การตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

โดย

นางสาวสุนิสา เรืองศรี รหัสนักศึกษา B4907251

นางสาวสุวารี ศรีอำพรรณ์ รหัสนักศึกษา B4910220

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม

หลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2545

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2552

โครงงาน การ ตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล
โดย 1. นางสาวสุนิสา เรืองศรี รหัสประจำตัว B4907251
2. นางสาวสุวารี ศรีอำพรรณ์ รหัสประจำตัว B4910220
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ร.อ. ดร. ประโยชน์ คำสวัสดิ์
สาขาวิชา วิศวกรรมโทรคมนาคม
ภาคการศึกษาที่ 3/2552

บทคัดย่อ

(Abstract)

การตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล เป็นโครงงานที่คณะผู้จัดทำไว้จัดทำขึ้นโดยนำความรู้จากการเรียนและประสบการณ์มาประยุกต์ใช้ โดยมุ่งเน้นที่จะนำเสนอระบบรักษาความปลอดภัยที่มีประสิทธิภาพใช้งานง่าย และราคาไม่สูง โดยอาศัยหลักการจับภาพของ กล้องเว็บแคม เพื่อบันทึกภาพเหตุการณ์ต่างๆที่เกิดขึ้น และประยุกต์การใช้เทคนิค ในการประมวลผลภาพดิจิตอล (Digital image processing) โดยทำการวิเคราะห์ภาพเพื่อค้นหาสิ่งของที่ถูกวางทิ้งไว้ และไม่มีการเคลื่อนไหวเป็นเวลานาน ซึ่งอาจเป็น กระเป๋า กล่องหรือวัตถุรูปทรงอื่นๆ ซึ่งสิ่งของดังกล่าวอาจจะมีวัตถุอันตรายซุกซ่อนอยู่ เมื่อตรวจพบสิ่งของถูกวางทิ้งไว้เป็นเวลานานตามที่โปรแกรมกำหนด ก็จะมีการแจ้งเตือนไปยังผู้ควบคุมระบบหรือพนักงานรักษาความปลอดภัย ให้ทำการตรวจสอบต่อไป

มหาวิทยาลัยเทคโนโลยีสุรนารี

กิตติกรรมประกาศ

(Acknowledgement)

การทำโครงงานเรื่อง “การตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล” ส่งผลให้คณะผู้จัดทำได้รับความรู้และประสบการณ์ในด้านต่างๆ มากมาย ไม่ว่าจะเป็นความรู้เกี่ยวกับลักษณะของการประมวลผลภาพดิจิตอล การใช้งานโปรแกรม MATLAB ขณะนี้โครงงานดังกล่าวพร้อมทั้งรายงานได้สำเร็จแล้ว ซึ่งโครงงานดังกล่าวนี้ได้รับความร่วมมือ คำปรึกษา ข้อแนะนำและการสนับสนุนจากบุคคลดังนี้

ผู้ช่วยศาสตราจารย์ ร.อ. ดร. ประโยชน์ คำสวัสดิ์ อาจารย์สาขา วิศวกรรมโทรคมนาคม ซึ่งท่านได้เป็นที่ปรึกษาโครงงานดังกล่าวนี้ ข้าพเจ้าคณะผู้จัดทำโครงงานทุกคน ใคร่ขอขอบพระคุณท่านอาจารย์เป็นอย่างยิ่งที่มีส่วนร่วมในการให้ข้อมูล คำแนะนำที่เป็นประโยชน์ต่อโครงงานนี้ และเป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับการใช้งานโปรแกรม MATLAB เป็นอย่างดีมาโดยตลอด ซึ่งข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

ขอบคุณและขอบใจ พี่ เพื่อน และน้อง สาขาวิศวกรรมโทรคมนาคมทุกคน ที่คอยถามไถ่ด้วยความห่วงใยถึงความก้าวหน้าของโครงงานและถามว่าเมื่อไหร่จะสำเร็จการศึกษา และขอขอบคุณพิเศษเพื่อนร่วมรุ่น TCE 14 ที่เอาใจช่วยตลอดมา รวมถึงผู้มีพระคุณทุกท่านที่มิได้เอ่ยนามไว้ ณ ที่นี้

สุดท้ายนี้ กราบขอบพระคุณบิดาและมารดา ที่ได้อบรมเลี้ยงดู สนับสนุน ให้โอกาสทางการศึกษา ให้คำปรึกษา เป็นกำลังใจ เป็นผู้ให้ทุกสิ่งทุกอย่างกับผู้พัฒนาโครงงาน

นางสาวสุนิสา เรืองศรี
นางสาวสุวารี ศรีอำพรรณ์

คณะผู้จัดทำรายงาน

สารบัญ

เรื่อง หน้า

บทคัดย่อ……ก
กิตติกรรมประกาศ……ข
สารบัญ……..ค
สารบัญรูป……..จ
สารบัญตาราง……ช
บทที่ 1 บทนำ
1.1 บทนำ……1
1.2 หลักการและเหตุผลความสำคัญของโครงงาน……1
1.3 วัตถุประสงค์ของโครงงาน……1
1.4 ขอบเขตการทำงาน……2
1.5 ประโยชน์ที่คาดว่าจะได้รับ……...2
1.6 ขั้นตอนการดำเนินงาน……..2
บทที่ 2 ทฤษฎีและหลักการ
2.1 บทนำ.....3
2.2 นิยามคำศัพท์พร้อมความหมาย……3
2.3 แนะนำสู่การประมวลผลภาพดิจิตอล.....4
2.4 ทฤษฎีการหาขอบภาพ (Edge detection).....6
2.5 รูปแบบการบีบอัดแฟ้มภาพแบบเจเพ็ก (JPEG)……..…….....13
2.6 ค่าสหสัมพันธ์ (Correlation) ……..…...14
2.7 ภาพถ่ายสีระดับเทา (Gray scale image)…….... ……..…….……..……..15
2.8 มิติสีต่าง ๆ (Color)……....16
บทที่ 3 การทดลอง
3.1 บทนำ……20
3.2 อุปกรณ์การทำงานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล……...……20
3.3 ขั้นตอนการทำงานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล……...22

สารบัญ (ต่อ)

เรื่อง หน้า

3.3.1 ขั้นตอนการรับข้อมูลภาพ……..22
3.3.2 การประมวลผลของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล……..... ขั้นตอน……..22
3.3.3 ขั้นตอนการทำงานของประมวลผลภาพ……22
3.4 การทดลองการหาขอบภาพ (Edge detection)……..……....…….……...…27
3.5 การทดลองในการตรวจจับวัตถุต้องสงสัยโดยการทำ Correation ……..29
3.6 Database ที่ใช้งานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล …...……32
3.7 แสดงการเก็บข้อมูลตัวอย่างงานที่ได้ของระบบการตรวจจับ ……..……33
3.8 การแสดง Data Flow Diagram (DFD) การทำงานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล…….……37
บทที่ 4 สรุปและข้อเสนอแนะ
4.1 บทนำ.....41
4.2 สรุปผลการทดลองการหาขอบภาพ……..……41
4.3 สรุปผลการทดลองในการตรวจจับวัตถุต้องสงสัยโดยการทำ Correlation…...…….41
4.4 ปัญหาและอุปสรรค……...42
4.5 ข้อจำกัดของระบบ……42
4.6 ข้อเสนอแนะในการพัฒนาต่อไป……...42
4.7 รูปภาพการปฏิบัติงาน……....43
ประวัติผู้เขียน……..…....44
บรรณานุกรม……....45
ภาคผนวก ก โปรแกรม MATLAB ที่ใช้งาน……..……46
ภาคผนวก ข โปรแกรมคอมพิวเตอร์ที่ใช้งาน……..……..93

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป

รายการ หน้า

รูปที่ 2.1 ตัวอย่างภาพการประมวลผลภาพดิจิตอล……..4
รูปที่ 2.2 การเปลี่ยนจากค่าของสีแต่ละ pixel เป็นเลขฐานสอง……..…6
รูปที่ 2.3 ตัวอย่างของเทมเพลตด้วยวิธีโซเบล……..……6
รูปที่ 2.4 กราฟการหาขอบด้วยวิธี Gradient method และ Laplacian method……...10
รูปที่ 2.5 ตัวอย่างการหาขอบภาพโดยใช้ Edge detection แบบต่างๆ……11
รูปที่ 2.6 ภาพขั้นตอนของ Canny edge detection……..11
รูปที่ 2.7 แผนภาพการกระจายของภาพดาวเทียม LANDSAT-TM แสดงจำนวนนับของจุดภาพที่มีค่าร่วมกัน ของ 2 แบนด์ ทั้งสองแกนเป็นค่าตัวเลขของจุดภาพของแบนด์นั้นๆ……...14
รูปที่ 2.8 การทำ Correlation……..15
รูปที่ 2.9 การเปลี่ยนจากภาพสี RGB ที่ความเข้มสีเท่ากันเป็นภาพระดับเทา……..……..16
รูปที่ 2.10 ผลกระทบของสีที่มีสีของภาพพื้นหลังที่แตกต่างกัน
(ก) ภาพพื้นหลังสีดำ
(ข) ภาพพื้นหลังสีขาว……16
รูปที่ 2.11 แบบจำลองมิติสีแบบต่าง ๆ……..17
รูปที่ 2.12 แบบจำลองมิติสี HSV……..18
รูปที่ 3.1 ตัวอย่างภาพก่อนนำไปประมวลผล……..……23
รูปที่ 3.2 ตัวอย่างภาพเป้าหมาย……23
รูปที่ 3.3 ตัวอย่างภาพต้นฉบับที่ผ่านการทำ Gray scale……..……..24
รูปที่ 3.4 ตัวอย่างภาพเป้าหมายที่ผ่านการทำ Gray scale……..……24
รูปที่ 3.5 ตัวอย่างภาพต้นฉบับที่ทำการหาขอบภาพ (edge detection)……...25
รูปที่ 3.6 ตัวอย่างภาพเป้าหมายที่ทำการหาขอบภาพ (edge detection)……..……25
รูปที่ 3.7 ตัวอย่างภาพที่นำมาเป็นตัวกรองเพื่อทำให้ตรงส่วนที่เป็นเป้าหมายยังเป็ภาพสี……..…...…26
รูปที่ 3.8 ภาพผลลัพธ์……..…26
รูปที่ 3.9 ตัวอย่างภาพต้นฉบับ……...27
รูปที่ 3.10 ตัวอย่างการหาขอบภาพแบบวิธี Sobel method……..……....27
รูปที่ 3.11 ตัวอย่างการหาขอบภาพแบบวิธี Prewitt method……..28
รูปที่ 3.12 ตัวอย่างการหาขอบภาพแบบวิธี Roberts method……...…….…28
รูปที่ 3.13 ตัวอย่างการหาขอบภาพแบบวิธี Laplacian of gaussian method……..……...……..28
รูปที่ 3.14 ตัวอย่างการหาขอบภาพแบบวิธี Canny method……...……..29
รูปที่ 3.15 ตัวอย่างภาพต้นฉบับ……...……29

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป(ต่อ)

รายการ หน้า

รูปที่ 3.16 ตัวอย่างภาพเป้าหมาย……...30
รูปที่ 3.17 ตัวอย่างภาพที่เปลี่ยนภาพระดับสีเทา (Gray scale)……..……..……30
รูปที่ 3.18 ตัวอย่างภาพที่ทำการหาขอบของภาพ (Edge detection)……..……..……....30
รูปที่ 3.19 ตัวอย่างภาพการหา Correlation ……...……...31
รูปที่ 3.20 ตัวอย่างภาพที่นำมาเป็นตัวกรองเพื่อทำให้ตรงส่วนที่เป็นเป้าหมายยังเป็นภาพสี…….....……..31
รูปที่ 3.21 ตัวอย่างภาพผลลัพธ์……..……31
รูปที่ 3.22 ตัวอย่าง Folder ที่ใช้เป็นฐานข้อมูลภาพของระบบ……32
รูปที่ 3.23 ตัวอย่างงานรูปแบบวัตถุทรงกลม……..…....33
รูปที่ 3.24 ตัวอย่างงานรูปแบบวัตถุกระป๋อง……..…34
รูปที่ 3.25 ตัวอย่างงานรูปแบบวัตถุกระเป๋า……..……35
รูปที่ 3.26 ตัวอย่างงานรูปแบบวัตถุกล่อง……..……..36
รูปที่ 3.27 ภาพแสดง Data Flow Diagram (DFD) การทำงานของประมวลผลภาพ…….…....37
รูปที่ 3.28 ภาพแสดง Data Flow Diagram (DFD) การประมวลผลของระบบการตรวจจับวัตถุต้อง
สงสัยด้วยการประมวลผลภาพดิจิตอล……..……..38
รูปที่ 4.1 ตัวอย่างภาพปฏิบัติงาน……..……43

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง

เรื่อง หน้า

ตารางที่ 3.1 แสดงการบันทึกการทดลองในการตรวจจับวัตถุรูปทรงเรขาคณิตแบบต่างๆ …39

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1
บทนำ

1.1 บทนำ

การตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล เป็นโครงงานที่คณะผู้จัดทำไว้จัดทำขึ้นโดยนำความรู้จากการเรียนและประสบการณ์มาประยุกต์ใช้ โดยมุ่งเน้นที่จะนำเสนอระบบรักษาความปลอดภัยที่มีประสิทธิภาพใช้งานง่าย และราคาไม่สูง โดยอาศัยหลักการจับภาพของเว็บแคม เพื่อบันทึกภาพเหตุการณ์ต่างๆที่เกิดขึ้น และประยุกต์การใช้เทคนิค ในการประมวลผลภาพดิจิตอล (Digital image processing) โดยทำการวิเคราะห์ภาพเพื่อค้นหา สิ่งของที่ถูกวางทิ้งไว้ และไม่มีการเคลื่อนไหวเป็นเวลานาน ซึ่งอาจเป็น กระเป๋า กล่องหรือวัตถุรูปทรงอื่นๆ ซึ่งสิ่งของดังกล่าวอาจจะมีวัตถุอันตรายซุกซ่อนอยู่ เมื่อตรวจพบสิ่งของถูกวางทิ้งไว้เป็นเวลานานตามที่โปรแกรมกำหนด ก็จะมีการแจ้งเตือนไปยังผู้ควบคุมระบบหรือพนักงานรักษาความปลอดภัย ให้ทำการตรวจสอบต่อไป

1.2 หลักการและเหตุผลความสำคัญของโครงงาน

เนื่องมาจากเหตุการณ์การก่อการร้ายที่เกิดขึ้นในประเทศต่างๆทั่วโลกในปัจจุบัน สังเกตได้ว่าการก่อการร้ายส่วนมากมักจะเป็นการวางระเบิด โดยจะเป็นการนำระเบิดซ่อนไว้ในวัตถุต่างๆ เช่น กล่อง กระเป๋า ถุงดำ ฯลฯ แล้วนำวัตถุนั้นมาวางในพื้นที่เป้าหมาย ซึ่งมักจะเป็นบริเวณที่มีผู้คนพลุกพล่าน ซึ่งถ้าหากระเบิดนั้นทำงานตามที่ผู้ก่อการร้ายได้วางแผนไว้ จะก่อให้เกิดความเสียหายต่อชีวิตและทรัพย์สินเป็นอย่างมาก ด้วยเหตุนี้เองผู้จัดทำโครงงานจึงตั้งใจที่จะจัดทำระบบรักษาความปลอดภัยต้นแบบ โดยใช้กล้องวงจรปิดเป็นอุปกรณ์หลักที่ใช้สังเกตการณ์ และติดตั้งในบริเวณซึ่งมีความเสี่ยงต่อการก่อการร้าย และทำการประยุกต์ภาพที่กล้องวงจรปิดบันทึกได้เข้ากับเทคนิคการประมวลผลภาพแบบดิจิตอล (Digital image processing) เพื่อทำการวิเคราะห์ภาพที่ได้ว่ามีวัตถุต้องสงสัยหรือไม่ ถ้าหากพบว่ามีวัตถุต้องสงสัย ระบบจะทำการแจ้งเตือนผู้ที่เกี่ยวข้องด้วยเสียงเตือน

1.3 วัตถุประสงค์ของโครงงาน

1. จัดทำระบบรักษาความปลอดภัยต้นแบบ ที่สามารถตรวจสอบบริเวณเป้าหมายได้ว่ามีวัตถุต้องสงสัยหรือไม่
2. จัดทำฐานข้อมูลที่สามารถบันทึกและเรียกดูภาพเหตุการณ์จากกล้องวงจรปิดย้อนหลังได้

3. จัดทำระบบการแจ้งเตือนผู้ที่เกี่ยวข้องเมื่อตรวจพบวัตถุต้องสงสัย

1.4 ขอบเขตการทำงาน

1. ระบบสามารถตรวจจับวัตถุ เพียงชิ้นเดียว ที่เพิ่มเข้ามาในบริเวณที่กำหนดได้ โดยวัตถุนั้นจะต้องไม่มีการเคลื่อนไหว
2. ระบบสามารถแจ้งเตือนผู้ที่เกี่ยวข้อง ผ่านเสียงเตือนได้
3. ระบบสามารถปรับค่าต่างๆได้โดยผู้ควบคุม เช่น ค่าเวลาที่เป็นมาตรฐานในการวัดว่าวัตถุที่ตรวจพบเป็นวัตถุต้องสงสัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. มีระบบรักษาความปลอดภัยต้นแบบที่ราคาไม่แพง
2. สามารถนำระบบตรวจจับวัตถุต้องสงสัยไปพัฒนาต่อเนื่องและนำไปใช้งานได้จริง
3. สามารถนำความรู้ที่ได้มาใช้ในการประกอบอาชีพ
4. สามารถนำความรู้ทางทฤษฎีมาประยุกต์ใช้ในทางปฏิบัติ
5. สามารถแก้ปัญหาจากการปฏิบัติงานจริง เช่น การแก้ปัญหาต่างๆ การทำงานเป็นทีม

1.6 ขั้นตอนการดำเนินงาน

1. ศึกษาการใช้งานโปรแกรม MATLAB เบื้องต้น
2. ฝึกการใช้งานโปรแกรม MATLAB โดยการประมวลผลภาพดิจิตอลแบบต่างๆ
3. ศึกษาหลักการประมวลผลภาพดิจิตอล
4. เขียนโปรแกรมการประมวลผลภาพดิจิตอลและจัดทำเป็นวีดีโอ
5. เขียนโปรแกรมตรวจสอบเสียงการแจ้งเตือน
6. เขียนโปรแกรมตรวจสอบการตั้งค่าเวลา
7. ทดสอบการทำงานของโปรแกรมตรวจสอบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล
8. จัดเก็บข้อมูลและทฤษฎีที่เกี่ยวข้อง และ จัดทำรูปเล่มรายงาน

บทที่ 3
การทดลอง

3.1 บทนำ

ในบทนี้จะกล่าวถึงอุปกรณ์ในการทำงาน ขั้นตอนการทำงาน การทดลอง และการแสดง Data Flow Diagram (DFD) ในการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล โดยในการทำงานจะใช้โปรแกรม MATLAB และ Image processing toolbox โปรแกรม MATLAB จัดเป็นโปรแกรมที่สามารถตอบสนองผู้ใช้และแสดงผลการคำนวณได้อย่างทันทีทันใด การเขียนโปรแกรมไว้ในรูปแบบที่ง่ายต่อการใช้งาน นอกจากนั้น ยังสามารถแสดงผลออกมาเป็นรูปภาพหรือกราฟได้ และเนื่องจากเป็นโปรแกรมที่มีความพัฒนามาอย่างต่อเนื่อง ทำให้ในปัจจุบันโปรแกรม MATLAB ได้มีฟังก์ชั่นการใช้งานต่างๆมากมายหรือที่เรียกว่า Toolbox ที่สามารถวิเคราะห์งานประเภทต่างๆ ได้อย่างง่ายดาย เช่น การใช้โปรแกรม MATLAB ในการเขียนฟังก์ชันการหาขอบภาพ (Edge detection) การทำ Correlation การ Crop ภาพ การใส่สี Gray scale เป็นต้น

3.2 อุปกรณ์การทำงานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

1.โปรแกรม MATLAB

2.กล้องแว็บแคม Logitech

3.ขาตั้งกล้อง

4.วัตถุรูปทรงเรขาคณิตแบบต่างๆ

5.ลำโพง

6.คอมพิวเตอร์ Notebook

3.3 ขั้นตอนการทำงานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

ระบบสามารถแบ่งขั้นตอนการทำงานหลักๆ ได้ดังนี้

3.3.1 ขั้นตอนการรับข้อมูลภาพ

เป็นส่วนของระบบที่ทำหน้าที่รับข้อมูลเพื่อนำไปใช้ในการตรวจสอบวัตถุต้องสงสัย โดยจะรับภาพจากกล้องเว็บแคม (webcam)เข้ามาเป็นภาพถ่ายดิจิตอล ลักษณะเป็นภาพสี RGB มีนามสกุลภาพเป็น .JPEG และมีขนาดไม่เกิน 640 x 480 pixels

3.3.2 ขั้นตอนการประมวลผลของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

จะเป็นส่วนของการประมวลผลของระบบโดยรวม มีการทำงานดังนี้

1) ทำการ Capture ภาพ และ บันทึกภาพเก็บไว้ใน Folder Work ในโปรแกรม MATLAB
2) แล้วทำการโหลดภาพจาก Folder Work ออกมาประมวลผล โดยการเช็คค่า Correlation ของภาพ
3) กรณีที่ไม่พบวัตถุจะทำการวนกลับไปทำในข้อ 1) และ 2) จนกว่าจะพบวัตถุ
4) กรณีที่พบวัตถุจะทำการจับเวลา 5 นาที แล้วกลับไปทำในข้อ 1) และ 2) เพื่อทำการเช็คว่าวัตถุอยู่หรือไม่ ถ้าไม่พบวัตถุก็จะกลับไปทำในข้อ 1) และ 2) ซ้ำอีกรอบ แต่ถ้าพบวัตถุก็จะทำการส่งเสียงเตือนว่าพบวัตถุต้องสงสัย

3.3.3 ขั้นตอนการทำงานของประมวลผลภาพ

จะเป็นส่วนของระบบที่ทำหน้าที่นำข้อมูลนำเข้ามาประมวลผล มีการทำงานดังนี้

1) โหลดภาพที่ทำการถ่ายเก็บออกมาแล้วทำการตัดภาพตรงส่วนที่เราต้องการให้เป็นเป้าหมายที่เราจะหา
2) ทำการเปลี่ยนภาพต้นฉบับกับภาพเป้าหมายจากภาพสี RGB เป็น Gray Scale
3) ทำการหาขอบภาพของภาพต้นฉบับกับภาพเป้าหมายโดยวิธีโซลเบล (Sobel)
4) นำภาพต้นฉบับกับภาพเป้าหมายที่ได้จาการหาขอบภาพมาทำการ Correlation กัน
5) ทำการหาตำแหน่ง max และ min เพื่อหาพิกัดของพื้นที่สี่เหลี่ยมของภาพเป้าหมาย
6) นำมาทำเป็นตัวกรองให้ตรงส่วนที่เป็นเป้าหมายยังคงเป็นภาพสีแบบ RGB เหมือนเดิม แต่ตรงส่วนที่ไม่ใช่เป้าหมายเราจะทำให้เป็นสีแบบ Gray scale
7) ทำการพล็อตกรอบให้กับวัตถุที่เป็นเป้าหมาย

(ก)

(ค)

(ข)

(ง)

รูปที่ 3.1 ตัวอย่างภาพก่อนนำไปประมวลผล

(ก)

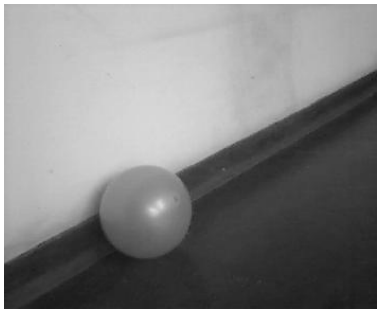
(ค)

(ข)

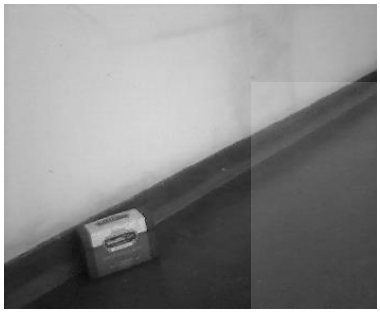
(ง)

รูปที่ 3.2 ตัวอย่างภาพเป้าหมาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

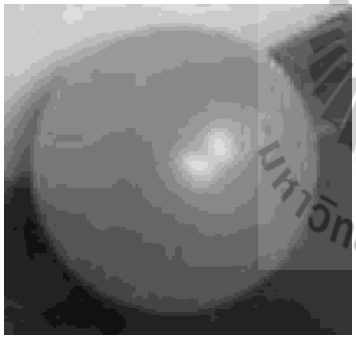


(ก) (ค)

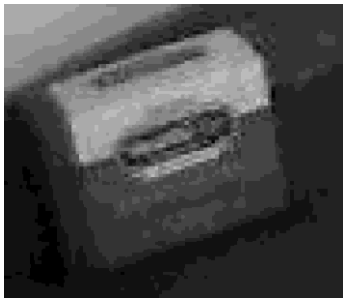


(ข) (ง)

รูปที่ 3.3 ตัวอย่างภาพต้นฉบับที่ผ่านการทำ Gray scale



(ก) (ค)



(ข) (ง)

รูปที่ 3.4 ตัวอย่างภาพเป้าหมายที่ผ่านการทำ Gray Scale

มหาวิทยาลัยเทคโนโลยีสุรนารี

(ก) (ค)

(ข) (ง)

รูปที่ 3.5 ตัวอย่างภาพต้นฉบับที่ทำการหาขอบภาพ (Edge detection)

(ก) (ค)

(ข) (ง)

รูปที่ 3.6 ตัวอย่างภาพเป้าหมายที่ทำการหาขอบภาพ (Edge detection)

มหาวิทยาลัยเทคโนโลยีสุรนารี

(ก)

(ค)

(ข)

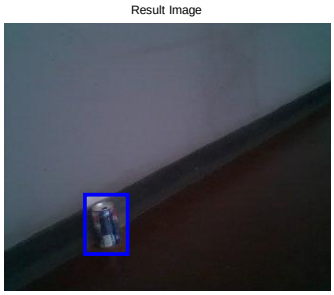
(ง)

รูปที่ 3.7 ตัวอย่างภาพที่นำมาเป็นตัวกรองเพื่อทำให้ตรงส่วนที่เป็นเป้าหมายยังเป็นภาพสี

(ก)

(ค)

(ข)

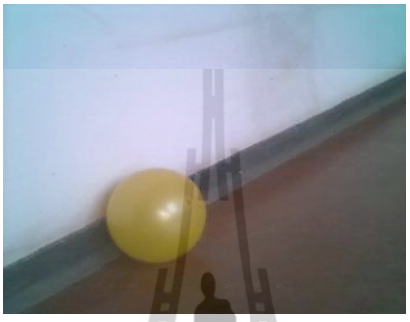


(ง)

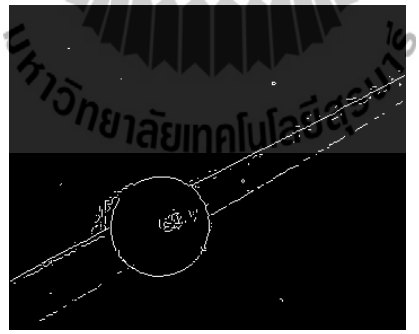
รูปที่ 3.8 ภาพผลลัพธ์

3.4 การทดลองการหาขอบภาพ (Edge detection)

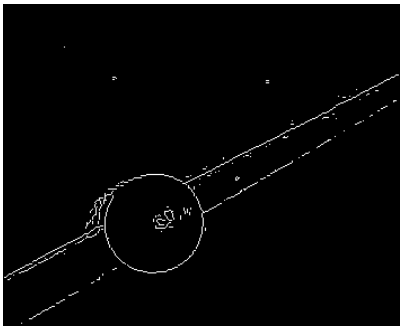
ในการหาขอบภาพจะใช้แบบ Sobel method Prewitt method Roberts method Laplacian of gaussian method and Canny method เพื่อที่จะเปรียบเทียบคุณภาพของภาพที่ผ่านการหาขอบภาพโดยวิธีการต่างๆกัน เพื่อที่จะหาความแตกต่างกันและศึกษาว่าวิธีการใดที่เหมาะสมในการนำไปใช้ในการทดลองมากที่สุด โดยภาพรูปที่ 3.9 เป็นภาพต้นฉบับที่จะทำการหาขอบภาพ และภาพอื่นๆเป็นวิธีการหาขอบภาพแบบต่างๆ



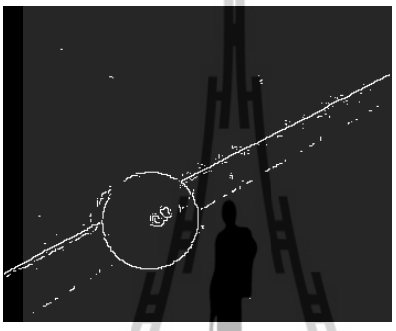
รูปที่ 3.9 ตัวอย่างภาพต้นฉบับ



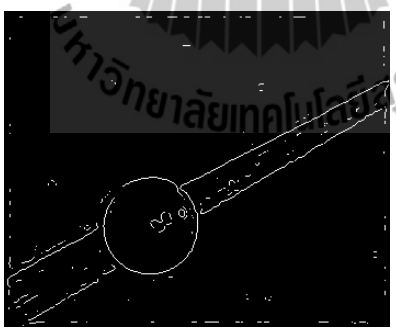
รูปที่ 3.10 ตัวอย่างการหาขอบภาพแบบวิธี Sobel method



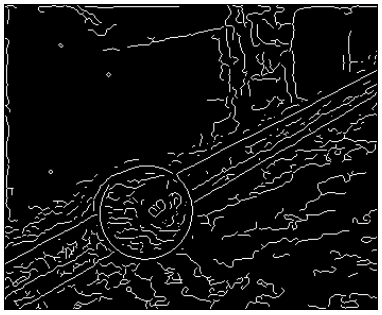
รูปที่ 3.11 ตัวอย่างการหาขอบภาพแบบวิธี Prewitt method



รูปที่ 3.12 ตัวอย่างการหาขอบภาพแบบวิธี Roberts method



รูปที่ 3.13 ตัวอย่างการหาขอบภาพแบบวิธี Laplacian of gaussian method



รูปที่ 3.14 ตัวอย่างการหาขอบภาพแบบวิธี Canny method

จากการแสดงตัวอย่างภาพนำภาพแต่ละแบบมาทำการเปรียบเทียบในการหาขอบภาพที่ผ่านมาเราจะสังเกตเห็นได้ว่าขอบภาพจะมีความชัดเจนของเส้นขอบที่แตกต่างกันออกไป โดยเลือกใช้แบบ Sobel method เพราะว่าเป็นแบบที่มีความชัดเจนของเส้นขอบไม่หนาและละเอียดเกินไปเพื่อที่จะนำมาหาวัตถุต้องสงสัย

3.5 การทดลองในการตรวจจับวัตถุต้องสงสัยโดยการทำ Correlation

ในการทดลองนี้จะเป็นการตรวจหาวัตถุต้องสงสัยโดยการทำ Correlation โดยการนำภาพต้นฉบับมาทำการ Correlation กับภาพเป้าหมายเพื่อหาตำแหน่งของภาพเป้าหมายและทำการ mask จุดในภาพต้นฉบับตรงส่วนที่เป็นเป้าหมาย แล้วทำการหาตำแหน่ง max , min เพื่อหาพื้นที่สี่เหลี่ยมตรงส่วนที่เป็นเป้าหมาย และนำมาทำเป็นตัวกรองให้ตรงส่วนที่เป็นเป้าหมายยังคงเป็นภาพสีแบบ RGB เหมือนเดิม แต่ตรงส่วนที่ไม่ใช่เป้าหมายเราจะทำให้เป็นสีแบบ Gray scale มีขั้นตอนการทำดังนี้

1.ภาพต้นฉบับที่ได้จากการ Capture เพื่อนำไปเป็นแม่แบบในการค้นหา

รูปที่ 3.15 ตัวอย่างภาพต้นฉบับ

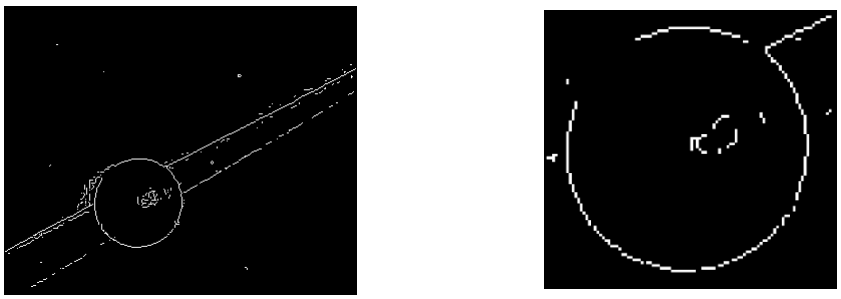
2.นำภาพต้นฉบับมาทำการตัด ตรงจุดที่เป็นเป้าหมายที่ต้องการค้นหา

รูปที่ 3.16 ตัวอย่างภาพเป้าหมาย

3.นำภาพรูปที่ 3.15 และรูปที่ 3.16 มาทำการเปลี่ยนภาพระดับสีเทา (Gray scale) เพื่อนำไปหาขอบภาพ

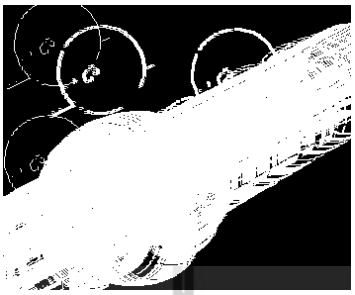
รูปที่ 3.17 ตัวอย่างภาพที่เปลี่ยนภาพระดับสีเทา Gray scale

4.นำภาพจากรูปที่ 3.17 มาทำการหาขอบของภาพ (Edge detection) โดยวิธี Sobel method



รูปที่ 3.18 ตัวอย่างภาพที่ทำการหาขอบของภาพ (Edge detection)

5.นำภาพจากรูปที่ 3.18 มาทำการหา Correlation เพื่อตรวจสอบค่า max ในการ Correlation โดยถ้ามีวัตถุจะได้ค่ามากกว่า 50 ขึ้นไป และถ้าไม่มีวัตถุจะได้ค่าน้อยกว่า 50



รูปที่ 3.19 ตัวอย่างภาพการหา Correlation

6.นำภาพจากรูปที่ 3.19 มาทำการหาตำแหน่ง max และ min จะได้พิกัดของพื้นที่สี่เหลี่ยมที่เป็นเป้าหมาย

รูปที่ 3.20 ตัวอย่างภาพที่นำมาเป็นตัวกรองเพื่อทำให้ตรงส่วนที่เป็นเป้าหมายยังเป็นภาพสี

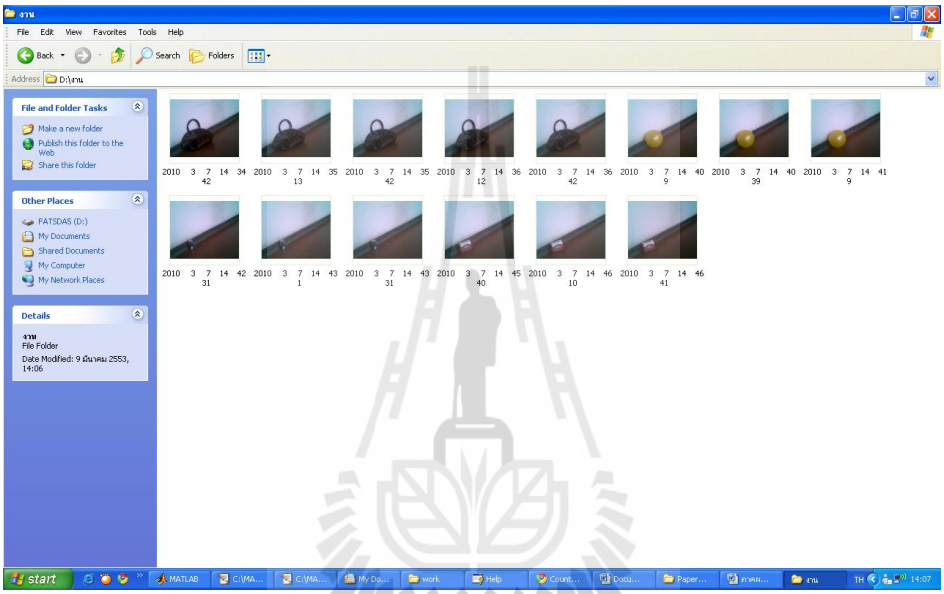
7.นำภาพจากรูปที่ 3.15 และรูปที่ 3.20 มาทำการคูณกันเพื่อต้องการให้ตรงส่วนที่เป็นเป้าหมายยังคงเป็นภาพสีแบบ RGB เหมือนเดิม แต่ตรงส่วนที่ไม่ใช่เป้าหมายเราจะทำให้เป็นสีแบบ Gray scale แล้วทำการพล็อตกรอบสี่เหลี่ยม

รูปที่ 3.21 ตัวอย่างภาพผลลัพธ์

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.6 Database ที่ใช้งานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

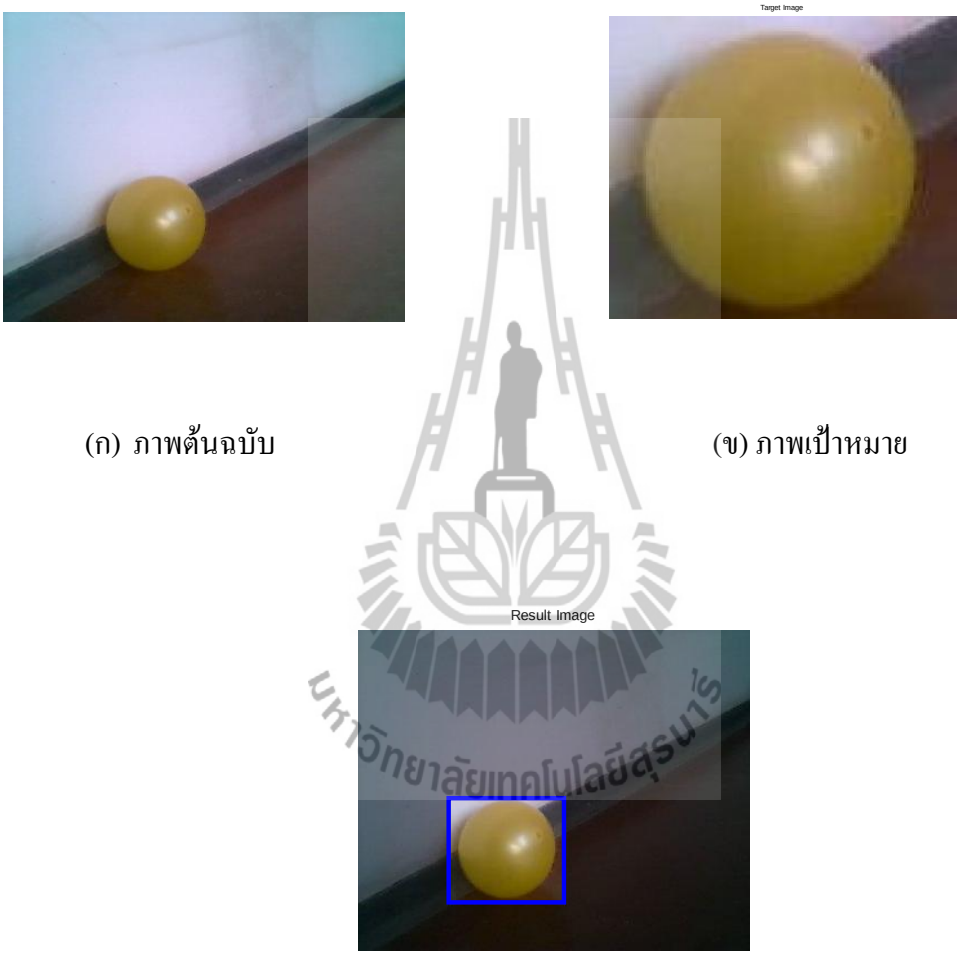
สำหรับฐานข้อมูลหรือ Database ที่เลือกใช้งาน ใช้ Folder ของ MATLAB เป็นเสมือนฐานข้อมูลในการจัดเก็บภาพและข้อมูลพื้นฐานต่างๆ เพื่อความสะดวกและรวดเร็วในการใช้งานระบบ เนื่องจากระบบไม่มีความแตกต่างของประเภทข้อมูลที่ต้องทำการจัดเก็บมากนัก โดยข้อมูลพื้นฐานที่จำเป็นต้องทำการจัดเก็บคือ ภาพจากกล้องและไฟล์ที่บันทึกการตั้งค่าของระบบ



รูปที่ 3.22 ตัวอย่าง Folder ที่ใช้เป็นฐานข้อมูลภาพของระบบ

3.7 แสดงการเก็บข้อมูลตัวอย่างงานที่ได้ของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

ตัวอย่างที่ 1 รูปแบบวัตถุทรงกลม

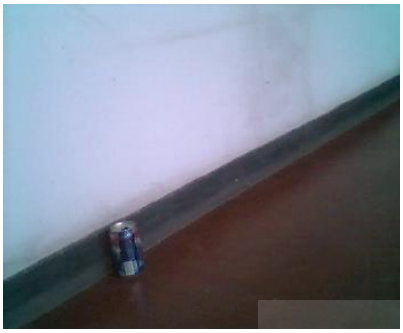


(ก) ภาพต้นฉบับ (ข) ภาพเป้าหมาย

(ค) ภาพผลลัพธ์

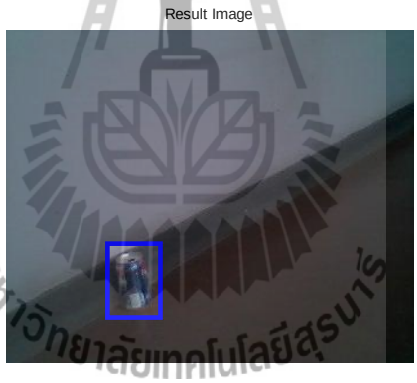
รูปที่ 3.23 ตัวอย่างงานรูปแบบวัตถุทรงกลม

ตัวอย่างที่ 2 รูปแบบวัตถุกระป๋อง



(ก) ภาพต้นฉบับ

(ข) ภาพเป้าหมาย



(ค) ภาพผลลัพธ์

รูปที่ 3.24 ตัวอย่างงานรูปแบบวัตถุกระป๋อง

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตัวอย่างที่ 3 รูปแบบวัตถุกระเป๋า

(ก) ภาพต้นฉบับ

(ข) ภาพเป้าหมาย

(ค) ภาพผลลัพธ์

รูปที่ 3.25 ตัวอย่างงานรูปแบบวัตถุกระเป๋า

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตัวอย่างที่ 4 รูปแบบวัตถุกล่อง

(ก) ภาพต้นฉบับ

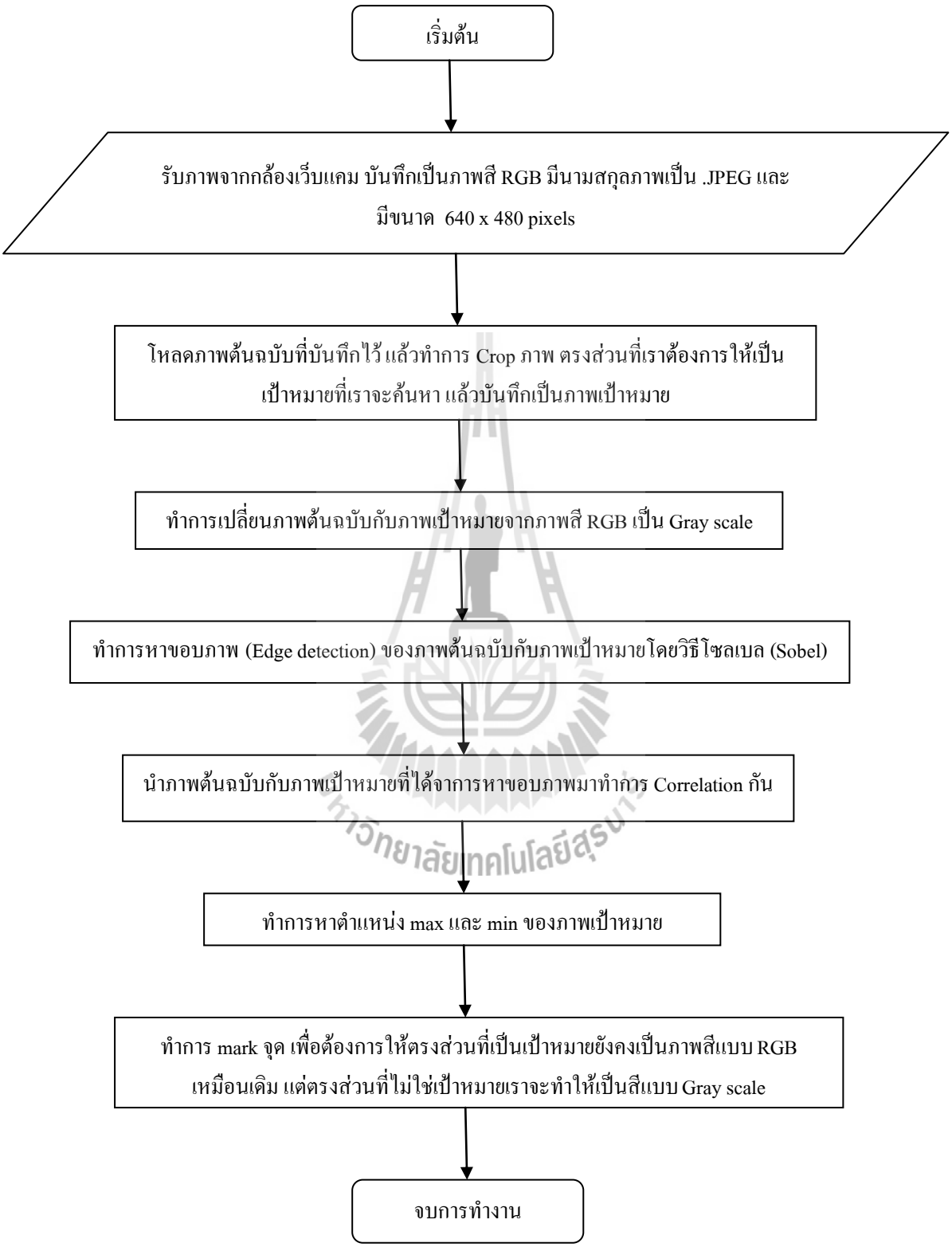
(ข) ภาพเป้าหมาย

(ค) ภาพผลลัพธ์

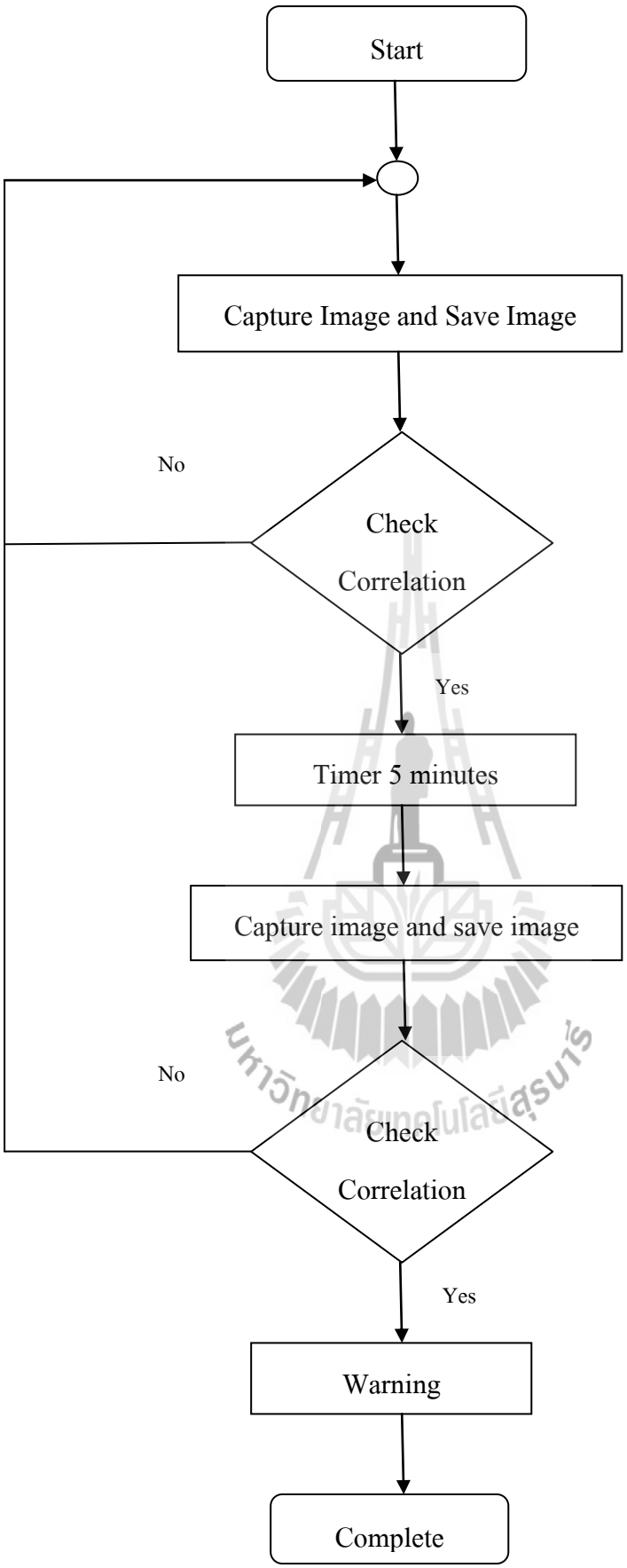
รูปที่ 3.26 ตัวอย่างงานรูปแบบวัตถุกล่อง

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.8 การแสดง Data Flow Diagram (DFD) การทำงานของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล



รูปที่ 3.27 ภาพแสดง Data Flow Diagram (DFD) การทำงานของประมวลผลภาพ



รูปที่ 3.28 ภาพแสดง Data Flow Diagram (DFD) การประมวลผลของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

อธิบายการทำงานของ Flow chart

เริ่มต้นทำการ Capture ภาพ เพื่อนำไปเช็คค่า Correlation ของภาพว่ามีวัตถุต้องสงสัยหรือไม่ ทำการเช็คค่า Correlation ในภาพโดยเช็คว่าค่า max ในการ Correlation เกิน 50 หรือไม่ ถ้าไม่เกิน50 แสดงว่าไม่พบวัตถุต้องสงสัย โปรแกรมจะทำการวนกลับไป Capture ภาพใหม่ ถ้าเกิน 50 แสดงว่าพบวัตถุต้องสงสัย โปรแกรมจะทำการจับเวลา 5 นาที แล้ววนกลับไป Capture ภาพใหม่แล้วเช็คค่า Correlation ซ้ำอีกครั้ง ค่า max ของการ Correlation เกิน 50 อีก พบว่ามีวัตถุต้องสงสัย โปรแกรมจะทำการส่งเสียงเตือน แต่ถ้าเช็คค่าแล้วค่า Correlation ไม่เกิน 50 โปรแกรมจะทำการวนกลับไป Capture ภาพใหม่

ตารางที่ 3.1 แสดงการบันทึกการทดลองในการตรวจจับวัตถุรูปทรงเรขาคณิตแบบต่างๆ

การทดลอง / รูปแบบวัตถุ	**ครั้งที่ 1**	**ครั้งที่ 2**	**ครั้งที่ 3**	**ครั้งที่ 4**	**ครั้งที่ 5**	**ครั้งที่ 6**	**ครั้งที่ 7**	**ครั้งที่ 8**	**ครั้งที่ 9**	**ครั้งที่ 10**	**สรุป**
กล่อง	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	ไม่พบวัตถุ	พบวัตถุ
กระเป๋า	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	ไม่พบวัตถุ	พบวัตถุ	ไม่พบวัตถุ	พบวัตถุ	พบวัตถุ
กระป๋อง	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	ไม่พบวัตถุ	พบวัตถุ	พบวัตถุ
ทรงกลม	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	พบวัตถุ	ไม่พบวัตถุ	พบวัตถุ

มหาวิทยาลัยเทคโนโลยีสุรนารี

วิเคราะห์ผลการทดลอง

จากตารางที่ 3.1 จะเห็นผลการทดลองพบว่าการทดลองในการตรวจจับวัตถุรูปทรงเรขาคณิตแบบต่างๆ ในการทดลองจะสามารถตรวจจับพบวัตถุได้ประมาณ 8-9 ครั้งมีเพียงแค่ 1-2 ครั้งที่ไม่สามารถตรวจจับวัตถุได้ จากการทดลอง 10 ครั้ง แล้วเมื่อนำมาสรุปรวมผลการทดลองจะสามารถตรวจจับพบวัตถุทุกชนิดได้ จากการทดลองเห็นได้ว่าโปรแกรมในการทำงานเกิดความผิดพลาดได้บ้าง เกิดจากโปรแกรมมีประสิทธิภาพในการทำงานได้ไม่สูงนัก

บทที่ 4

สรุปและข้อเสนอแนะ

4.1 บทนำ

ในบทนี้จะกล่าวถึงข้อสรุปการทำงาน ปัญหาอุปสรรคที่พบ ข้อจำกัดของระบบ และข้อเสนอแนะในการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล เป็นการพัฒนาโครงงาน “ระบบตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล ” พัฒนาได้สำเร็จตามจุดประสงค์ภายในเวลาที่กำหนด โดยปัจจุบันระบบตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล สามารถทำงานได้น่าพอใจระดับหนึ่ง คือสามารถทำงานพื้นฐานของระบบได้ดังนี้ บันทึกภาพ ตั้งค่าพื้นฐาน (เวลา ในการบันทึ ก ชื่อไฟล์ ที่ใช้จัดเก็บข้อมูล) ทำการประมวลผลภาพดิจิตอล และทำการส่งเสียงเตือนว่าพบวัตถุต้องสงสัย แต่ยังไม่สามารถทำงานได้ในสถานการณ์จริง เป็นเหตุการณ์ที่ลำลองขึ้นมา

4.2 สรุปผลการทดลองการหาขอบภาพ

จากการทดลองการหาขอบภาพ นั้นจะทำให้ได้ภาพจากการแสดงตัวอย่างภาพนำภาพแต่ละแบบมาทำการเปรียบเทียบในการหาขอบภาพที่ผ่านมาเราจะสังเกตเห็นได้ว่าขอบภาพจะมีความชัดเจนของเส้นขอบที่แตกต่างกันออกไป ซึ่งจะสังเกตเห็นได้ว่าแบบต่างๆจะมีความละเอียดชัดเจนของเส้นขอบที่ละเอียดเกินไป เส้นขอบหนาบางเกินไป โดยในการทดลองจะเลือกใช้แบบ Sobel method เพราะว่าเป็นแบบที่มีความชัดเจนของเส้นขอบไม่หนาและละเอียดเกินไปเพื่อที่จะนำมาหาวัตถุต้องสงสัย

4.3 สรุปผลการทดลองในการตรวจจับวัตถุต้องสงสัยโดยการทำ Correlation

จากการทดลองนี้จะเป็นการตรวจหาวัตถุต้องสงสัยโดยการทำ Correlation โดยการนำภาพต้นฉบับมาทำการ Correlation กับภาพเป้าหมายเพื่อหาตำแหน่งของภาพเป้าหมายและทำการ mask จุดในภาพต้นฉบับตรงส่วนที่เป็นเป้าหมาย แล้วทำการหาตำแหน่ง max , min เพื่อหาพื้นที่สี่เหลี่ยมตรงส่วนที่เป็นเป้าหมาย และนำมาทำเป็นตัวกรองให้ตรงส่วนที่เป็นเป้าหมายยังคงเป็นภาพสีแบบ RGB เหมือนเดิม แต่ตรงส่วนที่ไม่ใช่เป้าหมายเราจะทำให้เป็นสีแบบ Gray scale แล้วทำการพล็อตกรอบสี่เหลี่ยม ซึ่งการทำ Correlation จะสามารถทำให้ตรวจพบวัตถุต้องสงสัย

4.4 ปัญหาและอุปสรรค

4.4.1 ระบบสามารถวิเคราะห์วัตถุและแจ้งเตือนได้ถูกต้องเมื่อพบวัตถุเพียงชิ้นเดียว

4.4.2 กล้องที่ใช้เป็น webcam ซึ่งมีความละเอียดไม่มาก

4.4.3 เนื่องจากไม่เคยได้ศึกษาโปรแกรม MATLAB มาก่อนทำให้ไม่มีความชำนาญในการใช้โปรแกรมทำให้ช่วงแรกยังทำงานได้ไม่เต็มที่นักและยังมีข้อบกพร่องอยู่สมควร ต่อมาเมื่อสามารถปรับตัวและได้รับคำแนะนำจากอาจารย์ที่ปรึกษา จึงทำงานได้ดีขึ้นตามลำดับ

4.5 ข้อจำกัดของระบบ

4.5.1 ระบบทำงานได้ภายใต้สถานการณ์จำลอง

4.5.2 ระบบทำการวิเคราะห์วัตถุว่าเป็นวัตถุต้องสงสัยหรือไม่ได้เพียงครั้งละ 1 ชิ้น

4.5.3 ระบบไม่สามารถจำแนกวัตถุที่เพิ่มเข้ามาในตำแหน่งเดิมที่ตรวจพบวัตถุอยู่แล้ว

4.6 ข้อเสนอแนะในการพัฒนาต่อไป

4.6.1 พัฒนาส่วนของการวิเคราะห์และตรวจสอบวัตถุให้มีประสิทธิภาพสูงขึ้น

4.6.2 พัฒนาส่วนของการวิเคราะห์และตรวจสอบวัตถุให้สามารถวิเคราะห์และตรวจสอบวัตถุได้หลายชิ้น

4.6.3 สร้างตัวแบบการเรียนรู้เพื่อให้ระบบสามารถแยกแยะวัตถุออกจากสิ่งมีชีวิตได้ โดยมีทฤษฎีรองรับ (Neural Network: เครือข่ายประสาทเทียม)

4.6.4 พัฒนาระบบให้สามารถใช้ในสถานการณ์จริงได้

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.7 รูปภาพการปฏิบัติงาน

รูปที่ 4.1 ตัวอย่างภาพปฏิบัติงาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

ประวัติผู้เขียน

นางสาวสุนิสา เรืองศรี เกิดเมื่อวันที่ 1 มีนาคม พ.ศ. 25 30 ภูมิลำเนาอยู่ที่ ตำบลหนองผักนาก อำเภอสามชุก จังหวัดสุพรรณบุรี สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนสามชุกรัตนโภคาราม อำเภอสามชุก จังหวัดสุพรรณบุรี เมื่อปี พ.ศ. 254 8 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นางสาวสุวารี ศรีอำพรรณ์ เกิดเมื่อวันที่ 13 มกราคม พ.ศ. 25 30 ภูมิลำเนาอยู่ที่ ตำบลสามชุก อำเภอสามชุก จังหวัดสุพรรณบุรี สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนสามชุกรัตนโภคาราม อำเภอสามชุก จังหวัดสุพรรณบุรี เมื่อปี พ.ศ. 254 8 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี

บรรณานุกรม

[1] Digital Image processing using MATLAB / Rafael C. Gonzalez, Richard E. Woods and Steven L.Eddins

[2] Digital image processing / Rafael C. Gonzalez, Richard E. Woods

[3] http://sutir.sut.ac.th:8080/sutir/handle/123456789/2526

[4] http://sutir.sut.ac.th:8080/sutir/handle/123456789/1200

[5] http://fivedots.coe.psu.ac.th/~montri/Teaching/240-373/Chapter8.pdf

[6] www.docstoc.com/.../305434-Digital-Image-Processing-การประมวลผลภาพดิจิตอล

[7] http://guru.sanook.com/search/filter/

[8] http://fivedots.coe.psu.ac.th/~montri/Teaching/image/image.htm

[9] http://research.crma.ac.th/2549/index.php/SP49-IW_Final_Report

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ก

โปรแกรม MATLAB

MATLAB เป็นโปรแกรมคอมพิวเตอร์สมรรถนะสูงเพื่อใช้ในการคำนวณทางเทคนิค MATLAB ได้รวมการคำนวณ การเขียนโปรแกรมและการแสดงผลรวมกันอยู่ในตัวโปรแกรมเดียว ได้อย่างมีประสิทธิภาพ และอยู่ในลักษณะที่ง่ายต่อการใช้งาน นอกจากนี้ลักษณะของการเขียนสมการในโปรแกรมก็จะเหมือนการเขียนสมการคณิตศาสตร์ที่เราคุ้นเคยดีอยู่แล้ว งานที่ทั่วไปที่ใช้ MATLAB ก็เช่นการคำการคำนวณทั่วไป การสร้างแบบจำลองและการทดสอบแบบจำลอง การวิเคราะห์ข้อมูล การแสดงผลในรูปกราฟทั้งโดยทั่วไปและกราฟทางด้านทางวิทยาศาสตร์และวิศวกรรม สามารถสร้างโปรแกรมในลักษณะที่ติดต่อกับผู้ใช้ทางกราฟฟิกส์

การทำงานของ MATLAB จะสามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรง (Interactive)คือการเขียนคำสั่งเข้าไปทีละคำสั่ง เพื่อให้ MATLAB ประมวลผลไปเรื่อยๆ หรือสามารถที่จะรวบรวมชุดคำสั่งเรานั้นเป็นโปรแกรมก็ได้ ข้อสำคัญอย่างหนึ่งของ MATLAB ก็คือข้อมูลทุกตัวจะถูกเก็บในลักษณะของ array คือในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็กๆขึ้น (หรือจะได้รับการแบ่งเป็น element นั่นเอง) ซึ่งการใช้ตัวแปรเป็น array ในMATLAB นี้เราไม่จำเป็นที่จะต้องจอง dimension เหมือนกับการเขียนโปรแกรมในภาษาขั้นต่ำตัวไป ซึ่งทำให้เราสามารถที่จะแก้ปัญหาของตัวแปรที่อยู่ในลักษณะของ matrix และ vector ได้โดยง่าย ซึ่งทำให้เราลดเวลาการทำงานลงได้อย่างมากเมื่อเทียบกับการเขียนโปรแกรมโดย C หรือ Fortran

MATLAB เป็นโปรแกรมสำเร็จรูปที่ใช้กันอย่างแพร่หลายในแวดวงของนักวิทยาศาสตร์และวิศวกรในปัจจุบัน ชื่อโปรแกรม MATLAB นั้นย่อมาจาก Matrix Laboratory โดย MATLAB นั้นได้เริ่มต้นขึ้นเพื่อต้องการให้เราสามารถแก้ปัญหาตัวแปรที่มีลักษณะเป็น Matrix ได้ง่ายขึ้น สำหรับ MATLAB ได้เริ่มพัฒนาครั้งแรกโดย Dr. Cleve Molor ซึ่งเขียนโปรแกรมนี้ขึ้นมาด้วยภาษา Fortran โดยโปรแกรมนี้ได้พัฒนาภายใต้โครงการ LINPACK และ EISPACK

สำหรับในปัจจุบันนี้ MATLAB ได้ถูกเขียนขึ้นโดยใช้ภาษา C โดยบริษัท Math Works ภายใต้โครงการ LAPACK และ ARPACK ถ้าหากเราจะเริ่มนับจากโปรแกรมที่ออกเผยแพร่เป็นครั้งแรกที่มีผู้ร่วมเขียนโปรแกรมไม่กี่คน จนกระทั่งทุกวันนี้มีทีมงานขนาดใหญ่ที่ทำงานในการพัฒนาโปรแกรมให้มีประสิทธิภาพสูงขึ้น ซึ่งทำให้ทุกวันนี้ MATLAB เป็นโปรแกรมที่สุดยอดในการคำนวณที่คำนวณด้าน matrix สำหรับงานทางวิทยาศาสตร์และวิศวกรรมโปรแกรมหนึ่ง นับจากวันแรกที่ได้เริ่มโครงการขึ้นจนกระทั่งในไตรมาสสุดท้ายของปี ค.ศ.2000 ได้พัฒนาเป็นMATLAB 6.0 ซึ่งเป็นการปรับปรุงใหม่และออกสู่ตลาดเป็นครั้งที่ 12 สำหรับในมุมมองของการศึกษานั้น MATLAB ถือได้ว่าเป็นเครื่องมือที่สำคัญอันหนึ่งสำหรับนักศึกษาทางด้านวิทยาศาสตร์และ

เทคโนโลยีที่จะใช้เป็นเครื่องมือในการคำนวณ และขณะนี้หลายๆมหาวิทยาลัยได้ยกอันดับของ MATLAB ขึ้นจากโปรแกรมสำเร็จรูป ให้เป็นภาษาสำหรับการใช้งานทางด้านเทคโนโลยี นั่นคือมีระดับเป็นภาษาเหมือนกับภาษา C หรือ Fortran นั่นเองนอกเหนือจากเพื่อการเรียนการสอนในสถาบันการศึกษาแล้ว MATLAB ยังเป็นเครื่องมือสำคัญที่ใช้ในงานวิจัย งานพัฒนาและการวิเคราะห์ของหน่วยงานต่างๆมากมาย

นอกเหนือจากตัวโปรแกรม MATLAB เองแล้ว บริษัท Math Works ผู้ผลิต MATLAB ยังได้ผลิตเครื่องมือหรือที่เรียกว่า toolbox ซึ่งเป็นโปรแกรมที่เขียนขึ้นเพื่อประกอบกับการใช้ MATLAB สำหรับงานที่จำเพาะเจาะจงหลายประเภท Toolbox นั้นเป็นการนำเอาโปรแกรมที่เขียนขึ้นเป็นฟังก์ชันสำหรับMATLAB เพื่อรวมเข้าเพื่อให้ผู้ใช้งานมีความสะดวกในการเรียกใช้มากขึ้น ทำให้ผู้ใช้ไม่จำเป็นที่จะต้องสร้างโปรแกรมขึ้นมาใช้งานเอง โดย toolbox ที่สร้างขึ้นจะครอบคลุมการทำงานด้านต่างๆมากมายเช่น signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation เป็นต้น

ถ้าหากจะสรุปโดยรวมแล้วความสามารถหลักของ MATLAB ที่ทำให้เป็นโปรแกรมที่เหมาะสมกับการทำงานทางด้านวิศวกรรมด้วยเหตุผลดังนี้

- MATLAB เป็นโปรแกรมเพื่อการคำนวณและแสดงผลได้ทั้งตัวเลขและรูปภาพซึ่งมีประสิทธิภาพสูง โดยทางบริษัท Math Works ผู้ผลิตได้ให้นิยามว่าเป็น High-Performance Numeric Computation and Visualization Software
- MATLABจะควบคุมการทำงานด้วยชุดคำสั่งและยังสามารถรวบรวมชุดคำสั่งเป็นโปรแกรมได้อีกด้วย
- MATLAB มี function ที่เหมาะสมกับงานทางวิศวกรรมพื้นฐานมากมาย นอกจากนั้นผู้ใช้ยังสามารถเขียน function ขึ้นมาใหม่โดยสามารถใช้ประโยชน์จาก function ที่มีอยู่แล้วเพื่อให้เหมาะสมกับงานของผู้ใช้แต่ละกลุ่ม
- ลักษณะการเขียนโปรแกรมใน MATLAB จะใกล้เคียงการเขียนสมการทางคณิตศาสตร์ที่เราคุ้นเคยจึงง่ายกว่าการเขียนโปรแกรมโดยใช้ภาษาชั้นสูงเช่น C, FORTRAN หรืออื่นๆ
- MATLAB มีความสามารถในการเขียนกราฟและรูปภาพทั้ง 2 มิติและ 3 มิติได้อย่างมีประสิทธิภาพ
- MATLAB สามารถทำ Dynamic Link กับโปรแกรมอื่นๆได้ไม่ว่าจะเป็น Word, Excel หรืออื่นๆที่ร่วมทำงานอยู่บน windows

- MATLAB มี toolbox หรือชุด function พิเศษสำหรับผู้ใช้ที่ต้องการใช้งานเฉพาะทางหรืองานด้านวิศวกรรมขั้นสูงอื่นๆ

MATLAB มีโปรแกรมที่จำหน่ายแก่นักศึกษาโดยเฉพาะซึ่งจะมีราคาต่ำกว่าราคาปกติมากแต่มีประสิทธิภาพเท่าเทียมกัน แม้ว่าอาจจะมีการจำกัดขีดความสามารถของโปรแกรมบ้างแต่ก็เพียงพอสำหรับนักศึกษาที่จะใช้เพื่อการศึกษาในระดับอุดมศึกษาและ MATLABยังมีความสามารถและข้อดีอื่นๆอีกมากมายซึ่งจะได้กล่าวในรายละเอียดในภายหลัง

2. ข้อแนะนำเบื้องต้นเกี่ยวกับการทำงานของ MATLAB

2.1 ระบบการทำงานของ MATLAB

ในการทำงานของ MATLAB เพื่อให้การทำงานเป็นไปตามจุดมุ่งหมาย MATLAB ได้แบ่งส่วนการทำงานของโปรแกรมออกเป็นส่วนหลักที่สำคัญ 5 ส่วนด้วยกัน

1. Development Environment.
2. The MATLAB Mathematical Function Library.
3. The MATLAB Language.
4. Handle Graphics
5. The MATLAB Application Program Interface (API)

ซึ่งแต่ละส่วนจะมีหน้าที่ควบคุมในการทำงานแบบหนึ่งๆ และประสานการทำงานระหว่างส่วนต่างๆ ไปพร้อมกันด้วย สำหรับรายละเอียดในการทำงานของส่วนต่างๆ มีดังนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

1. Development Environment

ในส่วนนี้จะเป็นชุดเครื่องมือที่ช่วยให้เราสามารถที่จะใช้ฟังก์ชันและไฟล์ต่างๆ โดยเครื่องมือหลายตัวในนี้จะมีลักษณะเป็น graphical user interface ซึ่งรวมถึง MATLAB Desktop และ Command Windows, command history และ browsers สำหรับเพื่อใช้ดู help, workspace, files และ search path ซึ่งทั้งหมดนี้จะได้กล่าวถึงในรายละเอียดต่อไป

2. The MATLAB Mathematical Function Library

ในส่วนนี้จะเป็นที่รวบรวมส่วนของโปรแกรมที่ได้รวบรวมเป็นไฟล์ย่อยๆ ไว้ ไฟล์แต่ละไฟล์จะเป็นไฟล์ที่เขียนขึ้นมาเพื่อใช้กำหนดลักษณะในการคำนวณหรือ Algorithms แบบต่างๆ นับจากฟังก์ชันง่ายๆ เช่นการบวก ฟังก์ชันตรีโกณมิติพื้นฐาน เช่น sine, cosine ไปจนถึงฟังก์ชันที่มีความ

ซับซ้อนมีขั้นตอนในการคำนวณมากมาย เช่นการหา inverse ของ matrix การหา eigen values และ eigenvector หรือfast Fourier transforms เป็นต้น

3. The MATLAB Language

ส่วนนี้จะเป็นภาษาระดับสูงที่ใช้ตัวแปรเป็น Matrix หรือ array ซึ่งมีคำสั่งที่ใช้ในการควบคุมการทำงานของโปรแกรม การทำงานของฟังก์ชัน การกำหนดโครงสร้างของตัวแปรแบบต่างๆ กำหนด inputและ output ของโปรแกรม ซึ่งทั้งหมดนี้จะช่วยทำให้ในการเขียนโปรแกรม Matlab แต่ละโปรแกรมจะเป็นโปรแกรมที่มีขนาดเล็กกว่าเมื่อเทียบกับโปรแกรมที่ใช้เพื่อวัตถุประสงค์เดียวกัน แต่ผู้ใช้ต้องเขียนฟังก์ชันการทำงานทุกขั้นตอนขึ้นมาเอง

4. Handle Graphics

ส่วนนี้จะเป็นส่วนที่ใช้แสดงกราฟฟิกส์และรูปภาพต่างๆ รวมถึงคำสั่งระดับสูงที่ใช้ในการแสดงผลในสองและสามมิติ การจัดรูปแบบในลักษณะ image processing การทำภาพเคลื่อนไหว นอกจากนี้ในส่วนนี้ยังได้รวมเอาภาษาในระดับต่ำไว้เพื่อให้เราสามารถปรับแก้รูปภาพต่างๆให้เป็นไปตามที่เราต้องการได้มากที่สุด รวมถึงการสร้าง Graphic User Interface ภายใต้การทำงานของ MATLABด้วย

5. The MATLAB Application Program Interface (API)

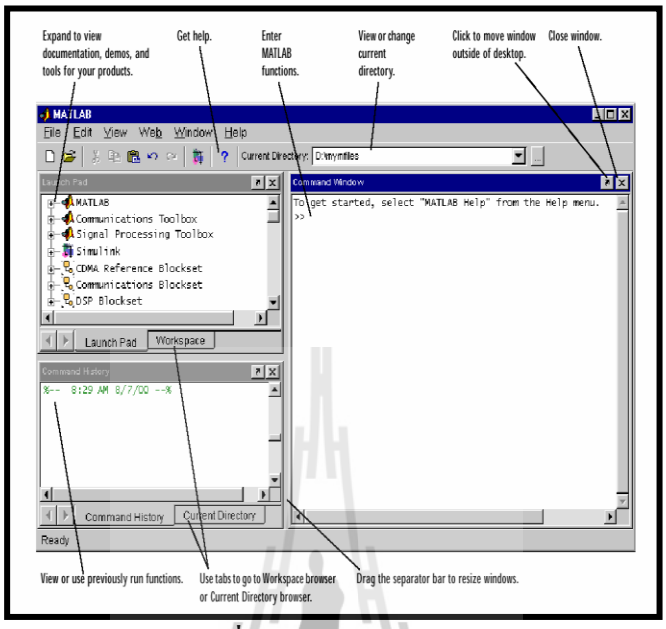
ส่วนนี้จะเป็น library ที่ให้เราสามารถที่จะเขียนโปรแกรมขึ้นในภาษา C หรือ Fortran แล้วมีการเชื่อมโยงการทำงานเข้ากับ MATLAB ซึ่งในส่วนนี้ยังได้รวมถึงการเขียนโปรแกรมขึ้นมาแล้วเรียกฟังก์ชันของ MATLAB ไปใช้งาน (dynamic linking), ซึ่งจะทำให้ MATLAB มีหน้าที่เสมือน engine ในการคำนวณ รวมถึงสามารถที่จะเขียนหรืออ่าน MAT-file ได้ด้วย

2.2 เริ่มการทำงานกับ MATLAB

สำหรับการทำงานภายใต้ระบบปฏิบัติการ Windows เราสามารถที่จะเริ่มการทำงานของ MATLAB ได้โดยการใช้เมาส์กดที่ shortcut ของ MATLAB ซึ่งจะปรากฏอยู่บน desktop หลังจากที่เราได้ติดตั้งโปรแกรมนี้ลงไปเรียบร้อยแล้ว หรืออาจจะใช้เมาส์กดที่ปุ่ม Start แล้วเลือก MATLAB ภายใต้เมนู Programs เหมือนกับการเปิดโปรแกรมอื่นๆ ใน Windows ก็ได้

เมื่อเราเริ่มเปิดโปรแกรม MATLAB 6 สิ่งแรกที่เราจะพบในครั้งแรกก็คือ MATLAB desktop ที่ประกอบด้วยหน้าต่างย่อยๆ อีกหลายหน้าต่าง โดยหน้าต่างแต่ละอันนั้นจะทำหน้าที่เป็นเครื่องมือที่

จะช่วยเราในการจัดการเกี่ยวกับไฟล์ตัวแปร และอื่นๆเกี่ยวกับการทำงานของ MATLAB โดย MATLAB desktop จะมีลักษณะดังรูปต่อไปนี้



รูปที่ ก.1 MATLAB desktop

แม้ว่าในบางกรณี Launch Pad ของเราอาจมีลักษณะแตกต่างไปจากรูปข้างบนนี้บ้าง ตามแต่จำนวนโปรแกรมหรือ toolbox ที่เราได้บรรจุเข้าไปในการติดตั้ง MATLAB เราสามารถที่จะเปลี่ยนแปลงลักษณะของ desktop ได้ด้วยการเปิด ปิด เคลื่อนย้ายและปรับขนาดของเครื่องมือเหล่านี้ได้ นอกจากนั้นเรายังสามารถที่จะย้ายเครื่องมือเหล่านั้นออกไปนอก desktop หรือย้ายกลับเข้ามา (docking) วางไว้กับ desktop ได้ โดยเครื่องมือบน desktop เหล่านี้จะช่วยการทำงานในขั้นตอนที่ใช้บ่อยๆไม่ว่าจะเป็น shortcut หรือ context menus อีกทั้งเรายังสามารถที่จะกำหนดลักษณะพิเศษต่างๆ ของ desktop ให้เป็นไปตามที่เราต้องการได้ ด้วยการเลือก Preferences จากเมนู File เพื่อเปลี่ยนลักษณะของตัวหนังสือที่ใช้ใน Command Window สำหรับรายละเอียดในการปรับแก้ลักษณะของ desktop นี้ดูได้จากการกดปุ่ม Help ภายใต้หน้าต่าง Preferences

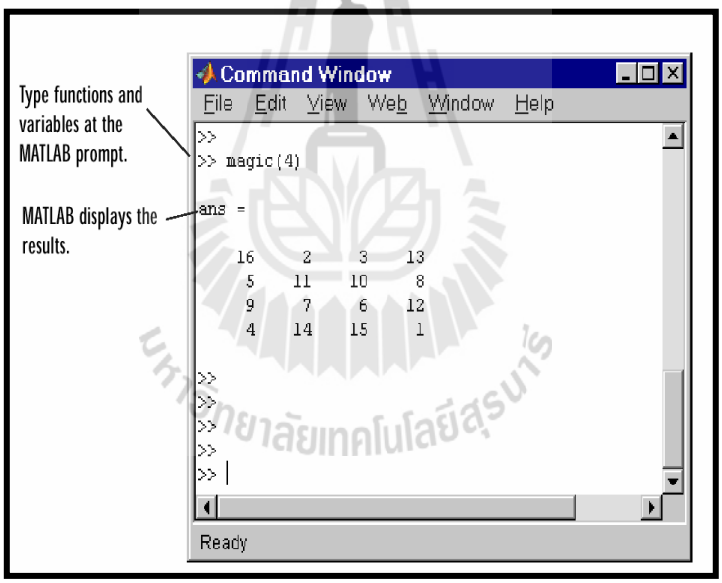
3. หน้าต่างที่มีใน MATLAB

ใน MATLAB 6.0 ซึ่งเป็น Version ใหม่ของ MATLAB จะประกอบด้วยหน้าต่างย่อยๆ หลายหน้าต่าง ในหัวข้อนี้จะเป็นการอธิบายถึงหน้าต่างที่อยู่ภายใต้ MATLAB desktop ซึ่งได้รับการพัฒนาขึ้นมาใช้ใน version 6 นี้ โดยแม้ว่าในความเป็นจริงเราสามารถที่จะใช้คำสั่งทั่วๆไปที่เป็นฟังก์ชันของ MATLABพิมพ์เข้าไปเพื่อให้ได้ผลตามที่เราต้องการได้เช่นกัน แต่การใช้หน้าต่างและ

เครื่องมือใน MATLAB desktop ซึ่งถือว่าเป็นรูปแบบใหม่ของ MATLAB 6 นี้ จะช่วยให้การใช้คำสั่งทำได้สะดวกและรวดเร็วขึ้นโดยเราสามารถสั่งคำสั่งเหล่านั้นผ่านเมาส์ได้และมีการปรับเปลี่ยนค่าต่างๆ ได้สะดวกและรวดเร็วขึ้นมาก สำหรับหน้าต่างที่สำคัญใน MATLAB Desktop จะมีอยู่ด้วยกัน 5 หน้าต่างคือ Command Windows, Command History Window, Current Directory Browser, Workspace Browser และ Launch Pad

3.1 Command Window

Command Window เป็นส่วนที่เราใช้ในการป้อนชุดคำสั่งเพื่อให้ MATLAB ทำงานตามคำสั่งนั้นและก็จะแสดงผลที่เป็นตัวหนังสือในหน้าต่างนี้ ซึ่งใน version ก่อนๆ ของ MATLAB ก็จะมีหน้าต่างนี้อยู่แล้ว ซึ่งเราสามารถที่จะกำหนดคำสั่งด้วยตัวอักษร เพื่อให้ MATLAB ทำงานตามที่เราต้องการได้นั่นเอง

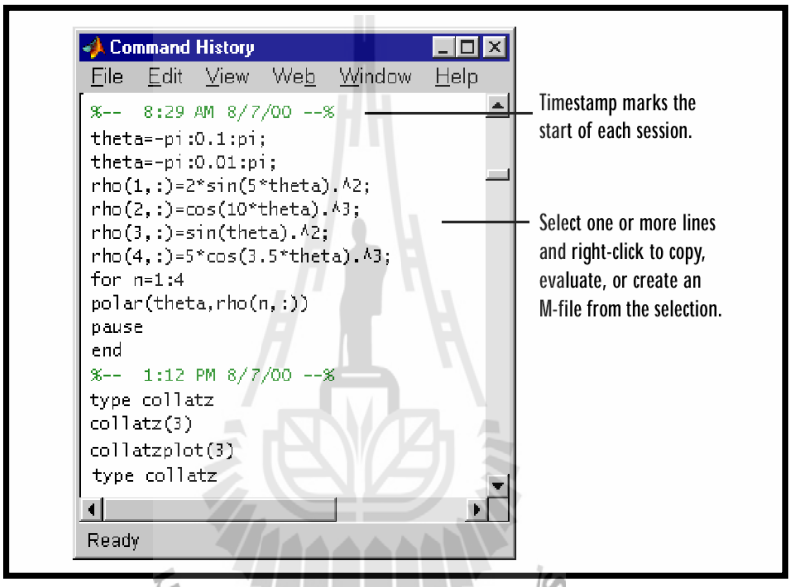


รูปที่ ก.2 หน้าต่าง Command Window

การที่เราจะป้อนคำสั่งให้ ที่ MATLAB Command Window โดย MATLAB จะรับคำสั่งเกือบทั้งหมดทางหน้าต่างนี้ ซึ่งทุกครั้งที่ MATLAB พร้อมที่จะรับคำสั่ง MATLAB จะแสดงเครื่องหมาย MATLAB prompt ในลักษณะ » ขึ้นสำหรับ Professional Edition หรือจะเป็นลักษณะ **EDU»** สำหรับ Student Edition เมื่อปรากฏเครื่องหมายดังกล่าวนี้ขึ้นก็แสดงว่า MATLAB พร้อมที่จะรับคำสั่งต่อไป แต่เพื่อความสะดวกสำหรับในเอกสารนี้เราจะไม่มีการแสดงเครื่องหมาย » หรือ **EDU»**หน้าชุดคำสั่ง

ต่างๆ เพราะเราจะได้ทราบในภายหลังว่าการป้อนชุดคำสั่งเหล่านี้อาจจะกำหนดผ่านทาง file ที่เขียนขึ้นเป็นชุดคำสั่งให้กับ MATLAB หรือที่เรานิยมเรียกกันสั้นๆว่า M-file

หน้าต่าง Command History นี้มีไว้เพื่อให้เราทราบว่าเราได้ใช้คำสั่งอะไรไปแล้วบ้าง โดยข้อมูลการใช้คำสั่งจะได้รับการบันทึกไว้ทุกครั้งที่มีการเปิดโปรแกรม MATLAB ขึ้นมาใช้ นอกจากนั้นยังบอกวัน-เวลาที่เราได้เข้ามาใช้โปรแกรมนี้ในแต่ละครั้งด้วย ในหน้าต่างนี้เราสามารถที่จะเลือกใช้คำสั่งที่เคยใช้มาก่อนหน้านี้แล้วอีกก็ได้ โดยการกดเมาส์สองครั้งที่คำสั่งนั้น หรือเราอาจจะเลือกที่จะทำสำเนาคำสั่งนั้นก็ได้

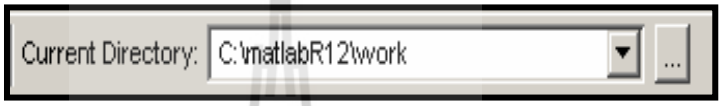


รูปที่ ก.3 หน้าต่าง Command History

เราสามารถที่จะลบ ข้อมูลในหน้าต่างนี้ได้ โดยการกดเมาส์ปุ่มขวาแล้วเลือกว่าจะลบเฉพาะตัวที่เลือก (Delete Selection) ลบตั้งแต่ต้นจนกระทั่งถึงตัวที่เราเลือก (Delete to Selection) หรือลบทั้งหมดเลย(Delete Entire History) ก็ได้นอกเหนือจากนั้นเรายังสามารถที่จะเลือกช่วงของคำสั่งที่เราใช้มาใน Command Windows นำมารวมกันแล้วสร้างเป็น M-file ได้อีกด้วย วิธีการก็คือเราเลือกช่วงคำสั่งที่เราต้องการขึ้นมาก่อนโดยการเลือกคำสั่งทีละคำสั่งแล้วกดปุ่ม Shift บนแป้นพิมพ์ค้างไว้ เมื่อได้คำสั่งครบตามต้องการแล้วให้กดเมาส์ปุ่มขวา แล้วเลือก Create M-File เราจะเข้าไปสู่ Editor เพื่อสร้าง M-file ต่อไป รายละเอียดในส่วนนี้จะกล่าวถึงอีกครั้งหนึ่งในการเขียน M-file

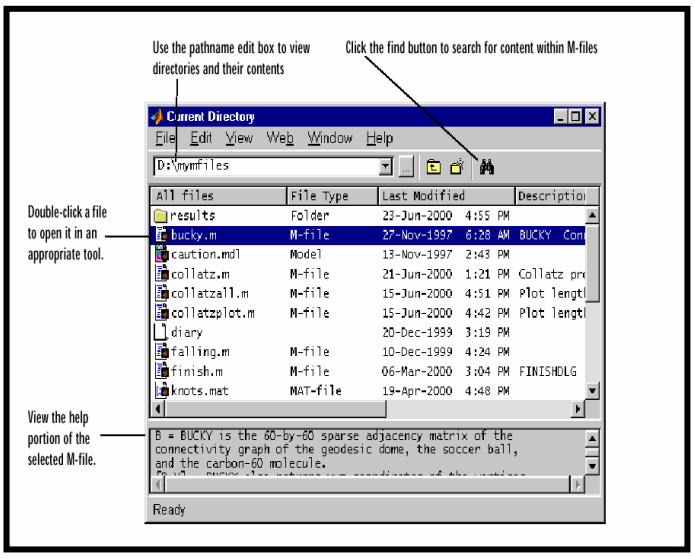
3.2 Current Directory Browser

คำสั่งที่กำหนดให้ MATLAB ทำนั้น MATLAB จะใช้ Current Directory และ Search Path เป็นจุดเริ่มต้นของการทำงานและเป็นพื้นที่ในการค้นหาข้อมูลหรือคำสั่งต่างๆ ตามที่ได้รับคำสั่งมา โดยการค้นหาจะจำกัดวงอยู่เฉพาะในสองส่วนหลักนี้เท่านั้น MATLAB จะไม่มีการค้นหา file หรือคำสั่งต่างๆนอกพื้นที่ดังกล่าว ดังนั้นคำสั่งหรือ M-file ต่างๆ ที่เราต้องการจะใช้งานนั้นจำเป็นอย่างยิ่งที่จะต้องอยู่ใน Current Directory หรือ Search Pathวิธีการที่จะดูว่าขณะนี้เราอยู่ใน Current Directory ใด ก็สามารถทำได้โดยดูที่แถบเครื่องมือ Current Directory ซึ่งอยู่ที่ Desktop Toolbar มีลักษณะตามที่แสดงในรูป นอกจากนั้นเรายังสามารถที่จะปรับเปลี่ยน Current Directory โดยการใช้แถบเครื่องมือนี้ได้อีกด้วย



รูปที่ ก.4 แถบเครื่องมือ Current Directory

โดยหากว่าเราต้องการจะปรับเปลี่ยนไปใช้ Current Directory ที่เราเคยใช้มาก่อนหน้านี้แล้ว เราสามารถกด เพื่อให้เมนูแสดง directory ที่เราเคยใช้เป็น Current Directory มาก่อน แต่ถ้าหากว่าเราต้องการที่จะเปลี่ยน Current Directory ไปอยู่ในdirectory ที่เราไม่เคยใช้มาก่อน เราจะต้องเลือกปุ่ม Browser ซึ่งจะเป็นการเปิดหน้าต่างใหม่เพื่อให้เราค้นหา directory ที่เราต้องการ เหมือนกับการค้นหา file ในระบบปฏิบัติการ Windows ทั่วๆไปนอกเหนือจากนั้นสำหรับการค้นหา ดู หรือ เปิด file ที่เราต้องการ เราสามารถที่จะทำได้โดยใช้ MATLAB Current Directory Browser ซึ่งเป็น desktop tool ที่มีหน้าที่เพื่อการนี้โดยเฉพาะ ลักษณะของ current Directory Browser จะมีลักษณะดังรูป



รูปที่ ก.5 หน้าต่าง Current Directory

เมื่อใช้ Current Directory Browser ตามรูปที่แสดงข้างบนนี้ เราสามารถที่จะค้นหา file โดยใช้ปุ่ม หรือเปิด file โดยการกดเมาส์สองครั้งที่ file ที่เราต้องการ นอกจากนั้นที่ส่วนล่างของหน้าต่างนี้ ยังแสดง help ของ M-file ที่เราได้เลือกในหน้าต่างส่วนบนด้วย สำหรับปุ่มและเมนูอื่นๆ ผู้ที่คุ้นเคยการทำงานกับ Windows คงจะทราบถึงความหมายของมันดีอยู่แล้ว เพราะจะมีลักษณะที่ใกล้เคียงกันนั่นเอง

Search Part

เพื่อที่จะให้ฟังก์ชันที่เราสั่งการจาก Command Window ทำงานนั้น Function นั้นรวมถึงฟังก์ชันที่ถูกเรียกใช้จากในฟังก์ชันไฟล์นั้นอีกทีหนึ่งนั้นจะต้องอยู่ใน Search Path โดย Search Path นี้หมายถึงกลุ่มของ directory ที่เราได้รวบรวมและบอก MATLAB ว่าจะเป็นกลุ่มของ directory ที่ MATLAB จะต้องค้นหาในการติดตั้ง MATLAB ไฟล์และ Toolbox ที่ได้รับการติดตั้งเข้าไปจะถูกรวมเข้าไปอยู่ใน Search Path นี้โดยอัตโนมัติอยู่แล้ว หากเราต้องการที่จะเพิ่ม Search Path เราสามารถที่จะทำได้โดย เลือกคำสั่ง Set Pathภายใต้ File Menu เพื่อเพิ่ม directory ที่เราต้องการเข้าไปอยู่ใน Search Pathหากเราต้องการที่จะดูว่ามี directory ใดบ้างที่อยู่ใน Search Path เราสามารถใช้คำสั่ง **path** ที่MATLAB Command Prompt เช่น เมื่อเราป้อนคำสั่ง

» path

เราจะได้ผลเป็น

MATLABPATH

C:*MATLABR*12*toolbox**MATLAB**general*

C:\MATLABR12\toolbox\MATLAB\ops

……..

C:\MATLABR12\toolbox\wavelet\wavedemo

C:\MATLABR12\toolbox\rtw\targets\xpc\target\kernel\embedded

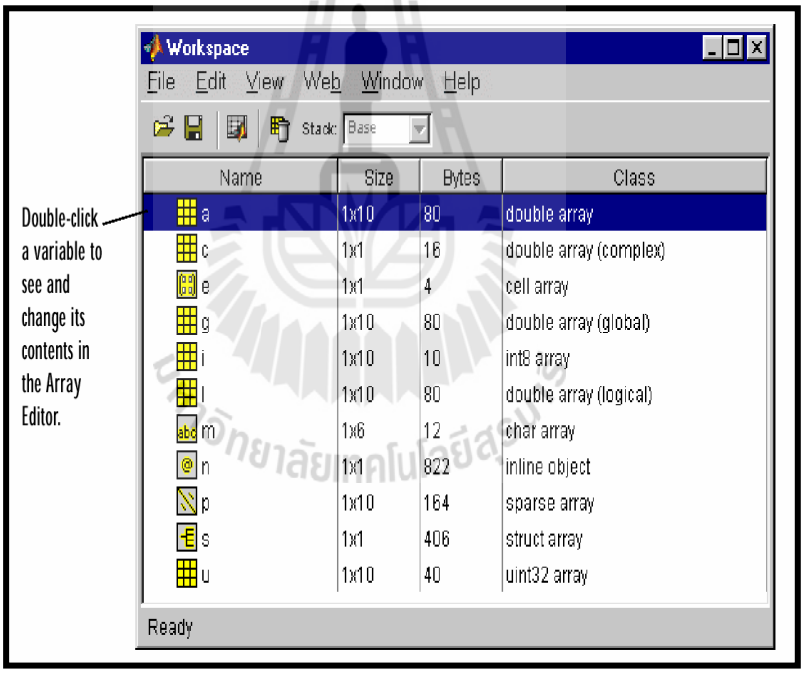
C:\MATLABR12\work

ซึ่งเป็นการแสดง Search Part ทั้งหมดที่เราได้กำหนดให้ MATLAB ในขณะนั้น และเรายังสามารถใช้คำสั่ง **addpath** เพื่อเพิ่ม Search Path หรือใช้ **rmpath** เพื่อตัด directory นั้นออกจาก Search Path ได้อีกด้วย

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.3 Workspace Browser

เมื่อเราได้มีการสร้างค่าตัวแปรหรือพารามิเตอร์ขึ้นใน MATLAB ค่าเหล่านั้นจะถูกเก็บไว้ในพื้นที่การทำงาน (Workspace) และหน่วยความจำของ MATLAB เราจะเพิ่มตัวแปรลงในพื้นที่ทำงานได้ด้วยการใช้คำสั่ง ให้ M-file ทำงานหรือ load ค่าที่บันทึกไว้เข้าสู่พื้นที่ทำงาน เพื่อที่จะดูว่าในขณะนั้นมีตัวแปรอะไรบ้างที่มีอยู่ในพื้นที่ทำงาน ใน MATLAB 6.0 นี้เราสามารถใช้ Workspace Browser ซึ่งเป็นหน้าต่างเครื่องมือหนึ่งใน Desktop Tool หรือในทุก version ของ MATLAB เราอาจใช้คำสั่ง who หรือ whos ที่Command Windows ก็ได้ Workspace Browser จะมีลักษณะโดยทั่วๆ ตามรูปต่อไปนี้

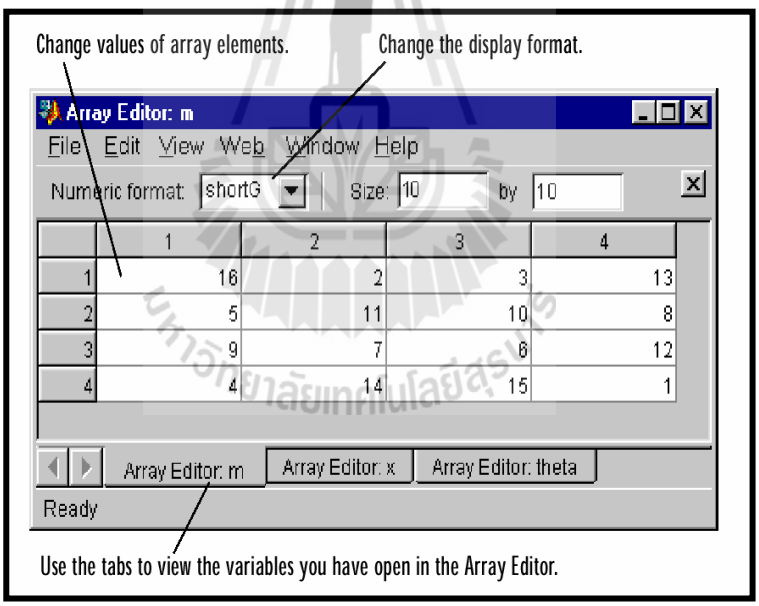


รูปที่ ก.6 หน้าต่างพื้นที่การทำงาน (Workspace Browser)

ที่ Workspace Browser นี้เราสามารถที่จะดูว่ามีตัวแปรหรือ array ตัวใดที่อยู่ใน workspace บ้างนอกจากนี้สำหรับตัวแปรแต่ละตัวก็จะมีข้อมูลที่บอกว่าตัวแปรแต่ละตัวนั้นเป็นประเภทใด มีขนาดเท่าใด ใช้หน่วยความจำมากเท่าใดอีกด้วย สำหรับตัวแปรแต่ละตัวที่ปรากฏอยู่ในรายการภายใต้work space Browser นี้เราสามารถที่จะลบมันออกจาก Workspace ได้สองวิธีคือ

- ด้วยการเลือกที่ตัวแปรตัวนั้น กดเมาส์ปุ่มขวาแล้วเลือก Delete Selection
- ด้วยการเลือกตัวแปรตัวนั้นแล้วกดปุ่ม ที่แถบเครื่องมือ เพื่อลบตัวแปรนี้ ออกไปเรายังสามารถที่จะแก้ไขข้อมูลของตัวแปรบางประเภทได้ เช่นตัวแปรที่เป็น array ซึ่งมีลักษณะเป็นเมตริกซ์ (รายละเอียดของเรื่องตัวแปรประเภทต่างๆ เราจะได้กล่าวถึงในบทต่อไป) เราสามารถที่จะแก้ไขที่ cell ใด cell หนึ่งเป็นการเฉพาะได้ โดยการเปิดตัวแปรนั้นขึ้นมา ซึ่งการเปิดตัวแปรนั้นสามารถทำได้หลายวิธีคือ
  - กดเมาส์สองครั้งที่ตัวแปรนั้น
  - เลือกตัวแปรนั้นกดเมาส์ปุ่มขวาแล้วเลือก Open Selection
  - เลือกตัวแปรนั้นแล้วกดปุ่ม ที่แถบเครื่องมือ

เมื่อเราทำการเปิดตัวแปรนั้นแล้ว เราจะได้หน้าต่างใหม่ขึ้นมา ซึ่งเราเรียกว่า Array Editor จะมีลักษณะคล้ายหน้าต่างของโปรแกรมประเภท space sheet คือมีลักษณะเป็น Matrix ซึ่งขนาดของ Matrixนั้นก็จะขึ้นอยู่กับขนาดของตัวแปร ลักษณะของ array editor จะมีลักษณะดังรูปต่อไปนี้



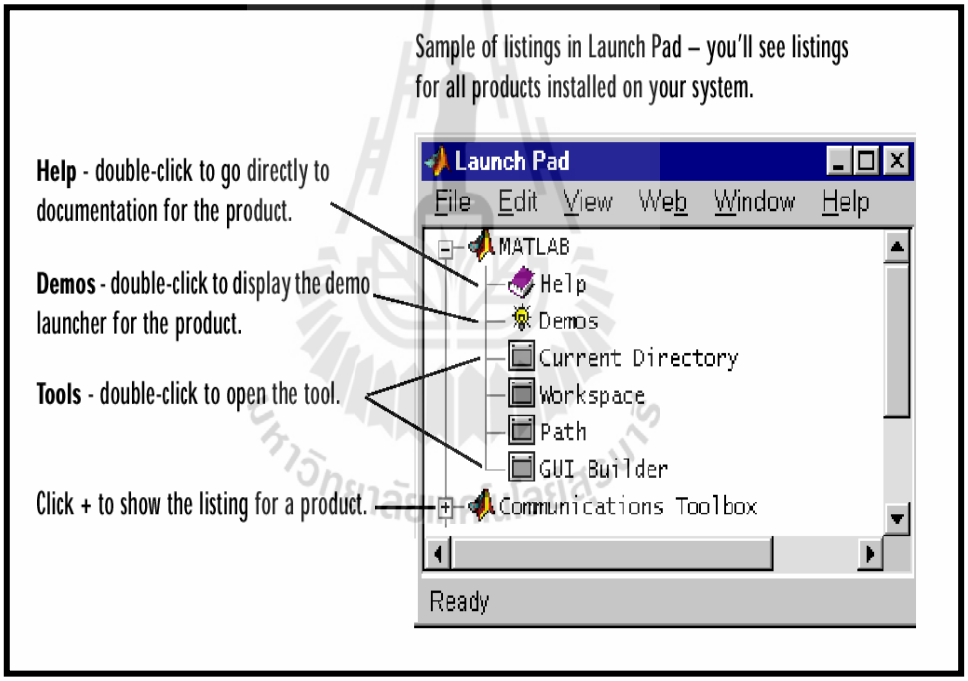
รูปที่ ก.7 หน้าต่าง Array Editor

บน Array Editor เราจะทราบขนาดของตัวแปร รูปแบบการแสดงผล (จะกล่าวถึงในหัวข้อต่อไป)และเราสามารถที่จะปรับเปลี่ยนค่า cell แต่ละ cell ได้อย่างอิสระ และเราสามารถใช้ Array Editor นี้แสดงผลตัวแปรได้หลายๆตัวพร้อมกัน โดยเลือกจะให้แสดงผลตัวใดได้โดยใช้แถบเลือกข้างล่างนอกจากนี้เรายังสามารถที่จะปรับเปลี่ยนขนาดของ array ได้ โดยการเปลี่ยนค่า size ให้เป็นไปตามต้องการ หากว่าเราทำการเพิ่มขนาด cell ที่เราไม่ได้กำหนดค่าจะมีค่าเท่ากับศูนย์ ข้อควร

ระวังคือการลดขนาดตัวแปรจะทำให้มีค่าของตัวแปรบางส่วนหายไป และเราไม่สามารถที่จะ undo เพื่อเรียกข้อมูลคืนมาได้

3.4 Launch Pad

Launch Pad เป็นหน้าต่างที่แสดง toolbox ต่างๆที่เราได้ติดตั้งไว้ในเครื่องของเรา และทำให้เราสามารถที่จะเข้าสู่ เครื่องมือ ตัวอย่าง และเอกสารที่เกี่ยวข้องกับ MATLAB หรือ Toolbox ต่างๆ ได้โดยง่ายลักษณะของ Launch Pad ก็จะเหมือนกับการแสดง file ใน Windows Explorer คือสามารถที่จะขยายหรือลดการแสดงรายละเอียดใน Toolbox ต่างๆ ได้ และเมื่อเรากดเมาส์สองครั้งในหัวข้อที่ต้องการเราก็จะได้เห็นตัวอย่าง หรือเอกสารที่เกี่ยวข้องกับหัวข้อนั้นได้ทันที โดยไม่ต้องเสียเวลาในการค้นหา ลักษณะของ Launch Pad จะเป็นตามรูปต่อไปนี้



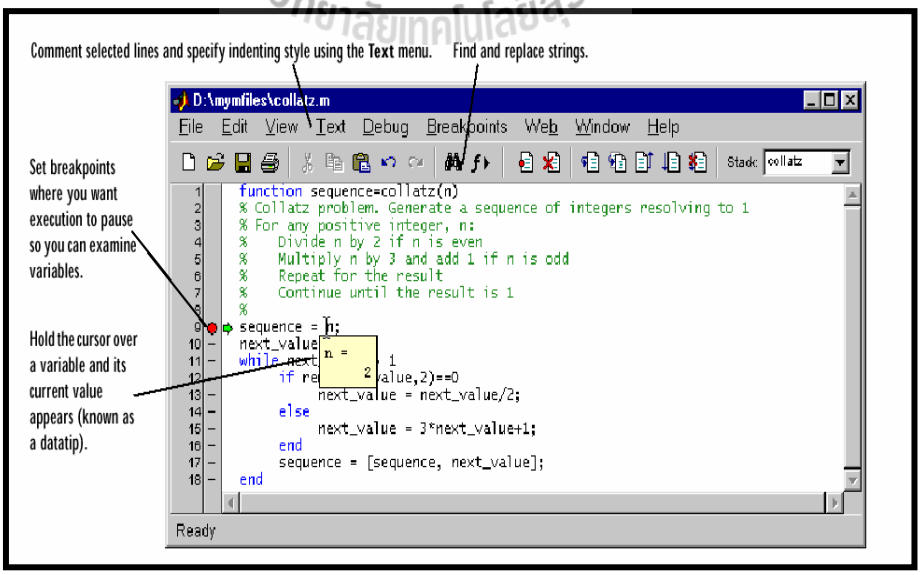
รูปที่ ก.8 หน้าต่าง Launch Pad

สำหรับหน้าต่างทั้ง 5 ที่เปิดขึ้นมาพร้อมกับ MATLAB Desktop นี้เราสามารถที่จะนำออกมาจากDesktop ได้หรือที่เรียกว่า undock โดยการกดที่ปุ่ม ที่อยู่ที่ขอบบนขวาของหน้าต่างนั้นๆได้ หรือเราอาจจะปิดหน้าต่างนั้นไปเลยก็ได้ ยกเว้น (Command Windows) และถ้าเราต้องการจะนำหน้าต่างนี้กลับมาอยู่ที่ Desktop เราก็สามารถที่จะเรียกมันกลับมาได้โดยการใช้คำสั่ง **Dock .....** ภายใต้เมนู **View** ของหน้าต่างนั้นเมื่ออยู่นอก desktopนอกจากนี้เรายังสามารถที่จะปรับเปลี่ยนรูปร่างของ

Desktop ของเราได้โดยการใช้คำสั่งหลายคำสั่งที่อยู่ภายใต้เมนู **View** ซึ่งเราจะไม่ขอกล่าวในรายละเอียดในที่นี้ แต่หากได้มีการทดลองใช้คำสั่งต่างๆ ภายใต้เมนูนี้ เราเชื่อว่าผู้ใช้คงจะสามารถเข้าใจในคำสั่งต่างๆ ได้โดยไม่ยากนัก นอกเหนือจากนั้นเรายังสามารถที่จะตั้งค่าต่างๆ ของ Desktop นี้ได้โดยการใช้คำสั่ง **Preference** ภายใต้เมนู **File** ซึ่งจะสามารถปรับค่าต่างๆ เช่นแบบตัวหนังสือ สีที่แสดง และอื่นๆได้อีกมาก รายละเอียดเหล่านี้ บางส่วนเราจะกล่าวถึงในบทต่อๆไป

3.5 หน้าต่าง Editor/Debugger

ในการเขียนโปรแกรมหรือที่เรียกว่า M-file จะเขียนด้วย Text Editor ธรรมดาเช่น Notepad ก็ได้เพราะ M-file จะเป็นโปรแกรมที่ใช้ตัวอักษรในลักษณะ ASCII Code ธรรมดา และสำหรับ MATLAB version 5 เป็นต้นมานั้นจะมี editor มาพร้อมกับ MATLAB ด้วยเลยทำให้สะดวกในการใช้งานเป็นอย่างมากเพราะนอกจากจะเป็น editor แล้วยังมี debugger เพื่อช่วยในการแก้ไขโปรแกรมพร้อมอยู่ด้วยเราสามารถที่จะใช้ Editor/Debugger เพื่อสร้างและแก้ไข M-files ซึ่งเป็นการเขียนโปรแกรมที่จะเรียกชุดคำสั่งหรือฟังก์ชันต่างๆของ MATLAB ขึ้นมาทำงาน Editor/Debugger นี้จะทำหน้าที่เป็นทั้ง text editor เพื่อใช้ในการเขียนโปรแกรม และทำหน้าที่เป็น Debugger คือมีเครื่องมือที่ช่วยในการแก้ไขโปรแกรมกรณีที่โปรแกรมเกิดความผิดพลาดขึ้นฟังก์ชันหรือคำสั่งที่เราเขียนขึ้นเพื่อใช้กับ MATLABนั้นเราจะเรียกว่า MATLAB file หรือเพื่อความสะดวกเรานิยมที่จะย่อแล้วเรียกกันว่า M-file ลักษณะของEditor/Debugger จะมีลักษณะตามรูปต่อไปนี้

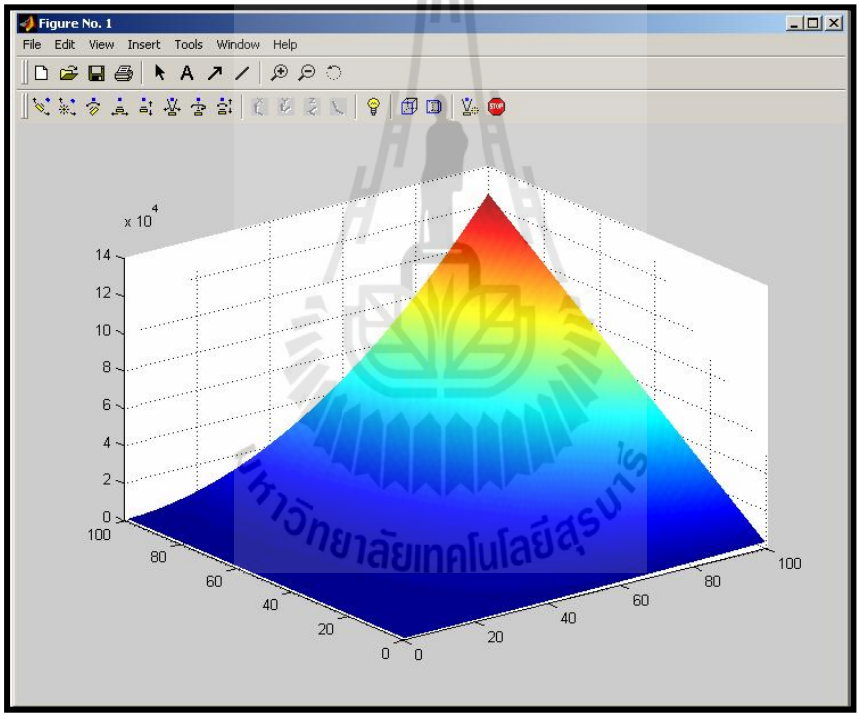


รูปที่ ก.9 หน้าต่าง Editor/Debugger

ซึ่งบน Editor/Debugger นี้จะมีเครื่องมือหลายอย่างช่วยเหลือให้เราสามารถที่จะเขียนโปรแกรมได้สะดวกขึ้น อย่างไรก็ตามรายละเอียดในการใช้ Editor/Debugger จะกล่าวถึงในบทที่เกี่ยวข้องกับการเขียน M-file

3.6 หน้าต่างแสดงรูปภาพ Graphic Windows

เมื่อได้รับคำสั่งให้เขียนกราฟ MATLAB จะแสดงผลบน Graphic Windows ซึ่งจะเรียกขึ้นใช้งานโดยอัตโนมัติ Graphic Window นี้อาจจะปรากฏขึ้นมากกว่าหนึ่งหน้าต่างในเวลาเดียวกันก็ได้ แล้วแต่คำสั่งที่กำหนดให้กับ MATLAB ซึ่งบนหน้าต่างนี้จะมี Menu Bar และอื่นๆอยู่ด้วย ลักษณะของ Graphic Windowจะมีลักษณะตามรูปต่อไปนี้



รูปที่ ก.10 หน้าต่าง Graphic Window

ซึ่งในหน้าต่างนี้นอกจากจะใช้แสดงผลรูปภาพแล้ว เรายังสามารถใช้สร้าง Graphical User Interface เพื่อใช้ทำโปรแกรมที่มีการติดต่อกับผู้ใช้โดยใช้ปุ่มต่างๆ เหมือนกับโปรแกรมที่ทำงานภายใต้windows ทั่วไปได้อีกด้วย นอกเหนือจากนี้ Graphic Window ยังมีเครื่องมือที่ช่วยในการเขียนกราฟของเราสะดวกขึ้น โดยเฉพาะใน version หลังๆ ของ MATLAB นี้ เราสามารถที่จะแก้ไขเพิ่มเติมกราฟที่เราที่Graphic Window ได้ ไม่ว่าจะเป็นการเพิ่มตัวหนังสือ การเพิ่มเส้น การเพิ่มชื่อ

แกนหรือชื่อกราฟ การปรับเปลี่ยนมุมมอง ปรับเปลี่ยนทิศทางของแสงที่ฉายมาที่รูป และอื่นๆอีกมาก และใน version 6 นี้เราสามารถที่จะทำการ fit curve ที่เราเขียนลงไปบน Graphic Window นี้ได้อีกด้วย สำหรับรายละเอียดในการใช้เครื่องมือต่างๆ บน Graphic Window เราจะกล่าวถึงในบทที่เกี่ยวข้องกับการใช้ Graphic ต่อไป

4. คำสั่งเบื้องต้น

ในบทต่อไปเราจะได้ทราบถึงคำสั่งต่างๆที่มีอยู่มากมายใน MATLAB อย่างไรก็ตามในส่วนนี้เราขอกล่าวถึงคำสั่งพื้นฐานเบื้องต้นและการใช้แป้นพิมพ์บางส่วนที่ใช้เป็นพื้นฐานในการทำงานภายใต้MATLAB สำหรับ คำสั่งต่างๆที่เราจะกล่าวถึงต่อไปในที่นี้นั้นจะเป็นคำสั่งตัวหนังสือที่เราจะใช้กับMATLAB โดยผ่าน Command Window ซึ่งคำสั่งพื้นฐานจะประกอบด้วยคำสั่งต่อไปนี้ แต่ก่อนอื่นเนื่องจากเอกสารนี้เขียนขึ้นโดยมุ่งหวังว่าผู้อ่านจะศึกษาการใช้งานขั้นพื้นฐานของ MATLAB ด้วยตัวเอง ดังนั้นจึงขอทำความเข้าใจถึงรูปแบบตัวอักษรและสัญลักษณ์ต่างๆที่ใช้ในเอกสารนี้ก่อน

4.1 รูปแบบและสัญลักษณ์ในเอกสาร

เนื่องจากในเอกสารชุดนี้จำเป็นต้องมีการยกตัวอย่างอยู่เสมอเพื่ออธิบายการใช้งาน MATLABดังนั้นเพื่อไม่ให้ผู้อ่านเกิดการสับสน ผู้เรียบเรียงได้ใช้รูปแบบของตัวหนังสือให้มีความแตกต่างกันโดยมีการใช้สัญลักษณ์ดังนี้

- ตัวหนังสือที่ใช้อธิบายปกติจะใช้ตัวหนังสือแบบ Angsana New Thai ทั้งภาษาไทยและภาษาอังกฤษ
- สำหรับคำสั่งที่เป็น input เพื่อให้ MATLAB คำนวณจะใช้ตัวหนังสือแบบ **Courier New** ตัวหนา เช่น **Airtime = B*C(1,2)**
- สำหรับ output ที่ได้จาก MATLAB จะใช้ตัวหนังสือแบบ Courier New แบบตัวเอียง *Airtime = 20*
- เครื่องหมาย % เป็นเครื่องหมาย comment ซึ่ง MATLAB จะไม่สนใจที่จะทำคำสั่งหรือข้อความต่างๆที่อยู่หลังเครื่องหมายนี้แต่มีประโยชน์ในการเขียนโปรแกรมเพราะจะสามารถบันทึกข้อความต่างๆลงไปได้
- เครื่องหมาย C แสดงถึงว่าเป็นข้อควรระวังหรือควรระลึกถึงในการใช้ MATLAB
- เครื่องหมาย ↲ หมายถึงว่าให้กด Enter key หลังจากจบคำสั่ง (ในบางกรณีที่ต้องกด Enter key อย่างชัดเจนอยู่แล้วอาจยกเว้นไม่ใส่เครื่องหมายนี้)

4.2 คำสั่งเบื้องต้นที่ควรทราบในการใช้ MATLAB

คำสั่งเบื้องต้นประกอบด้วยคำสั่งต่อไปนี้

- Quit หรือ exit เลิกการทำงานของ MATLAB
- clc ลบข้อความที่บรรจุอยู่ใน Command Window แต่ไม่ลบค่าตัวแปรใดๆ
- clf ลบรูปภาพที่บรรจุอยู่ใน Graphic Window
- clear ลบตัวแปรทุกตัวออกจากหน่วยความจำ
- save เป็นการรวบรวมค่าตัวแปรทุกตัวที่มีอยู่ในขณะนั้นบันทึกลงบน disk
- หากต้องการยกเลิกการคำนวณในขณะที่ MATLAB ยังทำการคำนวณไม่เรียบร้อย ให้กดแป้น Ctrl และ c พร้อมกัน

C สำหรับ MATLAB การใช้คำสั่งหรือ function ใดๆก็ตามจะต้องใช้อักษรตัวพิพม์เล็กเสมอ ไม่เช่นนั้นท่านจะได้ข้อความแสดงความผิดพลาด

4.3 Function Keys พิเศษ

ใน MATLAB มีการสำรองแป้นไว้สำหรับช่วยในการใช้งานให้ง่ายขึ้น โดยจะประกอบด้วย keysพิเศษต่างๆต่อไปนี้

Ctrl-p หรือ Z ใช้เรียกคำสั่งที่ทำไปในครั้งที่ผ่านมา

Ctrl-n หรือ b ใช้เรียกคำสั่งที่สั่งหลังจากคำสั่งที่กำลังสั่งอยู่

Ctrl-f หรือ Y เลื่อนไปทางขวา 1 ตัวอักษร

Ctrl-b หรือ a เลื่อนไปทางซ้าย 1 ตัวอักษร

Del หรือ **Backspace** ลบตัวอักษรครั้งละ 1 ตัว

Ctrl-l หรือ **Ctrl**- a เลื่อนไปทางซ้าย 1 คำ

Ctrl-r หรือ **Ctrl**- Y เลื่อนไปทางขวา 1คำ

Ctrl-a หรือแป้น **Home** เลื่อนไปที่ตัวอักษรแรกของบรรทัด

แป้น **End** เลื่อนไปที่ตัวอักษรสุดท้ายของบรรทัด

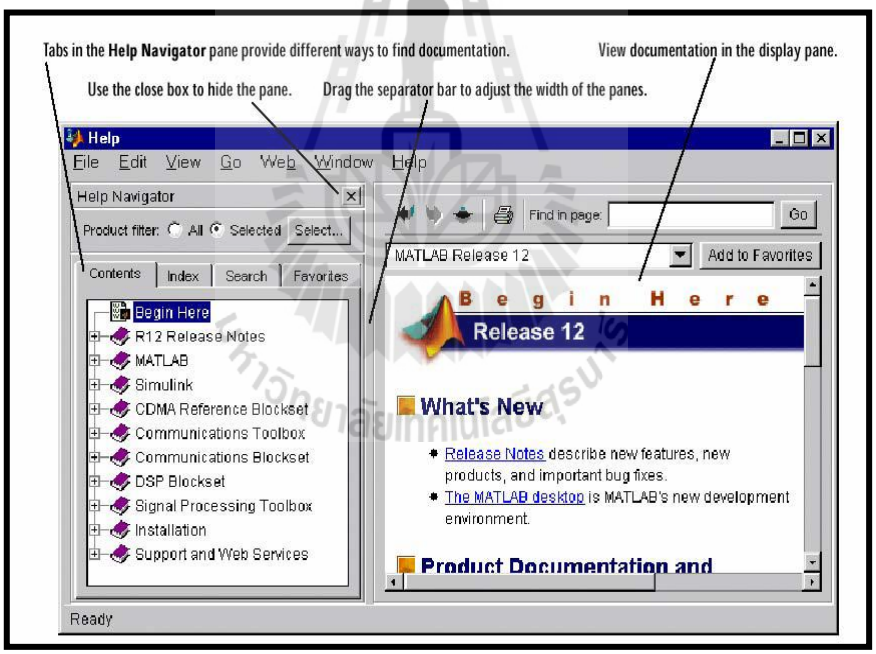
Ctrl-k ลบทุกตัวอักษรจากจุดที่อยู่ไปถึงตัวสุดท้ายของบรรทัด

4.4 การขอความช่วยเหลือในการใช้โปรแกรม

ถ้าในระหว่างการทำงานบน MATLAB แล้วมีปัญหาเกิดขึ้นในเรื่องของรูปแบบของการใช้คำสั่งเราอาจใช้คำสั่ง help เพื่อช่วยการทำงานได้

- ใช้ **help** อย่างเดียวเพื่อให้ MATLAB แสดงชื่อ function ที่มีบรรจุอยู่ใน help
- ใช้ **help** ตามด้วยชื่อ function เพื่อให้ MATLAB แสดงรายละเอียด function นั้น
- หากต้องการดูการสาธิตการทำงานของ MATLAB ให้ใช้คำสั่ง **demo**
- สำหรับการร้องขอ help อาจจะเรียกใช้จาก Menu bar ก็ได้
- Help Windows ซึ่งจะสะดวกในการค้นหามากขึ้น คำสั่งที่ใช้คือ **help win**

นอกเหนือจากนั้นใน version ใหม่นี้ยังได้มี Help Browser ที่สร้างขึ้นมาใน MATLAB เพื่อทำให้เราสามารถที่จะใช้เพื่อค้นหาและดูเอกสารที่เกี่ยวข้องกับ MATLAB ที่ผลิตขึ้นโดยบริษัท MathWorks ซึ่ง Help Browser นี้เป็นการนำ MATLAB Desktop นี้รวมเข้ากับ Web Browser เพื่อที่จะแสดงผลเอกสารประเภท HTML ซึ่งการทำงานของ Help Browser ก็จะเหมือนกับโปรแกรม Web Browser ทั่วๆไปนั่นเอง ลักษณะของ Help Browser จะมีลักษณะตามรูป



รูปที่ ก.11 หน้าต่างขอความช่วยเหลือ(Help Browser)

สำหรับรายละเอียดของการใช้ Help ภายใต้ Help Browser นั้นสามารถที่จะแยกเป็นส่วนการแสดงข้อมูลที่ช่วยในการใช้ MATLAB ออกเป็นส่วนๆได้หลายส่วน และเนื่องจาก Help นี้อยู่ในรูปแบบของ html ดังนั้นการใช้งานก็จะมีลักษณะเหมือนการใช้ Browser ทั่วๆไป เพียงแต่ข้อมูลทั้งหมดนั้นอยู่ภายในเครื่องของเรา (ถ้าเราเลือก install แบบนั้น) และเนื่องจากการจัดรูปของ Help อยู่

ในลักษณะของ browserเราจึงสามารถที่จะลงลึกเข้าไปในข้อมูลได้เรื่อๆ อย่างไรก็ตามหัวข้อที่สำคัญภายใต้หน้า help หน้าแรกที่จะพาเราท่องไปที่ส่วนต่างๆ จะมีที่สำคัญอยู่ดังนี้

ภายใต้ Using MATLAB จะมีข้อมูลที่จัดกลุ่มเป็นส่วนประกอบที่สำคัญ ต่างๆ ดังนี้

- "Development Environment" เป็นข้อมูลเกี่ยวกับการใช้ MATLAB desktop
- "Mathematics" อธิบายถึงขีดความสามารถในการทำงานด้านคณิตศาสตร์และสถิติของMATLAB
- "Programming and Data Types" อธิบายถึงวิธีการสร้างฟังก์ชันและการใช้ภาษา MATLAB
- "Graphics" อธิบายการเขียนกราฟและความสามารถของการทำงานด้านกราฟฟิกส์ของMATLAB
- "3-D Visualization" เป็นการแนะนำการใช้คำสั่ง view เพื่อกำหนดมุมมอง คำสั่ง lighting เพื่อกำหนดทิศทางและขนาดแสงในการมองภาพ 3 มิติ และการกำหนดลักษณะภาพสามมิติอื่นๆ เพื่อให้ได้ภาพที่ซับซ้อนออกมาในลักษณะที่เราต้องการ
- "External Interfaces/API" อธิบายวิธีการติดต่อกับโปรแกรมที่เขียนขึ้นโดยภาษา C และFortran รวมถึงการติต่อกับภาษา Java, ไฟล์ข้อมูล, serial port I/O, ActiveX และ DDE เป็นจ้น
- "Creating Graphical User Interfaces" อธิบายถึงวิธีการใช้เครื่องมือและการสร้างโปรแกรมMATLAB ประเภท graphical user interface

ภายใต้ Reference ได้มีการจัดเอกสารต่างๆ ดังนี้

- "MATLAB Function Reference" อธิบายถึงฟังก์ชันที่สำคัญของ MATLAB ทั้งหมด โดยจะบอกถึงวิธีการป้อนข้อมูลให้กับฟังก์ชัน การกำหนด syntax และระเบียบวิธีการทางคณิตศาสตร์ที่ฟังก์ชันนั้นใช้ เราสามารถที่จะเลือกวิธีการนำเสนอได้ทั้งให้แสดงฟังก์ชันแบบ "Function By Category" คือเรียงตามหมวดหมู่ของฟังก์ชันนั้น หรือ "Alphabetical List of Functions" คือเรียงตามลำดับอักษร
- "External Interfaces/API Reference" อธิบายฟังก์ชันของ MATLAB ที่เกี่ยวข้องกับการเชื่อมต่อกับภายนอก วิธีการเรียกโปรแกรมต่างๆ ค่าที่จะได้จากการคำนวณและตัวอย่างต่างๆที่เกี่ยวข้อง

นอกจากนี้ ยังรวม ซึ่งทำให้เราสามารถที่จะเข้าถึงคุณสมบัติของวัตถุที่เป็นรูปภาพต่างๆ

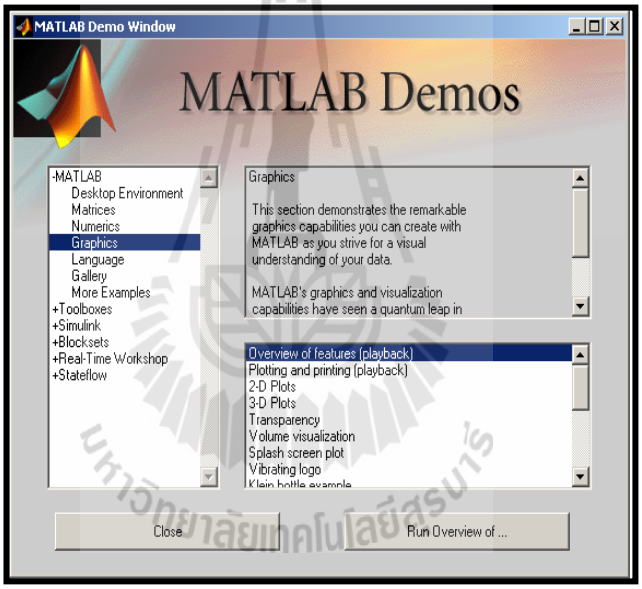
มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้โดยง่าย รายละเอียดเกี่ยวกับกราฟฟิกส์ของ MATLAB นั้นมีค่อนข้างมาก ซึ่งเราอาจต้องทำความเข้าในในเบื้องต้นเสียก่อน จึงจะสามารถใช้งานในส่วนนี้ได้อย่างจริงจัง MATLAB Demos จะเป็นส่วนที่แสดงตัวอย่างทั้งหมดที่มากับโปรแกรม จำนวนตัวอย่างที่แสดงนั้นก็จะขึ้นอยู่กับจำนวน Toolbox ที่เราได้ติดตั้งไว้บนเครื่องของเราด้วย ตัวอย่างเหล่านี้จะช่วยให้เราเข้าใจได้ว่าเราสามารถที่จะใช้ MATLAB ทำงานอะไรให้กับเราได้บ้าง

การเรียกดูตัวอย่างที่มีใน MATLAB สามารถทำได้โดยสั่ง

»demo

หลังจากนั้นก็จะปรากฏหน้าต่าง demo ขึ้นมีลักษณะดังรูปข้างล่างนี้ เราสามารถที่จะเลือก หัวข้อที่ต้องการจะดูตัวอย่างจากหน้าต่างย่อยทางซ้ายมือ จากนั้นจึงเลือกตัวอย่างทางหน้าต่างขวามืออีกครั้งหนึ่งแล้วจึงกดปุ่มเลือก Run..... เพื่อดูตัวอย่างการทำงานของ MATLAB ต่อไป



รูปที่ ก.12 หน้าต่าง MATLAB Demos

5. การใช้ MATLAB เบื้องต้น

5.1 แนะนำ Scalars, Vectors and Matrices

ในการแก้ปัญหาทางด้านวิศวกรรม เราพบว่ากลุ่มข้อมูลที่จะต้องพิจารณาอาจจะเป็นตัวเลขเพียงตัวเดียว หรือ scalar หรืออาจเป็นกลุ่มของข้อมูลที่จัดอยู่ในรูปของ vector หรือ matrix ก็ได้ สำหรับMATLAB แล้วโครงสร้างข้อมูลทั้งหมดจะได้รับการจัดให้อยู่ในรูปของ matrix ทั้งหมดคือ

- จะถือว่า scalar คือ matrix ขนาด [1 x 1]

- ส่วน vector สามารถจะจัดอยู่ในรูป row matrix [n x 1] หรือ column matrix [1 x n] ซึ่งนิยมเรียกอีกอย่างหนึ่งว่า row vector และ column vector ตามลำดับการตั้งชื่อตัวแปรใน MATLAB มีเงื่อนไขต่อไปนี้
- ชื่อตัวแปรต้องขึ้นต้นด้วยตัวอักษร
- ชื่อตัวแปรจะบรรจุด้วยตัวอักษร ตัวเลข และ underscore (_) ได้
- ชื่อตัวแปรจะยาวเพียงใดก็ได้แต่ MATLAB จะจดจำชื่อตัวแปรและแยกตัวแปรสองตัวออกจากกันด้วยอักษรเพียง 19 ตัวแรกเท่านั้น
- MATLAB เป็น case-sensitive variable ดังนั้นการใช้ตัวอักษรตัวใหญ่และตัวเล็กจะเก็บไว้ในหน่วยความจำต่างกัน เช่นตัวแปร xA, xa, XA จะเป็นตัวแปรคนละตัวกัน

***** ควรตั้งชื่อตัวแปรให้เหมาะสมเพื่อช่วยให้จำได้ว่าตัวแปรเหล่านั้นแทนค่าของอะไร เพื่อไม่ให้เกิดการสับสนเมื่อต้องการที่จะแก้ไขโปรแกรม**

1. การกำหนดค่าตัวแปร

การกำหนดค่าตัวแปร หรือ assignment statement จะใช้เครื่องหมาย '=' ซึ่งมีส่วนประกอบดังนี้

ตัวแปร = ค่าของตัวแปร

- ถ้าค่าของตัวแปรเป็นตัวเลขตัวเดียว อาจกำหนดเป็นตัวเลขไปเลยก็ได้ เช่น ถ้าต้องการให้ตัวแปร A มีค่า 1 **A** = 1
- ถ้าค่าของตัวแปรเป็น vector หรือ matrix ต้องกำหนดในเครื่องหมายวงเล็บใหญ่ [ ]
  - ★ ค่าที่จะกำหนดในลักษณะของแถวนอน (row) ค่าแต่ละ elements ใน row เดียวกันอาจจะแยกจากกันด้วยช่องว่าง (space) หรือ comma ก็ได้ เช่น

    A = [1, 2, 3]
  - ★ ใช้เครื่องหมาย semi-colon แยก row ออกจากกัน

    A = [1 2 ; 3 4]
  - ★ การแยกแต่ละ row ออกจากกันสามารถกระทำได้อีกวิธีหนึ่งคือแยกบรรทัด ด้วยการใช้ enter

    a = [1, 2 ↵

    3, 4]

- ในกรณีที่ row ยาวมาก จนต้องเขียนต่อบรรทัดใหม่ให้ใช้สัญลักษณ์ ellipsis คือ comma แล้วตามด้วยจุด 3 จุด เช่น

  a = [1, 2, 3 ,... ↵
  4 ,5, 6]

- MATLAB ยินยอมให้มีการกำหนดค่า matrix ใหม่โดยใช้ matrix เดิมเข้าช่วย
- โดยทั่วไป เมื่อมีการกำหนดค่าตัวแปร MATLAB จะมีการแสดงข้อมูลกลับ (echo) เพื่อให้ผู้ใช้รับทราบการยืนยันข้อมูลที่ส่งเข้าสู่หน่วยความจำ

***** ในการหัดใช้ MATLAB ในครั้งแรกๆท่านอาจจะไม่ชำนาญจึงอาจทำให้การป้อนข้อมูลหรือคำสั่งผิดพลาดได้ ดังนั้นจึงควรให้เครื่อง echo ค่ากลับมาก่อน เมื่อท่านชำนาญแล้วจึงงดใช้ echo**

- เราสามารถเปลี่ยนค่า element ของ matrix ได้โดยใช้ index กำหนดค่านั้น ๆ
- ถ้า matrix เป็น single row หรือ single column matrix อาจใช้ index ตัวเดียวได้
- ถ้า `a = [1 2 3]` แล้วเรากำหนด `a(7) = 5` เราจะได้ `a` มีค่าเป็น

  `a = [1 2 0 0 0 0 5]`
- ซึ่ง MATLAB จะเพิ่ม element จาก 3 เป็น 7 ทันที แล้วกำหนดค่า element ที่เพิ่มขึ้นโดยไม่มีการกำหนดค่าให้เป็นศูนย์โดยอัตโนมัติ
- การดูค่าตัวแปรที่บรรจุอยู่ในความจำของ MATLAB อาจใช้ คำสั่งได้ 3 คำสั่งคือ

  who จะบอกชื่อตัวแปร ทั้งหมดที่อยู่ในหน่วยความจำ

  whos จะบอกทั้งขนาด , จำนวน byte ที่ใช้ในหน่วยความจำ และ ชนิดของตัวแปรแต่ละตัวที่มีอยู่ทั้งหมดในขณะนั้น

  size จะบอกขนาดของตัวแปรที่เป็นทั้ง matrix และ array เช่นถ้า

  $$a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

  จะได้

  size(a) ans = 2 2

2. Colon operator

Colon operator เป็น operator ที่มีประโยชน์ในการกำหนดค่า matrix มาก โดยมีหน้าที่สำคัญดังต่อไปนี้

- เราสามารถกำหนดค่า vector จาก matrix ที่มีการกำหนดค่าเรียบร้อยแล้ว

- Colon operator สามารถใช้สร้าง matrix ใหม่ขึ้นมา เช่น ถ้าใช้ colon แยกจำนวนเต็ม 2 จำนวน เช่น k = 1:5 หมายถึงการสร้างให้ k มีค่าเป็น row vector มีค่าเป็นหรือเหมือนกับการใช้คำสั่ง k = [1, 2, 3, 4, 5]
- ถ้าใช้ colon แยกจำนวน 3 จำนวนหมายถึงให้มีค่าเริ่มจากค่าแรกเพิ่มขึ้นครั้งละเท่ากับค่าที่ 2และสิ้นสุดเมื่อค่าเท่ากับค่าที่ 3
- สามารถใช้ colon ในการนับถอยหลังได้
- สามารถใช้ colon operator แยก sub matrix ได้
- ถ้าใช้ a(:) จะได้ matrix column เดียวที่เริ่มต้นด้วย column ที่ 1 ของ a ต่อด้วย column ที่ 2, 3...ต่อไป
- Matrix ว่าง คือ matrix ที่มีขนาด 0 x 0 เช่น a = [ ]; หรือ b = 10:1:0; ซึ่ง matrix ว่างนี้แตกต่างจาก matrix ที่มี element ทุกค่าเป็นศูนย์
- Transpose Matrix คือการเปลี่ยน row เป็น column และ column เป็น row
- การเก็บค่าตัวแปร ตัวแปร 1 ตัว จะสามารถเก็บค่าได้เพียง 1 ค่า และจะเก็บค่าสุดท้ายเท่านั้น

3. ค่าพิเศษของ MATLAB

ค่าต่อไปนี้เป็นค่าที่กำหนดขึ้นพิเศษ เป็นค่าที่กำหนดไว้แต่ครั้งแรกเมื่อเรียก MATLAB ขึ้นมาใช้และไม่ควรจะมีการใช้ชื่อตัวแปรเหล่านี้กำหนดเป็นค่าอื่นเพราะจะทำให้สับสนได้ ค่าพิเศษมีดังต่อไปนี้

pi ใช้แทนค่า π ซึ่งมีค่าประมาณ 3.14159

i และ j ใช้แทน imaginary valve ซึ่งมีค่า $\sqrt{-1}$

Inf แทน infinity ส่วนมากจะเกิดจากการหารจำนวนใดๆ ด้วยศูนย์

Nan แทน Not-a-Number แทนค่าที่ทางคณิตศาสตร์ไม่ได้นิยาม (undefined) เช่น 0 ÷ 0

Clock เวลาปัจจุบันที่มีการเรียกค่านี้ใช้ซึ่งเป็นเวลาตามนาฬิกาของ Computer ที่กำลังทำงานอยู่

Date วันใน format วัน-เดือน-ปี เช่น 1-jan-97

eps แทน floating-point precision ของเครื่อง computer ที่กำลังทำงานอยู่

Ans แทนค่าที่คำนวณครั้งหลังสุดที่ไม่มีการกำหนดค่าอื่นใด

4. Matrix พิเศษของ MATLAB

Matrix พิเศษที่สามารถให้ MATLAB สร้างขึ้น ประกอบด้วย matrix หลายแบบด้วยกันคือ

rand(m,n) เป็นการสร้าง matrix ขนาด [m x n] ที่แต่ละ element เป็นค่าสุ่มจากตัวเลขในช่วง 0 – 1 หรือใช้ rand(m) สำหรับ square matrix

hilb(N) เป็นการสร้าง Hilbert matrix คือ matrix ขนาด [N x N] ที่ element มีค่าเท่ากับ

magic(N) เป็นการสร้าง square matrix ขนาด [N x N] จากจำนวนเต็ม 1 จนถึง N2 โดยที่ผลรวมของ element ในแต่ละ row, column และใน diagonal จะมีค่าเท่ากันสำหรับ N ที่มีค่าเท่ากับ 1, 3, 4, 5,…

zeros(m,n) เป็นการสร้าง matrix ขนาด [m x n] ที่มีทุก element เท่ากับศูนย์

ones(m,n) เป็นการสร้าง matrix ขนาด [m x n] ที่มีทุก element มีค่าเท่ากับ 1

eye(m,n) เป็นการสร้าง matrix ขนาด [m x n] ที่มี element ในแนว diagonal เท่ากับ 1นอกนั้นเป็นศูนย์ ซึ่ง matrix นี้ ไม่จำเป็นต้องเป็น square matrixlinspace(a,b,n)เป็นการสร้าง row vector ขนาด 1 x n โดยค่าแรกจะเท่ากับ a และค่าสุดท้ายจะเป็น b โดยค่อยๆ เพิ่มขึ้นเท่าๆ กัน จำนวน n ค่า ถ้าหากไม่กำหนดค่า n ไว้MATLAB จะใช้ค่า n เท่ากับ 100

logspace(a,b,n) เป็นการสร้าง row vector ขนาด 1 x n โดยค่าแรกจะเท่ากับ a และค่าสุดท้ายจะเป็น b โดยค่อยๆ เพิ่มขึ้นตาม logarithmic จำนวน n ค่า หากไม่กำหนดค่า n ไว้MATLAB จะใช้ค่า n เท่ากับ 100

5. String Variable

การกำหนดค่าตัวแปรนอกจะกำหนดเป็นตัวเลขแล้ว เรายังสามารถกำหนดเป็นตัวหนังสือได้อีกด้วย โดยการกำหนดค่าตัวแปรที่เป็นตัวหนังสือหรือที่เรียกว่า string จะต้องใส่ค่าที่เราต้องการกำหนดในเครื่องหมาย quotation “” ดังตัวอย่างเช่น

Str1='Tom';

Str2='Pookie';

การกำหนดค่าตัวแปรเป็น string จะมีประโยชน์ในกรณีที่ต้องส่งค่าตัวแปรที่เป็นตัวหนังสือไปให้ function เพื่อใช้ในการประมวลผลหรือแสดงคำสั่งที่ต้องการ

สำหรับการกำหนดตัวแปรให้เป็น matrix โดยมีค่าแต่ละ element เป็น string นั้นมีข้อจำกัดอย่างหนึ่งคือทุกตัวแปรจะต้องมีความยาวเท่ากัน ซึ่งข้อจำกัดนี้มาจากว่า matrix ต้องมีจำนวน column ในแต่ละrow เท่ากัน ซึ่งจากข้อจำกัดนี้ทำให้เราอาจต้องเพิ่ม blank space มาขั้นในตัวแปรที่เป็น string แต่ละตัว

5.2 Matrix Operation

1. Dot Product

Dot Product หรือ inner product เป็นการคำนวณหาค่า scalar value ของ vector ที่มีขนาดเท่ากัน ซึ่งถ้าA และ B เป็น vector ที่มีขนาดเท่ากันจะได้

$$A \cdot B = \sum_{i=1}^{N} aibi$$

เมื่อ N เป็นจำนวน element ของ A และ B

2. การคูณ Matrix

การที่ Matrix A และ B จะคูณกันได้ matrix ตัวแรกจะต้องมีจำนวน column เท่ากับจำนวน row ของตัวที่สอง และผลจะได้ matrix ที่มีขนาดเท่ากับ row ของตัวแรก และ column ของตัวหลัง เช่น A มีขนาด [m x n] และ B มีขนาด [p x q] AB จะมีค่าก็ต่อเมื่อ n = p และจะได้ matrix ขนาด m x q ดังนั้นโดยทั่วไป AB•BA หรือไม่มีการสลับที่การคูณของ matrix สำหรับ MATLAB ถ้า C = AB จะใช้C = A * B ถ้าจำนวน column A ไม่เท่ากับจำนวน row ของ B MATLAB จะแสดงข้อความบอกความผิดพลาด

3. การคูณ Matrix ด้วย Scalar

การคูณ matrix ด้วย scalar นั้นจะเป็นการคูณทุก element ของ matrix ด้วยค่าคงที่ เช่น A=[1 2]; จะได้

C =2*A

C =2 4

4. การบวก-ลบ Matrix

การที่ matrix จะบวก/ลบกันได้ต้องมีขนาดเท่ากัน และการบวก/ลบจะกระทำโดยบวก/ลบแต่ละelement ของ matrix

5. การยกกำลัง Matrix

Matrix ที่จะยกกำลังได้ต้องเป็น square matrix เท่านั้น โดย

$$A^2 = AA$$

6. Matrix Polynomial

สำหรับ polynomial function ของ x จะอยู่ในรูป

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots a_1 x + a_0$$

5.3 Matrix Functions

1. Matrix Inverse

ถ้า A เป็น square matrix จะได้ว่า Inverse Matrix ของ A นิยมเขียน A-1 จะนิยามโดย

$$AA^{-1} = A^{-1} A = I$$

เมื่อ I เป็น identity matrix สำหรับ matrix ที่ไม่สามารถหาค่า inverse ของ matrix ได้จะเรียกว่า singular matrixส่วน matrix ที่หาค่า inverse ได้ จะเรียก nonsingular matrix การหาค่า inverse matrix ที่มีขนาดใหญ่เป็นเรื่องยากในทางปฏิบัติ โดยเฉพาะการคำนวณตามหลักของพีชคณิต ดังนั้นส่วนใหญ่ในโปรแกรมคอมพิวเตอร์นิยมใช้วิธีเชิงตัวเลขมากกว่า สำหรับ MATLAB ก็เช่นกันโดยคำสั่งที่ใช้จะใช้คำสั่ง inv

มหาวิทยาลัยเทคโนโลยีสุรนารี

2. Rank of Matrix

ถ้าแต่ละ row ของ matrix แทนแต่ละสมการในพีชคณิตเชิงเส้น rank ของ matrix คือจำนวนสมการที่ไม่ขึ้นต่อกันของระบบสมการนั้น เช่น

$$a = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{pmatrix}$$

จะเห็นว่า row ที่ 2 เป็น 2 เท่าของ row แรก ดังนั้น row ทั้งสองจะขึ้นต่อกัน ทำให้ a นี้มี rank เท่ากับ 1 โดยทั่วไปแล้ว matrix ที่มี rank น้อยกว่าจำนวน row จะเป็น singular matrix สำหรับ MATLAB จะหา rankจากคำสั่ง rank

3. Determinant of Matrix

Determinant ของ square matrix เป็นการหาค่า scalar ของ matrix คำสั่งการหาค่า determinant ของsquare matrix a ใน MATLAB คือ

det(a) หาค่า determinant ของ a

สำหรับ matrix มี determinate เท่ากับศูนย์ matrix นั้นจะเป็น singular matrix การหา determinant โดย MATLAB

4. การหาร Matrix

ความจริงแล้ว matrix ไม่มีคุณสมบัติของการหาร แต่ MATLAB ได้สร้าง function นี้ขึ้นมาเพื่อสะดวกในการแก้ ระบบสมการ linear equations พิจารณา

$$Ax = B$$

ดังนั้น $x = A^{-1} B$

ซึ่ง MATLAB จะใช้ สัญลักษณ์ x=A\B

5. Matrix Decomposition

โดยทั่วไปแล้วเราสามารถจะแยก matrix A ออกเป็นผลคูณของ matrix 2 matrix ซึ่งเรียกว่า Decomposition หรือ Factorization ของ matrix ในปัญหาทางคณิตศาสตร์หลาย ๆ กรณี หากเราแยก matrixออกแล้ว จะสามารถลดการคำนวณต่างๆ ให้มีขั้นตอนน้อยลงได้ การ decomposed matrix ที่นิยมทำกันมี 3 วิธีคือ

1. LU-decomposition หรือบางครั้งเรียก Triangular Factorization คือการแยก matrix ออกเป็นผลคูณของสอง matrix โดยแยกเป็น lower triangular matrix และ upper triangular matrix

2. QR-Decomposition เป็นอีกวิธีหนึ่งในการแยก matrix ออกเป็น orthonormal matrix Q และ uppertriangular matrix, R (ซึ่งบางครั้งเรียก right hand matrix เพราะมีค่าเฉพาะด้านขวามือของ main diagonalเท่านั้น) โดย orthogonal matrix คือ matrix ที่มีคุณสมบัติ QQT = I

3. Singular Value decomposition (SVD) เป็นการ decomposed matrix ออกเป็นผลคูณของ matrix 3matrix คือ A = QSV

มหาวิทยาลัยเทคโนโลยีสุรนารี

6. Array Operation

ในหลายๆ กรณีเรามีความจำเป็นที่จะคูณหรือหารเฉพาะ element ของ matrix เท่านั้น ซึ่งนิยมเรียก element - by - element operation หรือ array operation

***** การใช้ Array operation สำหรับ matrix จะต้องใช้ได้กับ matrix ที่มีขนาดเท่ากันเท่านั้น**

7.ลำดับการคำนวณ

ขั้นตอนการคำนวณใน MATLAB ก็เช่นเดียวกันกับพีชคณิต ทั่วไปคือจะทำตามลำดับดังนี้คือ

(1) คำนวณในวงเล็บก่อน โดยเริ่มจากวงเล็บในสุดก่อน

(2) ยกกำลัง โดยเริ่มจากซ้ายไปขวา

(2) คูณหรือหาร โดยเริ่มจากซ้ายไปขวา

(3) บวก หรือ ลบ โดยเริ่มจากซ้ายไปขวา

*****เพื่อให้ลำดับการคำนวณเป็นไปตามที่ต้องการควรใช้วงเล็บเข้าช่วย เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้**

8. การจัดเรียง Matrix ใหม่

ในหลายกรณี เรามีความจำเป็นต้องจัด matrix ให้อยู่ในรูปอื่นๆ เพื่อความเหมาะสม เพื่อใช้ในการคำนวณที่เราต้องการ ซึ่ง MATLAB มี function ต่อไปนี้

rot90(A) จะหมุน matrix A ไป 90° ทวนเข็มนาฬิกา

rot90(A,n) จะหมุน matrix A ไป 90° เป็นจำนวน n ครั้ง ในทิศทวนเข็มนาฬิกา

fliplr(A) พลิก Matrix A จากซ้ายไปขวาคือเอา column ซ้ายสุดไปไว้ขวาสุดแล้วสลับที่กันต่อ ๆ ไปจนครบ

flipud(A) พลิก Matrix A จากบนลงล่างคือเอา row บนสุดไปไว้ล่างสุดแล้วสลับที่กันต่อๆ ไปจนครบ

reshape(A,m,n) จัด matrix A ใหม่ให้มีขนาดเป็น [m x n] โดย element ของ matrix A เดิมต้องมีจำนวนเท่ากับ mn ด้วย และ MATLAB จะพิจารณาเพิ่มหรือลดขนาดไปทีละ column ของ A

diag(x) ถ้า x เป็น matrix จะได้ผลเป็น vector ที่มี main diagonal ของ x เป็น element ถ้า x เป็น vector จะได้ผลเป็น matrix ที่มี vector A เป็น diagonal

triu(A) สร้าง square matrix จาก matrix A โดยตัดส่วนที่อยู่ต่ำกว่า main diagonal ของ A ให้เป็นศูนย์

triu(A,k) สร้าง square matrix จาก matrix A โดยตัดส่วนที่ต่ำกว่า diagonal ที่ k ของ A ให้เป็นศูนย์

tril(A) สร้าง square matrix จาก matrix A โดยตัดส่วนที่อยู่สูงกว่า main diagonal ของ A ให้เป็นศูนย์

tril(A,k) สร้าง square matrix จาก matrix A โดยตัดส่วนที่อยู่สูงกว่า diagonal ที่ k ของ A ให้เป็นศูนย์

5.4 Structures

สำหรับโครงสร้างตัวแปรแบบที่ 3 ที่บรรจุอยู่ใน MATLAB 5.x คือโครงสร้างตัวแปรที่เรียกว่าstructure ซึ่งทาง MATLAB จะกำหนดให้ structure นั้นเป็น object ในรูปแบบที่เป็นที่เก็บข้อมูล (Data Containers) หรือที่เรียกว่า fields สำหรับผู้ที่เคยผ่านงานการใช้โปรแกรมทางด้านฐานข้อมูล (database) คงคุ้นเคยกับศัพท์เหล่านี้เป็นอย่างดี ซึ่ง structure นั้นก็จะมีประโยชน์อย่างหนึ่งคือใช้ในการทำฐานข้อมูลเช่นกัน

1. สร้างการ Structure

การสร้างโครงสร้างฐานข้อมูลใน MATLAB 5.x ตามความคิดของผู้เรียบเรียงแล้วคิดว่าค่อนข้างจะง่ายกว่าการสร้างฐานข้อมูลบนโปรแกรมฐานข้อมูลทั่วไป ลำดับการสร้างฐานข้อมูลมีดังนี้

- ลำดับแรกเราต้องกำหนดว่าเราจะใช้ฐานข้อมูลชื่ออะไร
- ในฐานข้อมูลนั้นจะต้องการเก็บรายระเอียดอะไรบ้าง ซึ่งแต่ละรายละเอียดก็คือ field นั่นเอง
- กำหนดชื่อ field ทั้งหมด
- เราสามารถเริ่มใส่ข้อมูลได้เลย

สำหรับตัวอย่างสมมุติว่าเราต้องการจะรวมข้อมูลของชิ้นส่วนเครื่องจักรเครื่องหนึ่งไว้ในฐานข้อมูลชื่อ part โดยรายละเอียดที่ต้องการรวบรวมคือ ชื่อ(name) ผู้ผลิตชิ้นส่วน (make) ราคาที่ซื้อ (price) และหมายเหตุ (remark) ถ้ามี ดังนั้นเมื่อรวมแล้วเราจะมี filed ทั้งหมด 4 field สมมุติว่าเราจะตั้งชื่อfield แต่ละ field ตามชื่อภาษาอังกฤษ จากนั้นเราก็เริ่มใส่ข้อมูลได้

2. ลักษณะของ field

จากตัวอย่างที่ยกให้เห็นนั้นเป็นการยกตัวอย่างง่ายๆ เพื่อให้ผู้ที่ไม่เคยใช้ program ทางด้านฐานข้อมูลเข้าใจได้ สำหรับรายละเอียดที่มากกว่านั้นซึ่งเป็นคุณลักษณะของ MATLAB 5.x สามารถรวบรวมได้ดังนี้

- ค่าที่บรรทึกในแต่ละ field จะเป็น object ประเภท ตัวเลข, matrix, cell array หรือแม้แต่ Structure ก็ได้
- หากว่าต้องการจะดูชื่อ field ทั้งหมดเราสามารถทำได้โดยใช้คำสั่ง fieldname
- การสร้าง structure อาจสร้างได้โดยใช้คำสั่ง struct ได้
- ในแต่ละ field ที่กำหนดสามารถที่จะสร้างเป็น sub-field ต่อลงไปได้อีก

5.5 Output Options

มีวิธีการหลายวิธีที่จะแสดงค่าของตัวแปร หรือค่าคงที่ที่กำหนดลงไปในหน่วยความจำของ MATLAB วิธีที่ง่ายที่สุดที่พิมพ์ชื่อตัวแปรที่ Command Window แล้วกด Enter Key ซึ่ง MATLAB จะแสดงค่าตัวแปรนั้นออกมา ส่วนการแสดงผลนั้นมีวิธีกำหนดได้หลายวิธี ดังนี้

1. Display Format

รูปแบบที่จะแสดงผลตัวเลขแบบต่างๆ สามารถกำหนดให้การแสดงค่าเป็นไปตามที่ต้องการได้ดังต่อไปนี้

คำสั่ง MATLAB	การแสดงผล	ตัวอย่าง
format short	แสดงทศนิยม 4 ตำแหน่ง	22.1234
format long	แสดงทศนิยม 15 ตำแหน่ง	22.123456789012345
format bank	แสดงทศนิยม 2 ตำแหน่ง	22.12
format short e	แสดงทศนิยม 4 ตำแหน่งพร้อมเลขยกกำลังของ 10	2.123 e+01
format long e	แสดงทศนิยม 15 ตำแหน่งพร้อมเลขยกกำลังของ 10	2.12345678901234 e+01
format +	แสดงค่า + , - หรือว่าง	+

สำหรับค่าที่ตั้งเบื้องต้นคือ format short ซึ่งหากไม่กำหนดค่าเป็นอย่างอื่น MATLAB จะแสดงค่านี้ไปตลอดการใช้งาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

2. การแสดงผลตัวอักษรและค่าตัวแปร

นอกจาก MATLAB จะมี echo ซึ่งเป็นการแสดงค่าตัวแปรแล้ว ยังสามารถที่จะกำหนดรูปแบบให้MATLAB แสดงผลตามที่ต้องการได้อีกด้วย ดังมีการใช้คำสั่งต่อไปนี้

■ การแสดงผลอาจใช้คำสั่ง disp

เช่นถ้า A เป็นตัวแปร ซึ่งได้รับการกำหนดค่า disp(A) จะแสดงค่า A ส่วน disp('Display') MATLAB จะพิมพ์คำว่า Display ลงบนจอภาพ สำหรับตัวอักษร (string) นั้นจะต้องบรรจุอยู่ในเครื่องหมาย "' "'

■ การแสดงผลอย่างมีรูปแบบ

หากต้องการแสดงผลอย่างมีรูปแบบ ควรจะใช้คำสั่ง fprintf เพราะจะสามารถควบคุม output ได้ดีกว่าคำสั่ง disp สำหรับคำสั่ง fprintf นั้นสามารถกำหนดรูปแบบของ output variable ได้โดยใช้ตัวกำหนด (specifier) ซึ่งประกอบด้วย %e, %f และ %g

%e จะเป็นการแสดงผลในลักษณะ exponential notation เช่น 2.3456 + 2

%f จะแสดงเป็นเลขทศนิยม

%g จะเลือกแสดง %e หรือ %f แล้วแต่ว่าแบบใดจะสั้นกว่า

6. MATLAB FUNCTIONS

6.1 ฟังก์ชั่นทั่วไป

MATLAB มี function ทางคณิตศาสตร์พื้นฐาน เพื่อสะดวกแก่การใช้งานซึ่งได้แก่

abs(x) หาค่า absolute ของ x

sqrt(x) หาค่า รากที่ 2 ของ x

round(x) ทำให้ x เป็นจำนวนเต็ม โดยปัดค่าให้เป็นจำนวนเต็มที่ใกล้ x ที่สุด

fix(x) ทำให้ x เป็นจำนวนเต็ม โดยปัดค่า x ให้เป็นจำนวนเต็มที่ใกล้ศูนย์ที่สุด

floor(x) ทำให้ x เป็นจำนวนเต็ม โดยปัดค่า x ให้เป็นจำนวนเต็มที่ใกล้ x ไปทาง - infinity

ceil(x) ทำให้ x เป็นจำนวนเต็ม โดยปัดค่า x ให้เป็นจำนวนเต็มที่ใกล้ x ไปทาง + infinity

sign(x) บอกเครื่องหมายของ x โดยจะเป็น -1 ถ้า x < 0, เป็น 1 ถ้า x > 0 และเป็น 0 ถ้า x = 0

rem(x,y) หาเศษที่ได้จากการหาร x ด้วย y หรือ เศษของ x/y

exp(x) หาค่า	ex
log(x) หาค่า	ln(x) หรือ natural logarithm ของ x
log10(x) หาค่า	log10 x หรือ logarithm ฐาน 10 ของ x

6.2 Complex Number Functions

Complex number z นิยามด้วยจำนวนจริงสองจำนวนคือ

$$z = x + iy$$

เมื่อ x และ y เป็น real และ i = $\sqrt{-1}$

x เป็น real part ของ z ; x = Re(z)

y เป็น imaginary part ของ z ; y = Im (z)

สำหรับ MATLAB เมื่อเริ่มต้นจะกำหนดค่า i และ j ให้เท่ากับ$\sqrt{-1}$ไว้โดยจะใช้ i หรือ j เป็น imaginary number ก็ได้ สำหรับ function ของ complex number ที่มีใน MATLAB โดย parameter z = x + iy มีดังนั้น

conj(z) หา	complex conjugate ของ z คือ ถ้า z = x + iy, จะให้ค่า x - iy
real(z)	หา Real part ของ z คือ ถ้า z = x + iy, จะเท่ากับ x
imag(z) หา	imaginary part ของ z คือ ถ้า z = x + iy, จะเท่ากับ y
abs(z)	หาค่า absolute หรือขนาดของ z จะให้ค่าผลลัพธ์เป็นค่าของ x2 + y2
angle(z)	หาค่ามุมหรือ argument ของ z คือ angle (z) = $\tan^{-1}\frac{x}{y}$ โดยค่ามุมที่ได้จะอยู่ระหว่าง -π ถึง π

6.3 File Input/Output Functions

แม้ว่า MATLAB จะเป็นโปรแกรมที่มีความสามารถในการทำงานในการคำนวณได้มากมาย แต่ในหลายๆกรณีก็มีความจำเป็นที่จะต้องติดต่อ ส่งผ่านข้อมูลกับโปรแกรมอื่น หรือต้องมีการเก็บข้อมูลเพื่อที่จะนำไปใช้ต่อไป ดังนั้นจึงจำเป็นอย่างยิ่งที่จะต้องมีระบบการบันทึกและรับส่งข้อมูล สำหรับในหัวข้อนี้จะกล่าวถึงวิธีการเปิด-อ่าน-เขียน-ปิด file ของ MATLAB

สำหรับผู้ที่ยังไม่มีประสบการณ์มากนักกับการเปิด -ปิด file ขอให้พิจารณาหลักง่ายๆดังนี้ หากเราเดินเข้าไปที่ชั้นหนังสือ มองดูหนังสือที่มีอยู่มากมาย เรามีความต้องการที่จะทราบข้อมูลในหนังสือเล่มใดเล่มหนึ่งหรือหลายๆเล่มเราจะทำอย่างไร อันดับแรกก็คงจะเป็นการหยิบหนังสือที่มีชื่อ

ที่ตรงกับที่เราสนใจขึ้นมา เปิดหนังสือไปที่หน้าที่ต้องการ อ่านข้อความเมื่อเสร็จสิ้นแล้วก็ปิดหนังสือเล่มนั้นเก็บเข้าที่เดิม ในการเปิด file ก็เช่นเดียวกันขั้นแรกก็จะต้องบอกเครื่องคอมพิวเตอร์ว่าในจำนวน file ที่มีมากมายใน disk นั้นเราต้องการจะเปิด file ใด จากนั้นก็อ่านข้อมูล หรือเขียนข้อมูลลงไป แล้วก็ทำการปิดfile นั้นให้เรียบร้อย

1. การเปิดและปิด file

คำสั่งแรกที่จะให้ MATLAB เปิด file ขึ้นมาก่อนก็คือคำสั่ง fopen

Fid = fopen(filename, permission)

เป็นการเปิด file ตามชื่อ filename และกำหนดลักษณะการเปิด file ด้วย permission ถ้า file ที่เปิดขึ้นสำหรับที่จะอ่านค่าแล้วไม่ปรากฏใน directory ที่MATLAB กำลังทำงานอยู่ MATLAB จะไปหาใน search path. สำหรับผลรับ fid ที่ได้จะเป็น file identifiers หรือก็คือหมายเลขกำหนด file ว่าขณะที่กำลังทำงานอยู่นั้น MATLAB จะจำ file นี้ด้วยหมายเลขอะไร สำหรับเงื่อนไขลักษณะการเปิดหรือ permission เป็น strings ตัวใดตัวหนึ่งต่อไปนี้

'r'	เปิดเพื่ออ่านอย่างเดียว (read)
'w'	เปิดขึ้นเพื่อเขียนหากไม่มี file ชื่อดังกล่าวก็จะสร้างขึ้นใหม่
'a'	เพิ่มเติม (append) สร้างใหม่ถ้าจำเป็น
'r+'	อ่านและเขียน ไม่มีการสร้างใหม่
'w+'	ตัดหรือสร้างใหม่สำหรับการอ่านและเขียน
'a+'	อ่านและเพิ่มเติม สร้างใหม่ถ้าจำเป็น

สำหรับในเบื้องต้นนั้น file ที่เปิดจะมี format เป็น binary code หากว่าต้องการจะเปิด file ที่เป็น text จะต้องเพิ่ม 't' เข้าไปใน string ตัวอย่างเช่น 'rt' หรือ 'wt+' อย่าลืมว่าบนเครื่อง UNIX และ Macintosh นั้น text file และ binary file จะเหมือนกันแต่สำหรับเครื่อง PC file ทั้งสองจะแตกต่างกันโดยสิ้นเชิง สำหรับในกรณีที่การเปิด file ไม่สำเร็จ MATLAB จะให้ค่า fid = -1 นอกจากนี้ fid =1, 2 และ 3 MATLAB จะสำรองไว้ใช้งานเองสำหรับชื่อ file นั้นจะต้องเขียนเป็น string และสำหรับเครื่อง PC การกำหนด directory จะทำตามมาตรฐานของ DOS

Fids = fopen('all') จะให้ row vector ที่มี file identifiers สำหรับทุก file ที่กำลังเปิดทำงานอยู่ใน ขณะนั้น

ส่วนการปิด file จะใช้คำสั่ง fclose ดังมีรูปแบบต่อไปนี้

st=fclose(fid) ปิด file ตาม file identifier, fid, ที่กำหนด ซึ่งได้จากการใช้คำสั่ง fopen ก่อน หน้านี้ ถ้า st =0 แสดงว่าปิด file ได้เรียบร้อย และ st = -1 แสดงว่าไม่เรียบร้อย

st=fclose('all') ปิดทุก file ที่เปิดอยู่

2. การเขียนและอ่าน Binary File

สำหรับการเขียนและอ่าน binary file สามารถใช้คำสั่ง fread และ fwrite ซึ่งมีรูปแบบของคำสั่งดังต่อไปนี้คือ สำหรับการอ่าน [A,count]= fread(fid,size,precision)

อ่าน binary data จาก file ที่กำหนดโดย fid และกำหนดค่าที่ได้ให้เป็นค่าของ matrix A ส่วนตัวเลือกเพิ่มเติม count จะบอกว่าได้มีการอ่านข้อมูลสำเร็จไปเท่าใด นอกเหนือจากนั้นตัวกำหนดขนาด size จะเป็นตัวเลือกหากไม่มีการกำหนด MATLAB จะอ่านจนกระทั่งหมด file แต่ถ้าจะกำหนดจะสามารถกำหนดได้ดังนี้

N อ่าน N element เข้าเป็น column vector

inf อ่านจนจบ file (ค่าเบื้องต้น)

[M,N] อ่านข้อมูลให้เข้าสู่ matrix ขนาด M x N เรียงตาม column

count=fwrite(fid,a,precision)

เขียน element ของ matrix A ลงไปใน file ที่กำหนดโดย fid โดยMATLAB จะเขียนตามลำดับ column ส่วนตัวเลือก count เป็นค่าของจำนวน elements ที่เขียนได้สำเร็จ

สำหรับ precision เป็นการกำหนดรูปแบบตัวแปรและความแม่นยำเป็นไปตามตารางต่อไปนี้

String	ความหมาย
'char'	ตัวอักษร - 8 bits
'uchar'	ตัวอักษรแบบ unsigned character - 8 bits
'schar'	ตัวอักษรแบบ signed character - 8 bits
'int8', 'int16','int32','int64'	จำนวนเต็ม - 8,16,32 และ 64 bits
'uint8', 'uint16', 'uint32', 'uint64'	จำนวนเต็มแบบ unsigned integer - 8,16,32 และ 64 bits
'float32'	floating point - 32 bits
'float64'	floating point - 64 bits

นอกเหนือจากนั้นยังสามารถใช้ค่าต่อไปนี้ได้ แต่อาจจะมีปัญหาหากใช้เครื่อง computer ที่มีรูปแบบการจัดโครงสร้างที่ต่างกัน

3. การอ่านและเขียน Format File

สำหรับ text file หรือ file ที่มีรูปแบบเฉพาะจะมีขั้นตอนการอ่านและเขียนแตกต่างจาก binary file โดยคำสั่งที่ใช้จะเป็น fscanf และ fprintf ซึ่งมีรายละเอียดดังนี้ สำหรับการอ่าน

[a,count]=fscanf(fid,format,size)

อ่านค่าจาก format file ตาม fid ที่กำหนดจากนั้นเปลี่ยนให้มาเป็นรูปแบบที่กำหนดตามคำสั่ง format แล้วให้ค่าเป็น matrix a ส่วนตัวเลือก count เป็นค่าที่บอกว่าได้อ่านข้อมูลมาเท่าใดแล้ว ส่วน size เป็นตัวเลือกที่กำหนดว่าจะอ่านค่าครั้งละเท่าใด เหมือนที่ใช้กับคำสั่ง freadสำหรับ format string จะมีลักษณะเป็น เครื่องหมาย % ตามด้วยตัวอักษรต่อไปนี้คือ d, i, o, u, x, e, f, g, s, c, และ [. . .] (scanset). ซึ่งจะเป็นรูปแบบของภาษา C ที่ใช้กันทั่วไปก็เช่น %s เป็น character string,%f เป็น floating point และ %d เป็น decimal point integer เป็นต้น การอ่านข้อมูล MATLAB จะอ่านเรียงบรรทัดจากบนลงล่าง

สำหรับการเขียนจะใช้

Count = fprintf(fid,format,a)

เขียน format data เฉพาะในส่วน real part ของ matrix a ภายใต้เงื่อนไขของ format string ลงบน file ที่เปิดขึ้นตาม fid ส่วน countเป็นตัวเลือกเพื่อบอกจำนวนข้อมูลที่เขียนลงไปบน file ได้สำเร็จ

มหาวิทยาลัยเทคโนโลยีสุรนารี

7. การเขียน M-FILE

7.1 M-File

การเขียนโปรแกรมโดยใช้ MATLAB เป็นการช่วยให้การทำงานของวิศวกรง่ายขึ้นมากเมื่อเทียบกับการใช้ภาษาพื้นฐานเช่น C, FORTRAN, PASCAL,... ทั้งนี้เพราะว่า MATLAB มี function ทางคณิตศาสตร์และการเขียนกราฟรองรับไว้มากมายซึ่งช่วยลดเวลาการเขียนโปรแกรมลงไปได้อย่างมาก อย่างไรก็ตามในบางกรณีเรามีความจำเป็นต้องเขียน function ขึ้นมาใหม่เพื่อให้เหมาะสมกับงานของเรา

โปรแกรมที่เขียนโดย MATLAB จะ Save โดยใช้ extension เป็น " m " ซึ่งเรานิยมเรียกโปรแกรมที่เขียนโดยใช้ MATLAB ว่า M-file โดย M-file นี้จะแบ่งออกเป็น 2 ลักษณะคือเขียนในลักษณะของการบอกขั้นตอนหรือบอกบทการทำงานหรือที่นิยมเรียกว่า script file และอีกประเภท

หนึ่งจะเขียนขึ้นในลักษณะของ function ซึ่งผู้ใช้สามารถรวบรวมเอา function ต่าง ๆ ของ MATLAB มารวมเข้าด้วยกันแล้วเขียนขึ้นเป็น function ใหม่ M-file ในลักษณะนี้เรียก function file

M-file ใน MATLAB จะเขียนเป็น plain text format ธรรมดา ดังนั้นเราอาจใช้ program เล็ก ๆ เขียนเช่น Notepad เขียนก็ได้ และการ save file จะ save เป็นชื่อ file ที่ต้องการโดยมี extension เป็น m สำหรับMATLAB 5.x แล้ว จะมี MATLAB Editor/Debugger เพื่อใช้ในการเขียนโปรแกรมและแก้ไขโปรแกรม ส่วนการเรียกใช้ M-file นั้นก็เพียง พิมพ์ชื่อ file ที่ต้องการ โดยไม่ต้องมี extension เช่น ถ้าเราเขียน M-file ชื่อ myfile.m เวลาเรียกใช้ที่ command window ของ MATLAB ก็จะใช้คำสั่ง

» myfile

1. การเรียกใช้ M-file

การที่ MATLAB จะค้นหา M-file ที่สร้างขึ้นได้พบนั้น เราจะต้องกำหนดว่า directory ใด หรือ pathใดบ้าง ที่จะให้ MATLAB ค้นหาเพราะ MATLAB จะไม่ค้นหาทุก directory หรือ sub-directory ของทุก driveที่มีในเครื่องขณะนั้น แต่ MATLAB จะค้นหาเฉพาะใน search path ที่กำหนดเท่านั้น ดังนั้นการเขียน M-fileขึ้นเองจะต้องเก็บ file นี้ให้อยู่ใน path ที่ MATLAB จะค้นหา สำหรับการปฏิบัติที่เหมาะสมเราควรจะเก็บ M-file ที่เราสร้างขึ้นไว้ใน directory ของเราต่างหากเพื่อไม่ให้ปะปนกับ M-file ของ MATLAB จากนั้นจึงเพิ่ม MATLAB search path ให้รวม directory ที่เราสร้างขึ้นใหม่เพื่อให้ MATLAB ทราบว่าต้องค้นหา programที่เราเขียนขึ้นในที่ใด ขั้นตอนการเพิ่ม search path ทำได้หลายวิธีดังนี้

- การเพิ่ม search path ชั่วคราวสำหรับการทำงานแต่ละครั้งที่เรียกใช้ MATLAB ให้ใช้คำสั่ง addpath แล้วตามด้วยชื่อ directory ที่จะเพิ่มใน search path โดยการกำหนด directory ให้ใช้รูปแบบของ DOS
- การกำหนด search path อย่างถาวรใน MATLAB 5.X สามารถกระทำได้โดยง่ายโดยการ clickที่ File เลือก Set path... จากนั้นเพิ่ม path ที่ท่านต้องการ

7.2 Script Files & Function Files

ในการเขียน MATLAB เพื่อการคำนวณค่าต่างๆ นั้น เราสามารถป้อนค่าตัวแปรและคำสั่ง ซึ่งเป็นfunction ต่างๆ ที่สร้างขึ้นมาใน MATLAB เพื่อให้คำนวณและแสดงผลที่เราต้องการได้ โดยการป้อนคำสั่งนั้น อาจกระทำทีละบรรทัด ตามที่ตัวอย่างได้กล่าวมาแล้ว หรือ เราอาจจะเขียนเป็น

ชุดคำสั่งเหมือนกับการเขียน program computer ทั่ว ๆ ไป หรือที่เรียกกันว่าเขียน script files เพื่อสะดวกในการแก้ไขและมีขั้นตอนการคำนวณและแสดงผลเป็นไปตามที่เราต้องการ

นอกจาก MATLAB จะเขียนเป็นบท (script) แล้วยังสามารถจะสร้าง function file ที่ท่านต้องการเองได้อีกด้วย ข้อแตกต่างของ script file กับ function file คือ

- ใน Function file จะต้องมีการกำหนด input ที่จะเข้าสู่ file ที่แน่นอนและต้องมีการกำหนดoutput ที่ต้องการที่แน่นอน ส่วนใน script file ไม่จำเป็นต้องมี นั่นคือการเขียน function fileจะต้องมี parameter
- ใน script file ค่าตัวแปรที่มีอยู่ในหน่วยความจำทุกค่าจะถูกส่งเข้าสู่ M-file และ MATLAB จะเก็บค่าทุกค่าที่เกิดขึ้นในระหว่างการคำนวณไว้ในหน่วยความจำแต่ function file จะใช้เฉพาะค่าที่กำหนดเข้าไปเท่านั้น และจะแสดงค่าเฉพาะค่าที่เราต้องการเป็น output ส่วนค่าอื่นที่เกิดขึ้นในระหว่างการหาค่า function นั้น MATLAB จะไม่เก็บไว้ในหน่วยความจำ กล่าวสั้นๆก็คือตัวแปรของ script file จะเป็น global variable ส่วนของ function file จะเป็น local variable

ซึ่งจะเห็นว่าการใช้ function file อาจจะดูยุ่งยากกว่า แต่มีข้อได้เปรียบหลายประการ คือ

- ประหยัดหน่วยความจำกว่า script file
- การตั้งชื่อ variable ใน M-file ไม่ต้องเกรงว่าจะเกิดการซ้ำซ้อนกับค่าที่ผู้ใช้กำลังใช้อยู่ เพราะMATLAB จะเก็บในหน่วยความจำเฉพาะค่า input และ output
- ในหลายกรณี การใช้งาน function file จะสะดวกกว่า script file เพราะมีรูปแบบการกำหนดค่าparameter ต่าง ๆ ทั้งที่จะต้องให้เข้าไปและค่าที่ได้ออกมาเป็นรูปแบบที่แน่นอน

มหาวิทยาลัยเทคโนโลยีสุรนารี

*****ไม่ว่าจะเป็น script file หรือ function file ก็ตาม การเขียนก็จะเขียนเป็น M-file เหมือนกัน และจะต้องบรรจุอยู่ใน search path ของ MATLAB เช่นเดียวกัน**

การเขียน script file จะขึ้นต้นด้วยอะไรก็ได้ แต่สำหรับ function file จะต้องขึ้นต้นบรรทัดแรกด้วยรูปแบบนี้เสมอ

function mypro(c)

โดยคำสั่ง function จะบอก MATLAB ว่า file นี้เป็น function file

c เป็น input หรือ parameter

mypro เป็นชื่อ function ของเรา “และ file นี้จะต้อง save ในชื่อ mypro. m” เท่านั้น

***** Function file จะต้องขึ้นบรรทัดแรกด้วยคำสั่ง function เสมอและชื่อของ M-file ต้อง Save ด้วยชื่อเดียวกับชื่อของ function นั้น**

1. การใช้ Comment (%)

ในการเขียนโปรแกรมนั้นในหลาย ๆ กรณีจำเป็นต้องมีการเขียน comment หรือข้อช่วยบันทึกความจำลงไปด้วยเพื่อสะดวกในการแก้ไขหรือเป็นประโยชน์แก่ผู้ใช้ เครื่องหมาย % เป็นเครื่องหมาย comment ซึ่ง MATLAB จะไม่สนใจที่จะทำคำสั่งหรือข้อความต่าง ๆ ที่อยู่หลังเครื่องหมายนี้ ข้อสำคัญอีกประการหนึ่งสำหรับเครื่องหมาย % นี้ก็คือสามารถใช้เป็น help ของ M-file นั้นได้ด้วย นั่นก็คือ

- สำหรับ script file ข้อความที่อยู่หลัง % ทั้งหมดนับจากบรรทัดแรกจนกระทั่งถึงบรรทัดที่ไม่มีเครื่องหมายนี้จะปรากฏขึ้น ถ้าหากมีการพิมพ์ help แล้วต่อด้วยชื่อ file นั้น
- สำหรับ function file ข้อความที่อยู่หลัง % ทั้งหมดนับจากบรรทัดที่ต่อจากคำสั่ง functionจนกระทั่งถึงบรรทัดที่ไม่มีเครื่องหมายนี้จะปรากฏขึ้นหากมีการพิมพ์ help แล้วต่อด้วยชื่อfile นั้น

7.3 คำสั่งที่ควบคุมขั้นตอนการทำงานของ M-File

เช่นเดียวกับการเขียน program computer ทั่วไป MATLAB มีคำสั่งที่ใช้ในการควบคุมขั้นตอนการทำงานของ program เพื่อสะดวกในการทำงาน เช่นเดียวกับใน C, FORTRAN หรือ BASIC แต่ข้อแตกต่างจากการใช้ภาษาพื้นฐานเหล่านั้น คือ

- ใน MATLAB ไม่มี line number ดังนั้นการเขียนโปรแกรมหากต้องการข้ามชุดคำสั่งใดอาจต้องใช้ข้อความทางตรรกเข้าช่วยหรือใช้วิธีการใช้ subroutine หรือ function เข้าช่วย
- ใน MATLAB ไม่ต้องมีการบอกว่าตัวแปรใดเป็น string, integer หรือถ้าเป็นเลขทศนิยมก็ไม่จำเป็นต้องบอกว่าจะมี precision แบบใดยกเว้นว่าต้องการกำหนดเอง
- ไม่จำเป็นต้องจอง array หรือจอง dimension ของตัวแปร
- สามารถเรียกใช้ function ต่าง ๆ ที่มีอยู่ใน MATLAB หรือที่สร้างขึ้นเองได้ตลอดเวลา หรือ ไม่จำเป็นต้องเขียน subroutine ขึ้นมาใหม่

- MATLAB ทำงานได้โดยไม่ต้อง compile ก่อน ดังนั้น โดยทั่วไป MATLAB จะหยุดทำงานทันทีเมื่อพบ error

คำสั่งที่ใช้ควบคุมการทำงานหลัก ๆ มีดังนี้

› For loop Repetitive Control Structure

› If - elseif - else - end Conditional Control Structure

› While loop Repetitive Control Structure

1. For Loop

สำหรับ For loop นั้นก็จะเหมือนกับ for loop (หรือ Do loop) ของภาษาอื่น ๆ โดยมีโครงสร้างเป็น

```
for m = a: b
ชุดคำสั่งหรือการคำนวณ
End
```

โดยคำสั่งนี้จะกำหนดให้ MATLAB ทำเป็นวงรอบ (loop) โดยเริ่มจากค่า m = a แล้วทำบรรทัดต่อมา คือ คำนวณชุดคำสั่งไม่ว่าจะมีชุดคำสั่งเท่าใดก็ตาม จากนั้นเมื่อพบคำสั่ง end แล้ว MATLAB ก็จะกลับไปเริ่มต้นที่คำสั่ง for อีก แล้วทำงานต่อโดยใช้ค่า m = a + 1 ไปเรื่อย และ จะเลิกทำงานเมื่อคำนวณค่าสุดท้ายที่ m = b ซึ่ง ในกรณีการใช้คำสั่งนี้ a และ b จะเป็นจำนวนเต็ม หรือหากใช้

for m = a:b:c จะเป็นการทำค่า l เริ่มจาก a ถึง c โดยเพิ่มขึ้นครั้งละ b

มหาวิทยาลัยเทคโนโลยีสุรนารี

*****สำหรับผู้ที่ใช้ภาษา FORTRAN อยู่เป็นประจำอาจนิยมที่จะทำ loop ด้วยตัวแปร i หรือ j เพราะกำหนเป็น integer ของ FORTRAN อย่างไรก็ตามใน MATLAB ไม่มีความจำเป็นและไม่แนะนำให้ใช้เพราะค่าทั้งสอง MATLAB ได้สำรองไว้สำหรับ imaginary number $\sqrt{-1}$**

2. การใช้ For Loop ซ้อนกัน

ในหลายกรณีเรามีความจำเป็นต้องใช้ for loop ซ้อนกัน ซึ่งก็จะสามารถกระทำได้ เช่น ตัวอย่างต่อไปนี้

```
for k = 1 : 5
    for i = 1 : 5
        p(k,i) = k * i ;
```

```
        end
    end
```

โปรแกรมนี้จะสร้าง matrix p มีขนาด 5x5 โดยแต่ละ element จะมีค่าเท่ากับผลคูณของตำแหน่ง row และ column จากโปรแกรมข้างบนนี้ จะเริ่มเมื่อ k = 1 จากนั้นจะเข้า loop ที่สองซึ่งจะได้ค่า i = 1 ดังนั้น P(1,1) = 1*1=1 และเมื่อพบคำสั่ง end เครื่องจะกลับไปที่จุดเริ่มต้นของ loop ที่ 2 ดังนั้น เมื่อถึงจุดนี้ k ยังคงเป็น 1 แต่ i = 2 ดังนั้น P(1,2) = 1*2=2 และเครื่องจะทำงานต่อไปจนกระทั่ง i = 5 ซึ่งครบคำสั่ง for ของ loop ที่สอง จากนั้น เมื่อพบคำสั่ง end ของ loop ที่สองเครื่องจะทำงานในบรรทัดต่อไป แต่เมื่อพบคำสั่ง end ของ loop แรกแล้ว เครื่องจะกลับไปเริ่มต้นใหม่ โดยใช้ค่า k = 2 แล้วจะวกกลับมาเข้าloop ที่สองอีกครั้งหนึ่ง ซึ่งวนค่า i = 1, 2, 3, 4, 5 ไปเรื่อย ๆ และ loop นี้จะหยุดการทำงานก็ต่อเมื่อจำนวนค่า k = 5 และ l = 5

ใน MATLAB 5.x เนื่องจากเราเขียน M-file ใน Editer ของ MATLAB เอง ดังนั้น เมื่อมี loop ซ้อนกันหลาย loop MATLAB จะทำการจัดย่อหน้าของคำสั่งให้เอง เพื่อความสะดวกในการแก้ไข โดยคำสั่งใน loop เดียวกันจะย่อหน้าเท่ากัน และคำสั่ง end จะตรงกับคำสั่ง for ของ loop นั้น ซึ่งจะเป็นเช่นเดียวกันกับคำสั่งในการวน loop อื่น ๆ

3. If Statement

สำหรับการใช้ คำสั่ง If มีรูปแบบดังนี้

```
if
        เงื่อนไข
        ชุดคำสั่ง
end
```

โดยถ้าเงื่อนไขเป็นจริง MATLAB จะคำนวณชุดคำสั่ง แต่ถ้าไม่เป็นจริง MATLAB จะข้ามไปทำคำสั่งในบรรทัดที่ต่อจาก end ต่อไป ในเงื่อนไขหรือ condition จะต้องมีค่าเป็นจริงหรือเท็จเท่านั้นเครื่องหมายที่ใช้เปรียบเทียบ ส่วนใหญ่เป็นดังนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

ความหมาย	สัญลักษณ์คณิตศาสตร์	MATLAB
เท่ากับ	=	= =
ไม่เท่ากับ	≠	~ =
ความหมาย	สัญลักษณ์คณิตศาสตร์	MATLAB
มากกว่า	>	>
น้อยกว่า	<	<
มากกว่าหรือเท่ากับ	≥	= >
น้อยกว่าหรือเท่ากับ	≤	= <
และ	AND	&
หรือ	OR	|

ตัวอย่างเช่น if x == 0

a = 5

end

นั่นคือ ถ้า x = 0 แล้วจะได้ว่า a = 5 แต่ถ้า x ไม่เท่ากับศูนย์ จะไม่มีการกำหนดให้ค่า a = 5

4. If-Elseif-Else

การใช้ else และ elseif นั้นมีรูปแบบดังนี้

if เงื่อนไขที่ 1

ชุดคำสั่งที่ 1

elseif เงื่อนไขที่ 2

ชุดคำสั่งที่ 2

elseif เงื่อนไขที่ 3

ชุดคำสั่งที่ 4

.....

else เงื่อนไขที่ n

ชุดคำสั่งที่ n

End

ในกรณีนี้ถ้าเงื่อนไขที่ 1 เป็นจริง MATLAB จะทำชุดคำสั่งที่ 1 แล้วมาที่ end แต่ถ้าไม่เป็นจริงMATLAB จะพิจารณาเงื่อนไขที่ 2 ถ้าเป็นจริง MATLAB จะทำชุดคำสั่งที่ 2 แล้วมาที่ end แต่ถ้า

ยังไม่เป็นจริงอีก MATLAB จะพิจารณาเงื่อนไขที่ 3 ถ้าเป็นจริง MATLAB จะทำชุดคำสั่งที่ 3 แล้วไปที่ end ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงเงื่อนไขที่ n ถ้าไม่มีเงื่อนไขใดเลยที่เป็นจริง MATLAB ก็จะไม่ทำชุดคำสั่งใดแล้วมาที่ end เลย

5. While - Loop

สำหรับ while loop นั้นก็จะคล้าย ๆ กับ for loop จะต่างกันที่ while loop นี้จะไม่กำหนดจำนวนรอบเหมือนกับ for loop แต่จะเป็นการวน loop ไปเรื่อย ๆ ตราบเท่าที่เงื่อนไขที่ให้ยังเป็นจริงอยู่รูปแบบของ while-loop คือ

```
While        เงื่อนไข
             ชุดคำสั่ง
end
```

โดย loop นี้จะดำเนินไปเรื่อย ๆ จนกว่าเงื่อนไขนี้จะเป็นเท็จ เมื่อเงื่อนไขเป็นเท็จ MATLAB จะมาที่คำสั่งend แล้วไปทำคำสั่งบรรทัดต่อไป เช่น

```
          a = 0
while     a < 5
          a = a + 1
          b = 5 + a
end
```

เริ่มต้น a = 0 เมื่อเข้าสู่ loop while เงื่อนไข a < 5 จะเป็นจริง ดังนั้น MATLAB จะคำนวณค่า a = 0+ 1 = 1 และ b = 5 + 1 = 6 เมื่อถึง end แล้ว MATLAB จะกลับไปที่ while อีกครั้งหนึ่งในขณะนี้ a=1ดังนั้น เงื่อนไข a < 5 ยังคงเป็นจริงอยู่ ดังนั้น MATLAB ก็จะคำนวณชุดคำสั่งต่อไปอีก ซึ่งจะทำเป็นรอบไปเรื่อย ๆ และจะหยุดเมื่อ a มีค่ามากกว่า 5

***** ข้อควรระวัง ในการใช้คำสั่งเพื่อให้เกิด loop นั้นต้องตรวจสอบให้ดีไม่เช่นนั้น โปรแกรมอาจจะอยู่ในสภาพ “ติด loop” คือ วนทำงานอยู่ใน loop นั้นไม่สามารถออกมาทำงานในชุดคำสั่งอื่นได้ สภาพการเช่นนี้สังเกตได้จากเครื่องจะใช้เวลาในการคำนวณนานเกินความจำเป็นมาก และโปรแกรมไม่มีทีท่าว่าจะดำเนินต่อไปได้ การแก้ไขให้กด Ctrl-C เป็นยกเลิกชุดคำสั่ง MATLAB ทั้งหมด**

6. คำสั่ง ยกเลิก Loop

ใน program เอง หากว่า ต้องการยกเลิก loop ก็สามารถทำได้โดยใช้คำสั่ง

Break

เมื่อ MATLAB พบคำสั่งนี้จะออกจาก loop ที่กำลังทำอยู่ทันที แล้วดำเนินการคำนวณชุดคำสั่งหลังจาก end ต่อไปเช่น

```
for k = 1 : 5
        y = k^2
        if y = = 9
                break
        end
        x = 5/(y-9)
end
```

ในตัวอย่างนี้ MATLAB จะเริ่มเข้า loop และคำนวณหาค่า y ก่อน ถ้าหากว่า y ไม่เท่ากับ 9 MATLAB ก็จะทำงานต่อไป เพื่อหาค่า x แต่ว่าเมื่อใดก็ตาม y = 9 ทำให้เงื่อนไขเป็นจริง MATLAB จะทำงานในชุดคำสั่งของ If นั่นคือสั่งให้หยุด (break) ดังนั้นถึงจุดนี้ MATLAB จะข้ามมาทำบรรทัดต่อจากend ของ loop for เลย ซึ่งเป็นการออกจาก loop มานั่นเอง

สำหรับคำสั่ง control flow ที่เพิ่มขึ้นของ MATLAB 5.X คือคำสั่ง switch - case ซึ่งเป็นคำสั่งที่ช่วยให้การใช้งานกรณีที่ตัวแปรมีโอกาสมีค่าได้หลายค่าและค่าแต่ละค่าจะมีคำสั่งให้ทำงานต่างๆกันออกไปโดยมีโครงสร้างต่อไปนี้

```
Switch   ตัวแปร
case  ค่าตัวแปร     กรณีที่ 1
        ชุดคำสั่งที่ 1
case  ค่าตัวแปร     กรณีที่ 2
        ชุดคำสั่งที่ 2
Case      ค่าตัวแปร กรณีที่ 3
        ชุดคำสั่งที่ 3
        .....
otherwise
        ชุดคำสั่งที่ n
```

```
end
```

ลักษณะของการทำงานคือ เมื่อเรากำหนดให้พิจารณาตัวแปรตามคำสั่ง switch จากนั้น MATLABจะพิจารณาว่าค่าของตัวแปรที่กำหนดนั้นเข้ากับกรณีใด ก็จะทำชุดคำสั่งของกรณีนั้น พิจารณาตัวอย่างต่อไปนี้

```
x=input('กรุณาเลือกคำตอบจากข้อ 1-4 \n');
switch x
    case 1
        fprintf('ท่านได้เลือกข้อ 1 \n')
    case 2
        fprintf('ท่านได้เลือกข้อ 2 \n')
    case 3
        fprintf('ท่านได้เลือกข้อ 3 \n')
    case 4
        fprintf('ท่านได้เลือกข้อ 4 \n')
    otherwise
        fprintf('ท่านได้เลือกข้อที่ไม่มีกำหนดไว้ \n')
end
```

จากการที่กำหนดค่า x แล้วเราสั่งให้พิจารณาค่า x จากนั้น MATLAB จะพิจารณาว่าค่า x ที่กำหนดให้จะเข้ากับกรณีใด ก็จะทำตามคำสั่งนั้น เช่นถ้า x = 1 เครื่องก็จะพิมพ์ว่า ท่านได้เลือกข้อ 1 เป็นต้น

มหาวิทยาลัยเทคโนโลยีสุรนารี

7. คำสั่งควบคุมขั้นตอนอื่นๆ

คำสั่งควบคุมขั้นตอนการทำงานอื่น ๆ ของ MATLAB มีดังนี้

pause	ให้หยุดการทำงานแล้วรอจนกระทั่งมีการกด keyboard อันใดอันหนึ่งจึงจะทำงานต่อ
pause(n)	หยุดการทำงานเป็นเวลา n วินาที
return	เลิกการทำงานของ M-file นั้นทั้งหมด แล้วกลับไปที่จุด ซึ่งเรียก M-file นี้มาใช้งาน
pause	on หรือ off ให้รับหรือไม่รับคำสั่ง pause ที่จะมีตามมาใน M-file

นี้สำหรับการเขียน M-file นั้นอย่างที่กล่าวมาแล้วก็คือเหมือนกับการเขียนโปรแกรมภาษา

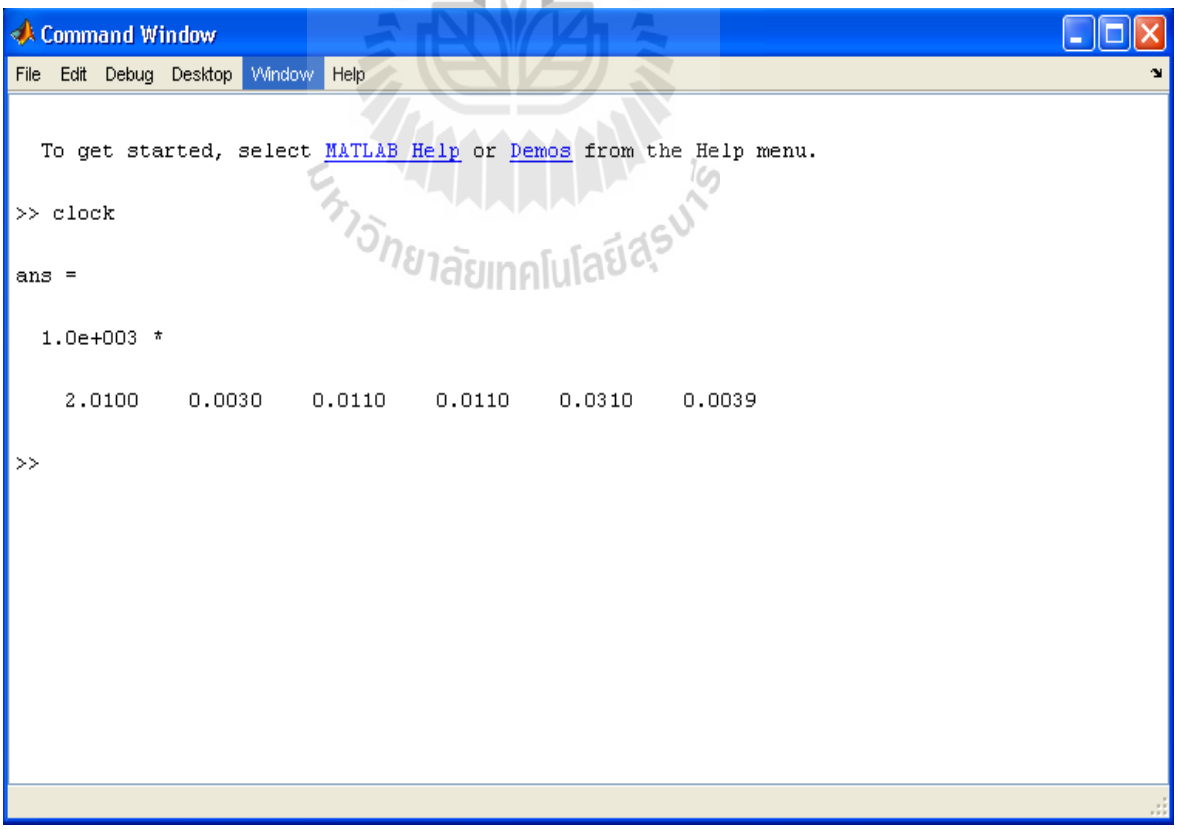
อื่นๆ โดยทั่วไปเพียงแต่จะมีรูปแบบที่ง่ายกว่าและสามารถจะเรียกใช้ function ของ MATLAB ที่มีอยู่เดิมได้ตลอดเวลา

8.ฟังก์ชันในการคำนวณทางเวลา

โปรแกรม MATLAB มีฟังก์ชันที่จัดการเกี่ยวกับเวลาเป็นจำนวนมาก คุณสามารถทำการตรวจสอบวัน เดือน และเวลาได้โดยโปรแกรม MATLAB จะเก็บข้อมูลเกี่ยวกับเวลาทั้งในปัจจุบัน อดีต และอนาคตซึ่งคุณสามารถสร้างปฏิทินส่วนตัวของคุณได้อย่างง่ายดายทั้งนี้เพราะรูปแบบของฟังก์ชั่นในการคำนวณทั้งง่าย มากแต่ค่อนข้างยากในการตีความของผลลัพธ์ที่แสดงออกมาในบางฟังก์ชั่น ซึ่งสามารถอธิบายได้ดังต่อไปนี้

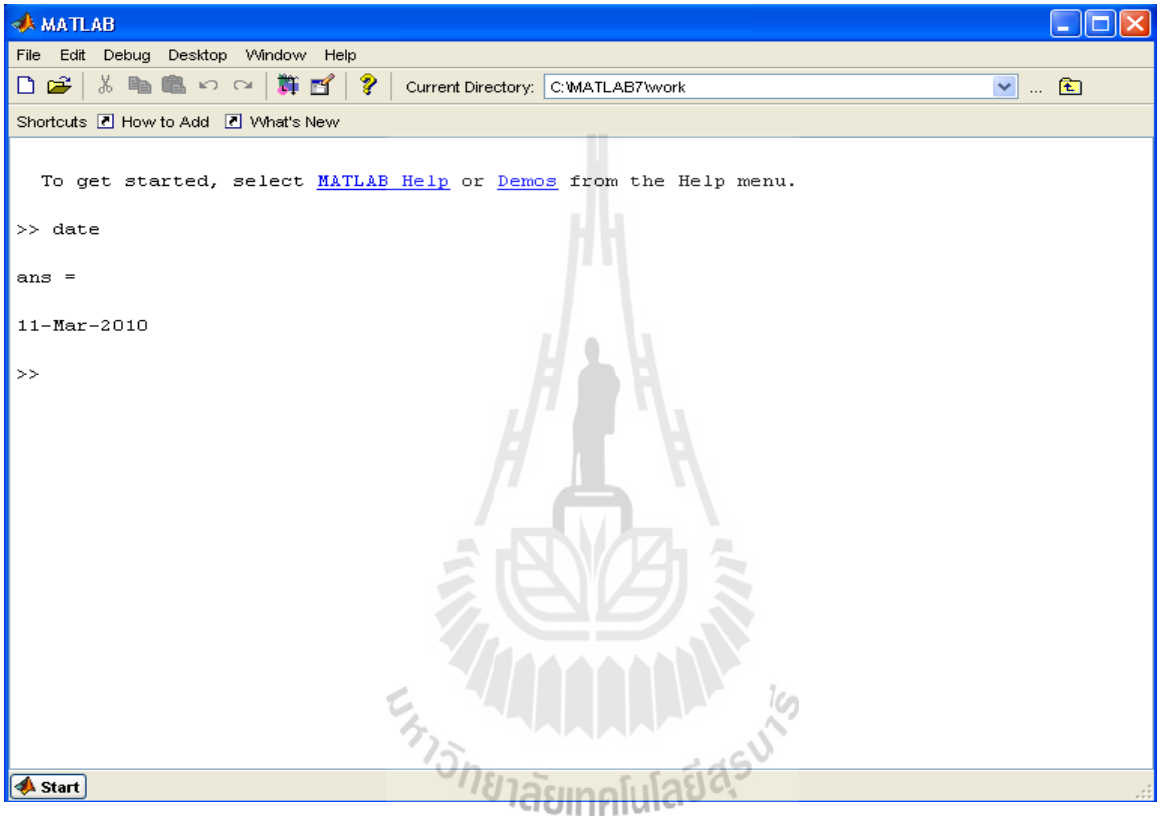
ฟังก์ชั่นเกี่ยวกับเวลาและวันเดือนปี

ฟังก์ชัน clock จะบอกถึงวันและเวลาปัจจุบันในอาร์เรย์ ตัวอย่างเช่น



คำสั่ง clock นี้จะบอกวันและเวลาปัจจุบันเป็น Time = [ปี เดือน วัน ชั่วโมง นาที วินาที] จากตัวอย่างนี้สามารถอธิบายได้ว่าวันเดือนปีเป็น 2010 เดือน 3 (มีนาคม) วันที่ 11 11 นาฬิกา 50 นาที 44.5 วินาที

ฟังก์ชั่น date จะบอกวันที่ปัจจุบันซึ่งเป็นสตริงในรูปแบบ วัน-เดือน-ปี ตัวอย่างเช่น

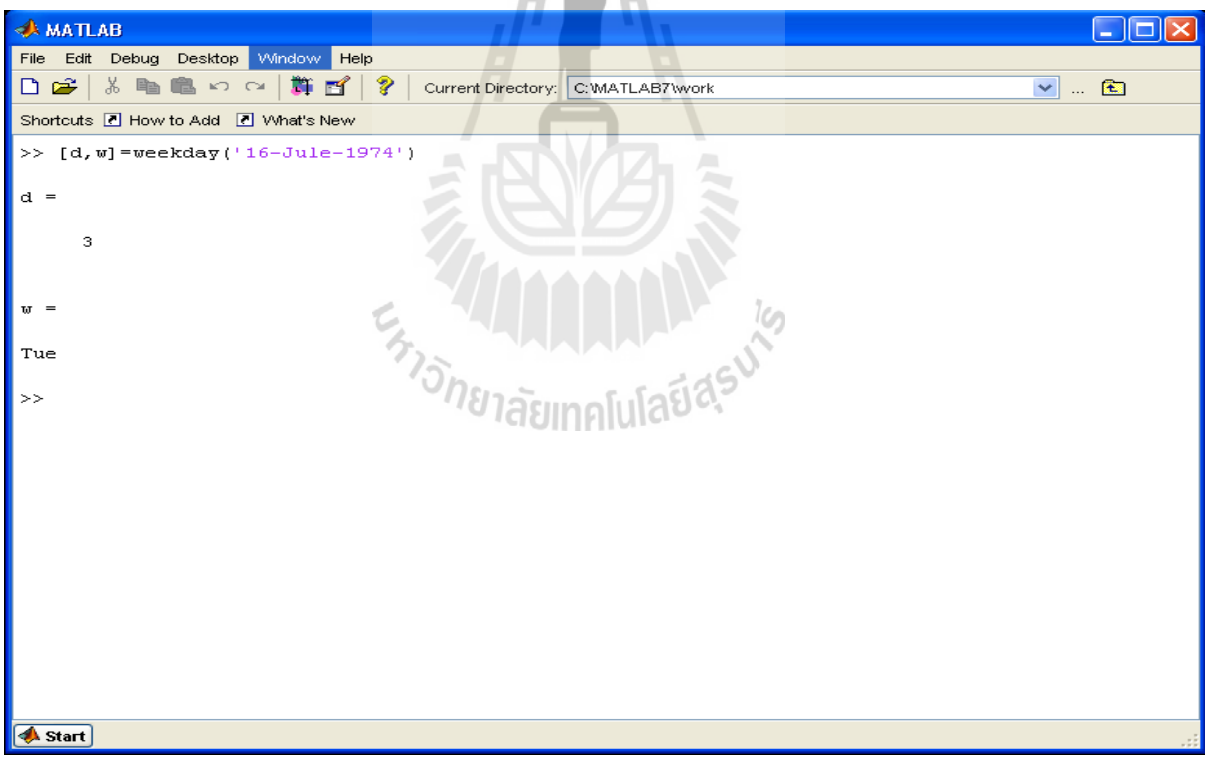


9.ฟังก์ชันวันที่

คุณสามารถหาวันต่างๆ ในสัปดาห์ได้โดยใช่คำสั่ง Weekday(“วัน-เดือน-ปี”) โดยโปรแกรม MATLAB จะทำการคำนวณและหาวันในสัปดาห์ที่คุณหาเป็นดังนี้

1. Sun
2. Mon
3. Tue
4. Wed
5. Thu
6. Fri
7. Sat

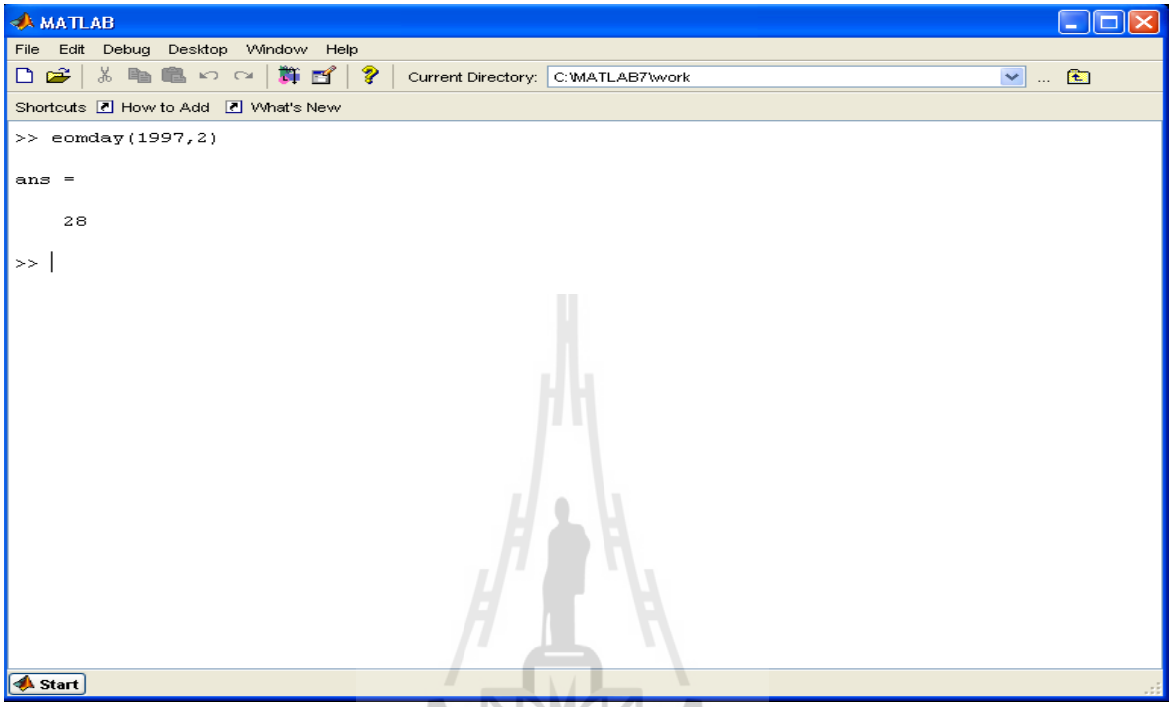
ตัวอย่างเช่น



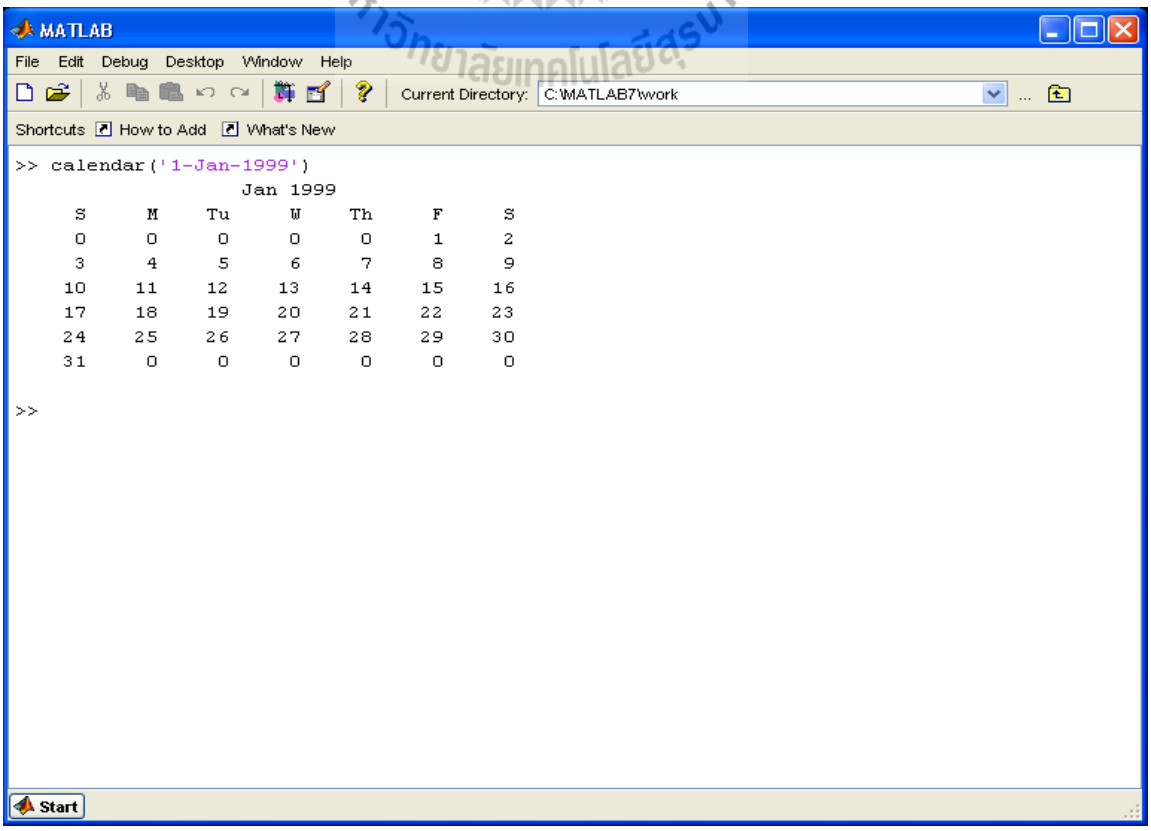
มหาวิทยาลัยเทคโนโลยีสุรนารี

คุณสามารถหาวันสุดท้ายของเดือนหรือเดือนหนึ่งมีกี่วันได้โดยการใช่คำสั่ง

Eomday (ปี. เดือน) ตัวอย่าง เช่น



โปรแกรม MATLAB สามารถสร้างปฏิทินสำหรับแต่ละเดือนที่คุณต้องการ โดยจะแสดงบนหน้าต่างคำสั่งได้ด้วยเมทริก 6x7 ด้วยการใช้คำสั่ง calendar (“เดือน / ปี”) ตัวอย่างเช่น



ภาคผนวก ข

1. โปรแกรมการประมวลผลของระบบการตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิตอล

```
clear all
close all
clc

%----------------------------------Set Timer-----------------------------------------------------------------------
t = timer('StartDelay', 2,'TimerFcn','disp("......")');
start(t)
wait(t)
ch = 0;
ch1=1;
while (ch1)

%----------------------------------Capture Image and save----------------------------------------------------------
if (ch==0)
obj = videoinput('winvideo', 1);
%preview(obj);
frame = getsnapshot(obj);
figure; imshow(frame, [])
delete(t);
date_num = datevec(datestr(now))

filename = strcat(num2str(date_num),'.jpg');
imwrite(frame,filename);

%----------------------------------Check correlation----------------------------------------------------------------
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
 [final,ch]  = check(filename);
 end
%----------------------------------Timing 5 minute-------------------------------------------------------------
 if (ch==1)
t = timer('StartDelay',300,'TimerFcn','disp("...5min....")');
start(t)
  wait(t)
  delete(t);
disp('1')

%---------------------------------- Capture Image and save ----------------------------------------------------
obj = videoinput('winvideo', 1);
frame = getsnapshot(obj);
figure; imshow(frame, [])
delete(t);
date_num = datevec(datestr(now))
filename = strcat(num2str(date_num),'.jpg');
imwrite(frame,filename);

%---------------------------------- Check correlation and  Beep silent-----------------------------------------
 [final,ch]  = check(filename);
if (ch==1)
silent_sound
 ch1=0;
end
end
disp('2')
ch=0;
end
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

2. โปรแกรมการประมวลผลภาพดิจิตอล

```
%---------------------------------- Read in the original image----------------------------------------------------------
        function [final,ch] = check(filename)
        w = imread(filename);
        w = im2double(w);
        sizw = size(w);
        figure,
        imshow(w)
        title('Input Image')

%---------------------------------- Read in the target image-----------------------------------------------------------
        t = imread('Cropped_Image.jpg');
        t = im2double(t);
        sizt = size(t);
        figure,
        imshow(t)
        title('Target Image')

%---------------------------------- Gray scale image-------------------------------------------------------------------
        ww = rgb2gray(w);
        tt = rgb2gray(t);

%---------------------------------Edge detection----------------------------------------------------------------------
        tedge = edge(tt,'sobel');
        wedge = edge(ww,'sobel');
        figure,
        imshow(tedge);
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
%---------------------------------Correlation by filter-------------------------------------------------------------
        out = filter2(tedge,wedge);
        figure,
        imshow(out);

%---------------------------------Calculation find value max,min---------------------------------------------------
        o = max(max(out));
        ch = 0;
        final=0;
        if (o>50)
        output = (1/o)*out;
        pixel = find(output == 1);
        pcolumn = fix(pixel / sizw(1));
        prow = mod(pixel,sizw(1));
        rdis = fix(sizt(1)/2);
        cdis = fix(sizt(2)/2);
        cmin = pcolumn - cdis;
        cmax = pcolumn + cdis;
        rmin = prow - rdis;
        rmax = prow + rdis;
        c = [cmin cmin cmax cmax];
        r = [rmin rmax rmax rmin];

%---------------------------------Cut position max,min-------------------------------------------------------------
        m = roipoly(ww,c,r);
        m = im2double(m);
        m = 0.5 * (m + 1);

%---------------------------------Maks point RGB-------------------------------------------------------------------
        mask(:,:,1) = m;
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
mask(:,:,2) = m;
mask(:,:,3) = m;


%-------------------------Compare original image with Maks point RGB image-------------------------------
final = mask .* w;
figure,
imshow(final)
title('Result Image')

%-------------------------Show original image campare  result Image ----------------------------------------------
subplot(1,2,1)
imshow(w)
title('Input Image')
subplot(1,2,2)
imshow(final),
title('Result Image')
rectangle('Position',[cmin,rmin,cmax-cmin,rmax-rmin],'EdgeColor','r');
imwrite(final,'Image.jpg');
ch = 1;
end
```

3. โปรแกรมเสียงเตือน

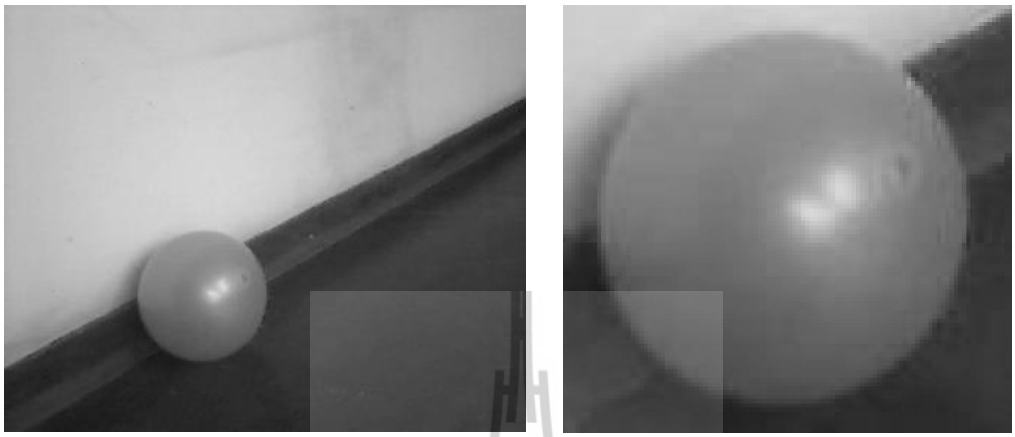
```
function silent_sound
a = wavread('silent.wav');
wavplay(a);
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

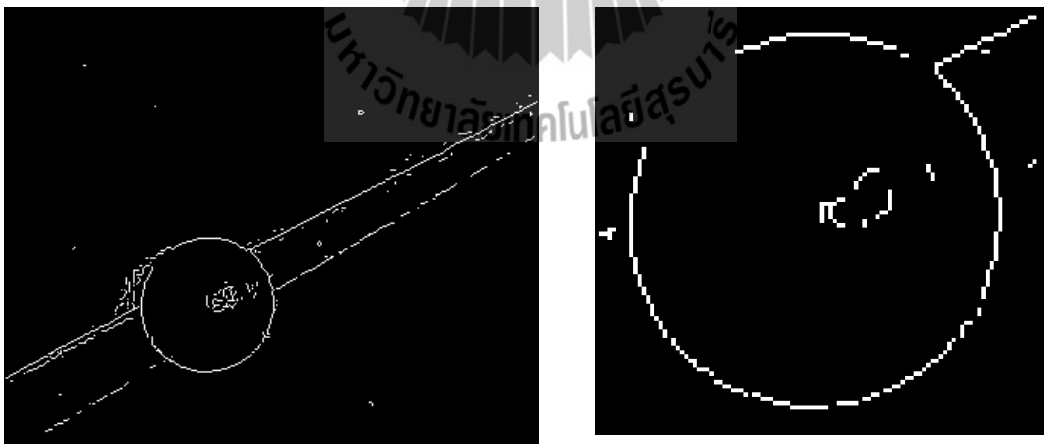
แสดงตัวอย่างภาพก่อนนำไปประมวลผล

แสดงตัวอย่างภาพที่ทำการ Crop

มหาวิทยาลัยเทคโนโลยีสุรนารี



แสดงตัวอย่างภาพต้นฉบับกับภาพเป้าหมายที่ทำเป็นระดับสีเทา(Gray Scale)



แสดงตัวอย่างภาพต้นฉบับที่ทำการหาขอบภาพ (edge detection) ภาพต้นฉบับกับภาพเป้าหมาย

แสดงตัวอย่างภาพในการทำ correlation โดยการ filter

มหาวิทยาลัยเทคโนโลยีสุรนารี

แสดงตัวอย่างภาพที่ทำการตัดจากตำแหน่ง max , min ที่ทำการคำนวณมาได้

แสดงตัวอย่างภาพผลลัพธ์ที่ได้

มหาวิทยาลัยเทคโนโลยีสุรนารี

ประวัติผู้เขียน

นางสาวสุนิสา เรืองศรี เกิดเมื่อวันที่ 1 มีนาคม พ.ศ. 25 30 ภูมิลำเนาอยู่ที่ ตำบลหนองผักนาก อำเภอสามชุก จังหวัดสุพรรณบุรี สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนสามชุกรัตนโภคาราม อำเภอสามชุก จังหวัดสุพรรณบุรี เมื่อปี พ.ศ. 254 8 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นางสาวสุวารี ศรีอำพรรณ์ เกิดเมื่อวันที่ 13 มกราคม พ.ศ. 25 30 ภูมิลำเนาอยู่ที่ ตำบลสามชุก อำเภอสามชุก จังหวัดสุพรรณบุรี สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนสามชุกรัตนโภคาราม อำเภอสามชุก จังหวัดสุพรรณบุรี เมื่อปี พ.ศ. 254 8 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี