สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปัญญาประดิษฐ์และการเลียนแบบสำหรับหุ่นยนต์เตะฟุตบอล

ARTIFICIAL INTELLIGENCE AND SIMULATION FOR ROBOCUP SMALL SIZE

T061440

โดย

นายสุรเดช อินทกรณ์

นายองครักษ์ อังคะรุด

เลขหมู่.....
เลขทะเบียน 61440
วัน,เดือน,ปี 17 ก.ค. 2549

b 11596338
ส.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARTIFICIAL INTELLIGENCE AND SIMULATION FOR ROBOCUP SMALL SIZE

By

MR.SURADEJ INTAGORN

MR.ONGKARAK UNGKARUD

A THESIS SUBMITTED IN PARTIAL FULLFILMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ปัญญาประดิษฐ์และการเลียนแบบสำหรับหุ่นยนต์เตะฟุตบอล
ชื่อนักศึกษา	นายสุรเดช อินทกรณ์ รหัสนักศึกษา 44010557
	นายองครักษ์ อังคะรุด รหัสนักศึกษา 44010572
อาจารย์ที่ปรึกษา	รศ.ดร.ปีติเขต สู้รักษา
	อ.บุญย์ชนะ ภู่ระหงษ์
ระดับการศึกษา	ปริญญาตรีวิศวกรรมศาสตรบัณฑิต
	สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2547

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(รศ.ดรปีติเขต สู้รักษา)

.....

(อ.บุญย์ชนะ ภู่ระหงษ์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	ปัญญาประดิษฐ์และการเลียนแบบสำหรับหุ่นยนต์เตะฟุตบอล
จัดทำโดย	นายสุรเดช อินทกรณ์ นายองครักษ์ อังคะรุด
อาจารย์ที่ปรึกษา	รศ.ดร.ปิติเขต สู้รักษา อ.บุญย์ชนะ ภู่ระหงษ์
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ในปัจจุบันการจัดการแข่งขันหุ่นยนต์ต่างๆ นั้นเริ่มเป็นที่จับตามองของผู้คนเพิ่มมากขึ้น ซึ่งการแข่งขันหุ่นยนต์แตะฟุตบอล(Soccer robot) ก็เป็นอีกการแข่งขันหนึ่งที่ได้รับความนิยมทั้งในและนอกประเทศ และโครงงานนี้ก็ได้รับแรงบันดาลใจจากการแข่งขั้นนี้เช่นกันซึ่งการทำหุ่นยนต์แตะฟุตบอลนี้ ต้องมีกระบวนการต่างๆ ทางวิศวกรรมมากมาย ที่ต้องนำมาใช้ร่วมกัน ไม่ว่าจะเป็นกระบวนการประมวลผลภาพ (Image Processing) หลักการด้านการต่อวงจรไฟฟ้า และปัญญาประดิษฐ์(Artificial Intelligence)

สำหรับโครงงานนี้จะทำเกี่ยวกับปัญญาประดิษฐ์โดยที่ปัญญาประดิษฐ์จะเป็นการรับอินพุตจากกระบวนการประมวลผลภาพมาแล้วนำไปตัดสินใจตามอัลกอริธึมที่ดีที่สุดจากนั้นจึงส่งคำสั่งให้แก่หุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Image Processing and Robot for Robocup small size
Student	Mr.SURADEJ INTAGORN ID.44010557
	Mr.ONGKARAK UNGKARUD ID.44010572
Advisor	Assoc.Prof.Dr.PITIKHATE SOORAKSA
	Mr.BOONCHANA PHURAHONG
Graduate level	Bachelor Degree of Information Engineering
Department	Information Engineering
Academic	2004

Abstract

At present,many people keep an eye on robot contest and the Soccer Robot is the favourites robot contest in Thailand and foreign country. This project has an idea from this contest too. The way for making Soccer Robot must have join several engineering process i.e. Image processing,circuit and Artificial Intelligence

This project presents Artificial Intelligence . Artificial Intelligence gets input from Image processing then determines to select the best algorithm and sends command to the robots

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จไปได้ หากไม่ได้รับความช่วยเหลือและความร่วมมือจากหลายๆฝ่ายด้วยกัน เริ่มจากอาจารย์ที่ปรึกษาปริญญานิพนธ์ซึ่งได้แก่ รศ.ดรปิติเขต สู้รักษา และอาจารย์บุญย์ชนะ ภู่ระหงษ์ ที่คอยให้ความช่วยเหลือในด้านต่างๆอย่างดีเยี่ยมมาโดยตลอดระยะเวลาที่ทำปริญญานิพนธ์นี้ จนกระทั่งโครงงานนี้ผ่านไปได้ด้วยดี

ขอขอบคุณ พี่ๆในห้อง Infodynamic laboratory ที่ให้คำแนะนำต่างๆและให้ความช่วยเหลือเป็นอย่างดี และขอขอบคุณคณาจารย์และเพื่อนๆในภาควิชาวิศวกรรมสารสนเทศทุกคนที่ให้คำติชม และคอยช่วยเหลือเสมอมา

สุดท้ายนี้คณะผู้จัดทำต้องขอกราบขอบพระคุณ บิดา มารดา บุคคลที่มีความสำคัญที่สุดในชีวิตที่ทำให้มีทุกวันนี้ คอยให้กำลังใจ ช่วยเหลือ และสนับสนุนในทุกๆด้าน จึงขอขอบพระคุณมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
สารบัญ	ง
สารบัญรูปภาพ	ฉ
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่ได้รับจากโครงงาน	3
บทที่ 2 ทฤษฎีและหลักการที่ใช้ในโครงงาน	4
2.1Artificial Intelligence	4
2.2Turing Test	4
2.3Application of AI	4
2.4Search Techniques in AI	4
2.5รูปแบบการคำนวณสำหรับกิ่งการตัดสินใจ	5
2.5.1การค้นหาทางลึกก่อน (depth first search)	6
2.5.2การค้นหาทางกว้างก่อน (breadth first search)	6
2.6ปัญหาทางปัญญาประดิษฐ์	7
2.7ยุทธวิธีในการแก้ปัญหา	8
2.8สรุปเปรียบเทียบระหว่างการค้นหาในแนวลึก กับในแนวกว้าง	16
บทที่3 โปรแกรมที่ใช้ในโครงงาน	
3.1 รายละเอียดของโปรแกรมที่ใช้พัฒนาและที่มา	17
3.2 ขั้นตอนการออกแบบโปรแกรม	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.2.1 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งผู้รักษาประตู	20
3.2.2 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งกองหลัง	22
3.2.3 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งกองกลางตัวรับ	22
3.2.4 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งกองกลางและกองหน้า	27
3.3 ปัญหาที่เกิดขึ้นระหว่างขั้นตอนการสร้างโปรแกรม	32
3.4 รายละเอียดและการใช้งานของโปรแกรม	33
บทที่ 4 วิธีการดำเนินการทดลอง	34
4.1อธิบายตำแหน่งต่างๆของ ROBOT ในสนาม	35
4.2 ขั้นตอนการดำเนินการทดลอง	35
4.3การทดลองด้าน simulator	42
4.4ผลนั้นมีความแตกต่างกันดังนี้	42
บทที่5 สรุปผลการทดลอง	43
5.1 สรุปผลการดำเนินการทดลอง	43
5.2 ปัญหาที่พบในระหว่างการทดลอง	43
5.3 แนงทางการพัฒนาโครงงาน	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

เรื่อง	หน้า
รูปที่ 3.1 ภาพแสดงระบบทั้งหมดของหุ่นยนต์เตะฟุตบอล	18
รูปที่ 3.2 ภาพแสดงโปรโตคอลอินพุตของปัญญาประดิษฐ์	19
รูปที่ 3.2.1 ภาพแสดงการเคลื่อนที่ผู้รักษาประตูแนวอัลกอริธึมที่ 1	20
รูปที่ 3.2.2 ภาพแสดงการเคลื่อนที่ผู้รักษาประตูแนวอัลกอริธึมที่ 2	21
รูปที่ 3.2.3 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 1 ขั้นตอนการหมุนสู่เส้นตรงที่ระยะทางสั้นที่สุด	22
รูปที่ 3.2.4 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 1 ขั้นตอนการเดินสู่เส้นตรงที่ระยะทางสั้นที่สุด	23
รูปที่ 3.2.5 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 1 ขั้นตอนการฟีดบอลเข้าแนวที่ต้องการ	24
รูปที่ 3.2.6 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 2 ขั้นตอนการเดินไปสู่จุดใต้บอล	25
รูปที่ 3.2.7 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 2 ขั้นตอนการเดินไปสู่จุดใต้บอล	26
รูปที่ 3.2.8 ภาพแสดงการเคลื่อนที่MC ขั้นตอนการคำนวนแนววิถีโจมตี	27
รูปที่ 3.2.9 ภาพแสดงผังงานของอัลกอริธึมแนวรุก	28
รูปที่ 3.2.10 ภาพแสดงผังงานของอัลกอริธึมแนวรับ	29
รูปที่ 3.2.11 ภาพแสดงแนวแรงของล้อทั้ง3ของหุ่นยนต	30
รูปที่ 3.4.1 แสดงโปรแกรมส่วน AI	32
รูปที่ 4.1.1 แสดงตำแหน่งต่างๆของ ROBOT ในสนาม	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

เนื่องจากปัจจุบันนี้ได้มีการนำระบบปัญญาประดิษฐ์ (Artificial Intelligence) มาใช้กับระบบงานต่างๆมากมาย เช่นการควบคุมอัตโนมัติต่างๆที่จะต้องให้หลายๆส่วนทำงานพร้อมกัน เนื่องจากมนุษย์นั้นมีขีดจำกัดในงานบางประเภทเช่นการจะต้องควบคุมเครื่องจักรกลหลายตัวพร้อมกันโดยให้ทำงานให้เข้าจังหวะกัน หรือแม้แต่ระบบผู้เชี่ยวชาญ(Expert System)ก็มีการนำระบบปัญญาประดิษฐ์ (Artificial Intelligence) มาใช้เช่นกันเช่นการนำข้อมูลจากฐานข้อมูลออกมาประมวลผลเพื่อสามารถให้คำแนะนำแก่บุคคลในสายงานนั้นๆอย่างคร่าวๆ โดยจะสามารถลดต้นทุนการจ้างผู้เชี่ยวชาญที่เป็นมนุษย์ได้ส่วนหนึ่ง

สำหรับโครงงานนี้นั้นจะเป็นการนำปัญญาประดิษฐ์ (Artificial Intelligence)มาใช้ควบคุมหุ่นยนต์เตะฟุตบอล(Soccer robot) 5 ตัวให้สามารถเล่นเป็นทีมร่วมกันก็จะพบปัญหาคือถ้าในขณะนั้นทีมยังสร้าง หุ่นยนต์เตะฟุตบอล(Soccer robot)ยังไม่เสร็จแล้วจะทำให้ไม่สามารถทดลองอะไรได้เลยทางผู้จัดทำจึงได้ทำการเพิ่มหัวข้อที่จะศึกษาเข้าไปอีก 1 หัวข้อคือ ตัวจำลองสถานการณ์ (Simulator) ซึ่งจะเป็นการสร้างสถานการณ์เสมือนเพื่อให้สามารถเห็นผลคร่าวๆของโปรแกรมปัญญาประดิษฐ์ (Artificial Intelligence) แม้ว่าความเร็วต่างๆอาจจะต่างไปแต่จะไม่อยู่ในขอบเขตปัญหาของเรา ขอบเขตปัญหาของเราจะมีเพียงแค่สามารถแสดงผลรูปแบบการเคลื่อนที่ของโปรแกรม ปัญญาประดิษฐ์ (Artificial Intelligence) พอโดยไม่สนความเร็วที่เกิดขึ้น

ปัญญาประดิษฐ์ (Artificial Intelligence) โดยนิยามแล้วจะกว้างมากแต่โดยทั่วไปจะเป็นการทำให้คอมพิวเตอร์สามารถเรียนรู้การแก้ปัญหาได้แต่ในโครงงานนี้ผู้จัดทำจะขอยกคำนิยามของ Alan Turing โดยเขาได้กล่าวว่า "A machine has artificial intelligence when there is no discernible difference between the conversation generated by the machine and that of an intelligent person." ซึ่งผู้จัดทำจะขอถอดความว่า การที่เครื่องจักรจะมีความเป็นปัญญาประดิษฐ์เมื่อไม่สามารถสังเกตุได้ถึงความแตกต่างของคำตอบที่ถูกสร้างระหว่างเครื่องจักรนั้นกับบุคคลที่เราถือว่าฉลาด หมายเหตุ conversation นั้นโดยความหมายแปลว่าการสนทนาแต่ในที่นี้น่าจะหมายถึงการโต้ตอบ ซึ่งผู้จัดทำได้เลือใช้คำว่าคำตอบแทน

ตัวจำลองสถานการณ์(Simulator) เป็นโปรแกรมที่จะจำลองสถานการณ์ที่เราต้องการขึ้นโดยหลักการคร่าวๆคือจะต้องเลียนแบบสถานการณ์จริงให้ได้ใกล้เคียงที่สุด คำว่าเลียนแบบให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ได้ใกล้เคียงที่สุดนั้นในที่นี้จะขอนิยามว่าเป็นการให้ outputของตัวจำลองสถานการณ์ได้ใกล้เคียงกับoutput ของจริงมากที่สุดเมื่อ input ที่ได้เหมือนกัน

1.2 วัตถุประสงค์

- หารูปแบบของ ปัญญาประดิษฐ์ (Artificial Intelligence) ให้มีความฉลาดมากที่สุด
- เพื่อเป็นประโยชน์ต่อการนำไปใช้การพัฒนางานด้านอื่นๆ เช่น ระบบผู้เชี่ยวชาญ
- เพื่อสามารถจำลองระบบหุ่นยนต์เตะฟุตบอล(Soccer robot)

1.3 ขอบเขตของโครงงาน

เป็นการทดลองเกี่ยวกับการหารูปแบบของ ปัญญาประดิษฐ์ (Artificial Intelligence) ซึ่งมีจุดมุ่งหมายเพื่อที่จะหาแนวทางในการเล่นที่ดีที่สุด แต่เนื่องจากผลการแข่งขันนั้นจะต้องขึ้นกับคู่แข่งขันด้วยเพราะฉะนั้นเราจึงจะกำหนดขอบเขตปัญหาเป็น 2 ส่วนโดยส่วนแรกจะเป็นการหารูปแบบปัญญาประดิษฐ์ (Artificial Intelligence) ที่สามารถทำประตูได้มากที่สุดจะให้มีการวางสิ่งกีดขวางไว้แล้วให้หุ่นยนต์ตัดสินใจในการพาลูกฟุตบอลไปทำประตู ส่วนที่ 2 นั้นจะเป็นการหารูปแบบปัญญาประดิษฐ์ (Artificial Intelligence) ที่สามารถป้องกันประตูได้โดยให้เสียประตูน้อยที่สุดโดยรูปแบการทดลองนั้นจะเป็นการให้มนุษย์ทดลงโดยการโยนลูกเข้าไปแล้วให้หุ่นยนต์ป้องกันประตู

ในส่วนของ ตัวจำลองสถานการณ์(Simulator)นั้นขอบเขตปัญหาของเราจะมีเพียงแค่สามารถแสดงผลรูปแบบการเคลื่อนที่ของ โปรแกรม ปัญญาประดิษฐ์ (Artificial Intelligence) พอโดยไม่สนความเร็วที่เกิดขึ้นหรืออีกนัยหนึ่งคือ ตัวจำลองสถานการณ์(Simulator) นั้นจะทำหน้าที่เหมือนกับตัวแสดง output ของโปรแกรม ปัญญาประดิษฐ์ (Artificial Intelligence)

1.4 ขั้นตอนการดำเนินงาน

- ศึกษาทฤษฎีของปัญญาประดิษฐ์ (Artificial Intelligence)
- ศึกษาทฤษฎีของการจำลองสถานการณ์(Simulation)
- สร้างโปรแกรมประยุกต์ขึ้นมาเพื่อใช้ทำการทดลองทฤษฎีและหลักการที่ศึกษาไว้
- ทำการทดลองระหว่างโปรแกรมปัญญาประดิษฐ์ (Artificial Intelligence) ที่สร้างไว้กับโปรแกรมตัวจำลองสถานการณ์(Simulator)
- ทำการทดลองระหว่างโปรแกรมปัญญาประดิษฐ์ (Artificial Intelligence) ที่สร้างไว้กับหุ่นยนต์ของจริงเนื่องจากจะต้องทำการปรับระยะและความเร็วให้เข้ากับของจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สรุปผลการทดลองและนำผลที่ได้ไปวิเคราะห์เพื่อหาวิธีแก้ไขข้อผิดพลาดจากโปรแกรมที่ได้สร้างไว้

1.5 ประโยชน์ที่ได้รับจากโครงงาน

จากโครงงานที่ได้ทำการทดลองนี้สามารถนำไปควบคุมหุ่นยนต์ในทีมทั้ง 5 ตัว ซึ่งจะซึ่งจะสามารถนำเอาประโยชน์จากโครงงานนี้ไปประยุกต์ใช้เกี่ยวกับการระบบ control ที่ต้องการความฉลาด และยังสามารถนำไปเป็นแนวทางในการสร้างระบบผู้เชี่ยวชาญ(Expert System) นอกจากนี้ในส่วนของ Simulator นั้นจะมีประโยชน์สำหรับผู้ที่จะมาพัฒนาต่อของระบบนี้เนื่องจากว่าจะสามารถลองอัลกอริธึมอย่างหยาบได้โดยไม่ต้องไปเสียเวลาตั้งระบบที่จะทำการทดลองกับของจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่ใช้ในโครงงาน

2.1Artificial Intelligence

Artificial Intelligence(AI) คือ การทำอย่างไรที่ทำให้ เครื่องคอมพิวเตอร์ สามารถเรียนรู้ได้เอง แต่ไม่ว่าอย่างไร คนก็ดีที่สุด

2.2Turing Test

1. เราจะรู้ได้อย่างไรว่า เครื่องสามารถคิดเองได้ Alan Turing เสนอวิธี Turing Test
2. ใช้คน 2 คน และเครื่อง 1 เครื่อง
3 .คนแรกเป็นผู้ตั้งคำถามโดยใช้คียบอร์ดและจอภาพ ซึ่งผู้ถามจะอยู่ห้องแยกจากเครื่องและคนที่สอง
4. ผู้ถามต้องแยกให้ได้ว่าคำตอบที่มาจากอีกห้องหนึ่ง นั้นเป็นของใครถ้าแยกได้แสดงว่าเครื่องนั้นไม่ใช่ปัญญาประดิษฐ์แต่ถ้าแยกไม่ได้แสดงว่าเครื่องนั้นมีความเป็นปัญญาประดิษฐ์

2.3Application of AI

1. Intelligent Retrival from Database
2.Expert System
3. Robotics
4. Automatic Programming - เป็นลักษณะ Input/Output ใช้ AI ในการสร้างโปรแกรม
5. Perception Problems การมองเห็น ใช้ Image Processing หาเส้นขอบ

Application of AI มีมากมายแต่ในการทำโปรเจคของกลุ่มผมนั้นเกี่ยวกับ Soccer Robot จึงทำการศึกษาในส่วนของ Robotics และ Image Processing แต่ในส่วนของ Image Processing นั้นจะเป็นการทำร่วมกันกับกลุ่มอื่นเพราะโปรเจคนั้นสอดคล้องกัน

2.4Search Techniques in AI

Exhausive Search

Breadth-First Search

Partial Seach

Depth-First Search

Search

Simulated Annealing

Heuristic Search

Tabu Search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า Search Techniques in AI นั้นมีมากมายแต่ในการทำโปรเจคของกลุมผมนั้นได้เลือกวิธีการ Breadth-First Search

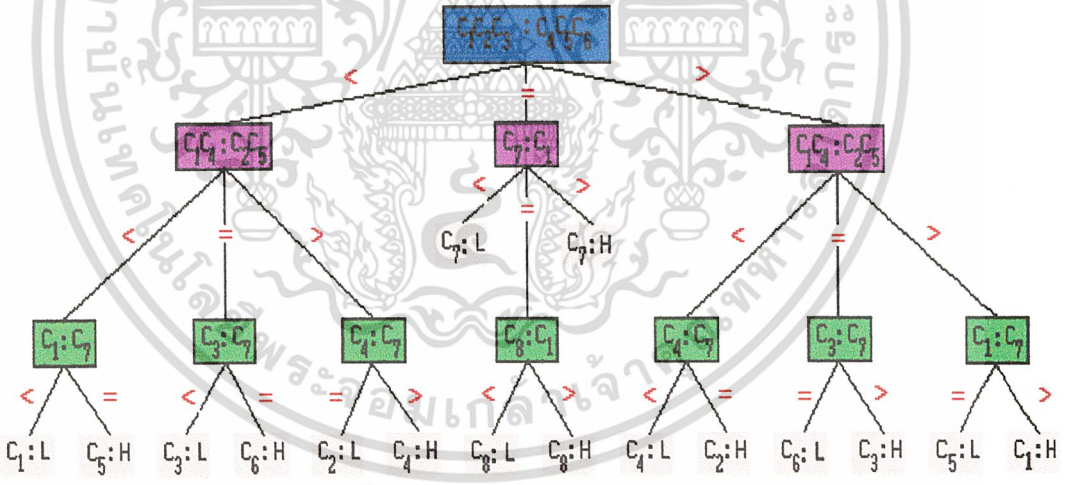
ในการอธิบายนั้นจะมีตัวอย่างและการเปรียนบเทียบระหว่าง Breadth-First Search กับ Depth-First Search เพื่อให้เกิดความเข้าใจมากยิ่งขึ้น

2.5รูปแบบการคำนวณสำหรับกิ่งการตัดสินใจ

การสร้างกิ่งการตัดสินใจ เป็นการนิยามขอบเขตของปัญหา หรือการสร้างกิ่งที่เป็นไปได้ แต่ทางเลือกในการคำนวณหรือเส้นทางที่จะคิดคำนวณคงต้องหาทางตามสภาพความเหมาะสมของปัญหา

ปัญหาเหรียญ 8 เหรียญ เหรียญ 8 เหรียญ มีลักษณะภายนอกเหมือนกัน แต่ในจำนวนนี้มีเหรียญหนึ่ง ที่มีความแตกต่างในเรื่องน้ำหนัก ซึ่งอาจจะหนักกว่าหรือเบากว่า โดยที่ถ้าดูจากภายนอกจะไม่รู้ ในการตรวจสอบจะใช้ตาชั่ง การหาเหรียญแปลกนี้คงต้องใช้ตาชั่ง และ สร้างปัญหานี้ให้เป็นกิ่งการตัดสินใจ

ในที่นี้สมมุติ มีเหรียญ 8 เหรียญ ที่ใช้ $C_1\ C_2 \ldots C_8$ แทนเหรียญทั้งแปด การเปรียบเทียบ คงทำได้ดังนี้



การสร้างกิ่งการตัดสินใจนี้ เป็นหนทางของการสร้างระบบเพื่อแก้ปัญหาและหาทางที่จะเข้าถึงคำตอบให้ได้เร็วและดีที่สุด

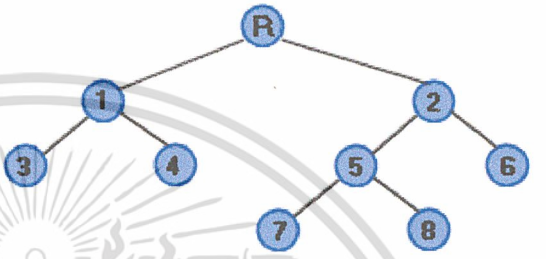
การเข้าถึงจากจุดยอดของโครงสร้างต้นไม้ (tree) นี้ มีวิธีการได้หลายทาง หนทางการเข้าถึงโดยเริ่มจากโหนดยอดสุด ไปยังเป้าหมายทาง เราเรียกขบวนการนี้ว่า การท่องไปในกิ่งต้นไม้ ซึ่งเทียบได้กับการค้นหา (research) เพื่อหาคำตอบ แน่นอนว่าขอบเขตของต้นไม้จึงเป็นขอบเขตของปัญหาที่เป็นไปได้ ซึ่งเราเรียกว่า problem space

การค้นหาใน problem space ทำได้หลายวิธี ซึ่งได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

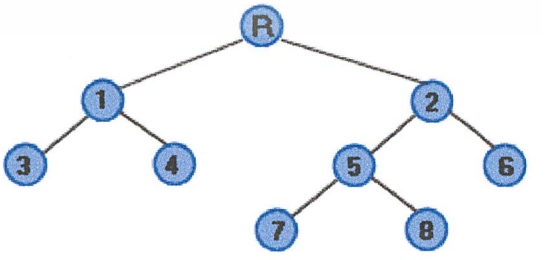
2.5.1การค้นหาทางลึกก่อน (depth first search)

ลักษณะการหาคำตอบในกิ่งของต้นไม้ โดยการคำนวณค่าหรือ ตรวจสอบค่าตามโหนดต่างๆ โดยเน้นทางลงระดับล่างก่อน การค้นหาทางลึก ใช้หลักการคือ เราจะลงไปคำนวณทางกิ่งที่อยู่ทางซ้ายสุดก่อน ถ้าสุดเส้นทางไปได้จึงจุดถอยกลับขึ้นบน และลงไปยังกิ่งที่ยังไม่เคยคำนวณโดยทำไล่จากทางซ้ายไป เริ่มจากรูท หรือ R ซึ่งเป็นจุด แรกของต้นไม้ การค้นหาทางลึก จะลงไปทางกิ่งทางซ้ายก่อน นั่นคือ ลงไปที่ 1, 3, 4, 2, 5, 7, 8, 6



2.5.2การค้นหาทางกว้างก่อน (breadth first search)

จากแนวความคิด <u>การค้นหาทางลึกก่อน</u> เน้นการคำนวณลงไปในแนวดิ่ง คือ ลงไปให้ถึงโหนดสุดท้ายทางแนวดิ่ง ที่อยู่ทางซ้ายมือก่อน แต่การคำนวณแบบการค้นหาทางกว้างก่อน จะลงไปคำนวณที่ความลึกทีละระดับ จากระดับแรกที่รูทของต้นไม้ ลงไประดับความลึกหนึ่ง และลงไปในระดับความลึกเป็นชั้นๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณตามระดับความกว้างก่อน
จะเริ่มจากรูทและลงไปคำนวณระดับ
ล่างคือ โหนด 1, 2 จากนั้นจะลงไปที่ 3,
4, 5, 6, และ 7, 8 จนครบทั้งขอบเขต
ของปัญหาทั้งหมดที่จะเป็นไปได้

ตัวอย่างของวิธีการแก้ปัญหาโดย **breadth first search** และ **depth first search** และมีการเปรียบเทียบ

2.6ปัญหาทางปัญญาประดิษฐ์

1. เกม OX (Tic-Tac-Toe): กำหนดกระดานสี่เหลี่ยมขนาด 3 คูณ 3 ที่มีทั้งหมด 9 ช่องโดยที่สี่เหลี่ยมแต่ละช่องสามารถเขียนตัวสัญลักษณ์ O หรือ X ลงไป
   - มีผู้เล่นอยู่สองคนต้องการเล่นเกมนี้โดยผลัดกันเขียนสัญลักษณ์ O สำหรับผู้เล่นคนแรก และสัญลักษณ์ X สำหรับผู้เล่นคนที่สอง
   - การเขียนสัญลักษณ์บนช่องใด ๆ ช่องนั้นจะต้องไม่มีสัญลักษณ์ใดปรากฏอยู่
   - เกมจะหยุดลงเมื่อผู้เล่นคนใดคนหนึ่งสามารถเขียนเครื่องหมายที่เหมือนกันเรียงกัน 3 ตัวในแนวดิ่ง แนวนอนหรือแนวทแยง

   จงหาแผนการที่ผู้เล่นแต่ละคนควรเล่นเพื่อเอาชนะผู้เล่นอีกคนหนึ่ง

1	2	3
4	5	6
7	8	9

2. เกมการตวงน้ำ (A Water jug problem): เราได้รับถังสองใบ ใบแรกมีความจุ 4 แกลลอน และใบที่สองมีความจุ 3 แกลลอน โดยที่ถังทั้งสองใบไม่มีรอยตวงให้สามารถวัดจำนวนปริมาณน้ำได้ เรายังมีปั๊มน้ำที่สามารถเติมน้ำในแต่ละถังให้เต็มได้ จงหาว่าเราจะตวงน้ำ 2 แกลลอนใส่ในถังขนาด 4 แกลลอนอย่างไรโดยไม่ใช้การประมาณ
3. เกมนักแสวงบุญและมนุษย์กินคน (The Missionaries and Cannibals Problem): มีมิสชันนารีสามคน และมนุษย์กินคนสามคนอยู่บนฝั่งแม่น้ำสายหนึ่ง เขาทั้งหมดต้องการข้ามแม่น้ำนี้อย่างปลอดภัย แต่ที่ฝั่งที่เขาอยู่ทั้งหมดมีเรืออยู่หนึ่งลำที่สามารถบรรทุกคนได้สูงที่สุดสองคน มิสชันนารีไม่แน่ใจว่าอะไรจะเกิดขึ้นถ้าจำนวนของมิสชันนารีมีน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มนุษย์กินคน นั้นคือมิสชันนารีต้องการวางแผนข้ามแม่น้ำอย่างไร เพื่อไม่ให้เกิดสถานะการณ์ที่จำนวนของมิสชันนารีมีน้อยกว่ามนุษย์กินคน

4. เกมหอคอยฮานอย (The Tower of Hanoi): ในเมือง Hanoi มีพระอยู่กลุ่มหนึ่งซึ่งพยายามที่จะทำงานที่สำคัญต่อไปนี้ให้เสร็จ คือมีเสาอยู่สามต้น โดยมีเสาต้นหนึ่งที่มี จานกลมเรียงกันจากขนาดเล็กไปขนาดใหญ่ซ้อนกันอยู่ 64 จาน โดยที่จานที่มีขนาดเล็กจะซ้อนอยู่บนจานที่มีขนาดใหญ่กว่า หน้าที่ของพระคือการโยกย้ายจานเหล่านี้จากเสาต้นหนึ่งไปอีกต้นหนึ่ง โดยกำหนดให้มีการโยกย้ายได้ทีละจาน และในระหว่างการโยกย้าย ต้องไม่มีการวางจานขนาดใหญ่ซ้อนบนจานขนาดเล็กกว่า และไม่อนุญาตให้วางจานนอกเหนือจากเสาทั้งสามต้นนี้ พระเหล่านี้ควรย้ายจานเหล่านี้อย่างไรให้เร็วที่สุดเท่าที่จะเป็นไปได้
5. เกมปริศนาเลขคณิตลึกลับ (Cryptarithmetic): กำหนดตัวอักษรหนึ่งชุดซึ่งเขียนในเทอมของการคำนวณทางคณิตศาสตร์เช่น SEND + MORE = MONEY ให้หาการกำหนดค่าของตัวหนังสือโดยใช้ตัวเลขระหว่าง 0 ถึง 9 แทนตัวอักษรแต่ละตัวแล้วทำให้การคำนวณทางคณิตศาสตร์ที่ปรากฏถูกต้อง เน้นว่าตัวอักษรต่างกันจะมีการกำหนดค่าตัวเลขที่ต่างกัน และไม่มีตัวเลขใดที่กำหนดให้กับตัวอักษรเกินกว่าหนึ่งตัว

2.7ยุทธวิธีในการแก้ปัญหา

1. **Tic-Tac-Toe:**
   - วิธีการแก้ปัญหาแบบที่ 1

การแทนปัญหาในคอมพิวเตอร์

กระดาน ใช้เวกเตอร์ขนาด 9 ตัวซึ่งมีตำแหน่งดังรูป

1	2	3
4	5	6
7	8	9

แต่ละค่าของเวกเตอร์จะใช้ 0 แทนตำแหน่งนั้นว่าง 1 แทน X และ 2 แทน O

ตารางการเล่น ใช้เวกเตอร์ขนาด 19683 (3^9) ตัวซึ่งแต่ละตัวแทนเวกเตอร์ขนาด 9 ที่ใช้ตำแหน่งของดัชนี ในการบอกการเล่นครั้งถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ขั้นตอนวิธี**
  1. แปลงเวกเตอร์ที่มีอยู่ในปัจจุบันเป็นเลขฐานสิบ (มองเวกเตอร์เป็นชุดตัวเลขฐานสาม)
  2. ใช้ตัวเลขที่คำนวณได้เป็นดัชนี โดยดึงตารางการเล่นที่ตำแหน่งของดัชนีนั้นมาเป็นคำตอบของการเล่นถัดไป
- **วิธีการแก้ปัญหาแบบที่ 2**

การแทนปัญหาใน

คอมพิวเตอร์

กระดาน ใช้เวกเตอร์ขนาด 9 ตัวซึ่งมีตำแหน่งดังรูป

1	2	3
4	5	6
7	8	9

แต่ละค่าของเวกเตอร์จะใช้ 2 แทนตำแหน่งนั้นว่าง 3 แทน X และ 5 แทน O

การเล่นตาต่อไป ใช้ตัวเลข 1 ถึง 9 โดยที่ 1 หมายถึงช่องแรก และ 9 หมายถึงช่องสุดท้าย

- **ขั้นตอนวิธี** จะใช้โปรแกรมย่อยสามตัวคือ Make2, Posswin(p) และ Go(n)

Make2 ส่งค่า 5 (ช่องกลาง) ในกรณีที่ช่องเบอร์ 5 นั้นเป็นช่องว่าง (มีค่าเป็น 2)

Posswin(p) ส่งค่า 0 ออกมาถ้า p ไม่สามารถชนะได้ ในกรณีอื่น ๆ จะให้ค่าของช่อง ที่จะทำให้ผู้เล่น p ชนะได้

เราจะใช้โปรแกรมย่อยนี้ในการเดินและการกัน เราจะใช้ Posswin(p) ทดสอบคำตอบในแนวแถว สดมภ์และทแยง โดยถ้าค่าในกลุ่มใดเป็น 18 (3 x 3 x 2) จะได้ว่า X จะชนะ, 50 (5 x 5 x 2) จะได้ว่า O จะชนะ ถ้าเราพบว่าแถวใดทำให้เราชนะได้ให้ส่งค่าของช่องว่างนั้นออกมา

Go(n) เป็นฟังก์ชันที่ลงการเดินตาต่อไป โดยกำหนดให้ค่าบนกระดานที่ตำแหน่ง n เป็น 3 ถ้าตานี้เป็นเลขคี่ หรือ 5 ถ้าตานี้เป็นเลขคู่

- **เราใช้ขั้นตอนวิธีนี้ดังนี้ คือถ้าการเดินเป็นครั้งที่คี่ การเดินนี้เป็นของ X และสำหรับครั้งที่เดินเป็นเลขคู่คือการเดินของ O**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Go(1).
2. If Board[5] is blank, Go(5), else Go(1).
3. If Board[9] is blank, Go(9), else Go(3).
4. If Posswin(X) is not 0, then Go(Posswin(X)) [i.e., block opponent's win], else Go(Make2).
5. If Posswin(X) is not 0, then Go(Posswin(X)) [i.e., win] else if Posswin(O) is not 0, then Go(Posswin(O)) [i.e., block win], else if Board[7] is blank, then Go(7), else Go(3).
6. If Posswin(O) is not 0, then Go(Posswin(O)), else if Posswin(X) is not 0, then Go(Posswin(X)), else Go(Make2).
7. If Posswin(X) is not 0, then Go(Posswin(X)), else if Posswin(O) is not 0, then Go(Posswin(O)), else go anywhere that is blank.
8. If Posswin(O) is not 0, then Go(Posswin(O)), else if Posswin(X) is not 0, then Go(Posswin(X)), go anywhere that is blank.
9. If Posswin(X) is not 0, then Go(Posswin(X)), else if Posswin(O) is not 0, then Go(Posswin(O)), else go anywhere that is blank.

- วิธีการแก้ปัญหาแบบที่ 3

การแทนปัญหาในคอมพิวเตอร์

กระดาน ใช้เวกเตอร์ขนาด 9 ตัวซึ่งมีตำแหน่งดังรูป

8	3	4
1	5	9
6	7	2

สังเกตได้ว่ากระดานที่ได้เป็น Magic Square คือทุกแถว ทุกสดมภ์และทุกแนวทแยงมีผลรวมเท่ากับ 15 แต่ละค่าของเวกเตอร์จะใช้ 2 แทนตำแหน่งนั้นว่าง 3 แทน X และ 5 แทน O

การเล่นตาต่อไป ใช้ตัวเลข 1 ถึง 9 โดยที่ 1 หมายถึงช่องแรก และ 9 หมายถึงช่องสุดท้าย

- **ขั้นตอนวิธี** จะใช้โปรแกรมย่อยสามตัวคือ Make2, Posswin(p) และ Go(n)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Make2 ส่งค่า 5 (ช่องกลาง) ในกรณีที่ช่องเบอร์ 5 นั้นเป็นช่องว่าง (มีค่าเป็น 2)

Posswin(p) ส่งค่า 0 ออกมาถ้า p ไม่สามารถชนะได้ ในกรณีอื่น ๆ จะให้ค่าของช่องที่จะทำให้ผู้เล่น p ชนะได้

เราจะใช้โปรแกรมย่อยนี้ในการเดินและการกัน กำหนดให้มีการเก็บข้อมูลของช่องที่ผู้เล่นแต่ละคนได้เล่นไปแล้ว การทดสอบจะใช้หลักการที่ว่า เราพิจารณาผู้เล่นคนหนึ่งโดยนำ ช่องที่ผู้เล่นนั้นได้เลือกไปแล้วมาสองช่อง บวกกันแล้วลบออกจาก 15 ถ้าค่าที่ได้แสดงถึงช่องที่ว่าง เราจะได้ว่าผู้เล่นคนนั้นเลือกช่องนี้จะชนะ

Go(n) เป็นฟังก์ชันที่ลงการเดินตาต่อไป โดยกำหนดให้ค่าบนกระดานที่ตำแหน่ง n เป็น 3 ถ้าตานี้เป็นเลขคี่ หรือ 5 ถ้าตานี้เป็นเลขคู่

- เราใช้ขั้นตอนวิธีนี้ดังนี้ คือถ้าการเดินเป็นครั้งที่คี่ การเดินนี้เป็นของ X และสำหรับครั้งที่เดินเป็นเลขคู่คือการเดินของ O

1. Go(1).
2. If Board[5] is blank, Go(5), else Go(1).
3. If Board[9] is blank, Go(9), else Go(3).
4. If Posswin(X) is not 0, then Go(Posswin(X)) [i.e., block opponent's win], else Go(Make2).
5. If Posswin(X) is not 0, then Go(Posswin(X)) [i.e., win] else if Posswin(O) is not 0, then Go(Posswin(O)) [i.e., block win], else if Board[7] is blank, then Go(7), else Go(3).
6. If Posswin(O) is not 0, then Go(Posswin(O)), else if Posswin(X) is not 0, then Go(Posswin(X)), else Go(Make2).
7. If Posswin(X) is not 0, then Go(Posswin(X)), else if Posswin(O) is not 0, then Go(Posswin(O)), else go anywhere that is blank.
8. If Posswin(O) is not 0, then Go(Posswin(O)), else if Posswin(X) is not 0, then Go(Posswin(X)), go anywhere that is blank.
9. If Posswin(X) is not 0, then Go(Posswin(X)), else if Posswin(O) is not 0, then Go(Posswin(O)), else go anywhere that is blank.

- วิธีการแก้ปัญหาแบบที่ 4

การแทนปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในคอมพิวเตอร์

กระดาน ใช้เวกเตอร์ขนาด 9 ตัวซึ่งมีตำแหน่งดังรูป

1	2	3
4	5	6
7	8	9

และรายการของกระดานที่เป็นไปได้ทั้งหมดจากการเดินหนึ่งตา พร้อมทั้งตัวเลขที่ประมาณค่าตัวเลขในการนำไปสู่การเล่นที่ฝ่ายเราจะชนะ เราเรียก Merit function แต่ละค่าของเวกเตอร์จะใช้ 2 แทนตำแหน่งนั้นว่าง 3 แทน X และ 5 แทน O

การเล่นตาต่อไป เลือกตาเดินที่มีค่า Merit function มากที่สุด

- **ขั้นตอนวิธี** การเลือกตาเดินเราจะใช้หลักการ มองไปข้างหน้า (Look ahead ซึ่งหมายถึงการเดินในตาต่อไป) ตัดสินใจว่าตาเดินใดให้ค่า Merit function ที่ดีที่สุด ทำการเลือกเพื่อไปให้ถึงค่าที่ดีที่สุดนั้น
  ในการตัดสินใจเลือกทางเดินที่ดีที่สุดให้พิจารณาดังนี้
  1. ถ้าการเลือกตาต่อไปทำให้เกิดการชนะ ให้กำหนดค่า Merit ให้มากที่สุด
  2. มิฉะนั้นพิจารณาตาเดินที่เป็นไปได้ของคู่ต่อสู้ถัดไป ตาเดินใดที่ทำให้เราแพ้หรือมีค่า Merit ต่ำ ให้ส่งค่าที่ต่ำนั้นเป็นการบอกทางเลือกนี้
  3. ทางเลือกที่ดีที่สุดคือทางเลือกที่มีค่า Merit สูงที่สุด

2. A Water Jug Problem: ปัญหาการตวงน้ำ

เรามีถังสองใบ ใบแรกมีความจุ 4 แกลลอนและใบที่สองมีความจุ 3 แกลลอน โดยที่ถังทั้งสองใบไม่มีรอยตวงให้สามารถวัดจำนวนปริมาณน้ำได้ มีปั๊มน้ำที่สามารถเติมน้ำในแต่ละถังให้เต็มได้ จงหาว่าเราจะตวงน้ำ 2 แกลลอนใส่ในถังขนาด 4 แกลลอนอย่างไร โดยไม่ใช้การประมาณ

1. ปริภูมิของปัญหา S = {(0, 0), (0, 1), ..., (4, 3)} โดยที่แต่ละสถานะแทนด้วย (x, y) เมื่อ x แทนจำนวนน้ำในถังแรก และ y แทนจำนวนน้ำในถังที่สอง
2. สถานะเริ่มต้นคือ (0, 0)
3. สถานะเป้าหมายคือ (2, 0), (2, 1), (2, 2), (2, 3)
4. เราต้องการเปลี่ยนสถานะจากสถานะหนึ่งไปอีกสถานะหนึ่ง เราอาจใช้
   1. ตารางบอกการเปลี่ยนสถานะ เช่น (0, 0) $\rightarrow$(0, 3) $\rightarrow$(3, 0) $\rightarrow$(3, 3) $\rightarrow$(4, 2) $\rightarrow$(0, 2) $\rightarrow$(2, 0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กฎการเปลี่ยนสถานะ เช่น (x, 0) $\rightarrow$(0, 0) หมายถึงการเทน้ำในถังแรกออก

5. เรากำหนดกฎทั้งหมดที่เป็นไปได้ โดยใช้ตัวแปร และกฎการอนุมาน ดังนี้
   1. กฎการเติมน้ำให้เต็มถัง
      ถ้า $x < 4$, $(x, y) \rightarrow (4, y)$
      ถ้า $y < 3$, $(x, y) \rightarrow (x, 3)$
   2. กฎการเทน้ำออกจากถังทั้งหมด
      ถ้า $x > 0$, $(x, y) \rightarrow (0, y)$
      ถ้า $y > 0$, $(x, y) \rightarrow (x, 0)$
   3. กฎการเทน้ำ จากถังที่สองมาถังแรกจนเต็ม
      ถ้า $x+y \geq 4$ และ $y > 0$, $(x, y) \rightarrow (4, y - (4-x))$
   4. กฎการเทน้ำ จากถังแรกมาถังที่สองจนเต็ม
      ถ้า $x+y \geq 3$ และ $x > 0$, $(x, y) \rightarrow (x - (3-y), 3)$
   5. กฎการเทน้ำ จากถังที่สองใส่ถังแรกทั้งหมด
      ถ้า $x+y \leq 4$ และ $y > 0$, $(x, y) \rightarrow (x+y, 0)$
   6. กฎการเทน้ำ จากถังแรกใส่ถังที่สองทั้งหมด
      ถ้า $x+y \leq 3$ และ $x > 0$, $(x, y) \rightarrow (0, x+y)$
   7. กฎการเทน้ำ 2 แกลลอนจากถังที่สอง มาใส่ถังแรก $(0, 2) \rightarrow (2, 0)$
6. เราแก้ปัญหานี้โดยเขียนโปรแกรม วนซ้ำจนกระทั่งได้สถานะเป้าหมาย โดยที่ทดสอบว่ากฎใดสามารถใช้กับสถานะการณ์ ปัจจุบันได้ แล้วใช้กฎเพื่อเปลี่ยนสถานะ ทดสอบว่าสถานะใหม่ที่ได้ใช่สถานะสุดท้ายหรือไม่ ถ้าใช่ให้หยุดการทำงาน มิฉะนั้น ให้วนซ้ำ

   สำหรับปัญหานี้ เราพบว่าผลเฉลยคือลำดับของการเติมน้ำ ซึ่งมีได้หลายวิธี โดยปกติแล้วเราจะสนใจลำดับที่มีการเปลี่ยน สถานะที่น้อยที่สุด

   นอกจากนี้เราต้องพิจารณา
   1. การเปรียบเทียบเงื่อนไข ข้างหน้าก่อนการนำกฎไปใช้ ตัวอย่างเช่นในกฎแรก ถ้าเราไม่บังคับให้ทดสอบว่า $x < 4$ ก่อนการนำกฎไปใช้เราจะพบว่า (4, 1) เมื่อใช้กฎแรกแล้วสถานะไม่มีการเปลี่ยนสถานะ การเติมกฎบังคับการนำกฎไปใช้จะช่วยทำให้การค้นหาผลเฉลยมีประสิทธิภาพเพิ่มขึ้น
   2. กฎบางกฎ อาจไม่จำเป็นในการนำไปใช้เช่น
      กฎการเทน้ำ จากถังแรกออกโดยไม่ทราบจำนวน
      ถ้า $x > 0$, $(x, y) \rightarrow (x - d, y)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราพบว่า กฎลักษณะนี้ เราไม่สามารถนำเราไปสู่ผลเฉลยที่ต้องการได้ การเขียนอธิบายปัญหาเราอาจไม่จำเป็นต้องเขียนกฎที่เป็นไปได้ทั้งหมด เราอาจลดกฎบางกฎที่ไม่จำเป็น

3. กฎบางกฎเป็นกฎเฉพาะ จากกฎทั่วไป เช่น กฎ 5 เป็นกรณีเฉพาะของกฎ 4 เมื่อ x = 0, y = 2 ในบางครั้งการกำหนดกฎเฉพาะจะช่วยให้การหาผลเฉลยเป็นไปได้ง่ายขึ้น

ตัวอย่างของลำดับที่เป็นไปได้

น้ำในถัง 4 แกลลอน	น้ำในถัง 3 แกลลอน	กฎที่ใช้
0	0	1
0	3	4
3	0	1
3	3	3
4	2	2
0	2	5
2	0	

เราแก้ปัญหาโดยใช้กฎที่ให้มา ร่วมกับ ยุทธวิธีเลือกกฎที่เหมาะสม ทำให้สามารถหาผลเฉลยของปัญหาได้ การค้นหานับว่าเป็นพื้นฐานอย่างหนึ่งที่สำคัญในการแก้ปัญหาทุกแบบ แต่ไม่ได้หมายความว่าการค้นหาจะเป็นวิธีเดียว ที่ใช้แก้ปัญหา เราอาจนำวิธีการอื่นมาประกอบการค้นหา ตัวอย่างเช่นในกฎที่ 3 เราใช้การคำนวณค่าของ y - (4-x) แทนที่การทดสอบค่าที่เป็นไปได้ การค้นหามักจะถูกใช้เมื่อวิธีการอื่นนั้นไม่สามารถนำมาใช้ได้

<u>กลยุทธการเลือก</u>

กลยุทธการเลือกที่เหมาะสมมีความสำคัญในการหาผลเฉลยของปัญหาซึ่งมีหลักการดังนี้

7. กลยุทธต้องทำให้เกิดการเปลี่ยนแปลงที่นำไปสู่ผลเฉลยหรือคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. กลยุทธต้องมีการกระทำอย่างเป็นระบบ ระเบียบและแบบแผน ตัวอย่างเช่น การกำหนดกลยุทธการเลือกกฎโดยการสุ่มทุกครั้ง อาจไม่ใช่กลยุทธที่ถูกต้อง แม้ว่าเราอาจไปถึงผลเฉลยที่ต้องการ แต่อาจต้องใช้เวลานานมาก เพราะเราจะมีโอกาสที่จะใช้กฎแล้วย้อนกลับมาที่สถานะเดิม ซึ่งไม่เป็นระบบ

ตัวอย่างของกลยุทธที่ดีสำหรับปัญหาการเติมน้ำเช่น ให้เก็บข้อมูลในแต่ละสถานะในลักษณะต้นไม้ โดยมีสถานะเริ่มต้น เป็นรากของต้นไม้การค้นหา (Search tree) เริ่มแตกกิ่งของต้นไม้โดยทดลองใช้กฎ แต่ละกฎและสร้างจุดยอดให้ใหม่ สำหรับแต่ละสถานะที่เกิดขึ้นใหม่ และสำหรับจุดยอดชุดใหม่ให้ทำซ้ำกับการกระทำที่ราก ไปจนพบผลเฉลยที่ต้องการ เราเรียกวิธีการนี้ว่า การค้นหาในแนวกว้าง **(Breadth-first search)**

Breadth-first search

ข้อมูลเข้า:

ข้อมูลออก: ลำดับการเติมน้ำในถังเพื่อให้ได้น้ำ 2 แกลลอนในถัง 4 แกลลอน

9. Set the NODE-LIST to contain the initial state (0,0)
10. while (NODE-LIST is not empty) then
11. Remove the first element from NODE-LIST, called it E.
12. For each rule that match the state of E do
13. Keep track of the rule applied for each E.
14. Apply the rule to generate the new state.
15. If the new state is the goal state, quit and return this solution.
16. Otherwise add the new state to the end of NODE-LIST.
17. In case, that NODE-LIST is empty, then report that this is impossible to do.

หรือเราอาจค้นกิ่งทีละกิ่งจากต้นไม้ของการค้นหา (Search tree) โดยพิจารณาจุดยอดไปในทางลึกจนพบผลเฉลยที่ต้องการ หรือในกรณีที่ไม่สามารถค้นหาต่อได้ เราสามารถย้อนกลับมาสถานะก่อนหน้าสถานะนั้น (backtracking) เราเรียกวิธีการนี้ว่า **การค้นหาในแนวลึก (Depth-first search)**

Depth-first search

ข้อมูลเข้า:

ข้อมูลออก: ลำดับการเติมน้ำในถังเพื่อให้ได้น้ำ 2 แกลลอนในถัง 4 แกลลอน

18. Set the NODE-LIST to contain the initial state (0,0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19. while (NODE-LIST is not empty) then
20. Remove the first element from NODE-LIST, called it E.
21. For each rule that match the state of E do
22. Keep track of the rule applied for each E.
23. Apply the rule to generate the new state.
24. If the new state is the goal state, quit and return this solution.
25. Otherwise add the new state to the end of NODE-LIST.

2.8สรุปเปรียบเทียบระหว่างการค้นหาในแนวลึก กับในแนวกว้าง

ข้อดีของ Depth-first search

1. Depth-first search ใช้หน่วยความจำที่น้อยกว่า Breadth-first search เพราะว่า เรามีการเก็บข้อมูลเฉพาะในวิถีที่เราพิจารณา ในขณะที่ Breadth-first search ต้องเก็บข้อมูลแต่ละระดับทีละระดับทั้งหมด
2. ในบางกรณี Depth-first search จะพิจารณาเพียงสถานะไม่มากเมื่อเทียบกับ ปริภูมิของสถานะทั้งหมด ในขณะที่ Breadth-first search จะพิจารณาสถานะทีละระดับทุกสถานะก่อนจะพิจารณาอีกระดับหนึ่ง

ข้อดีของ Breadth-first search

1. Breadth-first search จะไม่ถูกกักอยู่ในสถานะที่วนซ้ำไม่มีวันจบเหมือน Depth-first search ซึ่งอาจจะเกิดได้ถ้าไม่มีการเก็บข้อมูลของสถานะที่ได้ผ่านมาแล้ว
2. ถ้าปัญหานั้นมีผลเฉลย Breadth-first search จะหยุดเสมอ และถ้าผลเฉลยนั้นมีได้หลายผลเฉลย Breadth-first search จะได้ผลเฉลยที่ใช้ระยะที่สั้นที่สุด (ระยะคือจำนวนการใช้กฎในแต่ละกิ่งที่เกิดขึ้น) ในขณะที่ Depth-first search อาจได้ผลเฉลยที่ใช้ระยะมากกว่า เพราะว่าผลเฉลยที่ระยะสั้นที่สุดไม่ได้ถูกนำพิจารณาในการค้นนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บทที่3
โปรแกรมที่ใช้ในโครงงาน

ในบทนี้จะเป็นการอธิบายการทำงานและการใช้งานของโปรแกรมAI ที่เอาไว้ควบคุมหุ่นยนต์ทั้ง 5 ตัวเพื่อให้สามารถเล่นร่วมกันเป็นทีมโดยที่โปรแกรมนี้สร้างขึ้นมาจากโปรแกรม JAVA ซึ่งตัวโปรแกรมนี้เป็นไฟล์ที่ใช้ในการจัดการ Bytestream ที่ส่งมาเป็นอินพุตของการทำงานขของโปรแกรมโดยค่าของ Byte เหล่านั้นจะเป็นพารามิเตอร์ของฝั่งเราและฝั่งตรงข้าม เพื่อที่จะนำมาวางแนวทางที่จะทำการเล่นต่อไป ต่อไปจะกล่าวถึงรายละเอียดของโปรแกรมดังนี้

3.1 รายละเอียดของโปรแกรมที่ใช้พัฒนาและที่มา

1. JAVA 2 PLATFORM STANDARD EDITION DEVELOPMENT KIT 5.0 (jDK 5.0) จาก http://java.sun.com/j2se/1.5.0/download.jsp

2. Java Communications API **จาก**

http://java.sun.com/products/javacomm/downloads/

3. JCreator2.0 จาก

http://www.download.com/3120-20_4-0.html?qt=jcreator&tg=dl-2001

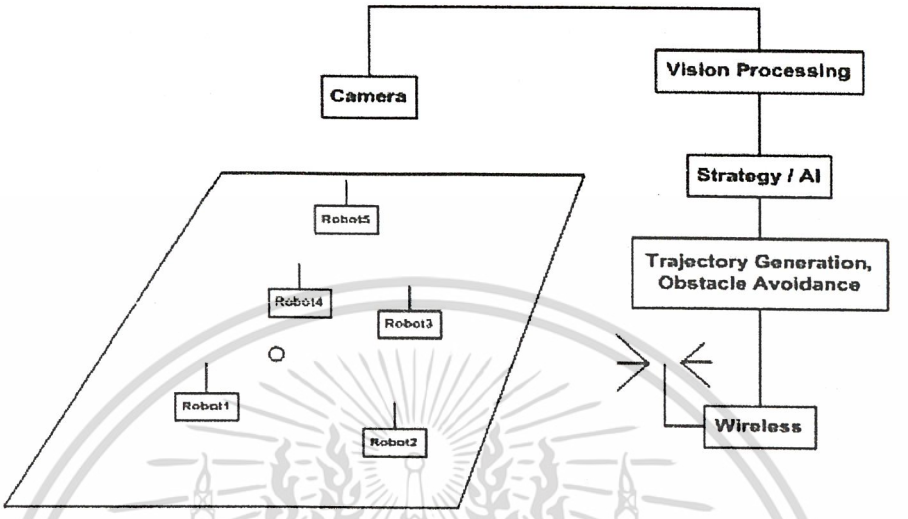
หมายเหตุ 1. URL ที่เขียนไว้ข้างต้นอาจเปลี่ยนแปลงได้ แก้ไขล่าสุดเมื่อ 14 มี.ค 2548

2. **JCreator2.0** เป็นเพียง IDE ผู้ที่จะนำไปพัฒนาต่อสามารถเลือกตัวอื่นก็ได้

3.2 ขั้นตอนการออกแบบโปรแกรม

ก่อนอื่นเราจะต้องมาดูไดอะแกรมการทำงานของระบบรวมทั้งหมดก่อนซึ่งในส่วนแรกเราจะทำการออกแบบในส่วนของอินพุตและเอาต์พุ ตของโปรแกรมซึ่งถ้าดูจากในรูปจะเห็นว่าตัวปัญญาประดิษฐ์(ในรูปจะรวมระหว่าง strategy/AI และ obstacle avoidance) จะเห็นว่ามันจะรับอินพุตคือ vision processing เข้ามาประมวลผลเพื่อควบคุมหุ่นยนต์แบบไร้สายต่อไปสำหรับค่าบิตเรตที่ออกแบบไว้สำหรับอินพุตของโปรแกรมนี้คือ 115200 และสำหรับเอาต์พุตให้แก่โปรแกรมคือ 4800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61440



รูปที่3.1 ภาพแสดงระบบทั้งหมดของหุ่นยนต์เตะฟุตบอล

สำหรับในส่วนเอาต์พุตของโปรแกรมนั้นเราจะออกแบบให้มันส่งข้อมูล 4 ไบต์

Protocol มีทั้งหมด 4 byte แบ่งเป็น

Byte ที่ 1 | 1bit | ID 3 bit | direction m1 | direction m2 | direction m3| dribble 1 bit|

Byte ที่ 2 | Vm1 4 bits | Vm2 4 bits |

Byte ที่ 3 | Vm3 4bits | shoot 1 bit | ID 3 bits|

Byte ที่ 4 | 1 1 1 1 1 1 1 1 |

โดยหลักการของโปรโตคอลนี้จะเป็นการส่งค่าความเร็วและทิศของล้อโดยปัญญาประดิษฐ์จะทำการคำนวนค่าต่างๆเพื่อส่งไปให้หุ่นยนต์เองนอกจากนี้โปรโตคอลนี้ยังมีรายละเอียดย่อยๆอีกคือการเลี้ยงและการยิงซึ่งจะสามารถสั่งได้จากโปรโตคอลรูปแบบนี้ได้เช่นกันโดยรูปแบบในการออกแบบนั้นจะออกแบบโดยอิงบนหลักฟิสิกส์นิวตันธรรมดาก่อนแต่ว่าหุ่นยนต์จริงนั้นแรงมอเตอร์แต่ละตัวไม่เท่ากันจึงจะต้องมาทำการปรับละเอียดอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Start byte	ID	Is player 0 in field	+-	X position player 0	+-	Y position player 0	+-	Degree player 0
	ID	Is player 1 in field	+-	X position player 1	+-	Y position player 1	+-	Degree player 1
	ID	Is player 2 in field	+-	X position player 2	+-	Y position player 2	+-	Degree player 2
	ID	Is player 3 in field	+-	X position player 3	+-	Y position player 3	+-	Degree player 3
	ID	Is player 4 in field	+-	X position player 4	+-	Y position player 4	+-	Degree player 4
	ID	Is enemy 0 in field	+-	X position enemy0	+-	Y position enemy0		
	ID	Is enemy 1 in field	+-	X position enemy1	+-	Y position enemy1		
	ID	Is enemy 2 in field	+-	X position enemy2	+-	Y position enemy2		
	ID	Is enemy 3 in field	+-	X position enemy3	+-	Y position enemy3		
	ID	Is enemy 4 in field	+-	X position enemy4	+-	Y position enemy4		
	Is Ball infield	+-	X position		+-	Y position	Stop byte	

รูปที่3.2 ภาพแสดงโปรโตคอลอินพุตของปัญญาประดิษฐ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการออกแบบในส่วนของอินพุตจาก vision processing นั้นเราจะออกแบบให้มีขนาดทั้งหมด 104 byteเนื่องจากข้อมูลส่งมาแบบเรียลไทม์ดังนั้นเราจึงจะต้องกำหนด start byte และ stop byte เอาไว้ตัดข้อมูลซึ่งเนื่องจากมีการส่งผ่านข้อมูลเป็นแบบbyte stream นี่เองทำให้โปรแกรมในส่วนประมวลผลภาพและปัญญาประดิษฐ์ภาษาที่ใช้จะไม่ขึ้นแก่กันนั่นคือตกลงโปรโตคอลเอาไว้จากนั้นก็นำแยกพัฒนากันได้

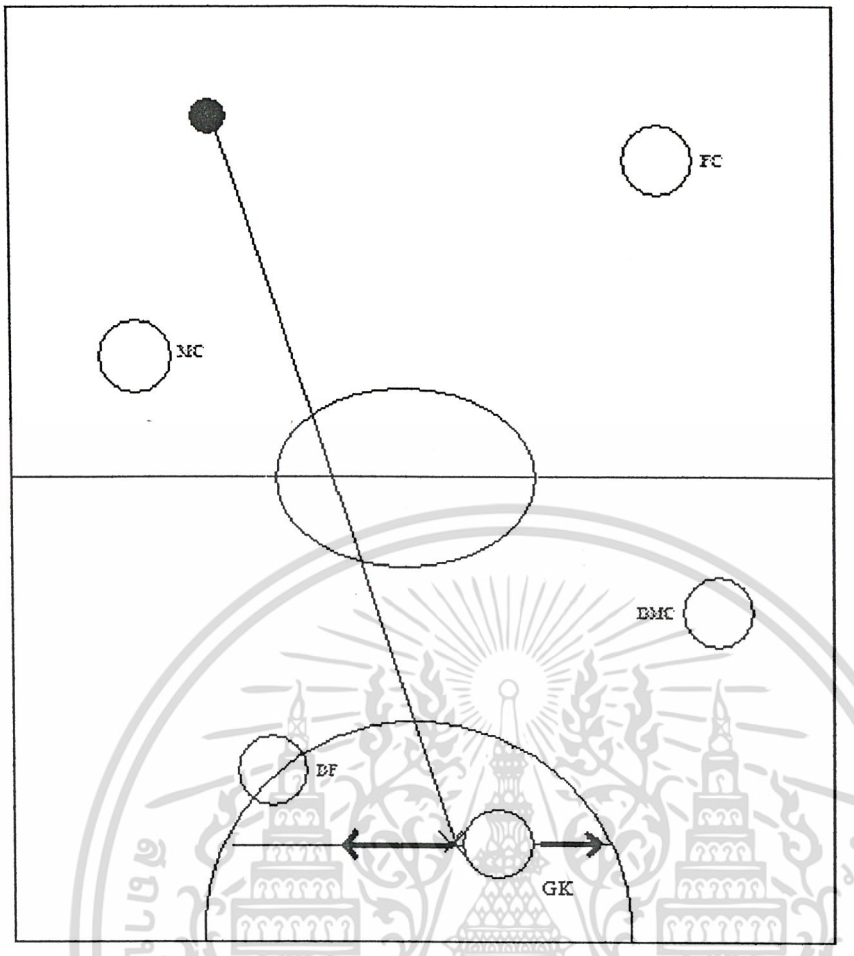
ในการออกแบบโปรแกรมได้เลือกใช้โปรแกรม java เป็นตัวพัฒนาทั้ง 2 โปรแกรมซึ่งการทำงานของโปรแกรมจะเริ่มตั้งแต่การกำหนดพารามิเตอร์เริ่มต้นต่างๆให้แก่โปรแกรม AI เช่นการกำหนดค่าให้ object serial port ใช้ bit rate เท่าใดจากนั้นก็จะนำพิกัดที่ได้มาจาก image processing ไปคำนวนค่าต่างๆเช่น เส้นการโจมตี,ระยะห่างจากบอลของหุ่นยนต์แต่ละตัว,จุดตัดบอลจากนั้นก็นำค่าที่ได้มาเพื่อเป็นเงื่อนไขในการตัดสินใจของหุ่นยนต์แต่ละตำแหน่งต่อไปเช่นนำค่าเส้นโจมตีมาสั่งให้หุ่นที่เล่นตำแหน่งแนวรุกเข้าไปโจมตีในแนวเส้นนั้น

3.2.1 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งผู้รักษาประตู

หลักการสำคัญของอัลกอริธึมนี้คือจะต้องทำการคำนวนจุดที่บอลน่าจะวิ่งผ่านเพื่อที่เราจะได้วิ่งไปป้องกันโดยเราจะทำการสมมุติว่าบอลจะเคลื่อนที่เป็นเส้นตรงจากนั้นเราจะทำการกำหนดเส้นทางการป้องกันขึ้นมาโดยที่ในตอนแรกผู้จัดทำได้ทำการอออกแบบไว้ 2 รูปแบบคือ

1. เคลื่อนที่เป็นแนวเส้นตรง
2. เคลื่อนที่เป็นแนวครึ่งวงกลม

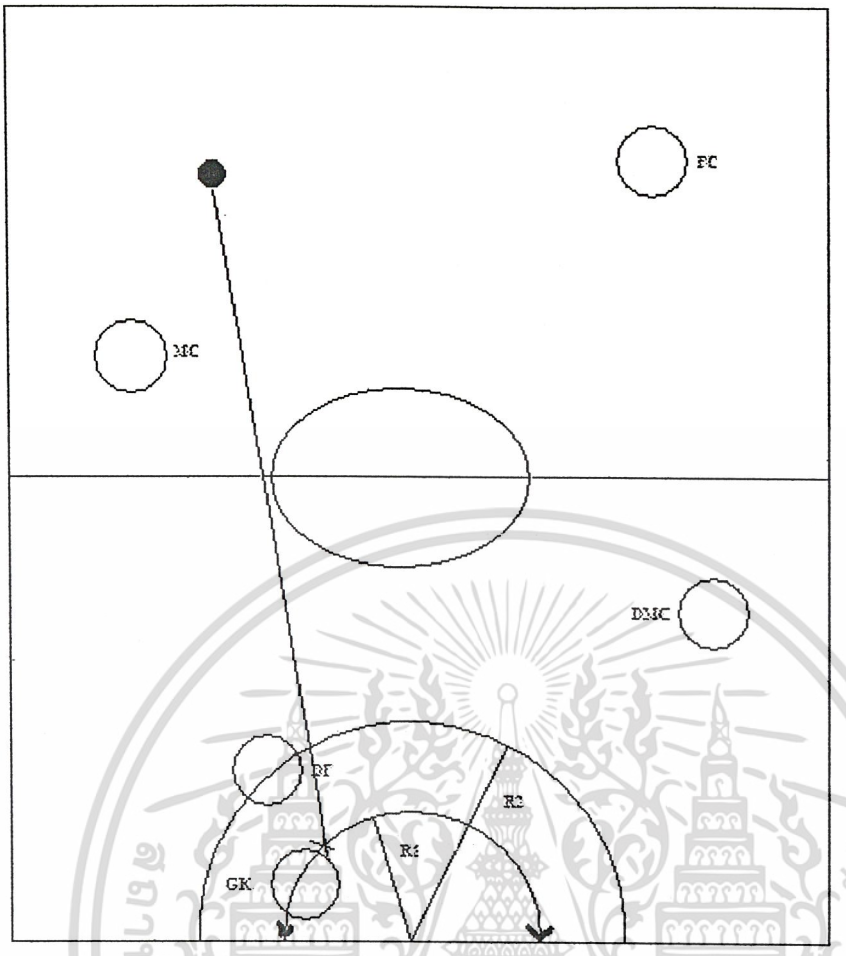
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.1 ภาพแสดงการเคลื่อนที่ผู้รักษาประตูแนวอัลกอริธึมที่ 1

แนวคิดอัลกอริธึมนี้จะให้หุ่นยนต์เคลื่อนที่ขนานเส้นแกน X ข้อดีของแนวคิดนี้คือจะไม่ซับซับซ้อนและเคลื่อนที่ได้รวดเร็วโดยอัลกอริธึมคือคือจะทำการหาทิศทางของลูกบอลที่จะเคลื่อนที่เข้ามาแล้ววิ่งออกไปตัดบอลให้ได้โดยจะมีการตัดสินใจเพียงแค่ 4 รูปแบบเท่านั้นคือหมุนตัวในทิศทวนเข็มและตามเข็มและเคลื่อนที่ไปในแนวซ้ายและขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.2 ภาพแสดงการเคลื่อนที่ผู้รักษาประตูแนวอัลกอริธึมที่ 2

แนวคิดอัลกอริธึมนี้จะให้หุ่นยนต์เคลื่อนที่ตามเส้นสมการครึ่งวงกลม ข้อดีของแนวคิดนี้จะทำให้สามารถบังมุมเอาไว้ได้ดีกว่าอีกอัลกอริธึม(ผู้อ่านสามารถดูรายละเอียดเกี่ยวกับผลการทดลองได้ในบทที่ 4) โ ดยรูปแบบการตัดสินใจจะมีหลายรูปแบบกว่า

3.2.2 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งกองหลัง

หัวข้อนี้จะไม่ลงรายละเอียดเนื่องจากว่าจะใช้หลักการทดสอบแบบเดียวกับอัลกอริธึมของผู้เล่นตำแหน่งผู้รักษาประตูเลยโดยคร่าวๆจะเป็นการช่วยตำแหน่งผู้รักษาประตูในการบังมุมโดยจะมีการเยื้ององศากันเล็กน้อยเพื่อเพิ่มองศาในการบังมากขึ้น

3.2.3 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งกองกลางตัวรับ

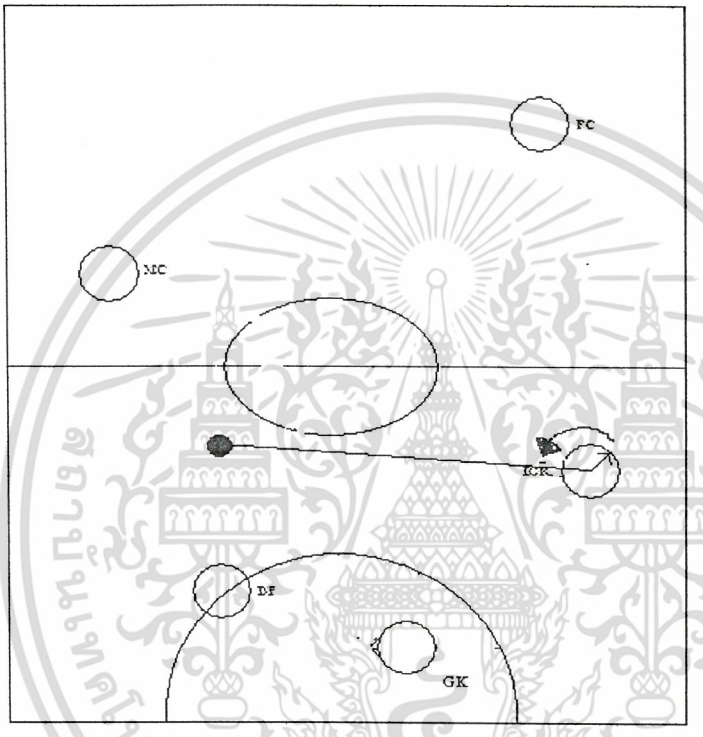
อัลกอริธึมนี้จะเป็นการเข้าหาลูกฟุตบอลให้ได้เร็วที่สุดโดยจะไม่สนถึงว่าหลังจากเข้าหาลูกบอลแล้วจะได้ผลลัพธ์เป็นอย่างไรเพียงแต่ขอให้ลูกบอลไปให้พ้นโกล(เป็นตำแหน่งการเกมเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลุดเกมรุกฝ่ายตรงข้าม)โดยแนวทางที่คิดไว้จะมีรูปแบบในการเข้าลูกฟุตบอล 2 รูปแบบซึ่งถ้าใช้หุ่นยนต์ของที่ลองแล้วจะพบว่าแนวทางที่ 2 จะให้ผลที่ดีกว่าเนื่องจากแนวทางแรกนั้นหุ่นยนต์ของทีมจะต้องสามารถฟีดบอลเอาไว้ให้ได้แต่ถ้าไม่ได้นั้นจะกลายเป็นบอลจะกลิ้งไปที่อื่นทำให้เสียโอกาส 2 แนวทางนั้นคือ

1 สร้างสมการเส้นตรงระหว่างหุ่นกับลูกบอลลูกนั้นแล้วหมุนไปที่เส้นตรง

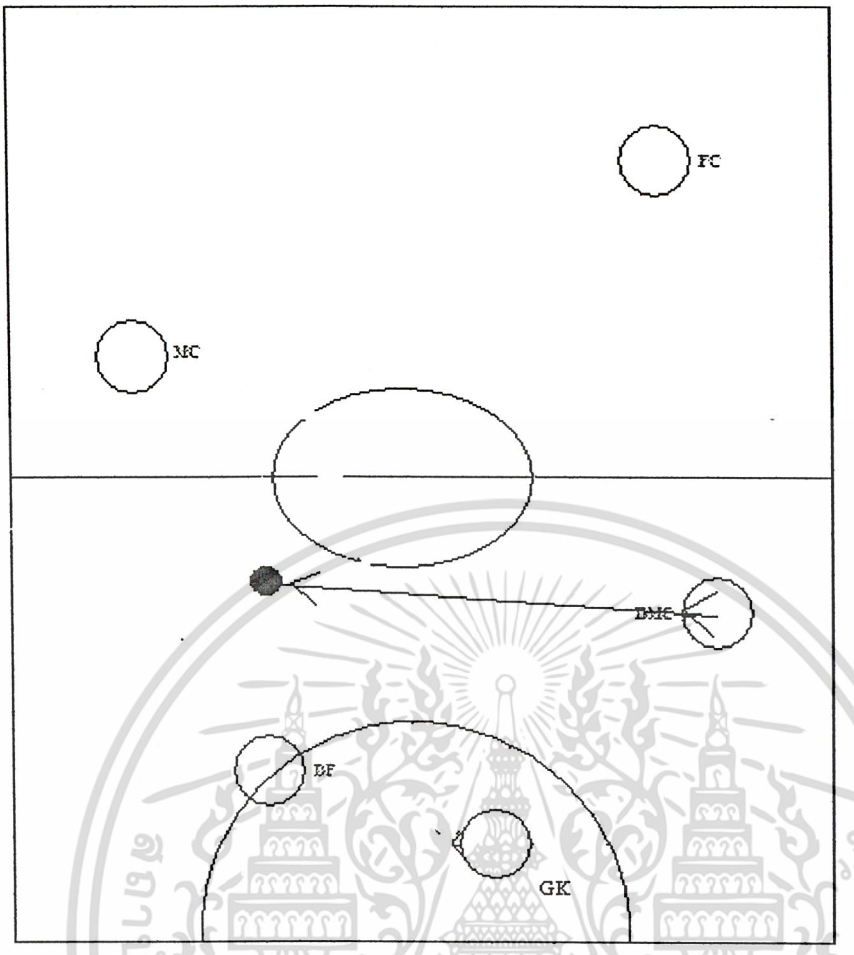
2 เดินไปที่จุดใต้บอลที่คำนวนไว้ก่อนจากนั้นค่อยเข้าหาลูกฟุตบอล



รูปที่3.3 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 1 ขั้นตอนการหมุนสู่เส้นตรงที่ระยะทางสั้นที่สุด

รูปที่ 3.3 นั้นจะเป็นการแสดงขั้นตอนใหญ่ๆขั้นตอนหนึ่งของอัลกอริธึมคือการหมุนเข้าสู้เส้นทางที่ใกล้ลูกฟุตบอลมากที่สุด(ในทางปฏิบัตินั้นยังต้องมีอัลกอริธึมในการเลือกหมุนอีกด้วย)แต่ว่าสำหรับขั้นตอนของอัลกอริธึมนี้ยังไม่จบเพียงเท่านี้ขั้นตอนต่อไปคือการเดินไปทางเส้นทางที่มีระยะทางระหว่างผู้เล่นกับลูกบอลสั้นที่สุด

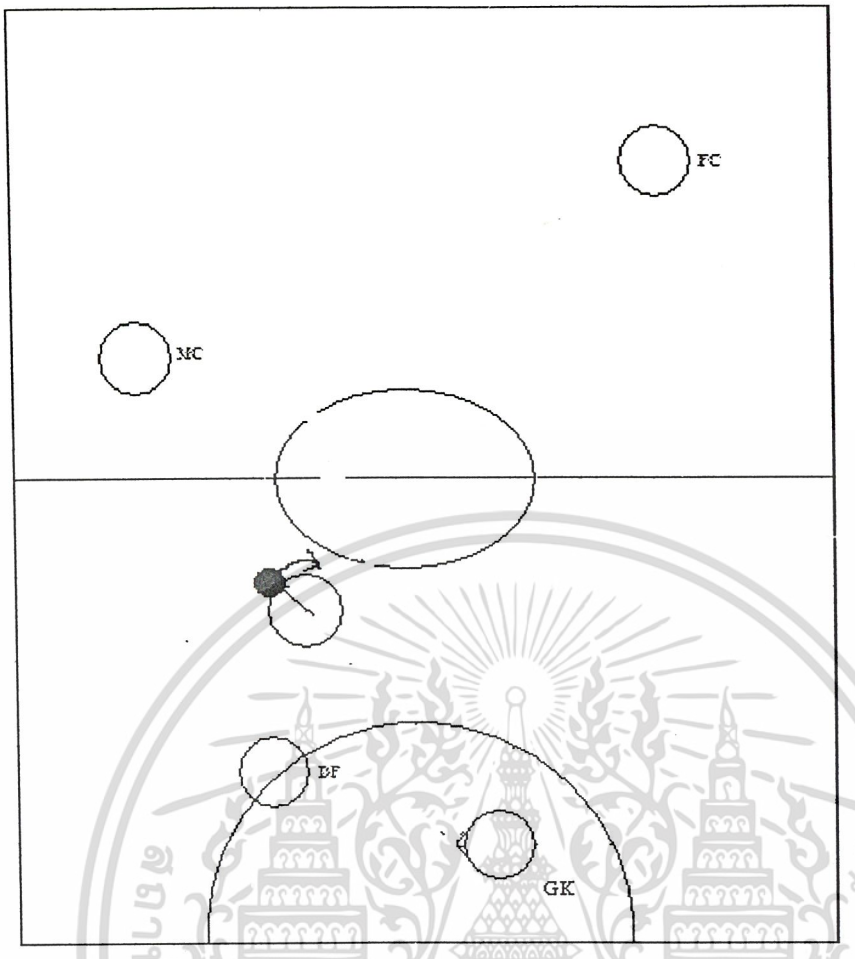
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.4 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 1 ขั้นตอนการเดินตามเส้นตรงที่ระยะทางสั้นที่สุด

รูปที่ 3.4 นั้นจะเป็นการแสดง แนวการเดินของหุ่นยนต์ซึ่งหลังจากขั้นตอนนี้ก็จะมีขั้นตอนอีกคือจะต้องฟีดบอลเพื่อให้เข้าไปสู่แนวทางที่ต้องการเนื่องจากว่าเส้นตรงที่สั้นที่สุดระหว่างหุ่นยนต์กับลูกบอลอาจไม่ใช้เส้นทางที่ดีที่สุดต่อเกมโดยรวมก็ได้สำหรับรูปต่อไปจะเป็นการแสดงการจับบอลให้ไปในทิศทางที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



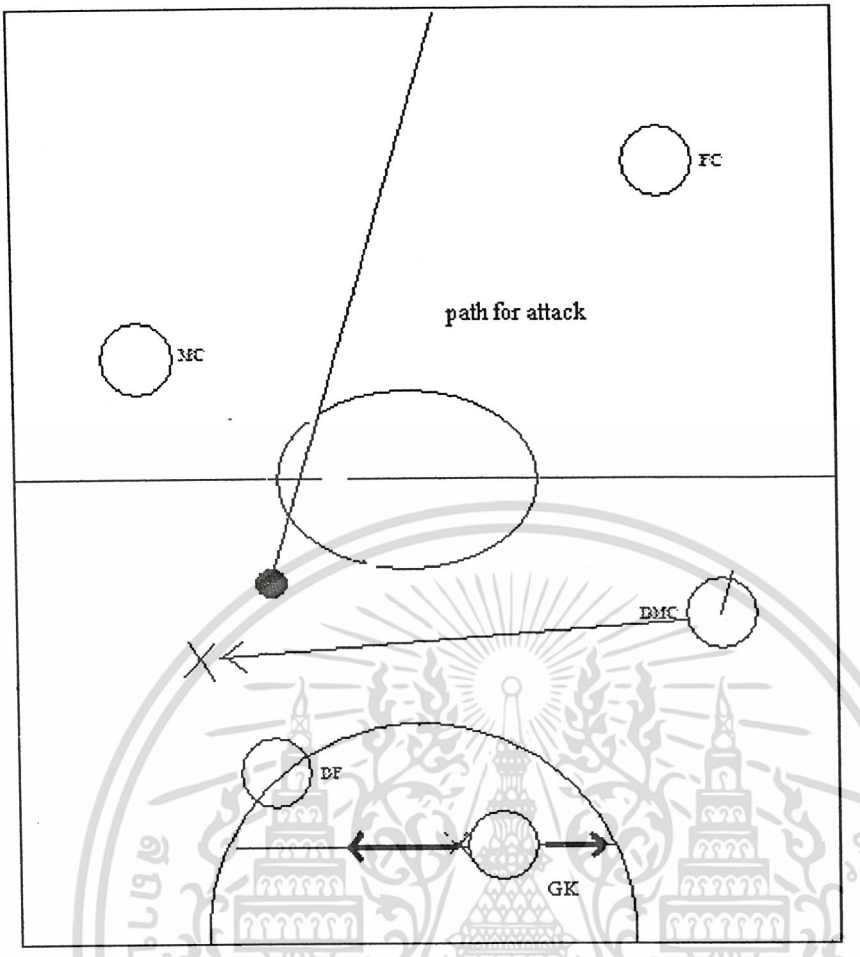
รูปที่3.4 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 1 ขั้นตอนการฟีดบอลเข้าแนวที่ต้องการ

จะเห็นว่าแม้อัลกอริธึมนี้จะเป็นการเดินในเส้นทางที่สั้นที่สุดแต่ว่าจะต้องมีการตรวจสอบเงื่อนไขในการหมุนหุ่นให้ได้ทิศที่ต้องการนานพอสมควรเพราะฉะนั้นผู้จัดทำจึงได้คิดแนวคิดที่ 2 เพิ่มคือจะเป็นแนวการเคลื่อนที่โดยใสนใจว่ามุมปัจจุบันเป็นเช่นไรแต่เอามุมมาคำนวนว่าเมื่อเคลื่อนที่ทิศนี้จะทำให้พิกัดใหม่ที่ได้น่าจะเป็นเท่าใด

หมายเหตุ 1.สำหรับแนวคิดที่ 2 นั้นเมื่อผู้ทำได้เอามาลองกับหุ่นของทีมพบว่าจะให้ประสิทธิภาพที่ดีกว่าแต่จากประสปการณ์ในการแข่งนั้นพบว่าถ้าหุ่นยนต์นั้นสามาคอนโทรลองศาได้แม่นยำแล้วแนวคิดที่ 1 จะมีความรวดเร็วกว่าและแม่นยำกว่า

2.ผู้จัดทำได้เลือกแนวคิดที่ 2 ปืนหลัก

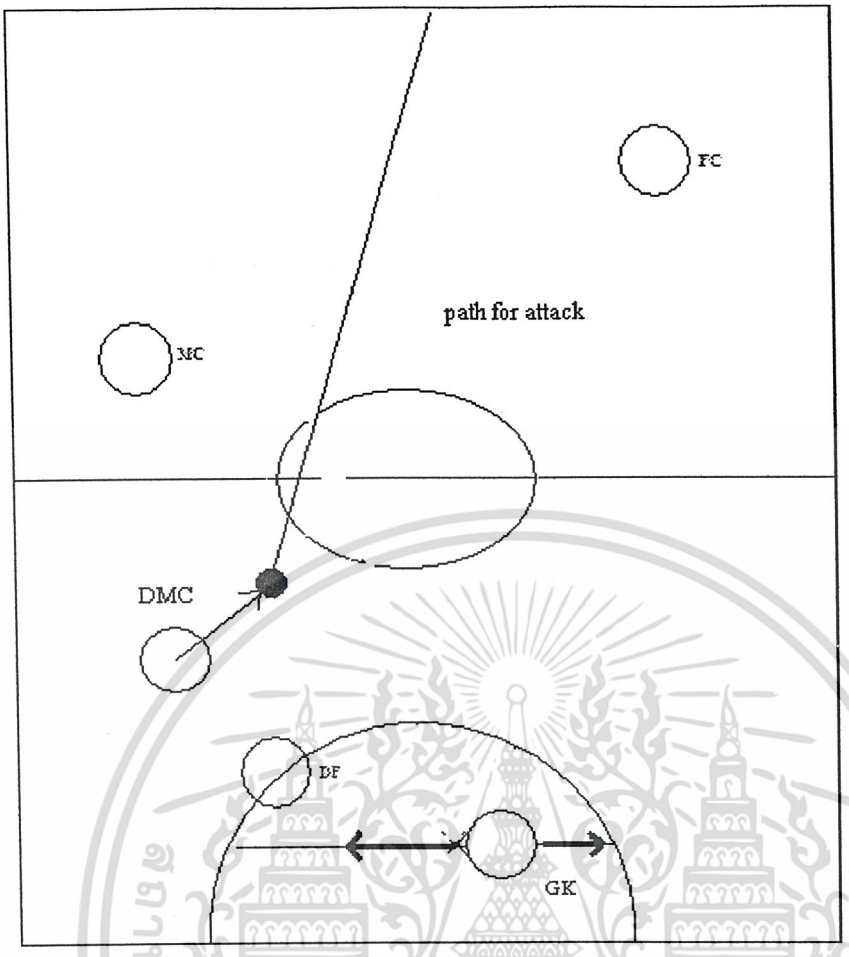
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.3 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 2 ขั้นตอนการเดินไปสู่จุดใต้บอล

จะเห็นว่าแม้อัลกอริธึมนี้จะเป็นมีค่าผลรวมระยะทางที่เยอะกว่าแต่ก็ใช้เวลาในการเข้าสู่เงื่อนไขได้ง่ายกว่าเนื่องจากมันจะเริ่มเคลื่อนที่เลยโดยอาจจะประกอบจากการเดินหลายรูปแบบที่ไม่ใช่เดินตรงอย่างเดียวเท่านั้นซึ่งแนวคิดนี้จะใช้เวลาโดยเฉลี่ยของแต่ในแต่ละสถาการณ์เท่ากันซึ่งต่างจากแนวคิดแรกที่ถ้าลูกบอลเคลื่อนที่มากๆแล้วจะทำให้ต้องเสียเวลาหมุนหาลูกบอลนานกว่าปกติหรือแม้แต่ถ้าลูกบอลอยู่ข้างหลังหุ่นยนต์แล้วจะทำให้ต้องเสียเวลาในการหมุนมากกว่าปกติ นอกจากนี้ในการแข่งจริงการฟีดบอลที่ผิดพลาดนั้นอาจทำให้เกิดผลเสียต่อทีมอย่างมากเช่นการทำเข้าประตูตัวเองโดยจะทำการคำนวนทิศทางที่จะต้องเดินทางจากค่ามุมปัจจุบันและและจุดที่มันต้องการจะเดินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



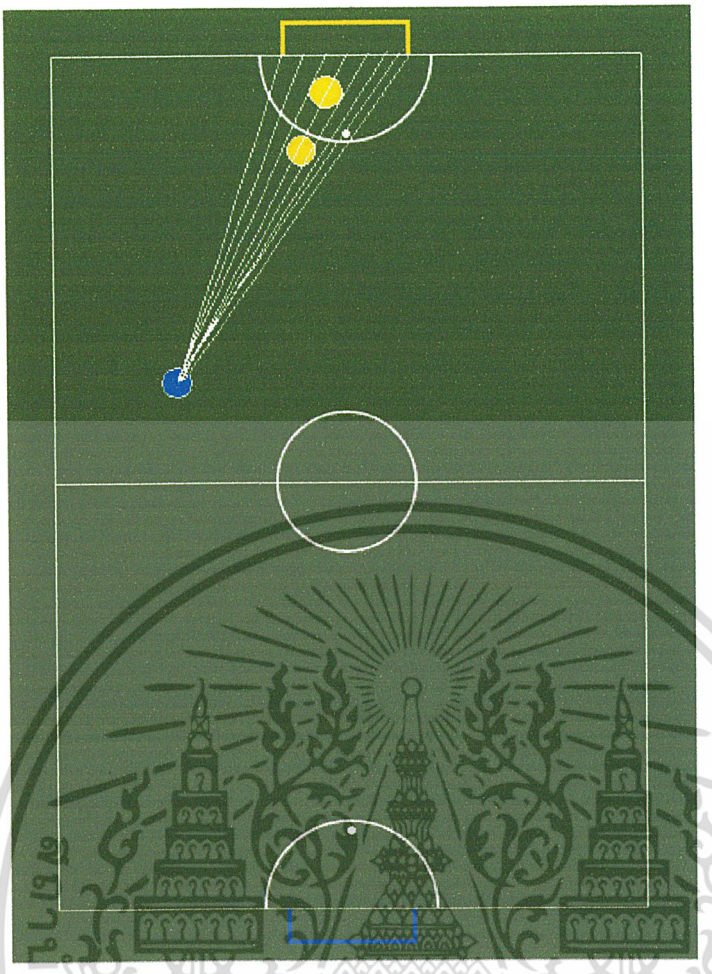
รูปที่3.3 ภาพแสดงการเคลื่อนที่DMCแนวอัลกอริธึมที่ 2 ขั้นตอนการเดินไปหาบอล

ขั้นตอนนี้ของอัลกอริธึมนี้คือการเข้าหาบอลจะเห็นว่าไม่ว่าหุ่นยนต์จะเคลื่อนที่ ไปหาบอลอย่างไรก็ตามจะสามารถพาบอลขึ้นไปฝั่งตรงข้ามได้

3.2.4 ขั้นตอนการออกแบบอัลกอริธึมของผู้เล่นตำแหน่งกองกลางและกองหน้า

หัวข้อนี้จะทำการรวมอัลกอริธึมของกองกลางกับกองหน้าเนื่องจากว่าจะมีความใกล้เคียงกันมากคล้ายๆกับตำแหน่งกองหลังกับผู้รักษาประตูที่ใกล้เคียงกันมาก โดยความแตกต่างจะอยู่ที่ผู้เล่นตำแหน่งกองกลางจะทั่วสนามและกองหน้านั้นการปรับความเร็วต่างๆอาจจะช้ากว่าหน่อยเนื่องจากสำหรับตำแหน่งกองหน้านั้นจะต้องการความปราณีตสูงกว่าและเนื่องจากว่าจะมีอัลกอริธึมบางส่วนที่เหมือนกับกองกลางตัวรับไปแล้วเพราะฉะนั้นเราจึงจะอธิบายเพียงส่วนที่กองกลางตัวรับไม่มีเท่านั้น

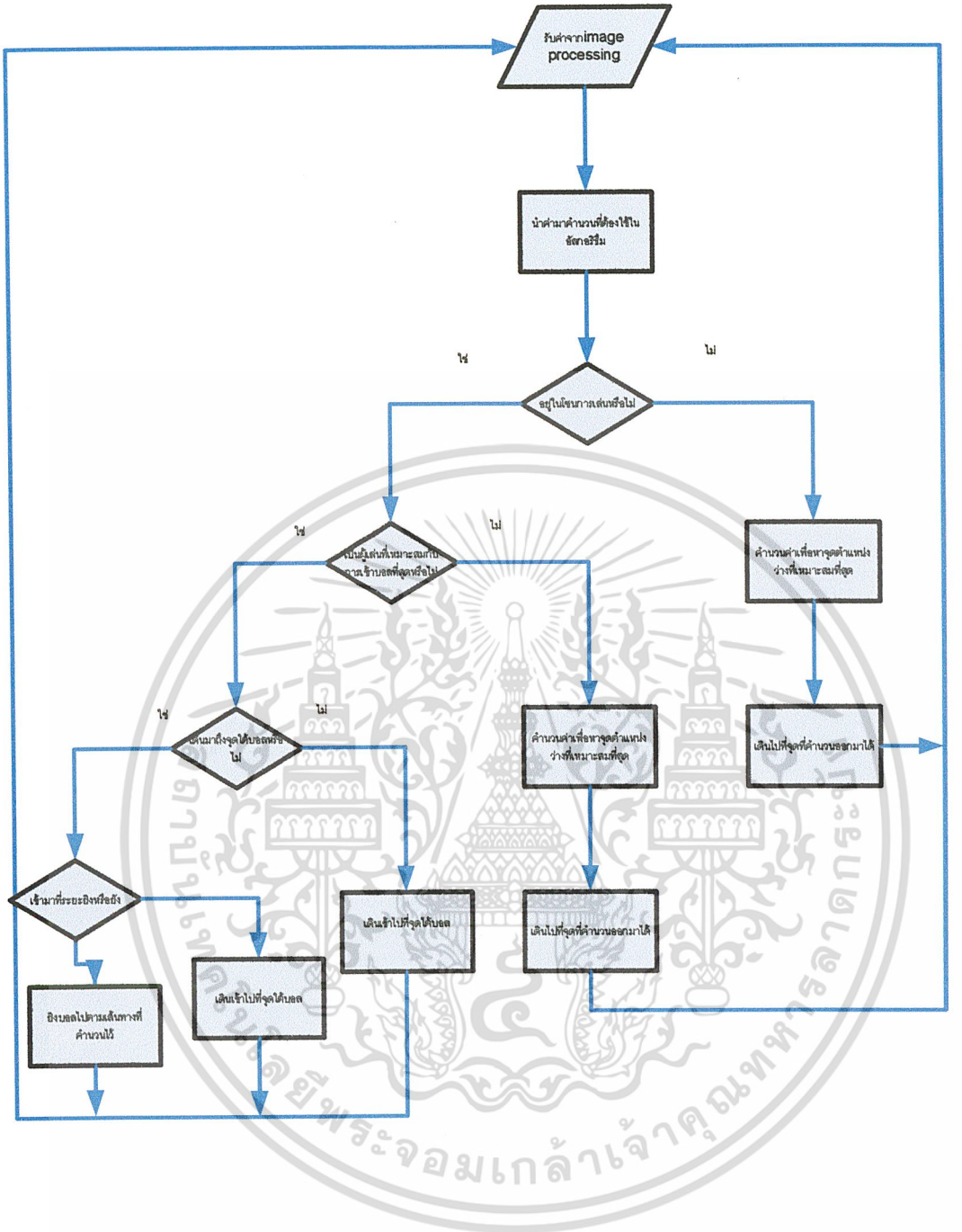
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.2.8 ภาพแสดงการเคลื่อนที่MC ขั้นตอนการคำนวนแนววิถีโจมตี

จะเป็นการสร้างสมการเส้นตรงขึ้นมาเพื่อหาหาเส้นทางการยิงเมื่อคำนวนได้แล้วก็จะทำการยิง จากที่กล่าวมาทั้งหมดนั้นจะเห็นว่าโดยรูปแบบอัลกอริธึมจริงๆนั้นจะมี 2 รูปแบบโดยจะแบ่งเป็น 2 ส่วนใหญ่ๆคืออัลกอริธึมการโจมตีกับการป้องกันโดยจะแสดงเป็นflow chart ได้ดังนี้

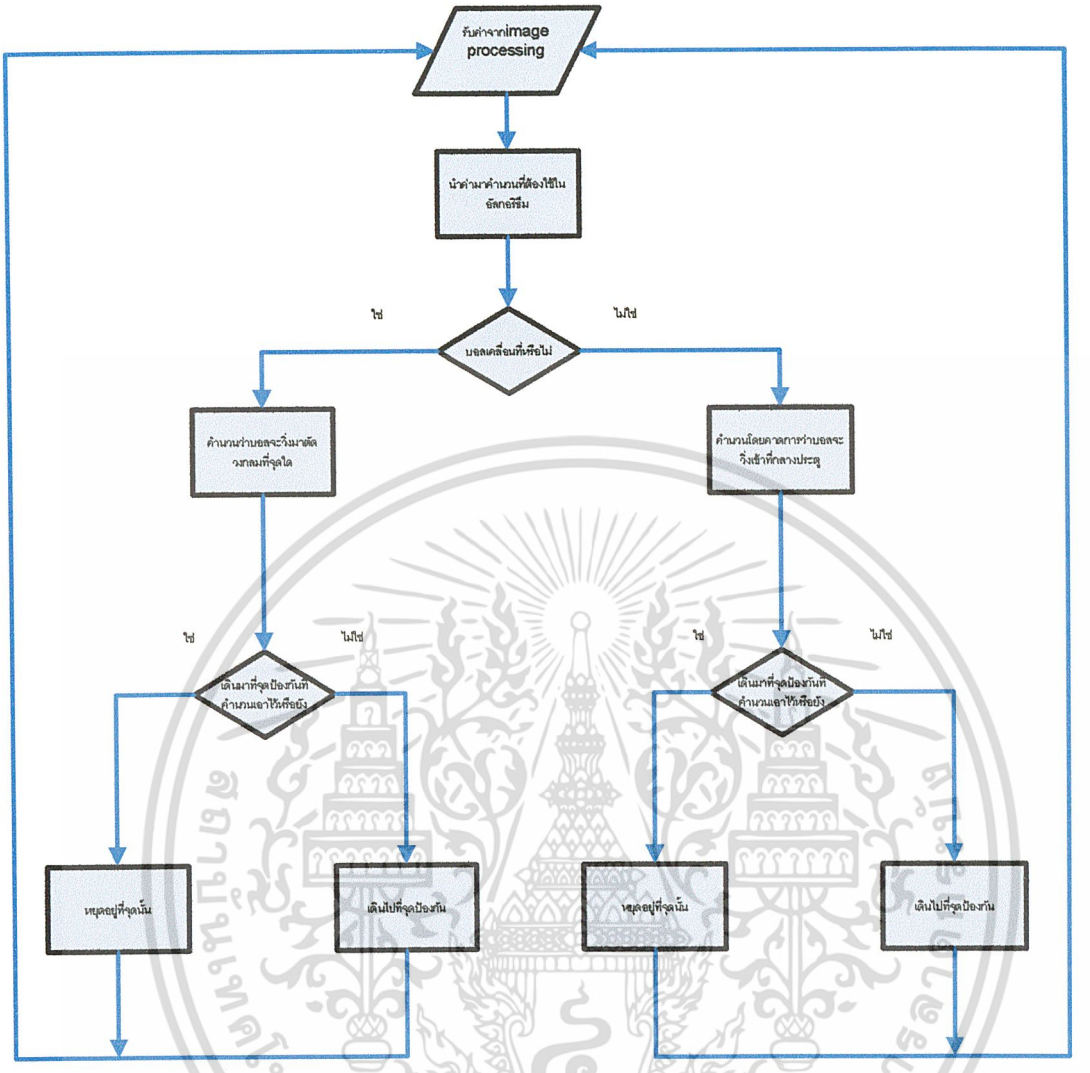
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.2.9 ภาพแสดงผังงานของอัลกอริธึมแนวรุก

จากผังงานที่เห็นจะเป็นการแสดงถึงอัลกอริธึมของแนวบุกโดยจะมีตำแหน่งที่ใช้อัลกอริธึมนี้ 3 ตำแหน่งด้วยกันคือ กองหน้า,กองกลาง,และ กองกลางตัวรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

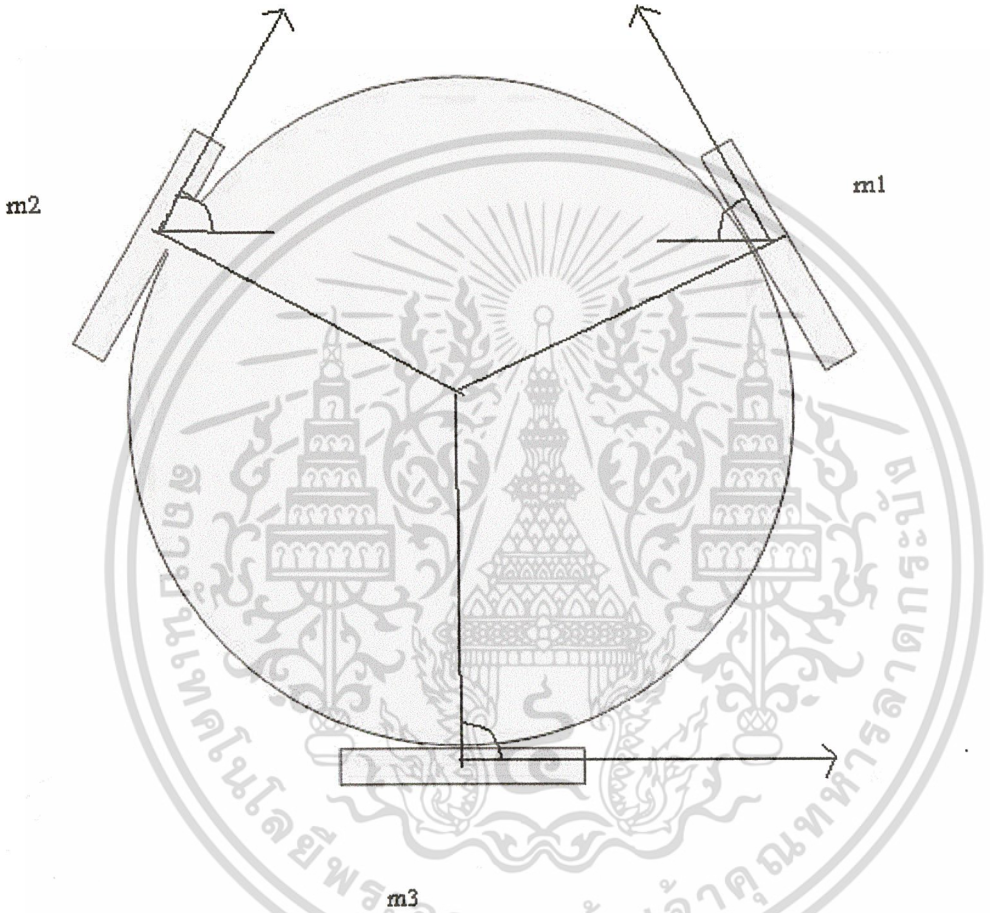


รูปที่3.2.10 ภาพแสดงอัลกอริธึมแนวรับ

จากผังงานที่เห็นจะเป็นการแสดงถึงอัลกอริธึมของแนวรับจะมีตำแหน่งที่ใช้อัลกอริธึมนี้ 2 ตำแหน่งด้วยกันคือ กองหลังและผู้รักษาประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนออกแบบของตัวจำลองสถาณการณ์นั้นก่อนอื่นเราจะต้องเข้าใจถึงตำแหน่งแรงของหุ่นยนต์ก่อน



รูปที่3.2.11 ภาพแสดงแนวแรงของล้อทั้ง3ของหุ่นยนต์

ซึ่งจากรูปนั้นเมื่อเราตทำการแตกแรงออกมาจะได้สมการเป็น

```
velo_y=(Math.cos(theta1)*binary_to_int(bits.substring(8,12))
      +Math.cos(theta2)*binary_to_int(bits.substring(12,16))
      +Math.cos(theta3)*binary_to_int(bits.substring(16,20)));
velo_x=(Math.sin(theta1)*binary_to_int(bits.substring(8,12))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
+Math.sin(theta2)*binary_to_int(bits.substring(12,16))
+Math.sin(theta3)*binary_to_int(bits.substring(16,20)));
```

Omega=Fm1+Fm2+Fm3

หมายเหตุ ค่าที่นำมาแสดงนำมาจาก code โปรแกรมโดยจะเป็นบิทที่นำมาใช้คิดผู้อ่านสามารถดูจาก protocol ในต้นบท 3 ได้

จากนั้นเราก็นำค่าที่ได้ไปใส่ให้กับหุ่นยนต์ในตัวจำลองสถาณการณ์ของเราเพื่อให้หุ่นยนต์เราวิ่งตามความเร็วแกน X,Y ที่คำนวนออกมาได้

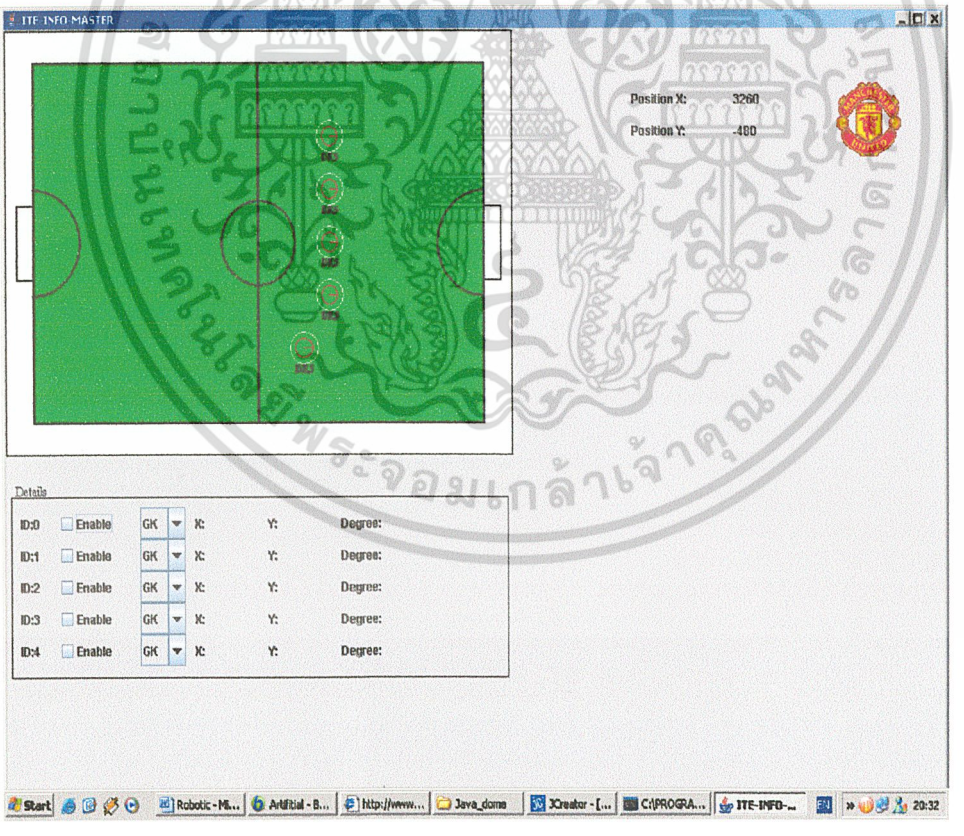
3.3 ปัญหาที่เกิดขึ้นระหว่างขั้นตอนการสร้างโปรแกรม

1. เนื่องจากการทดลองนี้ผลการทดลองที่แท้จริงจะอยู่กับตัวหุ่นยนต์เพราะฉะนั้นบางครั้งอัลกอริธึมแบบนึงจะเหมาะกับหุ่นรูปแบบหนึ่งเช่นถ้าเราใช้อัลกอริธึมที่เร็วแต่จะต้องการความแม่นยำจากฮารด์แวร์สูงก็จะไม่เหมาะกับฮารด์แวร์ที่ความแม่นยำต่ำเช่นหุ่นที่ไม่เป็น closeloop
2. เนื่องจากการทดลองนี้ได้ใช้ต้องใช้ image processing และหุ่นยนต์เพราะฉะนั้นความผิดพลาดบางจุดจะมาเนื่องจาก image processing และหุ่นยนต์เช่นแสดงเปลี่ยนจะจับพิกัดออกมาผิดหรือส่งสัญญาณไปแต่หุ่นยนต์รับสัญญาณไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 รายละเอียดและการใช้งานของโปรแกรม

1. สร้างเธรดขึ้นมาตามจำนวนตัวหุ่นยนต์
2. นำ byte stream ที่ได้จากimage processing มาทำการตัดส่วนหัวเพื่อเอามาเฉพาะส่วนต้องการก่อน
3. ขั้นแรกในการทำโปรแกรมนี้ จะเริ่มจากการแปลง byte stream เหล่านี้ให้อยู่ ในรูปตัวเลขซึ่ งความหมายของแต่ละ byte ผู้อ่านสามารถดูความหมายแต่ละ byte ได้จากต้นบทที่3
4. นำค่าได้เหล่านั้นมาคำนวนหาพารามิเตอร์พิเศษต่างๆที่ image processing ไม่ได้ส่งให้เช่นทิศการวิ่งของลูกบอล ระยะห่างจากบอล แนววิถีการโจมตี
5. นำค่าเหล่านั้นไปประมวลผลที่อัลกอริธึมต่างๆโดยที่แต่ละอัลกอริธึมก็อาจจะใช้พารามิเตอร์พิเศษที่สร้างขึ้นไม่เหมือนกัน m2
6. โปรแกรมวนลูปทำงานไปเรื่อยๆตามข้อ 1-5 จนกว่าจะทำลายการทำงานของเธรดทั้งหมด

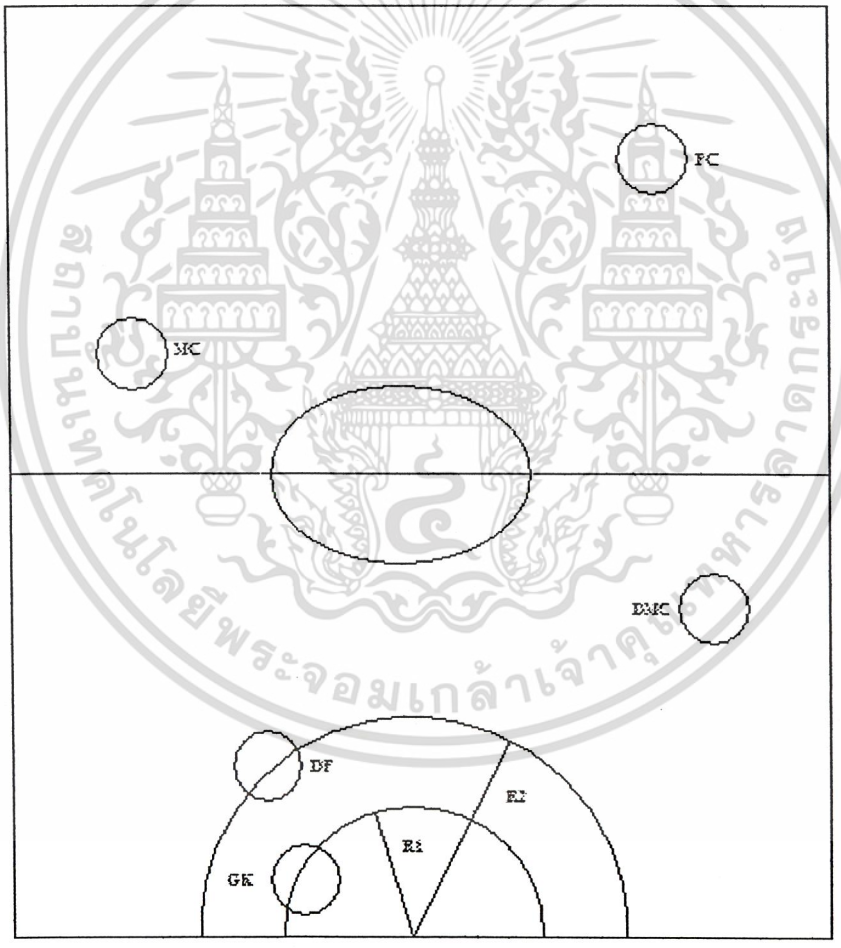


รูปที่ 3.4.1 แสดงโปรแกรมส่วน AI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
วิธีการดำเนินการทดลอง

อันดับแรกเราต้องทราบก่อนว่าระบบการทำงานของ SOCCER ROBOT ของกลุ่มเรานั้นมีศักยภาพอย่างไรเราจะมาอธิบายด้วยผังของสนามแข่ง และตำแหน่งต่างๆของ ROBOT แต่ละตัว เพราะตำแหน่งต่างๆของ ROBOTแต่ละตัวนั้น เป็นตัวกำหนดหน้าที่ของ ROBOT ว่าต้องทำอะไร ต้องเดินไปทางไหน ต้องป้องกันอย่างไร ต้องยิงเมื่อไหร่และวิธีดำเนินการทดลองของกลุ่มผมนั้น จะอธิบายถึงตำแหน่งต่างๆว่ามีความแม่นยำแค่ไหน โดยทำการทดลองอย่างจริงจังและทดลองเป็นจำนวนหลายๆครั้ง และนำมาหาค่าเฉลี่ย และคิดเป็น เปอร์เซนต์เพื่อที่จะง่ายต่อการอธิบายให้อาจารย์เข้าใจ



ภาพแสดงตำแหน่งต่างๆของ ROBOT ในสนาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1อธิบายตำแหน่งต่างๆของ ROBOT ในสนาม

GK ทำหน้าที่ดูแลรักษาประตู ที่พูดอย่างนี้ได้เพราะตำแหน่งนี้เป็นตำแหน่งที่อยู่ใกล้ประตูที่สุด และมีการสั่งการให้เคลื่อนที่ ตามรัศมี R1 โดยจะเคลื่อนไปบังบอลตามรัศมี R1 และอยู่ในครึ่งวงกลม(กรอบเขตโทษ)

DF ทำหน้าที่ดูแลรักษาประตูเช่นเดียวกันกับ GK แต่จะแตกต่างกันตรงที่ ตำแหน่งนี้นั้นจะเคลื่อนที่ตามรัศมี R2แต่ไม่สามารถเข้าไปอยู่ในครึ่งวงกลม(กรอบเขตโทษ)

DMC จะทำหน้าที่ในการไล่ติดตามบอลเฉพาะแดนหลังทั้งหมด แต่จะไม่เข้าไปในรัศมีของ R1

MC จะทำการไล่บอลทั้งหมดในสนามไม่ว่าจะเป็นแดนตนเองหรือ แดนฝ่ายตรงข้ามแต่จะไม่เข้าไปในรัศมีของ R1และเมื่อจับบอลได้จะทำการเลี้ยงบอลไปตำแหน่งที่เหมาะสมแล้วหุ่นยนต์จะหันหน้าเข้าหาประตูของฝ่ายตรงข้ามเมื่อตรงแล้วก็จะทำการยิงบอลทันทีโดยแนวคิดนี้นั้นคือยิ่งยิงบอลมากเท่าไรโอกาสที่จะได้ประตูก็ยิ่งมีสูงตามด้วย

FC ส่วนตำแหน่งนี้นั้นจะวิ่งไล่บอลในแดนของฝ่ายตรงข้ามเท่านั้นและเมื่อได้บอลแล้วจะทำการยิงทันที

4.2 ขั้นตอนการดำเนินการทดลอง

สมมุติฐาน

หุ่นยนต์ต้องมีความผิดพลาด

วิธีการดำเนินการทดลอง

การทดลองของกลุ่มผมนั้นจะเป็นการทดลองโดยการทำการโยนบอลลงในสนามในตำแหน่งที่ ROBOT แต่ละตัวรับผิดชอบเป็นตำแหน่งๆไป เพื่อดูว่าแต่ละตำแหน่งมีความถูกต้องแม่นยำแค่ไหนโดยจะทำการทดลองประมาณ 15 ครั้งแล้วทำการเทียบเป็นเปอร์เซนต์เพื่อง่ายต่อการอธิบาย

หมายเหตุ ถ้ามีสัญลักษณ์ **0** ที่ตรงไหนแสดงว่าเป็นสิ่งนั้น

และในการทดสอบGKนั้นจะทำการโยนบอลให้เข้าประตูเพื่อดูว่าGKสามารถป้องกันได้หรือไม่

โดยมี Algorithmในการทดลอง 2 แบบ

GKเมื่อใช้การป้องกันประตูเป็นรูปครึ่งวงกลม(R1)

จำนวนครั้ง	ไม่เข้า	เข้า	ทำไมจึงเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1	0		
2	0		
3	0		
4		0	หาลูกบอลไม่เจอ
5	0		
6		0	เคลื่อนที่ไปไม่ทัน
7	0		
8	0		
9	0		
10	0		
11	0		
12	0		
13	0		
14	0		
15	0		

คิดเป็นเปอร์เซนต์ความแม่นยำในการป้องกันประตู 86.67%

GKเมื่อใช้การป้องกันเป็นเส้นตรงแนวแกนX

จำนวนครั้ง	ไม่เข้า	เข้า	ทำไมจึงเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1		0	ป้องกันได้แต่เกิดการแฉลบเข้าประตู
2	0		
3	0		
4		0	ป้องกันได้แต่เกิดการแฉลบเข้าประตู
5	0		
6	0		
7	0		
8	0		
9		0	ป้องกันได้แต่เกิดการแฉลบเข้าประตู
10	0		
11		0	ป้องกันได้แต่เกิดการแฉลบเข้าประตู
12		0	ไม่ยอมเคลื่อนที่
13	0		
14		0	หาลูกบอลไม่เจอ
15	0		

คิดเป็นเปอร์เซนต์ความแม่นยำในการป้องกันประตู 59.99%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DF

จำนวนครั้ง	ไม่เข้า	เข้า	ทำไมจึงเข้า
1	0		
2	0		
3	0		
4	0		
5	0		
6	0		
7	0		
8	0		
9		0	หาลูกบอลไม่เจอ
10	0		
11	0		
12		0	ไม่ยอมเคลื่อนที่
13	0		
14	0		
15	0		

คิดเป็นเปอร์เซนต์ความแม่นยำในการป้องกันประตู 86.67%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DMC

จำนวนครั้ง	เจอบอล	ไม่เจอบอล	ทำไมไม่เจอบอล
1		0	หุ่นเคลื่อนที่ไม่ตรง
2	0		
3	0		
4	0		
5		0	หุ่นอยู่นิ่ง
6	0		
7	0		
8	0		
9	0		
10	0		
11	0		
12	0		
13	0		
14	0		
15	0		

คิดเป็นเปอร์เซนต์ความแม่นยำในการติดตามบอล 86.67%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลอง MC และ FC นั้นจะเป็นการทดลองโดยมีการจับเวลาในการทำประตูเป็นเวลา4นาที และมีสิ่งกีดขวาง

MC

จำนวนครั้ง	ยิงเข้า	ยิงไม่เข้า	ทำไมยิงไม่เข้า
1		O	หุ่นเคลื่อนที่ไม่ตรง
2	O		
3	O		
4	O		
5		O	หุ่นอยู่นิ่ง
6	O		
7	O		
8		O	เวลาหมด
9	O		
10		O	เวลาหมด
11	O		
12	O		
13		O	เวลาหมด
14		O	เวลาหมด
15	O		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คิดเป็นเปอร์เซนต์ความแม่นยำในการติดตามบอล 59.99

FC

จำนวนครั้ง	ยิงเข้า	ยิงไม่เข้า	ทำไมยิงไม่เข้า
1	O		
2	O		
3	O		
4	O		
5		O	ยิงไม่ตรง
6	O		
7		O	เวลาหมด
8	O		
9		O	เวลาหมด
10		O	เวลาหมด
11	O		
12	O		
13	O		
14	O		
15	O		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คิดเป็นเปอร์เซนต์ความแม่นยำในการติดตามบอล 73.33

4.3การทดลองด้าน simulator

การทดลองด้านนี้จะเป็นการทดสอบว่าการประมวลผลโดยให้ simulator แสดงผล กับการประมวลผลโดยให้หุ่นยนต์จริงๆแสดงผลนั้นแตกต่างกันอย่างไร

ผลการทดสอบนั้นจะทำการ Run Program โดยดูความแตกต่างว่าเป็นอย่างไรแล้วทำการบันทึกผล

4.4ผลนั้นมีความแตกต่างกันดังนี้

แสดงผลโดย simulator	แสดงผลโดย Soccer Robot
โปรแกรมสั่งการอะไร ผลออกมาใกล้เคียงกับที่สั่ง	มีความผิดเพี้ยนสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5

สรุปผลการทดลอง

5.1 สรุปผลการดำเนินการทดลอง

จะเห็นได้ว่าการทดลองจะแบ่งออกเป็น2ส่วน

ในส่วนแรกนั้นจะทดลองเกี่ยวกับความแม่นยำของหุ่นยนต์แต่ละตัวว่าแต่ละตำแหน่งมีความแม่นยำแค่ไหนโดยการ Run Program และมีการจับเวลาบ้างในบางตัว ซึ่งผลออกมาก็มีความผิดพลาดอยู่บ้าง ความผิดพลาดเกิดจากอะไรนั้นจะอธิบายในหัวข้อต่อไป

ในส่วนที่สองนั้นจะทดลองเกี่ยวกับ simulator ว่า simulator นั้นมีความแตกต่างจากหุ่นยนต์ในส่วนไหนบ้าง พบว่ามีความแตกต่าง

5.2 ปัญหาที่พบในระหว่างการทดลอง

ปัญหาอาจเกิดได้หลายกรณี

-ความไม่ชำนาญในการเขียน โปรแกรมภาษาจาวา(Java Programming) ซึ่งเป็นภาษาที่ใช้ในการสร้างและออกแบบโครงงาน

-การจับภาพของการทำ Image processing อาจจะจับภาพและทำการบอกตำแหน่งผิดพลาดทำให้หุ่นยนต์นั้นเกิดการเคลื่อนที่คลาดเคลื่อนเมื่อคลาดเคลื่อนแล้วก็จะทำให้หุ่นยนต์นั้นเตะได้ไม่ตรงเป้าหมายหรือป้องกันประตูได้ๆม่ดีเท่าที่ควร

-และอีกข้อที่ทำให้การควบคุมหุ่นยนต์นั้นไม่เที่ยงตรงก็อยู่ในส่วนของ Hardware การประกอบตัวหุ่นยนต์นั้นอาจจะล้อไม่ตรง ติดตั้งตัวฝีสไม่ดี มอเตอร์ไม่เสถียร แผงวงจรเกิดการผิดพลาดวิธีการที่จะแก้ไขนั้นก็คือ ต้องติดตั้งตัวหุ่นยนต์ด้วยความระมัดระวัง

-ในด้าน ของsimulator ที่แสดงผลออกมาแล้วไม่ตรงกับของหุ่นยนต์นั้น เพราะ simulator เขียน Program ตามหลักการของ Physic ผลที่ได้ย่อมมีความแม่นยำและแน่นอน ส่วนหุ่นยนต์นั้นที่ไม่เที่ยงตรงก็เนื่องมาจากส่วนของ Hardware ด้วย

5.3 แนวทางการพัฒนาโครงงาน

- ทำการพัฒนาเพิ่มเติมฟังก์ชัน ผู้ช่วยให้ผู้ใช้สามารถใช้งานได้หลากหลายมากขึ้น
- ทำ Image processing ให้จับภาพได้ดีขึ้น
- พยายามทำส่วนของตัวหุ่นยนต์ให้มีความเสถียรมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้