหลักๆ จะอยู่ในรูปของบล็อกไดอะแกรม หรือรูปภาพกราฟิก แสดงดังรูปที่ 2.11 ซึ่งรวบรวมไว้เป็นชุดไลบรารี (Blockset Libraries) มากมายหลายสาขา เช่นเดียวกันกับ MATLAB Desktop ได้แก่ Control System, Bioinformatics, Signal Processing, Aerospace,

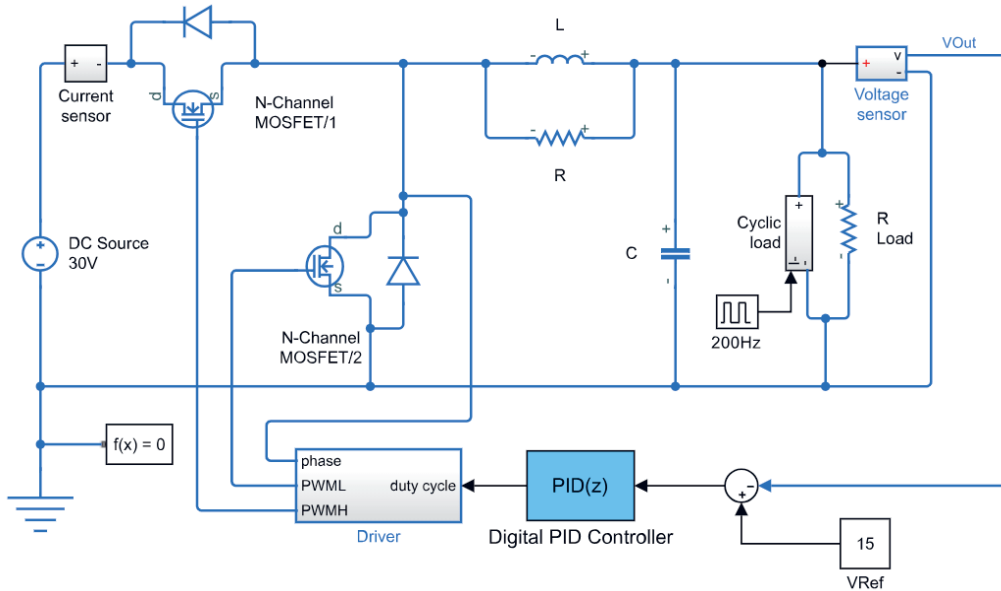
Image Processing, Econometrics และ Fuzzy Logic เป็นต้น โดยออกแบบให้สามารถทำงานร่วมกันกับโปรแกรม MATLAB ที่อยู่ในรูปของ .m-files ได้ คุณสมบัติที่กล่าวมาข้างต้น ทำให้ MATLAB Simulink ถูกนำไปใช้ในงานอย่างแพร่หลายในการออกแบบและสร้างแบบจำลองในระบบควบคุมอัตโนมัติ การประมวลผลสัญญาณดิจิทัล การประมวลผลภาพ เป็นต้น แสดงการนำโปรแกรม MATLAB Simulink ไปใช้งานเพื่อการจำลองระบบด้านต่างๆ ดังในรูปที่ 2.11 – 2.20



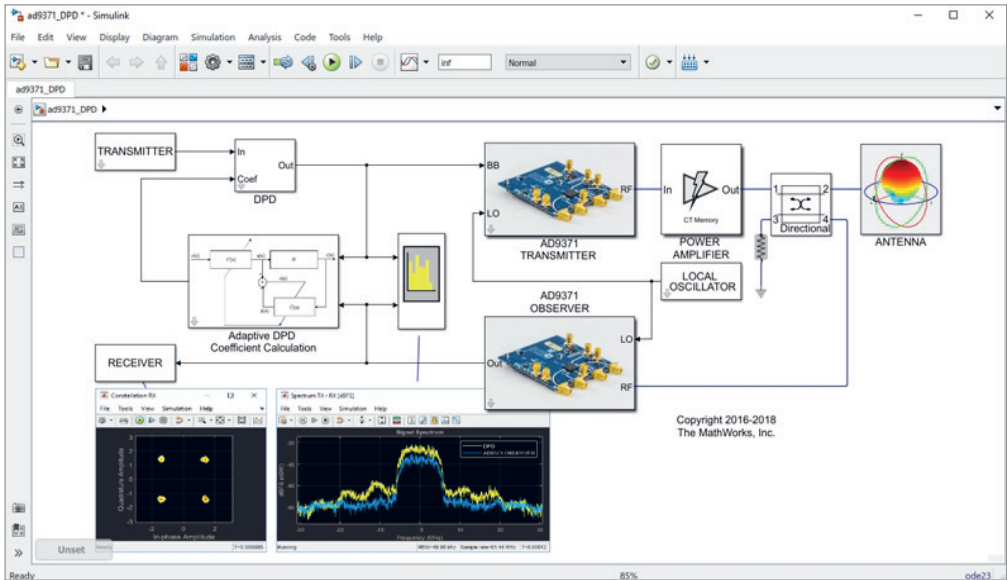
รูปที่ 2.11 โปรแกรม MATLAB Simulink ประกอบด้วยหน้าต่างทำงาน และชุดไลบรารี (Blockset Libraries) ต่างๆ รูปร่างลักษณะ บล็อกไดอะแกรมหรือไอคอนบล็อกเซต สำหรับเข้าถึงและจัดการบอร์ด Raspberry Pi



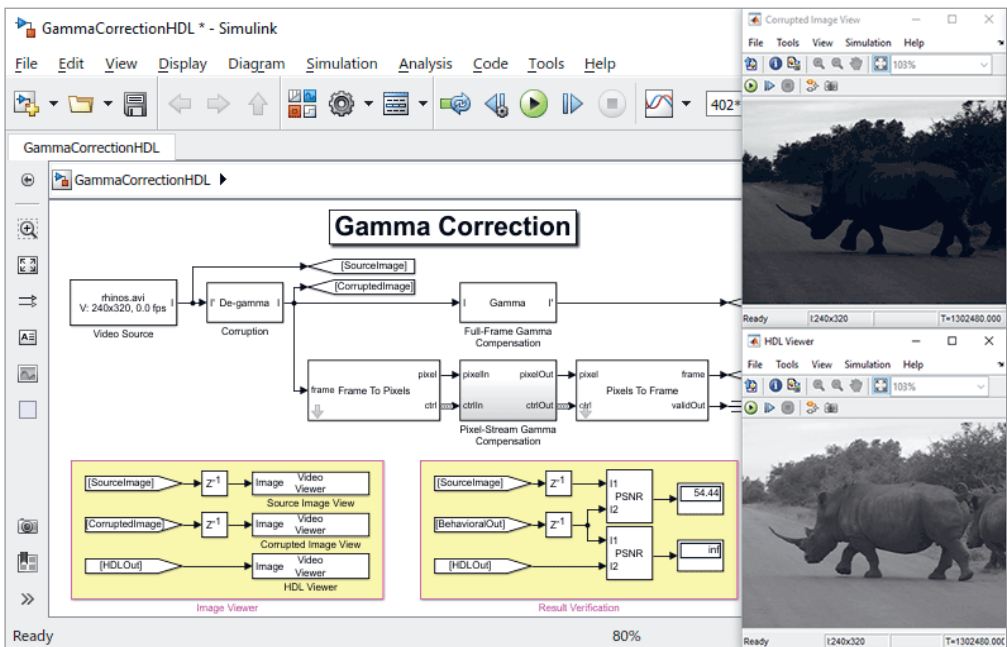
รูปที่ 2.12 MATLAB Simulink กับแบบจำลองการทำงานระบบ Control Systems



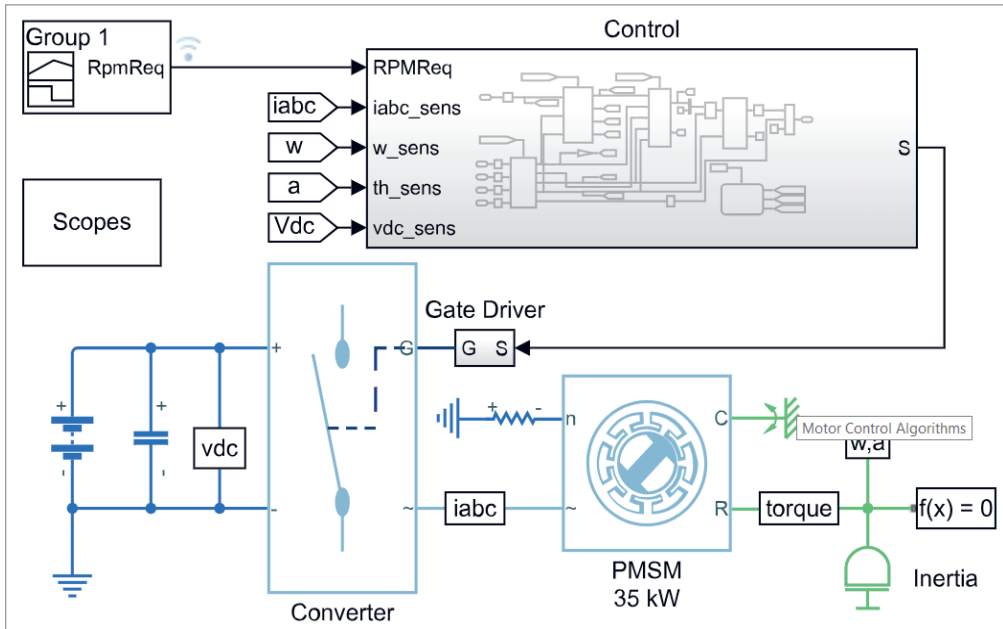
รูปที่ 2.13 MATLAB Simulink กับแบบจำลองด้าน Power Electronics Design; Design Digital Control for Power Converters



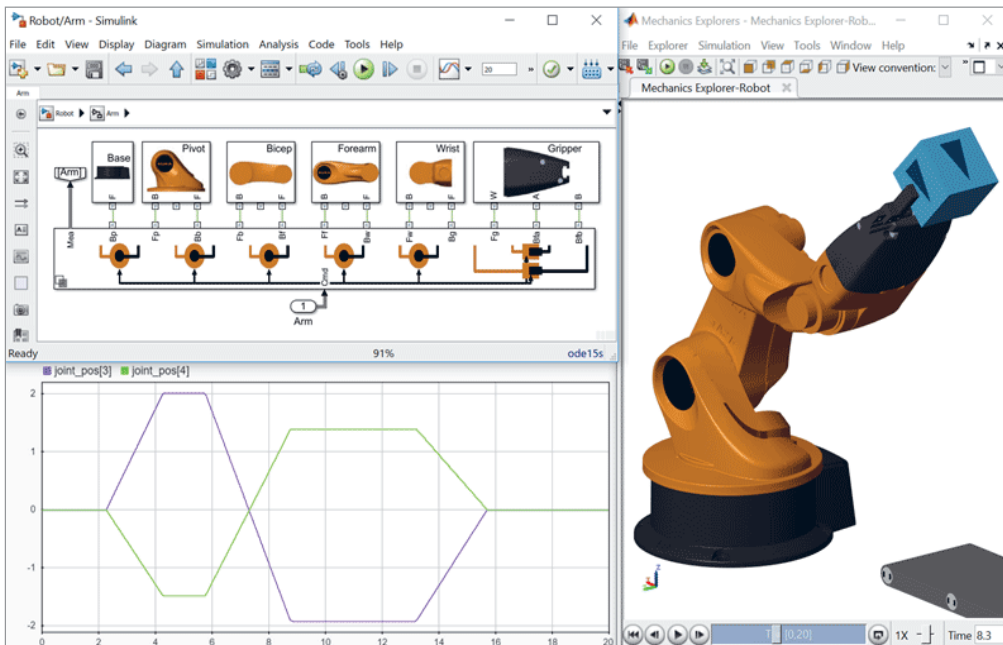
รูปที่ 2.14 MATLAB Simulink กับแบบจำลองด้าน Signal Processing; Wireless Communications



รูปที่ 2.15 MATLAB Simulink กับแบบจำลองด้าน Image Processing



รูปที่ 2.16 MATLAB Simulink กับแบบจำลองด้าน Power Electronics Control Design; Build and Tune Motor Control Algorithms



รูปที่ 2.17 MATLAB Simulink กับแบบจำลองด้าน Robotics

