หุ่นยนต์เคลื่อนที่แบบวิทัศน์โดยหลักการฟัซซี่ซิงเกิลตัน

Image Processing based Mobile Robot using Fuzzy Singleton

นายวุฒิพงศ์	หยองใหญ่	รหัส 45360401
นางสาวสุภาพร	ไชยชาญ	รหัส 45360518
นายอนรรฆพล	แสนทน	รหัส 45360559

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 / W.A. 2553 /
เลขทะเบียน 15010202
เลขเรียกหนังสือ วิ6ว7ท 2548
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	หุ่นยนต์เคลื่อนที่แบบวิทัศน์โดยหลักการฟัซซี่ชิงเกิลตัน		
ผู้ดำเนินโครงงาน	นายวุฒิพงศ์	หยองใหญ่	รหัส 45360401
	นางสาวสุภาพร	ไชยชาญ	รหัส 45360518
	นายอนรรฆพล	แสนทน	รหัส 45360559
อาจารย์ที่ปรึกษา	ดร. สมยศ	เกียรติวนิชวิไล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.สมยศ เกียรติวนิชวิไล)

.....กรรมการ
(ดร.สุรเชษฐ์ กานต์ประชา)

.....กรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ
(อาจารย์พนัส นัถฤทธิ์)

หัวข้อโครงงาน	หุ่นยนต์เคลื่อนที่แบบวิทัศน์โดยหลักการฟัซซี่ซิงเกิลตัน		
ผู้ดำเนินโครงงาน	นายวุฒิพงศ์	หยองใหญ่	รหัส 45360401
	นางสาวสุภาพร	ไชยชาญ	รหัส 45360518
	นายอนรรฆพล	แสนทน	รหัส 45360559
อาจารย์ที่ปรึกษา	ดร. สมยศ เกียรติวนิชวิไล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

บทคัดย่อ

การศึกษาโครงงานนี้แสดงให้เห็นถึงประสิทธิภาพในการใช้ทฤษฎีฟัซซี่ซิงเกิลตันเพื่อควบคุมการเคลื่อนที่ผ่านการสื่อสารไร้สายของหุ่นยนต์ ในขั้นแรกกล้องเว็บแคมที่ถูกกำหนดตำแหน่งการวางกล้องที่แน่นอนจะจับภาพซึ่งครอบคลุมพื้นที่ที่ใช้ในการวิเคราะห์ตำแหน่งของหุ่นยนต์, สิ่งกีดขวางและเป้าหมาย จากนั้นข้อมูลจะถูกส่งมาคำนวณค่าฟัซซี่ เพื่อใช้ในการตัดสินใจเคลื่อนที่หุ่นยนต์โดยส่งคำสั่งผ่านการสื่อสารไร้สายไปสู่หุ่นยนต์ จากการทดลองหุ่นยนต์จะสามารถเคลื่อนที่หลบหลีกสิ่งกีดขวางและเข้าสู่เป้าหมายได้ ทั้งในสิ่งแวดล้อมที่คงที่ (เป้าหมายและสิ่งกีดขวางอยู่ในตำแหน่งคงที่ตลอดการทดลอง) หรือสิ่งแวดล้อมที่มีการเคลื่อนที่ (เป้าหมายและสิ่งกีดขวางมีการเคลื่อนที่ในขณะที่หุ่นยนต์ทำงาน)

Project title	Image Processing based Mobile Robot using Fuzzy Singleton
Name	Mr. Wuttipong Yongyai ID. 45360401
	Miss Supaporn Chaichan ID. 45360518
	Mr. Anakkapon Seanton ID. 44360559
Project advisor	Dr. Somyot Kiattivanichvilai
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2005

.....

Abstract

This study demonstrates the effectiveness of using fuzzy singleton technique to control the movement of a wireless, mobile robot. First, a remote global fixed camera takes the images of the coverage area for analyzing for the position of the robot, obstacles, and targets. Next, this information is fed to the fuzzy singleton decision making module which decides the movement of the robot and wirelessly sends the instruction to the robot. Based on our experiment, the robot was able to reach the target while avoiding the obstacles in both static configuration (obstacles and target stayed at the same position throughout the experiment) or in a dynamic environment (obstacles and target were moved while the robot was moving)

กิตติกรรมประกาศ

ขอขอบพระคุณ ดร. สมยศ เกียรติวนิชวิไล อาจารย์ที่ปรึกษา และดร.สุรเชษฐ์ กานต์ประชา อาจารย์ที่ปรึกษาร่วม ที่คอยให้คำปรึกษา ความช่วยเหลือตลอดจนคำแนะนำต่างๆในการทำโครงงานชิ้นนี้ และขอขอบคุณ นายเศรษฐา ตั้งค้าวานิช ประธานชมรมโรบอท มหาวิทยาลัยนเรศวร ที่คอยให้ความช่วยเหลือและสนับสนุนอุปกรณ์ต่างๆในการสร้างหุ่นยนต์ สุดท้ายต้องขอขอบพระคุณอาจารย์ทุกท่านและเพื่อนๆทุกคนที่ยังไม่ได้เอ่ยนามที่ให้การสนับสนุนผู้จัดทำโครงงานให้สามารถทำโครงงานชิ้นนี้จนสำเร็จลุล่วงไปได้ด้วยดี

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบข่ายของโครงงาน 1
1.4 ขั้นตอนของการดำเนินงาน 2
1.5 แผนการดำเนินงาน 3
1.6 ผลที่คาดว่าจะได้รับ 4
1.7 งบประมาณของโครงงาน 4

บทที่ 2 หลักการและทฤษฎี

2.1 ทฤษฎีพื้นฐานของฟัซซี่ลอจิก 5
2.2 การทำงานของฟัซซี่ลอจิก 10
2.3 กฎของฟัซซี่ (Fuzzy Rules) 10
2.4 กระบวนการทำงานของฟัซซี่ลอจิก 11
2.5 กฎของฟัซซี่ซิงเกิลตัน (fuzzy singleton rule) 11
2.6 พื้นฐานทั่วไปเกี่ยวกับการประมวลผลภาพ 12
2.7 การสื่อสารข้อมูลแบบไร้สาย (Wireless) 15
2.8 การขับเคลื่อนมอเตอร์แบบกระแสตรง 16

สารบัญ (ต่อ)

บทที่ 3 วิธีการดำเนินการ

3.1 การควบคุมการเคลื่อนที่ของหุ่นยนต์จากคอมพิวเตอร์โดยการประมวลผลจากภาพ 20

3.2 การเขียนโปรแกรมของระบบควบคุมการตัดสินใจในการหลบหลีกสิ่งกีดขวางของหุ่นยนต์โดยใช้ทฤษฎีฟัซซี่ 21

3.3 การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกลตัน 25

3.4 การออกแบบโปรแกรมควบคุมหุ่นยนต์เคลื่อนที่แบบวิทัศน์โดยหลักการฟัซซี่ซิงเกลตัน ... 29

3.5 การเขียนโปรแกรมประมวลผลภาพ 30

3.6 การสร้างตัวหุ่นยนต์ 32

บทที่ 4 ผลการทดลอง

4.1 ผลการทดลองการเคลื่อนที่หลบหลีกสิ่งกีดขวางเข้าสู่เป้าหมายของหุ่นยนต์ 34

บทที่ 5 สรุปผล

5.1 ผลการทดลอง 43

5.2 ปัญหาและแนวทางแก้ไข 44

5.3 สรุปผลการทดลอง 45

5.4 ข้อเสนอแนะ 45

บรรณานุกรม 46

ภาคผนวก ก 47

ภาคผนวก ข 68

ภาคผนวก ค 71

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงจำนวนสีที่เป็นไปได้ซึ่งขึ้นกับจำนวนบิต 14

2.2 การทำงานของมอเตอร์เมื่อป้อนอินพุตในลักษณะต่างๆ 19

3.1 ตารางแสดงกฎของฟิชชี่ 29

สารบัญรูป

รูปที่		หน้า
2.1	แสดงการอ่านค่าฟังก์ชั่นจำนวนสมาชิก	6
2.2	ฟังก์ชันสมาชิกฟัซซี่ที่มีขอบเขตฟัซซี่เป็นฟังก์ชันรูปสามเหลี่ยม	7
2.3	เซตของฟัซซี่ลอจิกกับขอบเขตร่วม	7
2.4	ค่าภาวะสมาชิกฟัซซี่ที่แสดงโดยค่าเฉพาะ	8
2.5	การใช้โอเปอเรเตอร์ของฟัซซี่ลอจิก	9
2.6	ภาพแสดงการทำงานของฟัซซี่	10
2.7	ฟังก์ชันภาวะสมาชิกของฟัซซี่ที่มีขอบเขตเป็นฟังก์ชันของรูปสามเหลี่ยม	11
2.8	โหมดสี RGB	14
2.9	แสดงลักษณะของสัญญาณ FREQUENCY MODULATION	16
2.10	แสดงวงจรขยายแบบ H-BRIDGE	17
2.11	แสดงการไหลของกระแสไฟฟ้าผ่านทางขั้วบวกของมอเตอร์ทำให้มอเตอร์หมุนไปในทิศทางไปข้างหน้า	17
2.12	แสดงการไหลของกระแสไฟฟ้าผ่านทางขั้วลบของมอเตอร์ ทำให้มอเตอร์หมุนไปในทิศทางย้อนกลับ	18
2.13	รูปแสดงไอซีเบอร์ TA7279P	18
3.1	แสดงการทำงานการควบคุมหุ่นยนต์เคลื่อนที่	20
3.2	ช่วงขององศาที่มีผลต่อการตัดสินใจหลบหลีกสิ่งกีดขวางของหุ่นยนต์	21
3.3	การแบ่งขอบเขตของอินพุต	22
3.4	กราฟของตัวแปรอินพุต	22
3.5	แสดงตำแหน่งของสิ่งกีดขวาง	24
3.6	แผนภาพแสดงการทำงานของโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์	26
3.7	แผนภาพการทำงานของโปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยฟัซซี่ซิงเกลตัน	27
3.8	แผนภาพแสดงการประมวลผลภาพหาตำแหน่งวัตถุ	28
3.10	แสดงการหาตำแหน่งและรัศมีของวัตถุ	31
3.11	แสดงการตัด NOISE ออกจากระบบการประมวลผลภาพ	31

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.12	แสดงการประมวลผลภาพของสิ่งกีดขวางที่มากกว่า 1 ชิ้น	32
3.13	แสดงวิธีการสร้างตัวหุ่นยนต์	32
3.14	ภาพหุ่นยนต์	33
4.1	ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 1 ชิ้น	35
4.2	ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 2 ชิ้น	36
4.3	ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 3 ชิ้น	37
4.4	ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 4 ชิ้น	38
4.5	ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 5 ชิ้น	39
4.6	ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 6 ชิ้น	40
4.7	ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 7 ชิ้น	41
ข.1	วงจรการทำงานของ PIC16F877	68
ข.2	แสดงวงจรการส่ง OUTPUT ให้มอเตอร์ทำงาน	68
ข.3	แสดงรูปโปรแกรม MIKRO BASIC	69
ค.1	แสดงภาคส่งของสัญญาณ WIRELESS	71
ค.2	แสดงภาครับของสัญญาณ WIRELESS	71

บทที่ 1
บทนำ

ปัจจุบันเทคโนโลยีเข้ามามีบทบาทมากในชีวิตประจำวัน มนุษย์ใช้ความสามารถในการพัฒนาอุปกรณ์และเครื่องมืออำนวยความสะดวกในด้านต่าง ๆ เพื่อลดการใช้แรงงานมนุษย์ และเพิ่มความสะดวกสบายในการดำเนินชีวิต จึงก่อให้เกิดการแข่งขันกันในการพัฒนาอุปกรณ์ เครื่องมือนวัตกรรมใหม่ๆ จึงแพร่หลายอย่างรวดเร็วในระยะเวลาเพียงไม่กี่ปี หุ่นยนต์นับเป็นนวัตกรรมหนึ่งที่สำคัญและมีใช้กันอย่างแพร่หลายในประเทศที่พัฒนาแล้ว อาทิเช่น หุ่นยนต์แขนกลที่ช่วยหยิบจับวัตถุหรือทำงานแทนมนุษย์ในด้านต่างๆ การโปรแกรมควบคุมการทำงานของหุ่นยนต์ โดยมากจะกำหนดตามการตัดสินใจแบบตายตัวที่ผู้เขียนโปรแกรมกำหนดขึ้นเท่านั้น การตัดสินใจและการควบคุมในปัญหาบางประเภท จะมีความไม่เป็นเชิงเส้นสูง(Nonlinear System) ดังนั้นการออกแบบการควบคุมแบบเชิงเส้น (Linear System) จึงไม่ประสบผลสำเร็จเท่าที่ควร

1.1 ที่มาและความสำคัญของโครงงาน

ทฤษฎีฟัซซี่ซิงเกลตันเป็นทฤษฎีหนึ่งที่มีโครงสร้างการออกแบบควบคุมแบบไม่เป็นเชิงเส้น เนื่องจากเป็นทฤษฎีที่มีความคล้ายคลึงกับการตัดสินใจของมนุษย์ ดังนั้นจึงมีความเหมาะสมและน่าสนใจที่จะใช้ทฤษฎีฟัซซี่ในการควบคุมการสั่งงานการเคลื่อนที่หลบหลีกสิ่งกีดขวางของหุ่นยนต์

การหาตำแหน่งหุ่นยนต์และสิ่งกีดขวางในโครงงานนี้อาศัยการประมวลผลภาพเพื่อตำแหน่งของวัตถุที่ต้องการผ่านกล้อง Web cam โดยใช้ทฤษฎีฟัซซี่ซิงเกิลตันในการควบคุมการตัดสินใจเพื่อหลบหลีกสิ่งกีดขวาง โดยไม่ต้องใช้มนุษย์เป็นผู้ควบคุม

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อพัฒนาโปรแกรมการประมวลผลภาพ ในการหาตำแหน่งสิ่งกีดขวางและหุ่นยนต์

1.2.2 เพื่อสร้างต้นแบบหุ่นยนต์สามารถเคลื่อนที่หลบหลีกสิ่งกีดขวาง โดยใช้ทฤษฎีฟัซซี่ซิงเกิลตันได้

1.3 ขอบข่ายของโครงงาน

1.3.1 ประยุกต์โปรแกรมควบคุมการทำงานของหุ่นยนต์ในการหลบหลีกสิ่งกีดขวางและเคลื่อนที่เข้าหาเป้าหมาย ให้สามารถใช้งานได้กับหุ่นยนต์จริงด้วยหลักการฟัซซี่ซิงเกิลตัน

1.3.2 พัฒนาระบบประมวลผลภาพผ่านกล้อง Web cam เพื่อหาตำแหน่งของหุ่นยนต์และสิ่งกีดขวาง

1.4 ขั้นตอนของการดำเนินงาน

1.4.1 ศึกษาและค้นคว้าข้อมูลเกี่ยวกับทฤษฎีฟัซซี่ในส่วนที่เกี่ยวข้องกับการควบคุมการตัดสินใจของหุ่นยนต์ และศึกษาอุปกรณ์ที่เหมาะสมในการใช้ประดิษฐ์หุ่นยนต์

1.4.2 ค้นคว้าและศึกษาการเขียนโปรแกรมที่ใช้ในการควบคุมการเคลื่อนไหวของหุ่นยนต์

1.4.3 เขียนโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์สร้างตัวหุ่นยนต์ สนามทดสอบทฤษฎีและสิ่งกีดขวาง

1.4.4 เขียนโปรแกรมในส่วนของฟัซซี่ลอจิก เพื่อควบคุมการตัดสินใจของหุ่นยนต์ ในการหลบสิ่งกีดขวาง

1.4.5 ทดสอบการทำงานของหุ่นยนต์

1.4.6 แก้ไขความผิดพลาดในส่วนต่างๆของโครงงาน

1.4.7 สรุปผลการทดลองและจัดทำรูปเล่มโครงงาน

1.5 แผนการดำเนินงาน

แผนการดำเนินงานโครงงาน Vision based Mobile Robot using Fuzzy Singleton

กิจกรรม	ปี 2547		ปี2548							
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ส.ค.	ก.ย.
1. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับทฤษฎีฟัซซี่ในส่วนที่เกี่ยวข้องกับการควบคุมการตัดสินใจของหุ่นยนต์ และศึกษาอุปกรณ์ที่เหมาะสมในการใช้ประดิษฐ์หุ่นยนต์	←	→								
2. ค้นคว้าและศึกษาการเขียนโปรแกรมที่ใช้ในการควบคุมการเคลื่อนไหวของหุ่นยนต์	←	→								
3. เขียนโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์		←	→							
4. สร้างตัวหุ่นยนต์ สนามทดสอบทฤษฎี และสิ่งกีดขวาง			←	→						
5. เขียนโปรแกรมในส่วนของฟัซซี่ลอจิก เพื่อควบคุมการตัดสินใจของหุ่นยนต์ ในการหลบสิ่งกีดขวาง					←	→				
6. ทดสอบการทำงานของหุ่นยนต์							↔			
7. แก้ไขความผิดพลาดในส่วนต่างๆของโครงงาน							←	→		
8. สรุปผลการทดลองและจัดทำรูปเล่มโครงงาน									←	→

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถประยุกต์ใช้ทฤษฎีทาง การประมวลผลภาพ ในการหาตำแหน่งของสิ่งกีดขวางและเป้าหมาย

1.6.2 สามารถประยุกต์ใช้ทฤษฎีฟัซซี่ลอจิกในการควบคุมการหลบหลีกสิ่งกีดขวาง

1.6.3 สามารถสร้างหุ่นยนต์ที่สามารถหลบสิ่งกีดขวางได้จริง

1.7 งบประมาณของโครงงาน

1.7.1 ค่าถ่ายเอกสารและค่าเข้าเล่มโครงงาน

1.7.2 ค่าอุปกรณ์ Hardware

1.7.4 ค่าหมึกพิมพ์

รวมเป็นเงิน 3,000 บาท(สามพันบาทถ้วน)

มหาวิทยาลัยนเรศวร

บทที่ 2
หลักการและทฤษฎี

ฟัซซี่ลอจิก เป็นสาขาหนึ่งของคณิตศาสตร์ และเป็นวิธีการทางตรรกศาสตร์ ที่มีการเชื่อมโยงระบบการคำนวณทั้งสองระบบเข้าด้วยกัน ซึ่งระบบหนึ่งมีผลการคำนวณหรือมีวิธีการคำนวณเป็นไปตามกฎทางคณิตศาสตร์กฎใดกฎหนึ่งอย่างชัดเจน และอีกระบบมีผลการคำนวณหรือมีวิธีการคำนวณ โดยการผสมผสานกฎทางคณิตศาสตร์เข้าด้วยกัน

2.1 ทฤษฎีพื้นฐานของฟัซซี่ลอจิก

ฟัซซี่ลอจิก ถูกคิดค้นโดย Dr.Lotfi Zadeh ในปี ค.ศ.1965 โดยมีการใช้คอมพิวเตอร์เข้ามาช่วยในการตัดสินใจ ซึ่งจะจำลองการตัดสินใจแบบสมองของมนุษย์ เนื่องจากมนุษย์จะตัดสินใจอะไร ย่อมมีปัจจัยหลายอย่างที่เข้ามาเกี่ยวข้อง และมีกระบวนการคิดที่ซับซ้อนและไม่แน่นอน ตัวอย่างเช่น "หนาว" เป็นคำพูดที่แสดงออกมาทางภาษาที่เข้าใจ โดยไม่พูดเป็นคำพูดที่แน่นอน เช่น หนาวที่อุณหภูมิ 0 อาศา จะเห็นได้ว่า เป็นการตัดสินใจแบบคลุมเครือ (Shades of gray) ไม่ใช่การตัดสินใจแบบเจาะจงว่า "ใช่" หรือ "ไม่ใช่" อย่างไรก็ตามการตัดสินใจนี้สามารถนำไปใช้ในการควบคุมระบบไม่เป็นเชิงเส้นได้ดี

ระบบฟัซซี่ลอจิก สามารถทำการคำนวณทางคณิตศาสตร์จากคอมพิวเตอร์ และลักษณะของฟัซซี่ จะมีความใกล้เคียงภาษามนุษย์มากกว่าที่เป็นแค่ตัวเลขตายตัว ทฤษฎีต่อไปนี้อธิบายถึงพื้นฐานของฟัซซี่ที่จำเป็นในโครงงานนี้

2.1.2 เซตของฟัซซี่ (Fuzzy sets)

เซตของฟัซซี่มีลักษณะคล้ายกับเซตในทฤษฎีเซต แตกต่างกันที่ทฤษฎีฟัซซี่จะมีค่าจำนวนสมาชิก ซึ่งค่าจำนวนสมาชิกเป็นตัวเลขจำนวนจริงจะอยู่ในช่วงเวลาระหว่าง [0, 1]

กำหนดให้ U เป็นเซตเอกภพ และ A เป็นสับเซตของฟัซซี่ถ้า A คือ เซตของคู่อันดับของสมาชิก $x \in U$ และค่าของฟังก์ชันภาวะสมาชิก (membership function)

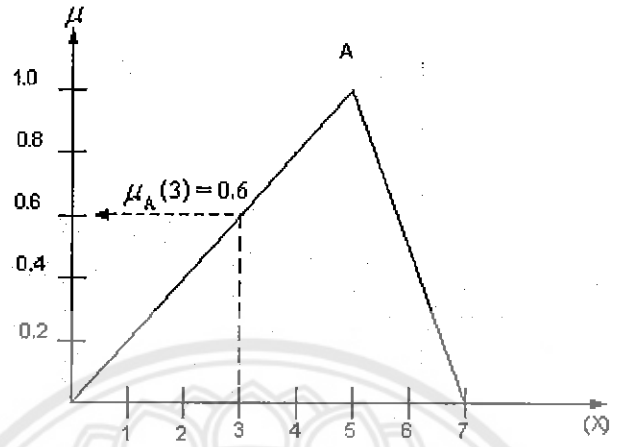
$$\mu_A(X) \in [0, 1]$$

กล่าวคือ

$$A = \{(x, \mu_A(x)): x \in U\} \qquad (2.1)$$

2.1.2 ฟังก์ชันภาวะสมาชิก

ฟังก์ชั่นภาวะสมาชิกเป็นฟังก์ชั่นที่ใช้นิยามอินพุตเพื่อหาค่าสภาวะสมาชิก โดยฟังก์ชันภาวะสมาชิกมีค่าระหว่าง 0 ถึง 1 ดังรูปที่ 2.1



รูปที่ 2.1 แสดงการอ่านค่าฟังก์ชั่นจำนวนสมาชิก

ตัวอย่างเช่น $\mu_A(x)$ หมายถึง ฟังก์ชั่นภาวะสมาชิกของ x ในเซต A ดังแสดงในรูปที่ 2.1

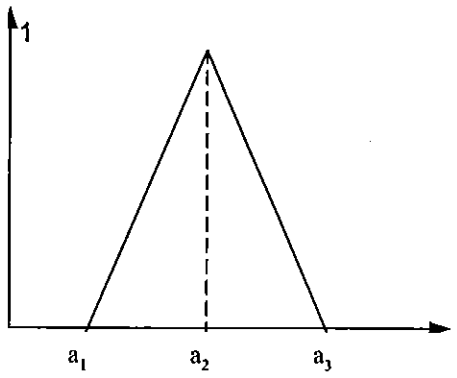
$\mu_A(3)$ หมายถึง ฟังก์ชันภาวะสมาชิกของอินพุต x = 3 ในเซต A ค่า $\mu_A(3)$ จะเท่ากับ 0.6

2.1.2 ขอบเขตฟัซซี่ (Fuzzy regions)

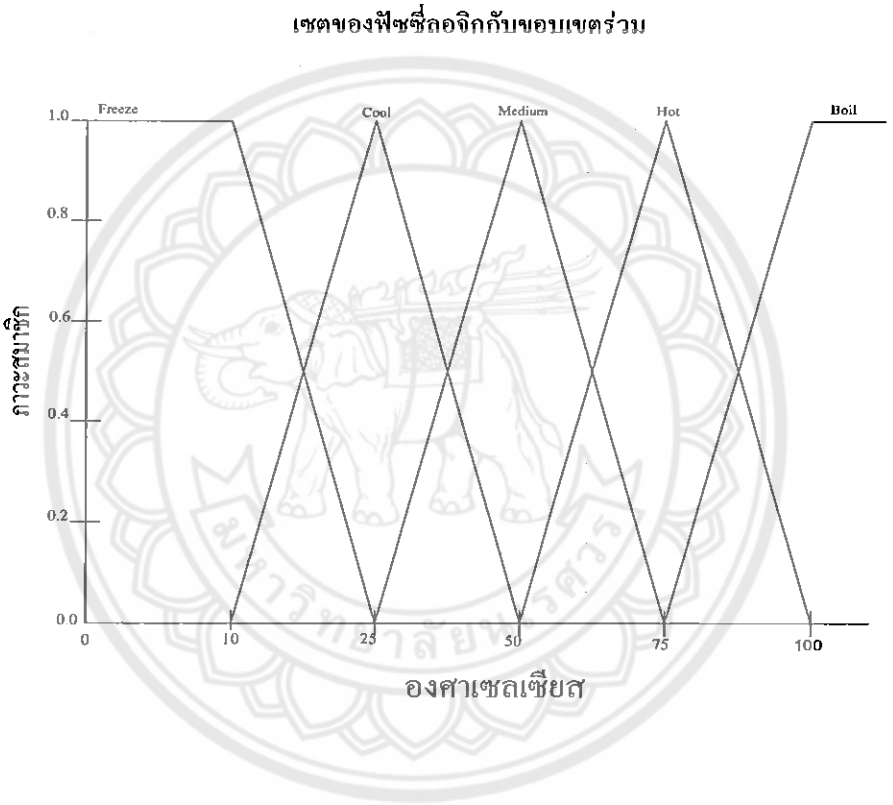
ขอบเขตฟัซซี่ใช้แสดงถึงความสัมพันธ์ระหว่างค่าภาวะสมาชิกกับฟัซซี่เซต ขอบเขตฟัซซี่ไม่จำเป็นต้องมีการเหลื่อมล้ำหรือซ้อนทับกันแต่จะต้องสอดคล้องกับกฎของฟัซซี่

รูปแบบของขอบเขตฟัซซี่สามารถนิยามได้โดยฟังก์ชันภาวะสมาชิก ถึงแม้ว่ารูปของขอบเขตฟัซซี่สามารถใช้ฟังก์ชันที่มีความซับซ้อน เช่น ฟังก์ชั่นระฆังคว่ำ ฟังก์ชั่นสามเหลี่ยม ฯลฯ ได้ อย่างไรก็ตามฟังก์ชั่นที่ง่ายต่อการคำนวณ เช่น ฟังก์ชันรูปสามเหลี่ยม ดังรูปที่ 2.2 ซึ่งเป็นที่นิยม เนื่องจากเป็นฟังก์ชันที่ไม่ซับซ้อนและใช้งานได้จริง สมการต่อไปนี้แสดงถึงฟังก์ชั่นสามเหลี่ยม

$$\mu_A(x) = \begin{cases} (x-a_1)/(a_2-a_1) & \text{ถ้า } a_1 \le x < a_2 \\ (a_3-x)/(a_3-a_2) & \text{ถ้า } a_2 \le x < a_3 \\ 0 & \text{ถ้า } x \le a_1 \text{ หรือ } x > a_3 \end{cases} \qquad (2.3)$$



รูปที่ 2.2 ฟังก์ชันสมาชิกฟัซซี่ที่มีขอบเขตฟัซซี่เป็นฟังก์ชันรูปสามเหลี่ยม[2]



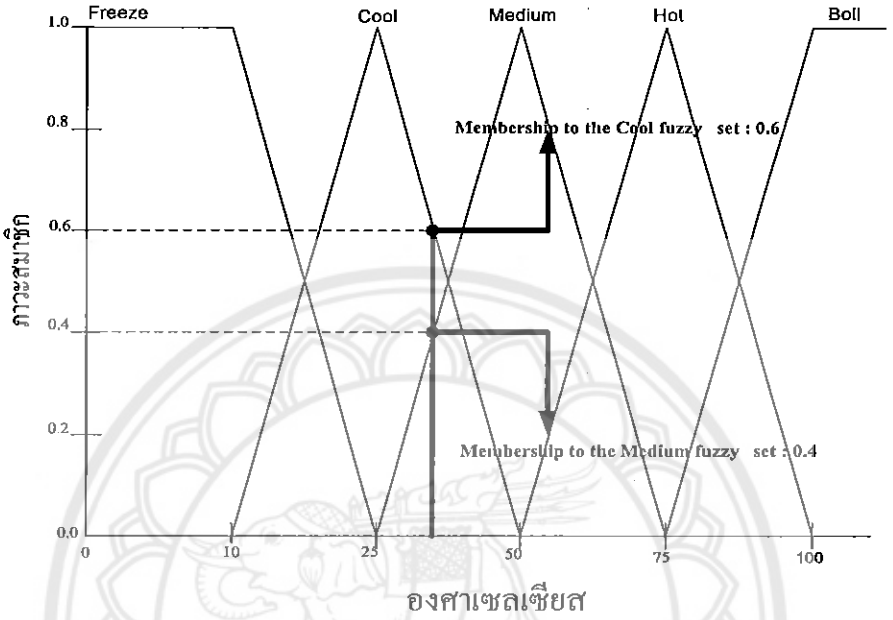
รูปที่ 2.3 เซตของฟัซซี่ลอจิกกับขอบเขตร่วม

จากรูป 2.3 ถ้าพิจารณาที่อุณหภูมิ 35 องศาเซลเซียส จะได้ค่าภาวะสมาชิกค่าดังต่อไปนี้ $\mu_{cool}(35)$ หมายถึงที่ อุณหภูมิ 35 องศาเซลเซียส ค่าภาวะสมาชิกของอุณหภูมิน้ำเย็น(Cool) ของเซตอุณหภูมิมีค่าเป็น 0.6 ดังรูป 2.4

$\mu_{medium}(35)$ หมายถึงที่ อุณหภูมิ 35 องศาเซลเซียส ค่าภาวะสมาชิกของอุณหภูมิน้ำอุ่น(medium) ของเซตอุณหภูมิมีค่าเป็น 0.4 ดังรูป 2.4

$\mu_{freeze}(35) = \mu_{hot}(35) = \mu_{boil}(35) = 0$ หมายถึงที่ อุณหภูมิ 35 องศาเซลเซียส ค่าภาวะสมาชิกของอุณหภูมิน้ำแข็ง(freeze), น้ำร้อน(hot) และอุณหภูมิของน้ำเดือด(boil) จะมีค่าเป็นศูนย์

เซตของภาวะสมาชิกกับอุณหภูมิของน้ำ



รูปที่ 2.4 ค่าภาวะสมาชิกฟัซซี่ที่แสดงโดยค่าเฉพาะ

2.1.4 ตัวแปรเชิงภาษา (linguistic variable)

ตัวแปรเชิงภาษา เป็นตัวแปรที่สร้างขึ้นเพื่อแทนความรู้สึก โดยการนำความรู้สึกเหล่านั้นมาระบุค่าเป็นเลขฟัซซี่

ตัวอย่างเช่น ถ้าให้ค่าของอุณหภูมิของน้ำอยู่ในช่วงระหว่าง 0 – 100 องศาเซลเซียส ตัวเลขตั้งแต่ 0 – 100 องศา สามารถแสดงความรู้สึกได้หลากหลายออกไป มีค่าที่เป็นไปได้ดังนี้

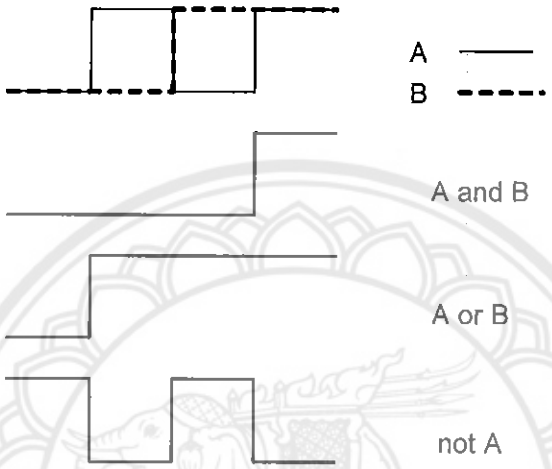
น้ำเย็น , น้ำอุ่น , น้ำร้อน

จะเห็นได้ว่า อุณหภูมิของน้ำที่แตกต่างออกไปในแต่ละองศา สามารถแสดงค่าตัวแปรเชิงภาษาได้แตกต่างกันออกไป ขึ้นอยู่กับความรู้สึกมนุษย์ ซึ่งความรู้สึกเหล่านั้น สามารถแสดงเป็นค่า

ของเลขฟัซซี่ได้ เช่น ที่อุณหภูมิ 100 องศา หมายถึง "จุดเดือด" ในสภาวะนี้ทางลอจิก จะมีค่าเป็น 1 ซึ่งค่าของ "จุดเดือด" นั้นจะเรียกว่า Antecedent ในขณะเดียวกัน น้ำที่อุณหภูมิ 0 องศา จะมีค่าทางลอจิกเป็น 0 ฟัซซี่ลอจิกแสดงค่าการเปลี่ยนของตัวเลขทางภาษานี้ให้มีค่าที่อยู่ระหว่าง 0 – 1 ได้

2.1.5 โอเปอเรเตอร์ของฟัซซี่

เป็นการรวมเซตของฟัซซี่เข้าด้วยกันโดยโอเปอเรเตอร์ของฟัซซี่ซึ่งได้แก่ AND, OR และ NOT ถ้านำ A และ B ที่เป็นเซตของฟัซซี่และเป็น Antecedent มารวมกันโดยโอเปอเรเตอร์ลอจิก AND, OR และ NOT ตามลำดับ สามารถแสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 การใช้โอเปอเรเตอร์ของฟัซซี่ลอจิก[2]

จะเห็นว่า A AND B คือการเลือกใช้ค่าน้อยที่สุดระหว่าง A หรือ B ส่วน A OR B คือการเลือกใช้ค่าสูงสุดระหว่าง A หรือB สุดท้ายการ NOT A คือการเลือกใช้ค่าตรงกันข้ามกับ A

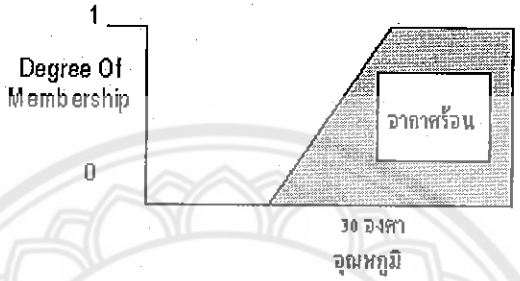
2.1.6 ค่าเฉลี่ยและค่ากลางของฟัซซี่

ตำแหน่งของเซตของฟัซซี่สามารถอธิบายได้โดยค่าเฉลี่ยของฟัซซี่ (fuzzy mean) หรือ M บางครั้งอาจเรียกว่าจุดศูนย์ถ่วง (Center of gravity) หรือ centroid ถ้าค่ากลางของฟัซซี่ของเซตของฟัซซี่ A ถูกนิยามบนเส้นจำนวนจริง M(A) สามารถหาได้โดยตรงจาก

$$M(A) = \frac{\int_{-\infty}^{+\infty} t\mu_A(t)dt}{\int_{-\infty}^{+\infty} \mu_A(t)dt} \qquad (2.3)$$

2.2 การทำงานของฟัซซี่ลอจิก

ฟัซซี่ลอจิกทำงานโดยอาศัยสับเซตของฟัซซี่ ซึ่งจะมีค่าความเป็นสมาชิก ตัวอย่างเช่น "อากาศร้อน" หมายความว่า เซตนี้จะประกอบด้วยอุณหภูมิของอากาศ ซึ่งอาจหมายถึงว่า เมื่ออุณหภูมิมากกว่า 30 องศา แล้ว "อากาศร้อน" มีค่าความเป็นสมาชิกเท่ากับ 1 อย่างไรก็ตามที่อุณหภูมิ 29 องศา อาจจะยังรู้สึกร้อนอยู่ เพราะอุณหภูมิต่างกันไม่มาก เมื่อเทียบกับเซตของฟัซซี่แล้ว ค่าสมาชิกของเซตจะอยู่ในช่วง 0 – 1 ซึ่งอาจกำหนดเป็นฟังก์ชั่นลาดเอียงของกราฟที่เปลี่ยนแปลงจาก "อากาศเย็น" และ "อากาศร้อน" ดังรูป 2.6



รูปที่ 2.6 ภาพแสดงการทำงานของฟัซซี่

จากรูป แสดงให้เห็นว่า เซตของฟัซซี่ จะแบ่งระดับระหว่าง "ต่ำ" กับ "สูง" ออกเป็นช่วงก่อนหน้า 30 องศา และหลังช่วง 30 องศา

2.3 กฎของฟัซซี่ (Fuzzy Rules)

กฎจะถูกออกแบบเพื่ออธิบายและควบคุมการทำงานของฟัซซี่ อย่างไรก็ตาม การออกแบบกฎในฟัซซี่ลอจิกจะอาศัยประสบการณ์ของผู้ออกแบบเอง

กฎฟัซซี่ประกอบด้วยประพจน์ $A_{i,k}$ ในรูปแบบของเซตของฟัซซี่ที่มีค่าฟังก์ชั่นภาวะสมาชิกส่วนนี้เรียก Antecedent และผลที่เกิดตามมาภายหลัง (Consequence) B_i ซึ่งเป็นเซตของฟัซซี่เช่นกัน

$$\text{If } a_1 \text{ is } A_{i,1} \circledcirc a_2 \text{ is } A_{i,2} \circledcirc \ldots \circledcirc a_k \text{ is } A_{i,k} \text{ then } B_i \qquad (2.4)$$

สัญลักษณ์โอเปอร์เรเตอร์ $\circledcirc$ ในที่นี้อาจเป็น AND หรือ OR ประโยค "a_k is $A_{i,k}$" อาจเขียนแทนด้วย $A_{i,k}$ เพื่อความง่าย ดังนี้

$$\text{If } A_{i,1} \circledcirc A_{i,2} \circledcirc \ldots \circledcirc A_{i,k} \text{ then } B_i \qquad (2.5)$$

2.4 กระบวนการทำงานของฟัซซี่ลอจิก

1. การทำให้เป็นฟัซซี่ (Fuzzification) ทำการแปลงค่าของอินพุตให้อยู่ในรูปค่าฟังก์ชั่นสมาชิก

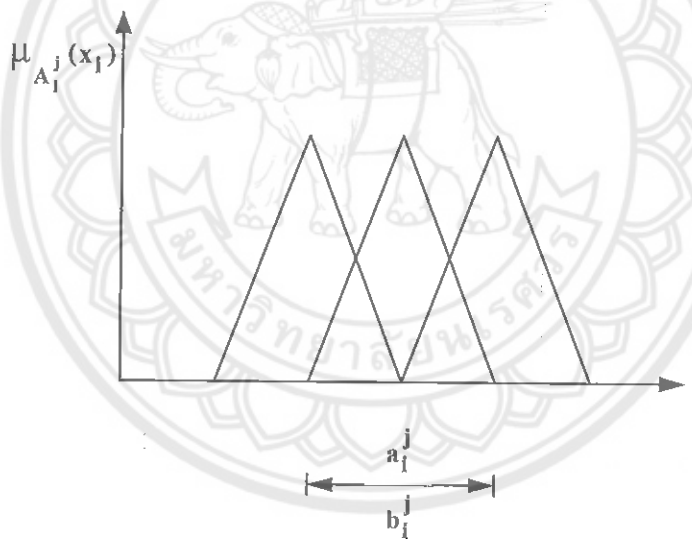
2. การอนุมาน (Inference) เป็นการสร้างกฎฟัซซี่

3. การแปลงกลับของฟัซซี่ (Defuzzification) เป็นการหาค่าเอาท์พุตของฟัซซี่จากกฎและฟัซซี่เซตของอินพุต การแปลงกลับของฟัซซี่มีอยู่ด้วยกันหลายวิธี ทั่วไปที่ใช้กันมากกว่าวิธีอื่นคือ Centroid

2.5 กฎของฟัซซี่ซิงเกิลตัน (fuzzy singleton rule)

ฟัซซี่ซิงเกิลตัน เป็นวิธีการหนึ่งที่ใช้หาเอาท์พุตเมื่อฟังก์ชั่นภาวะสมาชิกมีขอบเขตเป็นฟังก์ชั่นสมมาตร เช่น ฟังก์ชั่นของรูปสามเหลี่ยม, ฟังก์ชั่นของรูปสี่เหลี่ยมคางหมูหรือฟังก์ชั่นของรูประฆัง เป็นต้น ฟัซซี่ซิงเกลตันมีขั้นตอนต่างๆ ดังนี้

ขั้นตอนที่ 1 การทำให้เป็นฟัซซี่ เพื่อความง่ายต่อการคำนวณจะแสดงโดยใช้ฟังก์ชันของรูปสามเหลี่ยมดังรูป 2.7



รูปที่ 2.7 ฟังก์ชันภาวะสมาชิกของฟัซซี่ที่มีขอบเขตเป็นฟังก์ชันของรูปสามเหลี่ยม[2]

สามารถเขียนเป็นสมการได้เป็น

$$\mu_{A_i^j}(X_i) = 1 - \frac{2\left|x_i - a_i^j\right|}{b_i^j} \qquad i = 1,2,\ldots,n \qquad j = 1,2,\ldots,M \tag{2.6}$$

เมื่อ x_i เป็นค่าของอินพุต, a_i^j เป็นศูนย์กลางของสามเหลี่ยม, b_i^j เป็นความกว้างของฐานสามเหลี่ยมและ M เป็นจำนวนกฎฟัซซี่ กฎเหล่านี้ถูกเรียกว่า ฟัซซี่ซิงเกิลตัน

<u>ขั้นตอนที่ 2</u> การอนุมาน

กฎ j : *If* x_i is A_1^j *and* x_2 is A_2^j *and* ... *and* x_n is A_n^j *then* u_n is w_j.

เมื่อ A_i^j เป็นเทอมของตัวแปรเชิงภาษาซึ่งจะแสดงค่าด้วยฟังก์ชันภาวะสมาชิก $\mu_{A_i^j}(x_i)$, w_j เป็นเลขจำนวนที่กำหนดในเอาท์พุตแต่ละกฎ

<u>ขั้นตอนที่ 3</u> การแปลงกลับของฟัซซี่ จากกฎของฟัซซี่ซิงเกิลตันสามารถคำนวณหาเอาท์พุตได้ดังนี้

$$u_n = \frac{\sum_{j=1}^{M} \mu_j w_j}{\sum_{j=1}^{M} \mu_j} \tag{2.7}$$

เมื่อ $\mu_j = \mu_{A_1^j}(x_i)\,\mu_{A_2^j}(x_i)\ldots\mu_{A_n^j}(x_i)$

โดย M คือจำนวนกฎของฟัซซี่ และ μ_n คือค่าเอาท์พุตของฟัซซี่

2.6 พื้นฐานทั่วไปเกี่ยวกับการประมวลผลภาพ

ในโครงงานนี้หาระยะทางของหุ่นยนต์ด้วยการประมวลผลภาพในหัวข้อนี้จะกล่าวถึงทฤษฎีการประมวลผลภาพ

2.6.1 ชนิดของรูปภาพทั่วไป

รูปภาพสามารถแบ่งวิธีการจัดเก็บได้เป็น 2 ชนิดใหญ่ๆ ดังต่อไปนี้

รูปภาพเวคเตอร์ (Vector Graphic) เป็นรูปภาพที่ไม่ขึ้นกับความละเอียดของภาพ เนื่องจากภาพชนิดนี้ถูกสร้างขึ้นจากสมการเส้นต่างๆ ไม่ว่าจะเป็นเส้นตรง เส้นโค้ง เมื่อเราทำการย่อขยายภาพแบบนี้ คอมพิวเตอร์จะทำการคำนวณภาพใหม่ทำให้ภาพคมชัดเสมอ ตัวอย่างของรูปภาพแบบนี้ที่เห็นได้ชัด คือ รูปภาพที่สร้างจากโปรแกรม Adobe IIlustrator , CorelDRAW , Macromedia Freehand เป็นต้น

รูปภาพแบบบิตแมป (bitmap Image) เป็นรูปภาพที่เกิดจากจุดเล็กๆมาประกอบกันขึ้นมาจนเห็นเป็นภาพขึ้นมา คุณภาพของรูปภาพชนิดนี้จะขึ้นอยู่กับความละเอียด หากภาพมีความละเอียดมากก็ยิ่งมีความชัดเจนมากขึ้น เมื่อเราทำหารย่อขยายรูปภาพ คอมพิวเตอร์จะทำการขยายภาพขึ้นด้วยความละเอียดที่มีอยู่ทำให้ภาพที่ได้มีลักษณะที่หยาบ ตัวอย่างรูปภาพนี้ คือ รูปภาพที่สร้างจากโปรแกรม Adobe Photoshop , CorelPHOTO Paint เป็นต้น

2.6.2 ความละเอียดของภาพ

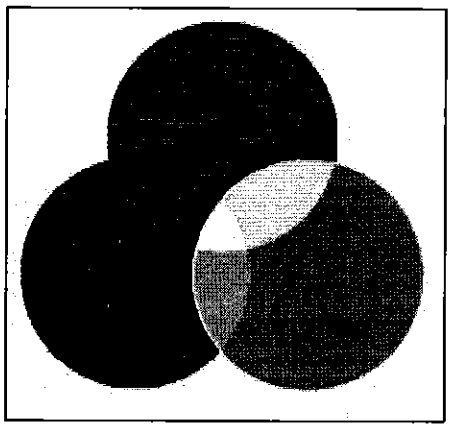
เป็นสิ่งบอกถึงคุณภาพของภาพนั้นนั้น หน่วยที่เรานิยมจะใช้บอกถึงความละเอียดของภาพ คือ พิกเซลต่อนิ้ว (Pixel / Inch) ค่านี้บอกให้เราทราบว่าภาพมีจำนวนพิกเซลกี่พิกเซลในหนึ่งนิ้ว และคุณยังสามารถคำนวณหาจำนวนจุดทั้งหมดของภาพได้อีกด้วย ตัวอย่าง เช่น ภาพขนาด 1×1 นิ้วที่มีความละเอียดเท่ากับ 8 พิกเซลต่อนิ้ว ภาพนี้จะมีพิกเซลทั้งหมดเท่ากับ 64 พิกเซล

การเลือกใช้ภาพที่มีความละเอียดสูงๆนั้นเป็นสิ่งที่ดี แต่การใช้ความละเอียดมากกว่าอุปกรณ์แสดงผล เราจะไม่สามารถใช้ประโยชน์จากความละเอียดที่เพิ่มขึ้นมานั้น อีกทั้งยังทำให้เครื่องคอมพิวเตอร์ทำงานล่าช้าลงอีกด้วย วิธีการเลือกความละเอียดที่ถูกต้องคือ เลือกความละเอียดตามอุปกรณ์แสดงผลที่คุณใช้งาน

2.6.3 พิกเซล (Pixel) และ (Dot)

พิกเซล คือ จุดเล็กที่สุดของภาพ พิกเซลหนึ่งสามารถแสดงได้หลายสี ส่วนดือทจะเป็นจุดเล็กที่ใช้ในกระบวนการพิมพ์ การสร้างพิกเซลขึ้นมาหนึ่งพิกเซลจะต้องใช้ดือทหลายดือท เพื่อทำให้เกิดภาพความเข้มและสีต่างๆกัน หน่วยดือทต่อนิ้ว (dpi) จะใช้บอกความละเอียดของเครื่องพิมพ์ ส่วนหน่วยพิกเซลต่อนิ้ว (ppi) จะใช้บอกความละเอียดของเครื่องสแกน และจอภาพ

2.6.4 โหมดสีแบบ RGB



รูปที่ 2.8 โหมดสี RGB

จากรูป 2.8 การผสมสีบนคอมพิวเตอร์นั้นอาศัย primary hues 3 สีคือ สีแดง สีเขียว และ น้ำเงิน หรือที่เรา เรียกว่า RGB colors (Red-Green-Blue) นั้นเอง ความเหมือนจริงของสีคอมพิวเตอร์นั้น ขึ้นอยู่กับในหนึ่งจุด (pixel) ของการแสดงผลนั้นใช้ระดับของสี หรือ color depths ว่ามีค่าเท่าไหร่ เช่นถ้าสี RGB มี color depths เป็น 8 planes นั่นคือ เราใช้ 8 บิตเก็บข้อมูลหนึ่งสี หมายความว่า เฉพาะ primary hues เช่น สีแดง สีเดียวก็จะมีความเป็นแดง อยู่ถึง $2^8 = 256$ ระดับ เมื่อเราผสมสี หนึ่งสี จาก แดง-เขียว-น้ำเงิน (RGB) เราต้องใช้สีแดง กี่ส่วนจาก 0 ถึง 255 ส่วน ใช้เขียว กี่ส่วนจาก 0 ถึง 255 และเช่นเดียวกัน สีน้ำเงินกี่ส่วน จาก 0 ถึง 255 สีที่เกิดขึ้นก็จะเกิดจากการผสมของสีทั้งสาม ในอัตราส่วนต่างๆ กัน ตัวอย่าง สีเหลืองธรรมดา เกิดจากการผสมสี แดง 255 ส่วน สีเขียว 255 ส่วน และสีน้ำเงิน 0 ส่วน

ซึ่งระดับของสีแดง มีถึง 256 ระดับ สีเขียว 256 ระดับ สีน้ำเงิน 256 ระดับ ดังนั้น RGB ทั้งหมดใช้ 24 บิต (8+8+8) ในการแสดงสี RGB ของหนึ่งจุด (pixel) ซึ่งสามารถแสดงสีได้มากถึง 256x256x256 = 16.7 ล้านสี ตารางข้างล่างแสดงถึง จำนวนสีที่เป็นไปได้ ซึ่งขึ้นกับจำนวนบิต ที่ใช้กำหนดระดับของสี

ตารางที่ 2.1 แสดงจำนวนสีที่เป็นไปได้ซึ่งขึ้นกับจำนวนบิต

จำนวนบิตที่ใช้เก็บสีต่อหนึ่งจุด	Color Mode Name	จำนวนสีที่แสดงได้
1	Black and White	2
4	16-Color (EGA)	16
8	Pseudo Color	256
16	HI-Color	65,536
24	True Color	16,777,216

2.7 การสื่อสารข้อมูลแบบไร้สาย (Wireless)

ระบบการสื่อสารแบบไร้สายมีเทคโนโลยีที่นิยมกันอยู่ 2 ชนิด คือการใช้คลื่นวิทยุ (RF) ในการส่ง และใช้รังสีอินฟาเรด (IR) ที่คล้ายกับรีโหมดของเครื่องรับโทรทัศน์ในการส่ง

ระบบพื้นฐานโดยทั่วไปที่ใช้กันอยู่ในทุกวันนี้จะอธิบายได้ว่าระบบสื่อสารข้อมูลดิจิตอลแบบไร้สายที่ใช้คลื่นวิทยุ (RF) ในการส่งนั้นแบ่งออกเป็น 3 ประเภทใหญ่ๆขึ้นกับกำลังการส่งของสัญญาณที่ใช้ Federal Communications Commission หรือ (FCC) เป็นผู้แยกระดับพลังงานของสัญญาณสำหรับเครื่องรับและเครื่องส่ง

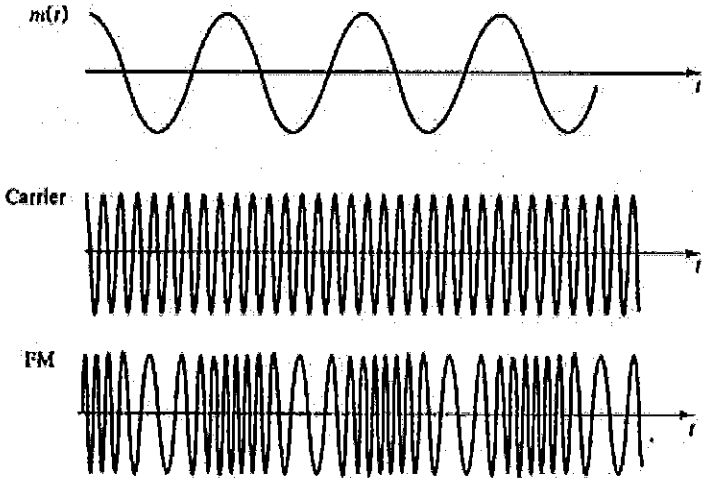
การบอกว่าสัญญาณที่ใช้ขึ้นอยู่กับ FCC หรือนั้นจำเป็นต้องมีใบอนุญาต เพียงแต่ต้องทำให้ระบบของเรามีคุณภาพใกล้เคียงกับมาตรฐานที่กำหนด

ในโครงงานนี้ อาศัยการควบคุมผ่านทางไมโครคอมพิวเตอร์ผ่านทางระบบไร้สายผ่านทางไมโครโปรเซสเซอร์ การอินเตอร์เฟสกับคอมพิวเตอร์หรือไมโครโปรเซสเซอร์ คือ การทำงานติดต่อร่วมกันระหว่างซีพียูกับอุปกรณ์อื่นๆกับการโอนถ่ายข้อมูลระหว่างอุปกรณ์ต่างๆ นอกเหนือจากจะต้องทำงานติดต่อกับ RAM, ROM แล้วยังต้องมีการติดต่อกับอุปกรณ์ภายนอกที่มีการส่งข้อมูลอินพุท, เอาท์พุทอีกทางหนึ่ง ซึ่งเป็นการเพิ่มประสิทธิภาพให้ระบบสมบูรณ์ ในระบบต่างของอุปกรณ์อิเล็กทรอนิกส์ จะทำงานต่อเนื่องเป็นลูกโซ่ ดังเช่น การส่งรับข้อมูลจากซีพียูไปยังส่วนอื่นๆ เป็นต้น

การแปลงสัญญาณข้อมูล (Modulation) ในการส่งสัญญาณข้อมูลผ่านช่องทางการสื่อสารนั้น พาหะ หรือตัวนำสัญญาณในการเคลื่อนย้ายข้อมูลต้องอาศัยพลังงานไฟฟ้า ซึ่งขั้นตอนในการเพิ่มพลังงานไฟฟ้า เรียกว่า Modulation

การแปลงสัญญาณข้อมูลสามารถทำได้หลายวิธี

ในโครงงานนี้เลือกใช้วิธีการแปลงสัญญาณแบบ Frequency Modulation (FM) ซึ่งจะมีลักษณะของ Amplitude คงที่ แต่ความถี่ของสัญญาณจะไม่คงที่เปลี่ยนไปตามขนาด (Amplitude) ของสัญญาณข้อมูล คุณภาพของ สัญญาณ FM นั้นจะดีกว่าแบบ AM แต่ระบบการทำงานจะซับซ้อนมากกว่า ปัจจุบันมีการเพิ่มวงจร Phase-Lock-Loop เข้าไปในระบบเพื่อปรับความถี่ และเฟสของสัญญาณ ทำให้สัญญาณมีคุณภาพ และถูกต้องยิ่งขึ้น โดยลักษณะสัญญาณก่อนที่จะทำการ modulation สัญญาณพาห์และสัญญาณที่ได้ทำการ modulation แบบ FM แล้ว แสดงได้ดังรูปที่ 2.9



รูปที่ 2.9 แสดงลักษณะของสัญญาณ Frequency Modulation

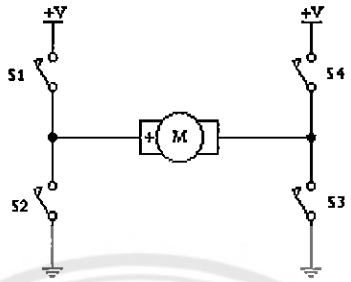
2.8 การขับเคลื่อนมอเตอร์แบบกระแสตรง

มอเตอร์แบบกระแสตรง (DC motor) ในปัจจุบันสามารถทำได้หลายประเภท ขึ้นอยู่กับการสร้างและวิธีการควบคุม สำหรับการหมุนของมอเตอร์กระแสตรงจะใช้ไฟกระแสตรง (Direct Current) จ่ายให้กับลวดสนามไฟฟ้า หรือขดลวดผลิตแรงเคลื่อนไฟฟ้าภายใน (Motor Armature) สำหรับแรงบิดของมอเตอร์จะขึ้นอยู่กับเส้นแรงแม่เหล็ก ถ้าหากเส้นแรงแม่เหล็กมีจำนวนมาก ก็จะทำให้มีแรงบิดสูง มอเตอร์กระแสตรงส่วนใหญ่ถ้าหมุนด้วยความเร็วสูงจะมีแรงบิดต่ำ และถ้าหมุนด้วยความเร็วต่ำจะมีแรงบิดสูง การควบคุมความเร็วในการหมุนนั้นจะสามารถทำได้โดยการควบคุมแรงดันไฟฟ้าให้กับมอเตอร์ แต่การทำงานแบบนี้จะต้องจ่ายแรงดันไฟฟ้าให้กับมอเตอร์ตลอดเวลา ซึ่งทำให้เกิดการสูญเสียกำลังไฟฟ้าและความร้อนที่เกิดขึ้นตลอดเวลา ดังนั้นจึงมีการใช้วิธีการจ่ายแรงดันไฟฟ้าเป็นช่วงๆ ให้กับมอเตอร์แทน การควบคุมความเร็วด้วยวิธีการนี้มีการใช้งานอยู่สองลักษณะคือ แบบ Pulse Frequency Modulation (PFM) คือ การควบคุมการจ่ายแรงดันและหยุดจ่ายแรงดันเพื่อเปิด-ปิดมอเตอร์เป็นช่วงๆ และค่าความถี่ของสัญญาณควบคุมจะไม่แน่นอน แบบที่สองแบบ Pulse Width Modulation (PWM) ควบคุมมอเตอร์แบบเดียวกันกับ PFM แต่ความถี่ในการควบคุมจะคงที่ ในโครงงานนี้จะควบคุมแบบ PWM โดยใช้วงจร H-Bridge ในการสร้างแรงดัน PWM ให้กับมอเตอร์

2.9.1 หลักการทำงานของวงจร H-Bridge

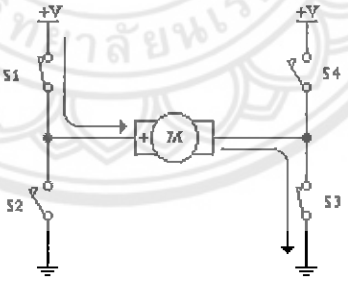
วงจรขยายที่ใช้สำหรับการขับมอเตอร์ที่นิยมใช้กันอย่างแพร่หลาย คือวงจรขยายสำหรับขับมอเตอร์ให้หมุนไปได้สองทิศทางโดยใช้หลักการทำงานของวงจร H-Bridge Switching

เหตุที่เรียกว่าวงจร H-Bridge Switch ก็เพราะว่าวงจรต่อเป็นรูปตัว H นั่นเอง หลักการของวงจรนั้นประกอบด้วย Switch 4 ตัวนั่นคือ S1, S2, S3 และ S4 ดังรูป 2.10



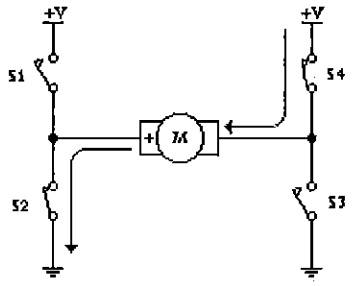
รูปที่ 2.10 แสดงวงจรขยายแบบ H-Bridge

ในสภาวะเริ่มต้นสวิตช์ทุกตัวอยู่ในภาวะ "เปิด" อยู่ จะไม่มีกระแสไฟฟ้าไหลเข้าสู่มอเตอร์ ดังรูปที่ 2.10 เมื่อสวิตช์ S1 และ S3 อยู่ในสภาวะ "ปิด" พร้อมกัน ตามรูปที่ 2.11 ทำให้มีกระแสไฟฟ้าไหลผ่านมอเตอร์ จากขั้วด้านหนึ่งของมอเตอร์ (สมมติให้เป็นขั้วบวก) ไปยังอีกขั้วของมอเตอร์ ทำให้มอเตอร์ทำงานในทิศทางไปข้างหน้า จะหมุนแบบตามเข็มนาฬิกา (CW) หรือทวนเข็มนาฬิกา (CCW) นั้น ขึ้นอยู่กับลักษณะของการพันขดลวดภายในมอเตอร์



รูปที่ 2.11 แสดงการไหลของกระแสไฟฟ้าผ่านทางขั้วบวกของมอเตอร์ทำให้มอเตอร์หมุนไปในทิศทางไปข้างหน้า

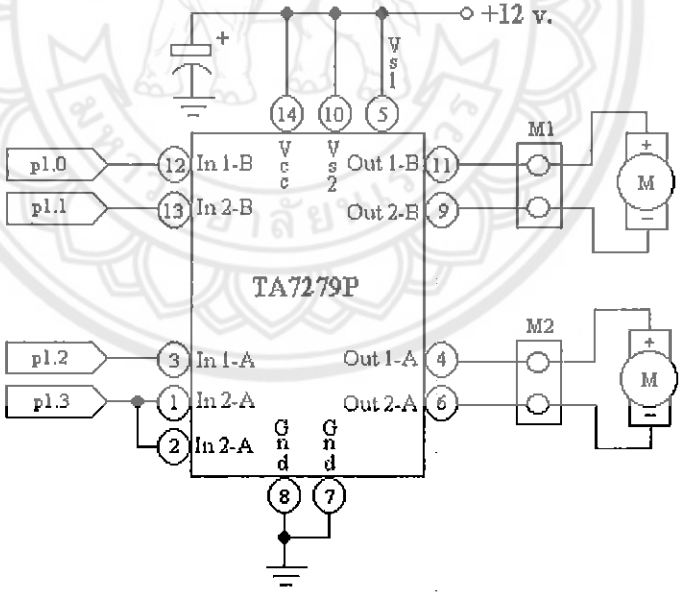
ในทางกลับกัน หากสวิตช์ S2 และ S4 "ปิด" พร้อมกัน ดังรูปที่ 2.12 จะทำให้เกิดกระแสไฟฟ้าไหลผ่านมอเตอร์ จากขั้วลบของมอเตอร์ไปยังขั้วบวก ทำให้มอเตอร์ทำงานในทิศทางตรงกันข้าม



รูปที่ 2.12 แสดงการไหลของกระแสไฟฟ้าผ่านทางขั้วลบของมอเตอร์ ทำให้มอเตอร์หมุนไปในทิศทางย้อนกลับ

2.9.2 การสร้างวงจร H-Bridge Switching จากไอซี เบอร์ TA7279P

เนื่องจากว่า IC TA7279P เป็นไอซีที่มีวงจร H-Bridge จำนวน 2 ชุดที่สามารถควบคุมมอเตอร์ได้ถึง 2 ตัว และยังมีฟังก์ชั่นหยุดการทำงานเมื่ออุณหภูมิสูงเกินไป อีกทั้งยังมีวงจรจำกัดกระแสอีกด้วยจึงทำให้ไอซีตัวนี้สามารถใช้งานได้ง่ายเหมาะกับการประกอบตัวหุ่นยนต์ที่ไม่ต้องการความยุ่งยากมากนัก ซึ่งการใช้งานเพียงแค่ต่อตัวเก็บประจุ 10 μF เพิ่มอีกหนึ่งตัว เพื่อใช้เป็นตัวบายพาส กันสัญญาณรบกวน ดังรูป 2.13



รูปที่ 2.13 รูปแสดงไอซีเบอร์ TA7279P

การควบคุมการหมุนของมอเตอร์เมื่อทำการต่อวงจรตามขาต่างๆเรียบร้อยแล้ว ก็สามารถป้อนอินพุตที่เป็นสัญญาณลอจิก "0" และ "1" เข้าที่ขาอินพุต 1 และ 2 เอาท์พุตที่ออกมาจะเป็นการสั่งให้มอเตอร์ทำงานอยู่ในโหมดใด ดังตาราง

ตารางที่ 2.2 การทำงานของมอเตอร์เมื่อป้อนอินพุตในลักษณะต่างๆ

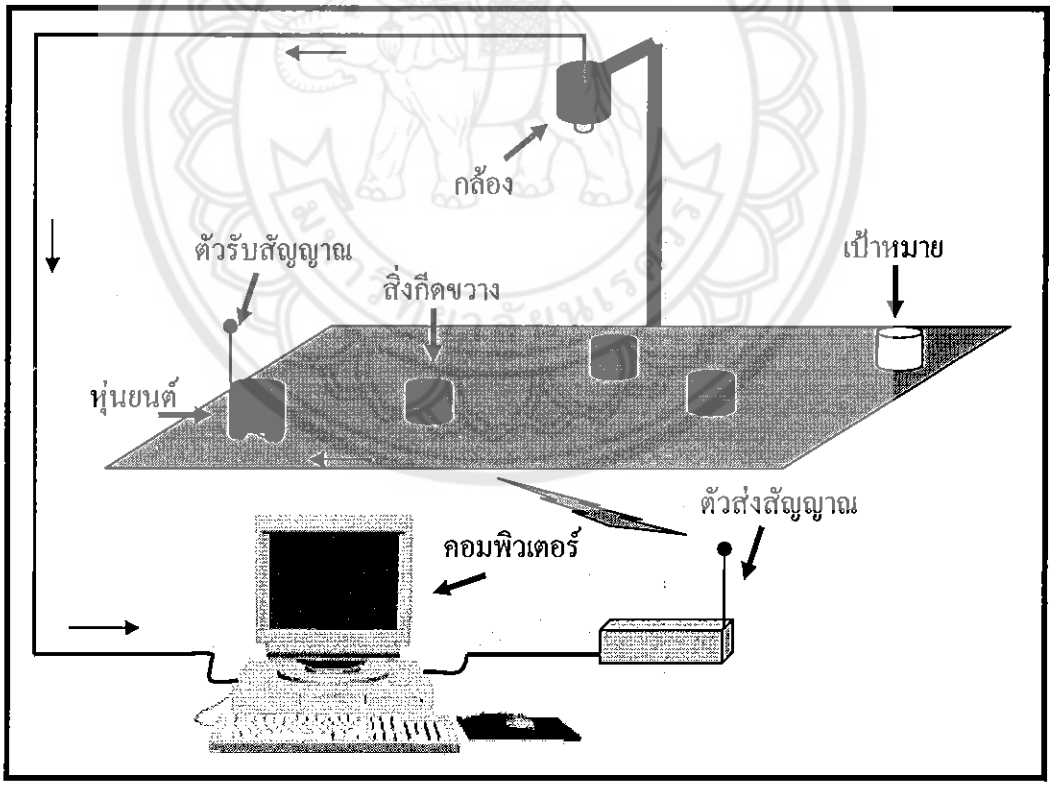
อินพุต		เอาท์พุต		โหมด
1	2	1	2	
0	0	High – Impedance		STOP
0	1	L	H	CW/CCW
1	0	H	L	CCW/CW
1	1	H	H	BREAK

บทที่ 3
วิธีการดำเนินการ

ปัจจุบันมนุษย์สามารถใช้อุปกรณ์และเครื่องมือที่ทันสมัย เพื่ออำนวยความสะดวกในด้านต่าง ๆ ซึ่งกล้อง Web Cam และอุปกรณ์ส่งสัญญาณแบบไร้สาย ก็เป็นเทคโนโลยีที่น่าสนใจ สามารถประยุกต์ใช้งานได้หลากหลายรูปแบบ และสามารถทำงานร่วมกับคอมพิวเตอร์ในการโปรแกรมการทำงานได้อย่างมีประสิทธิภาพ

3.1 การควบคุมการเคลื่อนที่ของหุ่นยนต์จากคอมพิวเตอร์โดยการประมวลผลจากภาพ

การเคลื่อนที่ของหุ่นยนต์จะถูกควบคุมโดยคอมพิวเตอร์ ซึ่งคอมพิวเตอร์รับภาพจากกล้อง Web Cam จากนั้นแปลงสัญญาณภาพเพื่อหาตำแหน่งของ หุ่นยนต์ และสิ่งกีดขวาง และใช้ทฤษฎีฟัซซี่ซิงเกิลตันตัดสินใจในการหลบหลีกสิ่งกีดขวาง เพื่อไปยังเป้าหมาย หลังจากตัดสินใจจะทำการส่งสัญญาณควบคุมผ่านทางพอร์ต RS-232 ไปยังอุปกรณ์ส่งสัญญาณแบบไร้สาย โดยจะมีตัวรับสัญญาณติดอยู่ที่ตัวหุ่นยนต์เพื่อไปควบคุมหุ่นยนต์ ดังในรูป 3.1



รูปที่ 3.1 แสดงการทำงานการควบคุมหุ่นยนต์เคลื่อนที่

3.2 การเขียนโปรแกรมของระบบควบคุมการตัดสินใจในการหลบหลีกสิ่งกีดขวางของหุ่นยนต์โดยใช้ทฤษฎีฟัซซี่

3.2.1 การออกแบบตัวแปรอินพุต

ในการออกแบบระบบการควบคุมการตัดสินใจของหุ่นยนต์นั้น ในขั้นแรกจะต้องนำอินพุตเข้าสู่ระบบ ซึ่งอินพุตที่ได้จะถูกส่งมาจากเอาท์พุตของการประมวลผลภาพ โดยอินพุตที่ได้รับมาจากการประมวลผลภาพจะสามารถบอกค่าของระยะห่างระหว่างหุ่นยนต์กับสิ่งกีดขวางเพื่อที่จะนำค่านั้นมาแสดงในอยู่ในรูปของระบบฟัซซี่ ในการตัดสินใจของหุ่นยนต์นั้น หุ่นยนต์จะสามารถหลบหลีกสิ่งกีดขวางได้ก็ต่อเมื่อสิ่งกีดขวางอยู่ในตำแหน่งที่อยู่ภายในรัศมี 180 องศาของตัวหุ่นยนต์เท่านั้น ดังรูป 3.2

รูปที่ 3.2 ช่วงขององศาที่มีผลต่อการตัดสินใจหลบหลีกสิ่งกีดขวางของหุ่นยนต์

จากรูป 3.2 แสดงให้เห็นช่วงของ 180 องศาที่มีผลต่อการตัดสินใจที่จะหลบหลีกสิ่งกีดขวางของหุ่นยนต์ โดยภายใน 180 องศานี้จะต้องแบ่งออกเป็นช่วงๆ ช่วงละ 60 องศาเพื่อที่สามารถบอกได้ว่าตำแหน่งของสิ่งกีดขวางนั้นอยู่ทางด้านขวามือ ซ้ายมือ หรือด้านหน้าของหุ่นยนต์ ดังรูป 3.3 ในกรณีที่หุ่นยนต์จะใช้ในการคำนวณองศาและทิศทางในการหลบหลีกสิ่งกีดขวางได้ตามต้องการ

สิ่งกีดขวางที่มีมุมอยู่ระหว่าง -90 ถึง -30 องศา จัดให้เป็นด้านซ้ายของหุ่นยนต์

สิ่งกีดขวางที่มีมุมอยู่ระหว่าง 30 ถึง 0 องศา หรือ มุมที่อยู่ระหว่าง 0 ถึง -30 องศา จัดให้เป็นด้านหน้าของหุ่นยนต์

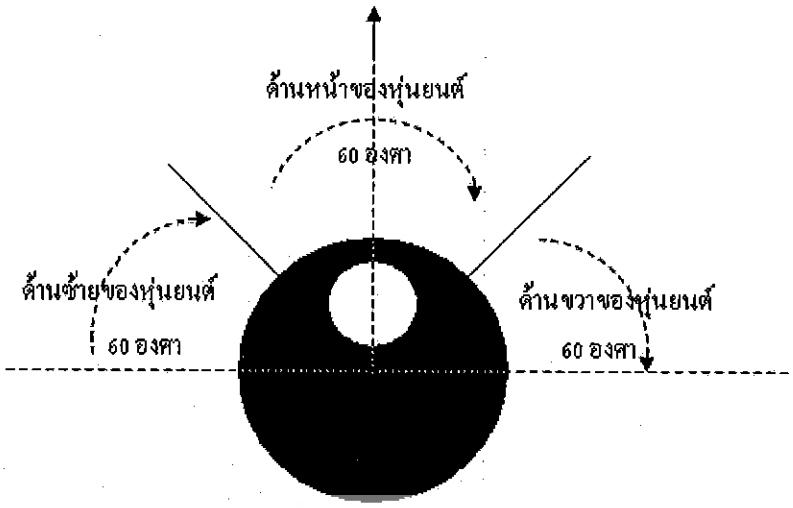
สิ่งกีดขวางที่มีมุมอยู่ระหว่าง 30 องศา ถึง 90 องศา จัดให้เป็นด้านขวาของหุ่นยนต์

ดังนั้นในขั้นตอนนี้เราจึงแบ่งตัวแปรอินพุตออกเป็น 3 ตัว คือ

อินพุตที่1 : รับระยะห่างที่อยู่ทางซ้ายที่ใกล้ที่สุดของหุ่นยนต์

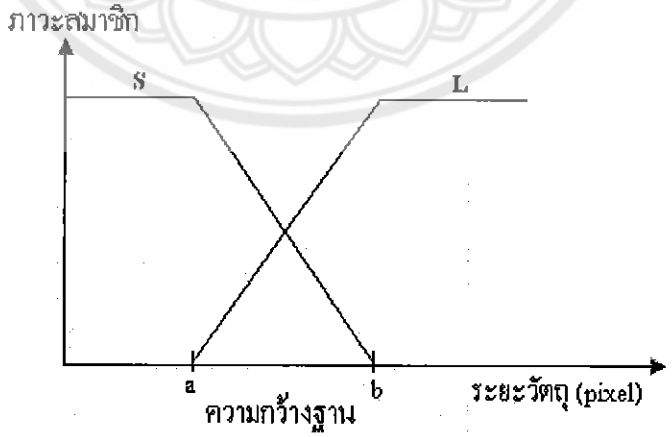
อินพุตที่2 : รับระยะห่างที่อยู่ด้านหน้าที่ใกล้ที่สุดของหุ่นยนต์

อินพุตที่3 : รับระยะห่างที่อยู่ทางขวาที่ใกล้ที่สุดของหุ่นยนต์



รูปที่ 3.3 การแบ่งขอบเขตของอินพุต

เมื่อทำการออกแบบตัวแปรอินพุตแล้ว การดำเนินงานขั้นตอนต่อไปก็ต้องทำตัวแปรอินพุตให้อยู่ในรูปแบบของฟัซซี่ โดยการอาศัยฟังก์ชั่นของสี่เหลี่ยมคางหมูมาช่วยในการอธิบาย ดังรูปที่ 3.4 สำหรับอินพุต 1 ตัวจะสามารถเขียนฟังก์ชั่นของภาวะสมาชิกได้ 2 ฟังก์ชั่น คือ ระยะวัตถุที่อยู่ใกล้ที่สุดน้อยมากและระยะวัตถุที่อยู่ไกลมาก ซึ่งช่วงของฟังก์ชั่นภาวะสมาชิกจะมีฐานกว้างเท่าไรขึ้นอยู่กับผู้ออกแบบการทดลอง โดยในที่นี้ผู้ออกแบบกำหนดให้สามารถเปลี่ยนความกว้างฐานได้ตามความเหมาะสมซึ่งจะใช้การวัดระยะห่างระหว่างหุ่นยนต์กับสิ่งกีดขวางที่มีหน่วยเป็น pixel เนื่องจากวัตถุแต่ละชิ้นอาจมีขนาดของวัตถุต่างกันตามแต่สถานการณ์ซึ่งสามารถวิเคราะห์ค่าความกว้างฐานได้จากภาพวิดีโอที่ใช้จับภาพวัตถุในการทดลอง



รูปที่ 3.4 กราฟของตัวแปรอินพุต

3.2.2 การออกแบบกฎฟัซซี่

การออกแบบกฎของฟัซซี่จากการสร้างอินพุตที่ผ่านมา ทำได้โดยการอนุมานจากตัวแปรอินพุตทั้ง 3 ตัวคือ ระยะห่างที่อยู่ทางซ้ายที่ใกล้ที่สุดของหุ่นยนต์, ระยะห่างที่อยู่ด้านหน้าที่ใกล้ที่สุดของหุ่นยนต์ และระยะห่างที่อยู่ทางขวาที่ใกล้ที่สุดของหุ่นยนต์ ประกอบกับแต่ละอินพุตจะมีฟังก์ชั่นภาวะสมาชิกทั้งหมดตัวละ 2 ฟังก์ชั่น ดังนั้นการออกแบบกฎของฟัซซี่ก็จะมีเท่ากับ 2^3 กฎ หรือ 8 กฎ ดังแสดงในตารางที่ 3.1

ตารางที่3.1 ตารางแสดงกฎของฟัซซี่

ระยะวัตถุ ทางซ้าย	ระยะวัตถุทางด้านหน้า	ระยะวัตถุทางด้านขวา	มุม (องศา)
SL	SF	SR	180
SL	SF	LR	0
SL	LF	SR	60
SL	LF	LR	30
LL	SF	SR	-60
LL	SF	LR	-30
LL	LF	SR	30
LL	LF	LR	0

SL : เป็นระยะวัตถุทางซ้ายน้อยมาก
SR : เป็นระยะวัตถุทางขวาน้อยมาก
LL : เป็นระยะวัตถุทางซ้ายมากมาก
LR : เป็นระยะวัตถุทางขวามากมาก
SF : เป็นระยะวัตถุด้านหน้าน้อยมาก
LF : เป็นระยะวัตถุด้านหน้ามากมาก

จากกฎของฟัซซี่ซิงเกลตันตามหัวข้อที่ 2.5 สามารถเขียนโดยอาศัยการอนุมานได้ตามรูปแบบของ *If* x_1 is A_1^j *and* x_2 is A_2^j *and* ... *and* x_n is A_n^j *then* u_n is w_j. ได้ดังต่อไปนี้

กฎ 1: If left is SL (ระยะวัตถุทางซ้ายน้อยมาก) and Font is SF (ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) and Right is SR (ระยะวัตถุอยู่ทางขวาน้อยมาก) Then Degree is 180

กฎ 2: If left is SL (ระยะวัตถุทางซ้ายน้อยมาก) and Font is SF (ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) Then Degree is 0

กฎ 3: If left is SL (ระยะวัตถุทางซ้ายน้อยมาก) and Font is LF (ระยะวัตถุอยู่ทางด้านหน้ามากมาก) and Right is SR (ระยะวัตถุอยู่ทางขวาน้อยมาก) Then Degree is 60

กฎ 4: If left is SL (ระยะวัตถุทางซ้ายน้อยมาก) and Font is LF (ระยะวัตถุอยู่ทางด้านหน้ามากมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) Then Degree is 30

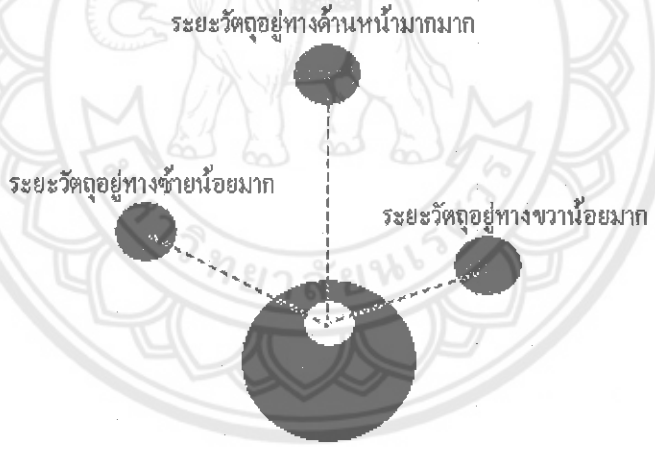
กฎ 5: If left is LL (ระยะวัตถุทางซ้ายมากมาก) and Font is SF (ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) and Right is SR (ระยะวัตถุอยู่ทางขวาน้อยมาก) Then Degree is -60

กฎ 5: If left is LL (ระยะวัตถุทางซ้ายมากมาก) and Font is SF (ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) Then Degree is -30

กฎ 6: If left is LL (ระยะวัตถุทางซ้ายมากมาก) and Font is LF (ระยะวัตถุอยู่ทางด้านหน้ามากมาก) and Right is SR (ระยะวัตถุอยู่ทางขวาน้อยมาก) Then Degree is 30

กฎ 6: If left is LL (ระยะวัตถุทางซ้ายมากมาก) and Font is LF (ระยะวัตถุอยู่ทางด้านหน้ามากมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) Then Degree is 0

เมื่อกำหนดให้ w_j เป็นเอาท์พุตของแต่ละกฎนั้น w_j จะได้มาจากการคำนวณซึ่งผลลัพธ์ที่ได้จากการคำนวณจะเป็นตัวกำหนดว่าหุ่นยนต์ควรจะหลบหลีกสิ่งกีดขวางเป็นมุมกี่องศา โดยจะคำนึงถึงกฎของฟัซซี่ซิงเกลตันทั้ง 8 กฎ ตัวอย่างเช่น ระยะวัตถุทางซ้ายน้อยมาก และระยะวัตถุอยู่ทางด้านหน้ามากมาก และ ระยะวัตถุอยู่ทางขวาน้อยมาก ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงตำแหน่งของสิ่งกีดขวาง

จากรูปที่ 3.5 เมื่อมีสิ่งกีดขวางอยู่ทางด้านขวามือ และซ้ายมือในระยะที่ใกล้น้อยที่สุดเหมือนกัน หุ่นยนต์จึงต้องตัดสินใจที่จะเดินตรงไป แต่จะเดินตรงไปที่องศาเท่าไรนั้นจะต้องเขียนโปรแกรมในส่วนกฎของฟัซซี่ซิงเกลตันเพื่อนำไปใช้ในการคำนวณสำหรับการหาเอาท์พุตของ

ปร. วิทย์ ๒๕๔๘ ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร 15010202

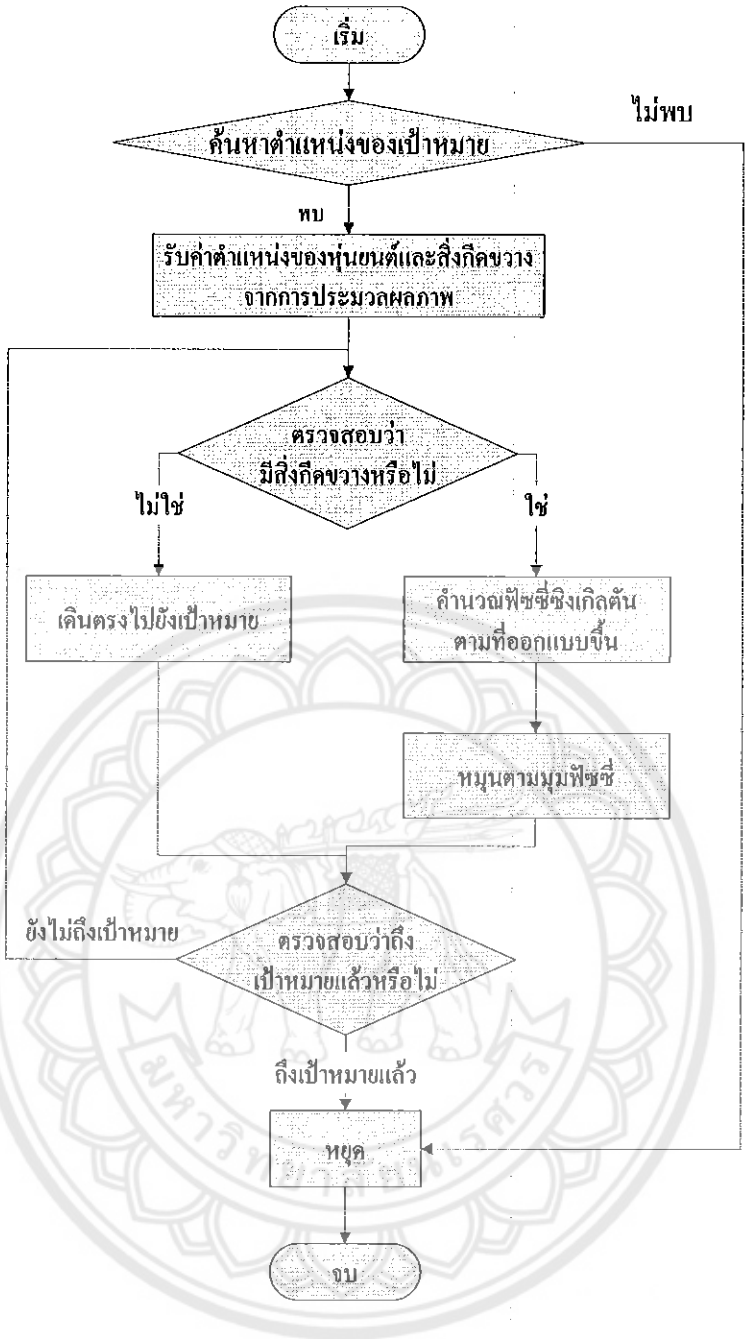
ระบบต่อไป จึงเป็นการคำนวณเพื่อหาค่าของ ของแต่ละกฎเท่านั้น เพื่อให้ทราบว่ากฎทุกกฎทำงานแล้ว โดยค่า $\mu_{,w,}$ ของแต่ละกฎจะถูกเก็บไว้ในตัวแปร Array ที่สามารถเก็บค่าได้ 8 ค่า [2]

3.2.3 เอาท์พุตของระบบ

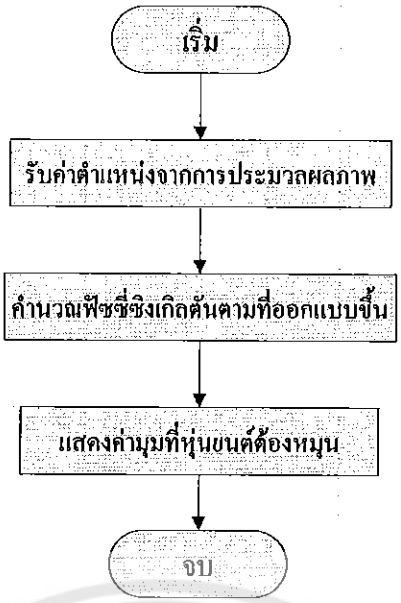
กฎทุกกฎจะถูกเรียกใช้งานและได้เอาท์พุตของแต่ละกฎ เอาท์พุตสุดท้ายซึ่งเป็นเอาท์พุตของระบบจะคำนวณหาได้จากทฤษฎีฟัซซี่ซิงเกลตัน ตามสมการที่ 2.7 จะต้องมีการเขียนโปรแกรมเพื่อให้คำนวณหาค่าของแต่ละกฎ และเก็บค่า $\mu_{,}$ ของแต่ละกฎไว้ในตัวแปร ValueRule2 เป็นตัวแปร Array ที่สามารถเก็บค่าได้ 8 ค่า

3.3 การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกลตัน

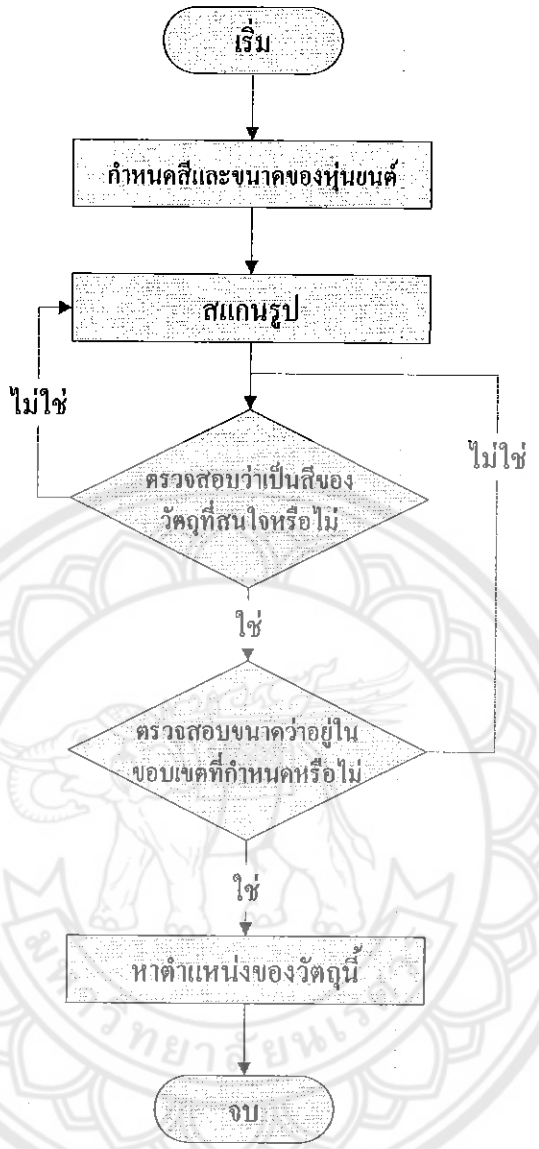
ตามโครงสร้างของโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์นั้นสามารถแบ่งออกได้เป็น 3 ส่วนคือ ส่วนของการควบคุมการเคลื่อนที่ของหุ่นยนต์ ส่วนของการควบคุมด้วยระบบฟัซซี่ลอจิกที่เป็นส่วนสำคัญที่ทำให้หุ่นยนต์สามารถตัดสินใจในการหลบหลีกสิ่งกีดขวางได้ตามต้องการ ในที่นี้ผู้ออกแบบโครงงานกำหนดให้มีสิ่งกีดขวางได้ตั้งแต่ 0 จนถึง 10 ตัว และส่วนสุดท้ายคือส่วนของการประมวลผลภาพเพื่อหาตำแหน่งของวัตถุทุกชิ้นโดยสามารถสรุปเป็นแผนภาพได้ดังนี้



รูปที่ 3.6 แผนภาพแสดงการทำงานของโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์



รูปที่ 3.7 แผนภาพการทำงานของโปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์โดยฟัซซี่ซิงเกลตัน



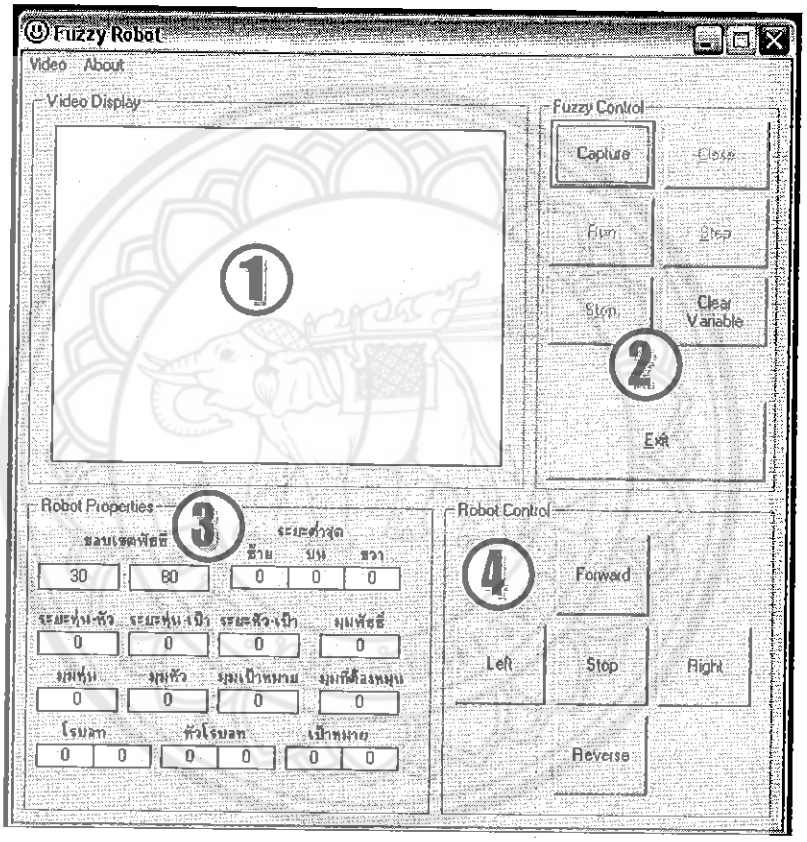
รูปที่ 3.8 แผนภาพแสดงการประมวลผลภาพหาตำแหน่งวัตถุ

จากแผนภาพข้างต้นแต่ละส่วนของโปรแกรมจะเป็นอิสระต่อกัน แต่ในการใช้งานจะต้องนำโปรแกรมทั้ง 3 ส่วนมารวมเข้ากัน เป็นโปรแกรมที่สามารถควบคุมหุ่นยนต์ให้สามารถตัดสินใจเคลื่อนที่หลบหลีกสิ่งกีดขวางได้

3.4 การออกแบบโปรแกรมควบคุมหุ่นยนต์เคลื่อนที่แบบวิทัศน์โดยหลักการฟัซซี่ซิงเกลตัน

โปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์ฟัซซี่นั้น จะเขียนด้วยโปรแกรมวิชวลเบสิกเวอร์ชั่น 6.0 เนื่องจากโปรแกรมวิชวลเบสิกเป็นโปรแกรมที่สามารถทำความเข้าใจได้ง่าย และมีออบเจ็คที่ช่วยในการสร้างหน้าตาของโปรแกรมเพื่อให้ผู้ใช้สามารถใช้งานได้ง่ายขึ้น

สำหรับขั้นตอนในการเขียนโปรแกรมหลังจากได้ทำความเข้าใจเกี่ยวกับความต้องการของระบบแล้วขั้นตอนต่อมาก็คือการออกแบบโปรแกรมโดยโปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์โดยหลักการฟัซซี่ซิงเกลตันสามารถแสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 แสดงหน้าตาของโปรแกรมการควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกลตัน

จากรูปที่ 3.9 จะเห็นได้ว่าโปรแกรมการควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกลตันนั้นมีส่วนประกอบหลักหลายส่วนโดยจะอธิบายเป็นส่วนๆ ดังนี้

ส่วนที่ 1 Video Display ฟอร์มจะมีหน้าตาเป็นสี่เหลี่ยมผืนผ้า ในส่วนนี้จะเป็นส่วนที่ใช้รับภาพจากกล้อง Web Cam ที่แสดงภาพวัตถุทั้งหมดในสนามทดสอบ เพื่อใช้ในการประมวลผลภาพ

ส่วนที่ 2 Fuzzy Control ในส่วนนี้จะใช้ในการ Capture ภาพเอาไว้ใช้ในการประมวลผลภาพ ปุ่ม Step ให้หุ่นยนต์เคลื่อนที่ไปทีละ1 จังหวะ, ปุ่ม Run กำหนดให้มีการเคลื่อนที่เป็นจังหวะหลายๆครั้งต่อเนื่องกัน, ปุ่ม Stop กำหนดให้หุ่นยนต์หยุดการเคลื่อนที่ และปุ่ม Exit ให้ออกจากโปรแกรม

ส่วนที่ 3 Robot Properties หลังจากที่มีการประมวลผลภาพแล้วจะแสดงรายละเอียดต่างๆ ของหุ่นยนต์ดังนี้ ขอบเขตฟัซซี่, ระยะของสิ่งกีดขวางที่อยู่ใกล้ที่สุด, ระยะจากหุ่นถึงหัวหุ่น, ระยะจากหุ่นถึงเป้าหมาย, ระยะหัวหุ่นถึงเป้าหมาย, มุมฟัซซี่, มุมหุ่นยนต์, มุมหัวของหุ่นยนต์, มุมเป้าหมาย, มุมที่หุ่นยนต์ต้องหมุน, ตำแหน่งของหุ่นยนต์, ตำแหน่งของเป้าหมาย และตำแหน่งหัวของหุ่นยนต์ ในส่วนนี้จะเป็นส่วนที่มีความสำคัญในการควบคุมการเคลื่อนที่ของหุ่นยนต์

ส่วนที่ 4 **Robot Control** เป็นส่วนที่ใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยไม่เกี่ยวข้องกับระบบฟัซซี่ลอจิก

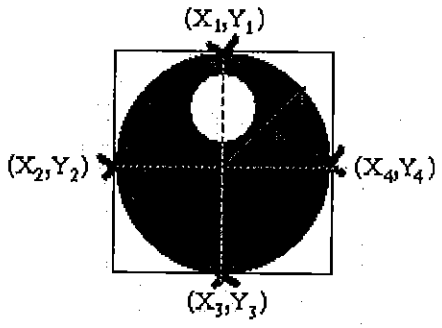
3.5 การเขียนโปรแกรมประมวลผลภาพ

ขั้นตอนในการประมวลผลภาพเริ่มจากการรับภาพมาจากกล้อง Web Cam แล้วนำภาพที่ได้มาทำการประมวลผลภาพโดยการแปลงให้อยู่ในรูปของเมตริกซ์ หลังจากนั้นใช้หลักการ RGB เข้ามาใช้ในการรับค่าของสีที่ต้องการจากวัตถุแต่ละตัว ซึ่งการกำหนดลักษณะสีให้กับวัตถุแต่ละชิ้นขึ้นอยู่กับผู้ออกแบบในที่นี้ผู้ออกแบบได้กำหนดดังนี้

- หุ่นยนต์ ผู้ออกแบบกำหนดให้หุ่นยนต์มีสีน้ำเงิน และมีวงกลมสีขาวเป็นตัวแสดงทิศทางของหุ่นยนต์ว่าหันหน้าไปทางด้านใด
- เป้าหมาย ผู้ออกแบบกำหนดให้ใช้สีขาว
- สิ่งกีดขวาง ผู้ออกแบบกำหนดให้ใช้สีแดง

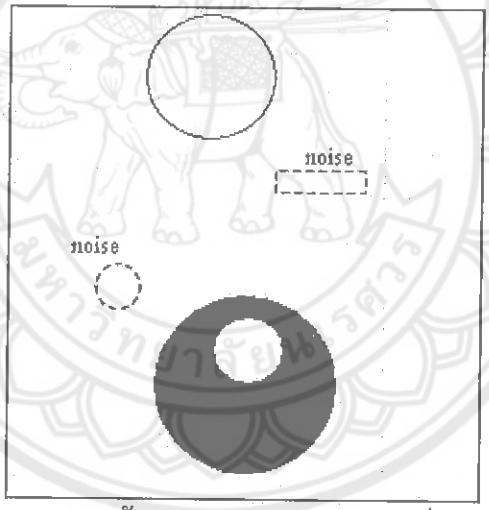
ในส่วนของโปรแกรมการประมวลผลภาพจะแยกเป็นการจับภาพวัตถุทีละตัว เริ่มจากการหาตำแหน่งของตัวหุ่นยนต์ก่อน จากนั้นก็หาตำแหน่งของเป้าหมาย และสุดท้ายจึงจะหาตำแหน่งของสิ่งกีดขวาง วิธีการหาตำแหน่งของวัตถุมีวิธีการที่คล้ายกัน คือ

ขั้นตอนที่ 1 ค้นหาภาพที่มีสีของวัตถุตัวที่เราต้องการหา เมื่อพบแล้วให้หาตำแหน่งขอบด้านบน ขอบด้านล่าง ขอบซ้าย และขอบขวาของวัตถุเพื่อทำการหารัศมีของวัตถุ ดังรูป 3.10



รูปที่ 3.10 แสดงการหาตำแหน่งและรัศมีของวัตถุ

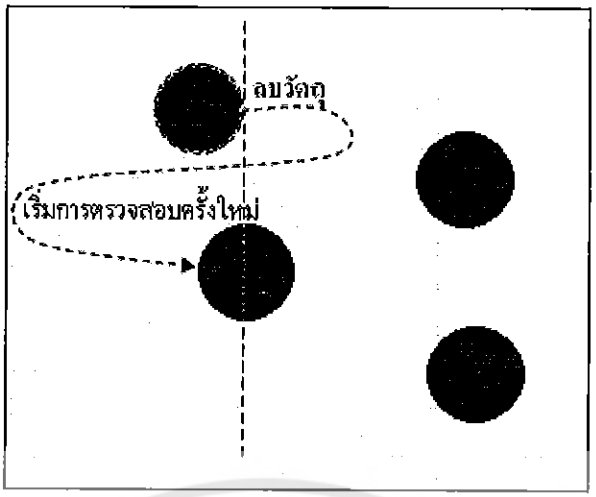
ขั้นตอนที่ 2 ตรวจสอบรัศมีของวัตถุว่าอยู่ในช่วงที่เราต้องการหรือไม่ ช่วงที่เราไม่ต้องการให้ถือว่าวัตถุนั้น เป็น Noise ไม่มีความสำคัญในการประมวลผลภาพ จากนั้นจึงจะทำการตรวจหาตำแหน่งวัตถุชิ้นต่อไป ในกรณีของตัวหุ่นยนต์ที่มีวัตถุสีขาวอยู่ด้วยหลังจากทำการหาตำแหน่งและรัศมีของตัวหุ่นยนต์เรียบร้อยแล้วจะทำการหาตำแหน่งรัศมีของสีขาวซึ่งมีเมตริกซ์สีเป็น RGB (255,255,255) เพราะรัศมีที่ต่างกันของเป้าหมายและหัวของหุ่นยนต์จึงทำให้โปรแกรมสามารถแยกแยะความแตกต่างของวัตถุสองตัวนี้ได้ ดังรูป 3.11



รูปที่ 3.11 แสดงการตัด Noise ออกจากระบบการประมวลผลภาพ

ขั้นตอนที่ 3 จัดเก็บตำแหน่งของวัตถุทุกชิ้นให้อยู่ในรูปของตัวแปรเพื่อใช้เป็นอินพุตในการคำนวณหาค่าฟัซซี่ต่อไป ซึ่งในการค้นหาตำแหน่งของสิ่งกีดขวางกรณีที่มีหลายชิ้น ทำได้โดยใช้วิธีการหาตำแหน่งแบบเดียวกับวัตถุอื่นๆ ในการค้นหาเริ่มจากวัตถุที่อยู่ซ้ายมือด้านบนสุด เมื่อค้นพบตำแหน่งของวัตถุชิ้นแรกและตรวจสอบแล้วว่าวัตถุนั้นเป็นสิ่งกีดขวาง จะทำการเก็บค่าตำแหน่งของสิ่งกีดขวางไว้ในตัวแปรจากนั้นทำการลบวัตถุชิ้นนั้นออก และตรวจสอบหาวัตถุชิ้น

ต่อไป โดยเริ่มที่มุมบนด้านซ้ายมืออีกครั้ง จนกระทั่งไม่พบวัตถุที่มีลักษณะเป็นสิ่งกีดขวาง ดังรูป 3.12



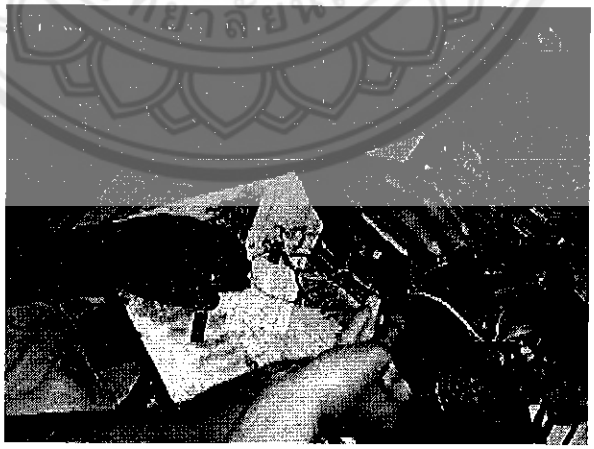
รูปที่ 3.12 แสดงการประมวลผลภาพของสิ่งกีดขวางที่มากกว่า 1 ชิ้น

3.6 การสร้างตัวหุ่นยนต์

การสร้างตัวหุ่นยนต์สามารถเขียนได้เป็นขั้นตอนดังนี้

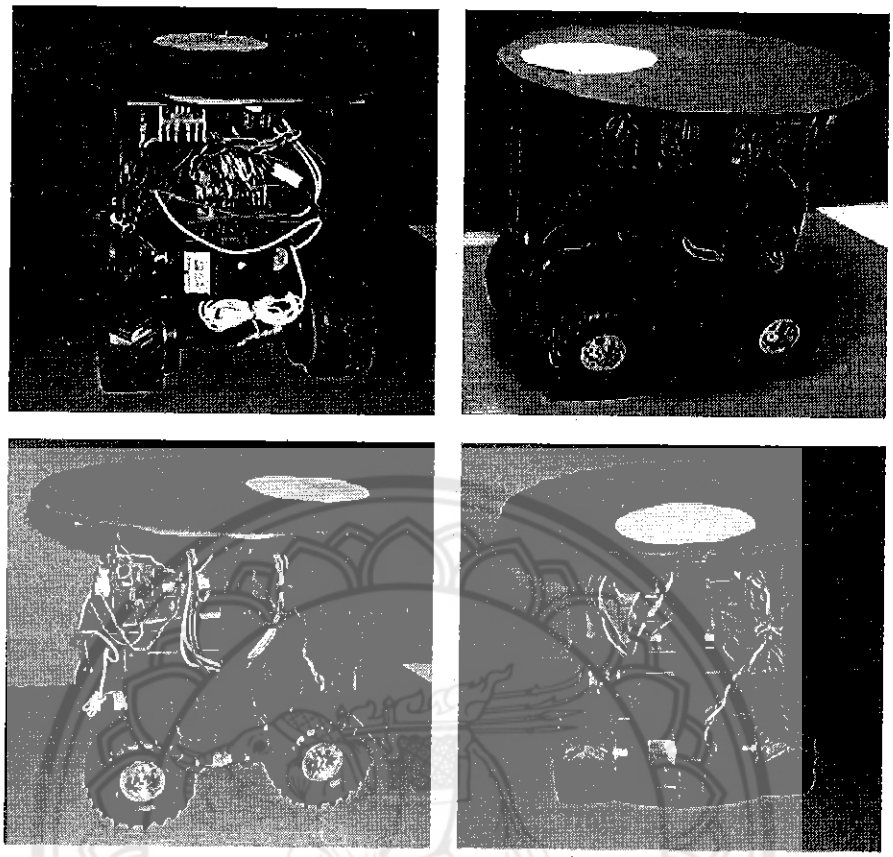
ขั้นตอนที่ 1 การสร้างแผงวงจร การสร้างแผงวงจร ต้องสร้างแผงวงจร 2 ตัว ตัวแรกเป็นแผงวงจร PIC16F877 Microcontroller เป็นตัวควบคุมการทำงานของหุ่นยนต์ และตัวที่สองเป็นตัวขับ (Driver) ใช้ในการขับมอเตอร์ให้ล้อสามารถเคลื่อนที่ได้

ขั้นตอนที่ 2 การสร้างตัวหุ่นยนต์ ต้องใช้แผงวงจรทั้ง 2 ติดเข้ากับโครงสร้างของตัวหุ่นยนต์ ดังรูป 3.13



รูปที่ 3.13 แสดงวิธีการสร้างตัวหุ่นยนต์

<u>ขั้นตอนที่ 3</u> ตกแต่งความสวยงามพร้อมใช้งาน ดังรูป 3.14



รูปที่ 3.14 ภาพหุ่นยนต์

บทที่ 4

ผลการทดลอง

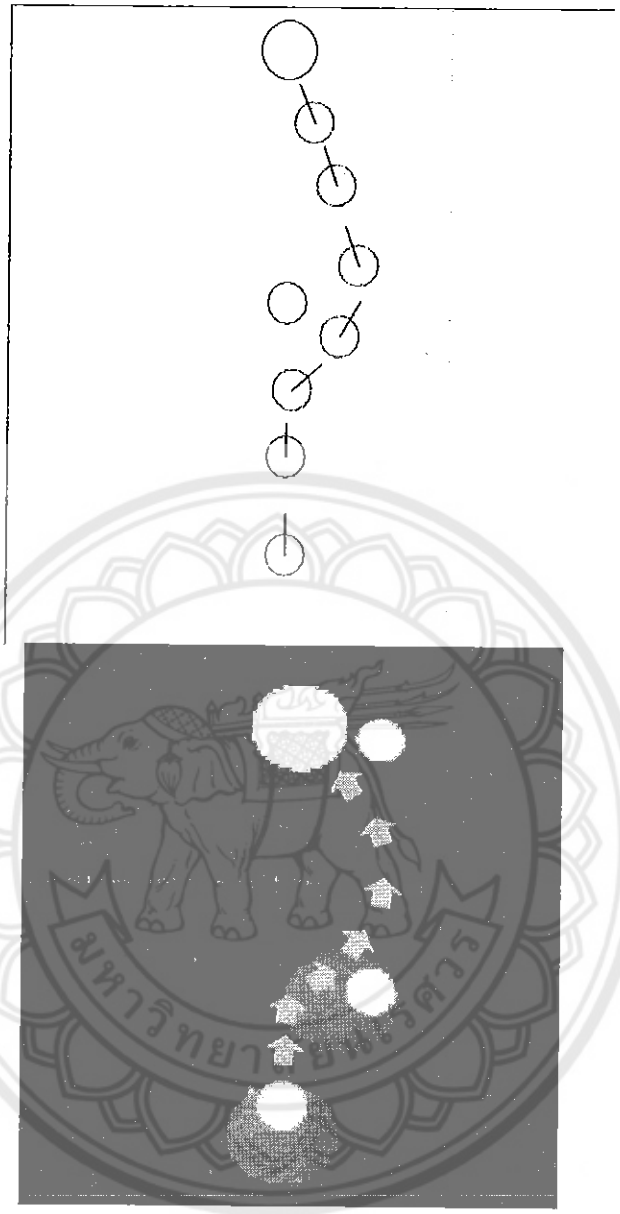
จากวิธีการดำเนินการในบทที่แล้ว จะสามารถแสดงภาพของผลการทดลองการเคลื่อนที่หลบหลีกสิ่งกีดขวางของหุ่นยนต์บนสนามทดสอบทฤษฎี โดยการจำลองสถานการณ์การวางสิ่งกีดขวางและจำนวนสิ่งกีดขวางในรูปแบบที่แตกต่างกันออกไปได้ดังนี้

4.1 ผลการทดลองการเคลื่อนที่หลบหลีกสิ่งกีดขวางเข้าสู่เป้าหมายของหุ่นยนต์

เมื่อเริ่มการทำงาน โปรแกรมจะรับค่าตำแหน่งของสิ่งกีดขวาง และตำแหน่งหุ่นยนต์ แล้วทำการประมวลผล เมื่อไม่มีสิ่งกีดขวางโปรแกรมจะสั่งให้หุ่นยนต์เดินไปยังเป้าหมาย แต่เมื่อมีสิ่งกีดขวาง โปรแกรมจะสั่งให้หุ่นยนต์เดินตามมุมฟัซซี่ เพื่อหลบสิ่งกีดขวาง

ในการทดลองการเคลื่อนที่หลบหลีกสิ่งกีดขวางของหุ่นยนต์ฟัซซี่ เมื่อผลการทดลองการเคลื่อนที่ของหุ่นยนต์จริงออกมาว่าจะต้องเคลื่อนที่ไปในทิศทางใดแล้ว จะนำผลการทดลองนั้นมาเปรียบเทียบกับโปรแกรมจำลองที่เคยมีผู้ศึกษาไว้ก่อนหน้า[2] เพื่อพิจารณาถึงข้อแตกต่างและข้อบกพร่องของโปรแกรมทั้ง 2 ตัว

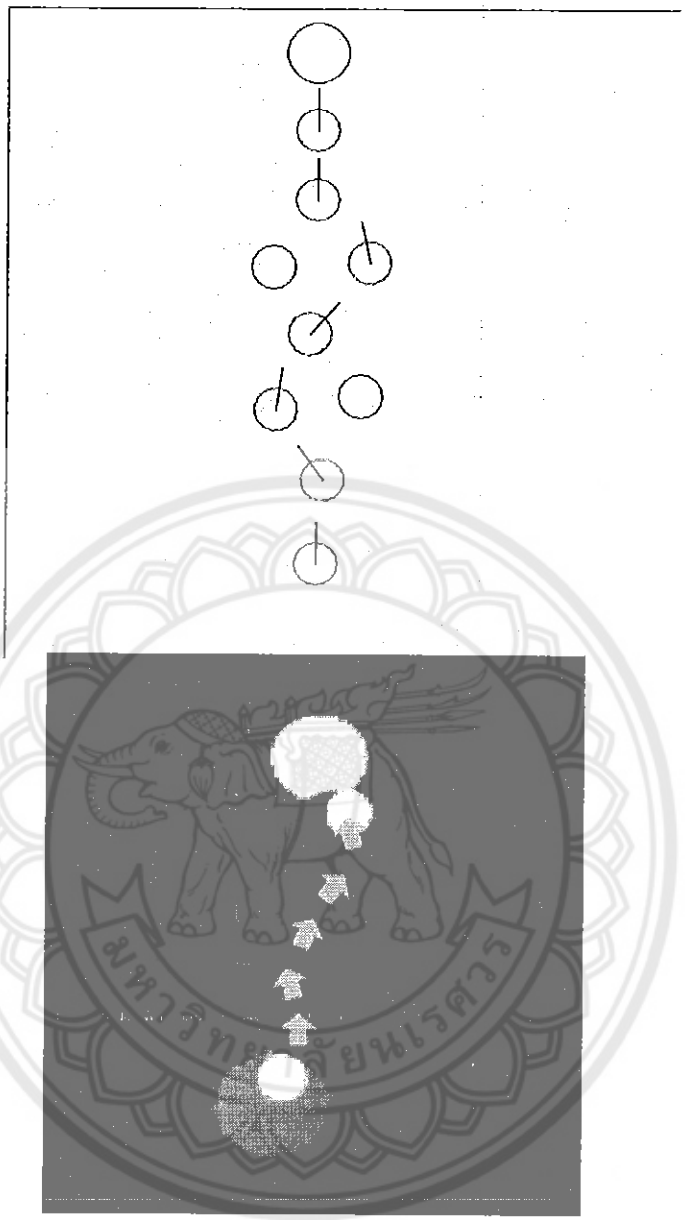
เมื่อกำหนดให้มีสิ่งกีดขวาง 1 ชิ้น ผลการทดลองที่ได้จะเป็นไปตามภาพ



รูปที่ 4.1 ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 1 ชิ้น

จากรูปที่ 4.1 ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์มีการหลบหลีกสิ่งกีดขวางไปทางด้านขวา เมื่อเปรียบเทียบกับโปรแกรม Simulate แล้วจะเห็นว่ามีลักษณะการเคลื่อนที่หลบหลีกสิ่งกีดขวางไปในทิศทางเดียวกัน

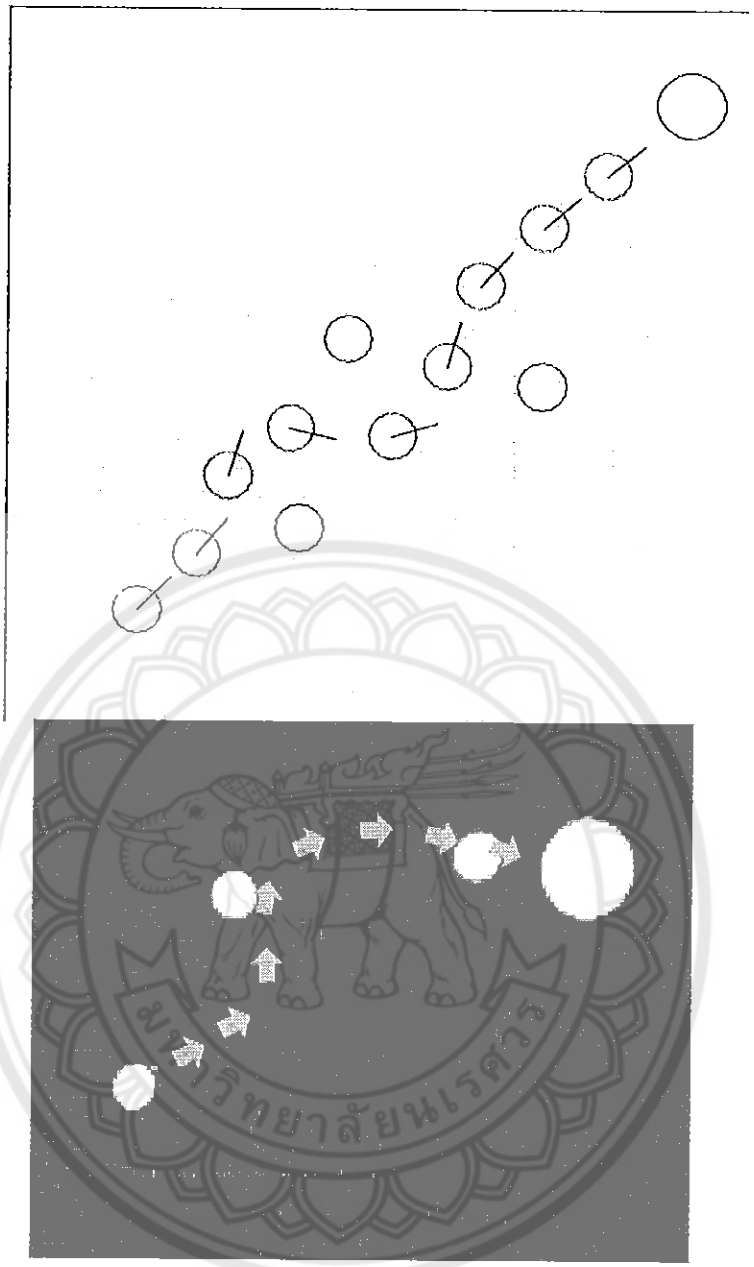
เมื่อกำหนดให้มีสิ่งกีดขวาง 2 ชิ้น ผลการทดลองจะเป็นไปตามภาพ



รูปที่ 4.2 ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 2 ชิ้น

จากรูปที่ 4.2 ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์มีการหลบหลีกสิ่งกีดขวาง 2 ชิ้น โดยมีลักษณะการเคลื่อนที่หลบหลีกสิ่งกีดขวางไปในทิศทางเดียวกันกับโปรแกรม Simulate คือหลบสิ่งกีดขวางชิ้นแรกไปทางด้านขวา และหลบสิ่งกีดขวางชิ้นที่ 2 ไปทางด้านซ้าย และเคลื่อนที่เข้าหาเป้าหมาย

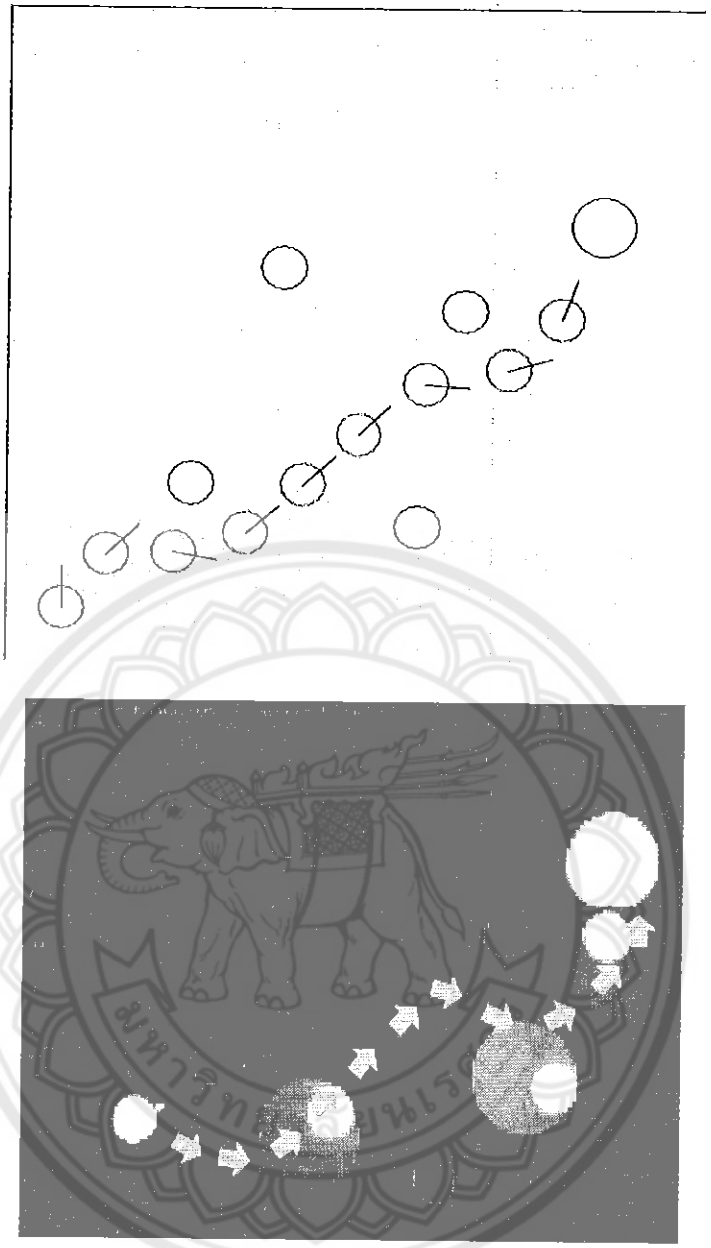
เมื่อกำหนดให้มีสิ่งกีดขวาง 3 ชิ้น ผลการทดลองจะเป็นไปตามภาพ



รูปที่ 4.3 ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 3 ชิ้น

จากรูปที่ 4.3 ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์มีการหลบหลีกสิ่งกีดขวาง 3 ชิ้น โดยมีลักษณะการเคลื่อนที่หลบหลีกสิ่งกีดขวางไปในทิศทางที่ต่างกันกับโปรแกรม Simulate คือหลบวัตถุชิ้นแรกไปทางด้านซ้าย และหลบสิ่งกีดขวางชิ้นที่ 2 ไปทางด้านซ้าย และเคลื่อนที่เข้าหาเป้าหมาย

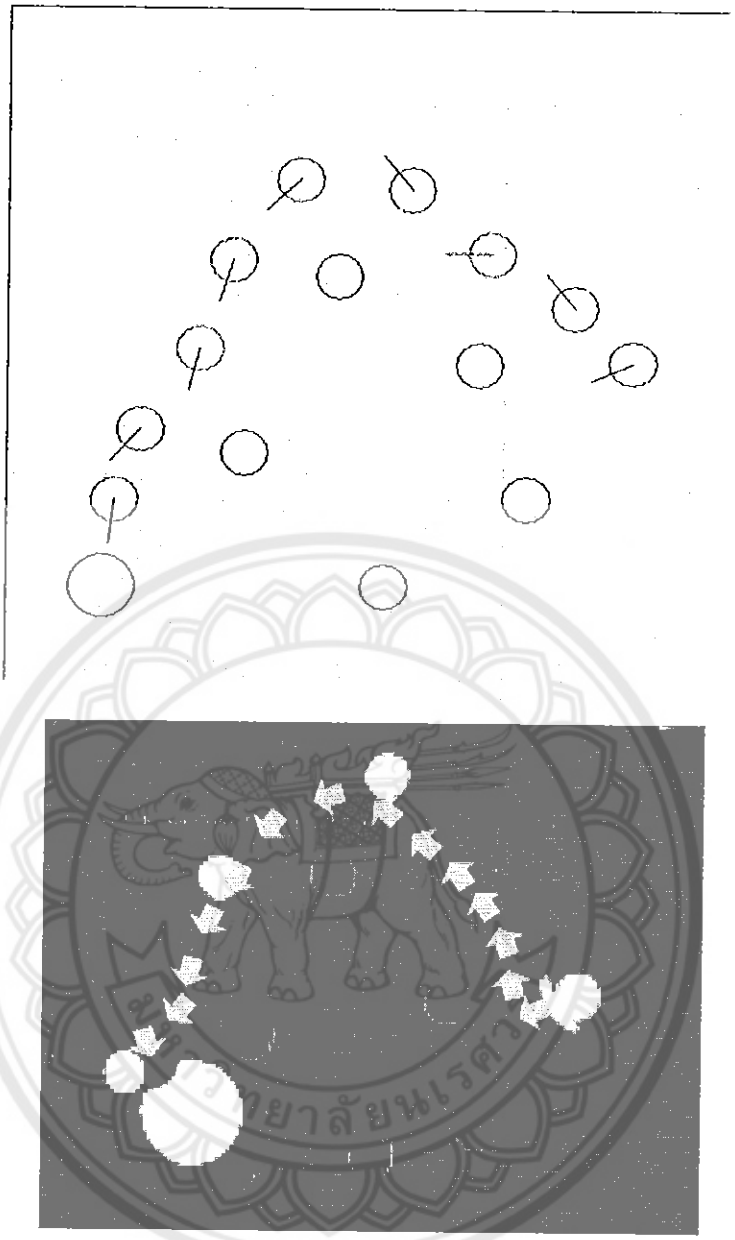
เมื่อกำหนดให้มีสิ่งกีดขวาง 4 ชิ้น ผลการทดลองจะเป็นไปตามภาพ



รูปที่ 4.4 ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 4 ชิ้น

จากรูปที่ 4.4 ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์มีการหลบหลีกสิ่งกีดขวาง 4 ชิ้น โดยมีลักษณะการเคลื่อนที่หลบหลีกสิ่งกีดขวางไปในทิศทางเดียวกันกับโปรแกรม Simulate คือหลบสิ่งกีดขวางชิ้นแรกไปทางด้านขวา เมื่อหุ่นยนต์เคลื่อนที่ไปพบสิ่งกีดขวางที่อยู่ทางขวา หุ่นยนต์จะหลบสิ่งกีดขวางไปทางด้านซ้าย และเคลื่อนที่ไปข้างหน้า พบสิ่งกีดขวางอยู่ด้านหน้า หุ่นยนต์จะตัดสินใจหลบสิ่งกีดขวางไปทางขวา และเคลื่อนที่เข้าหาเป้าหมาย

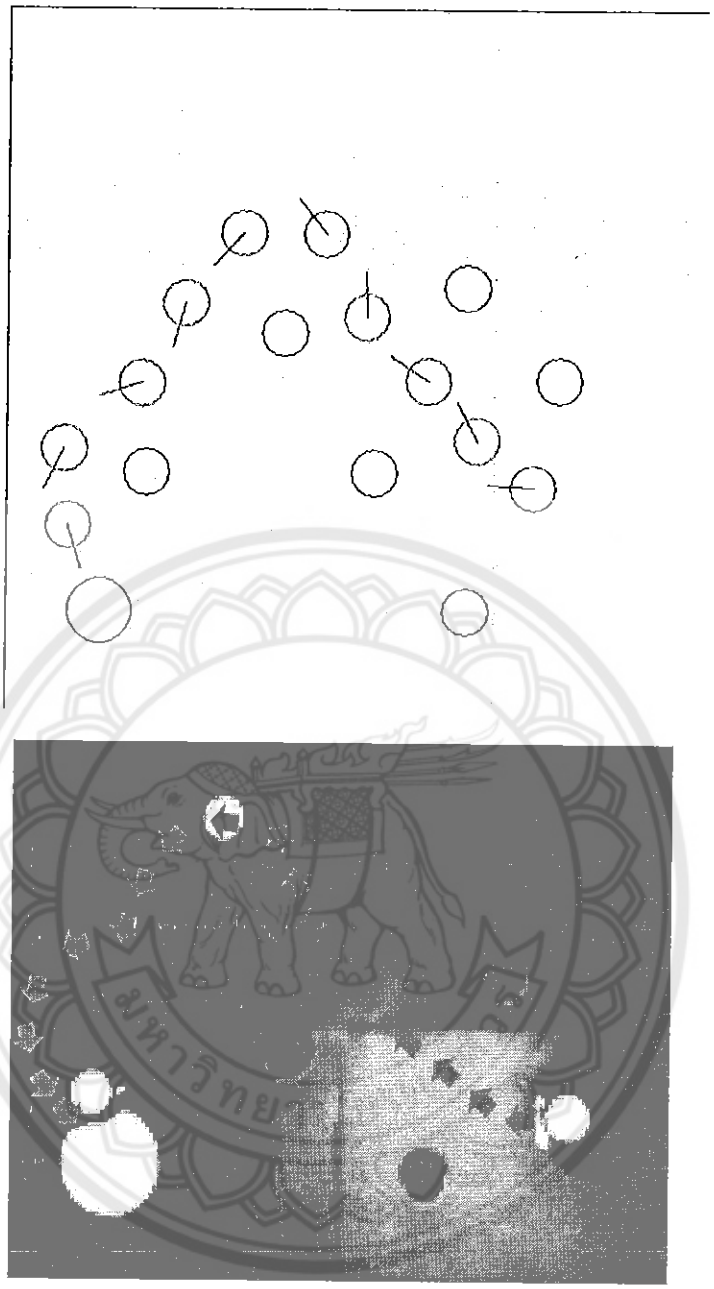
เมื่อกำหนดให้มีสิ่งกีดขวาง 5 ชิ้น ผลการทดลองจะเป็นไปตามภาพ



รูปที่ 4.5 ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 5 ชิ้น

จากรูปที่ 4.5 ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์มีการหลบหลีกสิ่งกีดขวาง 5 ชิ้น โดยมีลักษณะการเคลื่อนที่หลบหลีกสิ่งกีดขวางไปในทิศทางเดียวกันกับโปรแกรม Simulate คือหลบสิ่งกีดขวางชิ้นแรกไปทางด้านหน้า หุ่นยนต์จะหลบสิ่งกีดขวางไปทางขวา เมื่อเคลื่อนที่ไปข้างหน้าหุ่นยนต์พบสิ่งกีดขวางทางด้านซ้าย หุ่นยนต์จะหลบสิ่งกีดขวางไปทางขวา เมื่อเคลื่อนที่ไปข้าหน้าหุ่นยนต์พบสิ่งกีดขวางทางด้านซ้ายอีกครั้ง หุ่นยนต์จะหลบสิ่งกีดขวางไปทางขวา และเคลื่อนที่เข้าหาเป้าหมาย

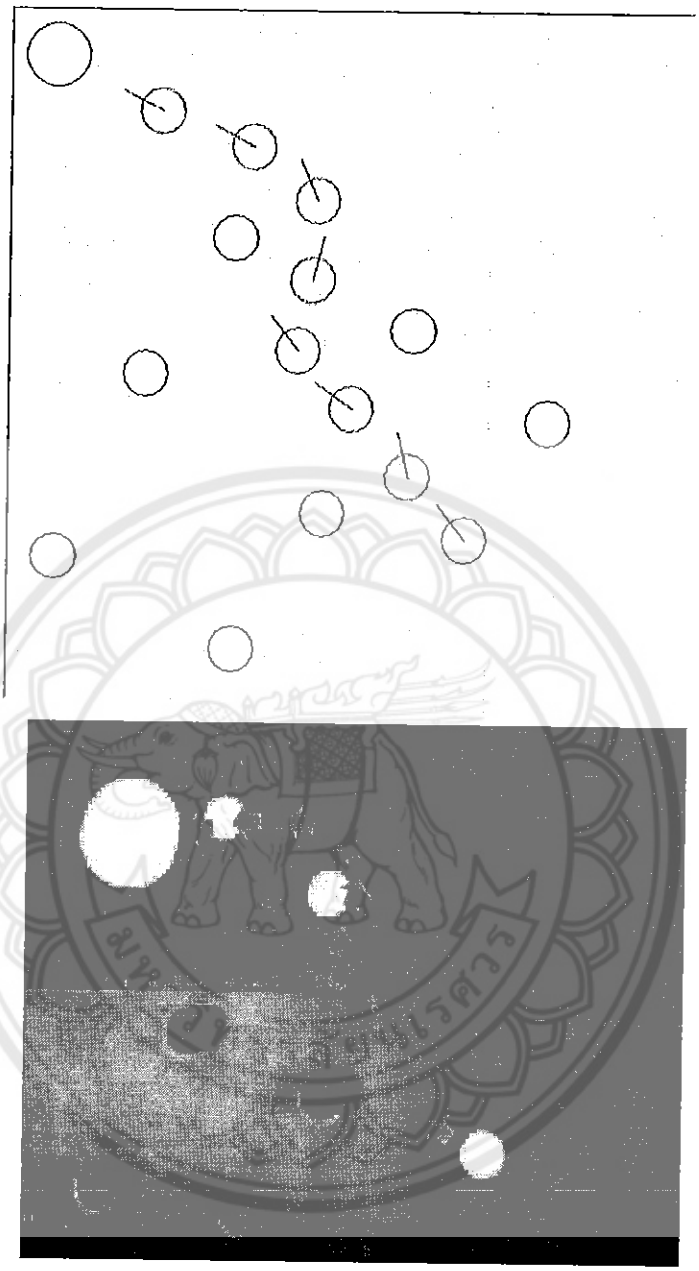
เมื่อกำหนดให้มีสิ่งกีดขวาง 6 ชิ้น ผลการทดลองจะเป็นไปตามภาพ



รูปที่ 4.6 ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 6 ชิ้น

จากรูปที่ 4.6 ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์มีการหลบหลีกสิ่งกีดขวาง 6 ชิ้น โดยมีลักษณะการเคลื่อนที่หลบหลีกสิ่งกีดขวางไปในทิศทางเดียวกันกับโปรแกรม Simulate คือ เมื่อหุ่นยนต์เคลื่อนที่ไปด้านหน้า พบสิ่งกีดขวางในหลายทิศทาง หุ่นยนต์ตัดสินหลบสิ่งกีดขวางที่อยู่ทั้งด้านซ้าย ด้านหน้า และด้านขวา และเคลื่อนที่เข้าหาเป้าหมาย

เมื่อกำหนดให้มีสิ่งกีดขวาง 7 ชิ้น ผลการทดลองจะเป็นไปตามภาพ



รูปที่ 4.7 ภาพแสดงผลการทดลองเมื่อมีสิ่งกีดขวาง 7 ชิ้น

จากรูปที่ 4.7 ผลการทดลองแสดงให้เห็นว่าหุ่นยนต์มีการหลบหลีกสิ่งกีดขวาง 7 ชิ้น โดยมีลักษณะการเคลื่อนที่หลบหลีกสิ่งกีดขวางไปในทิศทางเดียวกันกับโปรแกรม Simulate คือ เมื่อหุ่นยนต์เคลื่อนที่ไปด้านหน้า พบสิ่งกีดขวางในหลายทิศทาง หุ่นยนต์ตัดสินหลบสิ่งกีดขวางที่อยู่ทั้งด้านซ้าย ด้านหน้า และด้านขวา และเคลื่อนที่เข้าหาเป้าหมาย

จากผลการทดลองจะเห็นว่า การเคลื่อนที่จากโปรแกรมจำลอง และการเคลื่อนที่ของหุ่นยนต์จริงนั้น จะเดินไปในทิศทางเดียวกัน อาจมีบางการทดลองที่โปรแกรมจำลอง ไม่สามารถทำได้ คือการเคลื่อนที่ของหุ่นยนต์ที่ตำแหน่งของสิ่งกีดขวางไม่คงที่ และอาจมีบางสถานการณ์ที่หุ่นยนต์จริงมีการเคลื่อนที่ที่คลาดเคลื่อนไปจากโปรแกรมจำลอง ซึ่งสาเหตุอาจเกิดขึ้นจากความผิดพลาดทางด้านไดนามิคของหุ่นยนต์

บทที่ 5
สรุปผล

โครงงานนี้แสดงให้เห็นถึงประสิทธิผลของการนำหลักการฟัซซี่ซิงเกลตันร่วมกับทฤษฎีทางการประมวลผลภาพเพื่อหาตำแหน่งของวัตถุทุกชิ้นในสนามทดสอบทฤษฎีและนำทฤษฎีฟัซซี่ซิงเกลตันมาช่วยในการคำนวณหามุมและทิศทางการเคลื่อนที่ของหุ่นยนต์ให้สามารถหลบหลีกสิ่งกีดขวางและเคลื่อนที่เข้าหาเป้าหมายได้ ในระหว่างการเคลื่อนที่ของหุ่นยนต์ระยะห่างที่ใกล้ที่สุดระหว่างหุ่นยนต์กับวัตถุ (ที่อยู่ด้านซ้าย ด้านหน้าและด้านขวา) จะถูกใช้ในการคำนวณค่าฟัซซี่ จึงสังเกตได้ว่าการจำลองสถานการณ์ต่างๆที่มีจำนวนและตำแหน่งของสิ่งกีดขวางที่แตกต่างกันนั้น การเคลื่อนที่ของหุ่นยนต์จะไม่มีตำแหน่งในการเคลื่อนที่ที่แน่นอน เนื่องจากการใช้ทฤษฎีฟัซซี่ซึ่งเป็นทฤษฎีการควบคุมแบบไม่เป็นเชิงเส้น (Nonlinear System) มีความใกล้เคียงกับการตัดสินใจของสมองมนุษย์เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์

ซอฟต์แวร์ที่ใช้ในการควบคุมหุ่นยนต์นั้น กลุ่มผู้ศึกษาใช้ภาษาวิชวลเบสิก 6.0 ในการพัฒนา เนื่องจากเป็นภาษาที่ง่ายต่อการพัฒนาโปรแกรมและการออกแบบหน้าตาของโปรแกรม (User Interface) เพื่อให้ง่ายต่อการใช้งาน

5.1 ผลการทดลอง

การทดลองเราเริ่มจากสถานการณ์ที่สิ่งกีดขวางและเป้าหมายอยู่ในตำแหน่งคงที่ตลอดการทดลอง ผลการทดลองแสดงให้เห็นถึงประสิทธิผลในการประยุกต์ใช้ทฤษฎีฟัซซี่ซิงเกลตันในการควบคุมหุ่นยนต์ให้สามารถทำมุมหลบหลีกสิ่งกีดขวางและเคลื่อนที่เข้าหาเป้าหมายได้ เราได้ทดสอบในหลากหลายรูปแบบของการวางตำแหน่งสิ่งกีดขวางและเป้าหมาย ไม่ว่าสิ่งกีดขวางจะวางตำแหน่งใดในสนามทดสอบหรือจำนวนของสิ่งกีดขวางมีจำนวนมากน้อยเพียงใด (ไม่เกิน 10 ชิ้น) หุ่นยนต์ก็สามารถเคลื่อนที่หลบหลีกได้อย่างถูกต้องแม่นยำ อันแสดงให้เห็นถึงประสิทธิภาพของการเคลื่อนที่ของหุ่นยนต์ในสถานการณ์ที่กำหนดให้สิ่งกีดขวางและเป้าหมายอยู่ในตำแหน่งคงที่

ในโลกแห่งความเป็นจริงนั้น สิ่งกีดขวางและเป้าหมายอาจจะไม่อยู่คงที่ตลอดเวลาการใช้งาน กลุ่มผู้ทดลองจึงได้ทดสอบโปรแกรมที่พัฒนานั้นกับสถานการณ์ที่สิ่งกีดขวางและเป้าหมายมีการเคลื่อนย้าย ในขณะที่หุ่นยนต์กำลังเคลื่อนที่ โดยเราจะแบ่งเป็น 3 กรณีดังต่อไปนี้

1. กำหนดให้สิ่งกีดขวางสามารถเคลื่อนย้ายตำแหน่งได้
2. กำหนดให้เป้าหมายสามารถเคลื่อนย้ายตำแหน่งได้
3. กำหนดให้เป้าหมายและสิ่งกีดขวางสามารถเคลื่อนย้ายตำแหน่งได้

ในการทดลองที่กำหนดให้สิ่งกีดขวางสามารถเคลื่อนย้ายตำแหน่งได้ในระหว่างทำการทดลอง ผู้ทดลองได้เคลื่อนย้ายตำแหน่งของสิ่งกีดขวางไปมาในระยะระหว่างเป้าหมายกับตัวหุ่นยนต์ ในการทดลองหุ่นยนต์จำเป็นที่จะประมวลผลภาพเพื่อทำการหาตำแหน่งของสิ่งกีดขวางใหม่ทุกครั้งที่มีการเคลื่อนที่ โปรแกรมของเราก็สามารถปรับเปลี่ยนการเคลื่อนที่ของหุ่นยนต์ให้สอดคล้องกับตำแหน่งของสิ่งกีดขวางที่เปลี่ยนไป กล่าวอีกนัยหนึ่งคือหุ่นยนต์สามารถเคลื่อนที่ถึงตำแหน่งเป้าหมายได้ แม้ว่าตำแหน่งของสิ่งกีดขวางจะเปลี่ยนไปจากตำแหน่งเมื่อเริ่มต้น

เช่นเดียวกันกับกรณีการทดลองที่กำหนดให้เป้าหมายสามารถทำการเคลื่อนย้ายตำแหน่งได้ในระหว่างทำการทดลอง หุ่นยนต์จะทำการประมวลผลในการหาตำแหน่งของเป้าหมายและสิ่งกีดขวางใหม่ทุกๆครั้งที่มีการเคลื่อนที่ ด้วยเหตุนี้เองผลการทดลองจึงแสดงให้เห็นถึงประสิทธิภาพการประยุกต์ใช้ทฤษฎีฟัซซี่ชิงเกลตันเพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ให้หลบหลีกสิ่งกีดขวางและเข้าหาตำแหน่งใหม่ของเป้าหมายได้ในทุกๆตำแหน่งที่เป้าหมายเคลื่อนย้ายไป

และการทดลองท้ายสุดกลุ่มผู้ทำการทดลองได้เพิ่มความซับซ้อนด้วยการกำหนดให้ทั้งเป้าหมายและสิ่งกีดขวางสามารถเคลื่อนย้ายตำแหน่งได้ในระหว่างทำการทดลอง เพื่อทดสอบประสิทธิภาพทางความคิดของหุ่นยนต์ให้ใช้ความสามารถในการเคลื่อนที่หลบหลีกสิ่งกีดขวางและเข้าหาเป้าหมายได้ ไม่ว่าสิ่งกีดขวางและเป้าหมายจะอยู่ตำแหน่งใดในสนามทดสอบ ผลการทดลองก็ยังคงแสดงให้เห็นว่าหุ่นยนต์ยังสามารถปรับเปลี่ยนให้เข้ากับสภาวะแวดล้อมที่เปลี่ยนไปและมีความยืดหยุ่นกับทุกๆสถานการณ์ได้เป็นอย่างดี

กล่าวโดยสรุป หุ่นยนต์สามารถแสดงประสิทธิผล, ความยืดหยุ่น, ความสามารถในการปรับเปลี่ยนให้เข้ากับสภาพแวดล้อมที่มีการเปลี่ยนแปลงจากการประยุกต์ใช้ทฤษฎีฟัซซี่ชิงเกิลตันในการคำนวณหามุมเพื่อหลบหลีกสิ่งกีดขวางได้อย่างเหมาะสมและถูกต้อง

5.2 ปัญหาและแนวทางแก้ไข

ความแตกต่างระหว่างการทำงานของหุ่นยนต์ที่มีลักษณะการเดินที่แตกต่างออกไปจากโปรแกรมจำลองเกิดจากความผิดพลาดทางด้านไดนามิคของวงจรการควบคุมการเคลื่อนที่ของหุ่นยนต์ ต้องทำการพัฒนาให้สมบูรณ์ต่อไป

ระบบการประมวลผลภาพส่วนหนึ่งขึ้นอยู่กับประสิทธิภาพของกล้องที่ใช้ในการรับภาพ กล้องที่มีการมองภาพในมุมแคบ จะทำให้มองเห็นพื้นที่สนามทดสอบหุ่นยนต์มีขนาดเล็กลงในระยะความสูงของกล้องที่เท่ากันเมื่อเปรียบเทียบกับกล้องในมุมกว้าง อาจทำให้หุ่นยนต์ไม่สามารถแสดงประสิทธิภาพได้อย่างเต็มที่ อาจแก้ไขได้โดยเลือกใช้กล้องที่มีประสิทธิภาพมากขึ้น

การออกแบบกฎของฟัซซี่ขึ้นอยู่กับประสบการณ์ของผู้ออกแบบ ดังนั้นในแต่ละกฎอาจเกิดข้อผิดพลาดหรือไม่สามารถแก้ไขปัญหาได้ดีเท่าที่ควร โดยเฉพาะในกรณีที่ปัญหามีความซับซ้อน

มากๆ การออกแบบระบบควบคุมการเคลื่อนที่ของหุ่นยนต์ฟัซซี่ซิงเกลตันจะต้องสร้างเงื่อนไขหรือการออกแบบฟังก์ชั่นภาวะสมาชิกให้มีความละเอียดและครอบคลุมมากขึ้น

5.3 สรุปผลการทดลอง

จากผลการทดลองหุ่นยนต์สามารถแสดงประสิทธิภาพการประยุกต์ใช้ทฤษฎีฟัซซี่ซิงเกลตันในการคำนวณหามุมเพื่อหลบหลีกสิ่งกีดขวางและเคลื่อนที่เข้าหาเป้าหมายร่วมกับทฤษฎีการประมวลผลภาพในการหาตำแหน่งของวัตถุทุกชิ้นในสนามทดสอบได้ โดยหุ่นยนต์สามารถเคลื่อนที่หลบหลีกสิ่งกีดขวางและเข้าสู่เป้าหมายได้ทั้งในสภาวะที่คงที่ (เป้าหมายและสิ่งกีดขวางอยู่ในตำแหน่งคงที่ตลอดการทดลอง) หรือสภาวะที่มีการเคลื่อนที่ (เป้าหมายและสิ่งกีดขวางมีการเคลื่อนที่ในขณะที่หุ่นยนต์ทำงาน) ทั้งสองสภาวะนั้นหุ่นยนต์สามารถเคลื่อนที่ได้หลบหลีกสิ่งกีดขวางได้อย่างถูกต้องและสามารถเคลื่อนที่เข้าหาเป้าหมายได้อย่างแม่นยำ แสดงให้เห็นว่าหุ่นยนต์มีความสามารถในการปรับเปลี่ยนให้เข้ากับสภาพแวดล้อมต่างๆและมีสามารถในการยืดหยุ่นได้กับสถานการณ์ต่างๆที่เปลี่ยนไป

5.4 ข้อเสนอแนะ

1. อาจเปลี่ยนโปรแกรมภาษาที่ใช้ในการพัฒนาโปรแกรมการประมวลผลภาพและการควบคุมหุ่นยนต์ เพื่อให้หุ่นยนต์สามารถประมวลผลได้รวดเร็วยิ่งขึ้น

2.ในการสร้างตัวหุ่นยนต์สามารถทำให้หุ่นยนต์สามารถเคลื่อนที่ได้ด้วยวิธีการที่หลากหลาย อาจเลือกใช้ Servo Motor, Step Motor หรือมอเตอร์ชนิดอื่นๆในการพัฒนา

บรรณานุกรม

[1] ธีระวัฒน์ ประกอบผล. **ไมโครโปรเซสเซอร์ หลักการและประยุกต์ใช้งาน.** กรุงเทพฯ : แมคกรอ-ฮิล. 2541.

[2] นางสาวทัดดาว ทับอินทร์ และคณะ. "การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกลตัน". ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า, มหาวิทยาลัยนเรศวร. 2547

[3] จินดารักษ์ กลีบจินดา. "เครื่องคัดแยกผลิตผลทางการเกษตรด้วยกระบวนการวิเคราะห์ภาพ". ปริญญานิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า, มหาวิทยาลัยนเรศวร. 2543.

[4] C.C.Lee. **Fuzzy Logic in Control System : Fuzzy Logic Controller – Part I & II.** San Francisco :1990

[5] C.P Pappis, E.H. Mamdani. F.Gomide. **A Fuzzy Logic Controller For a Traffic Junction.** San Francisco:1977

ภาคผนวก ก

คำอธิบายโปรแกรม

โปรแกรมการ Capture ภาพ

ฟังก์ชัน Fullcolor()

```
For Y = 0 To VideoOCX1.GetHeight - 1
        For X = 0 To VideoOCX1.GetWidth - 1
            If (matrix(X, Y, 1) >= 180) And (matrix(X, Y, 0) >= 180) And (matrix(X, Y, 2) >=
180) Then
                matrix(X, Y, 2) = 0
                matrix(X, Y, 1) = 255
                matrix(X, Y, 0) = 255
            ElseIf (matrix(X, Y, 0) >= 180) And Not (matrix(X, Y, 1) >= 180) And Not
(matrix(X, Y, 2) >= 180) Then
                matrix(X, Y, 2) = 0
                matrix(X, Y, 1) = 0
                matrix(X, Y, 0) = 255
            ElseIf (matrix(X, Y, 2) >= 180) And Not (matrix(X, Y, 0) >= 180) And Not
(matrix(X, Y, 1) >= 180) Then
                matrix(X, Y, 2) = 255
                matrix(X, Y, 1) = 0
                matrix(X, Y, 0) = 0
            Else
                matrix(X, Y, 2) = 0
                matrix(X, Y, 1) = 0
                matrix(X, Y, 0) = 0
            End If
        Next X
    Next Y
```

จากโค้ดเป็นส่วนที่แยกสีโดยกำหนดให้ ถ้าจุดนั้นเป็นสีขาวให้เปลี่ยนเป็นสีเขียวเข้มที่ระดับสูงสุ ถ้าเป็นสีแดงให้เปลี่ยนเป็นสีแดงที่ระดับสูงสุด และถ้าเป็นสีน้ำเงินให้เปลี่ยนเป็นสีน้ำเงินเข้มที่ระดับสูงสุด

```
While (endloopY)
        loopCerY = loopCerY + 4
        If (loopCerY > VideoOCX1.GetHeight - 1) Then
                loopCerY = VideoOCX1.GetHeight - 1
                chkColor = 3
                endloopY = False
                endloopX = False
        Else
                loopCerX = 1
                endloopX = True
        End If

        While (endloopX)
                loopCerX = loopCerX + 4
                If (loopCerX > VideoOCX1.GetWidth - 1) Then
                        loopCerX = VideoOCX1.GetWidth - 1
                        endloopX = False
                End If
                If (matrix(loopCerX, loopCerY, 0) = 255) And Not (matrix(loopCerX,
loopCerY, 1) = 255) Then
                        chkColor = 0
                        '--2-B-------------------------------------------------------------------------
                        endloopY = False
                        endloopX = False
                        '--3-B-------------------------------------------------------------------------
                ElseIf (matrix(loopCerX, loopCerY, 1) = 255) Then
                        chkColor = 1
                        '--2-G-------------------------------------------------------------------------
                        endloopY = False
```

```
                endloopX = False
                '--3-G----------
            ElseIf (matrix(loopCerX, loopCerY, 2) = 255) Then
                chkColor = 2
                '--2-R----------
                endloopY = False
                endloopX = False
                '--3-R----------
            End If
        Wend
    Wend
```

จากโค้ดเป็นการวนลูป while เพื่อหาตำแหน่งของสีน้ำเงิน แดง และ เขียว ในแกน X ,Y จาก Image ที่ทำการ capturer มา แล้วทำการส่งค่า

```
While (readChk(1))  ' left
    If (matrix(lX, lY, chkColor) = 255) Or (matrix(lX, lY, chkColor2) = 255) Then
        lX = lX - 1
        If lX = 0 Then
            readOb(1, 1) = lX
            readOb(1, 2) = lY
            readChk(1) = False
        End If
    Else
        readOb(1, 1) = lX
        readOb(1, 2) = lY
        readChk(1) = False
    End If
Wend

While (readChk(2))  ' right
    If (matrix(rX, rY, chkColor) = 255) Or (matrix(lX, lY, chkColor2) = 255) Then
        rX = rX + 1
```

```
            If rX = VideoOCX1.GetWidth - 1 Then
                readOb(2, 1) = rX
                readOb(2, 2) = rY
                readChk(2) = False
            End If
        Else
            readOb(2, 1) = rX
            readOb(2, 2) = rY
            readChk(2) = False
        End If
    Wend
    cerX = (lX + rX) / 2
    tX = cerX
    While (readChk(3))  ' top
        If (matrix(tX, tY, chkColor) = 255) Or (matrix(lX, lY, chkColor2) = 255) Then
            tY = tY - 1
            If tY = 0 Then
                readOb(3, 1) = tX
                readOb(3, 2) = tY
                readChk(3) = False
            End If
        Else
            readOb(3, 1) = tX
            readOb(3, 2) = tY
            readChk(3) = False
        End If
    Wend
    dX = cerX
    While (readChk(4))  ' down
        If (matrix(dX, dY, chkColor) = 255) Or (matrix(lX, lY, chkColor2) = 255) Then
            dY = dY + 1
            If dY = VideoOCX1.GetHeight - 1 Then
```

```
                    readOb(4, 1) = dX
                    readOb(4, 2) = dY
                    readChk(4) = False
                End If
            Else
                readOb(4, 1) = dX
                readOb(4, 2) = dY
                readChk(4) = False
            End If
        Wend
        cerY = (readOb(3, 2) + readOb(4, 2)) / 2
        cerX = (tX + dX) / 2
        lY = cerY
        lX = cerX
        readChk(1) = True
        readChk(2) = True
        While (readChk(1))  ' left
            If (matrix(lX, lY, chkColor) = 255) Or (matrix(lX, lY, chkColor2) = 255) Then
                lX = lX - 1
                If lX = 0 Then
                    readOb(1, 1) = lX
                    readOb(1, 2) = lY
                    readChk(1) = False
                End If
            Else
                readOb(1, 1) = lX
                readOb(1, 2) = lY
                readChk(1) = False
            End If
        Wend
        rY = cerY
        rX = cerX
```

```
While (readChk(2))  ' right
        If (matrix(rX, rY, chkColor) = 255) Or (matrix(lX, lY, chkColor2) = 255) Then
          rX = rX + 1
          If rX = VideoOCX1.GetWidth - 1 Then
                readOb(2, 1) = rX
                readOb(2, 2) = rY
                readChk(2) = False
          End If
        Else
          readOb(2, 1) = rX
          readOb(2, 2) = rY
          readChk(2) = False
        End If
Wend
readObR(1) = (readOb(2, 1) - readOb(1, 1)) / 2
readObR(2) = (readOb(4, 2) - readOb(3, 2)) / 2
```

จากโค้ดที่ได้ทำการหาตำแหน่ง ด้านบนสุด, ล่างสุด, ซ้ายสุด และ ขวาสุด ของ Object นั้น โดยใช้ Loop While ในการวนหาตำแหน่งนั้นๆ จากนั้นทำการคำนวนหาค่ารัศมีของ Object นั้น ในแนวแกน X แกน Y เพื่อนำไปทำการเช็คค่าของ Object ว่าเป็นวัตถุที่ต้องการต่อไปต่อไป

```
If (readObR(1) > 9) Then
  Blue_r = Blue_r + 1
  Cent(2, 1) = (readOb(2, 1) + readOb(1, 1)) / 2
  Cent(2, 2) = (readOb(4, 2) + readOb(3, 2)) / 2
  For Y = readOb(3, 2) To readOb(4, 2)
        For X = readOb(1, 1) To readOb(2, 1)
              matrix(X, Y, 0) = 0
        Next X
  Next Y
Else
  For Y = readOb(3, 2) To readOb(4, 2)
```

```
            For X = readOb(1, 1) To readOb(2, 1)
                    matrix(X, Y, 0) = 0
            Next X
    Next Y
End If
endloopY = True
endloopX = True
```

เป็นการเช็คค่าสีเขียวจากรัศมีที่ได้ว่าเป็นเป้าหมายหรือหัวหุ่นที่ต้องการหรือไม่ ถ้าใช่ให้ทำการเก็บค่าตำแหน่งศูนย์กลางของวัตถุ แล้วทำการลบสีนั้นออกจากภาพ

```
If (readObR(1) > 9) Then
        If (readObR(1) > 15) Then
                green_b = green_b + 1
                Cent(4, 1) = (readOb(2, 1) + readOb(1, 1)) / 2
                Cent(4, 2) = (readOb(4, 2) + readOb(3, 2)) / 2

                For Y = readOb(3, 2) To readOb(4, 2)
                        For X = readOb(1, 1) To readOb(2, 1)
                                matrix(X, Y, 0) = 0
                                matrix(X, Y, 1) = 0
                        Next X
                Next Y
        Else
                green_s = green_s + 1
                Cent(5, 1) = (readOb(2, 1) + readOb(1, 1)) / 2
                Cent(5, 2) = (readOb(4, 2) + readOb(3, 2)) / 2

                For Y = readOb(3, 2) To readOb(4, 2)
                        For X = readOb(1, 1) To readOb(2, 1)
                                matrix(X, Y, 1) = 0
                        Next X
```

```
                Next Y

            End If
        End If

        endloopY = True
        endloopX = True
```

เป็นการเช็คค่าสีน้ำเงินจากรัศมีที่ได้ว่าเป็นหุ่นที่ต้องการหรือไม่ ถ้าใช่ให้ทำการเก็บค่าตำแหน่งศูนย์กลางของวัตถุ แล้วทำการลบสีนั้นออกจากภาพ

```
If (readObR(1) > 6) And (readObR(2) > 6) Then
        Red_r = Red_r + 1
        Red(Red_r, 5, 1) = (readOb(2, 1) + readOb(1, 1)) / 2
        Red(Red_r, 5, 2) = (readOb(3, 2) + readOb(4, 2)) / 2

        For Y = readOb(3, 2) To readOb(4, 2)
            For X = readOb(1, 1) To readOb(2, 1)
                matrix(X, Y, 2) = 0
            Next X
        Next Y
Else
        For Y = readOb(3, 2) To readOb(4, 2)
            For X = readOb(1, 1) To readOb(2, 1)
                matrix(X, Y, 2) = 0
            Next X
        Next Y
End If
endloopY = True
endloopX = True
```

เป็นการเช็คค่าสีแดงจากรัศมีที่ได้ว่าเป็นสิ่งกีดขวางที่ต้องการหรือไม่ ถ้าใช่ให้ทำการเก็บค่าตำแหน่งศูนย์กลางของวัตถุ แล้วทำการลบสีนั้นออกจากภาพ

โปรแกรมคำนวณเอาท์พุตแบบฟัซซี่ซิงเกลตัน

คำอธิบายของโปรแกรม

การประกาศตัวแปร

```
Dim j1, j2 As Boolean
Dim cX(10), cY(10) As Single
Dim tgX, tgY, htX, htY, rhX, rhY  As Single
Dim tgD, tgH, tgR As Single
Dim AngR, AngH, AngT As Single
Dim a(3), b(3) As Single
Dim Lmin, Tmin, Rmin, Oout As Single Dim PI As Single
Dim RegionsStart As Single
Dim RegionsStop As Single
Dim angle As Single
Dim FirstRun As Boolean
```

ฟังก์ชั่นจับภาพจากกล้อง

```
Private Sub cmdCapture_Click()
   cmdCapture.Enabled = False
   cmdClose.Enabled = True
   cmdStep.Enabled = True
   Walk.Enabled = True
   VideoOCX1.Mode = 0
   VideoOCX1.Init

   m_Image = VideoOCX1.GetColorImageHandle

   VideoOCX1.SetErrorMessages (False)
   If Not VideoOCX1.Init Then
        MsgBox VideoOCX1.GetLastErrorString
   Else
```

```
        m_Image = VideoOCX1.GetColorImageHandle
        VideoOCX1.SetPreview (True)
        VideoOCX1.Start
    End If
End Sub
```

จากฟังก์ชั่นจับภาพจากกล้อง ซึ่งจะใช้ Control ชื่อว่า VideoOCX ในการจับภาพจากกล้อง

ฟังก์ชั่นปิดการจับภาพจากกล้อง

```
Private Sub cmdClose_Click()
    cmdCapture.Enabled = True
    cmdClose.Enabled = False
    cmdStep.Enabled = False
    Walk.Enabled = False
    VideoOCX1.Stop
    VideoOCX1.Close
End Sub
```

ฟังก์ชั่นล้างตัวแปร

```
Private Sub cmdClear_Click()
    Dim i As Integer
    For i = 1 To 10
        cX(i) = 0
        cY(i) = 0
    Next i
    a(1) = 0
    b(1) = 0
    a(2) = 0
    b(2) = 0
    a(3) = 0
    b(3) = 0
    j1 = True
    j2 = True
```

```
    FirstRun = False
    tgR = 0
    tgD = 0
    tgH = 0
    AngR = 0
    AngH = 0
    AngT = 0
    Oout = 0
    Lmin = 0
    Tmin = 0
    Rmin = 0
    AngleRobotUse.Caption = 0
    Monitor
End Sub
```

จากฟังก์ชั่นล้างตัวแปรจะเป็นการกำหนดค่าเริ่มต้นให้กับโปรแกรมและตัวแปรต่าง ๆ

ฟังก์ชั่นกำหนดค่าเริ่มต้นเมื่อเริ่มเปิดโปรแกรม

```
Private Sub Form_Load()
   mscSender.CommPort = 1
   mscSender.Settings = "1200,n,8,1"
   mscSender.PortOpen = True
   PI = 3.14159265358979
   j1 = True
   j2 = True
   FirstRun = False
End Sub
```

จากฟังก์ชั่นกำหนดค่าเริ่มต้นเมื่อเปิดโปรแกรม จะมีการเรียกใช้งาน Com Port 1 ซึ่งมีการตั้งค่า Baud Rate อยู่ที่ 1200 bit/s

ฟังก์ชั่นการทำงานของหุ่น

```
Private Sub StepWalk()
```

```
    If Not FirstRun Then First
    FullColor
    init_robot
    RobotVector
    ObjVector
    RobotWalk
    Monitor
End Sub
```

ฟังก์ชั่นที่ทำงานเมื่อเริ่มทำงานครั้งแรก

```
Private Sub First()
    RegionsStart = StartReg.Text                    'รับค่าขอบเขตของฟัซซี่
    RegionsStop = StopReg.Text                      'รับค่าขอบเขตของฟัซซี่
    FirstRun = True
End Sub
```

ฟังก์ชั่นกำหนดค่าเริ่มต้นของหุ่นยนต์

```
Private Sub init_robot()
'อ่านค่าจำนวนสิ่งกีดขวางจากตัวแปร red_r
   For i = 1 To Red_r
      cX(i) = Red(i, 5, 1)                          'อ่านค่าจุดศูนย์กลางสิ่งกีดขวาง
      cY(i) = Red(i, 5, 2)
   Next i
End Sub
```

ฟังก์ชั่นหาระยะทางและมุมของหุ่นกับเป้าหมาย

```
Private Sub RobotVector()
   a(1) = Cent(2, 1)                                'อ่านค่าจุดศูนย์กลางหุ่น
   b(1) = Cent(2, 2)                                'อ่านค่าจุดศูนย์กลางหุ่น
   a(2) = Cent(4, 1)                                'อ่านค่าจุดศูนย์กลางเป้าหมาย
   b(2) = Cent(4, 2)                                'อ่านค่าจุดศูนย์กลางเป้าหมาย
   a(3) = Cent(5, 1)                                'อ่านค่าจุดศูนย์กลางหัวหุ่น
   b(3) = Cent(5, 2)                                'อ่านค่าจุดศูนย์กลางหัวหุ่น
```

```
    'ส่วนคำนวณหาเวกเตอร์และระยะทาง หัวหุ่นกับเป้าหมาย
    htX = a(2) - a(3)
    htY = b(2) - b(3)
    tgH = Sqr(htX ^ 2 + htY ^ 2)                    'ระยะระหว่างหัวกับเป้าหมาย
    'ส่วนคำนวณหาเวกเตอร์และระยะทาง หุ่นกับเป้าหมาย
    tgX = a(2) - a(1)
    tgY = b(2) - b(1)
    tgD = Sqr(tgX ^ 2 + tgY ^ 2)                    'ระยะทางจากหุ่นไปเป้าหมาย
    'ส่วนคำนวณหาเวกเตอร์และระยะทาง หุ่นกับหัวหุ่น
    rhX = a(3) - a(1)
    rhY = b(3) - b(1)
    tgR = Sqr(rhX ^ 2 + rhY ^ 2)                    'ระยะทางหุ่นถึงหัวหุ่น
'คำนวนหามุม
    AngT = AngleCal(tgR, tgD, tgH)                  'มุมเป้าหมาย
    AngH = AngleCal(tgD, tgR, tgH)                  'มุมหัว
    AngR = AngleCal(tgH, tgD, tgR)                  'มุมหุ่น
    'เวกเตอร์หุ่น-หัว ครอส หุ่น-เป้าหมาย
    If CrossVector(tgX, tgY) < 0 Then
      AngR = AngR * (-1)
    End If
End Sub
```

ฟังก์ชั่นหาเวกเตอร์หาทิศทางของเวกเตอร์

```
Private Function CrossVector(ByVal X As Single, ByVal Y As Single) As Single
    CrossVector = (rhX * Y) - (rhY * X)
End Function
```

จากฟังก์ชั่น เป็นการหาทิศทางของมุม ว่าอยู่ด้านขวาหรือด้านซ้ายของหุ่น โดยใช้หลักการการครอสกันของเวกเตอร์ เมื่อครอสกันแล้วได้ค่าเป็นบวก จะได้ว่ามุมอยู่ทางด้านซ้าย และถ้ามุมเป็นลบ จะได้ว่ามุมอยู่ทางด้านขวา

ฟังก์ชั่นหาระยะทางและมุมของหุ่นกับสิ่งกีดขวาง

```
Private Sub ObjVector()
   Dim j, k As Single
   Dim n, m As Single
   Dim DistRO, DistHO  As Single
   Dim AngleObj As Single
   Dim dL, dR, dT As Single
   Dim i As Integer

   Rmin = 1000
   Lmin = 1000
   Tmin = 1000

   For i = 1 To Red_r
     'เวกเตอร์จากโรบอทไปวัตถุ
     j = cX(i) - a(1)
     k = cY(i) - b(1)
     DistRO = Sqr(j ^ 2 + k ^ 2)
     'เวกเตอร์จากหัวไปวัตถุ
     n = cX(i) - a(3)
     m = cY(i) - b(3)
     DistHO = Sqr(n ^ 2 + m ^ 2)
     AngleObj = AngleCal(DistHO, DistRO, tgR)
     If CrossVector(j, k) < 0 Then
        AngleObj = AngleObj * (-1)
     End If

     If AngleObj <= 30 And AngleObj >= -30 Then
          If DistRO < Tmin Then Tmin = DistRO
     ElseIf AngleObj >= -90 And AngleObj < -30 Then
          If DistRO < Lmin Then Lmin = DistRO
```

```
    ElseIf AngleObj <= 90 And AngleObj > 30 Then
        If DistRO < Rmin Then Rmin = DistRO
    End If
  Next i
End Sub
```

จากฟังก์ชั่นหาค่ามุมและระยะทางของสิ่งกีดขวาง เมื่อได้ค่ามุมออกมาแล้ว จะแบ่งช่วงของมุมออกเป็น 3 ช่วง คือช่วง -30 – 30 องศา เป็นช่วงด้านหน้าของหุ่น ช่วง 30 – 90 องศา เป็นเป็นช่วงด้านขวาของหุ่น และช่วง -30 – -90 เป็นช่วงด้านซ้ายของหุ่น

ฟังก์ชั่นสั่งให้หุ่นเดิน

```
Private Sub RobotWalk()
  Dim tmp As Integer
  Dim i As Single
  Oout = Rule

  If tgD < 40 Then
    cmdStopWalk_Click
    MsgBox "Target Complete!", , "Report"
    Exit Sub
  End If
  If Oout = 0 Then
    tmp = Round(AngR)
  Else
    tmp = Round(Oout)
  End If
  AngleRobotUse.Caption = tmp

  If tmp >= 5 And tmp < 15 Then
      For i = 0 To 4
        cmdRight
      Next
```

```
        cmdStop
      noRotate (4)
    ElseIf tmp >= 15 And tmp < 30 Then
        For i = 0 To 8
          cmdRight
        Next
        cmdStop
      noRotate (4)
    ElseIf tmp >= 30 And tmp < 180 Then
        For i = 0 To 12
          cmdRight
        Next
        cmdStop
      noRotate (4)
    ElseIf tmp <= -4 And tmp > -15 Then
        For i = -4 To 0
          cmdLeft
        Next
        cmdStop
      noRotate (4)
    ElseIf tmp <= -15 And tmp > -30 Then
        For i = -8 To 0
          cmdLeft
        Next
        cmdStop
      noRotate (4)
    ElseIf tmp <= -30 And tmp > -180 Then
        For i = -12 To 0
          cmdLeft
        Next
        cmdStop
      noRotate (4)
```

```
ElseIf tmp = 180 Then
    cmdReverse
Else
  noRotate (4)
End If

End Sub
```

จากฟังก์ชั่น เป็นการเดินของหุ่น ใช้มุมจากมุมฟัซซี่ (Oout) ซึ่งถ้ามุมฟัซซี่เป็น 0 แล้ว จะใช้มุมหุ่นกับเป้าหมาย เพื่อให้หุ่นเดินไปยังเป้าหมาย โดยมีการแบ่งช่วงของมุม เป็นช่วง ๆ ในการกำหนดให้หุ่นเดินหน้า,หมุนซ้าย หรือ หมุนขวา

ฟังก์ชั่นคำนวณหามุม

```
Public Function AngleCal(ByVal a As Single, ByVal b As Single, ByVal c As Single) As Single
If b <> 0 And c <> 0 Then
  AngleCal = Acos((b ^ 2 + c ^ 2 - a ^ 2) / (2 * b * c))
Else
  AngleCal = 0
End If
End Function
```

ฟังก์ชั่นคำนวณหามุม โดยจะให้หลักของมุมในสามเหลี่ยม ซึ่งมีอินพุตเป็นด้าน 3 ด้านของสามเหลี่ยม จะได้เอาท์พุตเป็นมุม ซึ่งจากโค้ด จะเป็นมุมตรงข้ามกับด้าน a

ฟังก์ชั่นคำนวณหา Arccos

```
Public Function Acos(Xcos As Variant) As Single
 If Xcos > 1 Or Xcos < -1 Then
  Acos = 0
 ElseIf Xcos = 1 Then
  Acos = 0
 ElseIf Xcos = -1 Then
  Acos = (PI / 2) * 180 / PI
 Else
  Acos = (Atn(-Xcos / Sqr(-Xcos * Xcos + 1)) + 2 * Atn(1)) * 180 / PI
```

```
  End If
End Function
```

ฟังก์ชั่นแสดงตำแหน่ง,ระยะทางและมุมของหุ่น

```
Private Function Monitor()
   DistanceRH.Caption = Round(tgR, 2)
   DistanceRT.Caption = Round(tgD, 2)
   DistanceHT.Caption = Round(tgH, 2)
   AngleRobot.Caption = Round(AngR, 2)
   AngleHead.Caption = Round(AngH, 2)
   AngleTarget.Caption = Round(AngT, 2)
   PointRX.Caption = a(1)
   PointRY.Caption = b(1)
   PointTX.Caption = a(2)
   PointTY.Caption = b(2)
   PointHX.Caption = a(3)
   PointHY.Caption = b(3)
   AngleFuzzy.Caption = Round(Oout, 2)
   DistanceLeft.Caption = Round(Lmin, 2)
   DistanceTop.Caption = Round(Tmin, 2)
   DistanceRight.Caption = Round(Rmin, 2) End Function
```

ฟังก์ชั่นคำนวณฟัซซี่

```
Public Function Rule()
   Dim Lt1 As Single
   Dim Lt2 As Single
   Dim Rt1 As Single
   Dim Rt2 As Single
   Dim F1 As Single
   Dim F2 As Single
   Dim ValueRule1, ValueRule2 As Single
   Dim L, R, F As Single
```

```
    L = Lmin
    R = Rmin
    F = Tmin
    If L > RegionsStop Then L = RegionsStop
    If R > RegionsStop Then R = RegionsStop
    If F > RegionsStop Then F = RegionsStop
'คำนวนหาขอบเขตของฟัซซี่
    Lt1 = Fuzzy1(L)
    Lt2 = Fuzzy2(L)
    Rt1 = Fuzzy1(R)
    Rt2 = Fuzzy2(R)
    F1 = Fuzzy1(F)
    F2 = Fuzzy2(F)
'คำนวนหาเอาท์พุตจากกฎของฟัซซี่
    ValueRule1 = Output1(Lt1, Rt1, F1, 180)
    ValueRule1 = ValueRule1 + Output1(Lt1, Rt1, F2, 0)
    ValueRule1 = ValueRule1 + Output1(Lt1, Rt2, F1, 60)
    ValueRule1 = ValueRule1 + Output1(Lt1, Rt2, F2, 30)
    ValueRule1 = ValueRule1 + Output1(Lt2, Rt1, F1, (-60))
    ValueRule1 = ValueRule1 + Output1(Lt2, Rt1, F2, (-30))
    ValueRule1 = ValueRule1 + Output1(Lt2, Rt2, F1, 30)
    ValueRule1 = ValueRule1 + Output1(Lt2, Rt2, F2, 0)
    ValueRule2 = Output2(Lt1, Rt1, F1)
    ValueRule2 = ValueRule2 + Output2(Lt1, Rt1, F2)
    ValueRule2 = ValueRule2 + Output2(Lt1, Rt2, F1)
    ValueRule2 = ValueRule2 + Output2(Lt1, Rt2, F2)
    ValueRule2 = ValueRule2 + Output2(Lt2, Rt1, F1)
    ValueRule2 = ValueRule2 + Output2(Lt2, Rt1, F2)
    ValueRule2 = ValueRule2 + Output2(Lt2, Rt2, F1)
    ValueRule2 = ValueRule2 + Output2(Lt2, Rt2, F2)
' คำนวณหาเอาท์พุตค่าที่ได้คือองศาที่จะต้องหมุน
```

```
    If ValueRule2 <> 0 Then
      Rule = ValueRule1 / ValueRule2
    Else
      Rule = 0
      MsgBox "Div by 0", , "error"
    End If
End Function
```

ฟังก์ชั่นหาขอบเขตฟัซซี่

```
Public Function Fuzzy1(ByVal InputReg As Single) As Single
    If InputReg >= 0 And InputReg <= RegionsStart Then
      Fuzzy1 = 1
    ElseIf InputReg > RegionsStart And InputReg <= RegionsStop Then
      Fuzzy1 = (RegionsStop - InputReg) / (RegionsStop - RegionsStart)
    Else
      Fuzzy1 = 0
    End If
End Function
```

ฟังก์ชั่นหาขอบเขตฟัซซี่

```
Public Function Fuzzy2(ByVal InputReg As Single) As Single
    If InputReg >= 0 And InputReg <= RegionsStart Then
      Fuzzy2 = 0
    ElseIf InputReg > RegionsStart And InputReg <= RegionsStop Then
      Fuzzy2 = (InputReg - RegionsStart) / (RegionsStop - RegionsStart)   '(x-a1)/(a2-a1)
    Else
      Fuzzy2 = 1
    End If
End Function
```

ฟังก์ชั่นหาเอาพุต

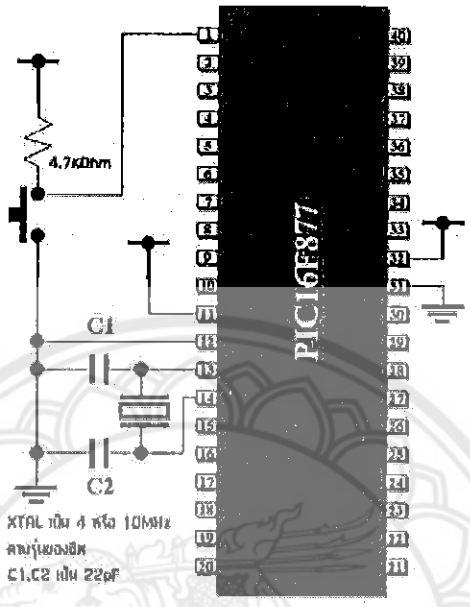
```
Public Function Output1(ByVal a As Single, ByVal b As Single, ByVal c As Single, ByVal
Angle As Single) As Single
   Output1 = a * b * c * Angle
End Function
```

ฟังก์ชั่นหาเอาพุต

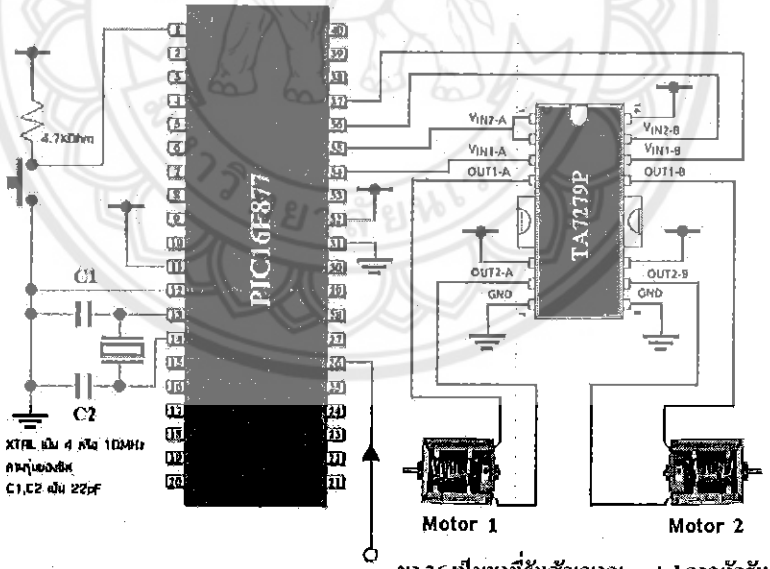
```
Public Function Output2(ByVal a As Single, ByVal b As Single, ByVal c As Single) As Single
   Output2 = a * b * c
End Function
```

ภาคผนวก ข

ส่วนของวงจรหุ่นยนต์



รูปที่ ข.1 วงจรการทำงานของ PIC16F877



ขา 26 เป็นขาที่รับสัญญาณ serial จากตัวรับ Wirless

รูปที่ ข.2 แสดงวงจรการส่ง Output ให้มอเตอร์ทำงาน

จากรูป เป็นวงจรการทำงานของ PIC16F877 ส่ง Output ไปยัง TA7279 ให้สั่งงาน Motor ทั้ง 2 ตัว เพื่อบังคับการทำงานของหุ่น

<u>ส่วนโค้ดโปรแกรมของ PIC16F877</u>

ในส่วนนี้ได้ใช้โปรแกรม mikroBasic ในการเขียนโปรแกรม ซึ่งโปรแกรมที่เขียนเป็นภาษาเบสิก

รูปที่ ข.3 แสดงรูปโปรแกรม Mikro Basic

โค้ดของโปรแกรม

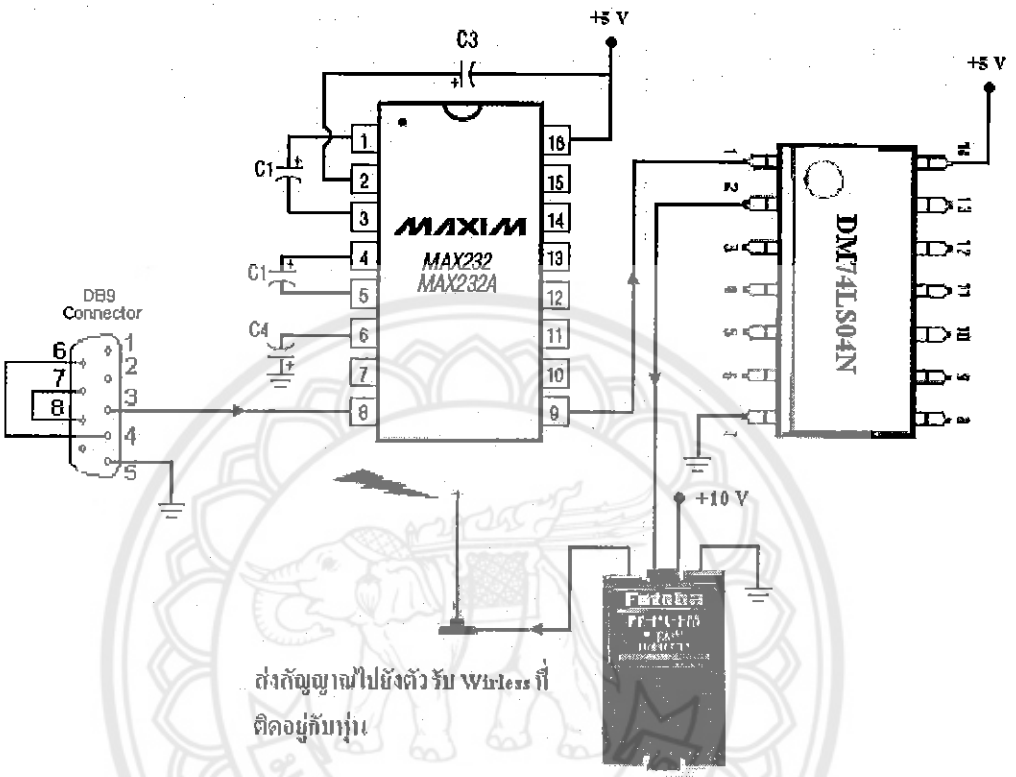
```
program FuzzyRobot
dim data as byte
main:
        TRISB = 0
        Usart_init(1200)
        while true
                if USART_data_ready = 1 then
                data = USART_read
                USART_Write(data)
                PORTB = data
        End If
        Wend
end.
```

จากโค้ดโปรแกรมกำหนดให้ port B เป็น Output และกำหนด baud rate เท่ากับ 1200 แล้ววนลูป while รับค่า serial

ภาคผนวก ค.

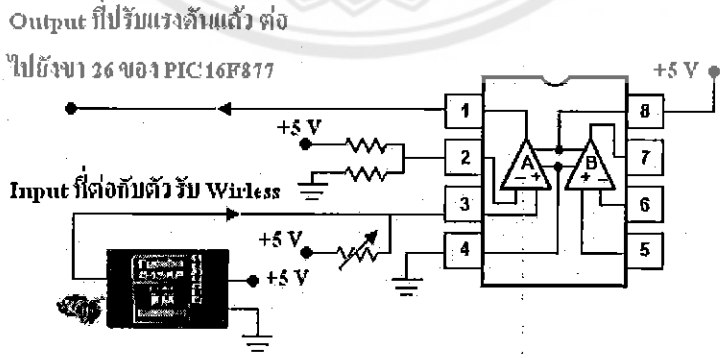
Wireless

ภาคส่งสัญญาณ Wireless



รูปที่ ค.1 แสดงภาคส่งของสัญญาณ Wireless

ภาครับสัญญาณ Wireless



รูปที่ ค.2 แสดงภาครับของสัญญาณ Wireless