การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกิลตัน

Mobile Robot Control using fuzzy singleton

นางสาวทัดดาว ทับอินทร์ รหัส 44362226

นางสาวธนพร คงพันธุ์ รหัส 44362234

นายวิริยะ บุญสนอง รหัส 44362390

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 พ.ค. 2553
เลขทะเบียน 150166911
เลขเรียกหนังสือ ช ร
มหาวิทยาลัยนเรศวร
2547
C.2

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2547

หัวข้อโครงงาน	**การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ชิงเกิลตัน**	
ผู้ดำเนินโครงงาน	**นางสาวทัดดาว**	**ทับอินทร์ รหัส 44362226**
	นางสาวชนพร	**คงพันธุ์ รหัส 44362234**
	นายวิริยะ	**บุญสนอง รหัส 44362390**
อาจารย์ที่ปรึกษา	ดร.สมยศ	เกียรติวนิชวิไล
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2547	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

..... **ประธานกรรมการ**

(ดร. สมยศ เกียรติวนิชวิไล)

..... **กรรมการ**

(ดร. ธนิต มาลากร)

..... **กรรมการ**

(ดร. สุรเชษฐ์ กานต์ประชา)

หัวข้อโครงงาน	การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกิลตัน		
ผู้ดำเนินโครงงาน	นางสาวทัดดาว	ทับอินทร์	รหัส 44362226
	นางสาวธนพร	คงพันธุ์	รหัส 44362234
	นายวิริยะ	บุญสนอง	รหัส 44362390
อาจารย์ที่ปรึกษา	ดร.สมยศ	เกียรติวนิชวิไล	
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2547		

บทคัดย่อ

โครงงานนี้ศึกษาและพัฒนาโปรแกรมสำหรับการควบคุมหุ่นยนต์เคลื่อนที่โดยใช้ทฤษฎีฟัซซี่ซิงเกิลตัน ฟัซซี่ลอจิกเป็นเครื่องมือที่มีประโยชน์สำหรับการแก้ไขปัญหาในการใช้งานจริงที่เกิดจากความไม่เที่ยงตรงและความคลาดเคลื่อนของข้อมูล ตัวควบคุมแบบฟัซซี่ลอจิกจึงถูกใช้ในการควบคุมในการใช้งานต่างๆอย่างแพร่หลาย ในโครงงานนี้ได้ออกแบบและจำลองผลการทำงานด้วยคอมพิวเตอร์ของฟัซซี่ซิงเกิลตันในการควบคุมการหลบหลีกสิ่งกีดขวางของหุ่นยนต์ โปรแกรมที่พัฒนาขึ้นนี้จะถูกทดสอบและทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ XP ภาษาที่ใช้ในการพัฒนาโปรแกรมคือ ภาษาวิชวลเบสิกเวอร์ชั่น 6.0

โปรแกรมแบ่งออกเป็นสองส่วน ส่วนแรกเป็นส่วนของกราฟฟิกแสดงรูปหุ่นยนต์เคลื่อนที่และการสร้างสิ่งกีดขวาง ส่วนที่สองจะประกอบไปด้วยส่วนของฟัซซี่ไฟล์เออร์, กฎของฟัซซี่และฟัซซี่ซิงเกิลตันของตัวควบคุมหุ่นยนต์เคลื่อน ผลการจำลองการทำงานของคอมพิวเตอร์แสดงถึงประสิทธิผลของตัวควบคุมแบบฟัซซี่ในการควบคุมหุ่นยนต์เคลื่อนที่

Project title	Mobile Robot Control using fuzzy singleton		
Name	Miss Thuddoaw	Thab-in	ID. 44362226
	Miss Tanaporn	Kongpun	ID. 44362234
	Mr. Viriya	Bunsanong	ID. 44362390
Project advisor	Dr. Somyot	Kiattivanichvilai	
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic year	2004		

.....

Abstract

This project studies and develop a program for control mobile robot using fuzzy singleton. Fuzzy logic is a useful tool for solving many real-world problems characterized by imprecise and uncertain information. Fuzzy logic controllers are widely used to control in many application. A development this program test and runs on Microsoft Windows XP Operating System. This program is develop using Visual Basic version 6.0 language. In this project, Obstacle avoidance control of a mobile robot using fuzzy singleton is designed and simulated.

The simulation is implemented in two parts. The first part pertains to the graphical aspects depicting robot movement and plotting obstacles. The second part deals with the fuzzifier, the fuzzy rule base and the fuzzy singleton of the mobile robot controller. The simulation results show the effectiveness of the fuzzy controller to control mobile robot.

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่ออังกฤษ ..... ข
สารบัญ ..... ค
สารบัญตาราง ..... จ
สารบัญรูป ..... ฉ

บทที่1 บทนำ
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 2
1.3 ขอบข่ายของโครงงาน 2
1.4 ขั้นตอนของการดำเนินงาน 2
1.5 แผนการดำเนินงาน 3
1.6 ผลที่คาดว่าจะได้รับ 4
1.7 งบประมาณของโครงงาน 4

บทที่2 ทฤษฎีพื้นฐานของฟัซซี่ลอจิก
2.1 ระบบฟัซซี่ 5
2.2 การทำงานของฟัซซี่ลอจิก 10
2.3 กฎของฟัซซี่ 11
2.4 กระบวนการทำงานของฟัซซี่ 11
2.5 กฎของฟัซซี่ซิงเกิลตัน 15

บทที่3 การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกิลตัน
3.1 การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกิลตัน 17
3.2 การออกแบบและการเขียนโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์ 18
3.3 การออกแบบและการเขียนโปรแกรมของระบบควบคุม
การตัดสินใจในการหลบหลีกสิ่งกีดขวางของหุ่นยนต์โดยใช้ทฤษฎีฟัซซี่ 22

สารบัญ (ต่อ)

บทที่4 ผลการทดลอง

4.1 ผลการทดลองการหลบหลีกสิ่งกีดขวางของหุ่นยนต์ 31

4.2 ผลการทดลองการเคลื่อนที่เข้าหาจุดเป้าหมายของหุ่นยนต์..... 32

บทที่5 สรุปผล 38

ภาคผนวก 1 39

ภาคผนวก 2 50

มหาวิทยาลัยนเรศวร

สารบัญตาราง

ตารางที่ หน้า

3.1 ตารางแสดงกฎของฟัซซี่ 25

สารบัญรูป

รูปที่ หน้า

2.1 แสดงการอ่านค่าฟังก์ชั่นจำนวนสมาชิก 6

2.2 ฟังก์ชันสมาชิกฟัซซี่ที่มีขอบเขตฟัซซี่เป็นฟังก์ชั่นรูปสามเหลี่ยม 7

2.3 เซตของฟัซซี่ลอจิกกับขอบเขตร่วม 7

2.4 ค่าภาวะสมาชิกฟัซซี่ที่แสดงโดยค่าเฉพาะ 8

2.5 การใช้โอเปอเรเตอร์ของฟัซซี่ลอจิก 9

2.6 เซตทั่วไปและเซตของฟัซซี่ที่แสดงความสูง 10

2.7 เซตทั่วไปและเซตของฟัซซี่ที่แสดงความสูงและความต่ำ 11

2.8 อินพุตของฟัซซี่ 12

2.9 การประยุกต์ใช้โอเปอเรเตอร์ 13

2.10 การประยุกต์ใช้วิธีการส่อความ 13

2.11 การรวมเอาต์พุต 14

2.12 การแปลงกลับของฟัซซี่ 15

2.13 ฟังก์ชันภาวะสมาชิกของฟัซซี่ที่มีขอบเขตเป็นฟังก์ชันของรูปสามเหลี่ยม 15

3.1 แผนภาพแสดงการทำงานของโปรแกรมการควบคุมหุ่นยนต์เคลื่อนที่ 17

3.2 แผนภาพการทำงานของโปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์
โดยฟัซซี่ซิงเกิลตัน 18

3.3 Form1ของโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์ 19

3.4 สร้างพื้นที่ในการแสดงสิ่งกีดขวาง 19

3.5 ขั้นตอนที่3ของโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์ 20

3.6 การวางรูปแบบการรับค่าต่างๆ 20

3.7 สร้าง CommandBotton 21

3.8 จัดแต่ง Form ของโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์ 21

3.9 การสร้างส่วนของการทำงานเพื่อให้มีการทำงานเป็นรอบ 22

3.10 ช่วงขององศาที่มีผลและไม่มีผลต่อการตัดสินใจหลบหลีกสิ่งกีดขวางของหุ่นยนต์ 22

3.11 การแบ่งขอบเขตของอินพุต 23

3.12 ระยะที่น้อยที่สุดที่หุ่นยนต์จะตัดสินใจว่าควรจะหลบสิ่งกีดขวาง 24

3.13 กราฟของตัวแปรอินพุต 1 24

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.14 รูปแสดงตำแหน่งของสิ่งกีดขวาง 26

3.15 Form1ของโปรแกรมส่วนฟัซซี่ 27

3.16 ขั้นตอนที่ 2ของโปรแกรมส่วนฟัซซี่ 28

3.17 แสดงผลของโปรแกรมส่วนฟัซซี่ 28

3.18 ตกแต่ง Formของโปรแกรมส่วนฟัซซี่ 30

4.1 หุ่นยนต์จะเคลื่อนที่เพียงเพื่อหลบแต่วัตถุ โดยไม่สนใจทิศทางที่กำลังเคลื่อนที่ไป 32

4.2 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่1 34

4.3 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่2 35

4.4 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่3 36

4.5 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่4 37

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ประเทศไทยจัดได้ว่าเป็นประเทศเกษตรกรรม แต่ปัจจุบันอุตสาหกรรมได้เข้ามามีบทบาทต่อเศรษฐกิจของประเทศไทย เนื่องจากรายได้ที่มีมูลค่ามากนั้นมาจากภาคอุตสาหกรรม ทั้งนี้เพราะการนำผลผลิตทางการเกษตรมาเป็นทุนทางวัตถุดิบในภาคอุตสาหกรรมเริ่มแพร่หลาย ในด้านการผลิตหุ่นยนต์และเครื่องจักรกลเป็นเทคโนโลยีที่ถูกทำมาใช้เพื่อทดแทนกำลังคนเพื่อลดต้นทุนในการจ้างแรงงานคนและเพิ่มประสิทธิภาพในการผลิต

นอกจากนี้แล้วหุ่นยนต์หรือเครื่องจักรกลยังไม่ได้ถูกใช้ในภาคอุตสาหกรรมเพียงอย่างเดียวเท่านั้น รัฐบาลและองค์กรต่างๆเล็งเห็นความสำคัญถึงการพัฒนาเทคโนโลยีเหล่านี้ซึ่งจะมีส่วนช่วยในการพัฒนาประเทศชาติและองค์กรดังจะเห็นได้จากการจัดการแข่งขันหุ่นยนต์ซึ่งถูกจัดขึ้นมากมายในแต่ละปีมหาวิทยาลัยรัฐบาลและเอกชนมีการจัดสรรงบประมาณเพื่อสนับสนุนงานวิจัยทางด้านนี้

การควบคุมหุ่นยนต์หรือเครื่องจักรกลนั้นทำได้อยู่หลายวิธีตามแต่บริษัทผู้ผลิตจะกำหนด ไม่ว่าจะเป็นระบบไมโครคอนโทรลเลอร์ (microcontroller system)หรือระบบควบคุมแบบอื่นๆ ซึ่งก็มีให้เลือกใช้กันตามความเหมาะสม ในการตัดสินใจเลือกใช้ระบบควบคุมมักมีองค์ประกอบหลายๆด้านที่ต้องคำนึงถึงทั้งนี้เพื่อความสะดวกที่จะใช้ควบคุม ความทันสมัยตลอดจนเพื่อความเหมาะสมกับงานนั้นๆ อีกทั้งยังขึ้นกับความถนัดของผู้ออกแบบเองเช่นกัน

ระบบควบคุมโดยฟัซซี่ลอจิกเป็นอีกระบบควบคุมหนึ่งที่ถูกนำมาใช้เนื่องด้วยความสามารถเฉพาะตัวและคุณสมบัติของฟัซซี่ลอจิกเองที่มีความสามารถในการตัดสินใจได้ดีใกล้เคียงกับการตัดสินใจของมนุษย์ โดยฟัซซี่ลอจิกสามารถรวมระบบแนวคิดของสองระบบที่แตกต่างกันเข้าด้วยกันได้ เช่น ระบบสมองกับระบบคอมพิวเตอร์ โดยจะเห็นว่าสมองนั้นเป็นระบบการทำงานที่มีปัจจัยหรือตัวแปรค่อนข้างสูงอีกทั้งยังมีความไม่แน่นอนสูงด้วย แต่คอมพิวเตอร์เป็นระบบการทำงานที่มีกฎเกณฑ์ชัดเจน เช่น การคำนวณเป็นต้น ด้วยเหตุผลนี้จึงสามารถนำฟัซซี่ลอจิกจึงถูกนำไปประยุกต์ใช้ในศาสตร์หลายๆแขนงเพื่อควบคุมการทำงานของระบบได้

จากคุณสมบัติของฟัซซี่ลอจิกที่มีความสามารถในการตัดสินใจได้ดีใกล้เคียงกับการตัดสินใจของมนุษย์ ผู้ทำโครงงานจึงสนใจที่จะนำศาสตร์ของฟัซซี่ลอจิกมาใช้เพื่อควบคุมการจำลองการเคลื่อนไหวของหุ่นยนต์ โดยจะควบคุมให้มีการตัดสินใจเพื่อหลบหลีกสิ่งกีดขวางโดยไม่ต้องมีมนุษย์เป็นผู้ควบคุม นั่นคือหุ่นยนต์ในแบบจำลองจะต้องทำการตัดสินใจเพื่อหลบหลีกสิ่งกีดขวางที่เกิดขึ้นได้โดยอาศัยระบบควบคุมแบบฟัซซี่

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาหลักการทำงานของฟัซซี่ลอจิกในส่วนที่เกี่ยวข้องกับการควบคุมการตัดสินใจของหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวาง

1.2.2 เพื่อสร้างระบบควบคุมการตัดสินใจของหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวางโดยทฤษฎีของฟัซซี่ลอจิก

1.2.3 สามารถแสดงผลเพื่อให้เห็นการตัดสินใจของหุ่นยนต์ในการหลบหลีกสิ่งกีดขวางโดยใช้การจำลองการเคลื่อนที่ของหุ่นยนต์

1.2.4 เพื่อศึกษาการเขียนโปรแกรมโดยใช้ภาษาวิชวลเบสิค (Visual Basic) ในส่วนที่เกี่ยวข้องกับโครงงาน

1.3 ขอบข่ายของโครงงาน

1.3.1 สร้างโปรแกรมการจำลองการทำงานของหุ่นยนต์ในการหลบหลีกสิ่งกีดขวางและเคลื่อนที่เข้าหาเป้าหมาย

1.3.2 การประยุกต์ใช้ฟัซซี่ซิงเกิลตันเพื่อควบคุมการหลบหลีกสิ่งกีดขวางของหุ่นยนต์เคลื่อนที่

1.4 ขั้นตอนของการดำเนินงาน

1.4.1 ศึกษาและค้นคว้าข้อมูลเกี่ยวกับฟัซซี่ลอจิกในส่วนเกี่ยวกับการควบคุมการตัดสินใจของหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวาง

1.4.2 ศึกษาและค้นคว้าเกี่ยวกับโปรแกรมที่นำมาออกแบบการจำลองการเคลื่อนที่ของหุ่นยนต์

1.4.3 เขียนโปรแกรมเพื่อจำลองการเคลื่อนที่ของหุ่นยนต์

1.4.4 เขียนโปรแกรมในส่วนของฟัซซี่ลอจิกเพื่อควบคุมการตัดสินใจของหุ่นยนต์ในการหลบหลีกสิ่งกีดขวาง

1.4.5 ทดสอบการทำงาน

1.4.6 สรุปผลการทดลองและจัดทำรูปเล่มโครงงาน

1.5 แผนการดำเนินงาน

กิจกรรม	ปี2546		ปี2547									
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
1. ศึกษาและค้นคว้าข้อมูลเกี่ยวกับทฤษฎีฟัซซี่ในส่วนที่เกี่ยวข้องกับการควบคุมการตัดสินใจของหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวาง	←	—	—	→								
2. ค้นคว้าและศึกษาเกี่ยวกับโปรแกรมที่จะนำมาออกแบบการจำลองการเคลื่อนที่ของหุ่นยนต์					←	—	→					
3. เขียนโปรแกรมเพื่อจำลองการเคลื่อนที่ของหุ่นยนต์							←	—	→			
4. เขียนโปรแกรมในส่วนของฟัซซี่ลอจิกเพื่อควบคุมการตัดสินใจของหุ่นยนต์ในการหลบหลีกสิ่งกีดขวาง									←	↔		
5. ทดสอบการทำงาน										←	↔	
6. สรุปผลการทดลองและจัดทำรูปเล่มโครงงาน											←	→

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถเข้าใจหลักการของฟัซซี่ลอจิกและนำไปประยุกต์ใช้ในส่วนของการควบคุมการตัดสินใจของหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวาง

1.6.2 สามารถใช้ฟัซซี่ลอจิกควบคุมการตัดสินใจของหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวางได้

1.6.3 สามารถแสดงผลการจำลองการเคลื่อนที่ของหุ่นยนต์เพื่อแสดงให้เห็นการตัดสินใจของหุ่นยนต์ในการหลบหลีกสิ่งกีดขวางได้

1.7 งบประมาณของโครงงาน

1.7.1 ค่าถ่ายเอกสารและค่าเข้าเล่มรายงาน

1.7.2 ค่าแผ่นซีดี

1.7.3 ค่าหนังสือข้อมูลเกี่ยวกับฟัซซี่ลอจิก

1.7.4 ค่าหมึกพิมพ์

รวมเป็นเงิน 3,000 บาท(สามพันบาทถ้วน)

บทที่ 2
ทฤษฎีพื้นฐานของฟัซซี่ลอจิก

ฟัซซี่ลอจิก เป็นวิธีการทางคณิตศาสตร์แบบตรรกศาสตร์ สามารถเชื่อมโยงระบบการคำนวณสองระบบเข้าด้วยกัน โดยที่ระบบหนึ่งเป็นผลการคำนวณหรือมีวิธีการคำนวณไม่เป็นไปตามกฎทางคณิตศาสตร์กฎใดกฎหนึ่งเพียงอย่างเดียว ส่วนอีกระบบหนึ่งเป็นผลการคำนวณหรือมีวิธีการคำนวณเป็นไปตามกฎทางคณิตศาสตร์กฎใดกฎหนึ่งชัดเจน

ยังมีคำอธิบายความหมายอีกอย่างหนึ่งของฟัซซี่ลอจิก โดยเน้นระบบที่เหมาะสมสำหรับฟัซซี่ลอจิกคือ ตรรกะหรือวิธีที่จะใช้จัดการกับระบบที่มีความแปรปรวนของกระบวนการสูง และผลที่ออกมาก็มีความไม่แน่นอนสูง ตัวอย่างที่ดีที่สุดสำหรับฟัซซี่ลอจิกคือสมองมนุษย์นี้เอง ที่มีกระบวนการทำงานออกมาเป็นความคิดซับซ้อนและไม่แน่นอน และไม่ขึ้นอยู่กับปัจจัยอย่างใดอย่างหนึ่งเพียงอย่างเดียว

ฟัซซี่ลอจิกถูกคิดค้นโดย Dr.Lotfi Zadeh ในปี ค.ศ.1965 ซึ่งถือว่าเป็นสาขาหนึ่งของวิชาคณิตศาสตร์ โดยการใช้คอมพิวเตอร์เข้ามาจำลองแบบอย่างที่คนส่วนใหญ่มักตัดสินใจกัน ซึ่งไม่เหมือนกับคอมพิวเตอร์โดยทั่วไป เนื่องจากการตัดสินใจของมนุษย์จะไม่มีความไม่เที่ยงตรงโดยใช้ความคิดและเหตุผลเข้ามามีส่วน แสดงออกมาในรูปของภาษา เช่น "ร้อน" หรือ "เร็ว" มากกว่าที่จะแสดงคำพูดที่มีความเที่ยงตรงทางตัวเลขแทน เช่น ร้อนที่อุณหภูมิ 100 องศาหรือที่ความเร็ว 70 ไมล์ต่อชั่วโมง เป็นเรื่องปกติที่มนุษย์มักมีการตัดสินใจแบบครึ่งๆกลางๆ (Shades of gray) ไม่ใช่การตัดสินใจที่เจาะจงลงไปได้ว่า "ใช่" หรือ "ไม่ใช่" ด้วยระบบฟัซซี่ลอจิกนี้เองสามารถทำให้การคำนวณของคอมพิวเตอร์มีระบบการตัดสินใจแบบที่มนุษย์ตัดสินใจ

ฟัซซี่ลอจิกกับความน่าจะเป็นนั้นมีความแตกต่างกัน เนื่องจากว่าความน่าจะเป็น เป็นการคาดการเหตุการณ์ที่จะเกิดขึ้นในอนาคต ส่วนฟัซซี่ลอจิกเป็นการบอกถึงเหตุการณ์ที่เกิดขึ้นไปแล้ว

2.1 ระบบฟัซซี่ (Fuzzy System)

ระบบฟัซซี่ประกอบด้วยส่วนต่างๆดังนี้

2.1.1 เซตของ ฟัซซี่ (Fuzzy sets)

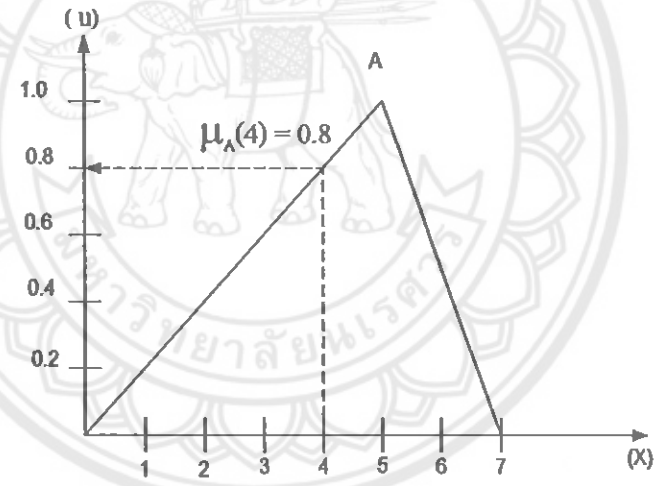
เซตของฟัซซี่คล้ายๆกับเซตในทฤษฎีเซต ความแตกต่างระหว่างเซตของฟัซซี่กับทฤษฎีเซตอยู่ที่การบวกของค่าจำนวนสมาชิก ซึ่งค่าจำนวนสมาชิกจะอยู่ในช่วงเวลาระหว่าง [0, 1]

ถ้าให้ U เป็นเซตเอกภพและ 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 ค่านี้อาจเรียกว่าค่าภาวะสมาชิก (degree of membership)



รูปที่2.1 แสดงการอ่านค่าฟังก์ชั่นจำนวนสมาชิก

ตัวอย่างเช่น $\mu_A(x)$ หมายถึง ฟังก์ชั่นภาวะสมาชิกของ X ในเซต A

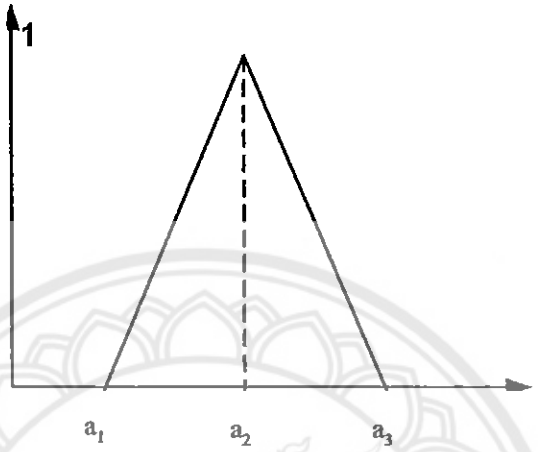
$\mu_A(4)$ หมายถึง ฟังก์ชันภาวะสมาชิกของอินพุต $X = 4$ ในเซต A เท่ากับ 0.8

2.1.3 ขอบเขตฟัซซี่ (Fuzzy regions)

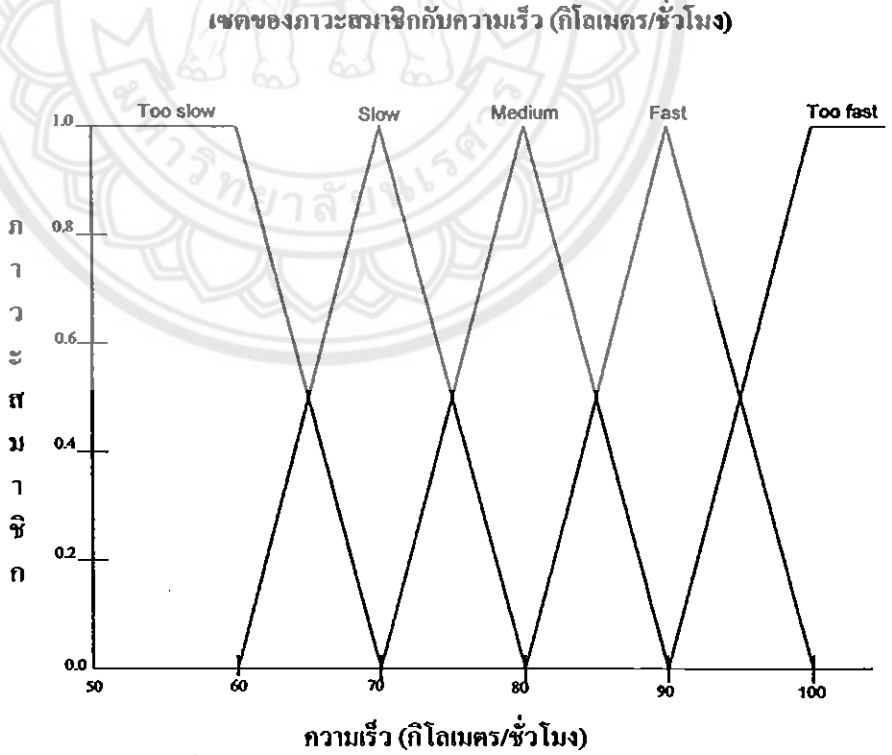
ขอบเขตฟัซซี่ใช้แสดงถึงความสัมพันธ์ระหว่างค่าภาวะสมาชิกกับฟัซซี่เซต ขอบเขตฟัซซี่ไม่จำเป็นต้องมีการเหลื่อมล้ำหรือซ้อนทับกันแต่จะต้องสอดคล้องกับกฎของฟัซซี่

รูปแบบของขอบเขตฟัซซี่สามารถนิยามได้โดยฟังก์ชันภาวะสมาชิก ถึงแม้ว่ารูปของขอบเขตฟัซซี่สามารถใช้ฟังก์ชันที่มีความซับซ้อน เช่น ฟังก์ชั่นระฆังคว่ำ ฟังก์ชั่นสามเหลี่ยม ฯลฯ แต่ที่นิยมที่สุดสำหรับขอบเขตฟัซซี่คือฟังก์ชันรูปสามเหลี่ยม เนื่องจากเป็นฟังก์ชันที่ไม่ซับซ้อน

$$\mu A(x) = \begin{cases} (x-a_1)/(a_2-a_1) & \text{ถ้า } a_1 \le x < a_2 \\ (a_3-x)/(a_3-a_2) & \text{ถ้า } a_2 \le x < a_3 \\ 0 & \text{ถ้า } x \le a_1 \text{ หรือ } x > a_3 \end{cases} \quad (2.2)$$



รูปที่2.2 ฟังก์ชันสมาชิกฟัซซี่ที่มีขอบเขตฟัซซี่เป็นฟังก์ชันรูปสามเหลี่ยม



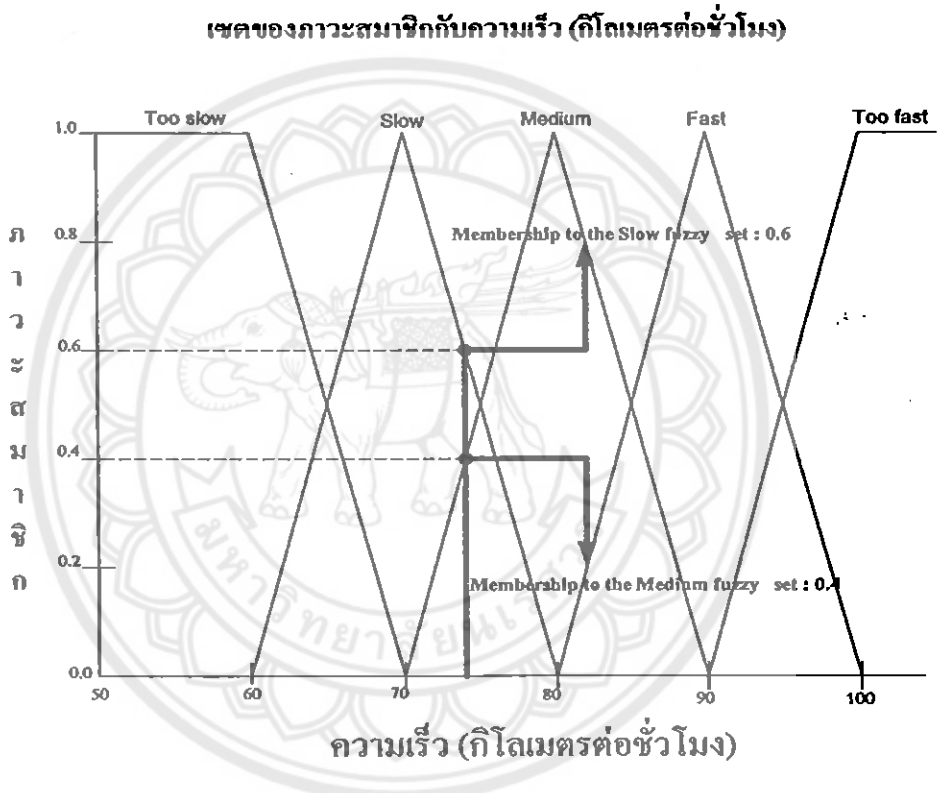
รูปที่ 2.3 เซตของฟัซซี่ลอจิกกับขอบเขตร่วม

ถ้าพิจารณาที่ความเร็ว 74 km/h จะได้ค่าภาวะสมาชิกของความเร็วระดับต่างนั่นมีค่าดังต่อไปนี้

μ slow (74) หมายถึงที่ความเร็ว 74 km/h ค่าภาวะสมาชิกของความเร็วระดับช้า(Slow)ของเซตความเร็วมีค่าเป็น 0.6

μ medium (74) หมายถึงที่ความเร็ว 74 km/h ค่าภาวะสมาชิกของความเร็วระดับกลาง(Medium)ของเซตความเร็วมีค่าเป็น 0.4

μ too slow (74) = μ fast (74) =μ too fast (74) หมายถึงที่ความเร็ว 74 km/h ค่าภาวะสมาชิกของความเร็วระดับช้ามาก(Too Slow), ระดับเร็ว(fast)และระดับเร็วที่สุด(Too Fast)มีค่าเป็นศูนย์



รูปที่ 2.4 ค่าภาวะสมาชิกฟัซซี่ที่แสดงโดยค่าเฉพาะ

2.1.4 ตัวแปรเชิงภาษา (linguistic variable)

ตัวแปรเชิงภาษาคือตัวแปรที่เป็นคำพูดในภาษามนุษย์ ในทางฟัซซี่ลอจิกจะระบุค่าที่เป็นคำพูดของตัวแปรเชิงภาษาด้วยเซตของฟัซซี่

ตัวอย่างเช่นความสูงของมนุษย์ ถ้าให้ U = [0,300] (cm) เป็นเอกภพความสูงมีค่าตั้งแต่ 0 ถึง 300 เซนติเมตร ค่าที่เป็นไปได้สำหรับตัวแปรเชิงภาษาได้แก่

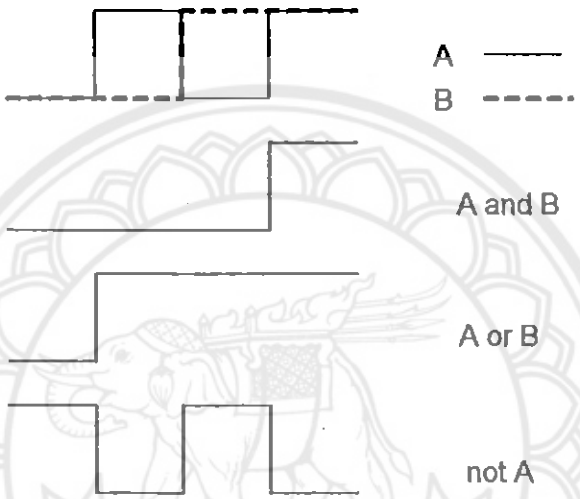
T = tall + very tall + not tall + small + very small + normal + rather small ...

เซตของฟัซซี่แต่ละเซตมีการใช้ค่าภาวะสมาชิก(membership value)ที่เป็นฟังก์ชันร่วมกัน โดยระบุเป็นตัวเลขที่แน่นอนของเซตนั้นๆเช่น ถ้าเราให้ "ส่วนสูง" เป็นตัวแปรแล้วใช้เงื่อนไขในการหา

ค่า "ส่วนสูง" (Height) ในเซตของ "สูง" ยกตัวอย่างเช่น ถ้าส่วนสูงมีค่าเป็น 6 ฟุต นั่นคือ "ส่วนสูง" เท่ากับ "สูง" (Height is TALL) ซึ่งประโยค "ส่วนสูง" เท่ากับเซต "สูง" นั้นเราเรียกว่า Antecedent

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

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

2.1.6 ค่าเฉลี่ยและค่ากลางของฟัซซี่

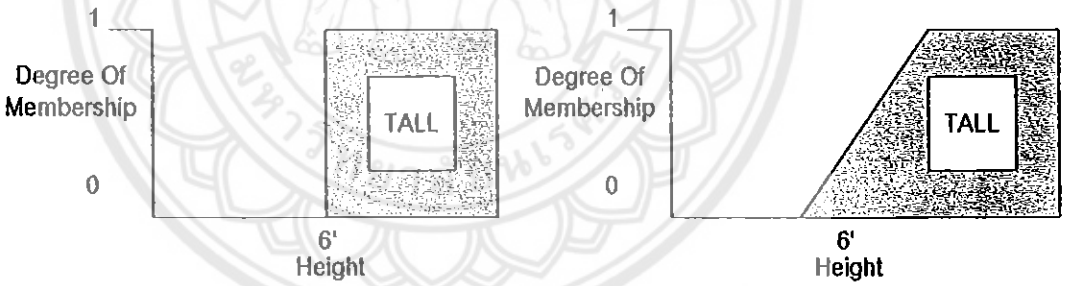
ตำแหน่งของเซตของฟัซซี่สามารถอธิบายได้โดยค่าเฉลี่ยของฟัซซี่ (fuzzy mean) หรือ M บางครั้งอาจเรียกว่าจุดศูนย์ถ่วง(Center of gravity) หรือ centroid ถ้าค่ากลางของฟัซซี่ของเซตของฟัซซี่ A ถูกนิยามบนเส้นจำนวนจริง M(A) สามารถหาได้โดยตรงจาก

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

2.2 การทำงานของฟัซซี่ลอจิก

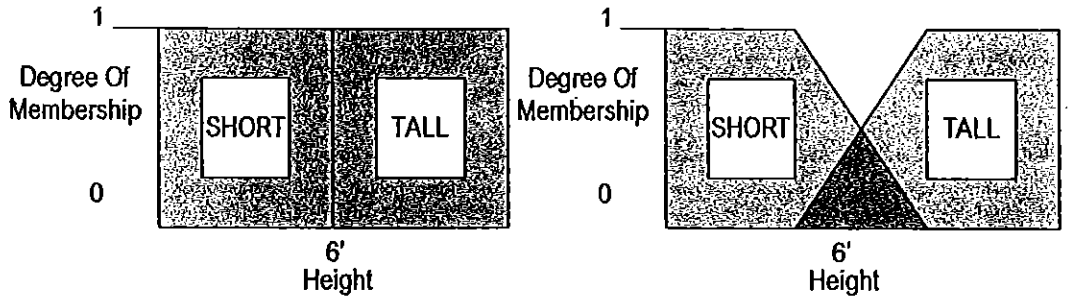
ฟัซซี่ลอจิกทำงานโดยใช้สับเซตของฟัซซี่ซึ่งไม่เหมือนกับเซตโดยทั่วไปที่ประกอบไปด้วยสมาชิกที่มีค่าตามเงื่อนไขที่ถูกกำหนดเท่านั้น หรืออาจกล่าวได้ว่าเซตนี้เมื่อมีสมาชิกแล้วจะมีค่าเป็น " 1 " และถ้าเซตนี้ไม่มีสมาชิกเลยจะมีค่าเป็น " 0 " ยกตัวอย่างเช่น ถ้าพูดถึงเซต " สูง " (TALL) ของผู้ชาย ให้หมายความว่าเซตนี้ประกอบไปด้วยผู้ชายที่ความสูงตั้งแต่ 6 ฟุตขึ้นไป ซึ่งเราจะบอกได้ว่าผู้ชายที่มีความสูง 6 ฟุตนั้น " สูง " และถ้าสูง 5 ฟุต 11 นิ้ว ก็จัดได้ว่า " ไม่สูง " ซึ่งในความเป็นจริงแล้วมันอาจจะไม่เป็นเช่นนั้นเสมอไป เพราะคนส่วนใหญ่ยังถือว่าผู้ชายที่มีความสูง 5 ฟุต 11 นิ้วนั้นจัดว่า " สูง " แต่ถึงอย่างนั้นก็ตามเซตโดยทั่วไปจะบอกได้แค่เพียงว่า " สูง " หรือ " ไม่สูง " ไม่มีกลุ่มที่อยู่ระหว่างนั้น

เมื่อเทียบกันแล้วเซตของฟัซซี่จะมีเงื่อนไขที่ยืดหยุ่นมากกว่า ซึ่งค่าของสมาชิกจะอยู่ในช่วง 0 ถึง 1 ไม่ใช่มีเพียงแค่ 0 หรือ 1 เหมือนในกรณีของเซตโดยทั่วไป โดยเซตของฟัซซี่จะทำให้เกิดการลาดเอียงของส่วนที่เป็นสมาชิกไปยังส่วนที่ไม่ใช่สมาชิก เช่น ผู้ชายที่สูง 6 ฟุตจะมีค่าในเซต " ความสูง " ของเซตของฟัซซี่เป็น 0.5 ผู้ชายที่สูง 5 ฟุต 6 นิ้วจะมีค่าในเซต " ความสูง " เป็น 0.25 และผู้ชายที่สูง 6 ฟุต 6 นิ้วจะมีค่าในเซต " ความสูง " เป็น 0.75 ดังรูปที่ 2.6



รูปที่ 2.6 เซตทั่วไปและเซตของฟัซซี่ที่แสดงความสูง

รูปที่ 2.6 แสดงให้เห็นถึงเซตทั่วไปกับฟัซซี่ในเซตของ " ต่ำ " กับ " สูง " โดย เซตทั่วไปจะกำหนดค่าระหว่าง " ต่ำ " กับ " สูง " ออกจากกันโดยใช้ความสูงที่ 6 ฟุตเป็นตัวตัดสิน แต่เมื่อเปรียบกับเซต " ต่ำ " กับ " สูง " ของฟัซซี่แล้ว ซึ่งจะแบ่งระดับระหว่าง " ต่ำ " กับ " สูง " ออกเป็นความสูงระหว่าง 5 ถึง 7 ฟุต



รูปที่ 2.7 เซตทั่วไปและเซตของฟัซซี่ที่แสดงความสูงและความต่ำ

แสดงให้เห็นถึงความเข้าใจเกี่ยวกับระดับขั้นของความสูงจึงขึ้นอยู่กับความคิดและประสบการณ์ของแต่ละบุคคลที่จะกำหนดวิธีที่ง่ายที่สุดในการออกแบบฟัซซี่เซต

2.3 กฎของฟัซซี่ (Fuzzy Rules)

กฎถูกออกแบบเพื่ออธิบายและเพื่อควบคุมการทำงานของฟัซซี่ อย่างไรก็ตามในบางครั้งเป็นการยากที่จะวางกฎเกณฑ์ในรูปแบบที่เที่ยงตรงเพราะการออกแบบกฎมาจากประสบการณ์ของมนุษย์

กฎฟัซซี่ประกอบด้วยข้อสนับสนุน $A_{i,k}$ ในรูปแบบของเซตของฟัซซี่ที่มี $\mu A_{i,k}$ เป็นฟังก์ชั่นภาวะสมาชิกส่วนนี้เรียก 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 $A_{i,k}$" อาจเขียนซ้ำๆด้วย $A_{i,k}$ เพื่อความง่ายก็ได้

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

2.4 กระบวนการทำงานของฟัซซี่ลอจิก

1. การทำให้เป็นฟัซซี่(Fuzzification) หน้าที่ของสมาชิกในฟังก์ชันจะทำหน้าที่เป็นตัวแปรอินพุตและถูกนำมาใช้เป็นค่าจริง เพื่อนำมาใช้หาค่าความจริงของแต่ละหลักเกณฑ์

2. การอนุมาน(Inference) เป็นการสร้างกฎฟัซซี่

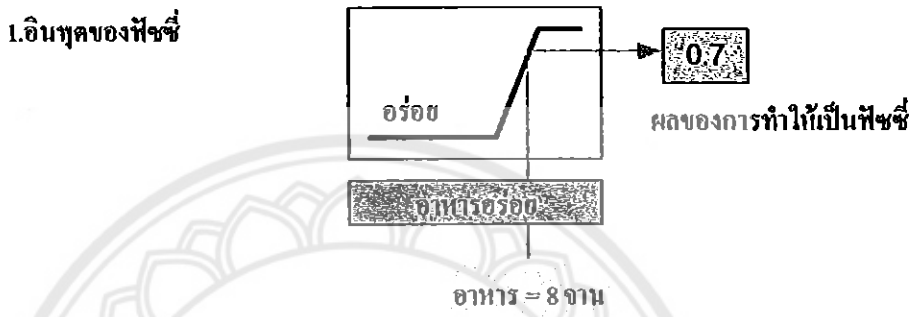
3. การแปลงกลับของฟัซซี่ (Defuzzification) หรือแปลงเอาต์พุตของฟัซซี่เป็นเอนาลอก การแปลงกลับของฟัซซี่มีด้วยกันอยู่หลายวิธี ทั่วไปที่ใช้กันมากกว่าวิธีอื่นคือ Centroid

สามารถจำลองการทำงานของกระบวนการต่างๆของฟัซซี่ลอจิกได้ดังนี้

ในที่นี้จะกล่าวถึงค่าตอบแทนอื่นๆที่พนักงานได้รับจากลูกค้าในร้านอาหารแห่งหนึ่ง

ขั้นที่1 อินพุตของฟัซซี่

ในขั้นแรกจะต้องกำหนดตัวแปรอินพุตและหาค่าของ สำหรับตัวอย่างจะสร้างฟังก์ชั่นภาวะสมาชิกขึ้นมา 3 ฟังก์ชั่น นั่นคืออินพุตจะมีฟังก์ชั่นภาวะสมาชิก 3 ฟังก์ชั่น แต่ละฟังก์ชั่นขึ้นอยู่กับการกระจายของอินพุตที่แตกต่างกันตามเซตของตัวแปรเชิงภาษา เช่น การบริการไม่ดี, การบริการดี, อาหารไม่อร่อย, อาหารอร่อย เป็นต้น

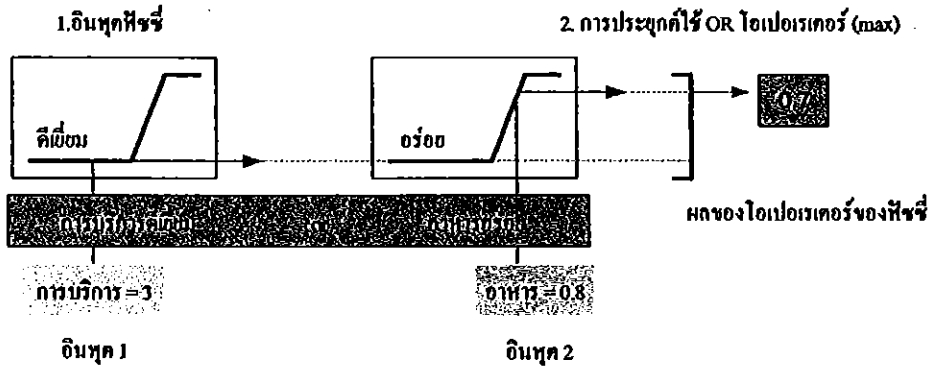


รูปที่2.8 อินพุตของฟัซซี่

รูปที่2.8 จะแสดงค่าของฟังก์ชั่นภาวะสมาชิกความอร่อยของอาหารว่าอยู่ในระดับไหนซึ่งจะขึ้นอยู่กับจำนวนอาหารที่ทานไป โดยสมมุติค่าของฟังก์ชั่นภาวะสมาชิกความอร่อยของอาหารมีค่าระหว่าง 0 ถึง 1 ตัวแปรเชิงภาษาในที่นี้คือ อร่อย กรณีนี้จากอินพุตคืออาหาร 8 จานจะได้ว่า μอร่อย = 0.7 สำหรับฟังก์ชั่นภาวะสมาชิกของความอร่อย

ขั้นที่ 2 การประยุกต์ใช้โอเปอเรเตอร์

หลังจากที่อินพุตถูกทำให้เป็นฟัซซี่แล้ว จะทราบระดับของแต่ละส่วนของ Antecedent ที่จะใช้สำหรับกฎแต่ละกฎ ถ้า Antecedent ของกฎมีมากกว่าหนึ่งแล้ว Antecedent โอเปอเรเตอร์ของฟัซซี่จะถูกนำมาใช้ โดยค่าที่ได้นี้จะถูกนำไปประยุกต์เพื่อหาฟังก์ชันเอาต์พุตต่อไป เพราะถึงแม้จะมีอินพุตของฟัซซี่ถึง 2 ค่า (2 Antecedents)หรือมากกว่า แต่เอาต์พุตของกฎหนึ่งกฎจะต้องมีเพียงค่าความจริงเดียวเท่านั้น

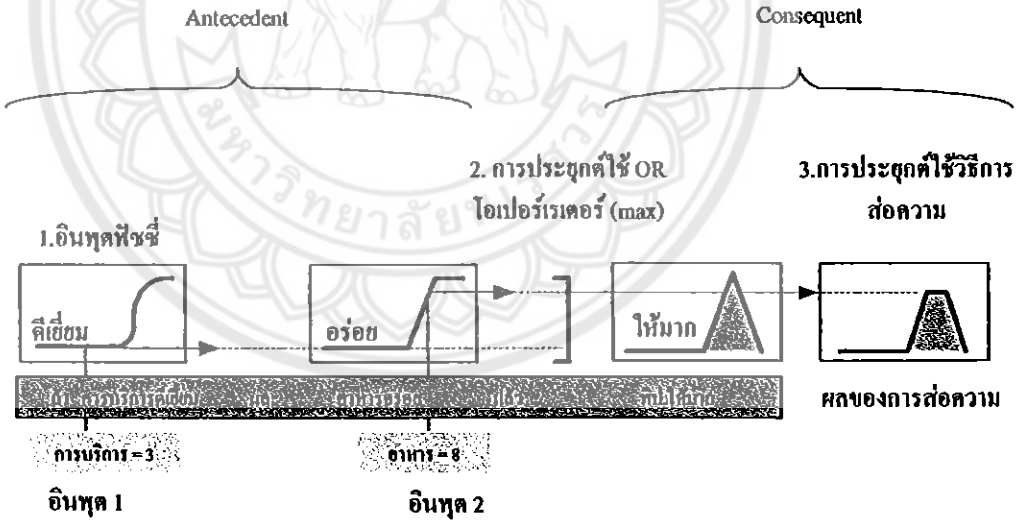


รูปที่2.9 การประยุกต์ใช้โอเปอเรเตอร์

รูปที่ 2.9 แสดงภาพการทำงานของ OR โอเปอเรเตอร์ ซึ่งการโอเปอเรเตอร์ของฟัซซี่ได้กล่าวถึงในหัวข้อที่ 2.1.5 แล้ว

ขั้นที่3 การประยุกต์ใช้วิธีการส่อความ(Implication Method)

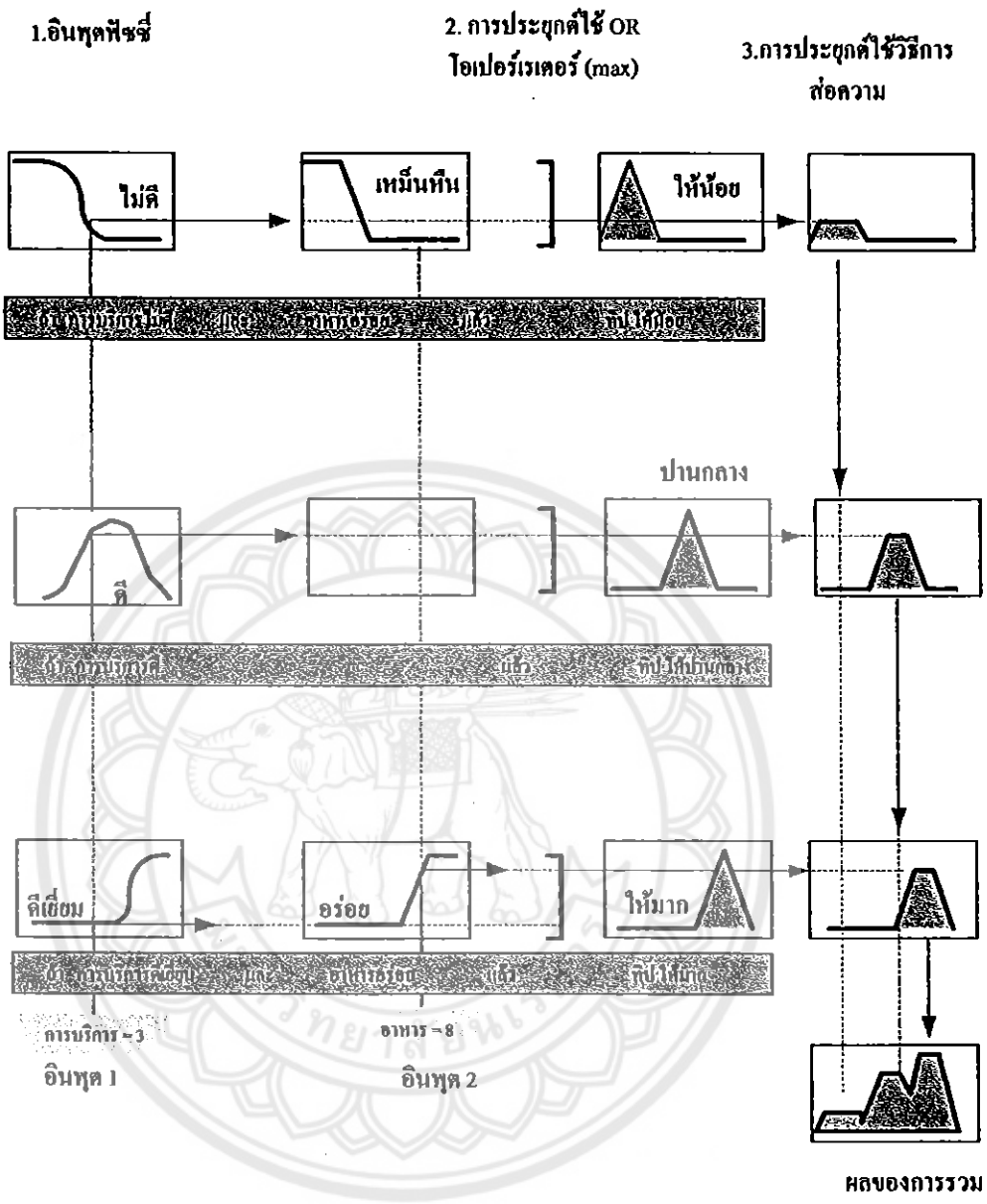
เป็นการหาเอาต์พุตของแต่ละกฎโดยจะแสดงได้ดังรูปที่2.10



รูปที่2.10 การประยุกต์ใช้วิธีการส่อความ

ขั้นที่ 4 การรวมเอาต์พุตทั้งหมด

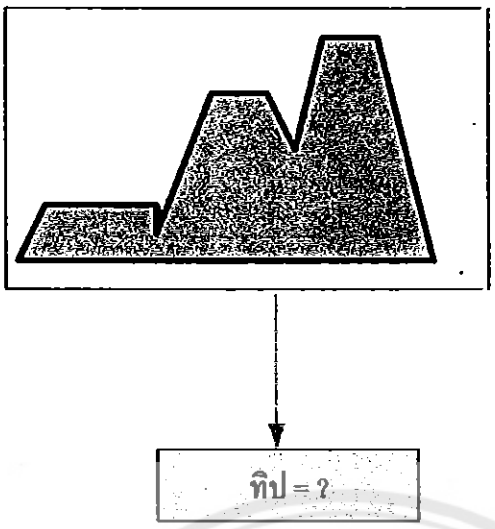
หลังจากที่ได้เอาต์พุตของแต่ละกฎออกมาแล้ว จะต้องทำการรวมเอาต์พุตเพื่อให้ได้เอาต์พุตเดียวเพื่อนำไปใช้ในการคำนวณสำหรับการแปลงกลับ รูปที่2.11 จะแสดงการรวมกันของเอาต์พุตของกฎ 3 กฎที่สมมติขึ้น โดยสิ่งที่ได้คือเซตของฟัซซี่เพียงหนึ่งเซต



รูปที่2.11 การรวมเอาต์พุต

ขั้นที่ 5 การแปลงกลับของฟัซซี่

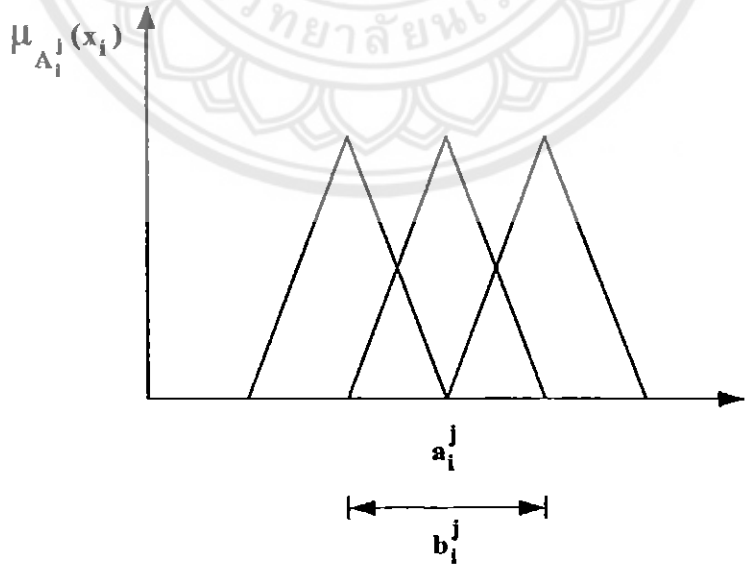
อินพุตสำหรับการแปลงกลับก็คือเอาต์พุตของการรวมกันในขั้นที่ 4 การแปลงกลับมีด้วยกันอยู่หลายวิธีดังที่ได้กล่าวไปเมื่อข้างต้นแล้ว โดยทั่วไปจะเลือกใช้วิธีการ centroid ซึ่งต้องมีการคำนวณพื้นที่ใต้กราฟ



รูปที่2.12 การแปลงกลับของฟัซซี่

2.5 กฎของฟัซซี่ซิงเกิลตัน (fuzzy singleton rule)

กฎของฟัซซี่ซิงเกิลตันเป็นอีกวิธีการหนึ่งที่ใช้หาเอาต์พุตเมื่อฟังก์ชันภาวะสมาชิกมีขอบเขตเป็นฟังก์ชันสมมาตรเช่นฟังก์ชันของรูปสามเหลี่ยม, ฟังก์ชันของรูปสี่เหลี่ยมคางหมูหรือฟังก์ชันของรูประฆังเป็นต้น เพื่อความง่ายต่อการคำนวณจะแสดงโดยใช้ฟังก์ชันของรูปสามเหลี่ยมดังนี้



รูปที่2.13 ฟังก์ชันภาวะสมาชิกของฟัซซี่ที่มีขอบเขตเป็นฟังก์ชันของรูปสามเหลี่ยม

สามารถเขียนเป็นสมการได้ดังนี้

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

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

กฎ j : *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.

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

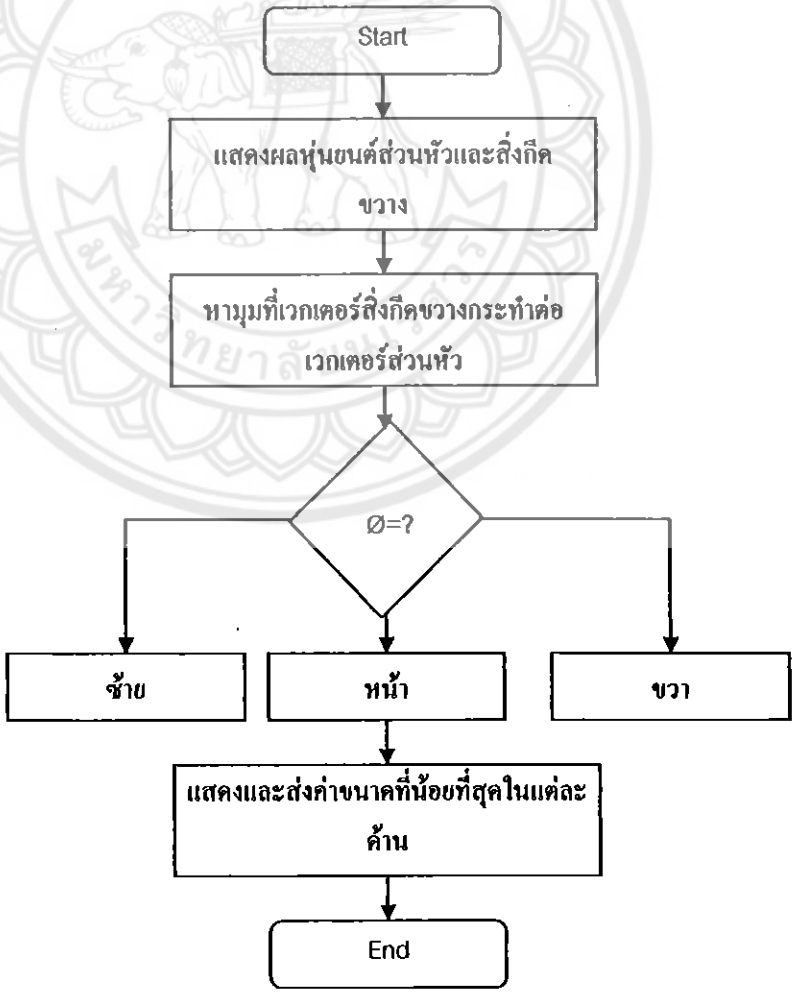
$$u_n = \frac{\sum_{j=1}^{M} \mu_j w_j}{\sum_{j=1}^{M} \mu_j} \tag{2.7}$$

เมื่อ $\mu_j = \mu_{A_1^j}(x_1)\,\mu_{A_2^j}(x_2)\ldots\mu_{A_n^j}(x_1)$ โดย n คือจำนวนของอินพุต

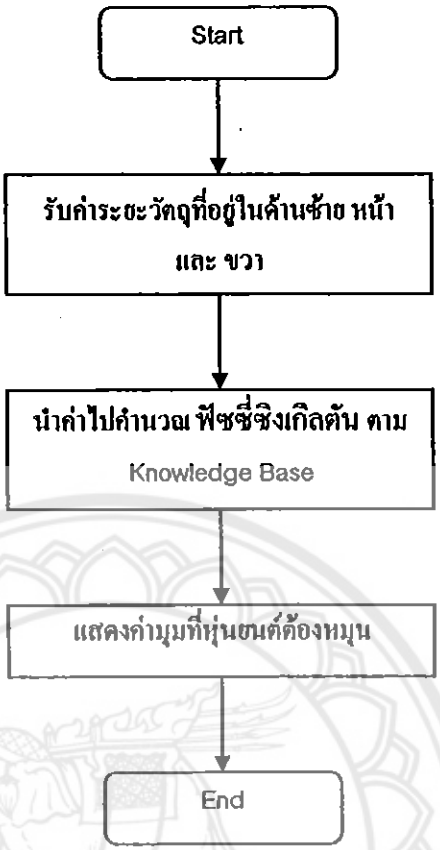
บทที่3
การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกิลตัน

3.1 การควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกิลตัน

โครงสร้างทั่วๆไปของการเขียนโปรแกรมสำหรับโครงงานนี้แบ่งออกเป็นสองส่วนใหญ่ๆ คือ ส่วนที่เป็นกราฟฟิกและส่วนของระบบฟัซซี่ ในส่วนของกราฟฟิกจะประกอบด้วยการออกแบบการจำลองการเคลื่อนที่ของหุ่นยนต์โดยอาจมีสิ่งกีดขวางตั้งแต่ศูนย์หรือตามแต่ความต้องการของผู้ออกแบบ สำหรับโครงงานนี้กำหนดให้มีสิ่งกีดขวางตั้งแต่ 0 ถึง10 อีกส่วนคือระบบฟัซซี่ซึ่งเป็นส่วนสำคัญเพื่อที่ทำให้หุ่นยนต์ในแบบจำลองมีการตัดสินใจในการหลบหลีกสิ่งกีดขวางที่เกิดขึ้น การทำงานในส่วนของการเคลื่อนที่ของหุ่นยนต์อาจเขียนสรุปเป็นแผนภาพได้ดังรูปที่ 3.1



รูปที่3.1 แผนภาพแสดงการทำงานของโปรแกรมการควบคุมหุ่นยนต์เคลื่อนที่



รูปที่3.2 แผนภาพการทำงานของโปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์โดยฟัซซี่ซิงเกิลตัน

จากแผนภาพจะเห็นว่าโปรแกรมทั้งสองส่วนถูกเขียนขึ้นโดยเป็นอิสระต่อกัน แล้วจึงนำมารวมกันเป็นโปรแกรมแสดงภาพกราฟฟิกที่มีการเคลื่อนที่ของหุ่น โดยมีระบบฟัซซี่เป็นระบบควบคุมการตัดสินใจในการหลบหลีกเมื่อมีสิ่งกีดขวางอยู่ในเส้นทางเดินของหุ่นยนต์

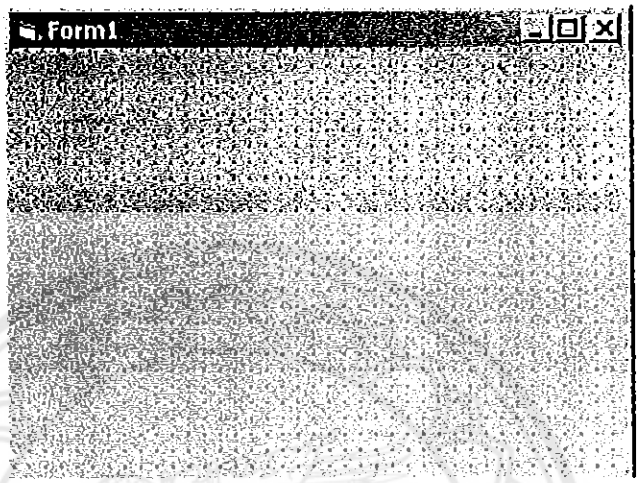
3.2 การออกแบบและการเขียนโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์

การออกแบบและการเขียนโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์นั้น โปรแกรมจะถูกเขียนด้วยไมโครซอฟท์วิชวลเบสิคเวอร์ชั่น 6.0 เนื่องจากเป็นโปรแกรมที่มีวิธีการเขียนโปรแกรมที่ทำความเข้าใจได้ง่าย และสามารถแสดงผลออกมาเป็นกราฟฟิกได้ง่ายอีกด้วย

การทำงานของโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์ ซึ่งประกอบด้วยส่วนของกราฟฟิกภาพจำลองการเคลื่อนที่ของหุ่นยนต์, การจำลองสิ่งกีดขวางและการคำนวณเพื่อใช้เป็นอินพุตของฟัซซี่โดยจะขออธิบายเป็นลำดับขั้นตอนได้ดังนี้

ขั้นตอนที่1

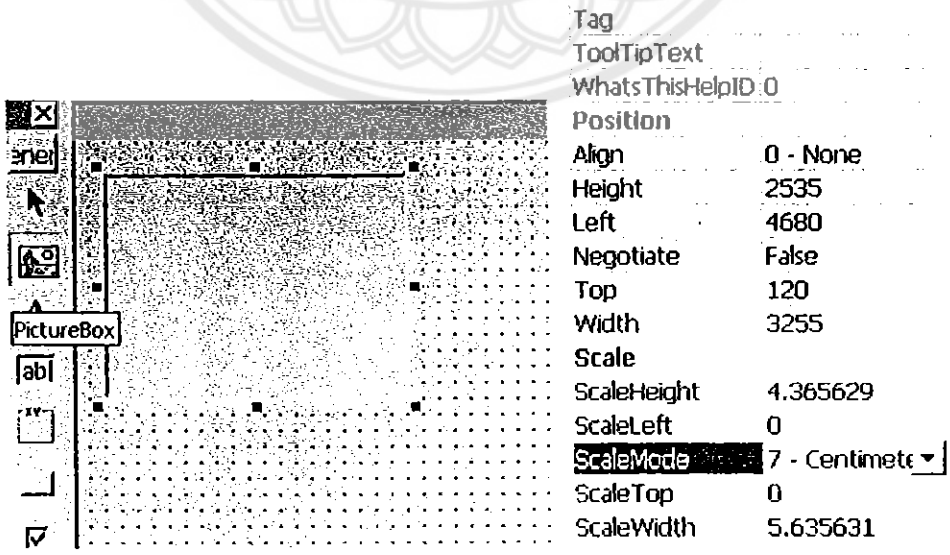
สร้าง Form1 ดังรูปที่3.3



รูปที่3.3 Form1ของโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์

ขั้นตอนที่2

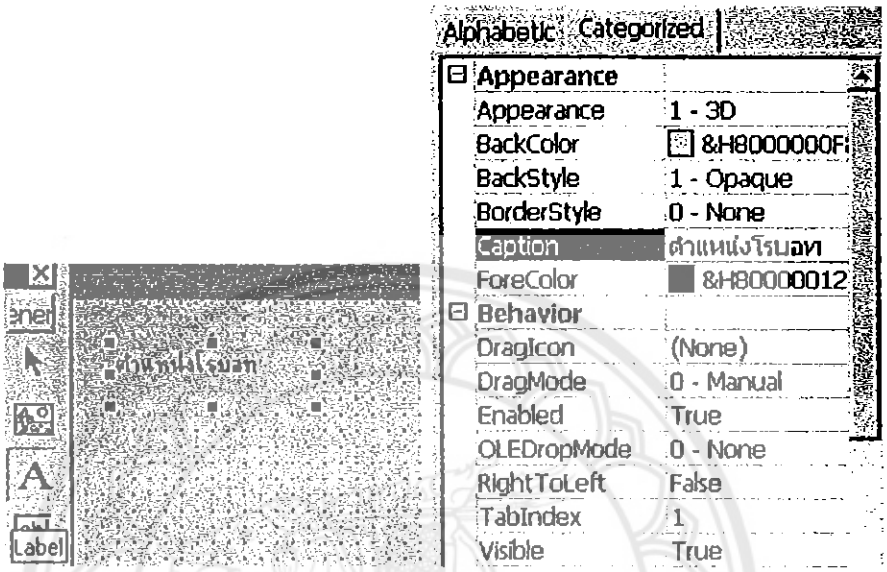
กำหนดพื้นที่ที่จะใช้ในการแสดงสิ่งกีดขวาง(วัตถุ) รวมถึงการเคลื่อนที่ของหุ่นยนต์ โดยใช้คำสั่ง PictureBox วาดลงไปแล้วกำหนดให้ ScaleModeในส่วนของ Propertiesเป็น 7 –Centimeter โดยกำหนดขนาดของ PictureBox เป็น 15 x 15 เซนติเมตรดังรูปที่3.4



รูปที่3.4 สร้างพื้นที่ในการแสดงสิ่งกีดขวาง

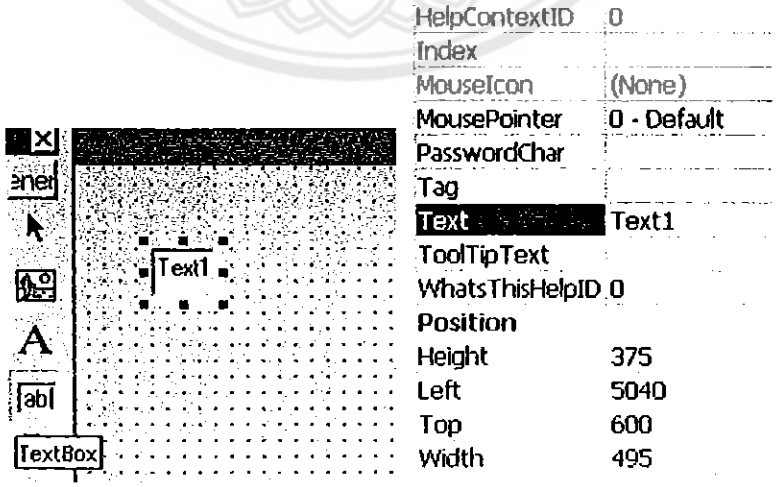
<u>ขั้นตอนที่3</u>

วางตัวหนังสือต่างๆโดยใช้คำสั่ง Label แล้วพิมพ์ข้อความลงใน Caption ในส่วนของ Properties ดังรูปที่ 3.5



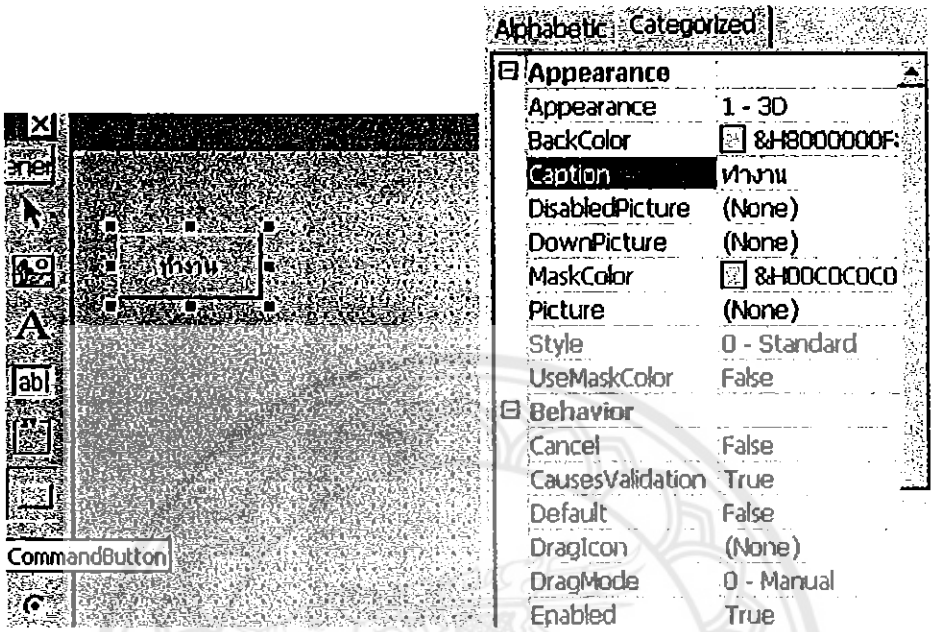
รูปที่3.5 ขั้นตอนที่3ของโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์

และวางรูปแบบการรับค่าต่างๆ และแสดงผลโดยใช้คำสั่ง TextBox ดังรูปที่ 3.6



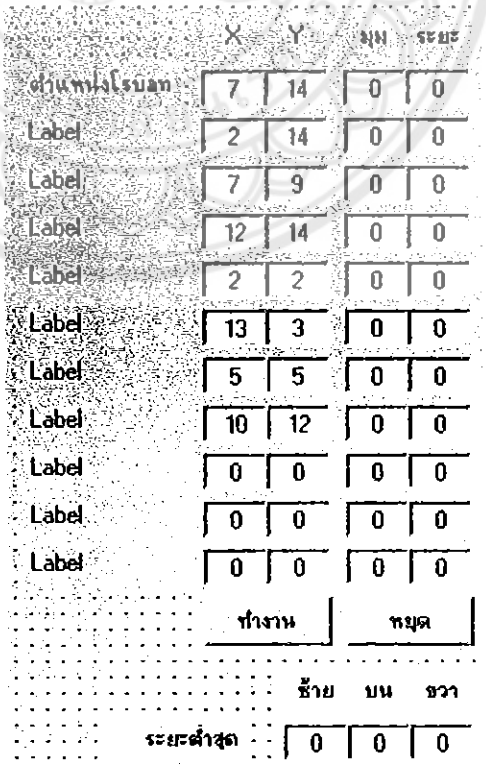
รูปที่3.6 การวางรูปแบบการรับค่าต่างๆ

สร้างปุ่มกดโดยใช้คำสั่ง CommandBottonแล้วพิมพ์ข้อความลงใน Caption ในส่วนของ Properties ดังรูปที่3.7



รูปที่3.7 สร้างCommandBotton

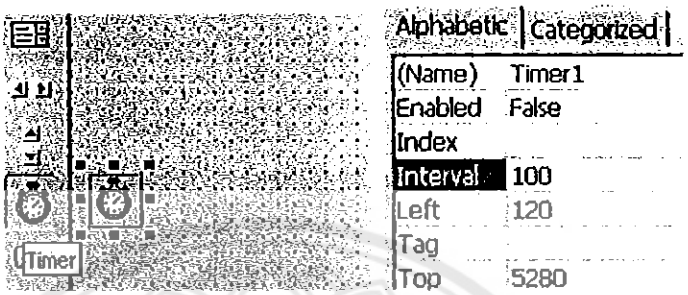
นำแต่ละส่วนมาจัดรูปแบบให้เหมาะสมกับการทำงาน ดังรูปที่3.8



รูปที่3.8 จัดแต่ง Form ของโปรแกรมการจำลองการเคลื่อนที่ของหุ่นยนต์

ขั้นตอนที่4

เป็นการสร้างส่วนของการทำงานเพื่อให้มีการทำงานเป็นรอบโดยใส่ Timer เพื่อควบคุมการทำงาน โดย Interval ช่วงเวลาใช้ควบคุมการทำงานเป็น 100 ms และ Enable ปรับเป็น False เพื่อให้ตัว Timer ยังไม่ทำงานเมื่อโหลดโปรแกรม

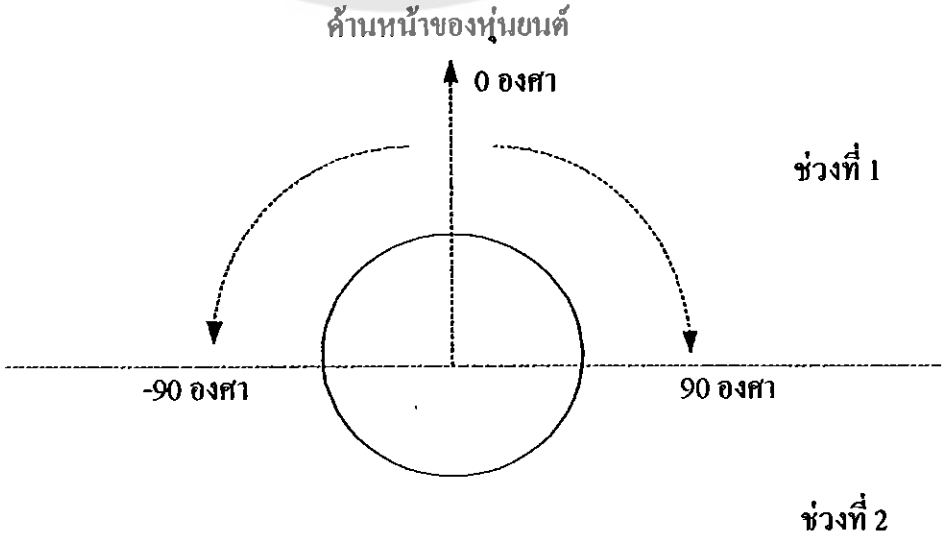


รูปที่3.9 การสร้างส่วนของการทำงานเพื่อให้มีการทำงานเป็นรอบ

3.3 การออกแบบและการเขียนโปรแกรมของระบบควบคุมการตัดสินใจในการหลบหลีกสิ่งกีดขวางของหุ่นยนต์โดยใช้ทฤษฎีฟัซซี่

3.3.1 การออกแบบตัวแปรอินพุต

การออกแบบระบบควบคุมการตัดสินใจในการหลบหลีกสิ่งกีดขวางของหุ่นยนต์โดยใช้ทฤษฎีฟัซซี่นั้นขั้นแรกต้องทำการกำหนดจำนวนตัวแปรอินพุต การกำหนดตัวแปรอินพุตจะขึ้นกับผู้ออกแบบตามแต่ความเหมาะสมที่จะนำไปใช้งาน โดยตัวแปรอินพุตที่กำหนดได้จะต้องทำให้อยู่ในรูปของฟัซซี่ ในที่นี้การตัดสินใจของหุ่นยนต์เพื่อหลบหลีกสิ่งกีดขวางจะเกิดขึ้นเมื่อสิ่งกีดขวางอยู่ในรัศมี 180 องศาดังจะแสดงให้เห็นในรูปที่3.10



รูปที่3.10 ช่วงขององศาที่มีผลและไม่มีผลต่อการตัดสินใจหลบหลีกสิ่งกีดขวางของหุ่นยนต์

จากรูปที่3.10 จะเห็นว่าช่วงที่1 เป็นช่วงที่มีผลต่อการตัดสินใจเพื่อหลบหลีกสิ่งกีดขวาง ส่วนช่วงที่2 จะเป็นช่วงที่มีไม่มีผลต่อการตัดสินใจเพื่อหลบหลีกสิ่งกีดขวาง กรณีที่มีสิ่งกีดขวางเกิดขึ้นนอกจากหุ่นยนต์จะต้องตัดสินใจว่าควรจะหลบสิ่งกีดขวางที่เกิดขึ้นนั้นๆหรือไม่แล้ว หากต้องมีหลบสิ่งกีดขวางควรจะหลบไปทางใดเป็นมุมองศาเท่าไร และเพื่อให้เพื่อให้หุ่นยนต์รับรู้ว่าสิ่งกีดขวางที่ตรวจจับได้อยู่ทางด้านใดและอยู่ในระยะที่ควรจะหลบหรือไม่โดยจะสร้างตัวแปรอินพุต3

อินพุต 1 : รับระยะความห่างของสิ่งกีดขวางที่อยู่ทางซ้ายมือของหุ่นยนต์

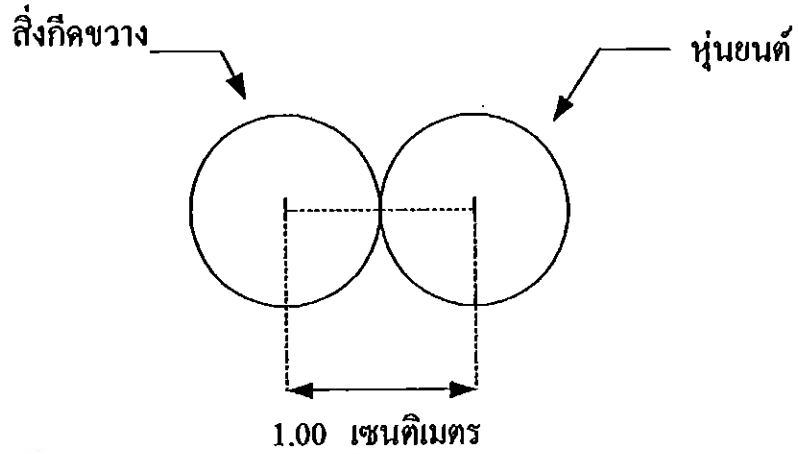
อินพุต 2 : รับระยะความห่างของสิ่งกีดขวางที่อยู่ทางขวามือของหุ่นยนต์

อินพุต 3 : รับระยะความห่างของสิ่งกีดขวางที่อยู่ด้านหน้าของหุ่นยนต์

สามารถแบ่งขอบเขตของอินพุตที่ต้องรับออกเป็น 3 ช่วงดังรูปที่3.11

รูปที่3.11 การแบ่งขอบเขตของอินพุต

เมื่อกำหนดจำนวนตัวแปรอินพุตแล้ว ขั้นตอนต่อมาคือการทำอินพุตให้อยู่ในรูปของฟัซซี่ ในที่นี้จะอาศัยฟังก์ชันของรูปสี่เหลี่ยมคางหมูมาใช้เพื่อความง่ายต่อการคำนวณ จากการออกแบบการจำลองการเคลื่อนที่ของหุ่นยนต์ในส่วนของการกำหนดขนาดของหุ่นยนต์จะเห็นว่าหุ่นยนต์มีลักษณะเป็นวงกลมรัศมี 0.5 เซนติเมตร เช่นเดียวกับสิ่งกีดขวาง จากรูปที่3.12 ระยะที่น้อยที่สุดที่หุ่นยนต์จะตัดสินใจว่าควรจะหลบสิ่งกีดขวางคือ 1 เซนติเมตร



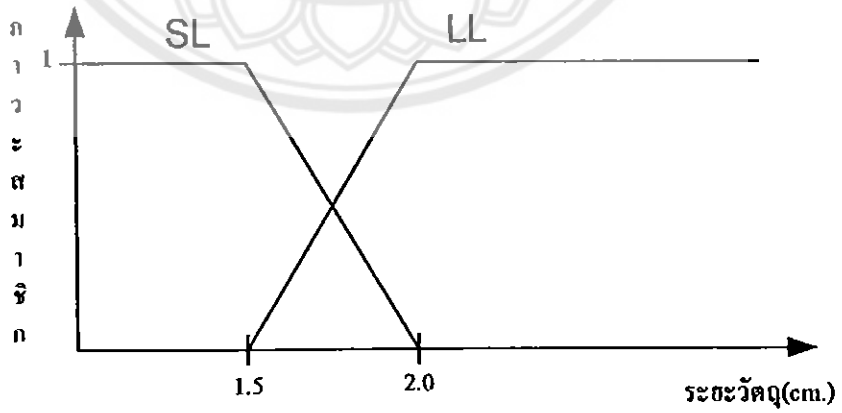
รูปที่3.12 ระยะที่น้อยที่สุดที่หุ่นยนต์จะตัดสินใจว่าควรจะหลบสิ่งกีดขวาง

เพื่อความเหมาะสมจึงกำหนดให้ระยะที่น้อยที่สุดที่หุ่นยนต์จะตัดสินใจว่าควรจะหลบสิ่งกีดขวางคือ 1.0 เซนติเมตร การทำอินพุตให้อยู่ในรูปของฟัซซี่ทั้งอินพุต 1, อินพุต 2และอินพุต 3 ถูกออกแบบให้เป็นไปในทางเดียวกัน ในที่นี้จะยกตัวอย่างเฉพาะการทำอินพุต 1ให้อยู่ในรูปของฟัซซี่ ให้อินพุต 1 มีฟังก์ชันภาวะสมาชิก 2 ฟังก์ชันมีตัวแปรเชิงภาษาดังต่อไปนี้

ระยะวัดฤทางซ้ายน้อยมากแทนด้วยตัวย่อ SL

ระยะวัดฤทางซ้ายมากมากแทนด้วยตัวย่อ LL

โดยช่วงของฟังก์ชันภาวะสมาชิกแต่ละตัวจะมีฐานกว้างเท่าไรนั้นขึ้นอยู่กับผู้ออกแบบ ในที่นี้กำหนดให้ SL อยู่ในช่วง 0 – 1.5 เซนติเมตร และ LLอยู่ในช่วง 1.5 – 2.0 เซนติเมตร



ตัวแปรอินพุต 1 : สิ่งกีดขวางอยู่ทางซ้ายของหุ่นยนต์

รูปที่3.13 กราฟของตัวแปรอินพุต 1

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

150166911

3.3.2 การออกแบบกฎ

เมื่อมีอินพุตแล้วขั้นตอนต่อมาคือการออกแบบกฎโดยอาศัยการอนุมาน ภายใต้เงื่อนไขของฟัซซี่ซิงกิลตัน ตัวแปรอินพุตมีทั้งหมด 3 ตัวแปร โดยแต่ละตัวแปรมีฟังก์ชันภาวะสมาชิก 2 ฟังก์ชัน แสดงว่าทั้งระบบต้องมีกฎทั้งหมดเท่ากับ 2^3 หรือ 8 กฎ

เมื่อ SL เป็นระยะวัตถุทางซ้ายน้อยมาก

LL เป็นระยะวัตถุทางซ้ายมากมาก

SR เป็นระยะวัตถุทางขวาน้อยมาก

LR เป็นระยะวัตถุทางขวามากมาก

SF เป็นระยะวัตถุทางด้านหน้าน้อยมาก

LF เป็นระยะวัตถุทางด้านหน้ามากมาก

ตารางที่3.1 ตารางแสดงกฎของฟัซซี่

ระยะวัตถุทางซ้าย (L)	ระยะวัตถุทางขวา (R)	ระยะวัตถุทางด้านหน้า (F)	มุม (องศา)
SL	SR	SF	180
SL	SR	LF	60
SL	LR	SF	0
SL	LR	LF	30
LL	SR	SF	-60
LL	SR	LF	30
LL	LR	SF	-30
LL	LR	LF	0

จากหัวข้อที่2.5 ว่าด้วยเรื่องกฎของฟัซซี่ซิงกิลตัน ตารางที่3.1สามารถเขียนโดยอาศัยการอนุมานภายใต้รูปแบบของ *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(ระยะวัตถุอยู่ทางซ้ายน้อยมาก) and Right is SR(ระยะวัตถุอยู่ทางขวาน้อยมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is 180

กฎ 2 : If left is SL (ระยะวัตถุอยู่ทางซ้ายน้อยมาก) and Right is SR(ระยะวัตถุอยู่ทางขวาน้อยมาก) and Font is LF(ระยะวัตถุอยู่ทางด้านหน้ามากมาก) Then Degree is 0

กฎ 3 : If left is SL (ระยะวัตถุอยู่ทางซ้ายน้อยมาก)and Right is LR(ระยะวัตถุอยู่ทางขวามากมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is 60

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

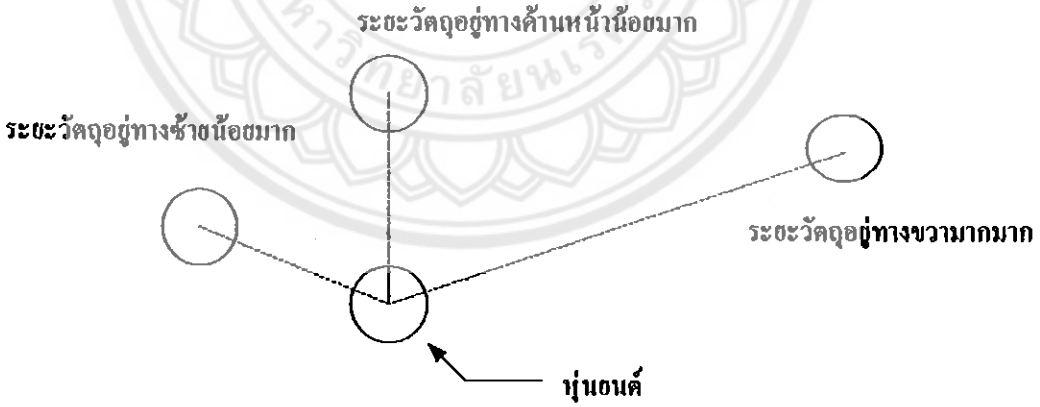
กฎ 5 : If left is LL (ระยะวัตถุอยู่ทางซ้ายมากมาก) and Right is SR (ระยะวัตถุอยู่ทางขวาน้อยมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is -60

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

กฎ 7 : If left is LL (ระยะวัตถุอยู่ทางซ้ายมากมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is 30

กฎ 8 : If left is LL (ระยะวัตถุอยู่ทางซ้ายมากมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) and Font is LF(ระยะวัตถุอยู่ทางด้านหน้ามากมาก) Then Degree is 0

เมื่อ w_j เป็นเอาต์พุตของแต่ละกฎซึ่งเป็นเอาต์พุตแบบอนาลอกการจะตัดสินใจว่าแต่ละกฎควรมีเอาต์พุตเท่าไรนั่นขึ้นอยู่กับผู้ออกแบบนอกจากนี้แล้วยังต้องคำนึงถึงความเป็นจริงที่ควรจะเป็น เช่นเมื่อมีระยะวัตถุอยู่ทางซ้ายน้อยมากและทางขวามีวัตถุห่างออกไปเป็นระยะมากมากเกิดขึ้นประกอบกับมีระยะวัตถุอยู่ทางด้านหน้าน้อยมากโดยจะแสดงให้เห็นดังรูปที่3.14



รูปที่3.14 รูปแสดงตำแหน่งของสิ่งกีดขวาง

จากรูปถ้าสิ่งกีดขวางอยู่ในระยะที่มีผลต่อการตัดสินใจ จะเห็นว่าเอาต์พุตที่ได้ควรจะเป็นองศาบวก เพื่อแสดงให้ทราบว่าหุ่นยนต์ควรจะหลบโดยการหมุนหนีจากแนวแกนเดิมเป็นองศาบวก เท่าไรการเขียนโปรแกรมของกฎจะเป็นส่วนที่ต้องนำไปใช้สำหรับการหาเอาต์พุตของระบบต่อไป จึงเป็นการคำนวณเพื่อหาค่าของ $\mu_j w_j$ ของแต่ละกฎเท่านั่น เพื่อให้ทราบว่ากฎทุกกฎทำงานแล้ว โดยค่า $\mu_j w_j$ ของแต่ละกฎจะถูกเก็บไว้ในตัวแปร x1,x2, …, x7และx8

3.3.3 เอาต์พุตของระบบ

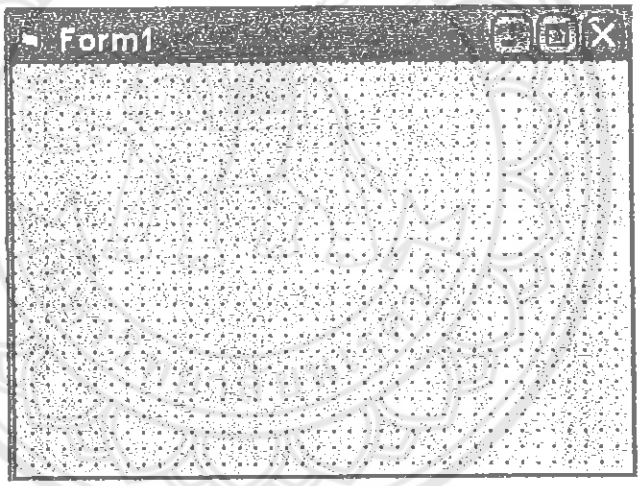
กฎทุกกฎจะถูกเรียกให้ทำงานและได้เอาต์พุตของแต่ละกฎ เอาต์พุตสุดท้ายซึ่งเป็นเอาต์พุตของระบบจะคำนวณหาได้จากทฤษฎีฟัซซี่ชิงกิลตันตามสมการ2.7 จะต้องมีการเขียนโปรแกรมเพื่อให้คำนวณหาค่า μ_j ของแต่ละกฎ และเก็บค่า μ_j ของแต่ละกฎ ไว้ในตัวแปร y1,y2, ..., y7และy8 สามารถเขียนโปรแกรมได้

3.3.4 การแสดงผลของโปรแกรมระบบควบคุมการตัดสินใจในการหลบหลีกสิ่งกีดขวางของหุ่นยนต์โดยใช้ทฤษฎีฟัซซี่

การแสดงผลหรือการสั่งให้โปรแกรมทำงาน เป็นส่วนที่ต้องมีการใส่ข้อมูลและมีการแสดงผลของเอาต์พุตที่คำนวณได้ โดยจะอธิบายเป็นลำดับขั้นตอนได้ดังต่อไปนี้

ขั้นตอนที่1

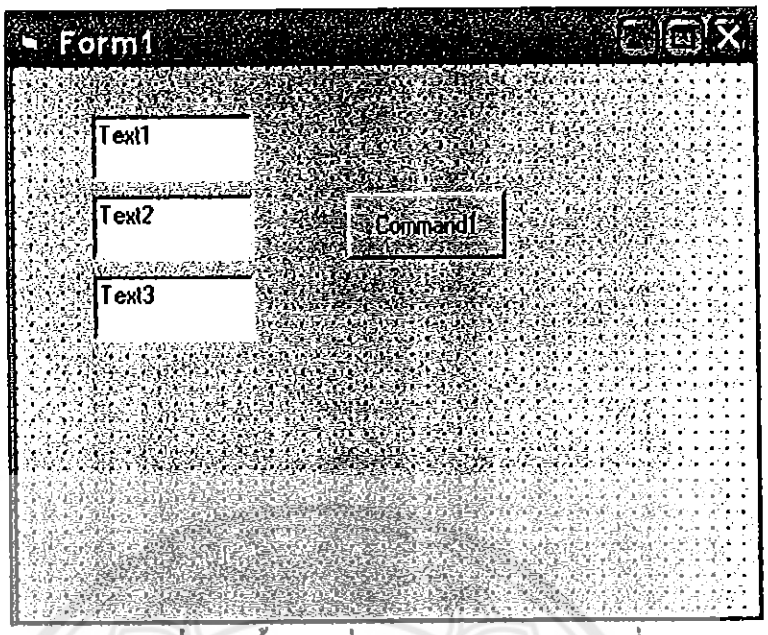
เริ่มจากการสร้าง Form1 ดังรูปที่3.15



รูปที่3.15 Form1ของโปรแกรมส่วนฟัซซี่

ขั้นตอนที่2

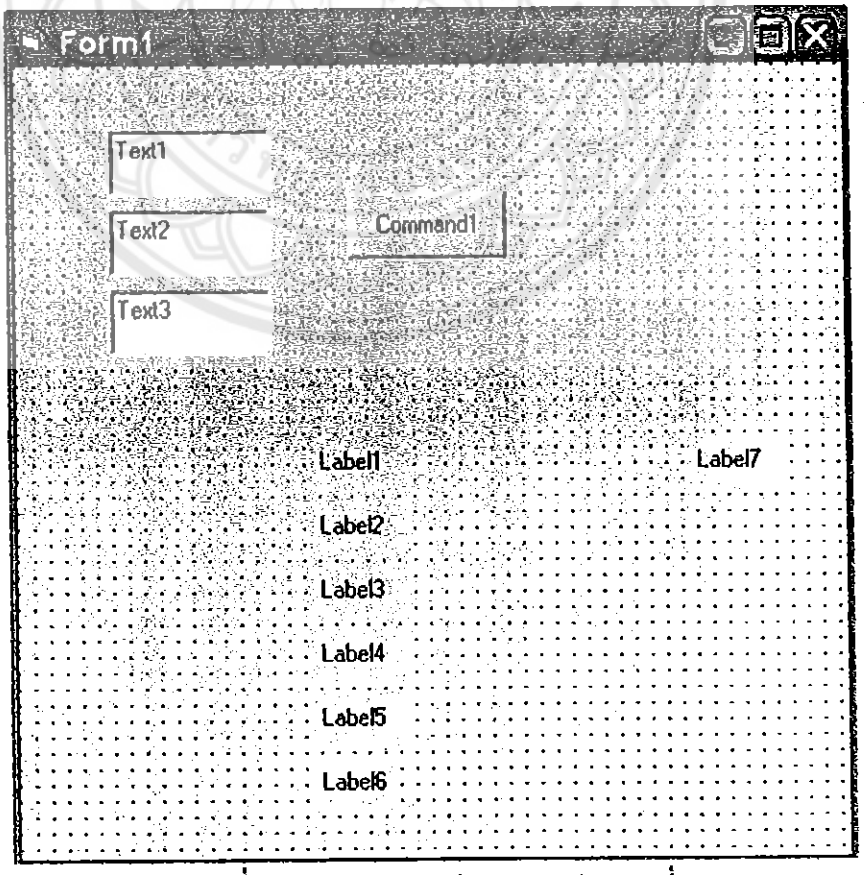
สร้าง TextBox ขึ้นมา 3 TextBox เพื่อใส่ค่าอินพุต1, อินพุต2 และอินพุต3 สร้าง CommandButtom เพื่อที่จะเขียนคำสั่งต่างๆ เพื่อเรียกการทำงานของฟังก์ชันที่สร้างขึ้นในหัวข้อที่ผ่านมา



รูปที่3.16 ขั้นตอนที่ 2ของโปรแกรมส่วนฟัซซี่

<u>ขั้นตอนที่3</u>

สร้าง Label ตามรูปที่3.17 ขึ้นมา 7 Label เป็นส่วนที่จะแสดงผลที่ได้จากการคำนวณ



รูปที่3.17 แสดงผลของโปรแกรมส่วนฟัซซี่

ให้

Label1 แสดงค่าของฟังก์ชันภาวะสมาชิก SL

Label2 แสดงค่าของฟังก์ชันภาวะสมาชิก LL

Label3 แสดงค่าของฟังก์ชันภาวะสมาชิก SR

Label4 แสดงค่าของฟังก์ชันภาวะสมาชิก LR

Label5 แสดงค่าของฟังก์ชันภาวะสมาชิก SF

Label6 แสดงค่าของฟังก์ชันภาวะสมาชิก LF

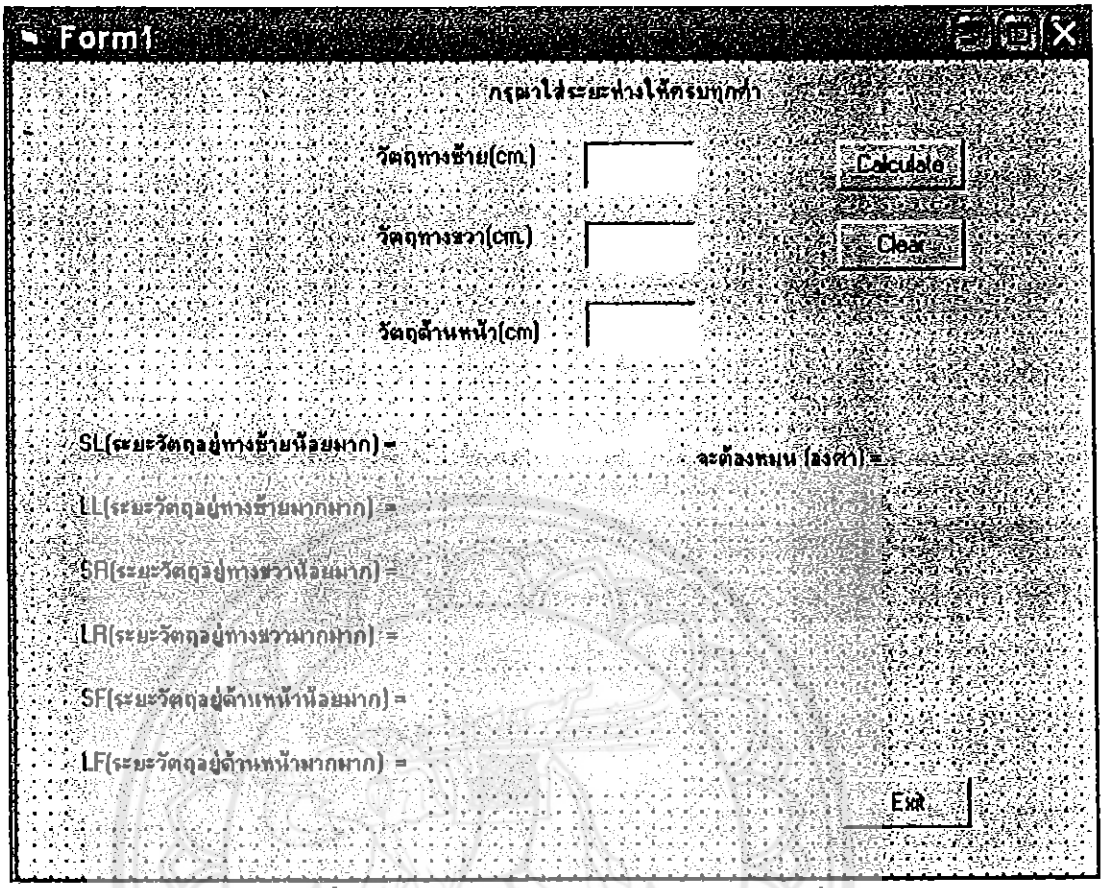
Label7 แสดงค่าเอาต์พุตระบบ

เขียนโปรแกรมใน Command1

```
Label1.Caption = Lt1
Label2.Caption = Lt2
Label3.Caption = Rt1
Label4.Caption = Rt2
Label5.Caption = F1
Label6.Caption = F2
Label7.Caption = Z
```

ขั้นตอนที่4

เป็นขั้นตอนสุดท้ายเพื่อตกแต่ง Form ดังรูปที่3.18 ใส่คำอธิบายเพื่อให้เข้าใจว่าค่าที่แสดงผลเป็นค่าของตัวแปรใด



รูปที่3.18 ตกแต่ง Form ของโปรแกรมส่วนฟัซซี่

บทที่4

ผลการทดลอง

4.1 ผลการทดลองการหลบหลีกสิ่งกีดขวางของหุ่นยนต์

จากโปรแกรมที่ 1 ให้ผลลัพธ์ออกมา 3 ค่า แล้วส่งค่าทั้ง 3 นี้ไปคำนวณในส่วนของ Fuzzy Singleton เพื่อหาค่ามุมที่จะใช้ให้หุ่นยนต์เคลื่อนที่ต่อไป โดย

L = Lmin

R = Rmin

F = Tmin

ตัวแปร L, R, F รับค่าไปใช้คำนวณในส่วนของ Fuzzy Singleton ซึ่งเป็นค่าขนาดที่น้อยที่สุด แล้ว Fuzzy Singleton จะส่งค่ามุมออกมา นำค่ามุมที่ได้ใหม่ไปรวมกับค่ามุมของ Head คำนวณค่าจุดปลายใหม่ โดย

c(1) = c(1) + มุมจาก Fuzzy

สั่งให้หุ่นยนต์เดินไปยังทิศที่คำนวณได้ใหม่เป็นระยะทาง 0.1 เซนติเมตร แล้วกลับไปเริ่มการคำนวณใหม่ตั้งแต่การหาตำแหน่งถัดไปของหุ่นยนต์ โดย

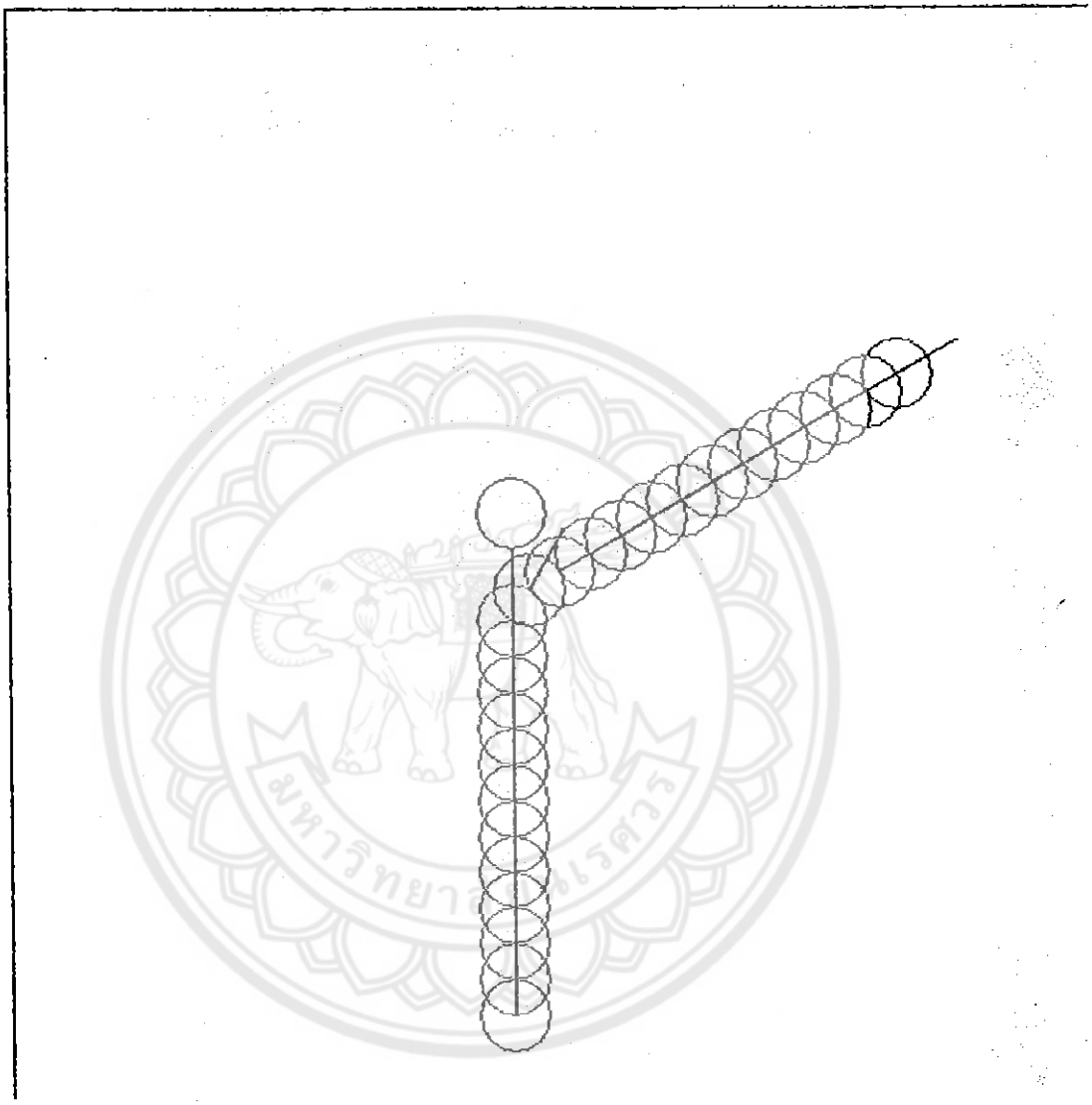
zO = (90 - c(1)) * Pi / 180

a(1) = a(1) + 0.1 * Cos(zO)

b(1) = b(1) - 0.1 * Sin(zO)

เครื่องรับค่า a(1) ใหม่ลงใน a(1) ตัวเดิมซึ่งทำให้ไม่จำเป็นต้องเปลี่ยนตัวแปร เมื่อค่า a(1) และ b(1) เปลี่ยน ก็ทำให้ค่าอื่นๆเปลี่ยนตามไปด้วย

ถ้ากำหนดให้วัตถุมีจุดศูนย์กลางอยู่ที่ (7, 7) และหุ่นยนต์มีจุดศูนย์กลางอยู่ที่ (7, 4) จะได้ผลดังรูปที่ 4.1



รูปที่4.1 หุ่นยนต์จะเคลื่อนที่เพียงเพื่อหลบแต่วัตถุ โดยไม่สนใจทิศทางที่กำลังเคลื่อนที่ไป

4.2 ผลการทดลองการเคลื่อนที่เข้าหาจุดเป้าหมายของหุ่นยนต์

เป็นส่วนที่มีการกำหนดจุดหมาย เพื่อกำหนดให้หุ่นยนต์มีจุดมุ่งหมายในการเคลื่อนที่ โดยการรับค่าจุดศูนย์กลางของ จุดหมาย แล้วนำไปคำนวณ โดย

กำหนดเงื่อนไขว่า ให้หุ่นยนต์หมุน Head ไปทาง จุดหมาย แล้วเคลื่อนที่ไปในทิศทางนั้นเมื่อ Oout มีค่าเป็น 0 หมายความว่าถ้าหุ่นยนต์ไม่จำเป็นต้องเปลี่ยนทิศทางในการเคลื่อนที่แล้วให้หมุน Head แล้วเคลื่อนที่ไปในทิศทางนั้น

```
a(2) = Text95.Text
b(2) = Text96.Text
tgX = a(2) - a(1)
tgY = b(2) - b(1)
tgD = Sqr(tgX ^ 2 + tgY ^ 2)
```

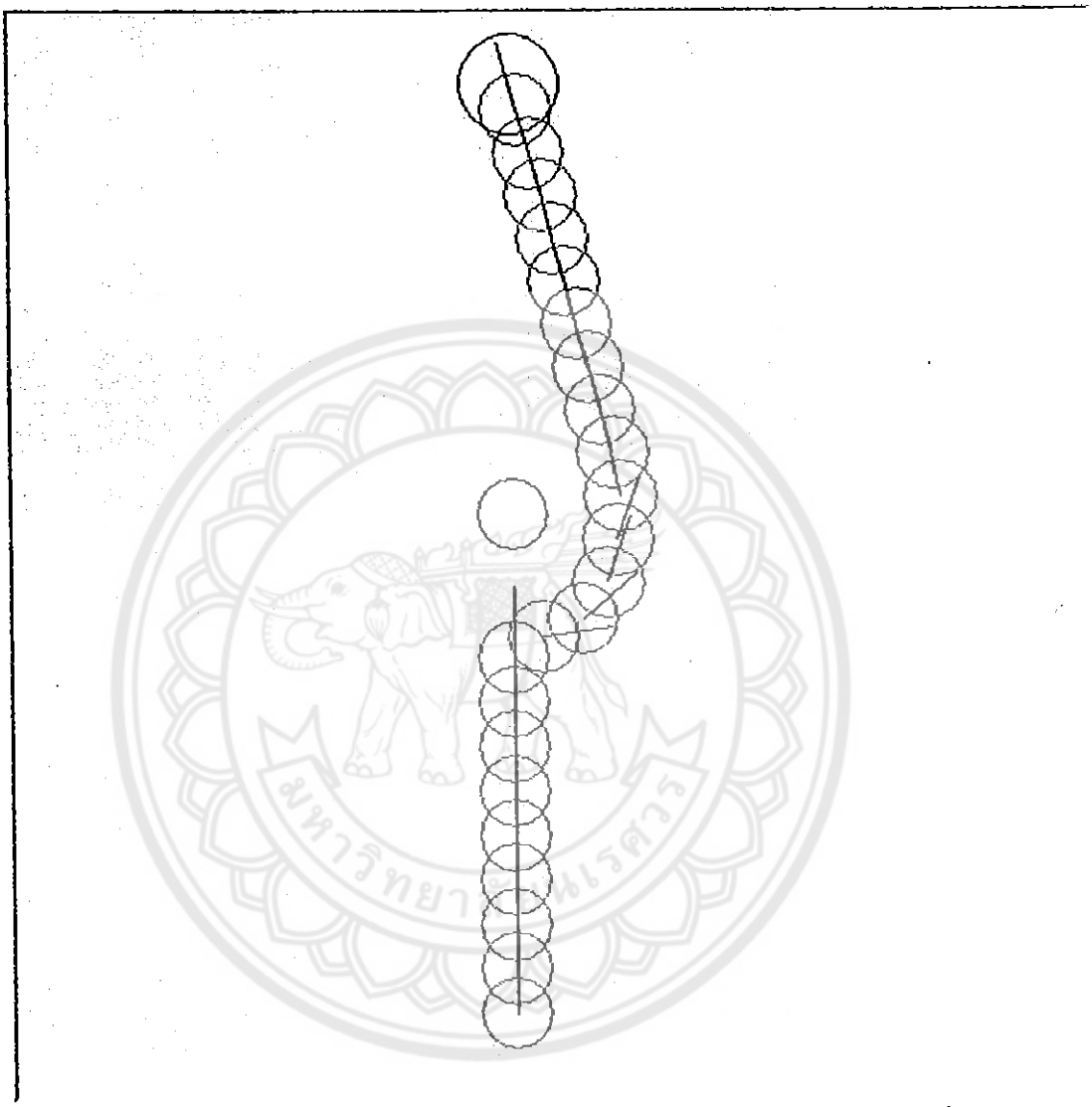
รับค่าจุดศูนย์กลางของ จุดหมาย ที่ตัวแปร a(2) และ b(2) แล้วหาเวกเตอร์และขนาดเหมือนกับส่วนวัตถุ

```
m(1) = a(1)
n(1) = b(1) - 1
j(0) = 0
k(0) = -1
d(0) = Sqr(j(0) ^ 2 + k(0) ^ 2)
```

ให้ Head ทำมุม 0 เป็นแนวแกนอ้างอิง แล้วหาเวกเตอร์และขนาดเพื่อใช้หามุมโดยใช้สมการเดิม แล้วนำมุมที่ได้นั้นไปคำนวณหาจุดศูนย์กลางจุดใหม่ที่หุ่นยนต์ต้องเคลื่อนที่ไป โดยใช้ Code

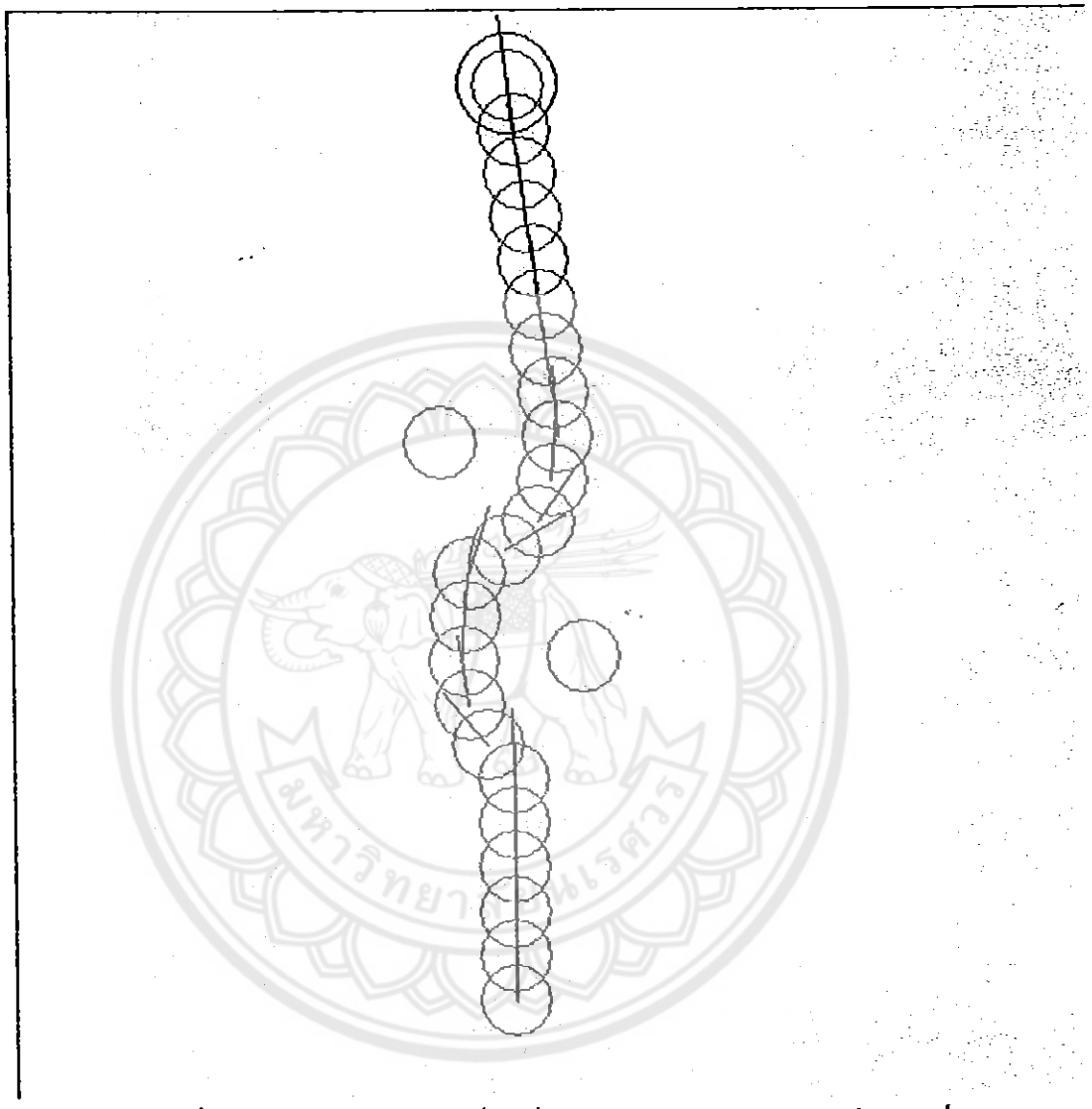
```
a(1) = a(1) + 0.02 * Cos(zO)
b(1) = b(1) - 0.02 * Sin(zO)
```

ถ้ากำหนดให้วัตถุมีจุดศูนย์กลางอยู่ที่ (7, 7) จุดหมายมีจุดศูนย์กลางอยู่ที่ (7, 1) และหุ่นยนต์มีจุดศูนย์กลางอยู่ที่ (7, 14) จะได้ผลดังภาพ



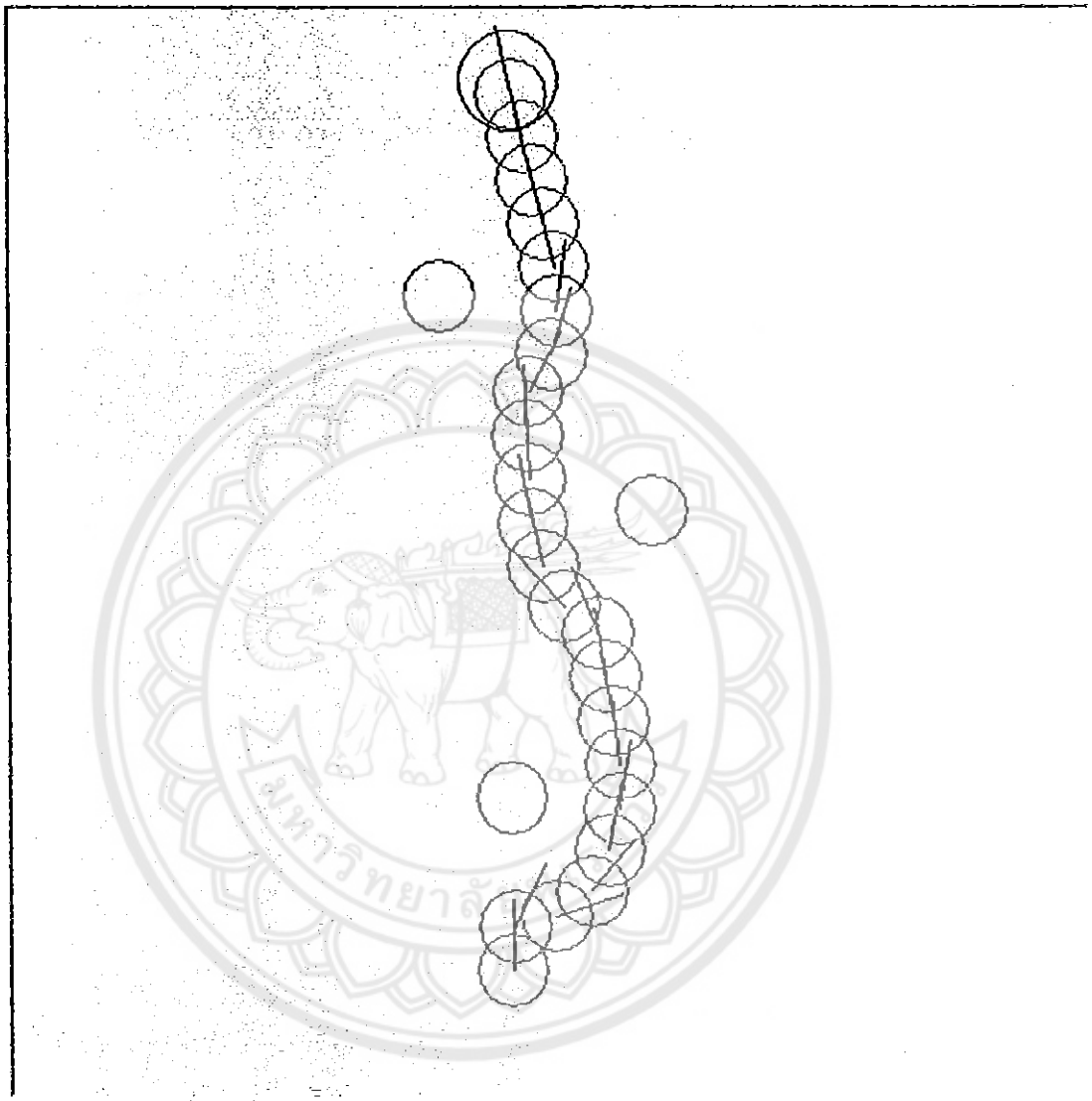
รูปที่4.2 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่1

วัตถุมีจุดศูนย์กลางอยู่ที่ (8, 9) และ (6, 6) หมายมีจุดศูนย์กลางอยู่ที่ (7, 1) และหุ่นยนต์มีจุดศูนย์กลางอยู่ที่ (7, 14) จะได้ผลดังรูปที่



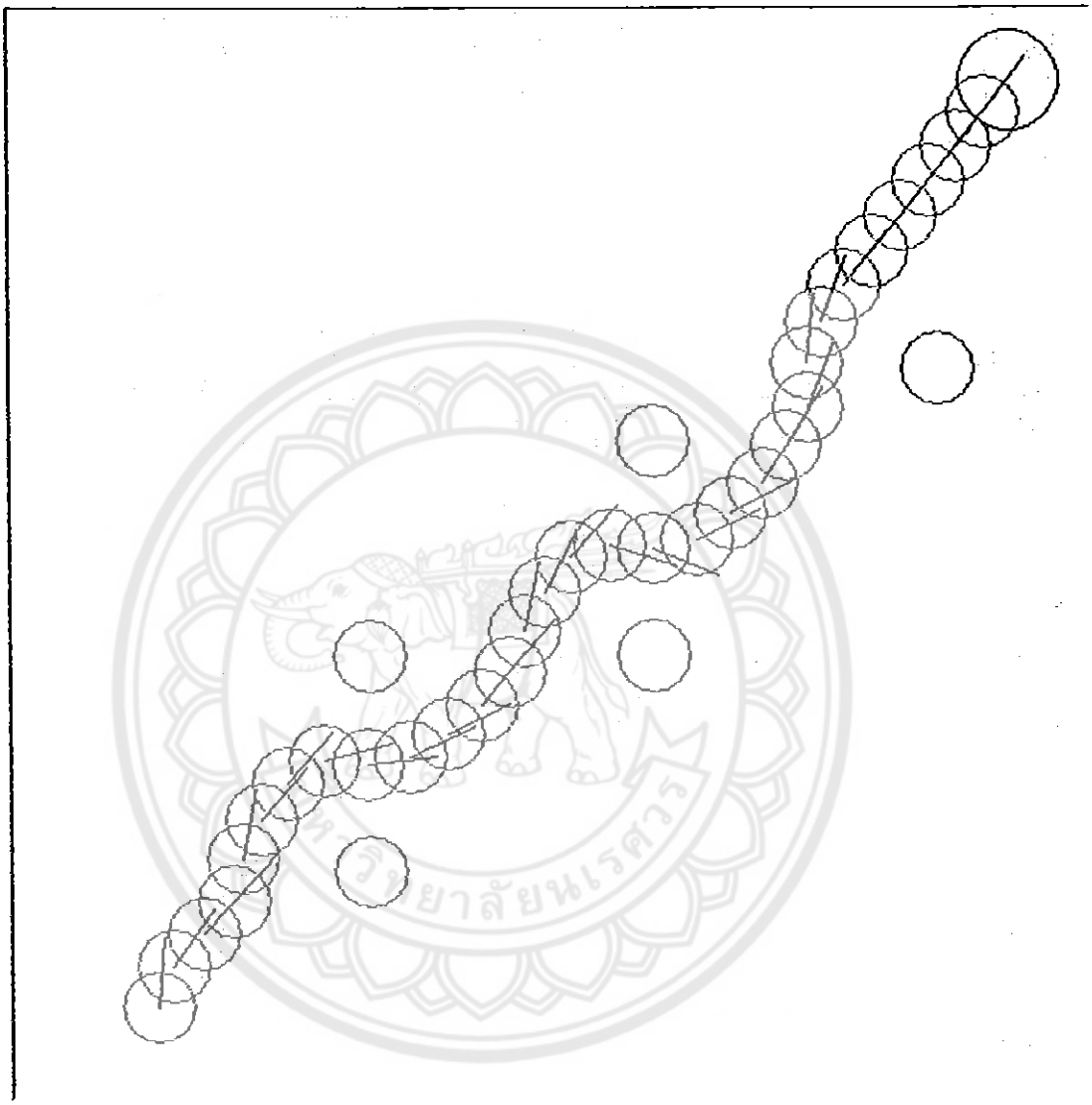
รูปที่4.3 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่2

วัตถุมีจุดศูนย์กลางอยู่ที่ (7, 11), (9, 7) และ (6, 4) จุดหมายมีจุดศูนย์กลางอยู่ที่ (7, 1) และหุ่นยนต์มีจุดศูนย์กลางอยู่ที่ (7, 14) จะได้ผลดังภาพ



รูปที่4.4 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่3

วัตถุมีจุดศูนย์กลางอยู่ที่ (5, 9), (5, 12), (9, 6), (9, 9) และ (13, 5) จุดหมายมีจุดศูนย์กลางอยู่ที่ (14, 1) และหุ่นยนต์มีจุดศูนย์กลางอยู่ที่ (2, 14) จะได้ผลดังภาพ



รูปที่4.5 ตัวอย่างแสดงการเคลื่อนที่ของหุ่นยนต์โดยกำหนดให้เป็นภาพที่4

บทที่5
สรุปผล

โครงงานนี้ศึกษาและพัฒนาโปรแกรมสำหรับการควบคุมหุ่นยนต์เคลื่อนที่โดยใช้ทฤษฎีฟัซซี่ซิงเกิลตัน โครงงานได้ออกแบบและจำลองผลการทำงานด้วยคอมพิวเตอร์ของฟัซซี่ซิงเกิลตันในการควบคุมการหลบหลีกสิ่งกีดขวางของหุ่นยนต์

ในส่วนของโปรแกรมฟัซซี่การออกแบบทำได้หลายวิธีขึ้นอยู่กับผู้ออกแบบ เนื่องจากกฎพื้นฐานของฟัซซี่มาจากประสบการณ์การเรียนรู้ของมนุษย์หรือ Knowledge Base หมายความว่าในการออกแบบกฎแต่ละกฎอาจเกิดข้อผิดพลาด หรือไม่สามารถแก้ปัญหาได้ดีเท่าที่ควร

การควบคุมระบบฟัซซี่สามารถนำไปประยุกต์ใช้เพื่อแก้ปัญหาที่เกิดจากความไม่เที่ยงตรงและความคลาดเคลื่อนของข้อมูล ในกรณีที่ปัญหามีความซับซ้อนสูงการออกแบบระบบควบคุมแบบฟัซซี่อาจทำโดยการสร้างเงื่อนไขหรือออกแบบฟังก์ชันภาวะสมาชิกให้มีความละเอียดและคลอบคุมมากขึ้น

ใช้ตัว Timer ในการควบคุมการทำงานของโปรแกรมโดยการดับเบิลคลิกที่ "Timer 1" ในหน้าต่างของ Object ที่วินโดวส์จะแสดง Code ขึ้นมา

```
Private Sub Timer1_Timer()

End Sub
```

ซึ่ง Code ต่อจากนี้ไปจะเขียนภายในส่วนของ Timer 1 นี้เท่านั้น

เริ่มในส่วนของสิ่งกัดขวางหรือวัตถุ ขั้นแรกต้องรับค่าจุดศูนย์กลางของวัตถุจาก TextBox ซึ่งรับค่าในรูปแบบของ x และ y แยกจากกัน ดังนั้นต้องสร้าง TextBox จำนวน 2 ชุด ต่อ วัตถุ 1 ชิ้น แล้วดับเบิลคลิกที่ "Timer 1" แล้วพิมพ์ Code ลงไปว่า

```
x(1) = Text1.Text
y(1) = Text2.Text
```

เป็นการรับค่า x และ y จาก TextBox ที่ชื่อ Text1 และ Text2 ลงไปใน x(1) และ y(1) ตามลำดับ ในกรณีที่ออกแบบให้มีวัตถุมากกว่า 1 ให้พิมพ์ Code เพิ่มไปเป็น

```
x(2) = Text3.Text
y(2) = Text4.Text
```

รับค่าจุดศูนย์และมุมเริ่มต้นของโรบอทจาก TextBox โดยวิธีเดียวกัน

```
a(1) = Text91.Text
b(1) = Text92.Text
c(1) = Text93.Text
```

เป็นการรับค่า x และ y จาก TextBox ที่ชื่อ Text91 และ Text92 ลงไปใน a(1) และ b(1) ตามลำดับ แลรับค่ามุมที่อยู่ในช่วง ±90 จาก Text93 ลงไปใน c(1)

คำนวณหาจุดปลายของตัวบอกทิศ(Head ขนาด 1 เซนติเมตร) โดยการนำค่า a(1), b(1) และ c(1) ไปคำนวณตาม Code ข้างล่าง

```
Pi = 3.14159265358979
zO = (90 - c(1)) * Pi / 180
m(1) = a(1) + Cos(zO)
n(1) = b(1) - Sin(zO)
```

กำหนดให้ค่า Pi เป็นค่าคงที่หมายถึง π แล้วคำนวณมาค่ามุมโดยการนำ 90 ไปลบด้วยมุมที่เก็บค่ามาจาก c(1) (เนื่องจากจุดอ้างอิงที่มุมเป็น 0 ของโรบอทอยู่บนแนวแกน y โดยมีทิศทำมุมตั้งฉากกับแกน x ทำให้ค่ามุมของ Head จะทำมุมเท่ากับ c(1) บนแกน y จำเป็นต้องนำ 90 ไปลบเพื่อให้ได้ค่ามุมที่อยู่บนแกน x) แล้วทำการแปลงค่ามุมจาก Degrees ไปเป็น Radian เพราะว่าค่าฟังก์ชันตรีโกณมิติใน Microsoft Visual Basic 6.0 นี้คิดเป็นแบบ Radian แล้วทำการคำนวณหาจุดปลายของ Head โดย m(1) = a(1) + Cos(z) ได้จุดปลายของ Headที่อยู่บนแกน x เก็บค่าไว้ที่ m(1) และ n(1) = b(1) - Sin(z) ได้จุดปลายของ Headที่อยู่บนแกน y เก็บค่าไว้ที่ n(1) แล้วนำค่าที่ได้ไปหาค่าเวกเตอร์บนแนวแกนและหาขนาดจากจุดศูนย์กลางของหุ่นยนต์ไปยัง Head

```
j(0) = m(1) - a(1)
k(0) = n(1) - b(1)
d(0) = Sqr(j(0) ^ 2 + k(0) ^ 2)
```

โดย j(0) จะเก็บค่า m(1) - a(1) ซึ่ง m(1) เป็นจุดปลายลบ a(1) ซึ่งเป็นจุดเริ่มต้นหรือจุดศูนย์กลาง จะได้ขนาดของเวกเตอร์บนแนวแกน x
ส่วน k(0) จะเก็บค่า n(1) - b(1) ซึ่ง n(1) เป็นจุดปลายลบ b(1) ซึ่งเป็นจุดเริ่มต้นหรือจุดศูนย์กลาง จะได้ขนาดของเวกเตอร์บนแนวแกน y นำเวกเตอร์ทั้งสองมาหาขนาดความยาว ซึ่งในตอนแรกได้กำหนดค่าความยาวเป็น 1 เซนติเมตร แต่เพื่อต้องการตรวจสอบค่าความถูกต้องที่ได้จากการคำนวณโดยค่าขนาดความยาวที่ออกมาต้องมีค่าเป็น 1

นำค่าต่างๆใน ไปวาดเส้นตรงเพื่อแสดงส่วน Head และวงกลมในส่วนหุ่นยนต์โดยใช้คำสั่ง

```
Picture1.Line (a(1), b(1))-(m(1), n(1))
```

เครื่องจะลากเส้นตรงจากจุดเริ่มต้นหรือจุดศูนย์กลางของโรบอท (a(1), b(1)) ไปยังจุดปลายที่ได้จากการคำนวณ (m(1), n(1))

```
Picture1.Circle (a(1), b(1)), 0.5, RGB(255, 0, 0)
```

เครื่องจะสร้างตัวหุ่นยนต์ให้เป็นวงกลมรัศมี 0.5 เซนติเมตร โดยโรบอทมีจุดศูนย์กลางอยู่ที่ (a(1),b(1)) สีแดง

เขียนฟังก์ชัน Arcsin และ Arccos ไว้ส่วนล่างสุด เพื่อใช้ในการคำนวณหาค่ามุมระหว่าง Head กับวัตถุโดย

```
Public Static Function Arcsin(x As Double) As Double
Arcsin = (Atn(x / Sqr(-x * x + 1)))
End Function
```

เป็นการประกาศฟังก์ชัน Arcsin โดยการรับรับค่าที่จะนำมาคำนวณ (x) อยู่ในรูปแบบ Double แล้วนำค่าที่รับมานั้นไปคำนวณในสมการ Arcsin = (Atn(x / Sqr(-x * x + 1))) ถ้าค่า x ที่นำมาคำนวณมีค่าเป็น 1 หรือ –1 จะไม่สามารถหาค่ามุมได้

```
Public Static Function Arccos(x As Double) As Double
Arccos = (Atn(-x / Sqr(-x * x + 1)) + 2 * Atn(1))
End Function
```

เป็นการประกาศฟังก์ชัน Arccos เหมือนกับ การประกาศฟังก์ชัน Arcsin

เนื่องจากโปรแกรมนี้ออกแบบให้รับค่าวัตถุได้ 10 ค่า แล้วต้องนำค่าของวัตถุแต่ละชิ้นนำไปวาดเป็นวงกลมและคำนวณหาขนาด จึงเขียน Code ให้มีการวนคำสั่ง 10 รอบ เพื่อให้ง่ายต่อการเขียน Code โดยใช้คำสั่ง

```
For i = 1 To 10

Next I
```

เครื่องจะกำหนดให้ค่า i =1 แล้วจะทำงานที่อยู่ระหว่าง For และ Next จนเมื่อโปรแกรมรันถึงบรรทัด Next i เครื่องจะบวกค่า i เพิ่มอีก 1 แล้วทำการวนกลับมาทำงานที่บรรทัด For i = 1 To 10 อีกครั้ง แล้ว For จะทำการตรวจสอบค่า i ถ้าค่า i มีค่าเท่ากับ 11 แล้วเครื่องจะข้ามไปทำงานคำสั่งที่อยู่ต่อจากบรรทัด Next I

เริ่มส่วนของวัตถุ อันดับแรกต้องทำการตรวจสอบจุดศูนย์กลางของวัตถุที่เก็บค่ามาจาก TextBox ถ้าหากจุดศูนย์กลางของวัตถุมีค่าเป็น 0 ทั้ง x และ y เครื่องจะป้อนค่าเอาท์พุตต่างๆ (ความหมายของตัวแปรเอาท์พุตอยู่ในหัวข้อถัดไป) เป็น 0

```
If x(i) = 0 And y(i) = 0 Then
```

กำหนดเงื่อนไขให้ ถ้าทั้ง x และ y มีค่าเป็น 0 แล้วค่าเอาท์พุตต่างๆ จะเป็น 0

```
d(i) = 0
Oc(i) = 0
Os(i) = 0
R(i) = 0
L(i) = 0
T(i) = 0
dR(i) = 0
dL(i) = 0
dT(i) = 0
```

แต่ถ้าหากค่าจุดศูนย์กลางของวัตถุ มีค่าใดค่าหนึ่งไม่เป็น 0 เครื่องเครื่องจะข้ามมารันที่คำสั่งตั้งแต่บรรทัด Else ลงไป ซึ่งบรรทัดถัดไปจะเป็นการวาดวงกลมรัศมี 0.5 เซนติเมตร สีน้ำเงิน มีจุดศูนย์กลางอยู่ที่ x(i), y(i) ซึ่งเป็นคำสั่งเดียวกับตอนที่วาดตัวหุ่นยนต์

```
Else
Picture1.Circle (x(i), y(i)), 0.5, RGB(0, 0, 255)
```

แล้วหาค่าเวกเตอร์บนแนวแกนและหาขนาดจากจุดศูนย์กลางของหุ่นยนต์ไปยังจุดศูนย์กลางของวัตถุ ซึ่งเป็นคำสั่งเดียวกับตอนที่หาขนาดของ Head

```
j(i) = x(i) - a(1)
k(i) = y(i) - b(1)
d(i) = Sqr(j(i) ^ 2 + k(i) ^ 2)
```

ทำการแบ่งช่วงของมุมที่ Head กระทำต่อวัตถุออกเป็น 4 ช่วง โดยที่

ช่วงที่ 1 Top ทำมุมอยู่ในช่วง $\pm 30^{\circ}$

ช่วงที่ 2 Right ทำมุมอยู่ในช่วง มากกว่า 30° แต่ไม่เกิน 90°

ช่วงที่ 3 Left ทำมุมอยู่ในช่วงน้อยกว่า -30° แต่ไม่เกิน -90°

ช่วงที่ 4 มุมที่อยู่นอกเหนือจากนี้ให้มีขนาดมุมเป็น 0

การหามุมระหว่างเวกเตอร์ของ Head กระทำต่อ วัตถุ ส่วนแรกใช้สมการเหล่านี้

$$\vec{A} \cdot \vec{B} = |\vec{A}||\vec{B}| \mathrm{Cosn}\theta \qquad (3.1)$$

$$\theta = \mathrm{Cos}^{-1} \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|} \qquad (3.2)$$

$$\vec{A} \cdot \vec{B} = (X_1 \times X_2) - (Y_1 \times Y_2) \qquad (3.3)$$

แล้วทำการเขียน Code ลงไป โดยแยกมุมของ Sin^{-1} ออกมาเพื่อความสะดวกในการคำนวณ

```
o1(i) = ((j(0) * k(i)) - (j(i) * k(0))) / (d(0) * d(i))
```

ค่ามุมของ Sin^{-1} เก็บอยู่ในตัวแปร o1(i) แต่ทั้งนี้ เนื่องจากการแทนค่า o1(i) หรือ x ในสมการ Arcsin ค่า o1(i) หรือ x นี้ต้องไม่เป็น 1 ดังข้อ 13 ดังนั้นต้องทำการกำหนดเงื่อนไขในการรับค่าของ o1(i) ก่อนที่แทนลงในสมการโดย

```
If o1(i) = 1 Then
Os(i) = 90

ElseIf o1(i) = -1 Then
Os(i) = -90

Else
Os(i) = Arcsin(((j(0) * k(i)) - (j(i) * k(0))) / (d(0) * d(i))) * 180 / Pi
End If
```

เมื่อ o1(i) = 1 จะกำหนดให้ Os(i) = 90 เนื่องจาก $Sin^{-1}(1) = 90$ แต่ถ้า o1(i) = -1 จะกำหนดให้ Os(i) = -90 เนื่องจาก $Sin^{-1}(1) = -90$ และถ้า Os(i) มีค่านอกเหนือจากนี้แล้ว จะส่งค่าไปในสมการ Arcsin เมื่อได้ค่ามุมออกแล้วก็ทำการแปลงมุมให้อยู่ในรูป Degrees และส่วนที่ 2 จากสมการ 3.1 และสมการ3.2 โดยที่ค่าของ $\vec{A} \cdot \vec{B} = (X_1 \times X_2) - (Y_1 \times Y_2)$ แล้วทำการเขียน Code ลงไป โดยแยกมุมของ Cos^{-1} ออกมาเพื่อความสะดวกในการคำนวณ

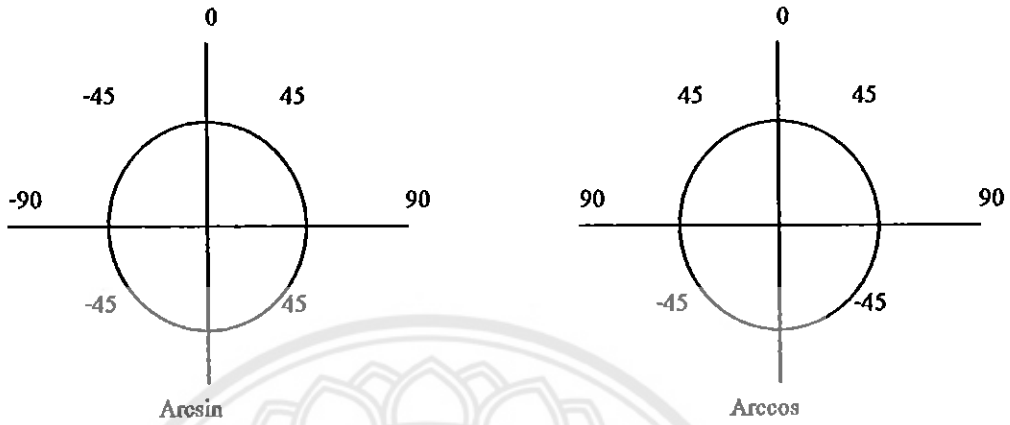
```
o2(i) = ((j(0) * j(i)) + (k(0) * k(i))) / (d(0) * d(i))
```

ค่ามุมของ Cos^{-1} เก็บอยู่ในตัวแปร o2(i) แต่ทั้งนี้ เนื่องจากการแทนค่า o2(i) หรือ x ในสมการ Arccos ค่า o2(i) หรือ x นี้ต้องไม่เป็น 1 ดังข้อ 13 ดังนั้นต้องทำการกำหมดเงื่อนไขในการรับค่าของ o1(i) ก่อนที่แทนลงในสมการโดยใช้วิธีเดียวกันกับ Arcsin

```
If o2(i) = 1 Or o2(i) = -1 Then
Oc(i) = 0

Else
 Oc(i) = Arccos(((j(0) * j(i)) + (k(0) * k(i))) / (d(0) * d(i))) * 180 / Pi
End If
```

เหตุที่ต้องหามุมทั้ง Arcsin และ Arccos เพราะว่า



รูปที่3.10 อธิบายการคำนวณหา Arcsin และ Arccos

ค่ามุมที่ได้จาก Arcsin จะได้ค่าทั้ง + และ – แต่จะไม่สามารถแยกวัตถุที่อยู่ด้านหน้าหรือหลังได้ เพราะมีมุมเท่ากันจึงต้องใช้ Arccos เข้ามาควบคุม ใส่เงื่อนไขในการแยกมุมออกเป็นช่วงๆ โดยช่วงแรกอยู่ในช่วงระหว่าง ±30° เก็บค่ามุมในช่วง Top โดย

```
If -30° <= Os(i) And Os(i) <= 30° And Oc(i) <= 30° Then
T(i) = Os(i)          ค่ามุมที่อยู่ในช่วง Top
R(i) = 0              ค่ามุมที่อยู่ในช่วง Right
L(i) = 0              ค่ามุมที่อยู่ในช่วง Left
dT(i) = d(i)          ขนาดที่อยู่ในช่วง Top
dR(i) = 0             ขนาดที่อยู่ในช่วง Right
dL(i) = 0             ขนาดที่อยู่ในช่วง Left
If dT(i) < Tmin Then Tmin = dT(i)
```

เก็บค่า Os(i) ใน T(i) เพื่อเก็บค่ามุมในส่วนTop และเก็บค่า d(i) ใน dT(i) เพื่อหาขนาดที่น้อยที่สุด โดยใช้คำสั่ง If dT(i) < Tmin Then Tmin = dT(i) ส่วนค่าอื่นจะเท่ากับศูนย์
ช่วงที่สองอยู่ในช่วงมากกว่า 30° แต่ไม่เกิน 90° เก็บค่ามุมในช่วง Right โดย

```
ElseIf 30° <Os(i)And Os(i)<=90° And Oc(i)<=90° Then
```

```
T(i) = 0
R(i) = Os(i)
L(i) = 0
dT(i) = 0
dR(i) = d(i)
dL(i) = 0
If dR(i) < Rmin Then Rmin = dR(i)
```

เก็บค่า Os(i) ใน R(i) **เพื่อเก็บค่ามุมในส่วน** Right **และเก็บค่า** d(i) ใน dR(i) **เพื่อหาขนาดที่น้อยที่สุด**
โดยใช้คำสั่ง If dR(i) < Rmin Then Rmin = dR(i) **ส่วนค่าอื่นจะเท่ากับศูนย์**
ช่วงที่สามอยู่ในช่วงน้อยกว่า -30° **แต่ไม่เกิน** -90° **เก็บค่ามุมในช่วง** Left **โดย**

```
ElseIf -30 > Os(i) And Os(i) >= -90 And Oc(i) <= 90
T(i) = 0
R(i) = 0
L(i) = Os(i)
dR(i) = 0
dL(i) = d(i)
dT(i) = 0
If dL(i) < Lmin Then Lmin = dL(i)      คำนวณหาขนาดที่น้อยที่สุด
```

เก็บค่า Os(i) ใน L(i) **เพื่อเก็บค่ามุมในส่วน** Right **และเก็บค่า** d(i) ใน dL(i) **เพื่อหาขนาดที่น้อยที่สุด**
โดยใช้คำสั่ง If dL(i) < Lmin Then Lmin = dL(i) ส่วนค่าอื่นจะเท่ากับศูนย์
ช่วงที่สี่ มุมที่อยู่นอกเหนือจากช่วงในสามช่วง ค่ามุมและขนาดในช่วงนี้จะมีค่าเป็น 0 **โดย**

```
Else
R(i) = 0
L(i) = 0
T(i) = 0
dR(i) = 0
dL(i) = 0
dT(i) = 0
End If
```

เมื่อเจอคำสั่ง Next I เครื่องจะเพิ่มค่า i อีก 1 แล้ววนกลับไปเริ่มใหม่ เพื่อหาค่าของวัตถุให้ครบทั้ง 10

แสดงขนาดที่น้อยที่สุดแต่ละช่วงใน TextBox เพื่อตรวจสอบความถูกต้องโดย

```
Text61.Text = Lmin
Text62.Text = Tmin
Text63.Text = Rmin
```

แล้วส่งค่าทั้ง 3 นี้ไปคำนวณในส่วนของ Fuzzy Singleton เพื่อหาค่ามุมที่จะใช้ให้หุ่นยนต์เคลื่อนที่ต่อไป โดย

```
L = Lmin
R = Rmin
F = Tmin
```

ตัวแปร L, R, F นำไปใช้คำนวณในส่วนของ Fuzzy Singleton ซึ่งจะกล่าวในตอนถัดไป เมื่อนำค่าขนาดที่น้อยที่สุดไปคำนวณแล้ว Fuzzy Singleton จะส่งค่ามุมออกมา นำค่ามุมที่ได้ใหม่ไปรวมกับค่ามุมของ Head แล้วคำนวณค่าจุดปลายใหม่ โดย

```
c(1) = c(1) + มุมจาก Fuzzy
```

สั่งให้หุ่นยนต์เดินไปยังทิศที่คำนวณได้ใหม่เป็นระยะทาง 0.02 เซนติเมตร แล้วกลับไปเริ่มการคำนวณใหม่ โดย

```
zO = (90 - c(1)) * Pi / 180
a(1) = a(1) + 0.02 * Cos(zO)
b(1) = b(1) - 0.02 * Sin(zO)
```

เครื่องรับค่า a(1) ใหม่ลงใน a(1) ตัวเดิมซึ่งทำให้ไม่จำเป็นต้องเปลี่ยนตัวแปร เมื่อค่า a(1) และ b(1) เปลี่ยน ก็ทำให้ค่าอื่นๆเปลี่ยนตามไปด้วย

ในส่วนของการเคลื่อนที่เข้าหาเป้าหมาย โดยถ้าส่วนโปรแกรมควบคุมส่งค่า Oout ออกมาเป็น 0 หมายความว่าตัวหุ่นยนต์ไม่จำเป็นต้องเปลี่ยนทิศการเคลื่อนที่ แสดงว่าไม่มีสิ่งกีดขวางที่จำเป็นต้องหลบหลีก ดังนั้นจะทำการหันหน้าหุ่นยนต์เข้าหาเป้าหมายโดยเริ่มต้น
รับค่าจุดศูนย์กลางของจุดหมายแล้วนำมาแสดงผลเป็นวงกลมรัศมี 0.7 เซนติเมตร สีดำ โดยใช้คำสั่ง

```
a(2) = Text95.Text
b(2) = Text96.Text
Picture1.Circle (a(2), b(2)), 0.7, RGB(0, 0, 0)
```

หาค่าเวกเตอร์และขนาดระหว่างจุดศูนย์กลางของหุ่นยนต์กับจุดหมาย โดยใช้วิธี ดังภาผนวก 1

```
tgX = a(2) - a(1)
tgY = b(2) - b(1)
tgD = Sqr(tgX ^ 2 + tgY ^ 2)
```

สร้างแกนอ้างอิงและแปลงเป็นเวกเตอร์ให้กับหุ่นยนต์เพื่อหาค่ามุมที่หุ่นยนต์ต้องหันหน้าไปโดย

```
a(0) = a(1)
b(0) = b(1) - 1
j(0) = 0
k(0) = -1
d(0) = Sqr(j(0) ^ 2 + k(0) ^ 2)
```

หามุมระหว่างหุ่นยนต์กับจุดหมายโดยใช้ Arcsin ได้

```
Os(0) = Arcsin(((j(0) * tgY) - (tgX * k(0))) / (d(0) * tgD)) * 180 / Pi
```

ส่งค่ามุมไปที่ c(1) แล้วทำการรันโปรแกรมตามปกติ ทำให้หุ่นยนต์หันหน้าไปทางทิศที่คำนวนได้ใหม่แล้วเริ่มเคลื่อนที่ไปในทิศทางนั้น

ภาคผนวก 2

โปรแกรมคำนวณเอาต์พุตแบบฟัซซี่ซิงเกิลตัน

โปรแกรมของตัวแปรอินพุต 1 เขียนได้ดังนี้

```
Public Function Fuzzy1(L As Double)
If L >= 0 And L <= 1.5 Then
Lt1 = 1
ElseIf L > 1.5 And L <= 4.5 Then
Lt1 = (L * (1 / (-3))) + 1.5
Else
Lt1 = 0
End If
End Function
Public Function Fuzzy2(L As Double)
If L > 1.5 And L <= 4.5 Then
Lt2 = (L * (1 / 3)) - 0.5
Else
Lt2 = 0
End If
End Function
```

คำอธิบายของโปรแกรม

Public Function Fuzzy1(L As Double)	เป็นการประกาศตัวแปรให้ L As Double บนคำสั่ง Function ให้ชื่อว่า Fuzzy1 โดย L จะเป็นค่าอินพุตที่รับมาจากโปรแกรมส่วนที่ 1 คือระยะของวัตถุที่ห่างออกไปทางซ้าย

If L >= 0 And L <= 1.5 Then
Lt1 = 1
ElseIf L > 1.5 And L <= 2 Then
Lt1 = (L * (-1)) + 2
Else
Lt1 = 0
End If
End Function
	ใช้คำสั่ง IF เพื่อสร้างเงื่อนไข โดยอาศัยสมการเชิงเส้นที่ทำให้เกิดรูปกราฟของตัวแปรอินพุต 1 Lt1 เป็นค่าของฟังก์ชันภาวะสมาชิก SL ซึ่งได้จากการคำนวณตามเงื่อนไขที่กำหนด

Public Function Fuzzy2(L As Double)	เป็นการประกาศตัวแปรให้ L As Double บนคำสั่ง Function ให้ชื่อว่า Fuzzy2 โดย L จะเป็นค่าอินพุตที่รับมาจากโปรแกรมส่วนที่ 1 คือระยะของวัตถุที่ห่างออกไปทางซ้าย

If L > 1.5 And L <= 2 Then
Lt2 = L - 1
Else
Lt2 = 0
End If
End Function
	ใช้คำสั่ง IF เพื่อสร้างเงื่อนไข โดยอาศัยสมการเชิงเส้นที่ทำให้เกิดรูปกราฟของตัวแปรอินพุต 1 Lt2 เป็นค่าของฟังก์ชันภาวะสมาชิก LL ซึ่งได้จากการคำนวณตามเงื่อนไขที่กำหนด

ให้อินพุต 2 มีฟังก์ชันภาวะสมาชิก 2 ฟังก์ชันมีตัวแปรเชิงภาษาดังต่อไปนี้

ระยะวัดฤทางขวาน้อยมากแทนด้วยตัวย่อ SR

ระยะวัดฤทางขวามากมากแทนด้วยตัวย่อ LR

โปรแกรมของตัวแปรอินพุต 2 เขียนได้ดังนี้

```
If R >= 0 And R <= 1.5 Then
Rt1 = 1
ElseIf R > 1.5 And R <= 2 Then
Rt1 = (R * (-1)) + 2
Else
Rt1 = 0
End If
End Function
Public Function Fuzzy4(R As Double)
If R > 1.5 And R <= 2 Then
Rt2 = R - 1
Else
Rt2 = 0
End If
End Function
```

คำอธิบายของโปรแกรม

Public Function Fuzzy3(R As Double)	เป็นการประกาศตัวแปรให้ R As Double บนคำสั่ง Function ให้ชื่อว่า Fuzzy3 โดย R จะเป็นค่าอินพุตที่รับมาจากโปรแกรมส่วนที่ 1 คือระยะของวัตถุที่ห่างออกไปทางขวา

If R >= 0 And R <= 1.5 Then
Rt1 = 1
ElseIf R > 1.5 And R <= 2 Then
Rt1 = (R * (-1)) + 2
Else
Rt1 = 0
End If
End Function
	ใช้คำสั่ง IF เพื่อสร้างเงื่อนไข โดยอาศัยสมการเชิงเส้นที่ทำให้เกิดรูปกราฟของตัวแปรอินพุต 2 Rt1 เป็นค่าของฟังก์ชันภาวะสมาชิก SR ซึ่งได้จากการคำนวณตามเงื่อนไขที่กำหนด

Public Function Fuzzy4(R As Double)	เป็นการประกาศตัวแปรให้ R As Double บนคำสั่ง Function ให้ชื่อว่า Fuzzy4 โดย L จะเป็นค่าอินพุตที่รับมาจากโปรแกรมส่วนที่ 1 คือระยะของวัตถุที่ห่างออกไปทางขวา

If R > 1.5 And R <= 2 Then
Rt2 = R - 1
Else
Rt2 = 0
End If
End Function
	ใช้คำสั่ง IF เพื่อสร้างเงื่อนไข โดยอาศัยสมการเชิงเส้นที่ทำให้เกิดรูปกราฟของตัวแปรอินพุต 2 Rt2 เป็นค่าของฟังก์ชันภาวะสมาชิก LR ซึ่งได้จากการคำนวณตามเงื่อนไขที่กำหนด

ให้อินพุต 2 มีฟังก์ชันภาวะสมาชิก 2 ฟังก์ชันมีตัวแปรเชิงภาษาดังต่อไปนี้

ระยะวัตถุทางด้านหน้าน้อยมากแทนด้วยตัวย่อ SF

ระยะวัตถุทางด้านหน้ามากมากแทนด้วยตัวย่อ LF

โปรแกรมของตัวแปรอินพุต 3 เขียนได้ดังนี้

```
Public Function Fuzzy5(F As Double)
If F >= 0 And F <= 1.5 Then
F1 = 1
ElseIf F > 1.5 And F <= 2 Then
F1 = (F * (-1)) + 2
Else
F1 = 0
End If

End Function
Public Function Fuzzy6(F As Double)
If F > 1.5 And F <= 2 Then
F2 = F - 1
Else
F2 = 0
End If
End Function
```

คำอธิบายของโปรแกรม

Public Function Fuzzy5(F As Double)	เป็นการประกาศตัวแปรให้ F As Double บนคำสั่ง Function ให้ชื่อว่า Fuzzy5 โดย F จะเป็นค่าอินพุตที่รับมาจากโปรแกรมส่วนที่ 1 คือระยะของวัตถุที่ห่างออกไปทางด้านหน้า

Public Function Fuzzy5(F As Double)
If F >= 0 And F <= 1.5 Then
F1 = 1
ElseIf F > 1.5 And F <= 2 Then
F1 = (F * (-1)) + 2
Else
F1 = 0
End If
End Function
	ใช้คำสั่ง IF เพื่อสร้างเงื่อนไข โดยอาศัยสมการเชิงเส้นที่ทำให้เกิดรูปกราฟของตัวแปรอินพุต 3 F1 เป็นค่าของฟังก์ชันภาวะสมาชิก SF ซึ่งได้จากการคำนวณตามเงื่อนไขที่กำหนด

Public Function Fuzzy6(F As Double)	เป็นการประกาศตัวแปรให้ F As Double บนคำสั่ง Function ให้ชื่อว่า Fuzzy6 โดย F จะเป็นค่าอินพุตที่รับมาจากโปรแกรมส่วนที่ 1 คือระยะของวัตถุที่ห่างออกไปทางด้านหน้า

If F > 1.5 And F <= 2 Then
F2 = F - 1
Else
F2 = 0
End If
End Function
	ใช้คำสั่ง IF เพื่อสร้างเงื่อนไข โดยอาศัยสมการเชิงเส้นที่ทำให้เกิดรูปกราฟของตัวแปรอินพุต 3 F2 เป็นค่าของฟังก์ชันภาวะสมาชิก LF ซึ่งได้จากการคำนวณตามเงื่อนไขที่กำหนด

```
Public Function Output1(a As Double, b As Double, c As Double)
x1 = a * b * c * 180
End Function
Public Function Output2(a As Double, b As Double, c As Double)
x2 = a * b * c * 0
End Function
Public Function Output3(a As Double, b As Double, c As Double)
x3 = a * b * c * 60
End Function
Public Function Output4(a As Double, b As Double, c As Double)
x4 = a * b * c * 30
End Function
Public Function Output5(a As Double, b As Double, c As Double)
x5 = a * b * c * (-60)
End Function

Public Function Output6(a As Double, b As Double, c As Double)
x6 = a * b * c * (-30)
End Function
Public Function Output7(a As Double, b As Double, c As Double)
x7 = a * b * c * 30
End Function
Public Function Output8(a As Double, b As Double, c As Double)
x8 = a * b * c * 0
End Function
```

คำอธิบายของโปรแกรม

Public Function Output1(a As Double, b As Double, c As Double)

x1 = a * b * c * 180

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output1 เป็นฟังก์ชันของกฎ 1

If left is SL(ระยะวัตถุอยู่ทางซ้ายน้อยมาก) and Right is SR(ระยะวัตถุอยู่ทางขวาน้อยมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is 180

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output1

Public Function Output2(a As Double, b As Double, c As Double)

x2 = a * b * c * 0

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output2 เป็นฟังก์ชันของกฎ 2

If left is SL (ระยะวัตถุอยู่ทางซ้ายน้อยมาก) and Right is SR(ระยะวัตถุอยู่ทางขวาน้อยมาก) and Font is LF(ระยะวัตถุอยู่ทางด้านหน้ามากมาก) Then Degree is 0

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output2

Public Function Output3(a As Double, b As Double, c As Double)

x3 = a * b * c * 60

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output3 เป็นฟังก์ชันของกฎ 3

If left is SL (ระยะวัตถุอยู่ทางซ้ายน้อยมาก)and Right is LR(ระยะวัตถุอยู่ทางขวามากมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is 60

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output3

Public Function Output4(a As Double, b As Double, c As Double)

x4 = a * b * c * 30

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output4 เป็นฟังก์ชันของกฎ 4

If left is SL (ระยะวัตถุอยู่ทางซ้ายน้อยมาก)and Right is LR(ระยะวัตถุอยู่ทางขวามากมาก) and Font is LF(ระยะวัตถุอยู่ทางด้านหน้ามากมาก) Then Degree is 30

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output4

```
Public Function Output5(a As Double, b As Double, c As Double)
x5 = a * b * c * (-60)
End Function
```

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output5 เป็นฟังก์ชันของกฎ 5

If left is LL (ระยะวัตถุอยู่ทางซ้ายมากมาก) and Right is SR (ระยะวัตถุอยู่ทางขวาน้อยมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is -60

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output5

```
Public Function Output6(a As Double, b As Double, c As Double)
x6 = a * b * c * (-30)
End Function
```

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output6 เป็นฟังก์ชันของกฎ 6

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

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output6

```
Public Function Output7(a As Double, b As Double, c As Double)
x7 = a * b * c * 30
End Function
```

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output7 เป็นฟังก์ชันของกฎ 7

If left is LL (ระยะวัตถุอยู่ทางซ้ายมากมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) and Font is SF(ระยะวัตถุอยู่ทางด้านหน้าน้อยมาก) Then Degree is 30

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output7

```
Public Function Output8(a As Double, b As Double, c As Double)
x8 = a * b * c * 0
End Function
```

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output8 เป็นฟังก์ชันของกฎ 8

If left is LL (ระยะวัตถุอยู่ทางซ้ายน้อยมาก) and Right is LR (ระยะวัตถุอยู่ทางขวามากมาก) and Font is LF(ระยะวัตถุอยู่ทางด้านหน้ามากมาก) Then Degree is 0

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชัน Output8

เอาต์พุตของระบบ

```
Public Function Output9(a As Double, b As Double, c As Double)
y1 = a * b * c
End Function

Public Function Output10(a As Double, b As Double, c As Double)
y2 = a * b * c
End Function

Public Function Output11(a As Double, b As Double, c As Double)
y3 = a * b * c
End Function

Public Function Output12(a As Double, b As Double, c As Double)
y4 = a * b * c
End Function

Public Function Output13(a As Double, b As Double, c As Double)
y5 = a * b * c
End Function
Public Function Output14(a As Double, b As Double, c As Double)
y6 = a * b * c
```

```
End Function

Public Function Output15(a As Double, b As Double, c As Double)
y7 = a * b * c
End Function

Public Function Output16(a As Double, b As Double, c As Double)
y8 = a * b * c
End Function

Public Function Output9(a As Double, b As Double, c As Double)
y1 = a * b * c
End Function
```

คำอธิบายโปรแกรม

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output9 เป็นฟังก์ชันหาค่า μ_j ของกฎ 1

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput9

```
Public Function Output10(a As Double, b As Double, c As Double)
y2 = a * b * c
End Function
```

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output10 เป็นฟังก์ชันหาค่า μ_j ของกฎ 2

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput10

```
Public Function Output11(a As Double, b As Double, c As Double)
y3 = a * b * c
End Function
```

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output11 เป็นฟังก์ชันหาค่า μ_j ของกฎ 3

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput11

Public Function Output12(a As Double, b As Double, c As Double)

y4 = a * b * c

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output12

เป็นฟังก์ชันหาค่า μ_j ของกฎ 4

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput12

Public Function Output13(a As Double, b As Double, c As Double)

y5 = a * b * c

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output13

เป็นฟังก์ชันหาค่า μ_j ของกฎ 5

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput13

Public Function Output14(a As Double, b As Double, c As Double)

y6 = a * b * c

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output14

เป็นฟังก์ชันหาค่า μ_j ของกฎ 6

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput14

Public Function Output15(a As Double, b As Double, c As Double)

y7 = a * b * c

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output15

เป็นฟังก์ชันหาค่า μ_j ของกฎ 7

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput15

Public Function Output16(a As Double, b As Double, c As Double)

y8 = a * b * c

End Function

ประกาศตัวแปร a As Double, b As Doubleและ c As Double บนคำสั่ง Function ให้ชื่อว่า Output16 เป็นฟังก์ชันหาค่า μ_j ของกฎ 8

โดย a, bและ c รับค่าอะไรก็ได้ที่ป้อนให้เมื่อมีการเรียกใช้ฟังก์ชันOutput16

เมื่อทุกๆ $\mu_j w_j$ และ μ_j ของแต่ละกฎถูกคำนวณและเก็บไว้ในตัวแปรที่กำหนดไว้ โปรแกรมส่วนสุดท้าย สำหรับการควบคุมหุ่นยนต์เคลื่อนที่โดยหลักการฟัซซี่ซิงเกิลตันคือโปรแกรมการหาเอาต์พุต

```
Public Function SUM1()
x = x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8
End Function

Public Function SUM2()
y = y1 + y2 + y3 + y4 + y5 + y6 + y7 + y8
End Function

Public Function SUM()
Z = x / y
End Function
```

คำอธิบายโปรแกรม

```
Public Function SUM1()
x = x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8
End Function
```

ใช้คำสั่ง Function เพื่อสร้างฟังก์ชันสำหรับคำนวณหาผลบวกของ $\mu_j w_j$ ทั้งหมด ให้ชื่อฟังก์ชันว่า SUM1 เก็บค่าที่ได้จากการคำนวณไว้ในตัวแปร x

```
Public Function SUM2()
y = y1 + y2 + y3 + y4 + y5 + y6 + y7 + y8
End Function
```

ใช้คำสั่ง Function เพื่อสร้างฟังก์ชันสำหรับคำนวณหาผลบวกของ μ_j ทั้งหมด ให้ชื่อฟังก์ชันว่า SUM2 เก็บค่าที่ได้จากการคำนวณไว้ในตัวแปร y

```
Public Function SUM()
Z = x / y
End Function
```

ใช้คำสั่ง Function เพื่อสร้างฟังก์ชันสำหรับคำนวณหาเอาต์พุต ตามสมการ2.47 ให้ชื่อฟังก์ชันว่า SUM เก็บค่าเอาต์พุตไว้ในตัวแปร Z

เขียนคำสั่งเพื่อประกาศตัวแปรและเรียกฟังก์ชันได้ดังนี้

```
Dim Lt1 As Double                ใช้คำสั่ง Dim เพื่อประกาศตัวแปร General
Dim Lt2 As Double
Dim Rt1 As Double
Dim Rt2 As Double
Dim F1 As Double
Dim F2 As Double

Private Sub Command1_Click()     ตั้งแต่บรรทัดนี้จะเป็นการเขียนโปรแกรมใน Command1

Dim L As Double                  ใช้คำสั่ง Dim เพื่อประกาศตัวแปรใน Command1
Dim R As Double
Dim F As Double

L = Text1.Text                   Text1 รับค่า L คือระยะวัตถุห่างออกไปทางซ้าย
R = Text2.Text                   Text2 รับค่า R คือระยะวัตถุห่างออกไปทางขวา
F = Text3.Text                   Text3 รับค่า F คือระยะวัตถุห่างออกไปทางด้านหน้า
If L > 4.5 Then
L = 4.5                          สร้างเงื่อนไขขึ้นเพื่อให้ทราบว่าระยะมากที่สุดที่ห่างออกไป
End If                           ทางซ้ายแล้วจะมีผลต่อการตัดสินใจในการหลบหลีกสิ่งกีด
                                 ขวางคือ 4.5 cm

If R > 4.5 Then
R = 4.5                          สร้างเงื่อนไขขึ้นเพื่อให้ทราบว่าระยะมากที่สุดที่ห่างออกไป
End If                           ทางขวาแล้วจะมีผลต่อการตัดสินใจในการหลบหลีกสิ่งกีด
                                 ขวางคือ 4.5 cm

If F > 4.5 Then
F = 4.5                          สร้างเงื่อนไขขึ้นเพื่อให้ทราบว่าระยะมากที่สุดที่ห่างออกไป
End If                           ทางด้านหน้าแล้วจะมีผลต่อการตัดสินใจในการหลบหลีกสิ่ง
                                 กีดขวางคือ 4.5 cm

Fuzzy1 (L)
                                 เป็นการเรียกฟังก์ชันเพื่อหาค่าของฟังก์ชันก์ภาวะสมาชิก SL
```

Fuzzy2 (L)	เป็นการเรียกฟังก์ชันเพื่อหาค่าของฟังก์ชันก์ภาวะสมาชิก LL
Fuzzy3 (R)	เป็นการเรียกฟังก์ชันเพื่อหาค่าของฟังก์ชันก์ภาวะสมาชิก SR
Fuzzy4 (R)	เป็นการเรียกฟังก์ชันเพื่อหาค่าของฟังก์ชันก์ภาวะสมาชิก LR
Fuzzy5 (F)	เป็นการเรียกฟังก์ชันเพื่อหาค่าของฟังก์ชันก์ภาวะสมาชิก SF
Fuzzy6 (F)	เป็นการเรียกฟังก์ชันเพื่อหาค่าของฟังก์ชันก์ภาวะสมาชิก LF
Output1 Lt1, Rt1, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 1
Output2 Lt1, Rt1, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 2
Output3 Lt1, Rt2, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 3
Output4 Lt1, Rt2, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 4
Output5 Lt2, Rt1, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 5
Output6 Lt2, Rt1, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 6
Output7 Lt2, Rt2, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 7
Output8 Lt2, Rt2, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 8
Output9 Lt1, Rt1, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 1
Output10 Lt1, Rt1, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 2
Output11 Lt1, Rt2, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 3

Output12 Lt1, Rt2, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 4
Output13 Lt2, Rt1, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 5
Output14 Lt2, Rt1, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 6
Output15 Lt2, Rt2, F1	แสดงให้ทราบว่ามีการเข้าถึงกฎ 7
Output16 Lt2, Rt2, F2	แสดงให้ทราบว่ามีการเข้าถึงกฎ 8
SUM1	เรียกฟังก์ชัน SUM1 ให้ทำงานเพื่อหาผลรวมของ $\mu_j w_j$
SUM2	เรียกฟังก์ชัน SUM2 ให้ทำงานเพื่อหาผลรวมของ μ_j
SUM	เรียกฟังก์ชัน SUM ให้ทำงานเพื่อหาเอาต์พุต
End Sub	จบการทำงานของ Command1

เอกสารอ้างอิง

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

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

[3] J.Favilla, A. Machion, F.Gomide. **Fuzzy Traffic Control: Adaptive Strategy.** 2nd. San Francisco : IEEE, Int. 1993

[4] M.Nakatsuyama, H.Nagahashi, N.Nishizara. **Fuzzy Logic Controller For a Traffic Junction in The One – Way Arterial Road.** 9th. Budapest : IFAC – World Congress. 1984

[5] R.L.Kelsey, K.R.Bisset. **A Simulation Environment For Fuzzy Control of Traffic Systems.** 12th. Sydney : IFAC-World Congress. 1993

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวทัดดาว ทับอินทร์

ภูมิลำเนา 63/1 ม.15 ต.เกยไชย อ.ชุมแสง จ.นครสวรรค์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนชุมแสงชนูทิศ
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : thuddoaw@hotmail.com

ชื่อ นางสาวธนพร คงพันธุ์

ภูมิลำเนา 3/46 หมู่บ้านเชียงใหม่เทรดเซนเตอร์ ถ.สามล้าน ต.พระสิงห์ อ.เมือง จ.เชียงใหม่

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนวัฒโนทัยพายัพ
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : tan_angle@hotmail.com

ชื่อ นายวิริยะ บุญสนอง

ภูมิลำเนา 79/2 ม.5 ต.คลองตาล อ.ศรีสำโรง จ.สุโขทัย

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนสุโขทัยวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : Demonic@pop.co.th