

การควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขาคู่ด้วยโครงข่ายประสาทเทียม  
แบบเซลล์ลาร์

THE CONTROLLING SIX-LEGGED ROBOTS MOVEMENTS VIA CELLULAR  
NEURAL NETWORK

ไกรมณ มณีพิลป์  
KRAIMON MANEESILP

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

ISBN 974-15-1076-4

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียม  
แบบเซลล์ลูลาร์

THE CONTROLLING SIX-LEGGED ROBOT'S MOVEMENTS VIA CELLULAR  
NEURAL NETWORK

ไกรมอน มณีศิลป์  
KRAIMON MANEESILP



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมสารสนเทศ  
บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พศ. 2547

ISBN 974-15-1076-4

เลขหมู่.....  
เลขทะเบียน..... 58512  
วัน,เดือน,ปี 25 ส.ค. 2549



THE CONTROLLING SIX-LEGGED ROBOT'S MOVEMENTS VIA CELLULAR  
NEURAL NETWORK

KRAIMON MANEESILP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF ENGINEERING IN INFORMATION ENGINEERING  
SCHOOL OF GRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

ISBN 974-15-1076-4

COPYRIGHT 2004

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

---

หัวข้อวิทยานิพนธ์      การควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขาคู่ด้วยโครงข่ายประสาทเทียม  
แบบเซลล์ลูลาร์  
THE CONTROLLING SIX-LEGGED ROBOT'S MOVEMENTS VIA  
CELLULAR NEURAL NETWORK

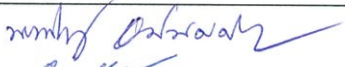



ชื่อนักศึกษา            นายไกรมน      มณีศิลป์

รหัสประจำตัว            45061102

ปริญญา                    วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา                วิศวกรรมสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์      รศ.ดร.ปิติเขต      ผู้รักษา

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.นภพินท์	อนันตรศิริชัย	
รศ.อรลาภ	แสงอรุณ	
รศ.ดร.วันชัย	วีรุจจา	
รศ.ดร.ปิติเขต	ผู้รักษา	

วัน/เดือน/ปี ที่สอบ 14 พฤษภาคม 2547 เวลา 14.15-16.15 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้องE 12-404)

  
บัณฑิตวิทยาลัยรับรองแล้ว  
  
(ผศ.ดร.จารุวัตร เจริญสุข)  
คณบดีบัณฑิตวิทยาลัย

วันที่.....14.....เดือน.....กรกฎาคม.....พ.ศ.....๒๕๔๗.....

หัวข้อวิทยานิพนธ์	การควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์
นักศึกษา	นายไกรมน มณีศิลป์
รหัสประจำตัว	45061102
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2547
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.ปิติเขต สุรักษา

### บทคัดย่อ

งานวิจัยนี้นำเสนอการออกแบบและการประยุกต์กับงานโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์เพื่อนำมาใช้ควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขา โดยทำให้โครงข่ายเซลล์ประสาทเทียมแบบเซลล์ลูลาร์เกิดปรากฏการณ์ฮอปโตเวฟขึ้น เนื่องจากการควบคุมในลักษณะนี้จะทำให้การเคลื่อนที่คล้ายกระบวนการเดินของแมลงอย่างเป็นธรรมชาติ จึงเหมาะสมกับการนำมาประยุกต์กับการควบคุมการเคลื่อนที่ของหุ่นยนต์แมลง 6 ขาได้ ผลจากการทดลองพบว่าวิธีการควบคุมดังกล่าวเป็นผลให้หุ่นยนต์สามารถเคลื่อนที่ในพื้นที่ลาดชันได้ถึง 20 องศาและใช้กำลังงานเฉลี่ยต่ำกว่าแบบใช้การเปิดตาราง

<b>Thesis Title</b>	The Controlling Six-legged Robot's Movements via Cellular Neural Network
<b>Student</b>	Mr. Kraimon Maneesilp
<b>Student ID.</b>	45061102
<b>Degree</b>	Master of Engineering
<b>Programme</b>	Information Engineering
<b>Year</b>	2004
<b>Thesis Adviser</b>	Assoc.Prof. Dr. Pitikhate Sooraksa

## ABSTRACT

This research presents a design and application of Cellular Neural Network (CNN) for controlling a six-legged robot's movements in forward motion. Because Cellular Neural Network to generate the Autowaves Phenomena in the network, the robot's movement then has a gait like that of a real insect. The phenomena can be applied to a six-legged robot motion controls. According to the experimental results, the purposed CNN algorithm shows satisfactory movement on 20-degree-slope terrain whereas the looked-up table one fails to move in the same condition, in comparison. In addition, the CNN architecture for the robot's motion outperforms the looked-up table counterpart in light of less power consumption.

## กิตติกรรมประกาศ

การทำวิทยานิพนธ์ครั้งนี้ สำเร็จลงได้ด้วยปัจจัยหลาย ๆ ด้าน ผู้เขียนขอกล่าวขอบพระคุณไว้ ณ ที่นี้คือ

- อ. พินัย จุลประภา แผนกอิเล็กทรอนิกส์ สถาบันเทคโนโลยีราชมงคล วช.พระนครเหนือ บุคคลแรกที่จุดประกายด้านช่างอิเล็กทรอนิกส์
- อ. นิวัตร จารูวาระกุล แผนกเทคนิคคอมพิวเตอร์ สถาบันเทคโนโลยีราชมงคล วช.พระนครเหนือ บุคคลแรกที่สอนเกี่ยวกับ Microcontroller ให้ผู้เขียน
- อ.วณพันธ์ ้วยวุฒิ แผนกเทคนิคคอมพิวเตอร์ สถาบันเทคโนโลยีราชมงคล วช.พระนครเหนือ ผู้ชี้ให้เห็นถึงศักยภาพ ซึ่งผู้เขียนเองไม่เคยเห็นมาก่อน
- อ.รังสรรค์ วงศ์สรรค์ ภาควิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี ผู้แสดงมุมมองที่แตกต่างและมีผลอย่างมากกับผู้เขียนในเวลาต่อมา
- ดร.วรวัดณ์ ลิ้มโกคา ดร.บุญธีร์ เครือตราฐ และ อ.กฤตวัน ศิริบุญรณ์ ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าฯลาดกระบัง ผู้แนะนำวิธีการทำงานที่ดีแก่ผู้เขียน ซึ่งพัฒนาประสิทธิภาพของผู้เขียนขึ้นเป็นอย่างมาก
- อ.กฤตากร กล่อมการ ภาควิชาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าฯลาดกระบัง ผู้ให้คำแนะนำในงานวิจัยนี้ หลาย ๆ ด้าน
- อ.บุญยชนะ ภูระหงษ์ ภาควิชาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าฯลาดกระบัง อาจารย์และเพื่อนที่ดี และให้โอกาสมาตลอด
- อนุรักษ์ จันทร์ศรี เพื่อนและน้องผู้ให้แรงผลักดันในการทำงาน

และที่ลืมไม่ได้ รศ.ดร.ปิติเขต สุรักษา อาจารย์ผู้ควบคุมวิทยานิพนธ์นี้ ผู้ซึ่งให้ความเชื่อมั่นในตัวผู้เขียนจนบางครั้งมากกว่าที่ผู้เขียนควรจะได้ และสุดท้าย ขอขอบพระคุณ บิดามารดา ที่ทำให้ผู้เขียนได้พบโอกาสอันดีเลิศเช่นนี้

การทำวิทยานิพนธ์นี้ได้รับการสนับสนุนจาก งบประมาณอุดหนุนการวิจัยสำหรับนักศึกษา ระดับบัณฑิตศึกษาประจำปีงบประมาณ 2547 เพื่อเป็นค่าใช้จ่ายสนับสนุนการทำวิทยานิพนธ์ จึงขอขอบคุณมา ณ โอกาสนี้ด้วย

ไกรมน มณีศิลป์

# สารบัญ

	หน้า
บทคัดย่อ .....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง .....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย .....	2
1.3 สมมติฐานของการวิจัย .....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย .....	2
1.5 ขอบเขตการวิจัย.....	3
1.6 ขั้นตอนของการวิจัย .....	3
บทที่ 2 การควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขา.....	4
2.1 โครงสร้างหุ่นยนต์ 6 ขา.....	4
2.2 การเดินแบบขั้นตอนวิธีการเปิดตารางของหุ่นยนต์ 6 ขา.....	6
2.2.1 การหารูปแบบการเคลื่อนที่.....	6
2.2.2 รูปแบบการเคลื่อนที่ในรูปตารางสถานะ.....	7
2.2.3 การสร้างชุดควบคุมจากตารางสถานะ .....	8
2.3 ความสามารถและข้อจำกัด.....	9
2.4 สรุปท้ายบท .....	9
บทที่ 3 โครงข่ายประสาทเทียมแบบเซลล์ูลาร์ .....	10
3.1 ทฤษฎีโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ .....	10
3.2 ปรัชญาการณ้อโตเวฟ .....	13
3.3 การควบคุมการเคลื่อนที่โดยอาศัยอโตเวฟ.....	15
3.4 ผลทดลองการปรับค่าพารามิเตอร์ต่าง ๆ ของระบบ .....	17

## สารบัญ (ต่อ)

	หน้า
3.5 การควบคุมหุ่นยนต์ โดยการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลาร์ในคอมพิวเตอร์ส่วนบุคคล .....	23
3.5.1 ส่วนโปรแกรมจำลองการทำงานโครงข่ายประสาทเทียมแบบเซลล์ลาร์ในคอมพิวเตอร์.....	24
3.5.2 ส่วนควบคุมหุ่นยนต์ผ่าน RS232C .....	25
3.6 สรุปท้ายบท .....	25
บทที่ 4 การออกแบบวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลาร์และการประยุกต์ .....	26
4.1 แนวคิดในการออกแบบวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลาร์ .....	26
4.2 วงจรจำลองเซลล์ควบคุมการก้าวขา .....	32
4.3 วงจรจำลองเซลล์ควบคุมการยกขา.....	33
4.4 ผลการทดสอบวงจร.....	34
4.4.1 สัญญาณจากวงจรโครงข่ายประสาทเทียมแบบเซลล์ลาร์แบบ 2 เซลล์ .....	34
4.4.2 สัญญาณจากวงจรโครงข่ายประสาทเทียมแบบเซลล์ลาร์เพื่อควบคุมหุ่นยนต์...36	
4.4.3 การนำไปควบคุมหุ่นยนต์จริง .....	42
4.5 สรุปท้ายบท .....	43
บทที่ 5 ผลทดลองการทดสอบการเคลื่อนที่.....	44
5.1 ผลการควบคุมหุ่นยนต์ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลาร์กับวิธีเปิดตาราง ....	44
5.1.1 พฤติกรรมการไต่ขึ้นทางลาดชัน .....	44
5.1.2 การใช้พลังงานในการเคลื่อนที่ .....	47
5.2 สรุปท้ายบท .....	55
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ .....	56
เอกสารอ้างอิง .....	58

## สารบัญ (ต่อ)

	หน้า
ภาคผนวก ก.....	60
โปรแกรมควบคุมหุ่นยนต์ 6 ขาด้วยขั้นตอนวิธีการเปิดตาราง.....	60
วงจรควบคุมหุ่นยนต์ 6 ขาด้วยขั้นตอนวิธีการเปิดตาราง.....	69
ภาคผนวก ข.....	71
โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์รูปที่ 3.5 .....	71
โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์รูปที่ 3.6 .....	72
โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์รูปที่ 3.8-3.9 .....	73
โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์รูปที่ 3.10-3.11 ..	74
โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์รูปที่ 3.12-3.13 ..	75
โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์รูปที่ 3.14-3.15 ..	76
ภาคผนวก ค.....	77
โปรแกรมจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ (CNN Simulator Ver.1.0) .....	77
โปรแกรมควบคุมหุ่นยนต์ 6 ขาผ่าน RS 232C .....	98
ภาคผนวก ง .....	104
การแปลงสัญญาณแอนะล็อกเป็นสัญญาณควบคุมเซอร์โวมอเตอร์ .....	104
วงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์.....	106
ประวัติผู้เขียน.....	107

# สารบัญตาราง

ตารางที่	หน้า
2. 1 สถานะสำหรับการควบคุมการเดินแบบขั้นตอนวิธีการเปิดตาราง .....	7
3. 1 เอกลักษณ์ของเซลล์ที่ใช้ควบคุมหุ่นยนต์ 6 ขาในตำแหน่งต่าง ๆ .....	23

# สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างหุ่นยนต์ 6 ขา มองจากด้านบน.....	4
2.2 โครงสร้างหุ่นยนต์ 6 ขา มองจากด้านข้าง.....	5
2.3 โครงสร้างหุ่นยนต์ 6 ขา มองจากด้านหน้า.....	5
2.4 ชั้นตอนต่าง ๆ ในการควบคุมการเดินแบบขั้นตอนวิธีการเปิดตาราง .....	6
2.5 แผนภูมิสถานะของการควบคุมการเดินแบบขั้นตอนวิธีการเปิดตาราง .....	7
2.6 ชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาแบบขั้นตอนวิธีการเปิดตาราง.....	8
2.7 ชุดวงจรควบคุมสำเร็จรูปที่นำมาใช้ในการควบคุมเซอร์โวมอเตอร์.....	8
3.1 วงจรสมมูลของเซลล์.....	10
3.2 คุณสมบัติสมการเอาท์พุท ของโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ .....	11
3.3 ขนาดของ $N_r$ เมื่อ $r = 1, 2$ และ $3$ ตามลำดับ.....	12
3.4 แผนภูมิของโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ขนาด 2 เซลล์ที่เกิดสภาวะอลวน .....	13
3.5 เอาท์พุท ( $y_n$ ) ที่ได้จากสมการ (11), (12) เมื่อค่า $I_j = 0$ โดย $j = 1, 2, 3, \dots, 6$ .....	14
3.6 เอาท์พุท ( $y_n$ ) ที่ได้จากสมการ (13), (14) เมื่อค่า $I_j = 0$ โดย $j = 1, 2, 3, \dots, 6$ .....	15
3.7 แผนภูมิของโครงข่าย CNN สำหรับควบคุมหุ่นยนต์หกขา.....	16
3.8 เอาท์พุทของเซลล์ที่ 1 เมื่อกำหนดค่า $I_r$ เท่ากับ 0 และเปลี่ยนค่า $\mu$ จาก 0.4 ถึง 0.7 .....	17
3.9 เอาท์พุทของเซลล์ที่ 1 ให้แกน X เป็นแกนเวลาเทียบกับแกน Y คือ $\mu$ .....	17
3.10 เอาท์พุทของเซลล์ที่ 1 เมื่อกำหนดค่า $\mu$ เท่ากับ 0.65 และเปลี่ยนค่า $I_r$ จาก -0.8 ถึง 0.8... ..	18
3.11 เอาท์พุทของเซลล์ที่ 1 ให้แกน X เป็นแกนเวลาเทียบกับแกน Y คือ $I_r$ .....	18
3.12 เอาท์พุทของเซลล์ที่ 1 เมื่อกำหนดค่า $I_m$ เท่ากับ 0 และเปลี่ยนค่า $\beta$ จาก 0 ถึง 5 .....	19
3.13 เอาท์พุทของเซลล์ที่ 1 ให้แกน X เป็นแกนเวลาเทียบกับแกน Y คือ $\beta$ .....	20
3.14 เอาท์พุทของเซลล์ที่ 1 เมื่อกำหนดค่า $\beta$ เท่ากับ 4 และเปลี่ยนค่า $I_m$ จาก -2.5 ถึง 2.5 .....	20
3.15 เอาท์พุทของเซลล์ที่ 1 ให้แกน X เป็นแกนเวลาเทียบกับแกน Y คือ $I_r$ .....	21
3.16 ผลการเคลื่อนที่ของขาหุ่นยนต์.....	21
3.17 การจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ที่ออกแบบไว้เพื่อทดสอบ ค่าพารามิเตอร์.....	24
3.18 โปรแกรมจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ูลาร์บนคอมพิวเตอร์.....	24
4.1 วงจรขยายผลต่างส่วนหน้าของวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ูลาร์.....	26
4.2 วงจรขยายผลบวกส่วนหน้าของวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ .....	27

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3 วงจรเชิงเส้นอิมิตัวสำหรับสมการเอาท์พุด.....	29
4.4 ปัญหาจุดเชื่อมต่อของเอาท์พุดส่วนที่ 1 กับอินพุตส่วนที่ 2.....	29
4.5 วงจรจำลองโครงข่ายประสาทเทียมแบบเซลลูลาร์แบบ 2 เซลล์.....	30
4.6 วงจรจำลองโครงข่ายประสาทเทียมแบบเซลลูลาร์แบบ 2 เซลล์ที่ออกแบบใหม่.....	31
4.7 วงจรจำลองเซลล์ควบคุมการก้าวขาทั้ง 6 ขา.....	32
4.8 วงจรจำลองเซลล์ควบคุมหุ่นยนต์ 6 ขาทั้งหมด.....	33
4.9 เอาท์พุดของวงจรจำลองโครงข่ายประสาทเทียมแบบเซลลูลาร์แบบ 2 เซลล์.....	34
4.10 เอาท์พุดในภาวะเริ่มต้นของวงจรจำลองเซลล์ 2 เซลล์.....	34
4.11 เอาท์พุดในภาวะลดไฟเลี้ยงของวงจร.....	35
4.12 เอาท์พุดในภาวะตัดไฟเลี้ยงของวงจรชั่วคราว.....	36
4.13 เอาท์พุดของเซลล์ก้าวขา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 2.....	36
4.14 เอาท์พุดของเซลล์ก้าวขา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 3.....	37
4.15 เอาท์พุดของเซลล์ก้าวขา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 4.....	37
4.16 เอาท์พุดของเซลล์ก้าวขา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 5.....	38
4.17 เอาท์พุดของเซลล์ก้าวขา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 6.....	38
4.18 สัญญาณเอาท์พุดที่ขาหุ่นยนต์ด้านหน้าซ้าย.....	39
4.19 สัญญาณเอาท์พุดที่ขาหุ่นยนต์ด้านหน้าขวา.....	39
4.20 สัญญาณเอาท์พุดที่ขาหุ่นยนต์ตรงกลางด้านซ้าย.....	40
4.21 สัญญาณเอาท์พุดที่ขาหุ่นยนต์ตรงกลางด้านขวา.....	40
4.22 สัญญาณเอาท์พุดที่ขาหุ่นยนต์ด้านหลังซ้าย.....	41
4.23 สัญญาณเอาท์พุดที่ขาหุ่นยนต์ด้านหลังขวา.....	41
4.24 ชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลลูลาร์.....	42
4.25 วงจรจำลองการทำงานโครงข่ายประสาทเทียมแบบเซลลูลาร์สำหรับหุ่นยนต์ 6 ขา.....	42
5.1 (ก) - (ข) ลำดับขั้นตอนการเคลื่อนที่บนทางลาดชันประมาณ 20° โดยการเคลื่อนที่แบบ ขั้นตอนวิธีเปิดตาราง.....	45
5.2 (ก) - (ข) ลำดับขั้นตอนการเคลื่อนที่บนทางลาดชันประมาณ 20° ด้วยโครงข่ายประสาท เทียมแบบเซลลูลาร์.....	46

## สารบัญญรูป (ต่อ)

รูปที่	หน้า
5.3 วงจรตรวจวัดกระแสด้วยสโคป .....	47
5.4 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยขั้นตอนวิธีการเปิดตารางที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์ 30° .....	48
5.5 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลลูลาร์ที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์ 30° .....	48
5.6 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยขั้นตอนวิธีการเปิดตารางที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์ 60° .....	49
5.7 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลลูลาร์ที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์ 60° .....	49
5.8 วงจรเก็บข้อมูลกระแสทางสถิติ .....	50
5.9 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย $\bar{P}$ ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วยวิธีเปิดตารางกรณียกขาลอย โดยเก็บข้อมูลที่ (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง .....	51
5.10 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย $\bar{P}$ ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วย CNN กรณียกขาลอย โดยเก็บข้อมูลที่ (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง .....	52
5.11 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย $\bar{P}$ ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วยวิธีเปิดตารางกรณีเดินที่พื้น โดยเก็บข้อมูลที่ (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง .....	53
5.12 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย $\bar{P}$ ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วย CNN กรณีเดินที่พื้น โดยเก็บข้อมูลที่ (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง .....	54
ก.1 วงจรของชุดควบคุม CP-SPI/RD2 ของบริษัท ETT จำกัด .....	69
ก.2 ขั้วต่อใช้งานอุปกรณ์ภายนอกของชุดควบคุม CP-SPI/RD2 .....	70
ก.3 การต่อเซอร์โวมอเตอร์กับชุดควบคุม CP-SPI/RD2 .....	70
ง.1 สัญญาณที่ใช้ควบคุมเซอร์โวมอเตอร์ ในอุปกรณ์เครื่องเล่นวิทยุบังคับ โดยค่าของ $a = 0.5 \text{ ms}$ , $b = 0-2 \text{ ms}$ ขึ้นอยู่กับมุมที่ต้องการและ $c > 20 \text{ ms}$ .....	104
ง.2 สัญญาณที่สร้างขึ้นมาเปรียบเทียบกับสัญญาณ CNN โดยมีค่าของ $a = 0.5 \text{ ms}$ , $b = 2.5 \text{ ms}$ และ $c = 20 \text{ ms}$ .....	104
ง.3 วงจรรวมทั้งหมดของชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลลูลาร์ .....	106

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

การออกแบบหุ่นยนต์โดยอาศัยพฤติกรรมเคลื่อนที่ ตามแบบธรรมชาติของสิ่งมีชีวิตนั้น เป็นการออกแบบที่น่าสนใจเป็นอย่างมาก [12][13] เนื่องจากการเคลื่อนที่โดยธรรมชาติของสิ่งมีชีวิตจะมีความกลมกลืนกับวัตถุประสงค์การใช้งานเป็นอย่างมาก เพราะจะต้องผ่านการคัดเลือกตามธรรมชาติมาเป็นเวลานานแล้วว่า เหมาะสมกับลักษณะสภาพแวดล้อมและภูมิประเทศแบบนั้น ๆ เป็นอย่างดี [14][15] มิฉะนั้นก็อาจสูญพันธุ์ไปตามกาลเวลา ดังนั้นถ้าสามารถลอกเลียนแบบการเคลื่อนที่ตามธรรมชาติได้ ก็จะได้รับ การถ่ายทอดความสามารถพิเศษตามการเคลื่อนที่แบบนั้น ๆ มาด้วย [16][17]

แต่การควบคุมการเคลื่อนที่ของหุ่นยนต์เลียนแบบธรรมชาติ มักอาศัยวิธีการควบคุมแบบขั้นตอนวิธีการเปิดตาราง (Lookup Table Algorithm) ซึ่งมีลักษณะที่ไม่เหมือนจริงอยู่มาก ทำให้คุณสมบัติพิเศษบางอย่างของการเคลื่อนที่ตามแบบธรรมชาติขาดหายไป ซึ่งการจะกระทำให้การเคลื่อนที่แบบขั้นตอนวิธีการเปิดตารางเคลื่อนที่ให้เหมือนจริงมากขึ้นนั้น จำเป็นจะต้องอาศัยความสามารถของการคำนวณทางคอมพิวเตอร์อย่างหลีกเลี่ยงไม่ได้ แต่ข้อเสียข้อหนึ่งของคอมพิวเตอร์คือ เมื่อระบบได้รับสัญญาณรบกวนมากในระดับหนึ่งระบบจะอยู่ในสภาวะหยุดชะงัก(Halt) เนื่องจากสัญญาณรบกวนดังกล่าวไปรบกวนลำดับขั้นการทำงานของคอมพิวเตอร์ ดังนั้นจึงอาจกล่าวได้ว่า หุ่นยนต์ที่อาศัยวิธีการควบคุมแบบขั้นตอนวิธีการเปิดตารางโดยใช้คอมพิวเตอร์นั้น จะมีความทนทานต่อสัญญาณรบกวนต่ำ จากปัญหาดังกล่าว ถ้าสามารถหลีกเลี่ยงวิธีการควบคุมแบบขั้นตอนวิธีการเปิดตารางไปใช้ วิธีการที่ไม่จำเป็นต้องอาศัยคอมพิวเตอร์ (Von Neumann Architecture) ในการควบคุมการเคลื่อนที่ได้ การทนต่อสัญญาณรบกวนของระบบก็จะมากขึ้น ผลคือสามารถนำหุ่นยนต์นั้น ๆ ไปใช้งานในที่ซึ่งมีสัญญาณรบกวนสูงได้

ปัจจุบันได้มีการศึกษาการควบคุมการเคลื่อนที่ของเซลล์ประสาทของสิ่งมีชีวิต และได้ทำการจำลองมาใช้ในการควบคุมหุ่นยนต์ [18] วิธีนี้เรียกว่าการควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ เนื่องจากเป็นการเลียนแบบวิธีการควบคุมการเคลื่อนที่ของสิ่งมีชีวิตจริงจึงมีลักษณะคล้ายกับการเคลื่อนที่ตามธรรมชาติมากกว่าวิธีการควบคุมแบบเปิดตาราง อีกทั้งยังสามารถจำลองการทำงานมาอยู่ในวงจรแอนะล็อกได้อีกด้วย ทำให้ระบบสามารถทนทานต่อสัญญาณรบกวนได้ดีกว่า

ในการวิจัยนี้จะทดลองการควบคุมการเคลื่อนที่ของแมลง 6 ขา โดยทดสอบเปรียบเทียบข้อดีข้อเสียของทั้งสองวิธี โดยได้ทำการออกแบบทั้งระบบโครงสร้างเซลล์แบบใหม่ และใช้วงจรในการจำลองเซลล์แบบใหม่อีกด้วย

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย

งานวิจัยนี้ มีจุดมุ่งหมายหลักเพื่อศึกษาและพัฒนาวิธีการควบคุมการเคลื่อนที่โดยอาศัยหลักการทางของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ จากนั้นนำมาเปรียบเทียบกับวิธีควบคุมแบบขั้นตอนวิธีการเปิดตารางเพื่อศึกษาข้อดีข้อเสียของแต่ละวิธี โดยนำไปทดลองใช้กับการควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขา เพื่อเป็นกรณีศึกษา

## 1.3 สมมติฐานของการวิจัย

การควบคุมการเคลื่อนที่โดยอาศัยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์นั้น คือการเลียนแบบการทำงานของระบบประสาทรอบนอกของสิ่งมีชีวิต ซึ่งบางส่วนมีการทำงานคล้ายคลึงส่งต่อกันเป็นช่วง ๆ ไป เรียกปรากฏการณ์นี้ว่า ปรากฏการณ์ Autowaves รูปแบบของคลื่นนี้เป็นลักษณะเฉพาะตัว ขึ้นอยู่กับรูปแบบการตั้งค่าพารามิเตอร์ของสมการที่ใช้ในการควบคุม วิธีการควบคุมดังกล่าวอาศัยหลักการของวงจรแบบไม่เป็นเชิงเส้นนำมาป้อนกลับเข้าหากัน ดังนั้นจึงสามารถอาศัยอุปกรณ์ทางแอนะล็อกในการควบคุมแทนคอมพิวเตอร์ได้

วิธีการที่นำมาศึกษานี้เมื่อเปรียบเทียบกับวิธีการเดิมในเบื้องต้นปรากฏว่า การเคลื่อนที่ของหุ่นยนต์มีลักษณะคล้ายแมลงมากกว่าวิธีการเดิม ในกรณีหุ่นยนต์ 6 ขา เมื่อควบคุมหุ่นยนต์ให้ไต่ขึ้นพื้นที่ลาดชันโดยอาศัยอัลกอริทึมการเดินด้วยตารางเวลานั้น ที่ระดับความชันค่าหนึ่งหุ่นยนต์จะมีการลื่นไถลลงมาตรง ๆ แต่เมื่อนำหุ่นยนต์ที่อาศัยการจำลองจากโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ในการควบคุมที่ระดับความสูงเดียวกัน ผลปรากฏว่าการลื่นไถลของหุ่นยนต์จะมีลักษณะเป็นการเบี่ยงเบนไปในด้านข้างด้านใดด้านหนึ่ง จนกระทั่งระดับแนวจุดศูนย์ถ่วงปรับได้ระดับที่เหมาะสมหุ่นยนต์จะสามารถเคลื่อนที่ไปได้ หุ่นยนต์ก็จะไต่ขึ้นไปในลักษณะดังกล่าว จากคุณสมบัตินี้ทำให้เห็นว่า หุ่นยนต์มีข้อจำกัดในการเคลื่อนที่น้อยลง

## 1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

อาศัยทฤษฎีการเกิดปรากฏการณ์ออโตเวฟในโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ มาสร้างเป็นสัญญาณควบคุมหุ่นยนต์ โดยศึกษาจากงานวิจัยที่ผ่านมาแล้ว จากนั้นนำไปพัฒนาให้ง่ายต่อการออกแบบมากขึ้น

อย่างไรก็ตาม เพื่อศึกษาถึงข้อดีข้อเสียของการควบคุมการเคลื่อนที่ด้วยวิธีดังกล่าวได้ถูกต้อง จึงจำเป็นต้องจำลองการทำงานทั้งในโปรแกรมจำลองเช่น โปรแกรม MATLAB การจำลองโดยพัฒนาโปรแกรมขึ้นบนคอมพิวเตอร์ส่วนบุคคล แล้วส่งข้อมูลเอาท์พุทที่คำนวณได้ไปยังชุดควบคุมหุ่นยนต์ ตลอดจนพัฒนาวงจรแอนะล็อกที่ทำหน้าที่จำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ขึ้นมา ทั้งนี้จะต้องพัฒนาการควบคุมการเคลื่อนที่โดยขั้นตอนวิธีการเปิดตารางขึ้นเพื่อใช้ในการเปรียบเทียบอีกด้วย

## 1.5 ขอบเขตการวิจัย

ทำการออกแบบและทดสอบโครงสร้างของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ และศึกษาผลกระทบจากค่าพารามิเตอร์ต่าง ๆ ของระบบ นำมาออกแบบวงจรแอนะล็อกเพื่อใช้ควบคุมหุ่นยนต์ 6 ขา และเปรียบเทียบหุ่นยนต์ 6 ขาที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์กับการควบคุมแบบขั้นตอนวิธีการเปิดตาราง ในด้านพฤติกรรมการไต่ขึ้นทางลาดและพฤติกรรมการใช้พลังงานในการเคลื่อนที่

## 1.6 ขั้นตอนของการวิจัย

มีขั้นตอนการศึกษาดังนี้

1. ศึกษาความเป็นไปได้ของการควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขาด้วยอาศัยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์
2. พัฒนาหุ่นยนต์ 6 ขาที่ใช้การควบคุมแบบขั้นตอนวิธีการเปิดตาราง เพื่อการเปรียบเทียบ
3. จำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ด้วยคอมพิวเตอร์ เพื่อศึกษาผลกระทบของค่าพารามิเตอร์ต่าง ๆ ต่อระบบ
4. จำลองโครงข่ายประสาทเทียมด้วยโปรแกรมคอมพิวเตอร์ โดยใช้พารามิเตอร์ที่ได้จากการจำลองการทำงาน ไปควบคุมหุ่นยนต์ 6 ขาเพื่อให้เคลื่อนที่ตามการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์
5. พัฒนางจรแอนะล็อกเพื่อใช้ควบคุมหุ่นยนต์ โดยอาศัยพารามิเตอร์เบื้องต้นจากการจำลองการทำงานในขั้นตอนที่ 3 เพื่อใช้ควบคุมหุ่นยนต์ 6 ขาโดยตรง
6. ทดสอบเปรียบเทียบระหว่างการควบคุมหุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ กับการควบคุมแบบขั้นตอนวิธีการเปิดตาราง โดยใช้ชุดขับเคลื่อนหุ่นยนต์ 6 ขาชุดเดียวกัน
7. สรุปผลและวิจารณ์

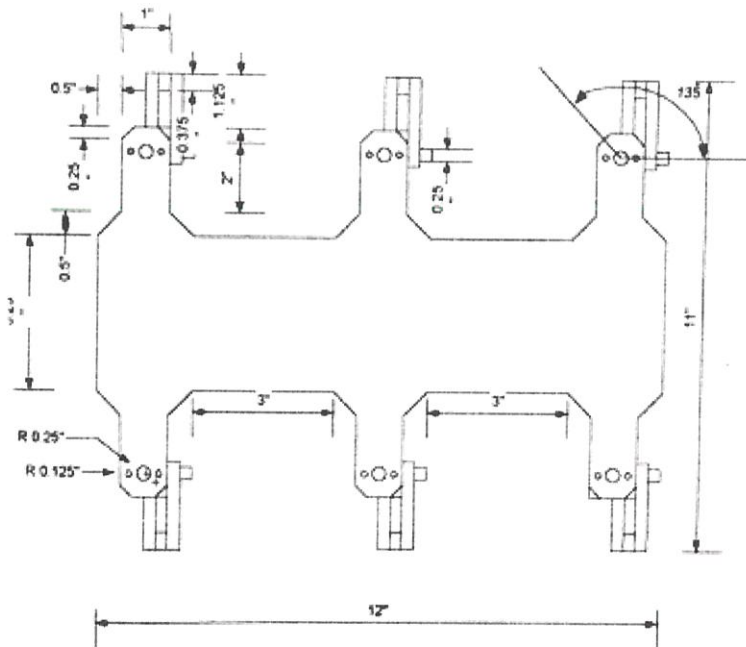
## บทที่ 2

### การควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขา

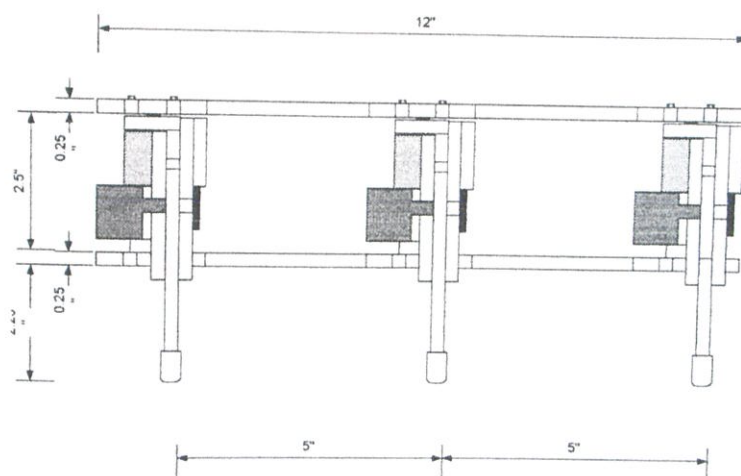
หุ่นยนต์ 6 ขา เป็นหุ่นยนต์ที่เลียนแบบการเคลื่อนที่ตามธรรมชาติชนิดหนึ่ง ซึ่งมีความสามารถในการไต่ไปในที่แคบและมีพื้นผิวขรุขระได้ดี การออกแบบขาของหุ่นยนต์ 6 ขามีด้วยกันหลายแบบ สำหรับหุ่นยนต์ 6 ขาที่นำมาใช้ในการทดสอบครั้งนี้มีโครงสร้างขาแบบ เพนโทกราฟ (Pantograph) ซึ่งเป็นโครงสร้างขาที่นิยมใช้กันอยู่ในปัจจุบันแบบหนึ่ง [19] เนื่องจากควบคุมได้ง่าย โครงสร้างของขาแบบดังกล่าวจะเป็นการอาศัยการเปลี่ยนแปลงมุมภายในของสี่เหลี่ยมด้านขนาน จึงทำให้ระยะเส้นทแยงมุมของมันเปลี่ยนแปลงทำให้ขาสามารถเคลื่อนที่ขึ้นลงได้ หุ่นยนต์จะอาศัยมอเตอร์จำนวน 2 ตัวในการเคลื่อนที่ของ 1 ขา ดังนั้น จึงต้องอาศัยมอเตอร์ทั้งหมด 12 ตัวเพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ทั้งหมด

#### 2.1 โครงสร้างหุ่นยนต์ 6 ขา

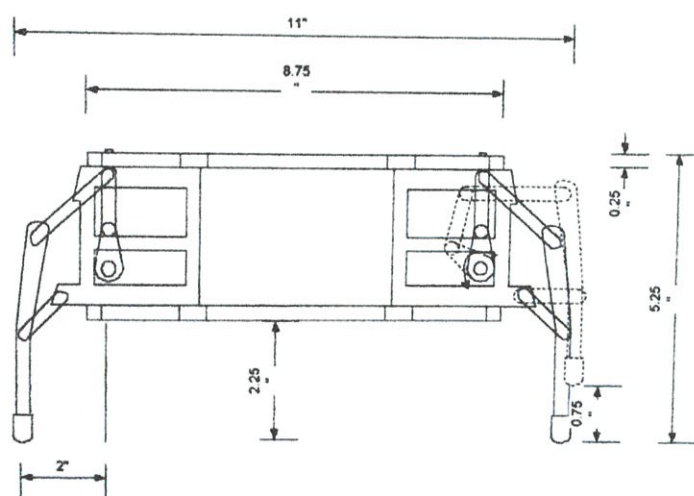
หุ่นยนต์ 6 ขาที่ใช้ เป็นหุ่นยนต์ที่สร้างขึ้นมาแล้ว จากโครงการวิจัยเดิม [21] เป็นหุ่นยนต์ที่สร้างขึ้นจากแผ่น PVC และอลูมิเนียม โดยมีโครงสร้างดังรูปที่ 2.1รูปที่ 2.2รูปที่ 2.3 ซึ่งขนาดของหุ่นยนต์ที่กำหนดในรูปมีหน่วยเป็นนิ้ว



รูปที่ 2.1 โครงสร้างหุ่นยนต์ 6 ขา มองจากด้านบน



รูปที่ 2.2 โครงสร้างหุ่นยนต์ 6 ขา มองจากด้านข้าง



รูปที่ 2.3 โครงสร้างหุ่นยนต์ 6 ขา มองจากด้านหน้า

จากรูป จะเห็นถึงโครงสร้างของขาแบบเพนโทกราฟ ซึ่งมีโครงสร้างเป็นคานสี่ท่อนขนานกัน เป็นรูปสี่เหลี่ยมด้านขนาน เมื่อขยับขาขึ้นก็คือจุดที่เส้นทแยงมุมทั้ง 2 เส้นมีขนาดเท่ากัน ส่วนการกดลงสุดคือจุดที่เส้นทแยงมุมของสี่เหลี่ยมต่างกันมากที่สุด ขาหุ่นยนต์ประเภทนี้จะยกขาได้ไม่มากนัก ดังเช่นหุ่นยนต์ที่ใช้ในการทดสอบนี้ จะสามารถยกขาได้เพียง 0.75 นิ้วเท่านั้น แต่เนื่องจากเป็นระบบขาที่ควบคุมและออกแบบได้ง่าย จึงเลือกมาใช้งาน

การหมุนขาไปด้านหน้าและหลังนั้น กรณีขาที่อยู่ที่มีมุมของหุ่นยนต์ จะสามารถหมุนออกมา นอกตัวหุ่นยนต์ได้ประมาณ  $90^\circ$  และหมุนเข้าไปในตัวหุ่นยนต์ได้ประมาณ  $45^\circ$  ส่วนขากลาง หุ่นยนต์ จะหมุนได้ด้านละ  $45^\circ$  เพื่อป้องกันไม่ให้ขาชนกัน

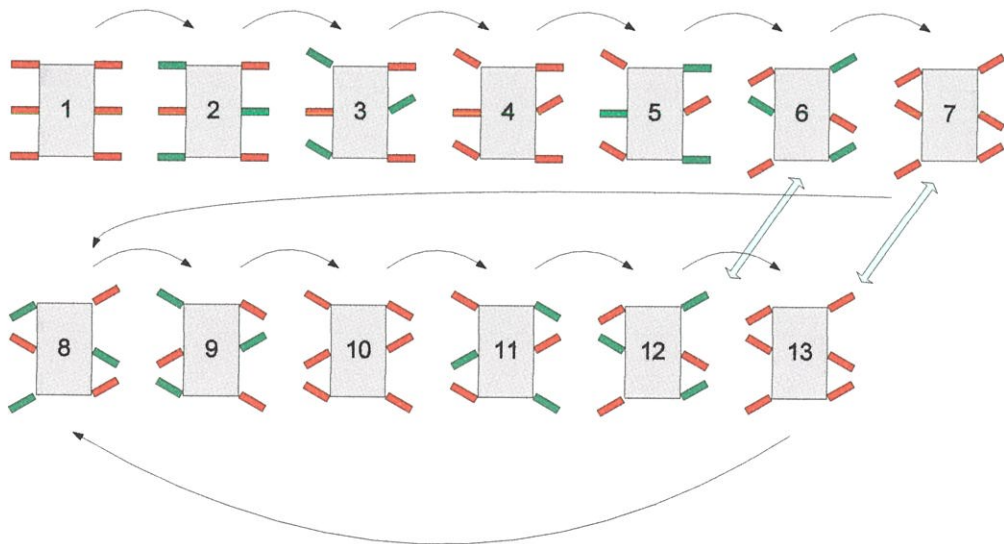
การบังคับกระทำโดยเซอร์โวมอเตอร์ 2 ตัวต่อขา โดยเป็นมอเตอร์ควบคุมการหมุนขา 1 ตัว และ มอเตอร์ควบคุมการยกขาอีก 1 ตัว รวมมอเตอร์ทั้งหมด 12 ตัว เซอร์โวมอเตอร์ที่ใช้เป็นเซอร์โวมอเตอร์ของอุปกรณ์เครื่องเล่นวิทยุบังคับของบริษัทฟูตาบา รุ่น S3001 และรุ่น FP-S148 มีแรงบิดเอาท์พุทที่ 3 กิโลกรัม ความเร็วในการหมุน  $0.22$  วินาที /  $60^\circ$  วิธีการควบคุมเซอร์โวมอเตอร์นั้นจะ ออกส่าวโดยละเอียดในภาคผนวก ง.

## 2.2 การเดินแบบขั้นตอนวิธีการเปิดตารางของหุ่นยนต์ 6 ขา

เพื่อเป็นการทดสอบประสิทธิภาพในการควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ จึงจำเป็นต้องสร้างหุ่นยนต์ที่ใช้วิธีควบคุมด้วยวิธีอื่นขึ้นมา วิธีที่เลือกนี้ได้แก่ การเดินแบบขั้นตอนวิธีการเปิดตาราง โดยการเดินแบบขั้นตอนวิธีการเปิดตาราง เป็นวิธีการควบคุมหุ่นยนต์แบบสถิต (Static) ที่ใช้งานกันอยู่โดยทั่วไป เนื่องจากสามารถสร้างและวิเคราะห์การทำงานได้โดยง่าย ซึ่งมีขั้นตอนการสร้างดังนี้

### 2.2.1 การหารูปแบบการเคลื่อนที่

ได้แก่การสรุปลำดับขั้นตอนต่าง ๆ ในการเคลื่อนที่ของหุ่นยนต์ สำหรับการเคลื่อนที่ของหุ่นยนต์ 6 ขา สามารถสรุปได้ดังรูปที่ 2.4

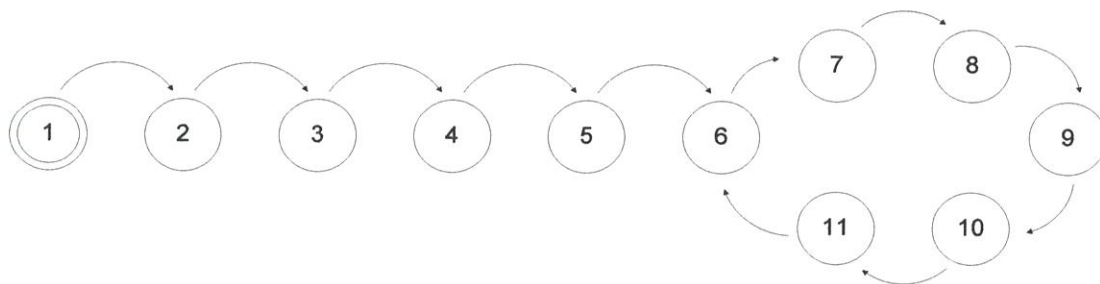


รูปที่ 2.4 ขั้นตอนต่าง ๆ ในการควบคุมการเดินแบบขั้นตอนวิธีการเปิดตาราง

จากรูปที่ 2.4 กำหนดให้การเดินเริ่มต้นตั้งแต่หุ่นยนต์หยุดนิ่ง อยู่ในสภาวะพร้อม ขาที่แสดงเป็นสีแดงหมายถึงขาที่วางอยู่กับพื้น ส่วนขาที่เป็นสีเขียวคือขาที่ถูกสั่งให้ยกขึ้น จะเห็นว่าการเคลื่อนที่ จะสั่งงานเป็นแนวทแยง คือ 2 ขาหน้าหลัง และขากลางของด้านตรงข้ามจะเคลื่อนที่พร้อมกัน

การเคลื่อนที่จะไล่จากจังหวะที่ 1 ไปตามลำดับจนกระทั่งถึงจังหวะที่ 12 จะเห็นว่ามีลักษณะการวางขาเหมือนกับจังหวะที่ 6 และเมื่อเคลื่อนที่ต่อไป จังหวะที่ 13 ก็จะไปซ้ำกับจังหวะที่ 7 เป็นเช่นนี้ไปเรื่อย ๆ

ดังนั้นจึงอาจสรุปได้ว่า จังหวะการก้าวเดินของหุ่นยนต์ 6 ขา จะมีจังหวะเริ่มต้นอยู่รวมทั้งสิ้น 5 จังหวะ จากนั้นจังหวะการเคลื่อนที่ก็จะวนเป็นวงรอบทั้งหมด 6 จังหวะ เป็นเช่นนี้ไปตลอด เมื่อสามารถวิเคราะห์จังหวะการก้าวเดินของหุ่นยนต์ได้แล้ว ลำดับต่อไป จะนำจังหวะการก้าวทั้งหมดมาเขียนให้อยู่ในรูปของแผนภูมิสถานะดังรูปที่ 2.5



รูปที่ 2.5 แผนภูมิสถานะของการควบคุมการเดินแบบขั้นตอนวิธีการเปิดตาราง

## 2.2.2 รูปแบบการเคลื่อนที่ในรูปตารางสถานะ

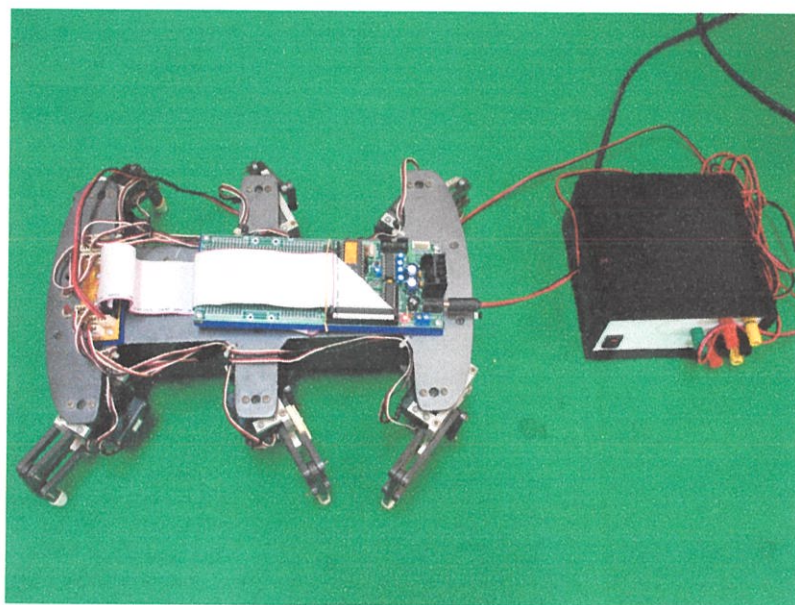
ตารางที่ 2. 1 สถานะสำหรับการควบคุมการเดินแบบขั้นตอนวิธีการเปิดตาราง

State No.	Left Front		Left Center		Left Rear		Right Front		Right Center		Right Rear	
	Rotate	move	Rotate	move	Rotate	move	Rotate	move	Rotate	move	Rotate	move
1	0	-1	0	-1	0	-1	0	-1	0	-1	0	-1
2	0	1	0	-1	0	1	0	-1	0	1	0	-1
3	1	1	0	-1	1	1	0	-1	1	1	0	-1
4	1	-1	0	-1	1	-1	0	-1	1	-1	0	-1
5	1	-1	0	1	1	-1	0	1	1	-1	0	1
6	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1
7	-1	-1	1	-1	-1	-1	1	-1	-1	-1	1	-1
8	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1
9	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
10	1	-1	-1	-1	1	-1	-1	-1	1	-1	-1	-1
11	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1

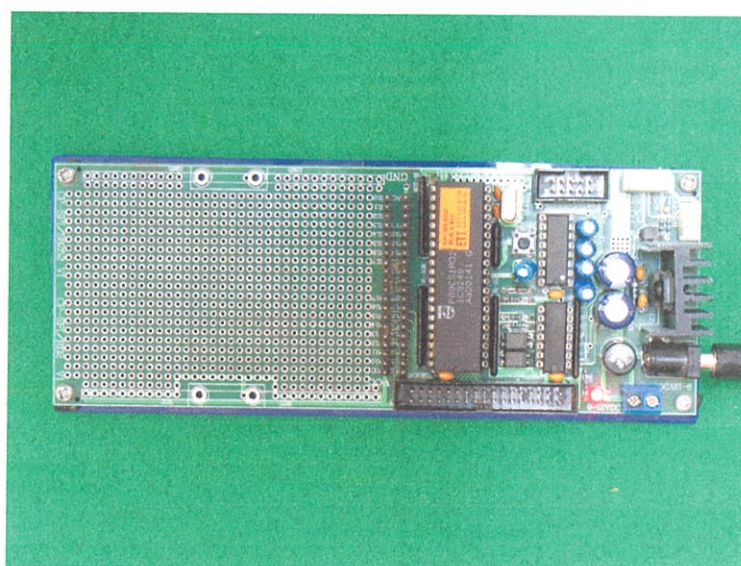
ได้ดังตารางที่ 2.1 โดยกำหนดให้กรณีการยกขา “1” หมายถึงยกขึ้น “-1” หมายถึงวางลง ส่วนกรณีหมุนขา “1” คือหมุนไปข้างหน้า “0” คืออยู่ตรงกลาง และ “-1” คือหมุนไปข้างหลัง

### 2.2.3 การสร้างชุดควบคุมจากตารางสถานะ

เมื่อได้ตารางสถานะแล้ว นำไปเขียนเป็นโปรแกรมควบคุมหุ่นยนต์ในชุดคอนโทรลเลอร์ โดยใช้หลักการอ่านข้อมูลจากตารางที่กำหนดไว้ตามสถานะปัจจุบัน จากนั้นห้วงเวลาไปจนกว่า จะถึงเวลาเปลี่ยนสถานะ จึงไปอ่านข้อมูลที่สถานะใหม่ ส่งไปควบคุมหุ่นยนต์ ทำเช่นนี้ไปตลอด



รูปที่ 2.6 ชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาแบบขั้นตอนวิธีการเปิดตาราง



รูปที่ 2.7 ชุดวงจรควบคุมสำเร็จรูปที่นำมาใช้ในการควบคุมเซอร์โวมอเตอร์

รูปที่ 2.6 เป็นหุ่นยนต์ที่ใช้การควบคุมแบบขั้นตอนวิธีการเปิดตาราง ในการทดสอบเปรียบเทียบการทำงานกับการควบคุมจากโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ส่วนรูปที่ 2.7 เป็นชุดควบคุมซึ่งเป็นวงจรควบคุมของบริษัทที่จำกัด ใช้ไมโครคอนโทรลเลอร์ เบอร์ P89C51RD2 โดยโปรแกรมที่ได้พัฒนาขึ้น สามารถควบคุมเซอร์โวมอเตอร์ได้ 16 ตัว ซึ่งในที่นี้ ใช้เซอร์โวมอเตอร์เพียง 12 ตัว สำหรับรายละเอียดการออกแบบโปรแกรมควบคุมดังกล่าว จะอธิบายในภาคผนวก ก.

### 2.3 ความสามารถและข้อจำกัด

จากการทดสอบ หุ่นยนต์สามารถเคลื่อนที่ได้ตามต้องการ ซึ่งเป็นการทำงานพื้นฐานในการทดลองครั้งนี้ อย่างไรก็ตามการพัฒนาหุ่นยนต์ดังกล่าวก็ไม่สามารถกระทำได้ด้วยตัวของมันเอง เนื่องจาก จำเป็นต้องให้มีลักษณะทางกายภาพให้เหมือนกันกับหุ่นยนต์ที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ทั้งนี้เพื่อให้การทดลองเป็นไปอย่างถูกต้องสมบูรณ์ ดังนั้นจำเป็นต้องปรับแต่งค่าสูงสุดต่ำสุด ในการเคลื่อนที่ของเซอร์โวมอเตอร์ให้ตรงกันกับหุ่นยนต์ที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์นั่นเอง

### 2.4 สรุปท้ายบท

บทนี้กล่าวถึง การสร้างชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาแบบขั้นตอนวิธีการเปิดตาราง จะเห็นว่าขั้นตอนที่ไม่ซับซ้อนมาก การออกแบบเริ่มจากกำหนดลักษณะการเคลื่อนที่ของหุ่นยนต์ แล้วนำมาวิเคราะห์ในรูปแบบผังภูมิสถานะจากนั้นจึงสร้างเป็นตารางเพื่อใช้ควบคุมการเคลื่อนที่ต่อไป

หุ่นยนต์ที่สร้างขึ้นนี้ เพื่อใช้เปรียบเทียบกับชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ที่ทำการศึกษา ซึ่งวิธีออกแบบโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ที่นำมาควบคุมหุ่นยนต์จะได้กล่าวถึงในบทที่ 4 และบทที่ 5 ต่อไป

### บทที่ 3

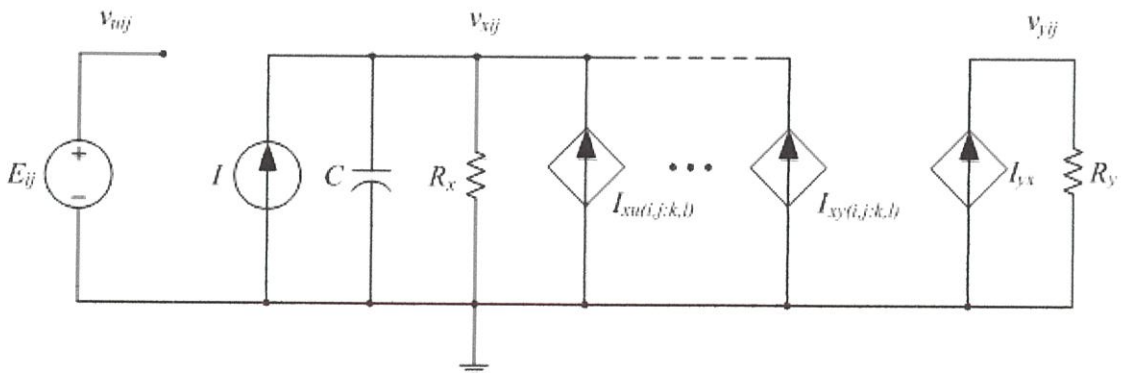
## โครงข่ายประสาทเทียมแบบเซลล์ูลาร์

โครงข่ายประสาทเทียมแบบเซลล์ูลาร์ เป็นโครงข่ายประสาทเทียมซึ่งเลียนแบบมาจากการทำงานของระบบประสาทรอบนอก (Peripheral Nervous System: PNS) [1] ซึ่งประกอบด้วย เซลล์ประสาทรับความรู้สึก (Sensory neuron) และ เซลล์ประสาทสั่งการ (Motor neuron) [2] ในทางวิทยาศาสตร์ทางระบบประสาท (Neural Science) ได้มีการศึกษาวิจัยเพื่อทำความเข้าใจ และพยายามนำไปประยุกต์กับการเชื่อมต่อระหว่างระบบประสาทของสิ่งมีชีวิตกับอุปกรณ์ภายนอก เช่น แขนและขาเทียมที่สามารถควบคุมได้จากสมองมนุษย์ หรือการเข้าไปควบคุมการเคลื่อนที่ของสัตว์ จากอุปกรณ์ภายนอกแทนสมองเป็นต้น

ในวงการวิศวกรรมไฟฟ้า ได้มีการนำความรู้ที่ได้นี้มาประยุกต์เป็นโครงข่ายประสาทเทียมขึ้น ผู้เสนอแนวความคิดนี้เป็นกลุ่มแรกคือ Prof. Leon O. Chua และ Lin Yang [3] [4] โดยใช้ชื่อเรียกว่า โครงข่ายประสาทเทียมแบบเซลล์ูลาร์ (Cellular Neural Network :CNN) ลักษณะพิเศษของโครงข่ายประสาทเทียมแบบเซลล์ูลาร์แตกต่างจากโครงข่ายประสาทเทียมทั่วไป ที่เซลล์ต่างๆ จับกลุ่มกันอยู่ และมีการป้อนกลับจากเอาต์พุตของตัวเองและเซลล์ข้างเคียงมายังอินพุต ทำให้มันเหมือนกลุ่มวงจรป้อนกลับ ซึ่งทำให้เกิดปรากฏการณ์ต่าง ๆ ในโครงข่ายดังจะได้กล่าวต่อไป

### 3.1 ทฤษฎีโครงข่ายประสาทเทียมแบบเซลล์ูลาร์

โครงข่ายประสาทเทียมแบบเซลล์ูลาร์นี้เมื่อนำมาเขียนเป็นวงจรสมมูลแล้วจะมีโครงสร้างหน่วยย่อยเรียกว่าเซลล์ดังรูปที่ 3.1



รูปที่ 3.1 วงจรสมมูลของเซลล์

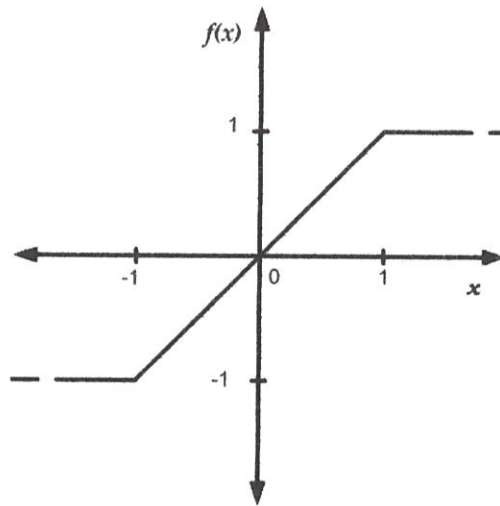
โดย  $V_{uij}$  คือค่าแรงดันที่อินพุตของเซลล์ที่ตำแหน่ง  $ij$   
 $V_{xij}$  คือค่าแรงดันภายในเซลล์ที่ตำแหน่ง  $ij$   
 $V_{yij}$  คือค่าแรงดันที่เอาต์พุตของเซลล์ที่ตำแหน่ง  $ij$

และ

$$I_{xy}(i, j : k, l) = A(i, j : k, l)v_{ykl} \quad (3.1)$$

$$I_{xu}(i, j : k, l) = B(i, j : k, l)v_{ukl} \quad (3.2)$$

และกำหนดให้ฟังก์ชันค่าเอาต์พุตของเซลล์เป็นไปตามวงจรรวมของ Chua มีคุณลักษณะดังรูปที่ 3.2



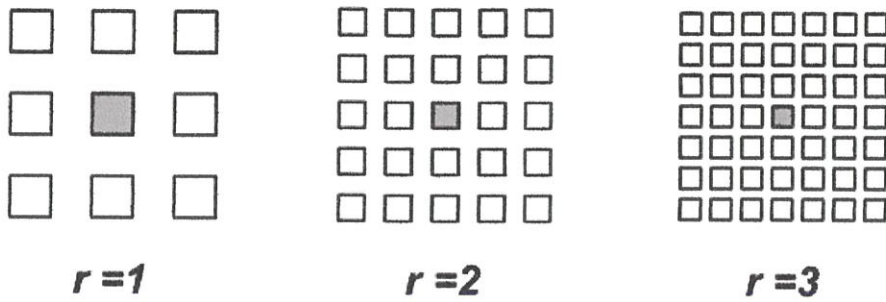
รูปที่ 3.2 คุณสมบัติสมการเอาต์พุตของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์

จะเห็นว่าเอาต์พุตของเซลล์มีลักษณะแบบไม่เป็นเชิงเส้น ทำให้ในบางครั้งโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์นี้จึงถูกเรียกว่า โครงข่ายไม่เป็นเชิงเส้นแบบเซลล์ลูลาร์ (Cellular Nonlinear Network) และเมื่อนำคุณสมบัติข้างต้นมาเขียนเป็นสมการจะได้ดังสมการที่ (3.3) นี้

$$v_{ykl} = \frac{1}{2} (|v_{xij}(t) + 1| - |v_{xij}(t) - 1|) \quad (3.3)$$

เนื่องจากโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์มีการทำงานร่วมกันเป็นกลุ่มก้อน จะมีขนาดเท่าใดขึ้นอยู่กับความต้องการของงานนั้น ๆ และได้นิยามการกำหนดเซลล์ที่เป็นสมาชิกรอบข้างของ  $N_r$  (r-neighborhood) มีดังนี้

$$N_r(i, j) = \{C(k, l) | \max\{|k - i|, |l - j|\} \leq r, 1 \leq k \leq M; 1 \leq l \leq N\} \quad (3.4)$$



รูปที่ 3.3 ขนาดของ  $N_r$  เมื่อ  $r=1, 2$  และ  $3$  ตามลำดับ

จากวงจรใน รูปที่ 3.1 และสมการที่ (3.1) ถึง (3.4) เมื่อนำไปเขียนให้อยู่ในรูปสมการสถานะ จะสามารถเขียนสมการได้ดังนี้

$$C \frac{dv_{xij}(t)}{dt} = -\frac{1}{R} v_{xij}(t) + \sum_{c(k,l) \in N_r(i,j)} A(i, j : k, l) v_{ykl}(t) + \sum_{c(k,l) \in N_r(i,j)} B(i, j : k, l) v_{uk}(t) + I \quad (3.5)$$

ถ้าที่  $M$  คือจำนวนเซลล์ทางแนวตั้ง  $N$  คือจำนวนเซลล์ทางแนวนอนของเซลล์นั้น ๆ โดยปกติแล้วค่า  $R_x$   $R_y$  และ  $C$  มักมีค่าเป็น “1” เพื่อให้ง่ายต่อการออกแบบ ซึ่งจะทำได้สมการสถานะของเซลล์คือ

$$\dot{x}_{ij} = x_{ij}(t) + A * y_{ij}(t) + B * u_{ij} + I \quad (3.6)$$

เมื่อ

$$1 \leq i \leq M, 1 \leq j \leq N$$

และ

$$T * v_{ij} = \sum_{c(k,l) \in N_r(i,j)} T(k-i, l-j) v_{kl} \quad (3.7)$$

จากการศึกษาพบว่า คุณสมบัติของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์จะเปลี่ยนแปลงตามค่าเมตริกซ์  $A$  และ  $B$  ซึ่งเรียกว่า Template เป็นกำหนดค่าน้ำหนักของอินพุตแต่ละตัวนั่นเอง ดังนั้นการวิจัยในช่วงแรก ๆ ของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์จึงเป็นการหาค่า Template ที่เหมาะสมในแต่ละงาน

เนื่องจากโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ มีหลักการทำงานเช่นเดียวกับระบบประสาทรอบนอก ดังนั้นจึงมีความสามารถในงานประเภทเดียวกันเช่น เป็นตัวประมวลผลก่อน(Preprocess) ในงานตรวจจับทั้งทางภาพ และ เสียง อีกทั้งยังมีการประยุกต์คุณสมบัติตามแบบเซลล์ประสาทสั่งการ ซึ่งนำมาใช้ในงานวิจัยนี้

### 3.2 ปรากฏการณ์ออโตเวฟ

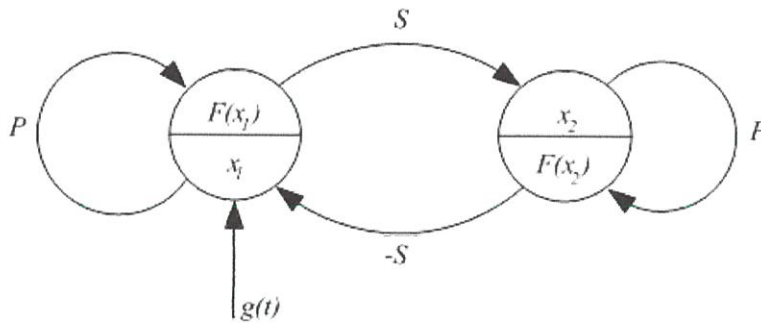
การประยุกต์ใช้งานโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ในการควบคุมการเคลื่อนที่นั้น อาศัยการเลียนแบบการทำงานของเซลล์ประสาทสั่งการของสิ่งมีชีวิต ซึ่งทำหน้าที่คล้ายตัวกำเนิดคลื่นและส่งต่อกันเป็นช่วง ๆ ไปเรียกว่า ปรากฏการณ์ออโตเวฟ (Autowaves Phenomena) [6] คลื่นที่ได้จะมีรูปแบบของสัญญาณอลวน (Chaotic Signal) จากงานวิจัยของ Prof. Josef A. Nessesk และ Prof. Fau Zou[7][8][9] แสดงให้เห็นว่าสามารถเกิดสภาวะอลวน (Chaotic State) ได้จากโครงข่ายแบบเซลล์ลูลาร์ที่มีเซลล์เพียง 2 เซลล์โดยอาศัยสมการสถานะดังนี้ [5]

$$\begin{aligned} \dot{x}_1 + x_1 &= pf(x_1) - sf(x_2) + g(t) \\ \dot{x}_2 + x_2 &= sf(x_1) + pf(x_2) \end{aligned} \quad (3.8)$$

เมื่อ  $p > 1, s > 0$  และสมการเอาท์พุทคือ

$$f(x(t)) = \frac{1}{2}(|x_i(t) + 1| - |x_i(t) - 1|), \quad i = 1, 2 \quad (3.9)$$

ซึ่งจะทำให้เกิด แผนภูมิได้ ดังรูปต่อไปนี้



รูปที่ 3.4 แผนภูมิของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ขนาด 2 เซลล์ที่เกิดสภาวะอลวน

จากสมการ  $g(t)$  คือค่าสัญญาณที่ต้องการนำไปแทรกแซง ซึ่งกรณีนี้มีค่าเป็น "0" และจากวงจรสมมูลในรูปที่ 3.1 จะมีค่ากระแสคงที่ภายใน  $I$  ของแต่ละเซลล์ ดังนั้นจึงสามารถเขียนสมการ (3.8) ได้ใหม่คือ

$$\begin{aligned} \dot{x}_1 &= -x_1 + (1 + \mu)y_1 - sy_2 + I_1 \\ \dot{x}_2 &= -x_2 + sy_1 + (1 + \mu)y_2 + I_2 \end{aligned} \quad (3.10)$$

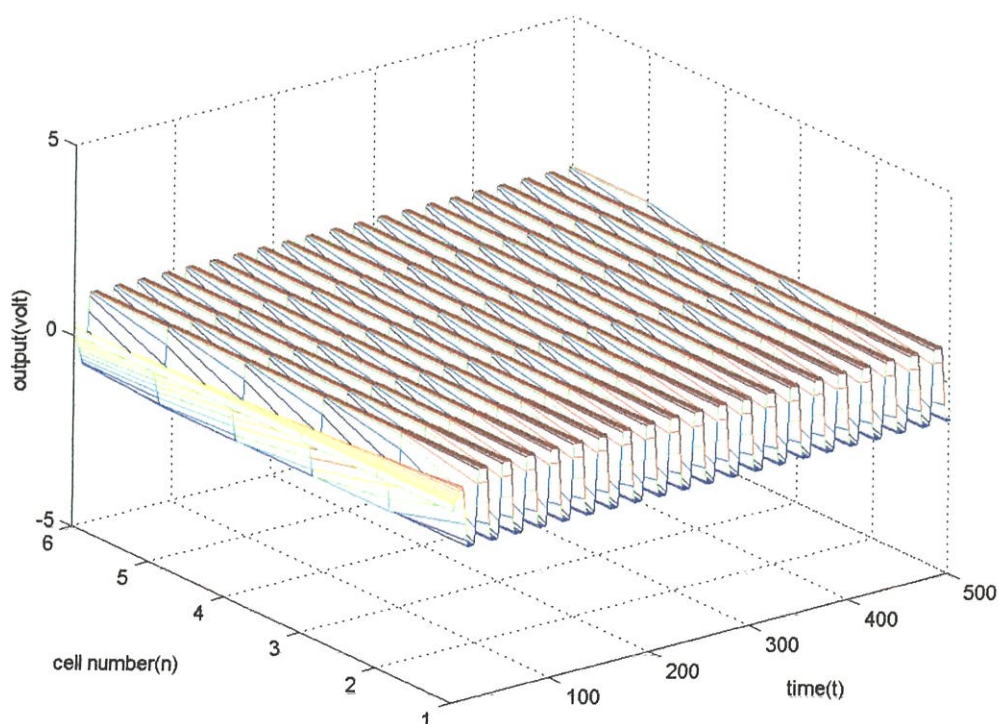
กำหนดให้  $1 + \mu = p$  โดยสัญญาณจะเกิดที่  $\mu, s > 0$  และ  $I_1, I_2$  เป็นค่าคงที่ในการเปลี่ยนแปลงเวลาของสัญญาณ กรณีเมื่อมีเซลล์ต่อกันไปเป็นจำนวน  $n$  เซลล์ สมการดังกล่าวจะเปลี่ยนแปลงไป โดยกำหนดให้  $1 + \mu = \mu_1$  และ  $s_n = \mu_{n+1}$  ซึ่งสมการของโครงข่ายประสาทเทียมแบบเซลล์ลูร์ที่มีขนาด  $n$  เซลล์ จะได้สมการที่ (3.11) และ (3.12)

$$\begin{aligned} \dot{x}_1 &= -x_1 + \mu_1 y_1 + \mu_2 y_2 + I_1 \\ \dot{x}_2 &= -x_2 + \mu_2 y_2 + \mu_3 y_3 + I_2 \\ \dot{x}_3 &= -x_3 + \mu_3 y_3 + \mu_4 y_4 + I_3 \\ &\vdots \\ \dot{x}_{n-1} &= -x_{n-1} + \mu_{n-1} y_{n-1} + \mu_n y_n + I_{n-1} \\ \dot{x}_n &= -x_n + \mu_n y_n - \mu_1 y_1 + I_n \end{aligned} \quad (3.11)$$

และ

$$y_i = \frac{1}{2}(|x_i + 1| - |x_i - 1|) \quad (3.12)$$

ถ้ามีเซลล์ทั้งหมด 6 เซลล์ กำหนดค่า  $I_1$  ถึง  $I_6$  มีค่าเป็น "0" โดยกำหนดให้ ค่าทางแกน  $X$  คือเวลา  $t$  ตั้งแต่ Sampling  $N = 1$  ถึง 500 ค่าทางแกน  $Y$  คือเซลล์ของ CNN ตั้งแต่ 1-6 และแกน  $Z$  คือค่าเอาต์พุตที่ได้ในแต่ละเซลล์ มีหน่วยเป็น volt จะได้สัญญาณดังรูปที่ 3.5



รูปที่ 3.5 เอาต์พุต ( $y_n$ ) ที่ได้จากสมการ (11), (12) เมื่อค่า  $I_j = 0$  โดย  $j = 1, 2, 3, \dots, 6$

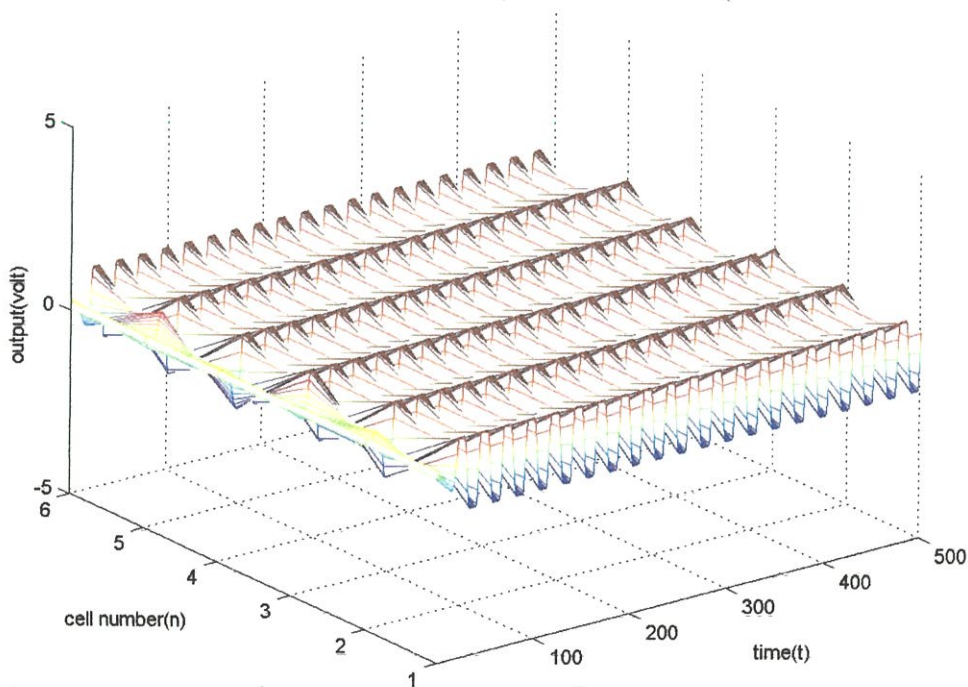
จะเห็นว่า ค่าสัญญาณที่ได้จะมีลักษณะเสมือนคลื่นเคลื่อนที่ออกไป โดยไล่จาก  $y_1$  ถึง  $y_6$

### 3.3 การควบคุมการเคลื่อนที่โดยอาศัยออตเวฟ

จากสมการ (3.11), (3.12) ถ้าระบบมีเซลล์จำนวน 6 เซลล์จะได้ผลตามรูปที่ 3.5 สังเกตว่า สัญญาณที่ได้มีเฟสกระจายอยู่ที่ครึ่งคาบแรกเท่านั้น ถ้าต้องการให้ได้สัญญาณมีเฟสกระจายเต็มคาบจะต้องใช้เซลล์ทั้งสิ้น 12 เซลล์ด้วยกัน และเพื่อประหยัดจำนวนเซลล์เราจึงเปลี่ยนค่าพารามิเตอร์ของสมการสถานะใหม่เพื่อให้คาบของสัญญาณสั้นลง จะได้สมการดังนี้

$$\begin{aligned} \dot{x}_1 &= -x_1 + \mu_1 y_1 - \mu_2 y_2 + I_1 \\ \dot{x}_2 &= -x_2 + \mu_2 y_2 - \mu_3 y_3 + I_2 \\ \dot{x}_3 &= -x_3 + \mu_3 y_3 - \mu_4 y_4 + I_3 \\ &\vdots \\ \dot{x}_{n-1} &= -x_{n-1} + \mu_{n-1} y_{n-1} - \mu_n y_n + I_{n-1} \\ \dot{x}_n &= -x_n + \mu_n y_n + \mu_1 y_1 + I_n \end{aligned} \quad (3.13)$$

จะสังเกตได้ว่า การบ้อนกลับจากเซลล์ถัดไปในแต่ละแถวเป็นการบ้อนกลับแบบลบ เมื่อใช้สมการที่ (3.12), (3.13) กำหนดให้  $I_1$  ถึง  $I_6$  มีค่าเป็น "0" และเมื่อนำมาแสดงในรูปแบบกราฟโดยกำหนดให้ ค่าทางแกน X คือเวลา  $t$  ตั้งแต่ Sampling  $N = 1$  ถึง 500 ค่าทางแกน Y คือ เซลล์ของ CNN ตั้งแต่ 1-6 และแกน Z คือค่าเอาต์พุตที่ได้ในแต่ละเซลล์ มีหน่วยเป็น volt จะได้ผลดังรูปที่ 3.6 จะเห็นว่าจำนวนเซลล์ที่ทำให้เอาต์พุตมีเฟสกระจายทั้งคาบจะลดลงมาเหลือ 6 เซลล์ดังนั้น จึงสามารถใช้จำนวนเซลล์เพียง 6 เซลล์ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ได้



รูปที่ 3.6 เอาต์พุต ( $y_n$ ) ที่ได้จากสมการ (13), (14) เมื่อค่า  $I_j = 0$  โดย  $j=1, 2, 3, \dots, 6$

การควบคุมดังที่กล่าวมานี้ เป็นการควบคุมการเคลื่อนที่ ไปข้างหน้าหรือไปข้างหลังของหุ่นยนต์ ซึ่งจำเป็นต้องควบคุมการยกขาขึ้นลงของหุ่นยนต์ด้วย ทั้งนี้จำเป็นต้องเข้าจังหวะกับการเคลื่อนที่หน้าหลัง ดังนั้นจึงต้องมีเซลล์อีกจำนวน 6 เซลล์เพื่อควบคุมการยกขาขึ้นลงด้วย ซึ่งเป็นเซลล์ที่มีสมาชิกรอบข้างเป็น 0 ( $N_0$ ) ดังนั้น สมการของเซลล์ยกขาคือ

$$\dot{x}_{m_i} = -x_{m_i} + \beta y_{r_i} + I_{m_i} \tag{3.14}$$

และ

$$y_{m_i} = \frac{1}{2} (|x_{m_i} + 1| - |x_{m_i} - 1|) \tag{3.15}$$

เมื่อ

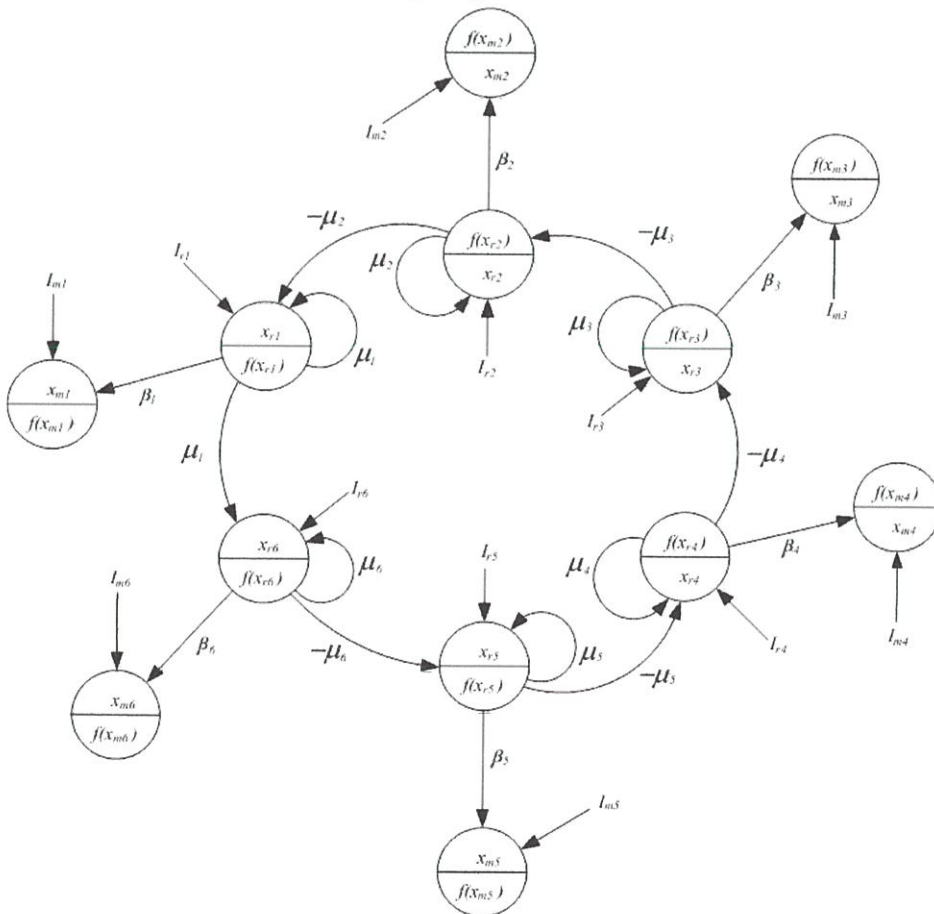
$\beta$  คือ ค่า Template ในกรณีของ  $N_0$  จะมีค่าเป็นค่าคงที่

$y_{m_i}$  คือ ค่าเอาต์พุตของเซลล์ควบคุมการยกขาที่  $i$

$y_{r_i}$  คือ ค่าเอาต์พุตของเซลล์ควบคุมการเคลื่อนที่หน้า/หลังที่  $i$

$I_{m_i}$  คือ ค่ากระแสคงที่ภายในเซลล์ควบคุมการยกขาที่  $i$

ในการใช้งาน จำเป็นต้องหาค่า  $\beta$  ที่เหมาะสม ซึ่งค่าดังกล่าวจะมีผลกับความเร็วในการยกขาของหุ่นยนต์ จากการออกแบบจะได้แผนภูมิดังรูปที่ 3.7

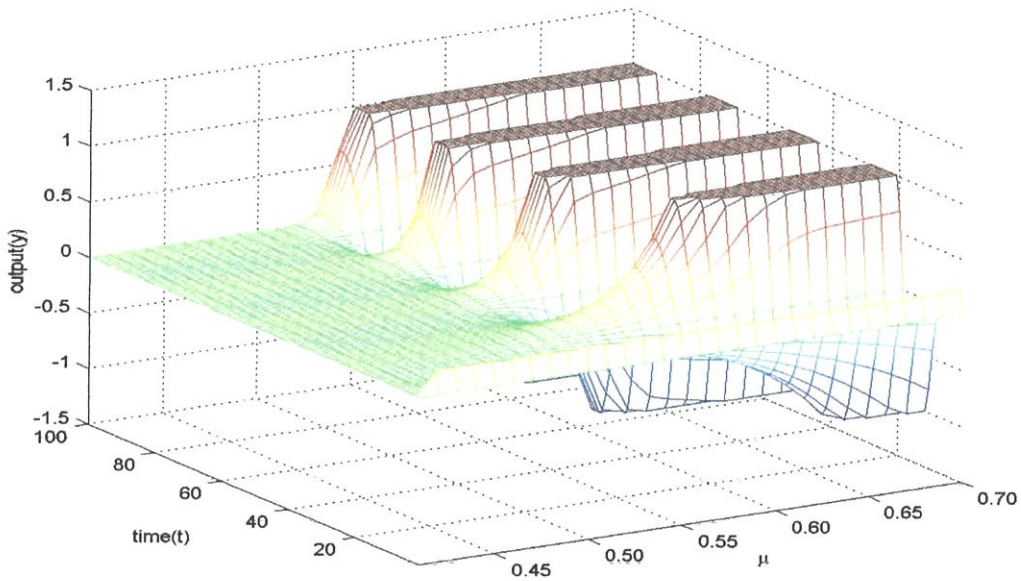


รูปที่ 3.7 แผนภูมิของโครงข่าย CNN สำหรับควบคุมหุ่นยนต์หกขา

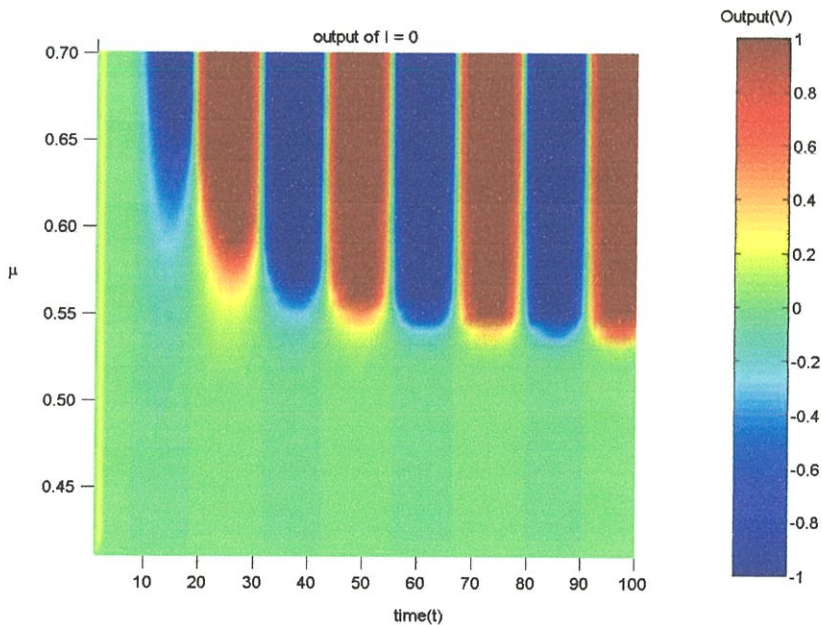
การอ่านแผนภูมิสถานะในรูปที่ 3.7 นี้ เหมือนกับกรณีของ Finite State Machine ปกติ

### 3.4 ผลทดลองการปรับค่าพารามิเตอร์ต่าง ๆ ของระบบ

จากการจำลองโดยอาศัยโปรแกรม MATLAB กำหนดค่าพารามิเตอร์  $I_r$  ทุกค่าให้เป็น 0 แล้วทดสอบเปลี่ยนค่าพารามิเตอร์ ให้  $\mu$  เป็นแกน X กำหนดค่าตั้งแต่ 0.4 ถึง 0.7 ซึ่งเป็นช่วงที่มีการเปลี่ยนแปลงของเอาต์พุตเป็นที่น่าสนใจ คำนวณที่เวลา  $t$  ตั้งแต่ Sampling  $N = 1$  ถึง 100 เป็นแกน Y ผลที่ได้จากเซลล์ 1 เป็นแกน Z มีหน่วยเป็น volt จะมีลักษณะดังนี้



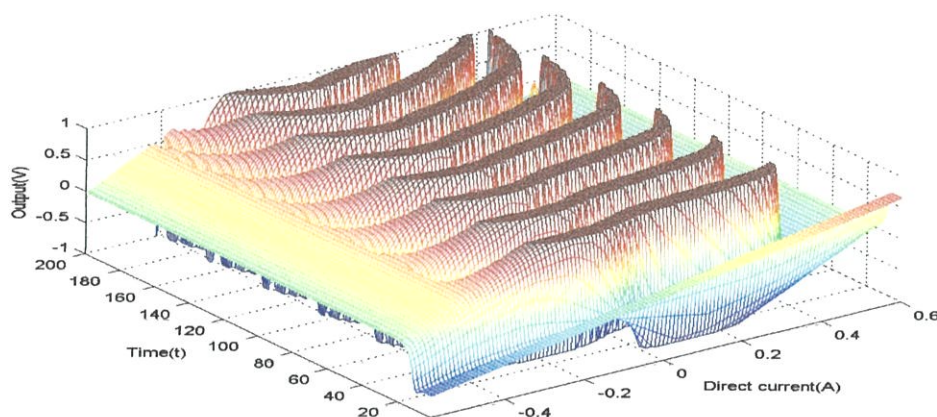
รูปที่ 3.8 เอาต์พุตของเซลล์ที่ 1 เมื่อกำหนดค่า  $I_r$  เท่ากับ 0 และเปลี่ยนค่า  $\mu$  จาก 0.4 ถึง 0.7



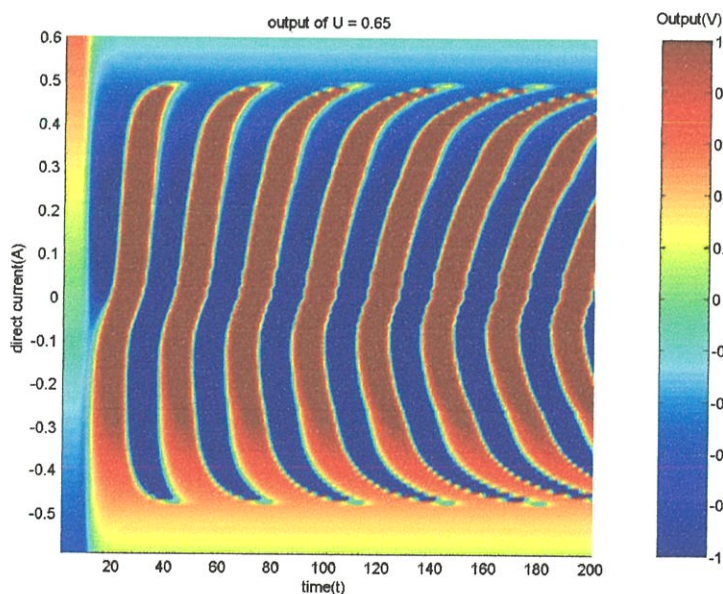
รูปที่ 3.9 เอาต์พุตของเซลล์ที่ 1 ให้แกน X เป็นเวลา เทียบกับแกน Y คือ  $\mu$

จะเห็นว่า ค่าของเอชท์พุดที่  $\mu < 0.55$  จะไม่มีคลื่นเกิดขึ้น ที่เอชท์พุดที่ค่า  $\mu \geq 0.55$  คุณสมบัติของคลื่นที่เอชท์พุดจะเริ่มแสดงออกมา และจะเข้าถึงค่าอิมพัลส์เร็วขึ้นเมื่อค่า  $\mu$  เพิ่มขึ้นเรื่อย ๆ และความชันของสัญญาณบริเวณจุดเปลี่ยนค่าระหว่างค่าอิมพัลส์ด้านบวก กับค่าอิมพัลส์ด้านลบจะมีความชันมากขึ้น จากการทดลองค่าความชันนี้จะเพิ่มขึ้นจนค่า  $\mu = 1$  จากนั้นจะมีค่าคงที่ ซึ่งความสัมพันธ์ของค่า  $\mu$  กับเอชท์พุดแสดงดังรูปที่ 3.9

นอกจากจะทดลองปรับค่าพารามิเตอร์  $\mu$  แล้ว ยังได้ทดลองปรับค่าพารามิเตอร์  $I_r$  โดยการกำหนดค่า  $\mu = 0.65$  จากนั้นทดลองปรับค่า  $I_r$  ที่แกน X ตั้งแต่ค่า -0.8 ถึง 0.8 Amp ซึ่งเป็นช่วงที่มีการเปลี่ยนแปลงของเอชท์พุดเป็นที่น่าสนใจ คำนวณที่เวลา  $t$  ตั้งแต่ Sampling  $N = 1$  ถึง 200 เป็นแกน Y ผลที่ได้จากเซลล์ 1 เป็นแกน Z มีหน่วยเป็น volt จะมีลักษณะดังนี้



รูปที่ 3.10 เอชท์พุดของเซลล์ที่ 1 เมื่อกำหนดค่า  $\mu$  เท่ากับ 0.65 และเปลี่ยนค่า  $I_r$  จาก -0.8 ถึง 0.8



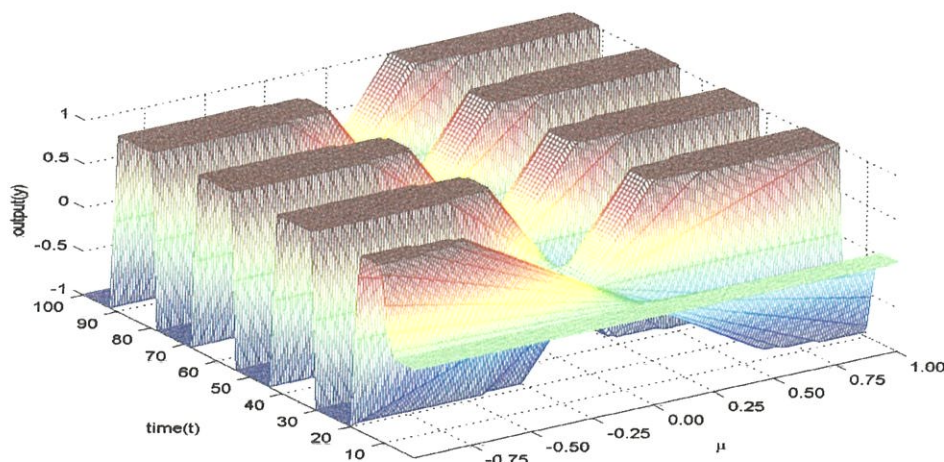
รูปที่ 3.11 เอชท์พุดของเซลล์ที่ 1 ให้แกน X เป็นเวลา เทียบกับแกน Y คือ  $I_r$ ,

จะเห็นว่า ค่าเอาต์พุตที่ค่า  $I_r$  ต่ำกว่า  $-0.5A$  และสูงกว่า  $0.5A$  สัญญาณที่ได้จะไม่มีลักษณะเป็นคลื่น โดยที่  $-0.5A$  เอาต์พุตเข้าสู่สภาวะส่ม่าเสมอโดยมีค่าประมาณ  $-0.5A$  และจะเปลี่ยนแปลงลดลงแบบเชิงเส้น เมื่อ  $I_r$  มีค่าลดลง จากการทดลองปรับค่า  $I_r$  ให้ต่ำกว่า  $-0.6A$  ปรากฏว่าค่าเอาต์พุตที่สภาวะส่ม่าเสมอจะต่ำลงแบบเชิงเส้น จนกระทั่งมีค่าเป็น  $-1A$  เมื่อ  $I_r$  มีค่าเป็น  $-1A$  และจะมีค่าเป็นเช่นนี้ไปจนถึง  $I_r = -\infty$  ในทำนองเดียวกัน เมื่อ  $I_r$  มีค่าที่  $0.5A$  เอาต์พุตเข้าสู่สภาวะส่ม่าเสมอโดยที่ค่า ประมาณ  $0.5A$  และเพิ่มขึ้นเป็นเชิงเส้น เมื่อ  $I_r$  เพิ่มขึ้น และเมื่อทดลองปรับค่า  $I_r$  สูงกว่า  $0.6A$  ปรากฏว่าค่าเอาต์พุตที่สภาวะส่ม่าเสมอจะเพิ่มขึ้นจนเป็น  $1$  เมื่อ  $I_r$  มีค่าเป็น  $1A$  และจะมีค่าเป็นเช่นนี้ไปจนถึง  $I_r = \infty$

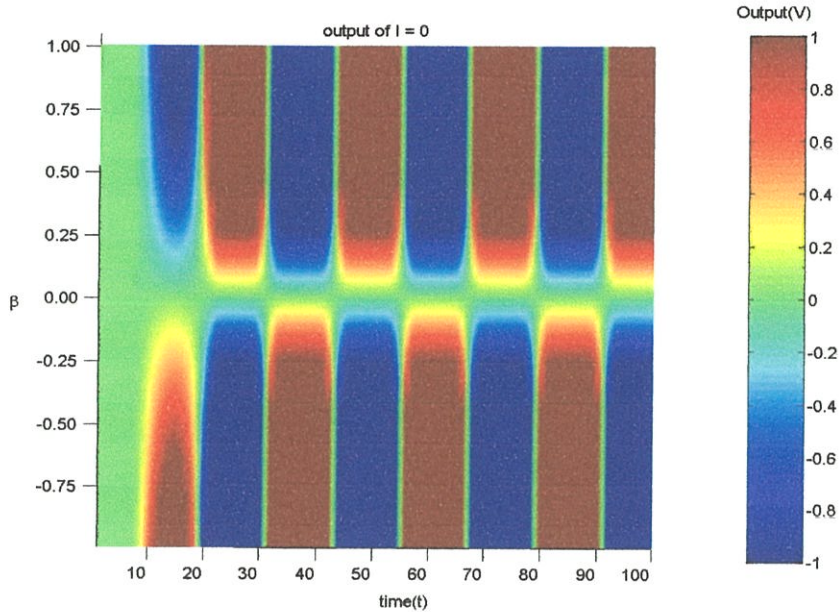
ค่าเอาต์พุตจะมีลักษณะเป็นคลื่นสมบูรณ์ที่สุดเมื่อ  $I_r$  เป็น  $0A$  โดยจะได้คลื่นที่มีด้านบวกและลบใกล้เคียงกัน มากที่สุด เมื่อค่า  $I_r$  มากกว่า  $0$  ค่าสัญญาณทางด้านลบจะมีน้อยลง และค่าต่ำสุดของสัญญาณจะไม่ถึง  $-1V$  เมื่อค่า  $I_r$  อยู่ที่  $0.4A$  ในทำนองเดียวกัน เมื่อลดค่า  $I_r$  ให้ต่ำกว่า  $0A$  ค่าสัญญาณทางด้านบวกจะมีน้อยลง และค่าสูงสุดของสัญญาณจะไม่ถึง  $1$  เมื่อค่า  $I_r$  อยู่ที่  $-0.4A$  และเมื่อค่าของ  $I_r$  มีค่าออกห่างจาก  $0A$  มากขึ้น ค่าคาบของสัญญาณจะยิ่งมีค่ามากขึ้น

จากความสัมพันธ์ของ  $\mu$  และ  $I_r$  ดังที่กล่าวมาแล้ว จึงเลือกค่า  $\mu = 0.65$  และ  $I_r = 0A$  ในการทำงานเนื่องจากเป็นช่วงที่มีความชันในการเปลี่ยนค่าจาก  $1$  เป็น  $-1$  ที่เหมาะสมในการทำงาน และมีสัญญาณทั้งด้านบวกและด้านลบใกล้เคียงกัน รวมทั้งง่ายในการนำไปคำนวณอีกด้วยเพราะพจน์  $I_r$  สามารถตัดทิ้งไปได้

จากนั้นทดลองปรับค่าพารามิเตอร์  $\beta$  ซึ่งเป็นพารามิเตอร์ของเซลล์ควบคุมการยกขา โดยกำหนดที่แกน  $X$  เป็นค่า  $\beta$  ตั้งแต่  $-5$  ถึง  $5$  ซึ่งเป็นช่วงที่มีการเปลี่ยนแปลงของเอาต์พุตเป็นที่น่าสนใจ คำนวณที่เวลา  $t$  ตั้งแต่ Sampling  $N = 1$  ถึง  $100$  เป็นแกน  $Y$  ผลที่ได้จากเซลล์  $1$  เป็นแกน  $Z$  มีหน่วยเป็น volt จะมีลักษณะดังนี้



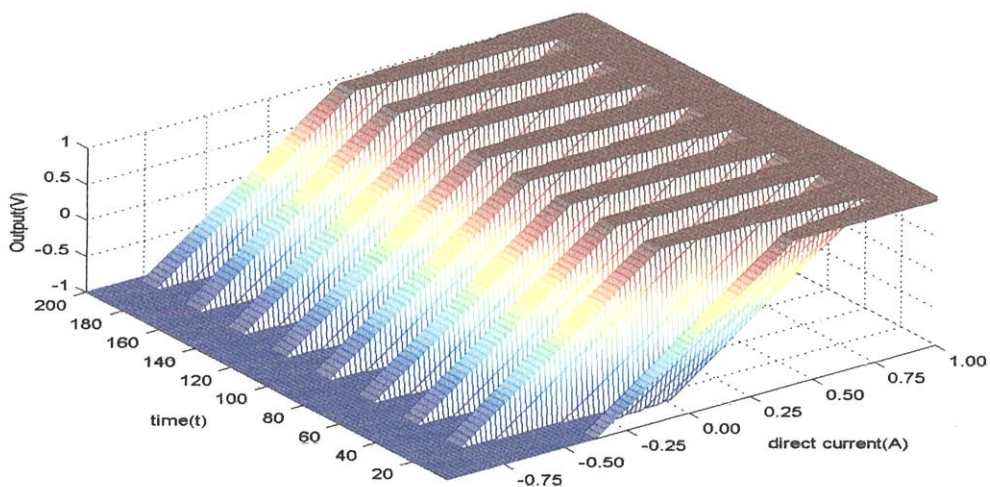
รูปที่ 3.12 เอาต์พุตของเซลล์ที่ 1 เมื่อกำหนดค่า  $I_m$  เท่ากับ 0 และเปลี่ยนค่า  $\beta$  จาก 0 ถึง 5



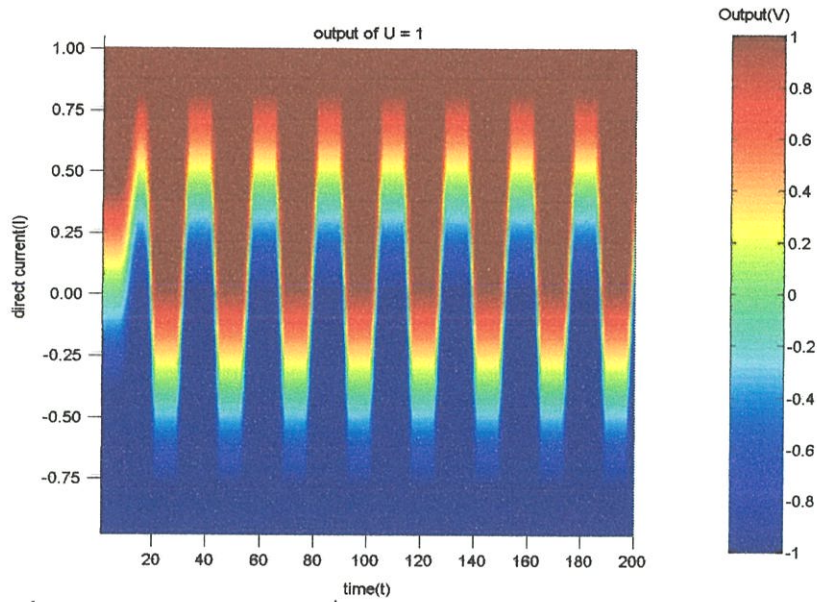
รูปที่ 3.13 เอาท์พุทของเซลล์ที่ 1 ให้แกน X เป็นเวลา เทียบกับแกน Y คือ  $\beta$

จากรูปจะเห็นว่า ค่าของเอาท์พุทจะมีค่าเป็น 0 ที่ค่า  $\beta = 0$  เมื่อเพิ่มค่าหรือลดค่า  $\beta$  ขนาดของสัญญาณจะเริ่มสูงขึ้น จนกระทั่ง ค่าสูงสุดและต่ำสุดอยู่ที่ 1 และ -1 ตามลำดับเมื่อค่า  $\beta = 1$  และ -1 แต่มีค่าเฟสของคลื่นตรงข้ามกัน จะเห็นว่าที่จุดนี้ยังปรากฏช่วงของสัญญาณได้ขึ้นและลงอยู่ ค่าความชันของการเปลี่ยนระดับระหว่าง 1 และ -1 จะมีค่าเข้าใกล้  $\infty$  และ  $-\infty$  ขึ้นเมื่อค่า  $\beta$  ห่างจาก 0 มากขึ้น จนประมาณ  $\pm \infty$  ที่ค่า  $\beta = \pm 5$

ส่วนค่าพารามิเตอร์  $I_m$  นั้น เมื่อกำหนดที่แกน X เป็นค่าระหว่าง -2.5A ถึง 2.5A และกำหนดค่า  $\beta = 4$  ได้ผลดังรูปที่ 3.14 คำนวนที่เวลา  $t$  ตั้งแต่ Sampling  $N = 0$  ถึง 200 เป็นแกน Y ผลที่ได้จากเซลล์ 1 เป็นแกน Z มีหน่วยเป็น volt จะมีลักษณะดังนี้



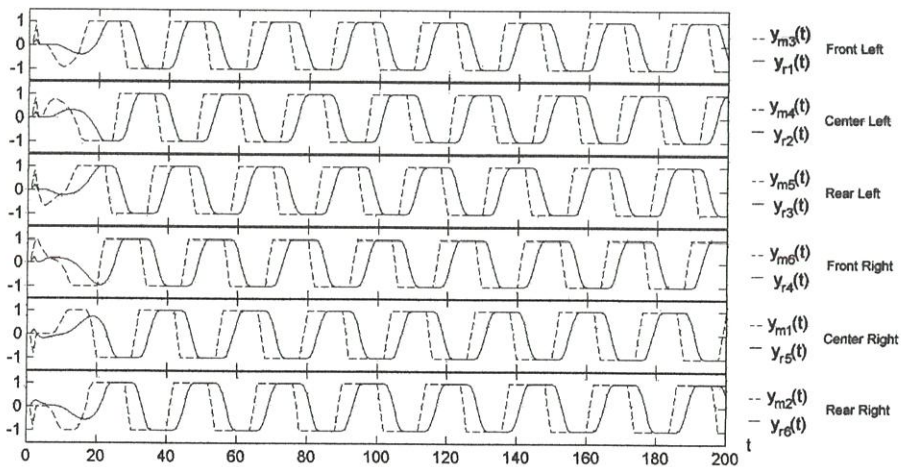
รูปที่ 3.14 เอาท์พุทของเซลล์ที่ 1 เมื่อกำหนดค่า  $\beta$  เท่ากับ 4 และเปลี่ยนค่า  $I_m$  จาก -2.5 ถึง 2.5



รูปที่ 3.15 เอาท์พุทของเซลล์ที่ 1 ให้แกน X เป็นเวลา เทียบกับแกน Y คือ  $I_m$

จากรูปค่าเอาท์พุทจะมีค่าระดับสูงสุดและต่ำสุดที่จุด 1V และ -1V ตามลำดับเมื่อค่า  $I_m$  เป็น 0 และมีค่าระดับต่ำสุดเพิ่มขึ้น เมื่อค่า  $I_m$  เพิ่มขึ้น จนเข้าสู่สภาวะสมำเสมอที่ 1V เมื่อ  $I_m = 2A$  เช่นเดียวกัน เมื่อ  $I_m$  ลดลง ค่าระดับสูงสุดของเอาท์พุทจะลดลงด้วย และเข้าสู่สภาวะสมำเสมอที่ -1V เมื่อ  $I_m = -2A$

ดังนั้นค่าพารามิเตอร์ในระบบนี้ ควรมีค่าเป็น  $\beta = 4$  เนื่องจากมีค่าความชันในการเปลี่ยนแปลงระหว่างค่าสูงสุดต่ำสุดเหมาะสมกับระบบ และให้ค่า  $I_m = 0 A$  เพื่อให้หุ่นประสาทเทียมแบบเซลล์ลาร์มีระยะยกขาเหมาะสมและสังเกตได้ง่าย เมื่อนำเฟสของสัญญาณมาเทียบกัน เพื่อให้สัญญาณสอดคล้องกันจึงปรับค่า  $\beta_1, \beta_2 = -4$  เพื่อให้มีเฟสที่กลับกับสัญญาณ  $\beta_1, \beta_2$  เดิม เมื่อนำค่าพารามิเตอร์มาแทนตามรูปที่ 3.7 โดยใช้ค่าที่กล่าวมาแล้ว และกำหนดให้แกน X คือเวลาแกน Y คือเอาท์พุทของแต่ละเซลล์จะได้ผลดังรูปที่ 3.16



รูปที่ 3.16 ผลการเคลื่อนที่ของขาหุ่นยนต์

จากรูปกำหนดให้ ค่าเอพท์พุตของ  $y_m(t)$  เป็น 1 หมายถึง ขาหุ่นยกขึ้นสุด และเป็น -1 ขาหุ่นจะยกลงต่ำสุด และ ค่าเอพท์พุตของ  $y_r(t)$  เป็น 1 หมายถึงขาหุ่นเคลื่อนที่ไปด้านหน้า และเป็น -1 ขาหุ่นเคลื่อนที่ไปด้านหลัง จะเห็นว่าหุ่นยนต์สามารถเคลื่อนที่ไปด้านหน้าได้เมื่อเกิดคลื่นดังในรูปที่ 3.16

เราได้ค่าพารามิเตอร์ที่เหมาะสมกับระบบ โดยได้ค่า  $\mu = 0.65$  และค่า  $\beta = 4$  และ  $I_r = 0$ ,  $I_m = 0$  ดังนั้นสมการระบบจึงเป็นดังนี้

$$\begin{aligned}
 \dot{x}_{r1} + x_{r1} &= \mu y_{r1} - \mu y_{r2} \\
 \dot{x}_{r2} + x_{r2} &= \mu y_{r2} - \mu y_{r3} \\
 \dot{x}_{r3} + x_{r3} &= \mu y_{r3} - \mu y_{r4} \\
 \dot{x}_{r4} + x_{r4} &= \mu y_{r4} - \mu y_{r5} \\
 \dot{x}_{r5} + x_{r5} &= \mu y_{r5} - \mu y_{r6} \\
 \dot{x}_{r6} + x_{r6} &= \mu y_{r6} + \mu y_{r1}
 \end{aligned} \tag{3.17}$$

และ

$$\begin{aligned}
 \dot{x}_{m1} + x_{m1} &= -\beta y_{r1} \\
 \dot{x}_{m2} + x_{m2} &= -\beta y_{r2} \\
 \dot{x}_{m3} + x_{m3} &= \beta y_{r3} \\
 \dot{x}_{m4} + x_{m4} &= \beta y_{r4} \\
 \dot{x}_{m5} + x_{m5} &= \beta y_{r5} \\
 \dot{x}_{m6} + x_{m6} &= \beta y_{r6}
 \end{aligned} \tag{3.18}$$

โดยสมการเอพท์พุตยังคงเป็นสมการที่ (3.12) และ (3.15) เช่นเดิม

จาก

$$x(n+1) = \frac{\dot{x}(n) + x(n)}{N} \tag{3.19}$$

เมื่อกำหนดให้  $N = 1$  หน่วย สมการที่ (3.19) จะเหลือเพียง

$$x(n+1) = \dot{x}(n) + x(n) \tag{3.20}$$

เมื่อแทนสมการ (3.20) ใน (3.17) และ (3.18) จะได้

$$\begin{aligned}
 x_{r1}(n+1) &= \mu y_{r1}(n) - \mu y_{r2}(n) \\
 x_{r2}(n+1) &= \mu y_{r2}(n) - \mu y_{r3}(n) \\
 x_{r3}(n+1) &= \mu y_{r3}(n) - \mu y_{r4}(n) \\
 x_{r4}(n+1) &= \mu y_{r4}(n) - \mu y_{r5}(n) \\
 x_{r5}(n+1) &= \mu y_{r5}(n) - \mu y_{r6}(n) \\
 x_{r6}(n+1) &= \mu y_{r6}(n) + \mu y_{r1}(n)
 \end{aligned} \tag{3.21}$$

และ

$$\begin{aligned}
 x_{m1}(n+1) &= -\beta y_{r1}(n) \\
 x_{m2}(n+1) &= -\beta y_{r2}(n) \\
 x_{m3}(n+1) &= \beta y_{r3}(n) \\
 x_{m4}(n+1) &= \beta y_{r4}(n) \\
 x_{m5}(n+1) &= \beta y_{r5}(n) \\
 x_{m6}(n+1) &= \beta y_{r6}(n)
 \end{aligned} \tag{3.22}$$

จากสมการที่ (3.21) และ (3.22) จะสามารถจำลองให้อยู่ในรูปโปรแกรมได้ง่ายขึ้น โดยกำหนดให้เซลล์ที่ใช้ควบคุมขาต่าง ๆ เป็นไปดังตารางที่ 3.1

ตารางที่ 3.1 เอ้าท์พุทของเซลล์ที่ใช้ควบคุมหุ่นยนต์ 6 ขาในตำแหน่งต่าง ๆ

ตำแหน่งขา		เคลื่อนที่หน้า-หลัง	เคลื่อนที่ขึ้น-ลง
หน้า	ซ้าย	$y_{r1}$	$y_{m3}$
	ขวา	$y_{r4}$	$y_{m6}$
กลาง	ซ้าย	$y_{r2}$	$y_{m4}$
	ขวา	$y_{r5}$	$y_{m1}$
หลัง	ซ้าย	$y_{r3}$	$y_{m5}$
	ขวา	$y_{r6}$	$y_{m2}$

### 3.5 การควบคุมหุ่นยนต์ โดยการจำลองโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ ในคอมพิวเตอร์ส่วนบุคคล

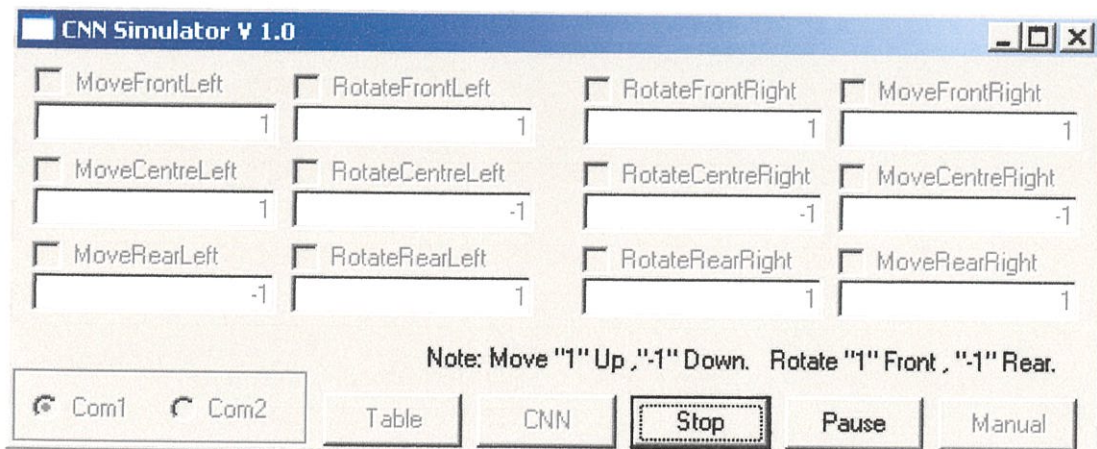
เพื่อทดสอบความถูกต้องในการออกแบบ จึงนำมาจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ูลาร์จากข้อมูลที่ได้ บนคอมพิวเตอร์ส่วนบุคคล โดยแบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนโปรแกรมจำลองโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ และส่วนควบคุมหุ่นยนต์ผ่าน RS232C ดังรูปที่ 3.17 โดยมีรายละเอียดดังนี้



รูปที่ 3.17 การจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ที่ออกแบบไว้เพื่อทดสอบค่าพารามิเตอร์

### 3.5.1 ส่วนโปรแกรมจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ในคอมพิวเตอร์

เป็นส่วนที่ทำการจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ โดยการกำหนดค่าพารามิเตอร์ต่าง ๆ ลงไปในโปรแกรม CNN Simulator โปรแกรมดังกล่าวจะทำหน้าที่คำนวณค่าของเอาต์พุตจากสมการที่ (3.21) และ (3.22) ของแต่ละเซลล์แล้วส่งไปให้ชุดควบคุมหุ่นยนต์ตามรหัสเซอร์โวมอเตอร์ที่ควบคุมจุดต่าง ๆ ของหุ่นยนต์ จากตารางที่ 3.1 ผ่าน พอร์ทอนุกรม RS232C ต่อไป



รูปที่ 3.18 โปรแกรมจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์บนคอมพิวเตอร์

โปรแกรมที่พัฒนาขึ้นนี้ จะคำนวณค่าเอาต์พุตของเซลล์อยู่ในรูปเลขจุดทศนิยม มีค่าตั้งแต่ -1 ถึง 1 โดยจะแสดงค่าที่คำนวณได้ในช่องของเซลล์นั้น ๆ จากนั้นจะทำการแปลงให้อยู่ในรูปพิกัดของเซอร์โวมอเตอร์เพื่อส่งไปให้ส่วนควบคุมหุ่นยนต์ผ่าน RS232C เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์

นอกจากโปรแกรมนี้ สามารถจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลาที่ออกแบบได้แล้ว ยังสามารถจำลองการทำงานแบบขั้นตอนวิธีการเปิดตารางได้อีกด้วย สำหรับรายละเอียดการทำงานของโปรแกรมกล่าวไว้ในภาคผนวก ค.

### 3.5.2 ส่วนควบคุมหุ่นยนต์ผ่าน RS232C

การทำงานในส่วนของการควบคุมหุ่นยนต์ผ่าน RS232C นั้น จะต้องทำหน้าที่รับข้อมูลรหัสเซอร์โวมอเตอร์และตำแหน่งที่ต้องการให้มอเตอร์แต่ละตัวหมุนไป จากส่วนโปรแกรมจำลองการทำงานของโครงข่ายประสาทเทียมแบบเซลล์ลูลา แล้วนำข้อมูลที่ได้ไปควบคุมตำแหน่งของเซอร์โวมอเตอร์ตามต้องการ บอร์ดควบคุมที่ให้นั้นใช้บอร์ดเดียวกับบอร์ดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาแบบขั้นตอนวิธีการเปิดตาราง เพียงแต่ใช้โปรแกรมควบคุมต่างกัน ส่วนการออกแบบโปรแกรมควบคุมหุ่นยนต์ผ่าน RS232C จะกล่าวถึงในภาคผนวก ค.

ผลที่ได้รับจากการจำลองข้างต้นนี้ พบว่าโครงข่ายประสาทเทียมแบบเซลล์ลูลาที่ได้ออกแบบไว้สามารถทำงานได้ แต่ต้องมีการปรับแต่งระยะเวลาการเคลื่อนที่ของเซอร์โวมอเตอร์ ให้เหมาะสมกับระยะเวลาการก้าวเดินของหุ่นยนต์ในการทดสอบอย่างคร่าว ๆ ปรากฏว่ามีพฤติกรรมบางอย่างที่เป็นข้อดีกว่าการควบคุมการเคลื่อนที่แบบขั้นตอนวิธีการเปิดตาราง ซึ่งจะได้ศึกษาและกล่าวโดยละเอียดต่อไปในบทที่ 5

## 3.6 สรุปท้ายบท

บทที่ 3 นี้กล่าวถึงทฤษฎีและการออกแบบโครงข่ายประสาทเทียมแบบเซลล์ลูลาที่นำมาควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขา รวมถึงผลกระทบของค่าพารามิเตอร์ต่าง ๆ ในระบบด้วย โดยจะเห็นว่า ค่าของ  $\mu$  มีผลต่อการเกิดคลื่นโดยตรง ส่วนค่า  $\beta$  จะมีผลต่อความเร็วในการยกขา ส่วนค่า  $\lambda$  จะทำให้จุดเริ่มต้นของการยกขาและคลื่นขาเปลี่ยนแปลง และยังมีผลต่อคาบเวลาการเกิดคลื่นอีกด้วยดังที่กล่าวมาแล้ว

ในบทที่ 4 จะกล่าวถึงการนำค่าพารามิเตอร์ที่ได้มาแล้ว ไปออกแบบเป็นวงจรแอนะล็อก ซึ่งเป็นวงจรที่จะนำไปควบคุมการเคลื่อนที่ของหุ่นยนต์ 6 ขาต่อไป

## บทที่ 4

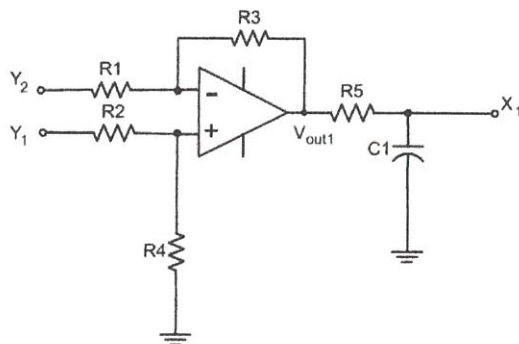
# การออกแบบวงจรจำลองโครงข่ายประสาทเทียมแบบ เซลล์ลูลาร์และการประยุกต์

การควบคุมการทำงานโดยวิธีการออกแบบวงจร CNN ซึ่งเป็นวงจรแอนะล็อก แม้จะมีความยุ่งยากมากกว่า และมีความยืดหยุ่นต่ำกว่าการออกแบบโดยอาศัยการจำลองเป็นโปรแกรมคอมพิวเตอร์ แต่มีข้อได้เปรียบในด้านความเร็วในการประมวลผล และยิ่งจะเห็นข้อแตกต่างได้ชัดเมื่อต้องการทำให้ระบบมีขนาดเล็กซึ่งอยู่ในรูปของไมโครคอนโทรลเลอร์ อีกทั้งการจำลองการทำงานเป็นไปได้ยาก เนื่องจากประสิทธิภาพในการประมวลผลของไมโครคอนโทรลเลอร์มีไม่สูงมากนัก แต่สำหรับวงจรแอนะล็อกแทบไม่พบปัญหาดังกล่าวเลยเนื่องจากการประมวลผลอยู่ในรูปของการประมวลสัญญาณแบบต่อเนื่องนั่นเอง นอกจากนี้ผลของสัญญาณรบกวนต่อคุณสมบัติของระบบยังมีน้อยกว่าไมโครคอนโทรลเลอร์อีกด้วย

บทนี้กล่าวถึงวิธีการออกแบบวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ และควบคุมหุ่นยนต์ 6 ขา โดยจะกล่าวถึงวิธีออกแบบเดิมที่นำเสนอโดย Paolo Arena และ Luigi Fortuna [10] จากนั้นจะนำเสนอวิธีการออกแบบใหม่ ซึ่งทำให้การออกแบบง่ายขึ้น

### 4.1 แนวคิดในการออกแบบวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์

เนื่องจากสมการสถานะของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ อยู่ในรูปของสมการเชิงเส้น อีกทั้งสมการเอาต์พุตเป็นสมการเชิงเส้นอิ่มตัว (Saturated Linear) ซึ่งง่ายต่อการนำเอาอปแอมป์มาประยุกต์ใช้งาน จากงานวิจัยของ Paolo Arena และ Luigi Fortuna ได้เสนอวิธีการแปลงสมการสถานะให้อยู่ในรูปวงจรรอบแอมป์ [5][10] เมื่อนำหลักการดังกล่าวมาประยุกต์ใช้งานกับสมการสถานะในสมการที่ (3.9) และ (3.10) โดยแบ่งวงจรเป็น 2 ส่วน ส่วนที่ 1 ได้แก่ส่วนขยายผลรวม- ผลต่าง และส่วนสมการเชิงเส้นอิ่มตัว กรณีผลต่างแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 วงจรขยายผลต่างส่วนหน้าของวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์

จากวงจรในรูปที่ 4.1 เมื่อพิจารณาในรูปสมการจะได้ว่า

$$C_1 \dot{X}_1 = \frac{V_{out1} - X_1}{R_5} \quad (4.1)$$

จากสมการออปแอมป์[11]

$$V_{out1} = \frac{R_4}{R_2} Y_1 - \frac{R_3}{R_1} Y_2 \quad (4.2)$$

จะได้ว่า

$$C_1 \dot{X}_1 = -\frac{X_1}{R_5} + \frac{R_4}{R_2 R_5} Y_1 - \frac{R_3}{R_1 R_5} Y_2 \quad (4.3)$$

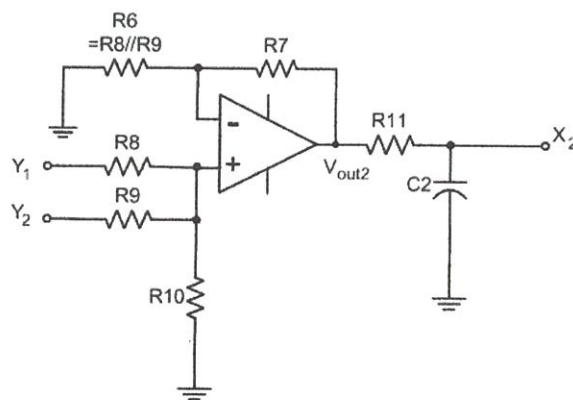
หรือ

$$C_1 R_5 \dot{X}_1 = -X_1 + \frac{R_4}{R_2} Y_1 - \frac{R_3}{R_1} Y_2 \quad (4.4)$$

เมื่อเทียบสัมประสิทธิ์กับสมการ (3.10) ได้ว่า

$$\mu = \left( \frac{R_4}{R_2} - 1 \right) \quad , \quad s = \frac{R_3}{R_1} \quad \text{และ} \quad C_1 R_5 \quad \text{เป็นตัวกำหนดคาบ} \quad (4.5)$$

สำหรับกรณีขยายผลบวกแสดงวงจรได้ดังรูปที่ 4.2



รูปที่ 4.2 วงจรขยายผลบวกส่วนหน้าของวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูการ์

ในการทำงานเดียวกันวงจรในรูปที่ 4.2 เมื่อพิจารณาในรูปสมการจะได้ว่า

$$C_2 \dot{X}_2 = \frac{V_{out2} - X_2}{R_{11}} \quad (4.6)$$

จากสมการออปแอมป์[11]

$$V_{out2} = \frac{R_{10}}{R_8} Y_1 + \frac{R_{10}}{R_9} Y_2 \quad (4.7)$$

จะได้ว่า

$$C_2 \dot{X}_2 = -\frac{X_1}{R_{11}} + \frac{R_{10}}{R_8 R_{11}} Y_1 + \frac{R_{10}}{R_9 R_{11}} Y_2 \quad (4.8)$$

หรือ

$$C_2 R_{11} \dot{X}_2 = -X_1 + \frac{R_{10}}{R_8} Y_1 + \frac{R_{10}}{R_9} Y_2 \quad (4.9)$$

เมื่อเทียบสัมประสิทธิ์กับสมการ (3.10) ได้ว่า

$$\mu = \left( \frac{R_{10}}{R_9} - 1 \right) \quad , \quad s = \frac{R_{10}}{R_8} \quad \text{และ} \quad C_2 R_{11} \quad \text{เป็นตัวกำหนดคาบ} \quad (4.10)$$

สำหรับส่วนที่ 2 คือส่วนสมการเอาท์พุท ซึ่งเป็นสมการเชิงเส้นอิมพัลส์ มีหลักการสร้างง่าย ๆ โดยนำสัญญาณที่ได้มาขยายโดยออปแอมป์ ด้วยอัตราขยายสูง ๆ ให้เกิดปรากฏการณ์ Overdrive จากนั้นนำสัญญาณเอาท์พุทที่ได้มาผ่านวงจรแบ่งแรงดันที่มีอัตราส่วนการลดทอนเท่ากับอัตราขยายของออปแอมป์ โดยมีสมการดังนี้

$$V_{out} = \frac{R_{15}}{R_{13}} X \quad (4.11)$$

และ

$$Y = \frac{R_{17}}{R_{16} + R_{17}} V_{out} \quad (4.12)$$

ดังนั้น

$$Y = \frac{R_{15}R_{17}}{(R_{16} + R_{17})R_{13}} X \quad (4.13)$$

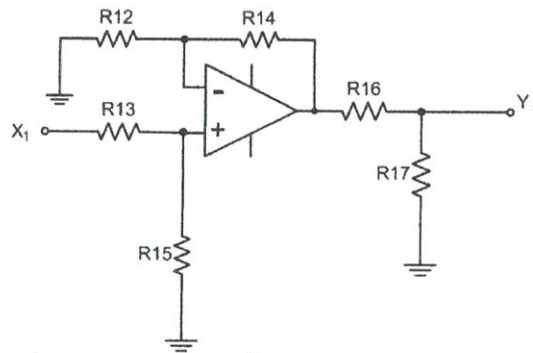
เนื่องจากอัตราขยายโดยรวมต้องเท่ากับ 1 ดังนั้น

$$R_{15}R_{17} = (R_{16} + R_{17})R_{13} \quad (4.14)$$

จากสมการที่ (4.12) หนึ่งกรณีที่เป็นไปได้คือ

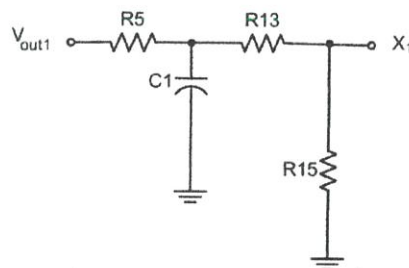
$$R_{13} = kR_{17} \quad \text{และ} \quad R_{15} = k(R_{16} + R_{17}) \quad (4.15)$$

เมื่อ  $k$  คือค่าคงที่ใด ๆ ซึ่งจะได้วงจรดังรูปที่ 4.3



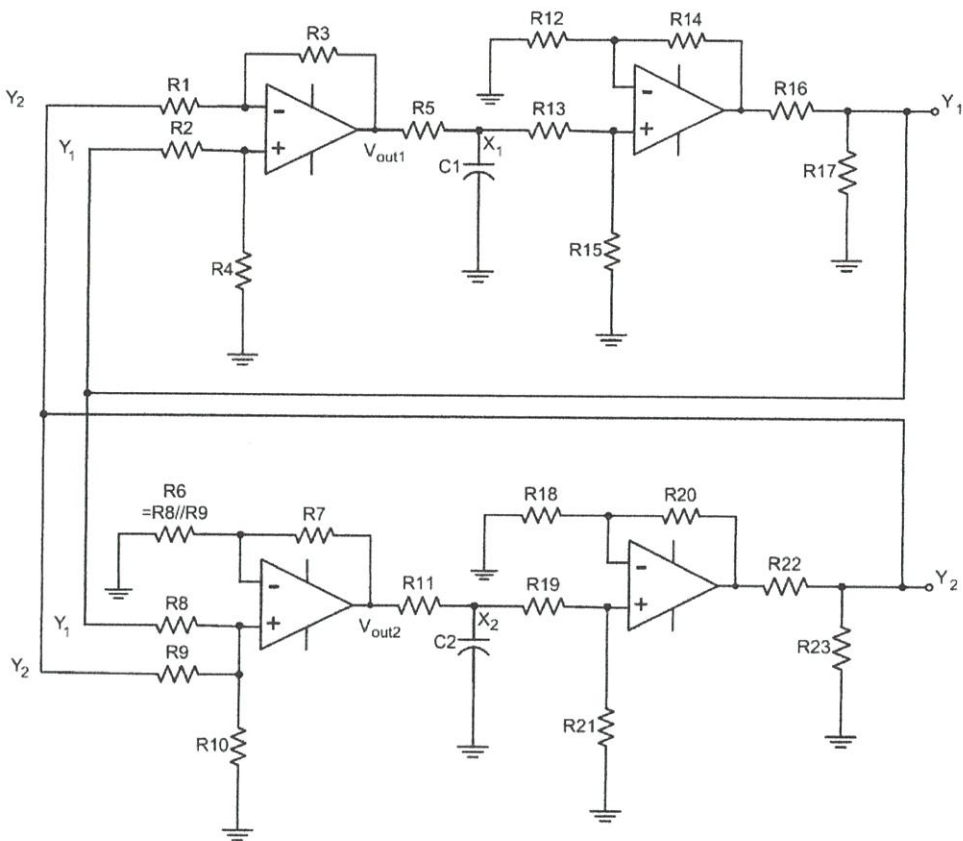
รูปที่ 4.3 วงจรเชิงเส้นอิมิตัวสำหรับสมการเอาร์ทพุต

เมื่อนำวงจรทั้งหมดมารวมกัน ก็จะได้วงจรเซลล์ทั้งหมด เมื่อพิจารณาจากวงจรในรูปที่ 4.4 ซึ่งเป็นจุดต่อของวงจร จะเห็นว่า ค่าของ  $R_7, R_9$  มีผลค่าคาบ  $R_5C_1$  ด้วย



รูปที่ 4.4 ปัญหาจุดเชื่อมต่อของเอาร์ทพุตส่วนที่ 1 กับอินพุตส่วนที่ 2

จากรูปที่ 4.4 จะเห็นว่า เมื่อค่า  $R_{13} + R_{15}$  มีค่าใกล้เคียงหรือต่ำกว่าค่า  $R_5$  เมื่อ  $C_1$  คาย ประจุจะมีประจุส่วนหนึ่ง คายผ่าน  $R_{13}$  และ  $R_{15}$  ทำให้สัญญาณขาลงมีช่วงเวลาที่สูงกว่าช่วงเวลาขาขึ้น เพื่อหลีกเลี่ยงปัญหาดังกล่าว ในการออกแบบค่าของ  $R_{13}, R_{15}$  จำเป็นต้องมีค่าสูงมาก ๆ เมื่อเทียบกับค่า  $R_5$  และในทำนองเดียวกันค่า  $R_{19}, R_{21}$  ก็จะต้องสูงกว่าค่า  $R_{11}$  มาก ๆ เช่นกัน เมื่อนำวงจรทั้งหมดมารวมกัน ก็จะได้วงจрдังรูปที่ 4.5



รูปที่ 4.5 วงจรจำลองโครงข่ายประสาทเทียมแบบเซลลูลาร์แบบ 2 เซลล์

เพื่อให้ง่ายในการออกแบบ จึงได้พัฒนางจรใหม่ โดยอาศัยหลักการนำมาตราส่วนจากเดิมให้ขอบเขตสัญญาณอยู่ที่ 1 ถึง -1 เป็นขอบเขตสัญญาณอยู่  $V_{sat}$  ถึง  $-V_{sat}$  เมื่อ  $V_{sat}$  คือค่าแรงดันอิ่มตัวของออปแอมป์ทางด้านบวก และ  $-V_{sat}$  คือค่าแรงดันอิ่มตัวของออปแอมป์ทางด้านลบตามลำดับ ซึ่งมีผลทำให้สมการ (4.5) และ (4.10) เปลี่ยนไปดังนี้

$$X_1 \Rightarrow X_1 V_{sat} \quad (4.16)$$

จากสมการ (4.4) จะได้ว่า

$$\frac{C_1 R_5}{V_{sat}} \dot{X}_1 = -X_1 + \frac{R_4}{R_2 V_{sat}} Y_1 - \frac{R_3}{R_1 V_{sat}} Y_2 \quad (4.17)$$

ดังนั้น

$$\mu = \left( \frac{R_4}{R_2 V_{sat}} - 1 \right), \quad s = \frac{R_3}{R_1 V_{sat}} \quad \text{และ} \quad \frac{C_1 R_5}{V_{sat}} \quad \text{เป็นตัวกำหนดคาบ} \quad (4.18)$$

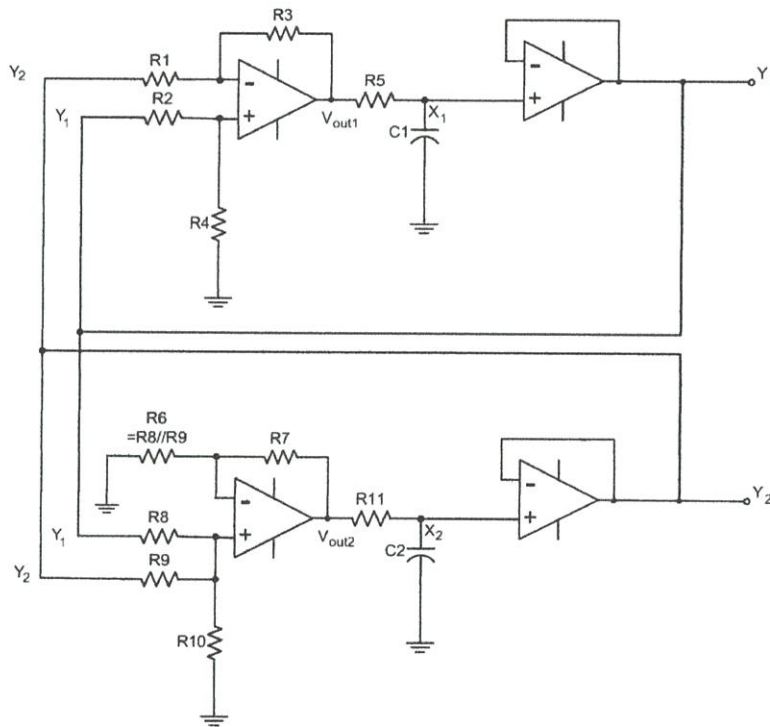
ในทำนองเดียวกันจากสมการ (4.9) จะได้ว่า

$$\frac{C_2 R_{11}}{V_{sat}} \dot{X}_2 = -X_2 + \frac{R_{10}}{R_8 V_{sat}} Y_1 + \frac{R_{10}}{R_9 V_{sat}} Y_2 \quad (4.20)$$

ดังนั้น

$$\mu = \left( \frac{R_{10}}{R_9 V_{sat}} - 1 \right), \quad s = \frac{R_{10}}{R_8} \quad \text{และ} \quad \frac{C_2 R_{11}}{V_{sat}} \quad \text{เป็นตัวกำหนดคาบ} \quad (4.21)$$

จากวิธีการดังกล่าวจะทำให้ภาคสมการเชิงเส้นอิมพัลส์ มีคุณสมบัติเป็นวงจรมัลติเพล็กซ์ไม่กลับเฟส ซึ่งมีความต้านทานอินพุตที่สูงมาก จึงไม่มีผลกระทบกับค่าคาบของสัญญาณ ดังนั้นจะได้วงจรใหม่ดังรูปที่ 4.6

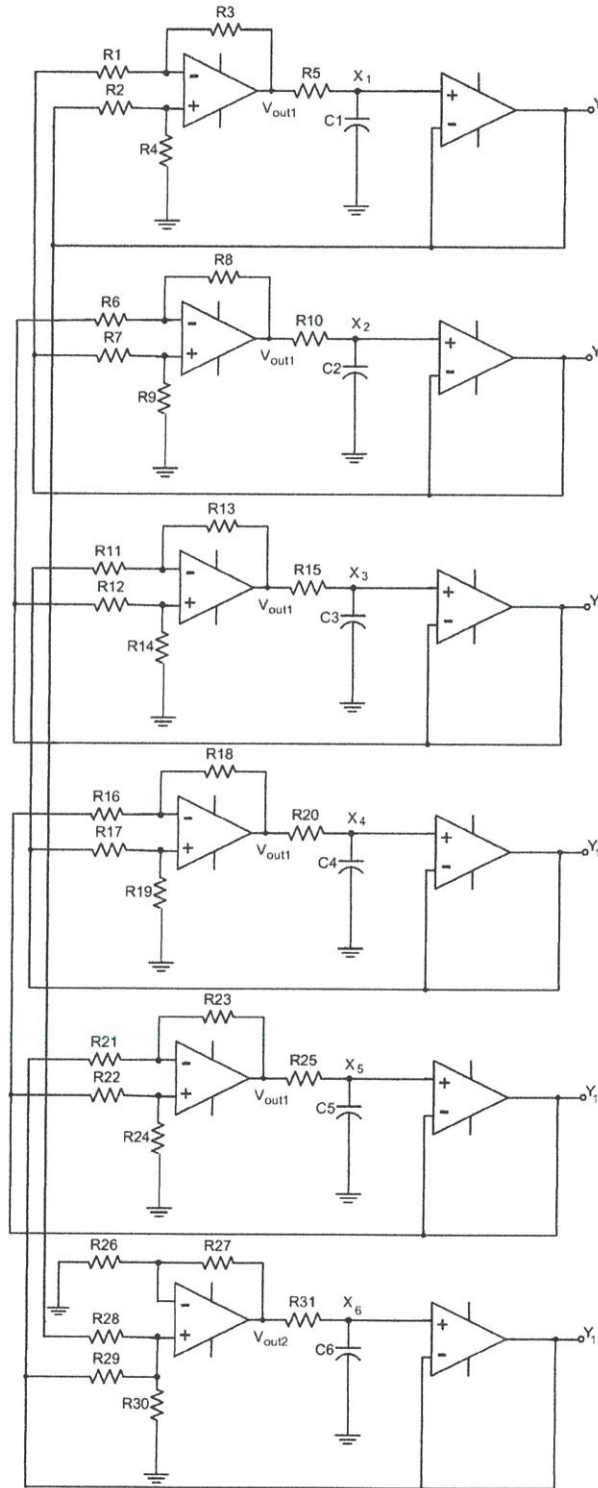


รูปที่ 4.6 วงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ูลาร์แบบ 2 เซลล์ที่ออกแบบใหม่

แม้ว่าการออกแบบโดยใช้แรงดันอิมพัลส์ของออปแอมป์ อาจทำให้เกิดปรากฏการณ์บางอย่าง เช่น การเกิดค่าพุ่งเกิน (Overshoot) ขึ้นเมื่อมีการเปลี่ยนระดับสัญญาณอย่างรวดเร็ว หรือ อาจเกิดความถี่ฮาร์โมนิกขึ้นเนื่องจากสัญญาณถูกตัดออกซึ่งจะเป็นสัญญาณรบกวนในระบบ [20] แต่เนื่องจากวงจรควบคุมโดยใช้โครงข่ายประสาทเทียมแบบเซลล์ูลาร์นั้น มีคาบความถี่ที่ต่ำมาก ทำให้ปรากฏการณ์ดังกล่าวมีผลกระทบต่อการทำงานน้อยมากจนสามารถละเลยได้

## 4.2 วงจรจำลองเซลล์ควบคุมการก้าวขา

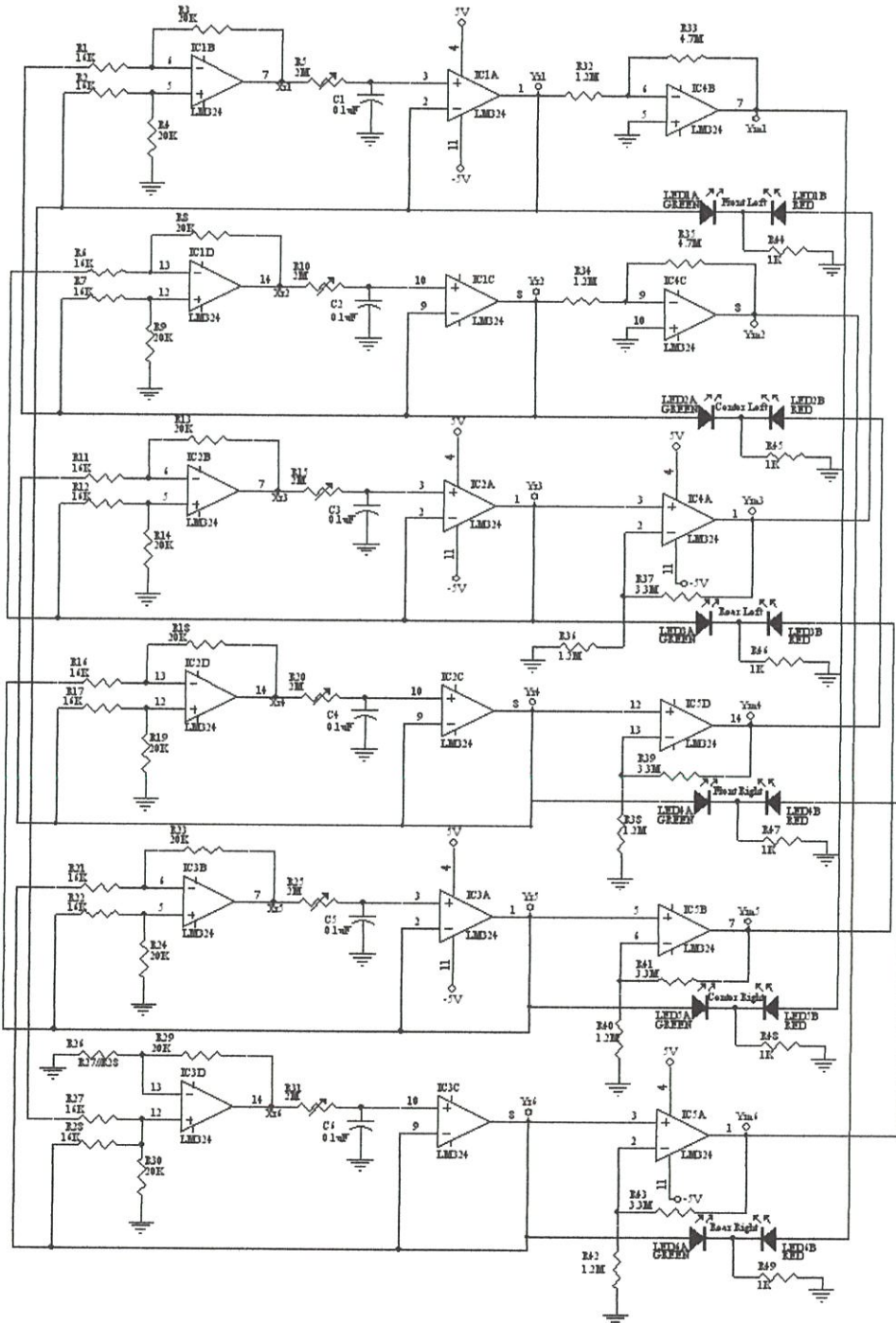
ในการออกแบบวงจรจำลองเซลล์ควบคุมการก้าวขานั้น อาศัยหลักการในหัวข้อ 4.1 และสมการ (3.12) และ (3.13) โดยกำหนดให้ค่า  $I$  เป็น 0 และ ค่า  $\mu \geq 0.65$  ซึ่งได้วงจรตามรูปที่ 4.7



รูปที่ 4.7 วงจรจำลองเซลล์ควบคุมการก้าวขาทั้ง 6 ขา

### 4.3 วงจรจำลองเซลล์ควบคุมการยกขา

การออกแบบวงจรถักขาหนี้อาศัยสมการ (3.14) และ (3.15) ในการออกแบบ เซลล์ชุดนี้ไม่ได้ทำหน้าที่ในการสร้างคลื่นเป็นแต่ด้วยขยายสัญญาณจากชุดเซลล์ควบคุมการก้าวเท่านั้น จึงมีลักษณะเป็นวงจขยายสัญญาณโดยจะแยกเป็น 2 กรณีคือ กรณีที่มีค่า  $\beta$  เป็นบวก และ กรณีที่มีค่า  $\beta$  เป็นลบ กรณี  $\beta$  เป็นบวก วงจรจะจัดอยู่ในรูปวงจขยายแบบไม่กลับเฟส ส่วนกรณีค่า  $\beta$  เป็นลบ จะเป็นวงจขยายกลับเฟสนั่นเอง เมื่อรวมกับเซลล์ก้าวขาแล้วจะได้วงจรในรูปที่ 4.8



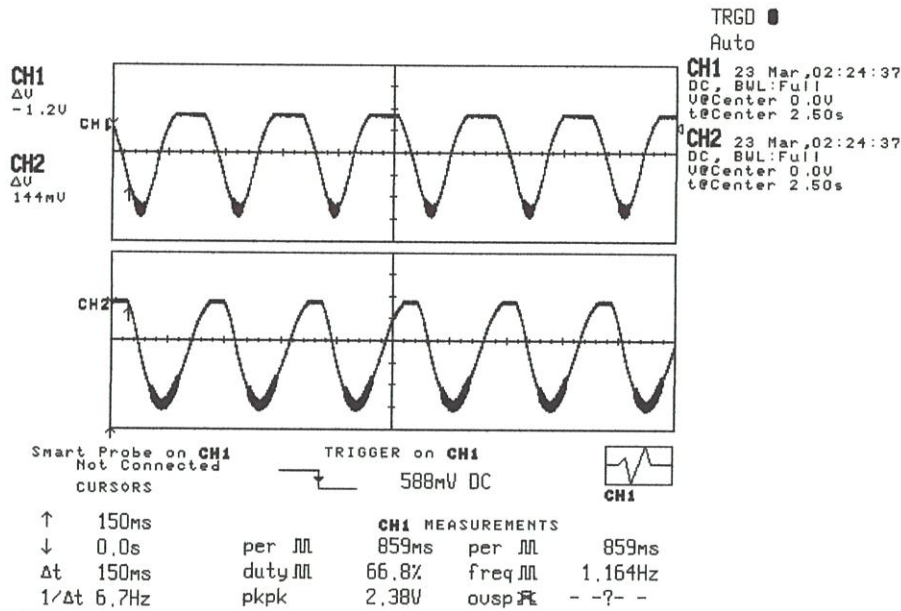
รูปที่ 4.8 วงจรจำลองเซลล์ควบคุมหุ่นยนต์ 6 ขาทั้งหมด

#### 4.4 ผลการทดสอบวงจร

เมื่อทดสอบวัดผลที่ได้จากวงจรที่ประกอบขึ้นในสถานะต่าง ๆ กันได้ผลดังนี้

##### 4.4.1 สัญญาณจากวงจรโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์แบบ 2 เซลล์

เมื่อทดสอบต่อวงจรตามรูปที่ 4.6 แล้วทดสอบวัดสัญญาณเอาต์พุต โดยกำหนดให้ช่องสัญญาณที่ 1 ต่อกับเอาต์พุตของเซลล์ที่ 1 และ ช่องสัญญาณที่ 2 ต่อกับเอาต์พุตของเซลล์ที่ 2 ได้ผลดังนี้



รูปที่ 4.9 เอาต์พุตของวงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์แบบ 2 เซลล์

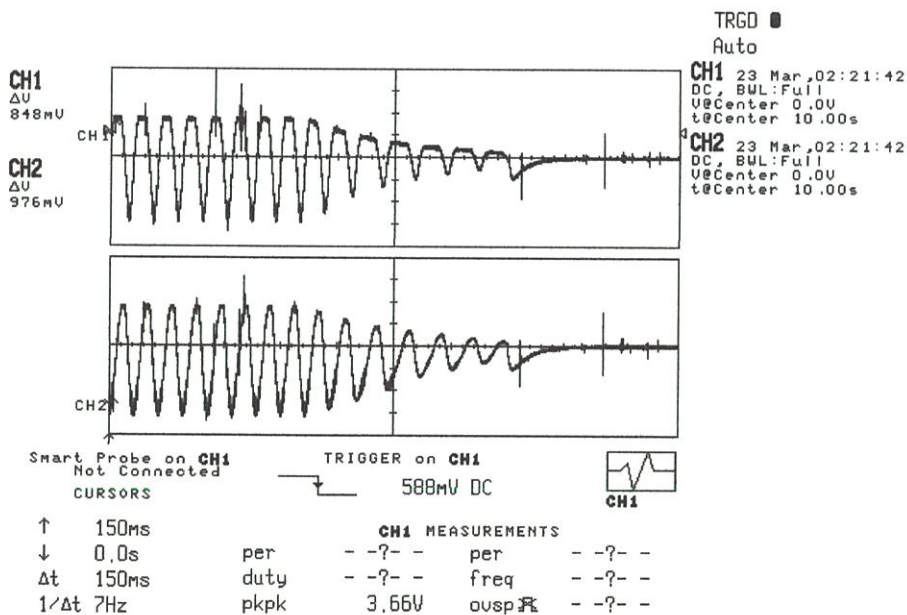


รูปที่ 4.10 เอาต์พุตในภาวะเริ่มต้นของวงจรจำลองเซลล์ 2 เซลล์

ในรูปที่ 4.9 แสดงเอาต์พุตในสภาวะเสถียร โดยตั้งค่าแรงดันต่อช่องไว้ที่ 0.5V และตั้งเวลาต่อช่องไว้ที่ 0.5 วินาที จะเห็นว่าสัญญาณที่ออกมาเป็นคลื่นความถี่ต่ำมากประมาณ 2.4 Hz และเอาต์พุตทั้งสองจะมีลักษณะเหมือนกัน การเหลื่อมกันนี้เป็นสาเหตุให้เกิดคลื่นขึ้น เนื่องจากเมื่อสัญญาณจากเซลล์ที่ 2 ถูกป้อนกลับไปเซลล์ที่ 1 จะไปรวมกับสัญญาณที่ถูกป้อนกลับมาจากเซลล์ที่ 1 เอง ในขณะที่ สัญญาณเอาต์พุตของเซลล์ที่ 2 ได้จากการหักล้างกันของเซลล์ที่ 1 และเซลล์ที่ 2 ดังนั้นจึงเกิดเป็นสภาวะไม่เสถียรขึ้นนั่นเอง

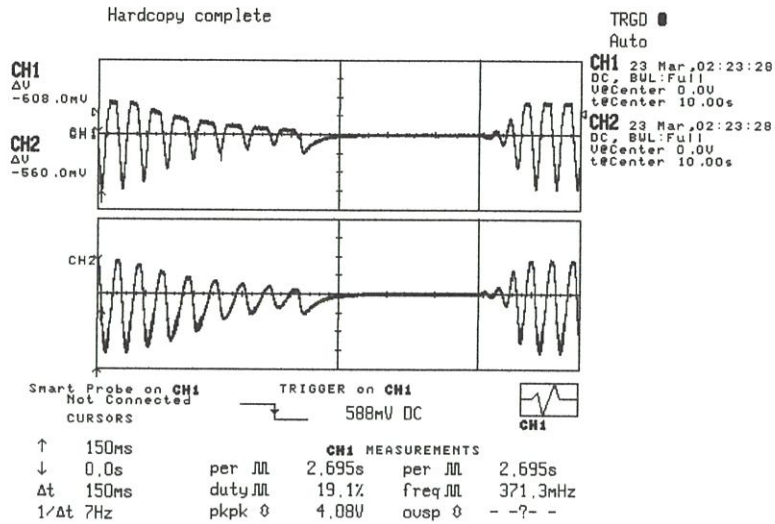
ในรูปที่ 4.10 เป็นสภาวะเริ่มต้นของเซลล์ เมื่อเริ่มจ่ายไฟเลี้ยงวงจรเพื่อศึกษาสภาวะเริ่มต้นของวงจร จะเห็นว่าต้องใช้เวลาช่วงเริ่มต้นในการสร้างคลื่นอยู่ระยะหนึ่งจึงจะสามารถผลิตคลื่นได้ โดยตั้งค่าแรงดันต่อช่องไว้ที่ 0.5V และตั้งเวลาต่อช่องไว้ที่ 2 วินาที

จะเห็นว่าการเกิดคลื่นจะอาศัยซึ่งกันและกันระหว่างเซลล์ที่ 1 และเซลล์ที่ 2 โดยขนาดของสัญญาณที่เซลล์ทั้งสองจะเพิ่มขึ้นทีละน้อย จนกระทั่งถึงค่าอิ่มตัวโดยจะมีเฟสที่เหลื่อมกันดังเหตุผลที่อธิบายไปแล้วข้างต้น



รูปที่ 4.11 เอาต์พุตในภาวะลดไฟเลี้ยงของวงจร

รูปที่ 4.11 เอาต์พุตเมื่อลดไฟเลี้ยงวงจร เพื่อศึกษาผลกระทบของไฟเลี้ยงต่อวงจร โดยตั้งค่าแรงดันต่อช่องไว้ที่ 0.5V และตั้งเวลาต่อช่องไว้ที่ 2 วินาที จะเห็นว่าความบิดเบี้ยวของสัญญาณจะมีมากขึ้นเมื่อไฟเลี้ยงวงจรลดลง และเป็นศูนย์ในที่สุด ทั้งนี้ค่าการเยื้องเฟสของเอาต์พุตทั้งสองยังคงมีอยู่จนกระทั่งสัญญาณหายไป

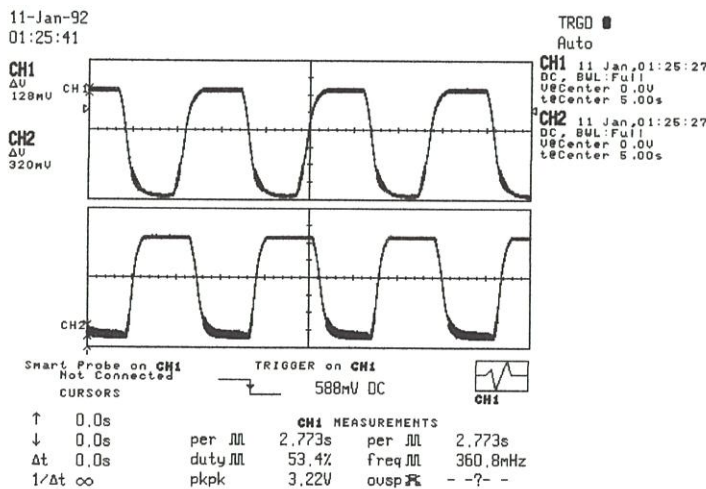


รูปที่ 4.12 เอาท์พุตในภาวะตัดไฟเลี้ยงของวงจรชั่วคราว

รูปที่ 4.12 เป็นการตัดไฟเลี้ยงวงจรชั่วคราว โดยตั้งค่าแรงดันต่อช่องไว้ที่ 0.5V และตั้งเวลาต่อช่องไว้ที่ 2 วินาที จะเห็นว่าการตัดไฟเลี้ยง ไม่มีผลต่อการทำงานของวงจรเลย โดยจะมีการผลิตสัญญาณในลักษณะเช่นเดียวกันกับรูปที่ 4.10 อีกครั้งเมื่อได้รับไฟเลี้ยงวงจรและเฟสของสัญญาณก็มีการเอียงกันด้วยค่าปกติ

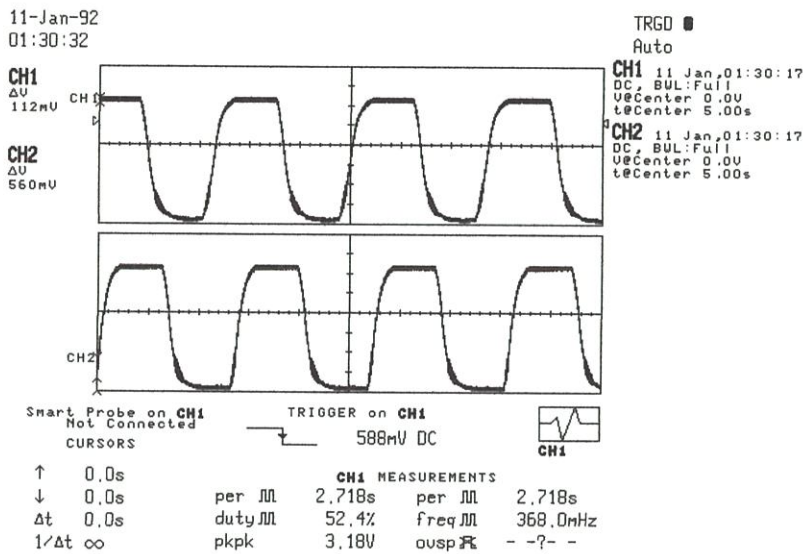
#### 4.4.2 สัญญาณจากวงจรโครงข่ายประสาทเทียมแบบเซลล์ลูร์เพื่อควบคุมหุ่นยนต์

เป็นการวัดเอาท์พุตจากวงจรในรูปที่ 4.8 โดยแบ่งออกเป็น 2 ช่วงคือ การเปรียบเทียบเฟสของเอาท์พุตแต่ละเซลล์ ซึ่งแสดงในรูปที่ 4.13รูปที่ 4.17 จะเป็นการวัดค่าเอาท์พุตของเซลล์ที่ 1 เปรียบเทียบกับทุก ๆ เซลล์เรียงทีละเซลล์ โดยเซลล์ที่ 1 อยู่ของสัญญาณที่ 1 การวัดนี้ ตั้งค่าแรงดันต่อช่องไว้ที่ 0.5V และตั้งเวลาต่อช่องไว้ที่ 1 วินาทีทุกรูป

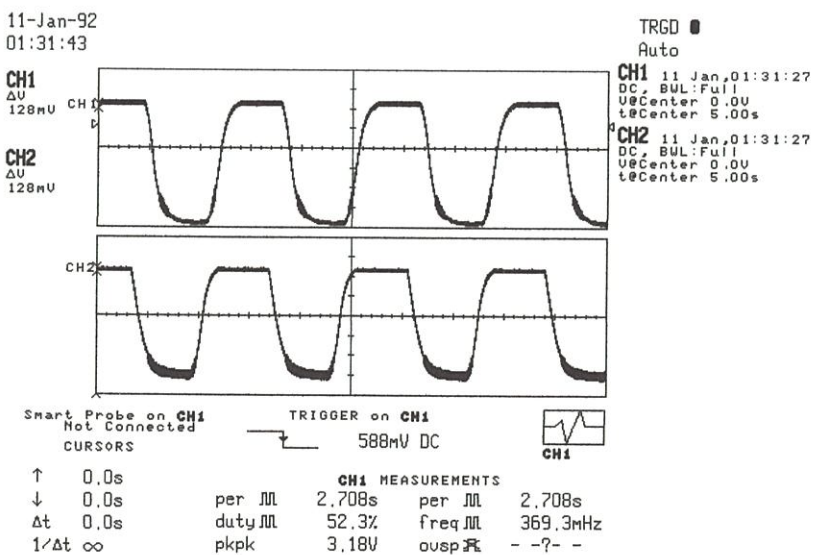


รูปที่ 4.13 เอาท์พุตของเซลล์ก้าวขา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 2

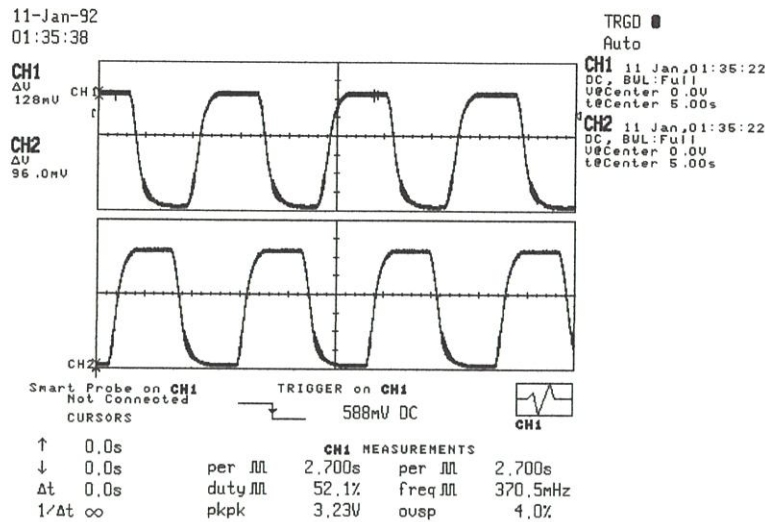
จากรูปที่ 4.13 เมื่อเปรียบเทียบเอาต์พุตของเซลล์ที่ 1 กับเซลล์ที่ 2 จะเห็นว่าเอาต์พุตของเซลล์ที่ 2 จะตามหลังเอาต์พุตของเซลล์ที่ 1 อยู่ เกือบจะ 90° ซึ่งเป็นไปตามที่จำลองได้ในรูปที่ 3.6 และเพื่อเป็นการง่ายในการตรวจสอบเฟส ในทำนองเดียวกันรูปที่ 4.14 - 4.17 จะเป็นการเปรียบเทียบสัญญาณเอาต์พุตของเซลล์ที่ 1 กับสัญญาณเอาต์พุตของเซลล์ที่ 3-6 ตามลำดับ จะเห็นว่าเฟสของเซลล์จะเลื่อนเพิ่มขึ้นไปเซลล์ละประมาณเกือบจะ 90° ทุก ๆ เซลล์ ซึ่งตรงตามรูปที่ 3.6 นั้นเอง



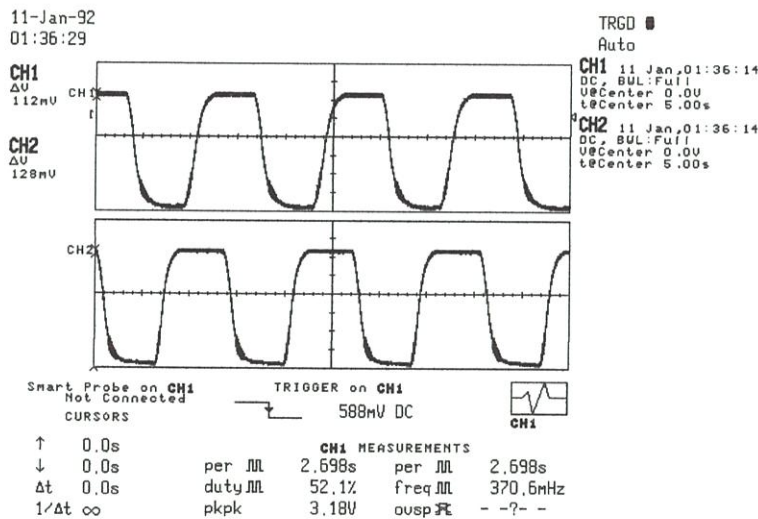
รูปที่ 4.14 เอาต์พุตของเซลล์กัววา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 3



รูปที่ 4.15 เอาต์พุตของเซลล์กัววา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 4



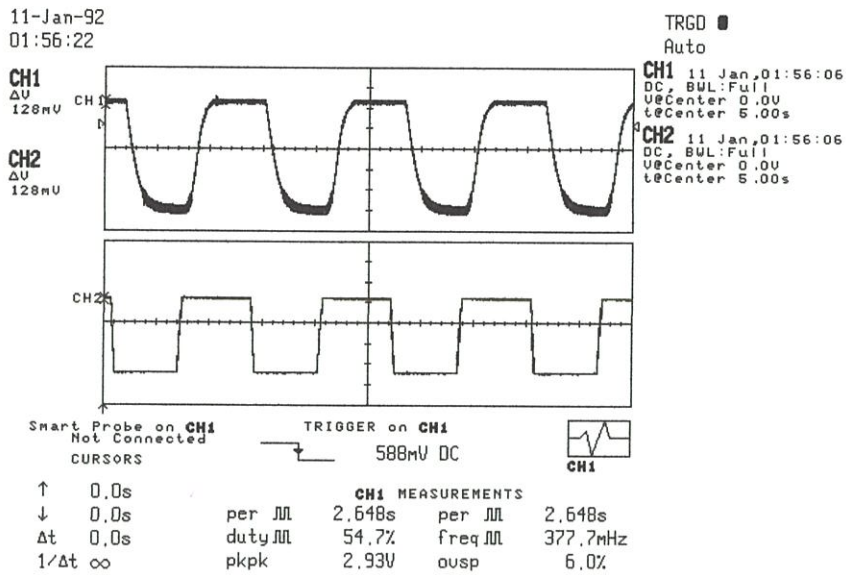
รูปที่ 4.16 เอาท์พุทของเซลล์กัววา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 5



รูปที่ 4.17 เอาท์พุทของเซลล์กัววา เซลล์ที่ 1 เปรียบเทียบกับเซลล์ที่ 6

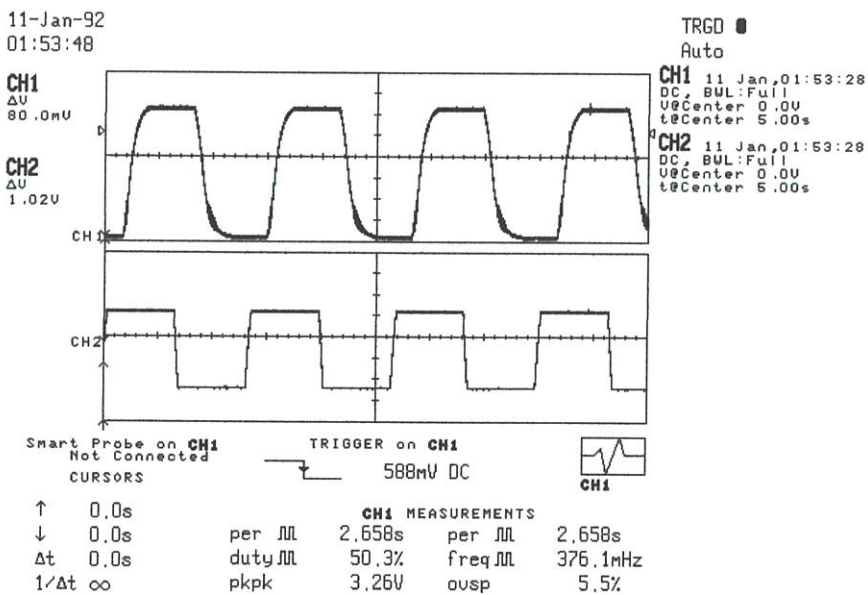
จากนั้นนำสัญญาณเอาท์พุทของเซลล์ที่ได้ข้างต้น ไปควบคุมการหมุนขาตามตารางที่ 3.1 แล้วนำไปเป็นอินพุทของเซลล์ยกขาโดยอาศัยตารางที่ 3.1 จากนั้นได้ทำการวัดสัญญาณเปรียบเทียบกันระหว่างเอาท์พุทเซลล์หมุนขา กับเซลล์ยกขาได้ดังนี้

รูปที่ 4.18 รูปที่ 4.23 แสดงการเปรียบเทียบระหว่างเอาท์พุทของเซลล์กัววาและเซลล์ยกขาแต่ละขา โดยค่า 1 ของเซลล์กัววาหมายถึงการก้าวไปข้างหน้า -1 หมายถึงการหมุนขามาด้านหลังและค่า 1 ของเซลล์ยกขาหมายถึงการยกขึ้น -1 หมายถึงกดลง ในการวัดกำหนดให้เอาท์พุทของเซลล์กัววาอยู่ช่องสัญญาณที่ 1 และ เอาท์พุทของเซลล์ยกขาอยู่ช่องสัญญาณที่ 2 ตั้งค่าแรงดันต่อช่องที่ช่องสัญญาณที่ 1 ไว้ที่ 0.5V ช่องสัญญาณที่ 2 ไว้ที่ 1V และตั้งเวลาต่อช่องไว้ที่ 1 วินาทีทุกรูป

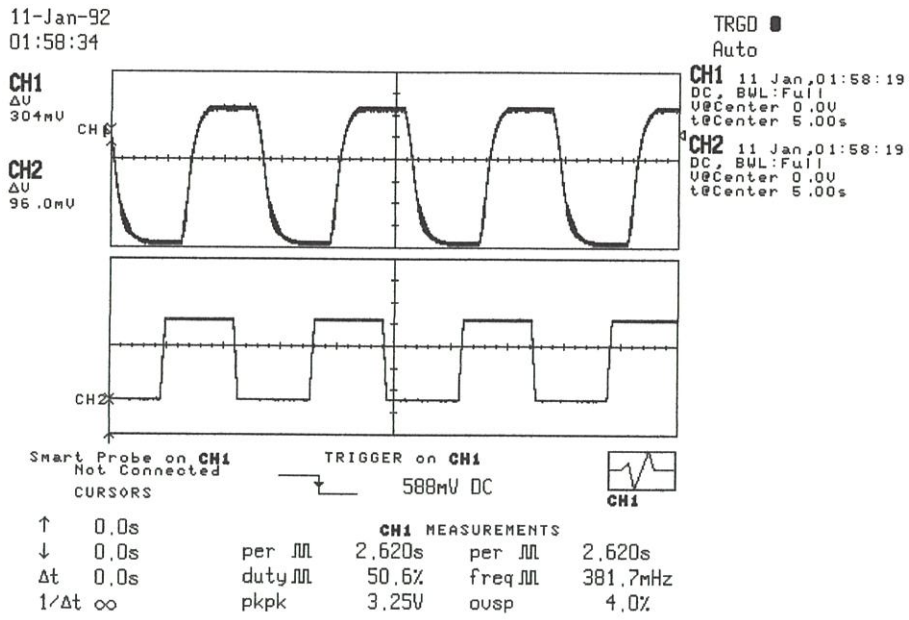


รูปที่ 4.18 สัญญาณเอาต์พุตที่ขาหุ่นยนต์ด้านหน้าซ้าย

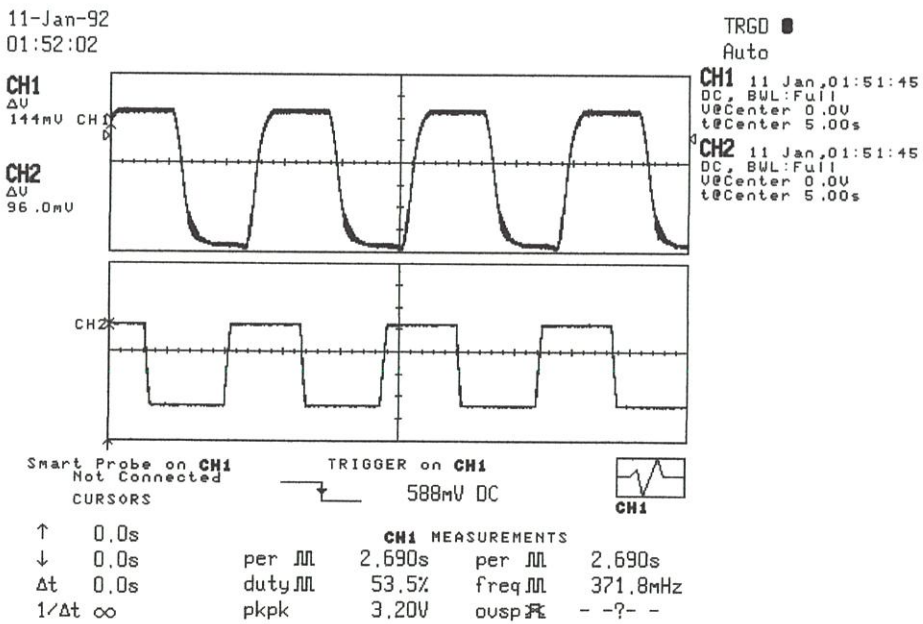
จากรูปที่ 4.18 ซึ่งเป็นสัญญาณควบคุมขาซ้ายหน้าจะเห็นว่าสัญญาณยกขาจะมีเฟสนำหน้าสัญญาณการก้าวขา โดยสัญญาณด้านบวกของการยกขาจะตรงกับช่วงที่สัญญาณก้าวขาเปลี่ยนจากลบเป็นบวก ทั้งนี้เนื่องจากช่วงที่ขาก้าวจากบวกไปลบเป็นช่วงที่ขาหมุนจากด้านหลังไปด้านหน้าซึ่งขาจะต้องถูกยกขึ้นนั่นเอง



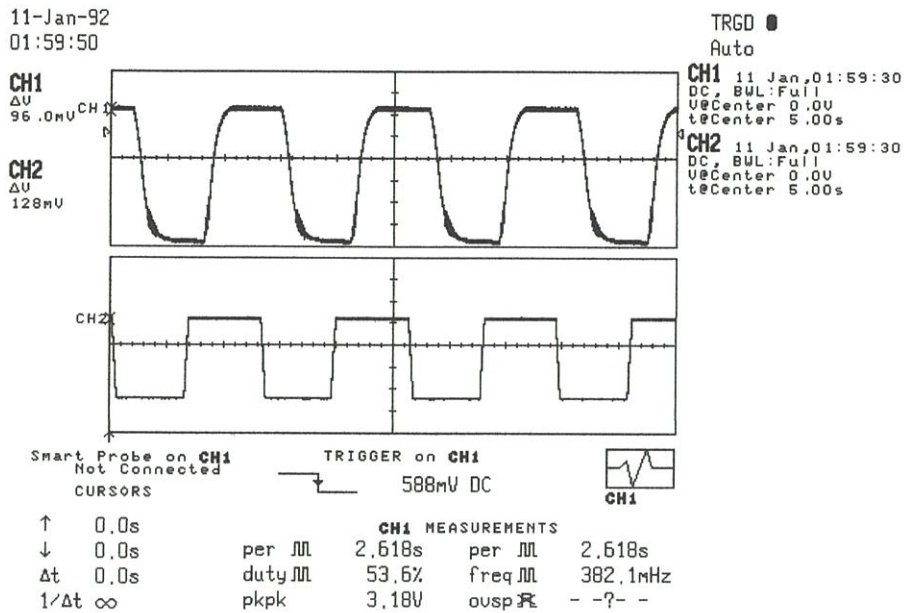
รูปที่ 4.19 สัญญาณเอาต์พุตที่ขาหุ่นยนต์ด้านหน้าขวา



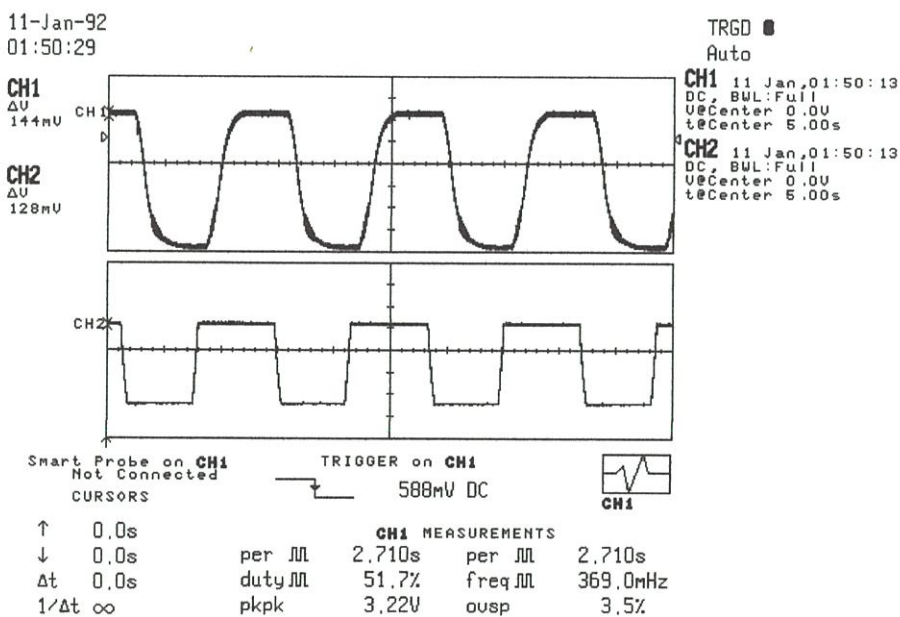
รูปที่ 4.20 สัญญาณเอาต์พุตที่ขาหุ่นยนต์ตรงกลางด้านซ้าย



รูปที่ 4.21 สัญญาณเอาต์พุตที่ขาหุ่นยนต์ตรงกลางด้านขวา



รูปที่ 4.22 สัญญาณเอาต์พุตที่ขาหุ่นยนต์ด้านหลังซ้าย

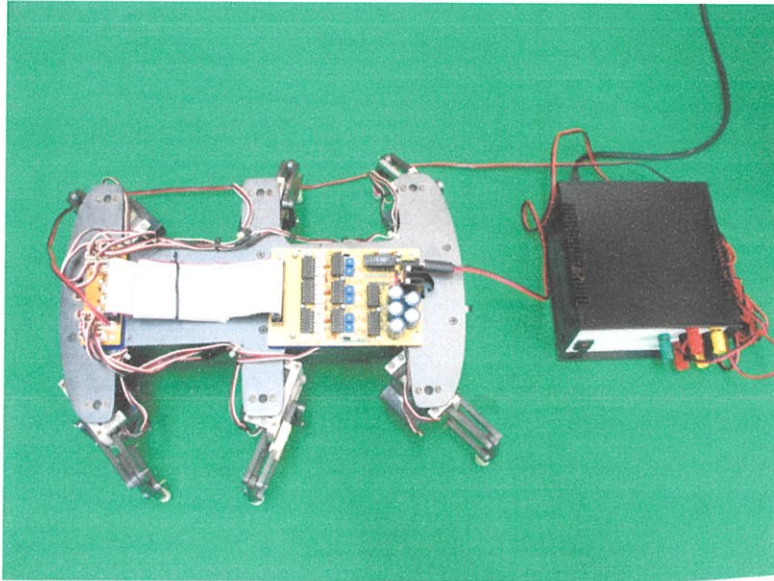


รูปที่ 4.23 สัญญาณเอาต์พุตที่ขาหุ่นยนต์ด้านหลังขวา

ในการทำงานเดียวกัน รูปที่ 4.19 - 4.23 เป็นการวัดสัญญาณควบคุมขาต่าง ๆ โดยตั้งค่าการวัดเช่นเดียวกันกับรูปที่ 4.18 จะเห็นว่าการจัดเฟสของสัญญาณควบคุมการก้าวขาและสัญญาณควบคุมการยกขาจะมีลักษณะการเยื้องที่เท่ากัน ดังนั้น การเคลื่อนที่ของขาหุ่นยนต์แต่ละขาจะมีลักษณะอย่างเดียวกัน แต่จะมีจังหวะการก้าวที่ไม่พร้อมกัน โดยลำดับการก้าวขึ้นอยู่กับลำดับเฟสของเอาต์พุตเซลล์ควบคุมการก้าวขา ซึ่งกล่าวถึงแล้วในรูปที่ 4.13 - 4.17 นั่นเอง

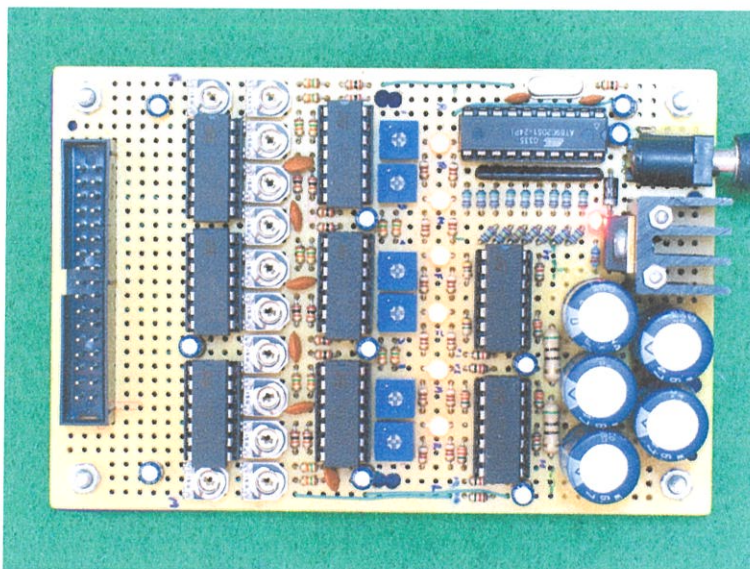
#### 4.4.3 การนำไปควบคุมหุ่นยนต์จริง

ในการควบคุมหุ่นยนต์จริงนั้น จำเป็นจะต้องเปลี่ยนสัญญาณเอาต์พุตของเซลล์ให้อยู่ในรูปของสัญญาณควบคุมเซอร์โวมอเตอร์ก่อน จากนั้นจึงนำไปควบคุมหุ่นยนต์ได้ ซึ่งรายละเอียดการออกแบบวงจรแปลงสัญญาณเอาต์พุตเป็นสัญญาณควบคุมเซอร์โวมอเตอร์จะกล่าวถึงในภาคผนวก ง. ต่อไป



รูปที่ 4.24 ชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์

จากรูปที่ 4.24 แสดงชุดทดสอบการควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ซึ่งประกอบด้วยส่วนหุ่นยนต์ 6 ขา วงจรควบคุมหุ่นยนต์ 6 ขาซึ่งวางอยู่บนตัวหุ่นยนต์ และแหล่งจ่ายไฟกระแสตรงเลี้ยงให้กับหุ่นยนต์



รูปที่ 4.25 วงจรจำลองการทำงานโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์สำหรับหุ่นยนต์ 6 ขา

ส่วนรูปที่ 4.25 เป็นการขยายวงจรควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบ เซลลูลาร์ ซึ่งประกอบด้วยภาคสร้างสัญญาณควบคุมการเคลื่อนที่ ภาคแปลงสัญญาณควบคุมการเคลื่อนที่เป็นสัญญาณเซอร์โวมอเตอร์ และภาคจ่ายไฟเลี้ยงวงจร ภาคสร้างสัญญาณควบคุมการเคลื่อนที่ประกอบด้วย IC ออปแอมป์ 3 ตัวในแถวที่ 2 และ 2 ตัวในแถวที่ 3 นับแถวจากซ้าย IC ดังกล่าวใช้เบอร์ LM324 โดย 3 ตัวในแถวที่ 2 ทำหน้าที่เป็นเซลล์สร้างสัญญาณการก้าวขา ส่วน 2 ตัวในแถวที่ 3 ทำหน้าที่เป็นเซลล์สร้างสัญญาณการยกขา ทางด้านขวาของ IC แถวที่ 2 จะมีตัวต้านทานปรับค่าได้ 6 ตัวทำหน้าที่เป็นตัวควบคุมคาบเวลาการก้าว โดยจะทำงานร่วมกับตัวเก็บประจุที่อยู่ทางซ้ายมือของ IC แถวที่ 2 เป็นวงจร RC Integator นั้นเอง จากนั้นสัญญาณจะถูกส่งไปที่เซลล์ยกขาในแถวที่ 3 แล้วนำสัญญาณทั้งสองไปป้อนให้กับ LED เพื่อแสดงผลการทำงาน ส่วนภาคแปลงสัญญาณควบคุมการเคลื่อนที่ เป็นสัญญาณเซอร์โวมอเตอร์ ประกอบด้วยส่วนสร้างสัญญาณเปรียบเทียบและส่วนวงจรเปรียบเทียบ เพื่อสร้างสัญญาณควบคุมเซอร์โวมอเตอร์ ส่วนสร้างสัญญาณเปรียบเทียบประกอบด้วย ไมโครคอนโทรลเลอร์ 89C2051 ทางด้านขวาบนสร้างสัญญาณเปรียบเทียบมาเป็นข้อมูลดิจิตอลจากนั้นส่งมาให้วงจร R2R Leader วางอยู่ด้านล่าง 89C2051 ซึ่งเป็นวงจรแปลงดิจิตอลเป็นแอนะล็อก แล้วส่งสัญญาณที่ได้ไปเปรียบเทียบกับสัญญาณควบคุมหุ่นยนต์จากภาคสร้างสัญญาณควบคุมการเคลื่อนที่ ที่ส่วนวงจรเปรียบเทียบ ซึ่งก็คือ IC แถวที่ 1 ในวงจร ส่วนตัวต้านทานปรับค่าได้ทางด้านขวาของ IC แถวที่ 1 มีหน้าที่ปรับระดับการยกและการก้าวขาของแต่ละเซลล์ ส่วนภาคจ่ายไฟเลี้ยงประกอบด้วยวงจรจ่ายไฟเลี้ยง 5V โดยใช้ IC 7805 เป็นตัวแปลงไฟ จากนั้นจะนำไปสร้าง Ground เสมือนโดยใช้ตัวเก็บประจุค่า 4700 $\mu$ F จำนวน 4 ตัว และตัวต้านทาน 50 $\Omega$  2 ตัวทางด้านขวาล่างของวงจร เพื่อใช้อ้างอิงในวงจรออปแอมป์ ส่วนวงจรทั้งหมดได้แสดงไว้ในภาคผนวก ง.

#### 4.5 สรุปท้ายบท

ที่กล่าวมาข้างต้น เป็นหลักการในการออกแบบวงจรจำลองการทำงานโครงข่ายประสาทเทียมแบบเซลลูลาร์สำหรับหุ่นยนต์ 6 ขา ซึ่งเป็นการนำหลักการที่มีผู้เสนออยู่แล้ว มาพัฒนาให้ออกแบบได้ง่ายยิ่งขึ้นอีกทั้งสามารถลดตัวต้านทานลงไป 6 ตัวต่อเซลล์ และทดสอบการทำงานว่าเป็นไปดังที่ต้องการหรือไม่ ซึ่งจากผลการทดสอบสรุปว่าเป็นไปตามที่ต้องการ

ในบทต่อไปจะกล่าวถึงการทดสอบเปรียบเทียบการทำงาน รวมทั้งเปรียบเทียบข้อดีข้อเสียของทั้ง 2 วิธีการ

## บทที่ 5

### ผลทดลองการทดสอบการเคลื่อนที่

การทดลองแบ่งเป็น 2 ส่วนคือการศึกษาพฤติกรรมการไต่ขึ้นที่ลาดชันซึ่งเป็นปรากฏการณ์ที่น่าสนใจของการควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์และพฤติกรรมการใช้พลังงาน ซึ่งเป็นความสามารถพิเศษในการประหยัดพลังงานของระบบ โดยจะใช้หุ่นยนต์ 6 ขาที่ควบคุมด้วยขั้นตอนวิธีการเปิดตารางเป็นระบบเปรียบเทียบ

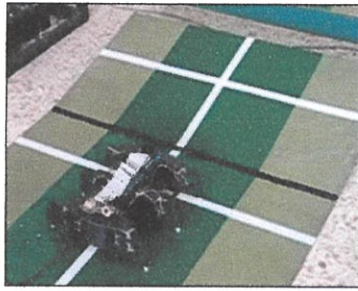
สำหรับการทดสอบไต่ขึ้นที่สูง จะใช้ทางลาดชันประมาณ  $20^\circ$  จากนั้นจะปล่อยหุ่นยนต์ที่ควบคุมการเคลื่อนที่ทั้งสองวิธีเดินไป ส่วนการศึกษาพลังงานที่หุ่นยนต์ที่ 2 แบบทำโดยการต่อวงจรเพื่อวัดค่ากระแสทั้งทางรูปคลื่นของสัญญาณ และข้อมูลทางสถิติ เพื่อหากระแสเฉลี่ยที่หุ่นยนต์ทั้งสองประเภท ซึ่งผลที่ได้จะกล่าวถึงโดยละเอียดต่อไป

#### 5.1 ผลการควบคุมหุ่นยนต์ด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์กับวิธีเปิดตาราง

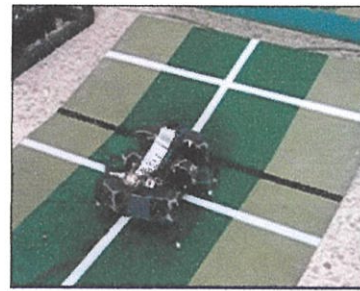
ในการศึกษาพฤติกรรมการไต่ขึ้นที่ลาดชันของหุ่นยนต์ เนื่องจากความสามารถเคลื่อนที่ผ่านที่ลาดชันขึ้นอยู่กับคุณสมบัติทางกายภาพของหุ่นยนต์ เช่นขนาดของหุ่นยนต์ ระยะห่างของแต่ละขา ลักษณะขาหุ่นยนต์ ลักษณะส่วนที่สัมผัสกับพื้นของแต่ละขา ลักษณะพื้นที่เดินผ่าน และความสูงของจุดศูนย์กลางถ่วงเป็นต้น เพื่อหลีกเลี่ยงผลกระทบจากเงื่อนไขเบื้องต้น จึงต้องอาศัยชุดควบคุมการเคลื่อนที่แบบเปิดตารางดังแสดงไว้ในบทที่ 2 และชุดควบคุมการเคลื่อนที่ด้วยวงจรโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ที่แสดงไว้ในบทที่ 4 มาควบคุมโครงของหุ่นยนต์ 6 ขาตัวเดียวกัน ในสภาวะแวดล้อมเดียวกันซึ่งได้ผลมีดังนี้

##### 5.1.1 พฤติกรรมการไต่ขึ้นทางลาดชัน

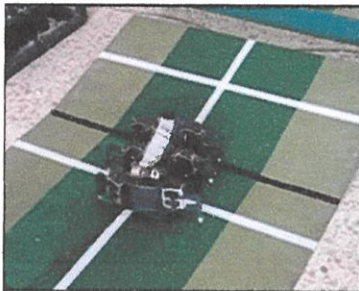
การทดลองกระทำที่มุมลาดชันประมาณ  $20^\circ$  ซึ่งเป็นมุมที่หุ่นยนต์ตัวอย่าง ไม่สามารถเคลื่อนที่ไปได้ วิธีสังเกตผล ทำโดยบันทึกภาพติดต่อกันเป็นจำนวน 6 ภาพ แต่ละภาพบันทึกห่างกัน 3 วินาที รวมเวลาในการบันทึกทั้งสิ้น 18 วินาที นำมาแสดงต่อกัน เพื่อให้ง่ายต่อการสังเกตการเคลื่อนที่ นำมาแสดงการเคลื่อนที่แบบวิธีการเปิดตารางในรูปที่ 5.1(ก) ถึง (จ) และการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ในรูปที่ 5.2 (ก) ถึง (จ) โดยที่การปล่อยหุ่นยนต์ จะปล่อยที่จุดกึ่งกลางบาทสี่ขาแรก จากนั้นจะให้หุ่นยนต์เดินไปข้างหน้าจนกระทั่งถึงจุดที่พื้นลาดชันซึ่งเป็นแนวเส้นสีดำ ผลที่ได้เป็นดังนี้



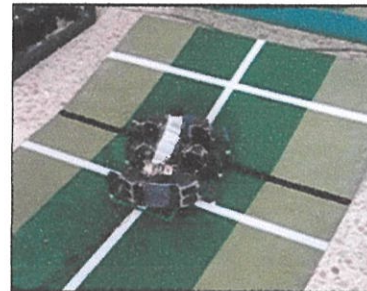
(ก)



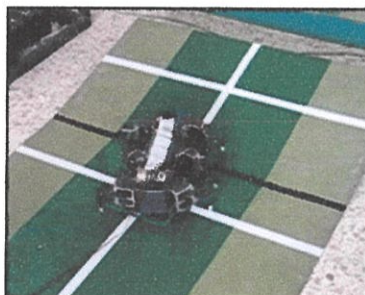
(ข)



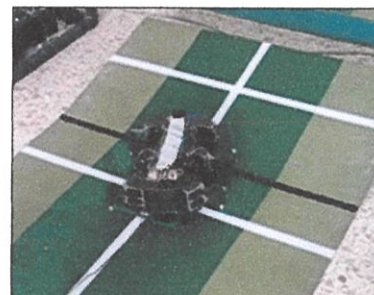
(ค)



(ง)



(จ)

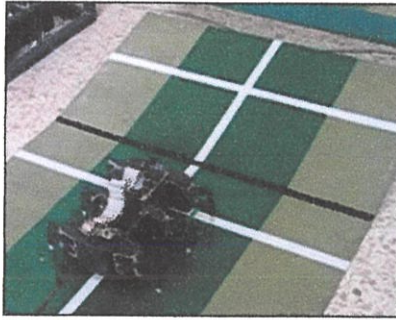


(ฉ)

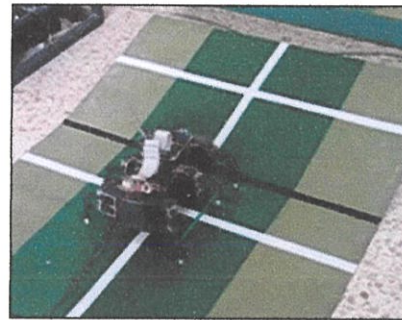
**รูปที่ 5.1 (ก) - (ฉ)** ลำดับขั้นตอนการเคลื่อนที่บนทางลาดชันประมาณ  $20^\circ$  โดยการเคลื่อนที่แบบขั้นตอนวิธีเปิดตาราง

เมื่อหุ่นยนต์เริ่มเคลื่อนที่จากจุดปล่อยในรูปที่ 5.1(ก) หุ่นยนต์จะเคลื่อนที่ตรงไปก่อนจนกระทั่งถึงแนวเส้นดำซึ่งแสดงจุดต่อของทางชันในรูปที่ 5.1(ค) ขาหุ่นยนต์จะเอนไปทางด้านซ้ายเล็กน้อยและเริ่มเกิดการลื่นไถลทั้ง 6 ขา จากนั้นหุ่นยนต์จะไม่สามารถเคลื่อนที่ไปได้อีกเลย หลังจากนั้นได้ทำการทดลองซ้ำด้วยวิธีการเดิมอีกมากกว่า 5 ครั้ง ผลที่ได้ก็ยังคงเป็นเช่นเดิม จึงพอสรุปเบื้องต้นได้ว่าเมื่อหุ่นยนต์เคลื่อนที่ด้วยวิธีการเปิดตารางตามบทที่ 2 โดยอาศัยโครงสร้างของหุ่นยนต์ตัวอย่าง เมื่อเคลื่อนที่มาถึงรอยต่อระหว่างพื้นราบกับทางลาดชัน หุ่นยนต์จะเกิดการลื่นไถลของตัวหุ่นยนต์ทำให้ไม่สามารถเคลื่อนที่ไปด้านหน้าได้

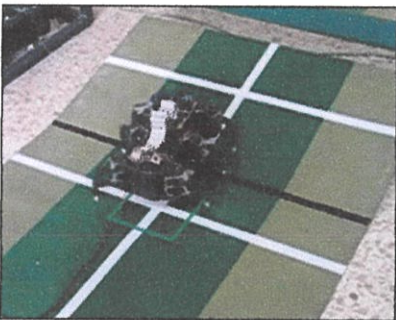
ลำดับต่อไปได้นำหุ่นยนต์ที่ควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ มาทดสอบการควบคุมหุ่นยนต์ตัวอย่างด้วยวิธีเดียวกันผลที่ได้เป็นไปตามรูปที่ 5.2



(ก)



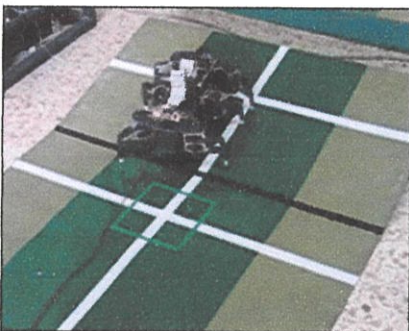
(ข)



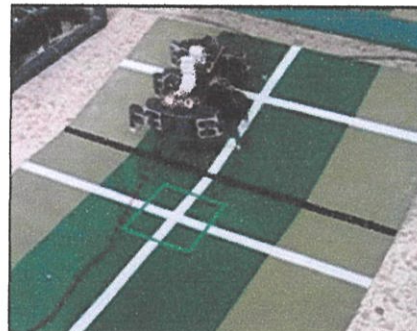
(ค)



(ง)



(จ)



(ฉ)

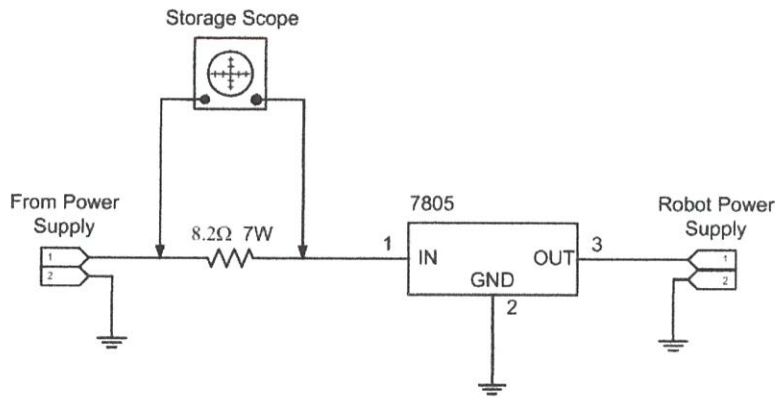
รูปที่ 5.2 (ก) - (ฉ) ลำดับขั้นตอนการเคลื่อนที่บนทางลาดชันประมาณ  $20^\circ$  ด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์

จากรูปที่ 5.2(ก) หุ่นยนต์จะเคลื่อนที่จากจุดปล่อยตรงไปยังแนวเส้นดำซึ่งแสดงจุดต่อของทางชัน เมื่อถึงจุดลาดชันในรูปที่ 5.2(ค) หุ่นยนต์จะเริ่มสั่นไถล โดยการสั่นไถลนี้จะทำให้หุ่นยนต์เอนตัวไปทางซ้ายเล็กน้อย จากนั้นหุ่นยนต์จะเคลื่อนที่ผ่านจุดลาดชันดังกล่าวได้ โดยการเดินจะยังคงลักษณะการเอนทางซ้ายเล็กน้อยอยู่ หลังจากนั้นได้ทำการทดลองซ้ำด้วยวิธีการเดิมอีก 5 ครั้งผลที่ได้ก็ยังเป็นเช่นเดิม จึงพอสรุปเบื้องต้นได้ว่าเมื่อหุ่นยนต์เคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ตามบทที่ 4 โดยอาศัยโครงร่างของหุ่นยนต์ตัวอย่าง ก็จะสามารถเคลื่อนที่ผ่านรอยต่อระหว่างพื้นราบกับทางลาดชันได้ โดยหุ่นยนต์จะเลี้ยวเบนไปทางซ้ายเล็กน้อย

อย่างไรก็ตามในการทดลองนั้นมียู 1 ครั้งที่หุ่นยนต์ที่ควบคุมการเคลื่อนที่แบบขั้นตอนวิธีเปิดตารางสามารถเลี้ยวเบนได้เช่นเดียวกับการควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ทั้งนี้เนื่องจาก มุมที่หุ่นยนต์กระทำกับทางลาดไม่ตั้งฉากจึงไม่เกิดปัญหาดังกล่าว

### 5.1.2 การใช้พลังงานในการเคลื่อนที่

จากแนวความคิดว่า ลักษณะและรูปแบบการขับเคลื่อนเซอร์โวมอเตอร์อาจมีส่วนกระทบกับปริมาณกระแสไฟเลี้ยงของระบบ ซึ่งเป็นที่มาของการศึกษาคุณลักษณะการใช้พลังงานในการขับเคลื่อน โดยมีสมมุติฐานว่าการควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์นั้น หุ่นยนต์จะทำการเคลื่อนที่แต่ละขาไม่พร้อมกัน ซึ่งควรจะทำให้การใช้กระแสไฟเลี้ยง กระจายตัวกันมากขึ้น ทำให้กระแสชั่วขณะมีค่าต่ำลง อีกทั้งระยะการเปลี่ยนแปลงมุมของเซอร์โวมอเตอร์ก็ควรจะมีผลต่อการใช้พลังงานของระบบด้วยจึงทำการทดลองดังกล่าวด้วย



รูปที่ 5.3 วงจรตรวจวัดกระแสด้วยสโคป

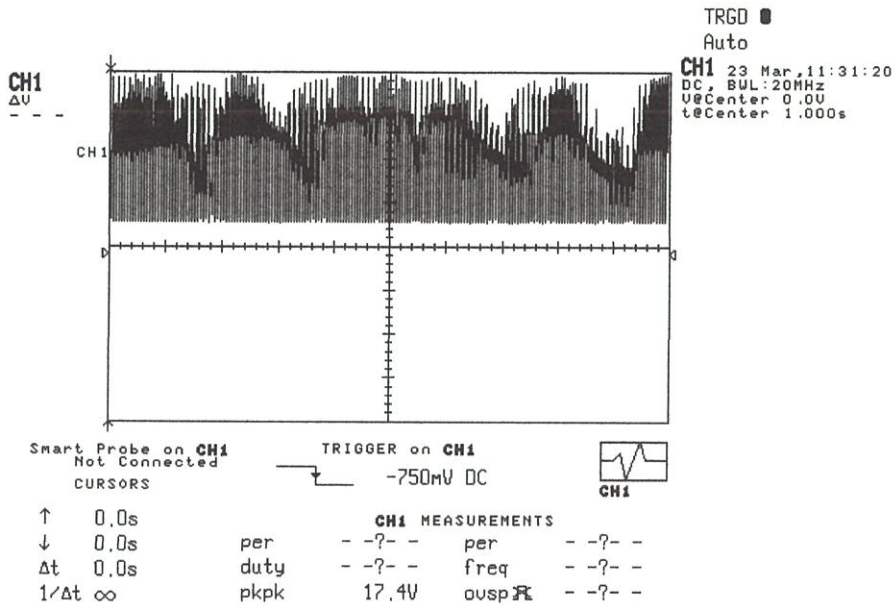
รูปที่ 5.3 เป็นการแสดงการวัดกระแสทางอ้อม โดยนำความต้านทานไปอนุกรมกับไฟเลี้ยงหุ่นยนต์ ดังนั้นกระแสที่หุ่นยนต์ต้องการจะเท่ากับค่าแรงดันที่ตัวต้านทานดังกล่าว โดยวัดจากสโคปหารด้วยค่าความต้านทานนั้น ค่าความต้านทานนี้จะมีผลต่อขนาดของสัญญาณที่สโคปจะวัดได้ ซึ่งค่าที่เหมาะสมคือ  $8.2\Omega$  เนื่องจากวงจรค่าแรงดันจากแหล่งจ่ายไฟกระแสตรง ต้องมีค่าสูงกว่า 5V แต่ไม่ควรเกิน 30V ส่วน LM340 ทำหน้าที่รักษาแรงดันที่ป้อนให้หุ่นยนต์เป็น 5V เสมอ ไม่ว่าหุ่นยนต์จะใช้กระแสไฟเท่าใดก็ตาม จากกฎของโอห์มสามารถเขียนสมการได้ดังนี้

$$I_{load} = \frac{V_{osc}}{8.2\Omega} \quad (5.1)$$

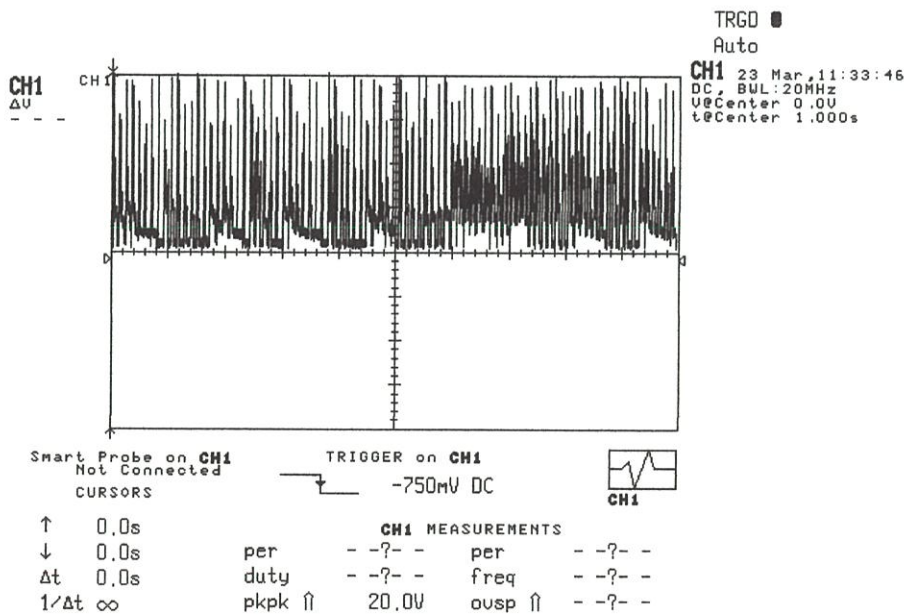
โดย

$I_{load}$  คือกระแสที่หุ่นยนต์ต้องการ และ  $V_{osc}$  คือแรงดันที่วัดได้จากสโคป

ในการตรวจสอบ จะทำการปรับค่าการเคลื่อนที่ของเซอร์โวมอเตอร์ไว้ 2 ระดับคือ การหมุนแต่ละครั้งเคลื่อนที่ 30° และ 60° ตามลำดับ ที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์ 30° มีผลดังนี้



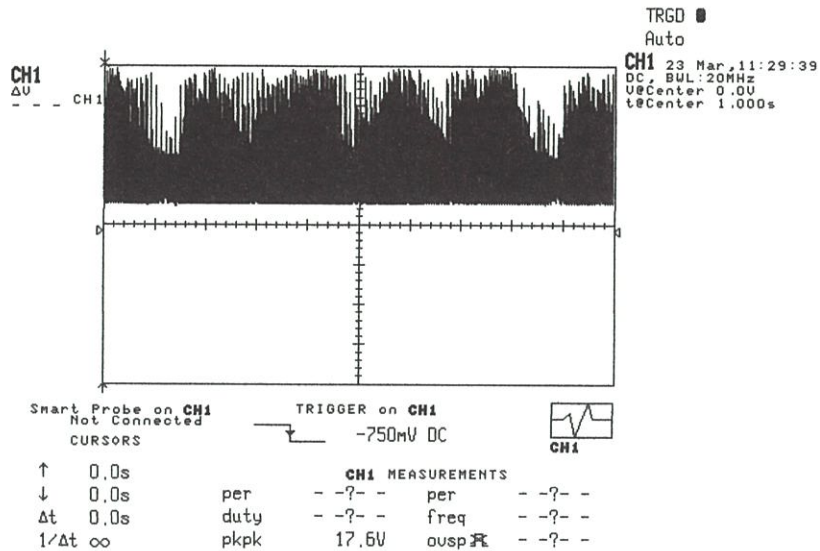
รูปที่ 5.4 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยขั้นตอนวิธีการเปิดตารางที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์ 30°



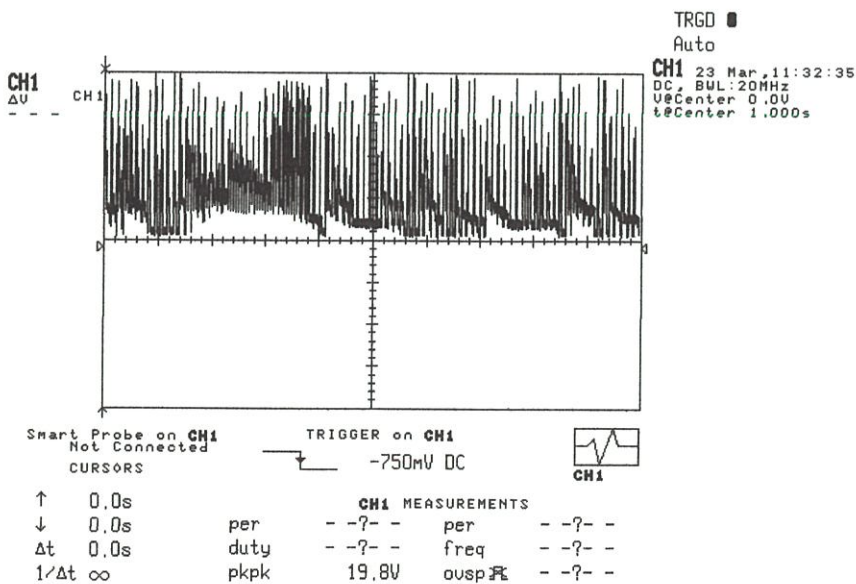
รูปที่ 5.5 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลล์ลาร์ที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์ 30°

จากรูปใช้ค่าแรงดันต่อช่องที่ 5V เมื่อแปลงเป็นกระแสแล้วจะมีค่ากระแสต่อช่องอยู่ที่ 0.6A หรือเมื่อมองเป็นวัตต์ และเวลาต่อช่องที่ 0.1 วินาที จะเห็นว่าค่าการใช้กระแสของหุ่นยนต์ที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์มีค่ากระแสเฉลี่ยที่ต่ำกว่า

เพื่อศึกษาผลกระทบทับมุมที่บังคับให้เซอร์โวมอเตอร์หมุนไป กับกระแสไฟเลี้ยงวงจร จึงทำการวัดด้วยวิธีเดียวกันกับข้างต้น แต่เปลี่ยนมุมการเคลื่อนที่ของเซอร์โวมอเตอร์ จาก  $30^\circ$  เป็น  $60^\circ$  มีผลดังนี้



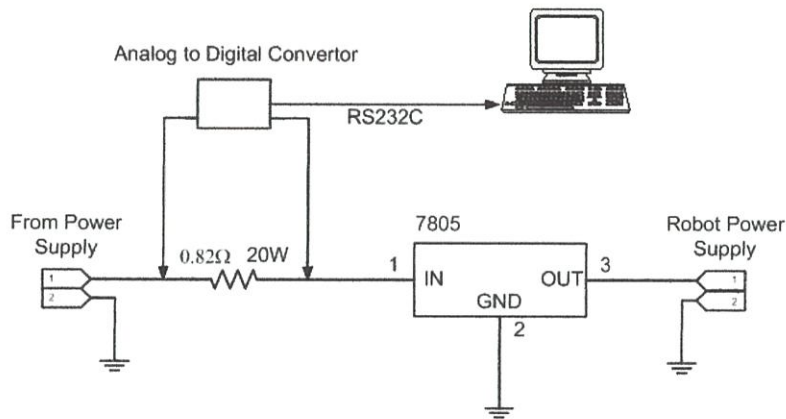
รูปที่ 5.6 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยขั้นตอนวิธีการเปิดตารางที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์  $60^\circ$



รูปที่ 5.7 การใช้กระแสไฟเลี้ยงของหุ่นยนต์ 6 ขาที่ควบคุมด้วยโครงข่ายประสาทเทียมแบบเซลล์ูลาร์ที่มุมการเคลื่อนที่ของเซอร์โวมอเตอร์  $60^\circ$

จากรูปที่ 5.4 รูปที่ 5.5 เมื่อเปรียบเทียบกับ รูปที่ 5.6 รูปที่ 5.7 จะเห็นว่า เมื่อมีมุมการเคลื่อนที่มากขึ้นอัตราการใช้กระแสก็จะมากขึ้นตามไปด้วย แต่ค่าการใช้กระแสของการควบคุมการเคลื่อนที่ด้วยวิธีเปิดตาราง ยังมีมากกว่าการควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์

นอกจากนั้น ยังได้ทำการเก็บข้อมูลทางสถิติ โดยอาศัยหลักการคล้ายกับวงจรในรูปที่ 5.3 แต่เปลี่ยนอุปกรณ์ที่ใช้ในการวัดจากสโคปเป็นวงจรแปลงแอนะล็อกเป็นดิจิตอลขนาด 8 บิต ดังนั้นค่าความต้านทานที่นำมาอนุกรมกับไฟเลี้ยงหุ่นยนต์จะต้องเปลี่ยนแปลงให้เหมาะสม เพื่อให้ค่าแรงดันตกคร่อมตัวต้านทานดังกล่าวต่ำกว่า 5V ซึ่งเป็นแรงดันสูงสุดที่วงจรแปลงแอนะล็อกเป็นดิจิตอลนี้จะรับได้ ค่าดังกล่าวคือ  $0.82\Omega$  ดังนั้นจะได้วงจรดัง ด้านล่างนี้



รูปที่ 5.8 วงจรเก็บข้อมูลกระแสทางสถิติ

ดังนั้น จะหาค่าแรงดันที่วงจรแปลงแอนะล็อกเป็นดิจิตอลวัดได้ จากสมการที่ (5.2)

$$V_{ADC} = \frac{X_i \times 5V}{256} \quad (5.2)$$

โดย

$V_{ADC}$  คือ แรงดันที่วงจรแปลงแอนะล็อกเป็นดิจิตอลวัดได้จากการเก็บข้อมูลครั้งที่  $i$

$X_i$  คือค่าข้อมูลที่เก็บได้

จากกฎของโอห์มเมื่อหุ่นใช้ไฟเลี้ยง 5V สามารถเขียนสมการกำลังงานที่หุ่นยนต์ต้องใช้ได้สมการที่ (5.3)

$$P_i = \frac{X_i \times 25}{256 \times 0.82\Omega} \quad (5.3)$$

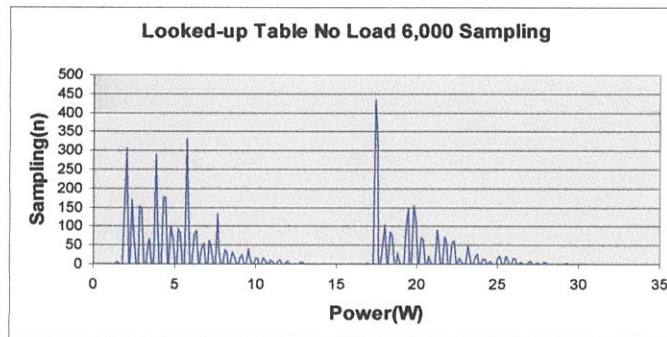
เมื่อ

$P_i$  คือกำลังงานที่วัดได้จากการเก็บข้อมูลครั้งที่  $i$

และกำลังงานเฉลี่ยของหุ่นยนต์หาได้จากสมการที่ (5.4)

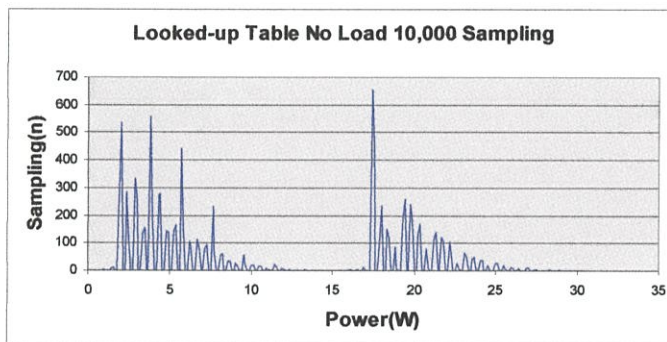
$$\bar{P} = \frac{\sum_{i=1}^m n_i P_i}{\sum_{i=1}^m n_i} \quad (5.4)$$

การเก็บข้อมูลจะกระทำทุก ๆ 10 มิลลิวินาที โดยวัดผลใน 2 กรณีคือ กรณียกขาลอย ให้หุ่นยนต์อยู่กับที่ และ กรณีนำหุ่นยนต์ไปเดินบนพื้นจริง แต่ละกรณีทำการวัด 3 ครั้ง แต่ละครั้ง กำหนดจำนวนการเก็บข้อมูลเป็น 6,000 ครั้ง 10,000 ครั้ง และ 60,000 ครั้ง ตามลำดับ ผลที่ได้มีดังนี้



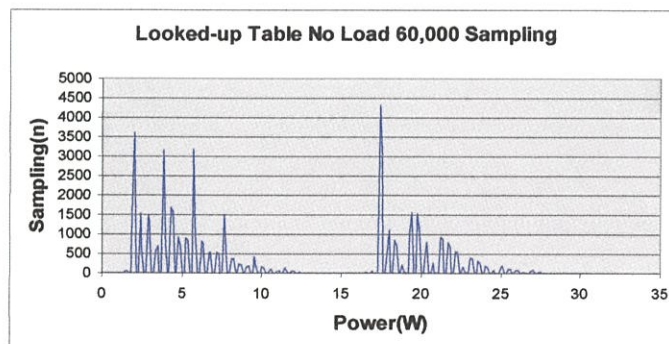
$$\bar{P} = 11.088W$$

(ก)



$$\bar{P} = 10.876W$$

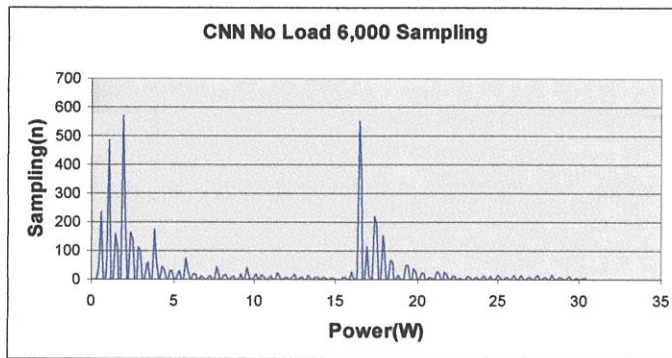
(ข)



$$\bar{P} = 11.002W$$

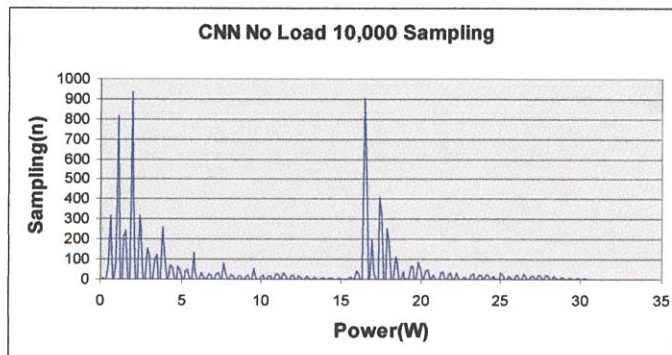
(ค)

รูปที่ 5.9 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย  $\bar{P}$  ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วยวิธีเปิดตารางกรณียกขาลอย โดยเก็บข้อมูลที (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง



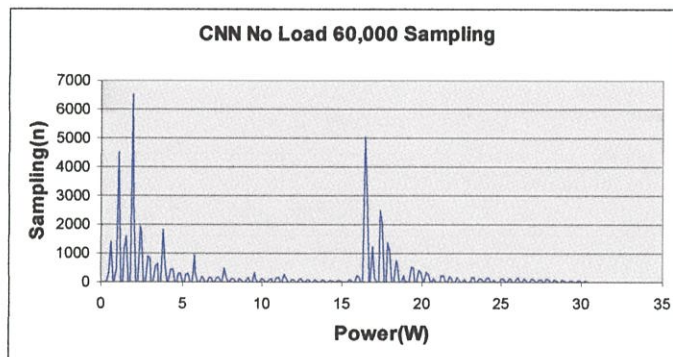
$$\bar{P} = 9.576W$$

(ก)



$$\bar{P} = 9.646W$$

(ข)



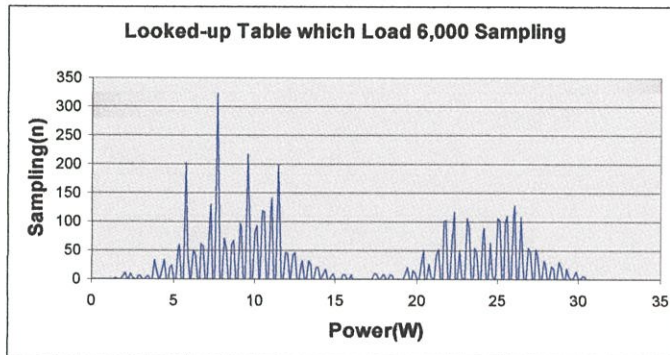
$$\bar{P} = 9.511W$$

(ค)

รูปที่ 5.10 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย  $\bar{P}$  ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วย CNN กรณียกขาลอย โดยเก็บข้อมูลที่ (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง

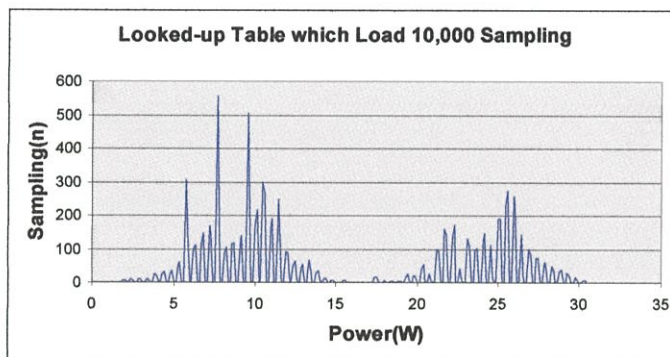
จากจากรูปที่ 5.9 และรูปที่ 5.10 จะเห็นว่าในกรณีที่ยกขาลอย ซึ่งตัดภาระจากน้ำหนักตัว หุ่นยนต์ให้กับเซอร์โวมอเตอร์ คงเหลือเพียงภาระจากความฝืดในกลไก และน้ำหนักของขาหุ่นยนต์ เท่านั้น ซึ่งมีค่าเท่ากันทั้งการควบคุมด้วยวิธีเปิดตาราง และการควบคุมด้วย CNN จากสถิติ จะเห็นว่ากำลังงานเฉลี่ยของหุ่นยนต์ที่ควบคุมด้วย CNN จะมีค่าเฉลี่ยอยู่ที่ 9.576W ในกรณี 6,000 ครั้ง อยู่ที่ 9.646W ในกรณี 10,000 ครั้ง และอยู่ที่ 9.511W ในกรณี 60,000 ครั้ง ซึ่งน้อยกว่า หุ่นยนต์ที่ควบคุมด้วยวิธีการเปิดตาราง ที่มีค่าเฉลี่ยอยู่ที่ 11.088W ในกรณี 6,000 ครั้ง อยู่ที่ 10.876W ในกรณี 10,000 ครั้ง และอยู่ที่ 11.002W ในกรณี 60,000 ครั้ง

เมื่อพิจารณาจากรูปจะเห็นว่า กราฟทางสถิติจะมีรูปแบบคล้ายกัน แต่กราฟของการควบคุมแบบเปิดตาราง จะมีค่าจำนวนครั้งที่ช่วงกำลังงานที่ 0-5W และที่ 15-20W สูงกว่าการควบคุมแบบ CNN จึงแสดงให้เห็นว่า มีการทำงานบางช่วงของหุ่นยนต์ที่ควบคุมด้วยการเปิดตารางที่ใช้กระแสไฟฟ้ามากกว่าเกิดขึ้น อันน่าจะเกิดจากวิธีการสั่งงานให้หุ่นยนต์เคลื่อนที่ด้วยวิธีที่แตกต่างกัน จากนั้นได้ทำการเก็บข้อมูลทางสถิติโดยให้หุ่นยนต์ที่ควบคุมด้วยวิธีทั้งสองแบบเดินบนพื้น และใช้วิธีการเก็บข้อมูลแบบเดิม ได้ผลดังนี้



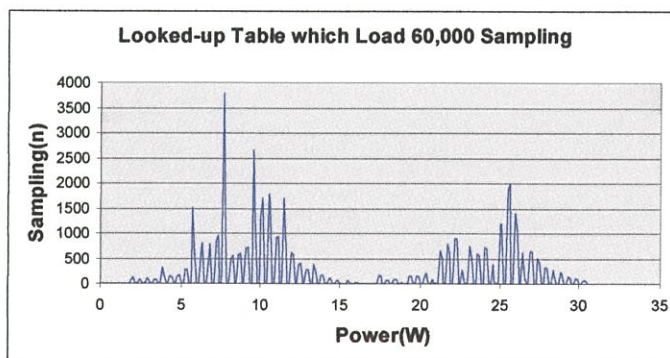
$$\bar{P} = 15.225W$$

(ก)



$$\bar{P} = 15.365W$$

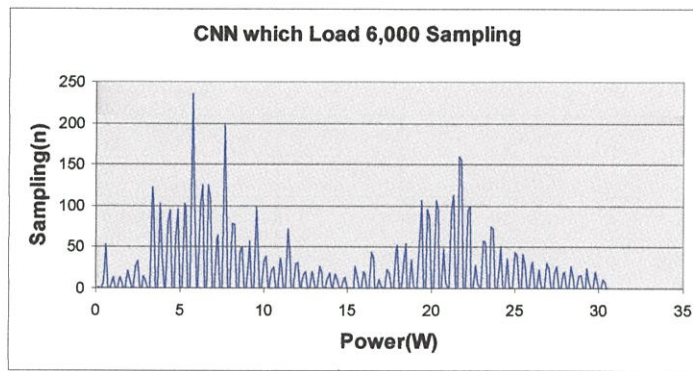
(ข)



$$\bar{P} = 15.278W$$

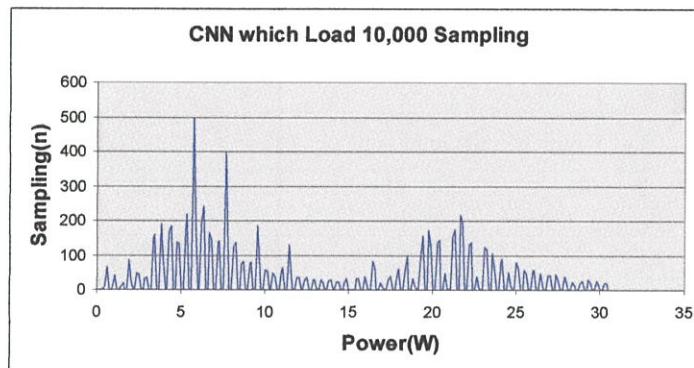
(ค)

รูปที่ 5.11 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย  $\bar{P}$  ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วยวิธีเปิดตารางกรณีเดินที่พื้น โดยเก็บข้อมูลที่ (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง



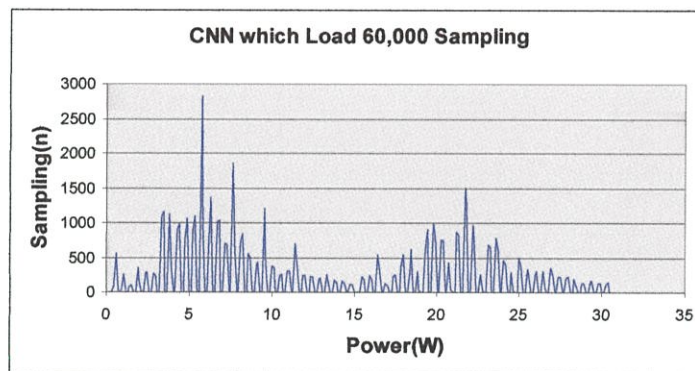
$$\bar{P} = 13.630W$$

(ก)



$$\bar{P} = 12.999W$$

(ข)



$$\bar{P} = 13.197W$$

(ค)

รูปที่ 5.12 ค่าสถิติของกำลังงานและกำลังงานเฉลี่ย  $\bar{P}$  ที่วัดได้ของหุ่นยนต์ที่ควบคุมด้วย CNN กรณีเดินที่พื้น โดยเก็บข้อมูลที่ (ก) 6,000 ครั้ง (ข) 10,000 ครั้ง และ (ค) 60,000 ครั้ง

ในลักษณะเดียวกัน ค่ากำลังงานเฉลี่ยที่หาได้จากการเก็บข้อมูลทางสถิติในกรณีเดินที่พื้น จะเห็นว่ากระแสเฉลี่ยของหุ่นยนต์ที่ควบคุมด้วย CNN จะมีกำลังงานค่าเฉลี่ยอยู่ที่ 13.630W ในกรณี 6,000 ครั้ง อยู่ที่ 12.999W ในกรณี 10,000 ครั้ง และอยู่ที่ 13.197W ในกรณี 60,000 ครั้ง ซึ่งน้อยกว่า หุ่นยนต์ที่ควบคุมด้วยวิธีการเปิดตาราง ที่มีค่ากำลังงานเฉลี่ยอยู่ที่ 15.225W ในกรณี 6,000 ครั้ง อยู่ที่ 15.365W ในกรณี 10,000 ครั้ง และอยู่ที่ 15.278W ในกรณี 60,000 ครั้ง

และเมื่อพิจารณาจากกราฟสถิติ จะเห็นว่า การควบคุมโดยอาศัยการเปิดตารางจะมีจำนวนครั้งที่ได้ข้อมูลอยู่ในช่วง 5-15W และช่วง 20-30W มากกว่า การควบคุมด้วย CNN ทั้งนี้จำนวนครั้งของข้อมูลในการควบคุมด้วย CNN จะไปเพิ่มที่ช่วง 3-12W และ 16-28W แทน ซึ่งมีค่าต่ำกว่า จึงทำให้กำลังงานเฉลี่ยของการควบคุมด้วย CNN มีค่าต่ำกว่านั่นเอง

สำหรับสาเหตุที่อาจทำให้การควบคุมการเคลื่อนที่ด้วย CNN ใช้กำลังงานไฟเลี้ยงต่ำกว่าก็คือ การควบคุมแบบนี้ ค่ามุมของเกาท์พุตจะค่อย ๆ เปลี่ยนแปลงไปที่ละน้อย ทำให้ไม่ต้องใช้พลังงานในการขับเคลื่อนแต่ละครั้งมาก และการควบคุมการเคลื่อนที่ด้วยขั้นตอนวิธีการเปิดตารางนั้น จะมีการเปลี่ยนแปลงจากค่าสูงสุดไปต่ำสุดในทันทีทันใด ทำให้ต้องจ่ายกระแสให้มอเตอร์ต่อเนื่องเป็นเวลานานกว่าด้วย

## 5.2 สรุปท้ายบท

จากการทดสอบในส่วนแรกคือการไต่ขึ้นทางชันนั้น จะเห็นว่าในหุ่นยนต์ที่ควบคุมการเคลื่อนที่ด้วยขั้นตอนวิธีการเปิดตาราง จะประสบปัญหาเมื่อหุ่นยนต์เดินเข้าหาทำมุมฉากกับระนาบชันที่สูงกว่าความสามารถในการไต่ได้ของหุ่นยนต์ โดยจะเกิดอาการสั่นไถลขึ้น ส่วนหุ่นยนต์ที่เคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์นั้น จะมีปรากฏการณ์เลี้ยวเบนไปด้านใดด้านหนึ่งจนกระทั่งมุมที่กระทำกับแนวระนาบต่ำลงจนสามารถไต่ขึ้น ซึ่งการเดินนี้จะเป็นทิศทางเดินเฉียงออกไปด้านข้าง ดังนั้นถ้าต้องการนำคุณสมบัตินี้มาใช้ จำเป็นจะต้องหาวิธีที่จะกลับทิศทางเดินเฉียงให้ตรงข้ามกันเป็นช่วง ๆ เพื่อให้สามารถไปถึงจุดหมายได้

ในตอนที่สองได้กล่าวถึงการทดสอบและเปรียบเทียบ การใช้กำลังงานของหุ่นยนต์ที่ใช้การควบคุมด้วยขั้นตอนวิธีการเปิดตาราง กับกำลังงานของหุ่นยนต์ที่ควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ โดยกระทำที่การควบคุมให้เซอร์โวมอเตอร์มีการเปลี่ยนแปลงที่  $30^\circ$  และ  $60^\circ$  ผลที่ได้คือ หุ่นยนต์ที่ใช้การควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ ใช้กระแสชั่วขณะต่ำกว่าหุ่นยนต์ที่ใช้การควบคุมด้วยขั้นตอนวิธีการเปิดตาราง ทั้งขนาดมุมเซอร์โวมอเตอร์ที่  $30^\circ$  และ  $60^\circ$  และที่มุมการเปลี่ยนแปลงของเซอร์โวมอเตอร์ที่  $30^\circ$  จะใช้กระแสชั่วขณะต่ำกว่าที่มุม  $60^\circ$  ด้วย

## บทที่ 6

### สรุปผลการวิจัยและข้อเสนอแนะ

วิทยานิพนธ์นี้นำเสนอเทคนิคใหม่ในการควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขา ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์และศึกษาค่าพารามิเตอร์ต่างๆของระบบ ในส่วนแรกกล่าวถึงการควบคุมการเคลื่อนที่แบบขั้นตอนวิธีการเปิดตาราง อีกทั้งได้แสดงการออกแบบชุดควบคุมด้วยวิธีดังกล่าวเพื่อใช้เป็นตัวเปรียบเทียบกับระบบใหม่ที่นำเสนอ

จากนั้นได้กล่าวถึงทฤษฎีของโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ และการประยุกต์ใช้ในการควบคุมที่ได้มีผู้ศึกษาพัฒนาก่อนหน้านี้แล้ว ซึ่งระบบดังกล่าวยังมีจุดด้อยอยู่ที่จำนวนเซลล์ที่ใช้ในการสร้างคลื่นเพื่อนำไปควบคุมการเคลื่อนที่ซึ่งต้องใช้ถึงชุดเซลล์ถึง 12 เซลล์ เนื่องจากการกระจายของคลื่นในสมการสถานะของระบบเดิมมีไม่มากนัก ดังนั้นจึงนำเสนอระบบสมการสถานะใหม่ซึ่งมีการกระจายของสัญญาณที่สูงกว่า ทำให้สามารถทำงานได้โดยอาศัยชุดเซลล์เพียงแค่ 6 เซลล์เท่านั้น แล้วนำเสนอสมการที่ได้ไปทดสอบผลกระทบต่อค่าพารามิเตอร์ต่าง ๆ

สำหรับค่าพารามิเตอร์ของชุดเซลล์ที่ใช้สร้างคลื่น ซึ่งชุดเซลล์ดังกล่าวเป็นชุดเดียวกันกับที่ใช้ในการควบคุมการก้าวเดินของหุ่นยนต์ 6 ขาที่ออกแบบโดยพารามิเตอร์ที่มีผลกับสมการคือค่า  $\mu$  และค่า  $I$  จากการทดสอบพบว่าค่า  $\mu$  มีผลโดยตรงต่อการทำให้เกิดคลื่น ทั้งนี้ถ้าค่า  $\mu$  มีค่าน้อยกว่า 0.45 จะไม่มีคลื่นเกิดขึ้นเลย ถ้าค่า  $\mu$  มีค่ามากกว่า 0.45 แต่ไม่เกิน 0.65 จะมีคลื่นเกิดขึ้น แต่ขนาดของคลื่นจะไม่ถึงจุดอิ่มตัว จนกระทั่งค่า  $\mu$  มีค่ามากกว่า 0.65 ค่าเอทพุตจะถึงจุดอิ่มตัวทันที และถ้ายิ่งเพิ่มค่า  $\mu$  ให้มากขึ้นไปอีก จะเห็นว่า ความชันขาขึ้นและขาลงของสัญญาณจะมากขึ้นด้วย ส่วนค่า  $I$  นั้น จะมีผลกับระดับของสัญญาณกล่าวคือ ที่ค่า  $I$  มีค่าเป็น 0 สัญญาณจะมีระดับอยู่ตรงกลาง เมื่อมีค่า  $I$  เปลี่ยนแปลงเพิ่มขึ้นหรือลดลง ค่าระดับเอทพุตก็จะเปลี่ยนแปลงไปด้วย แต่ถ้าเพิ่มค่า  $I$  ไปมากกว่า  $\pm 1$  จะไม่เกิดคลื่นขึ้นที่เอทพุต แต่จะเป็นกระแสไฟตรงออกมาแทน ผลกระทบอีกสิ่งหนึ่งของค่า  $I$  คือ เมื่อค่า  $I$  เข้าใกล้  $\pm 1$  ค่าคาบเวลาของคลื่นจะเปลี่ยนแปลงไปในทางลดต่ำลงด้วย

ในการพิจารณาค่าพารามิเตอร์ในส่วนของชุดเซลล์ยกขานั้น พบว่าชุดเซลล์ส่วนนี้ไม่ได้สร้างคลื่นขึ้นมา เพียงแต่เป็นเซลล์ที่รับสัญญาณคลื่นมาจากชุดเซลล์ก้าวขา โดยพารามิเตอร์ของเซลล์ชุดนี้มีค่า  $\beta$  และ  $I_m$  ซึ่งค่า  $\beta$  มีผลต่อความเร็วในการเคลื่อนที่ขึ้นลงของขา ส่วนค่า  $I_m$  มีผลต่อระดับค่าสูงสุดต่ำสุดที่ขาจะยกได้ จากนั้นได้แสดงการจับคู่เซลล์ทั้งสองชุดที่ใช้ควบคุมแต่ละขา แล้วนำผลที่ได้ไปทดสอบควบคุมหุ่นยนต์จริงโดยการจำลองในเครื่องคอมพิวเตอร์ แล้วส่งผลที่ได้ไปควบคุมหุ่นยนต์อีกครั้งหนึ่ง เมื่อทดสอบความถูกต้องแล้วจึงนำไปออกแบบวงจรแอนะล็อกต่อไป

ในการออกแบบวงจรแอนะล็อก ได้กล่าวถึงวิธีการออกแบบที่มีการศึกษาไว้แล้ว และได้แสดงให้เห็นว่า วิธีการดังกล่าวยังมีโอกาสผิดพลาดจากการออกแบบอยู่ จึงนำเสนอวิธีการออกแบบวงจรแอนะล็อกจากสมการใหม่เพื่อลดข้อผิดพลาดอันอาจเกิดขึ้นในการออกแบบ ในลำดับต่อมา ได้ทดลองวัดสัญญาณจากวงจรที่ออกแบบว่าเป็นไปตามที่ต้องการหรือไม่ ซึ่งผลเป็นไปตามที่คำนวณได้ แล้วจึงนำมาทดสอบเทียบกับชุดควบคุมการเคลื่อนที่แบบขั้นตอนวิธีการเปิดตารางที่ได้ออกแบบไปแล้วในข้างต้น

การทดสอบเปรียบเทียบกระทำใน 2 ส่วนคือ ส่วนที่ 1 การทดสอบการไต่ขึ้นทางลาดชัน โดยกำหนดให้ไต่ขึ้นทางลาดชันทำมุม  $20^\circ$  จากแนวระนาบ ผลปรากฏว่า หุ่นยนต์ที่ควบคุมการเคลื่อนที่ด้วยขั้นตอนวิธีการเปิดตารางจะเกิดการลื่นไถลไม่สามารถไต่ขึ้นไปได้ ส่วนหุ่นยนต์ที่ควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์จะเอนปรับตัวลงมาทางด้านข้าง ทำให้สามารถไต่ขึ้นต่อไปได้

ในส่วนที่ 2 คือการวัดค่ากระแสเลี้ยงเซอร์โวมอเตอร์ โดยวัดที่เซอร์โวมอเตอร์มุม  $30^\circ$  และ  $60^\circ$  ผลปรากฏว่า กระแสชั่วขณะของหุ่นยนต์ที่ควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ มีค่าต่ำกว่าของหุ่นยนต์ที่ควบคุมการเคลื่อนที่ด้วยขั้นตอนวิธีการเปิดตาราง ทั้งนี้ เหตุผลมาจากการเคลื่อนที่ของมอเตอร์เป็นไปโดยไม่พร้อมกัน อีกทั้งยังเคลื่อนที่เป็นมุมแคบ ๆ ไปทีละน้อยจึงมีโอกาสที่จะจ่ายกำลังมอเตอร์หลาย ๆ ตัวพร้อมกันจึงต่ำกว่า

แม้ว่าการควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ จะมีข้อดีกว่าการควบคุมการเคลื่อนที่ด้วยขั้นตอนวิธีการเปิดตาราง ในด้านที่กล่าวมาแล้วก็ตาม แต่ปัญหาของการควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ คือความไม่ยืดหยุ่นในการออกแบบ จึงเหมาะกับอุปกรณ์ที่ทำหน้าที่ใดหน้าที่หนึ่งโดยเฉพาะ สำหรับแนวทางวิจัยต่อไปนั้นสามารถศึกษาถึงวิธีในการบังคับหุ่นยนต์ให้ทำงานตามต้องการเช่น เดินตรง เลี้ยวซ้ายหรือขวา ทั้งนี้ควรเป็นการเปลี่ยนแปลงรูปแบบการกำเนิดคลื่นจากชุดเซลล์กำเนิด คือแทนที่จะมีชุดเซลล์แต่ละหน้าที่ยังคงนำมาสลับหน้าที่กันได้อีกด้วย

## เอกสารอ้างอิง

- [1] John Chad and Howard Wheal. **Cellular Neurobiology A Practical Approach**.  
Oxford: Information Press Ltd. 1991.
- [2] ชาติรี สุดทรวง. **ประสาทสรีรวิทยา**. จุฬาลงกรณ์มหาวิทยาลัย. พ.ศ.2539
- [3] Leon O. Chua and Lin Yang, "Cellular Neural Networks: Theory, " IEEE Trans. Circuits and Systems., vol.35, Oct 1988, pp 1257-1290.
- [4] Leon O. Chua and Lin Yang, "Cellular Neural Networks: Applications, " IEEE Trans. Circuits and Systems., vol.35, Oct 1988, pp 1257-1290.
- [5] Gabriele Manganro, Paolo Arena, Luigi Fortuna, **Cellular Neural Networks Chaos Complexity and VLSI Processing**, New York: Springer-Verlag, 1999.
- [6] Leon O. Chua, Mertin Hasler, George S. Moschytz and Jacques Neiryneck, "Autonomous Cellular Neural Networks: A Unified Paradigm for Pattern Formation and Active Wave Propagation, " IEEE Trans. Circuits and Systems., vol.42, Oct 1995, pp 559-577.
- [7] Fan Zou and Josef A. Nossek, "A Chaotic Attractor with Cellular Neural Networks, "IEEE Trans. Circuits and Systems., vol.38 no.7, Jul 1991, pp 811-812.
- [8] Fan Zou and Josef A. Nossek, "Bifurcation and Chaos in Cellular Neural Networks."IEEE Trans. Circuits and Systems., vol.40, Mar 1993, pp 166-173.
- [9] Fan Zou and Josef A. Nossek, "Stability of Cellular Neural Networks with Opposite-Sign Templates."IEEE Trans. Circuits and Systems., vol.38, Jun 1991, pp675-677 .
- [10] Paolo Arena and Luigi Fortuna, "Analog cellular locomotion control of hexapod robots." Control Systems Magazine, IEEE, vol. 22, Dec. 2002 , pp 21-36 .
- [11] Thomas M. Frederiksen, **Intuitive Operational Amplifiers**, McGraw-Hill, 1988.
- [12] Paolo Arena, Salvatore Castorina, Luigi Fortuna, Mattia Frasca and MarcoRuta, "A CNN-based chip for robot locomotion control." Circuits and Systems, ISCAS '03. Proceedings of the 2003 International Symposium, vol.3, 25-28 May 2003, pp III-510 - III-513 .

- [13] Csaba Rekeczky, Dávid Bálya, Gergely Timár and István Szatmári, "Bio-inspired flight control and visual search with CNN technology" *Circuits and Systems, ISCAS '03 Proceedings of the 2003 International Symposium*, vol.3, 25-28 May 2003, pp III-774 - III-777.
- [14] Nicola Giaquinto, Mario Savino and Sergio Taraglio, "A CNN-based passive optical range finder for real-time robotic applications.", *IEEE Trans. Instrumentation and Measurement*, 2 April 2002, pp 314 – 319.
- [15] Radu Dogaru, Ionescu T. and Manfred Glesner, "A design for emergence method applied to recurrent cellular computing systems with multi-nested cells.", 2003. *CAS 2003 International Semiconductor Conference* , Vol. 2, 28 Sept.-2 Oct. 2003, pp 386.
- [16] Paolo Arena, Adriano Basile, Luigi Fortuna, Mattia Frasca and Luca Patané, "Implementation of Turing patterns for bio-inspired motion control" *ISCAS '03. Proceedings of the 2003 International Symposium on Circuits and Systems*, Vol. 3, 25-28 May 2003, pp III-842 - III-845.
- [17] Paolo Arena, Claudia Bonomo, Luigi Fortuna and Mattia Frasca, "Electro-active polymers as CNN actuators for locomotion control" *ISCAS 2002. IEEE International Symposium on Circuits and Systems*, Vol. 4, 26-29 May 2002, pp IV-281 - IV-284.
- [18] Alexandru Gacsádi, Teodor Maghiar. and Virgil Tiponut, "A CNN path planning for a mobile robot in an environment with obstacles", *Proceedings of the 2002 7th IEEE International Workshop on Cellular Neural Networks and Their Applications*, 22-24 July 2002, pp 188 – 194.
- [19] David Wettergreen and Chuck Thorpe, "Developing planning and reactive control for a hexapod robot", *Proceedings of the 1996 IEEE International Conference on Robotics and Automation* Vol. 3, 22-28 April 1996, pp 2718-2723.
- [20] George B. Rutkowski, **Integrated – Circuit Operational Amplifier :second edition**, Prantice-Hall Inc., Englewood Cliffs, New Jersey, 1984.
- [21] Anita M. Flynn, Joseph L. Jones, Bruce A. Seiger and Bruce Seiger, **Mobile Robots Inspiration to Implementation**, A K Peters Wellesley, Massachusetts, 1993.

## ภาคผนวก ก

## โปรแกรมควบคุมหุ่นยนต์ 6 ขาด้วยขั้นตอนวิธีการเปิดตาราง

;This complete version of look-up table with philips P89C51RD2 on ETT CP-;SPI/RD2 board  
(Over Clock to 24MHz).

```

;
;It apply to controls CNN 6 legged robot in driver part.
;
;T2MOD=#00h
;T2CON=#00h
;by default
;register bank1
;      R0-R3  temp for process
;      R4     edit servo no
;      R5     count
;      R6     state_no
;      R7     temp_A
;start_time  F830h
;step_time   FFDAh
;stop_time   0000h
;count max   0DFh(223)

```

;\*\*\*\*\* 8052's Registor definition \*\*\*\*\*

```

      TF2  equ  0CFh
      TR2  equ  0CAh
      RCP2H equ 0CBh
      RCP2L equ 0CAh

```

;\*\*\*\*\* indirect addressing to registor \*\*\*\*\*

```

      DPTR_NO  equ  0A2h
      _R1     equ  09h
      _R2     equ  0Ah
      _R4     equ  0Ch

```

;\*\*\*\*\* define system constants \*\*\*\*\*

```

      DPTR0 equ  0000h
      DPTR1 equ  0100h
      start_h equ 0F8h
      start_l equ  30h
      step_h  equ  0FFh
      step_l  equ  0DAh
      stop_h  equ  00h
      stop_l  equ  00h
      ext1   equ  20h
      cnt_max equ  0E0h

```

;\*\*\*\*\* servo position constants \*\*\*\*\*

```

; 1st char      L:left  R:right
; 2nd char      F:front C:center R:rear
; 3rd char      M:move  R:rotate
; 4th char      H:high  C:center L:low

```

```

      LFMH equ 0BEh      ;190
      LFMC equ 84h      ;132

```

```

LFML equ 48h ;75
LFRH equ 8Ch ;140
LFRc equ 5Fh ;95
LFRL equ 32h ;50
LCMH equ 0BCh ;188
LCMC equ 83h ;131
LCML equ 49h ;73
LCRH equ 8Dh ;141
LCRC equ 60h ;96
LCRL equ 33h ;51
LRMH equ 0h
LRMC equ 32h ;50
LRML equ 64h ;100
LRRH equ 8Ah ;138
LRRC equ 5Dh ;93
LRRL equ 30h ;48
RFMH equ 5h
RFMC equ 37h ;55
RFML equ 69h ;105
RFRH equ 25h ;37
RFRc equ 52h ;82
RFRL equ 7Fh ;127
RCMH equ 0h
RCMC equ 32h ;50
RCML equ 64h ;100
RCRH equ 3Ah ;58
RCRC equ 57h ;87
RCRL equ 74h ;116
RRMH equ 9Fh ;159
RRMC equ 65h ;101
RRML equ 2Ch ;44
RRRH equ 25h ;37
RRRC equ 52h ;82
RRRL equ 7Fh ;127

```

```

;***** declare system variable *****

```

```

ext1ful equ 7Ah
t_state equ 7Bh
time equ 7Ch
temp_A equ 7Dh
temp_S0 equ 7Eh
temp_S1 equ 7Fh
servo_0 equ p0
servo_1 equ p2
err_out equ p1
persent equ 80h
next equ 90h

```

```

;***** declare servo variable *****

```

```

LFM equ 9Fh
LFR equ 9Eh
LCM equ 9Dh
LCR equ 9Ch
LRM equ 9Bh

```

```

LRR    equ    9Ah
RFM    equ    90h
RFR    equ    91h
RCM    equ    92h
RCR    equ    93h
RRM    equ    94h
RRR    equ    95h

        org    0000h
        ajmp   Init

        org    001Bh           ;T1    int_vector
        ajmp   T1_serv

Int_serv:clr    org    002Bh           ;T2    int_vector
                TF2
                xch    A,temp_A
                setb   RS0           ;set    Registor_bank1,Active
                mov    R7,A
                movx   A,@DPTR       ;mov    A,@DPTR0
                xrl   A,temp_S0
                mov    temp_S0,A
                mov    servo_0,A
                inc    DPTR           ; +2MC
                mov    DPTR_NO,#01h  ;set    DPTR1,Active
                movx   A,@DPTR       ;mov    A,@DPTR1
                xrl   A,temp_S1
                mov    temp_S1,A
                mov    servo_1,A
                inc    DPTR           ; +2MC
                mov    DPTR_NO,#00h  ;set    DPTR0,Active

begin_select:  cjne   R6,#01h,no_step_state ;step state (state 1)
                djnz  R5,non_over
                mov   RCP2H,#stop_h
                mov   RCP2L,#stop_1
                mov   R6,#02h         ;set    state_no,2
                mov   A,R7
                clr   RS0             ;set    Registor_bank0,Active
                xch  A,temp_A
                reti

non_over:      mov   RCP2H,#step_h
                mov   RCP2L,#step_1
                mov   A,R7
                clr   RS0             ;set    Registor_bank0,Active
                xch  A,temp_A
                reti

no_step_state: cjne   R6,#00h,no_start_state ;start state (state 0)
                mov   R5,#0DFh       ;clr    count
                mov   RCP2H,#step_h
                mov   RCP2L,#step_1
                mov   R6,#01h         ;set    state_no,1
                mov   A,R7
                clr   RS0             ;set    Registor_bank0,Active

                xch  A,temp_A
                reti

```

```

no_start_state:  cjne    R6,#02h,no_stop_state    ;stop state (state 2)

                push    DPH                    ;save   @SP,DPTR0
                push    DPL
                mov     DPTR_NO,#01h          ;set    DPTR1,Active
                push    DPH                    ;save   @SP,DPTR1
                push    DPL
                mov     DPTR_NO,#00h          ;set    DPTR0,Active

                mov     R0,#80h
                mov     R1,#90h
                mov     R3,#80h
find_loop_S0:    mov     A,R3
                rl      A
                mov     R3,A
                mov     A,@R0
                mov     _R2,@R1
                cjne   A,_R2,is_new_S0
is_new_S0:      ajmp   not_new_S0
                mov     DPL,@R0
                inc     DPTR
                movx   A,@DPTR
                XRL    A,R3
                movx   @DPTR,A
                mov     DPL,@R1
                inc     DPTR
                movx   A,@DPTR
                XRL    A,R3
                movx   @DPTR,A
not_new_S0:     mov     @R0,_R2
                inc     R0
                inc     R1
                cjne   R0,#88h,find_loop_S0

                mov     DPTR_NO,#01h          ;set    DPTR1,Active
find_loop_S1:   mov     A,R3
                rl      A
                mov     R3,A
                mov     A,@R0
                mov     _R2,@R1
                cjne   A,_R2,is_new_S1
is_new_S1:     ajmp   not_new_S1
                mov     DPL,@R0
                inc     DPTR
                movx   A,@DPTR
                XRL    A,R3
                movx   @DPTR,A
                mov     DPL,@R1
                inc     DPTR
                movx   A,@DPTR
                XRL    A,R3
                movx   @DPTR,A
not_new_S1:    mov     @R0,_R2
                inc     R0
                inc     R1
                cjne   R0,#90h,find_loop_S1

                mov     RCP2H,#step_h

```

```

        mov     RCP2L,#step_1

        pop     DPL                               ;get   DPTR1,@SP
        pop     DPH
        mov     DPTR_NO,#00h                     ;set   DPTR0,Active
        pop     DPL                               ;get   DPTR1,@SP
        pop     DPH
        mov     R6,#03h                           ;set   state_no,0

        mov     A,#00h                             ; if time_H==0 then
        add     A,time
        jnz     no_full
Over_flow:  mov     ext1ful,#01h                   ;       ext1ful=1
        mov     time,#ext1                       ;       time=ext1
no_full:   ; else
        dec     time                             ;       time--
        ;end if

        mov     A,R7
        clr     RS0                               ;set   Registor_bank0,Active
        setb    ES
        xch    A,temp_A
        reti

no_stop_state:  cjne  R6,#03h,state_error       ;disable com state (state 3)
        clr     ES
        mov     RCP2H,#start_h
        mov     RCP2L,#start_l
        mov     DPTR_NO,#01h                     ;set   DPTR1,Active
        mov     DPTR,#DPTR1                       ;mov   DPTR1,#DPTR1
        mov     DPTR_NO,#00h                     ;set   DPTR0,Active
        mov     DPTR,#DPTR0                       ;mov   DPTR0,#DPTR0
        mov     R6,#00h                           ;set   state_no,0
        mov     A,R7
        clr     RS0                               ;set   Registor_bank0,Active
        xch    A,temp_A
        reti

state_error:  mov     err_out,#01h                 ;out   error01
        mov     A,R7
        clr     RS0                               ;set   Registor_bank0,Active
        xch    A,temp_A
        reti

T1_serv:clr  TF1
        xch    A,temp_A
        setb   RS0                               ;set   Registor_bank1,Active

        clr     RS0                               ;set   Registor_bank0,Active
        xch    A,temp_A
        reti

; ***** End T2 Interrupt Service Routine ,begin System Initial *****

init:       mov     sp,#2Fh
        mov     RCP2H,#start_h
        mov     RCP2L,#start_l
        mov     TH2,#5Ah
        mov     TL2,#00h
        mov     IE,#0A0h

```

```

mov servo_0,#00h
mov temp_S0,#00h
mov servo_1,#00h
mov temp_S1,#00h

setb TR2
setb RS0 ;set Registor_bank1,Active
mov R7,A ;mov temp_A,A

mov DPTR,#DPTR0 ;mov DPTR0,#0000h
mov DPTR_NO,#01h ;set DPTR1,Active
mov DPTR,#DPTR1 ;mov DPTR1,#0100h
mov DPTR_NO,#00h ;set DPTR0,Active

mov A,#0FFh
movx @DPTR,A ;mov DPTR0,#0FFh
inc DPTR
mov DPTR_NO,#01h ;set DPTR1,Active
movx @DPTR,A ;mov DPTR1,#0FFh
inc DPTR
mov DPTR_NO,#00h ;set DPTR0,Active

table_loop:
mov R0,#persent
mov R1,#next
mov @R0,#80h
mov @R1,#80h
inc R0
inc R1
cjne R0,#(persent)+10h,table_loop

set_table_loop:
mov R1,#00h
mov R0,#80h
mov R2,#00h
loopservo_0_7:
mov A,R2 ;R2 is temp flag
RR A
mov R2,A
mov A,@R0
cjne A,_R1,NoEqu_0_7 ;cjne A,R1,NoEqu_0-7
orl _R2,#80h
NoEqu_0_7:
inc R0
cjne R0,#88h,LoopServo_0_7
mov A,R2
movx @DPTR,A ;mov DPTR0,A
inc DPTR ;inc DPTR0
mov 0A2h,#01h ;set DPTR1,Active

LoopServo_8_15:
mov R2,#00h
mov A,R2
RR A
mov R2,A
mov A,@R0
cjne A,_R1,NoEqu_8_15 ;cjne A,R1,NoEqu_8-15
orl _R2,#80h
NoEqu_8_15:
inc R0
cjne R0,#90h,LoopServo_8_15
mov A,R2
movx @DPTR,A ;mov DPTR1,A
inc DPTR ;inc DPTR1

```

```

mov    0A2h,#00h           ;set    DPTR0,Active

inc    R1
cjne   R1,#cnt_max,set_table_loop

mov    A,#00h
movx   @DPTR,A             ;mov    DPTR0,#00h
inc    DPTR
movx   @DPTR,A
mov    DPTR,#DPTR0         ;mov    DPTR0,#0000h
mov    DPTR_NO,#01h       ;set    DPTR1,Active
movx   @DPTR,A             ;mov    DPTR1,#00h
inc    DPTR
movx   @DPTR,A
mov    DPTR,#DPTR1        ;mov    DPTR1,#0100h
mov    DPTR_NO,#00h       ;set    DPTR0,Active
mov    R6,#00h            ;set    state_no,0

;                               ***** Table Init *****

mov    time,#ext1
mov    ext1ful,#00h
mov    t_state,#01h

mov    R0,#LFM             ;LFM=LFMC
mov    @R0,#LFMC
mov    R0,#LFR             ;LFR=LFRC
mov    @R0,#LFRC
mov    R0,#LCM             ;LCM=LCCM
mov    @R0,#LCCM
mov    R0,#LCR             ;LCR=LCRC
mov    @R0,#LCRC
mov    R0,#LRM             ;LRM=LRMC
mov    @R0,#LRMC
mov    R0,#LRR             ;LRR=LRRC
mov    @R0,#LRRC
mov    R0,#RFM             ;RFM=RFMC
mov    @R0,#RFMC
mov    R0,#RFR             ;RFR=RFRC
mov    @R0,#RFRC
mov    R0,#RCM             ;RCM=RCMC
mov    @R0,#RCMC
mov    R0,#RCR             ;RCR=RCRC
mov    @R0,#RCRC
mov    R0,#RRM             ;RRM=RRMC
mov    @R0,#RRMC
mov    R0,#RRR             ;RRR=RRRC
mov    @R0,#RRRC

;                               ***** Return Def *****

mov    A,R7                ;mov    A,temp_A
clr    RS0                 ;set    Registor_bank0,Active
ajmp   start_program

;-----
start_program:  mov    A,ext1ful           ;if ext1ful==1 then
                add    A,#00h
                jnz   ful
                ajmp  no_ful

```

```

ful:      mov     R1,t_state      ; if t_state==1 then
state1:   djnz   R1,state2
          mov     R0,#LFM       ; LFM=LFML
          mov     @R0,#LFML
          mov     R0,#LCM       ; LCM=LCML
          mov     @R0,#LCML
          mov     R0,#LRM       ; LRM=LRML
          mov     @R0,#LRML
          mov     R0,#RFM       ; RFM=RFML
          mov     @R0,#RFML
          mov     R0,#RCM       ; RCM=RCML
          mov     @R0,#RCML
          mov     R0,#RRM       ; RRM=RRML
          mov     @R0,#RRML
          ajmp   common_st
state2:   djnz   R1,state3      ; else if t_state==2 then
          mov     R0,#LFM
          mov     @R0,#LFMH     ; LFM=LFMH
          mov     R0,#LRM
          mov     @R0,#LRMH     ; LRM=LRMH
          mov     R0,#RCM
          mov     @R0,#RCMH     ; RCM=RCMH
          ajmp   common_st
state3:   djnz   R1,state4      ; else if t_state==3 then
          mov     R0,#LFR       ; LFR=LFRH
          mov     @R0,#LFRH
          mov     R0,#LRR       ; LRR=LRRH
          mov     @R0,#LRRH
          mov     R0,#RCR       ; RCR=RCRH
          mov     @R0,#RCRH
          ajmp   common_st
state4:   djnz   R1,state5      ; else if t_state==4 then
          mov     R0,#LFM       ; LFM=LFML
          mov     @R0,#LFML
          mov     R0,#LRM       ; LRM=LRML
          mov     @R0,#LRML
          mov     R0,#RCM       ; RCM=RCML
          mov     @R0,#RCML
          ajmp   common_st
state5:   djnz   R1,state6      ; else if t_state==5 then
          mov     R0,#LCM       ; LCM=LCMH
          mov     @R0,#LCMH
          mov     R0,#RFM       ; RFM=RFMH
          mov     @R0,#RFMH
          mov     R0,#RRM       ; RRM=RRMH
          mov     @R0,#RRMH
          sjmp   common_st
state6:   djnz   R1,state7      ; else if t_state==6 then
          mov     R0,#LFR       ; LFR=LFRL
          mov     @R0,#LFRL
          mov     R0,#LCR       ; LCR=LCRH
          mov     @R0,#LCRH
          mov     R0,#LRR       ; LRR=LRRL
          mov     @R0,#LRRL
          mov     R0,#RFR       ; RFR=RFRH
          mov     @R0,#RFRH
          mov     R0,#RCR       ; RCR=RCRL
          mov     @R0,#RCRL
          mov     R0,#RRR       ; RRR=RRRH
          mov     @R0,#RRRH

```

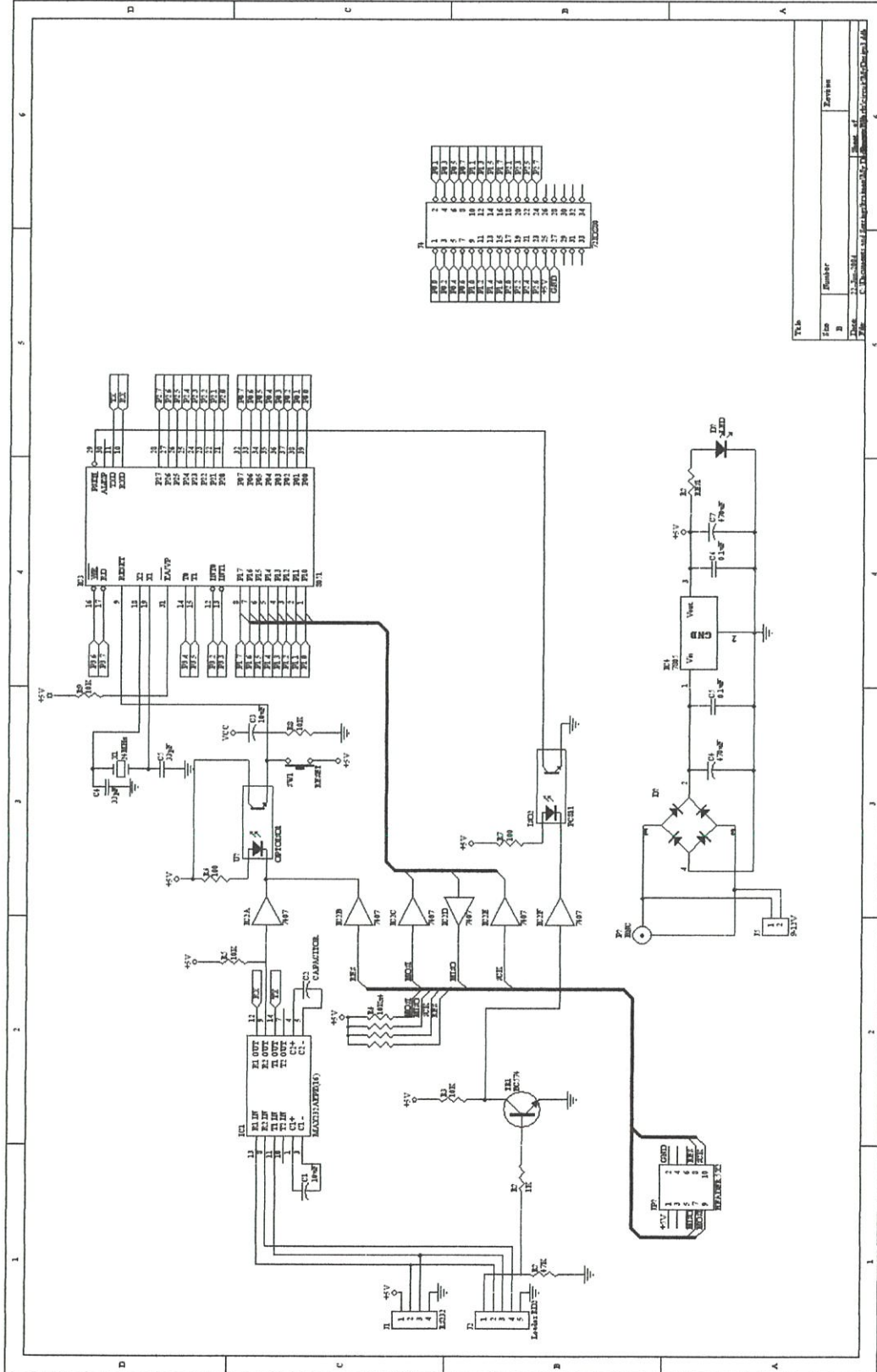
```

state7:      sjmp    common_st
             djnz   R1,state8          ; else if t_state==7 then
             mov    R0,#LCM           ; LCM=LCML
             mov    @R0,#LCML
             mov    R0,#RFM           ; RFM=RFML
             mov    @R0,#RFML
             mov    R0,#RRM           ; RRM=RRML
             mov    @R0,#RRML
state8:      sjmp    common_st
             djnz   R1,state9          ; else if t_state==8 then
             mov    R0,#LFM           ; LFM=LFMH
             mov    @R0,#LFMH
             mov    R0,#LRM           ; LRM=LRMH
             mov    @R0,#LRMH
             mov    R0,#RCM           ; RCM=RCMH
             mov    @R0,#RCMH
state9:      sjmp    common_st
             djnz   R1,state10         ; else if t_state==9 then
             mov    R0,#LFR           ; LFR=LFRH
             mov    @R0,#LFRH
             mov    R0,#LCR           ; LCR=LCRL
             mov    @R0,#LCRL
             mov    R0,#LRR           ; LRR=LRRH
             mov    @R0,#LRRH
             mov    R0,#RFR           ; RFR=RFRL
             mov    @R0,#RFRL
             mov    R0,#RCR           ; RCR=RCRH
             mov    @R0,#RCRH
             mov    R0,#RRR           ; RRR=RRRL
             mov    @R0,#RRRL
state10:     sjmp    common_st
             djnz   R1,state11         ; else if t_state==10 then
             mov    R0,#LFM           ; LFM=LFML
             mov    @R0,#LFML
             mov    R0,#LRM           ; LRM=LRML
             mov    @R0,#LRML
             mov    R0,#RCM           ; RCM=RCML
             mov    @R0,#RCML
state11:     sjmp    common_st
             djnz   R1,table_err       ; else if t_state==11 then
             mov    R0,#LCM           ; LCM=LCMH
             mov    @R0,#LCMH
             mov    R0,#RFM           ; RFM=RFMH
             mov    @R0,#RFMH
             mov    R0,#RRM           ; RRM=RRMH
             mov    @R0,#RRMH
common_st:   mov    t_state,#05h       ; t_state=5
             ; end if all
             inc    t_state            ; t_state++
no_ful:      mov    ext1ful,#00h       ; ext1ful=0
             ;else
table_err:   ajmp   start_program     ; infinite loop
             mov    err_out,#02h      ;out error02
             ajmp   start_program     ; infinite loop
             end

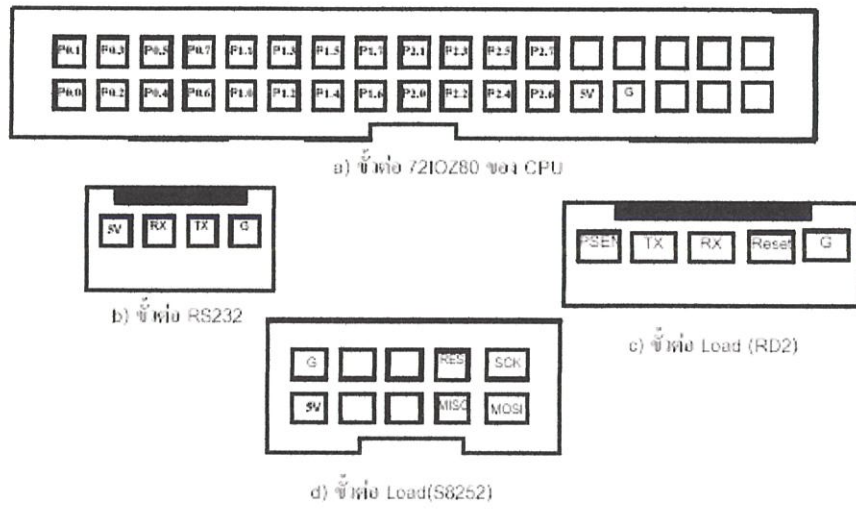
```

### วงจรควบคุมหุ่นยนต์ 6 ขาด้วยขั้นตอนวิธีการเปิดตาราง

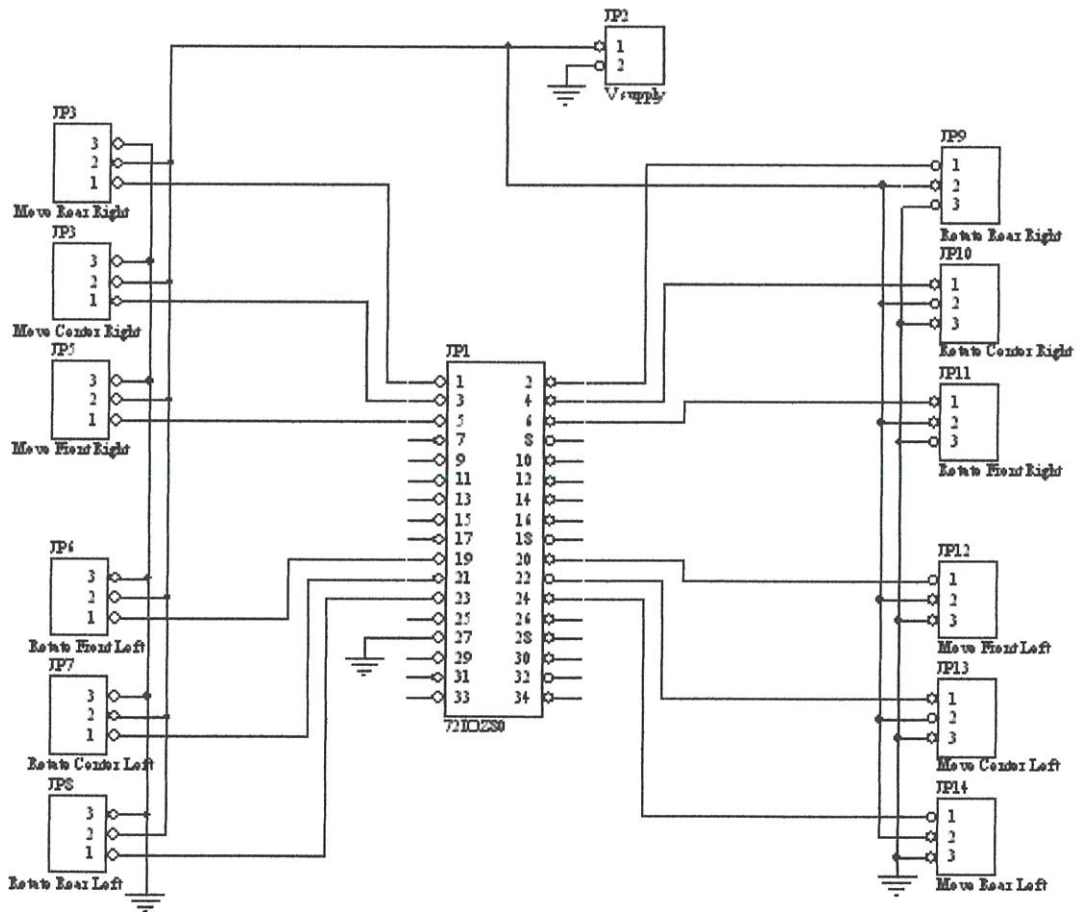
อาศัยชุดควบคุมรุ่น CP-SPI/RD2 ของบริษัท ETT จำกัด โดยใช้ พอร์ท 0 ควบคุมขา ด้านขวา และ พอร์ท 2 ควบคุมขา ด้านซ้าย โดยมีรายละเอียดของวงจรดังนี้



รูปที่ ก.1 วงจรของชุดควบคุม CP-SPI/RD2 ของบริษัท ETT จำกัด



รูปที่ ก.2 หัวต่อใช้งานอุปกรณ์ภายนอกของชุดควบคุม CP-SPI/RD2



รูปที่ ก.3 การต่อเซอร์โวมอเตอร์กับชุดควบคุม CP-SPI/RD2

## ภาคผนวก ข

## โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์รูปที่ 3.5

```

MAX_COUNT      = 200;
TURN_LEFT      = 100;
RETURN_FROM_LEFT = 200;
TURN_RIGHT     = 300;
RETURN_FROM_RIGHT = 400;
CELL_NUM       = 6;
U_ROTATE       = 0.6;
I_ROTATE       = 0;
X              = 0.2;
U_MOVE         = 1;
I_MOVE         = 0;

count=[1:MAX_COUNT];
Urot=U_ROTATE.*ones(CELL_NUM,1);
Irot=I_ROTATE.*ones(CELL_NUM,1);
Xrot=X.*ones(CELL_NUM,1);
ListYrot=0.*ones(CELL_NUM,1);
Umove=U_MOVE.*ones(CELL_NUM,1);
Imove=I_MOVE.*ones(CELL_NUM,1);
Xmove=X.*ones(CELL_NUM,1);
ListYmove=0.*ones(CELL_NUM,1);

for loop_count=1:MAX_COUNT-1
    Yrot=0.5.*(abs(Xrot+1)-abs(Xrot-1));
    for cell_count=1:CELL_NUM-1

        Xrot(cell_count,:)=Urot(cell_count,:)*Yrot(cell_count,:)+Urot(cell_count+1,:)*Yrot(c
        ell_count+1,:)+Irot(cell_count,:);
    end
    Xrot(CELL_NUM,:)=Urot(CELL_NUM,:)*Yrot(CELL_NUM,:)-
        Urot(1,:)*Yrot(1,:)+Irot(CELL_NUM,:);
    ListYrot=[ListYrot Yrot];
    Ymove=2*(abs(Yrot+0.25)-abs(Yrot-0.25));
    ListYmove=[ListYmove Ymove];
end

figure(1);
mesh(ListYrot);
axis([1 MAX_COUNT 1 CELL_NUM -1.5 1.5]);
figure(2);
subplot(4,1,1); plot(count,ListYrot(4,:),'r',count,ListYrot(5,:),'g',count,ListYrot(6,:),'b');
axis([1 MAX_COUNT -1.5 1.5]);
title('Plot CNN output [Yrot4(r) Yrot5(g) Yrot6(b) for L-legs rotate');
subplot(4,1,2);
plot(count,ListYmove(1,:),'r',count,ListYmove(2,:),'g',count,ListYmove(3,:),'b');
axis([1 MAX_COUNT -1.5 1.5]);
title('Plot CNN output [Yrot4(r) Yrot5(g) Yrot6(b) for L-legs rotate');
subplot(4,1,3); plot(count,ListYrot(1,:),'r',count,ListYrot(2,:),'g',count,ListYrot(3,:),'b');
axis([1 MAX_COUNT -1.5 1.5]);
title('Plot CNN output [Yrot4(r) Yrot5(g) Yrot6(b) for L-legs rotate');
subplot(4,1,4);
plot(count,ListYmove(4,:),'r',count,ListYmove(5,:),'g',count,ListYmove(6,:),'b');
axis([1 MAX_COUNT -1.5 1.5]);
title('Plot CNN output [Yrot4(r) Yrot5(g) Yrot6(b) for L-legs rotate');

```

### โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์รูปที่ 3.6

```

MAX_COUNT      = 200;
TURN_LEFT      = 100;
RETURN_FROM_LEFT = 200;
TURN_RIGHT     = 300;
RETURN_FROM_RIGHT = 400;
CELL_NUM       = 6;
U_ROTATE       = 0.6;
I_ROTATE       = 0;
X              = 0.2;
U_MOVE         = 1;
I_MOVE         = 0;

count=[1:MAX_COUNT];
Urot=U_ROTATE.*ones(CELL_NUM,1);
Irot=I_ROTATE.*ones(CELL_NUM,1);
Xrot=X.*ones(CELL_NUM,1);
ListYrot=0.*ones(CELL_NUM,1);
Umove=U_MOVE.*ones(CELL_NUM,1);
Imove=I_MOVE.*ones(CELL_NUM,1);
Xmove=X.*ones(CELL_NUM,1);
ListYmove=0.*ones(CELL_NUM,1);

for loop_count=1:MAX_COUNT-1
    Yrot=0.5.*(abs(Xrot+1)-abs(Xrot-1));
    for cell_count=1:CELL_NUM-1
        Xrot(cell_count,:)=Urot(cell_count,:)*Yrot(cell_count,:)-
            Urot(cell_count+1,:)*Yrot(cell_count+1,:)+Irot(cell_count,:);
    end

    Xrot(CELL_NUM,:)=Urot(CELL_NUM,:)*Yrot(CELL_NUM,:)+Urot(1,:)*Yrot(1,:)+Irot(CELL_NUM,:);
    ListYrot=[ListYrot Yrot];
    Ymove=0.5.*(abs(Xmove+1)-abs(Xmove-1));
    for cell_count=1:CELL_NUM-1
        Xmove(cell_count,:)=Umove(cell_count,:)*Ymove(cell_count,:)-
            Umove(cell_count+1,:)*Ymove(cell_count+1,:)+Imove(cell_count,:);
    end

    Xmove(CELL_NUM,:)=Umove(CELL_NUM,:)*Ymove(CELL_NUM,:)+
        Umove(1,:)*Ymove(1,:)+Imove(CELL_NUM,:);
    ListYmove=[ListYmove Ymove];
end

figure(1);
mesh(ListYrot); shading interp
axis([1 MAX_COUNT 1 CELL_NUM -1.5 1.5]);
figure(2);
plot(count,ListYrot(1,:),'r',count,ListYrot(2,:),'g',count,ListYrot(3,:),'b',count,ListYrot(4,:),'y',
    count,ListYrot(5,:),'m',count,ListYrot(6,:),'c');
axis([1 50 -1.5 1.5]); title('Plot CNN output [Yrot1(r) Yrot2(g) Yrot3(b) Yrot4(y) Yrot5(m)
Yrot6(c) ]');
```

## โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์รูปที่ 3.8

### – 3.9

```

MAX_COUNT      = 100;
TURN_LEFT      = 100;
RETURN_FROM_LEFT = 200;
TURN_RIGHT     = 300;
RETURN_FROM_RIGHT = 400;
CELL_NUM       = 6;
MAX_U          = 100;
I_ROTATE       = 0;
X              = 0.2;
U_MOVE         = 1;
I_MOVE         = 0;
count=[1:MAX_COUNT];
ListY1=0.*ones(MAX_COUNT,1);
Umove=U_MOVE.*ones(CELL_NUM,1);
Imove=I_MOVE.*ones(CELL_NUM,1);
ListYmove=0.*ones(CELL_NUM,1);
for u_count=1:MAX_U
    Urot=((u_count)/100).*ones(CELL_NUM,1);
    Xrot=X.*ones(CELL_NUM,1);
    Irot=I_ROTATE.*ones(CELL_NUM,1);
    Xmove=X.*ones(CELL_NUM,1);
    ListYrot=0.*ones(CELL_NUM,1);
    for loop_count=1:MAX_COUNT-1
        Yrot=0.5.*(abs(Xrot+1)-abs(Xrot-1));
        for cell_count=1:CELL_NUM-1
            Xrot(cell_count,:)=Urot(cell_count,:)*Yrot(cell_count,:)-
                Urot(cell_count+1,:)*Yrot(cell_count+1,:)+Irot(cell_count,:);
        end
        Xrot(CELL_NUM,:)=Urot(CELL_NUM,:)*Yrot(CELL_NUM,:)+Urot(1,:)*Yrot(1,:)+Irot(CELL_NUM,:);
        ListYrot=[ListYrot Yrot];
    end
end
if u_count>40 & u_count<70
    temp=ListYrot(1,:);
    ListY1=[ListY1 temp];
end
end

figure(1);
surf(ListY1); shading interp
xlabel('U')
ylabel('time(t)')
zlabel('output(y)')
title('output of I = 0');
axis([1 30 1 MAX_COUNT -1.5 1.5]);
set(gca,'XTickLabel',{'0.45';'0.50';'0.55';'0.60';'0.65';'0.70'})
colorbar

```

## โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลาร์รูปที่

### 3.10 – 3.11

```

MAX_COUNT      = 200;
TURN_LEFT      = 100;
RETURN_FROM_LEFT = 200;
TURN_RIGHT     = 300;
RETURN_FROM_RIGHT = 400;
CELL_NUM       = 6;
MAX_I          = 200;
U_ROTATE       = 0.65;
X              = 0.2;
U_MOVE         = 1;
I_MOVE         = 0;
Z = peaks; surf(Z);
axis tight
count=[1:MAX_COUNT];
ListY1=0.*ones(MAX_COUNT,1);
Umove=U_MOVE.*ones(CELL_NUM,1);
Imove=I_MOVE.*ones(CELL_NUM,1);
ListYmove=0.*ones(CELL_NUM,1);
for i_count=1:MAX_I
    Urot=U_ROTATE.*ones(CELL_NUM,1);
    Xrot=X.*ones(CELL_NUM,1);
    Irot=((i_count)-100)/100.*ones(CELL_NUM,1);
    Xmove=X.*ones(CELL_NUM,1);
    ListYrot=0.*ones(CELL_NUM,1);
    for loop_count=1:MAX_COUNT-1
        Yrot=0.5.*(abs(Xrot+1)-abs(Xrot-1));
        for cell_count=1:CELL_NUM-1
            Xrot(cell_count,:)=Urot(cell_count,:)*Yrot(cell_count,:)-
                Urot(cell_count+1,:)*Yrot(cell_count+1,:)+Irot(cell_count,:);
        end

            Xrot(CELL_NUM,:)=Urot(CELL_NUM,:)*Yrot(CELL_NUM,:)+Urot(1,:)*Yrot(1,:)+
                Irot(CELL_NUM,:);
        ListYrot=[ListYrot Yrot];
    end
    if i_count>40 & i_count<160
        temp=ListYrot(1,:);
        ListY1=[ListY1 temp];
    end
end
end
ListY1(:,1)=ListY1(:,2);
ListY1(1,:)=ListY1(3,:);
ListY1(2,:)=ListY1(3,:);
figure(1);
xlabel('I value')
ylabel('time(t)')
zlabel('output(y)')
title('output of U =')
title(j);
surf(ListY1);shading interp
axis([1 120 1 MAX_COUNT -1.5 1.5]);

```

## โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลาร์รูปที่

### 3.12 – 3.13

```

MAX_COUNT      = 100;
TURN_LEFT      = 100;
RETURN_FROM_LEFT = 200;
TURN_RIGHT     = 300;
RETURN_FROM_RIGHT = 400;
CELL_NUM       = 6;
U_ROTATE       = 0.6;
I_ROTATE       = 0;
X              = 0.2;
MAX_U          = 200;
I_MOVE         = 0;

count=[1:MAX_COUNT];

ListY1=0.*ones(MAX_COUNT,1);
Urot=U_ROTATE.*ones(CELL_NUM,1);
Imove=I_MOVE.*ones(CELL_NUM,1);
for u_count=1:MAX_U
    Umove=((u_count-100)/25).*ones(CELL_NUM,1);
    Xrot=X.*ones(CELL_NUM,1);
    Irot=I_ROTATE.*ones(CELL_NUM,1);
    Xmove=X.*ones(CELL_NUM,1);
    ListYrot=0.*ones(CELL_NUM,1);
    ListYmove=0;
    for loop_count=1:MAX_COUNT-1
        Yrot=0.5.*(abs(Xrot+1)-abs(Xrot-1));
        for cell_count=1:CELL_NUM-1
            Xrot(cell_count,:)=Urot(cell_count,:)*Yrot(cell_count,:)-
Urot(cell_count+1,:)*Yrot(cell_count+1,:)+Irot(cell_count,:);
            end
        Xrot(CELL_NUM,:)=Urot(CELL_NUM,:)*Yrot(CELL_NUM,:)+Urot(1,:)*Yrot(1,:)+Irot(CELL_NUM,
M,:);
        ListYrot=[ListYrot Yrot];
        Ymove=0.5.*(abs(Umove(1,:)*Yrot(1,:)+1)-abs(Umove(1,:)*Yrot(1,:)-1));

        ListYmove=[ListYmove Ymove];
    end
    temp=ListYmove(1,:);
    ListY1=[ListY1 temp];
end
ListY1(:,1)=ListY1(:,2);
ListY1(1,:)=ListY1(3,:);
ListY1(2,:)=ListY1(3,:);
figure(1);
surf(ListY1);shading interp
xlabel('U')
ylabel('time(t)')
zlabel('output(y)')
title('output of I = 0');
axis([1 200 1 MAX_COUNT -1 1]);
set(gca,'XTick',[25 50 75 100 125 150 175 200]);
set(gca,'XTickLabel',{'-0.75';'-0.50';'-0.25';'0.00';'0.25';'0.50';'0.75';'1.00'})
colorbar

```

## โปรแกรม MATLAB ในการจำลองโครงข่ายประสาทเทียมแบบเซลล์ลาร์รูปที่

### 3.13 – 3.14

```

MAX_COUNT      = 200;
TURN_LEFT      = 100;
RETURN_FROM_LEFT = 200;
TURN_RIGHT     = 300;
RETURN_FROM_RIGHT = 400;
CELL_NUM       = 6;
U_ROTATE       = 0.65;
I_ROTATE       = 0;
X              = 0.2;
U_MOVE         = 1;
MAX_I          = 100;

count=[1:MAX_COUNT];

ListY1=0.*ones(MAX_COUNT,1);
Umove=U_MOVE.*ones(CELL_NUM,1);
Irot=I_ROTATE.*ones(CELL_NUM,1);
Urot=U_ROTATE.*ones(CELL_NUM,1);
for i_count=1:MAX_I
    Imove=(((i_count)-50)/20).*ones(CELL_NUM,1);
    Xrot=X.*ones(CELL_NUM,1);
    Xmove=X.*ones(CELL_NUM,1);
    ListYrot=0.*ones(CELL_NUM,1);
    ListYmove=0.*ones(CELL_NUM,1);
    for loop_count=1:MAX_COUNT-1
        Yrot=0.5.*(abs(Xrot+1)-abs(Xrot-1));
        for cell_count=1:CELL_NUM-1
            Xrot(cell_count,:)=Urot(cell_count,:)*Yrot(cell_count,:)-
                Urot(cell_count+1,:)*Yrot(cell_count+1,:)+Irot(cell_count,:);
        end

        Xrot(CELL_NUM,:)=Urot(CELL_NUM,:)*Yrot(CELL_NUM,:)+Urot(1,:)*Yrot(1,:)+Irot(C
            ELL_NUM,:);
        ListYrot=[ListYrot Yrot];
        Xmove=Umove.*Yrot+Imove;
        Ymove=0.5.*(abs(Xmove+1)-abs(Xmove-1));
        ListYmove=[ListYmove Ymove];
    end
    temp=ListYmove(1,:);
    ListY1=[ListY1 temp];
end
ListY1(:,1)=ListY1(:,2);
ListY1(1,:)=ListY1(3,:);
ListY1(2,:)=ListY1(3,:);
figure(1);
surf(ListY1);shading interp
xlabel('direct current(I)')
ylabel('time(t)')
title('output of U = 1');
axis([1 100 1 MAX_COUNT -1 1]);
set(gca,'XTick',[12.5 25 37.5 50 62.5 75 87.5 100 ]);
set(gca,'XTickLabel',{'-0.75','-0.50','-0.25','0.00','0.25','0.50','0.75','1.00'})
colorbar

```

## ภาคผนวก ค

## โปรแกรมจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์ (CNN Simulator Ver.1.0)

'This code for simulate 6 legged robot controls signal which can generate signal by 'lookup table algorithm, CNN and manual control. It is send data to com port '(RS232C) at 2400 bps which selected by user at run time. It is send data in form of:

'  
' <servonumber[n]>,<position[n]>  
'

' where

' servonumber is number code of servo will be controls by:

' 0F0h=P0.0, 0F1h=P0.1,..., 0F7h=P0.7, 0F8h=P2.0, 0F9h=P2.1,...,0FFh=P2.7

' position is position of servo moving.

' It values is between 0-128(0-7Fh).  
'

'The format of RS232 is 1 start, 1 parity and 2 stop bit  
'

<b>Dim table</b>	<i>As Boolean</i>	'Table mode flag
<b>Dim CNN</b>	<i>As Boolean</i>	'CNN mode flag
<b>Dim Pause</b>	<i>As Boolean</i>	'Pause mode flag
<b>Dim State</b>	<i>As Integer</i>	'State No.
<b>Dim Xrot(6)</b>	<i>As Double</i>	'X rotate (6 cells)
<b>Dim Yrot(6)</b>	<i>As Double</i>	'Y rotate „
<b>Dim Xmov(6)</b>	<i>As Double</i>	'X move „
<b>Dim Ymov(6)</b>	<i>As Double</i>	'Y move „
<b>Dim I</b>	<i>As Double</i>	' I constant
<b>Dim Urot</b>	<i>As Double</i>	' $\mu$ rotate (6 cells)
<b>Dim Umov</b>	<i>As Double</i>	' $\mu$ move „
<b>Dim MFL</b>	<i>As Double</i>	'Move Front Left ( $-1 \leq values \leq 1$ )
<b>Dim RFL</b>	<i>As Double</i>	'Rotate Front Left „
<b>Dim MCL</b>	<i>As Double</i>	'Move Center Left „
<b>Dim RCL</b>	<i>As Double</i>	'Rotate Center Left „
<b>Dim MRL</b>	<i>As Double</i>	'Move Rear Left „
<b>Dim RRL</b>	<i>As Double</i>	'Rotate Rear Left „
<b>Dim MFR</b>	<i>As Double</i>	'Move Front Right „
<b>Dim RFR</b>	<i>As Double</i>	'Rotate Front Right „
<b>Dim MCR</b>	<i>As Double</i>	'Move Center Right „
<b>Dim RCR</b>	<i>As Double</i>	'Rotate Center Right „
<b>Dim MRR</b>	<i>As Double</i>	'Move Rear Right „
<b>Dim RRR</b>	<i>As Double</i>	'Rotate Rear Right „
<b>Dim CNNMFL</b>	<i>As Double</i>	'CNN of Move Front Left ( $-1 \leq values \leq 1$ )
<b>Dim CNNRFL</b>	<i>As Double</i>	'CNN of Rotate Front Left „
<b>Dim CNNMCL</b>	<i>As Double</i>	'CNN of Move Center Left „
<b>Dim CNNRCL</b>	<i>As Double</i>	'CNN of Rotate Center Left „

<b>Dim</b> CNNMRL	<i>As Double</i>	'CNN of Move Rear Left	„
<b>Dim</b> CNNRRL	<i>As Double</i>	'CNN of Rotate Rear Left	„
<b>Dim</b> CNNMFR	<i>As Double</i>	'CNN of Move Front Right	„
<b>Dim</b> CNNRFR	<i>As Double</i>	'CNN of Rotate Front Right	„
<b>Dim</b> CNNMCR	<i>As Double</i>	'CNN of Move Center Right	„
<b>Dim</b> CNNRCR	<i>As Double</i>	'CNN of Rotate Center Right	„
<b>Dim</b> CNNMRR	<i>As Double</i>	'CNN of Move Rear Right	„
<b>Dim</b> CNNRRR	<i>As Double</i>	'CNN of Rotate Rear Right	„
<b>Dim</b> TempMFL	<i>As Double</i>	'Temp of Move Front Left	(-1 ≤ values ≤ 1)
<b>Dim</b> TempRFL	<i>As Double</i>	'Temp of Rotate Front Left	„
<b>Dim</b> TempMCL	<i>As Double</i>	'Temp of Move Center Left	„
<b>Dim</b> TempRCL	<i>As Double</i>	'Temp of Rotate Center Left	„
<b>Dim</b> TempMRL	<i>As Double</i>	'Temp of Move Rear Left	„
<b>Dim</b> TempRRL	<i>As Double</i>	'Temp of Rotate Rear Left	„
<b>Dim</b> TempMFR	<i>As Double</i>	'Temp of Move Front Right	„
<b>Dim</b> TempRFR	<i>As Double</i>	'Temp of Rotate Front Right	„
<b>Dim</b> TempMCR	<i>As Double</i>	'Temp of Move Center Right	„
<b>Dim</b> TempRCR	<i>As Double</i>	'Temp of Rotate Center Right	„
<b>Dim</b> TempMRR	<i>As Double</i>	'Temp of Move Rear Right	„
<b>Dim</b> TempRRR	<i>As Double</i>	'Temp of Rotate Rear Right	„
<b>Dim</b> TMFL(3)	<i>As Double</i>	'Table of Move Front Left	(-1 ≤ values ≤ 1)
<b>Dim</b> TRFL(3)	<i>As Double</i>	'Table of Rotate Front Left	„
<b>Dim</b> TMCL(3)	<i>As Double</i>	'Table of Move Center Left	„
<b>Dim</b> TRCL(3)	<i>As Double</i>	'Table of Rotate Center Left	„
<b>Dim</b> TMRL(3)	<i>As Double</i>	'Table of Move Rear Left	„
<b>Dim</b> TRRL(3)	<i>As Double</i>	'Table of Rotate Rear Left	„
<b>Dim</b> TMFR(3)	<i>As Double</i>	'Table of Move Front Right	„
<b>Dim</b> TRFR(3)	<i>As Double</i>	'Table of Rotate Front Right	„
<b>Dim</b> TMCR(3)	<i>As Double</i>	'Table of Move Center Right	„
<b>Dim</b> TRCR(3)	<i>As Double</i>	'Table of Rotate Center Right	„
<b>Dim</b> TMRR(3)	<i>As Double</i>	'Table of Move Rear Right	„
<b>Dim</b> TRRR(3)	<i>As Double</i>	'Table of Rotate Rear Right	„
<b>Public Sub</b> <i>InitTable()</i>		'Initial table for table mode	
TMFL(0) = -1			
TMFR(0) = 1			
TMCL(0) = 1			
TMCR(0) = -1			
TMRL(0) = -1			
TMRR(0) = 1			
TRFL(0) = 0			
TRCL(0) = 0			
TRRL(0) = 0			
TRFR(0) = 0			
TRCR(0) = 0			
TRRR(0) = 0			

TMFL(1) = 0  
 TMFR(1) = 0  
 TMCL(1) = 0  
 TMCR(1) = 0  
 TMRL(1) = 0  
 TMRR(1) = 0

TRFL(1) = -1  
 TRCL(1) = 1  
 TRRL(1) = -1  
 TRFR(1) = 1  
 TRCR(1) = -1  
 TRRR(1) = 1

TMFL(2) = 1  
 TMFR(2) = -1  
 TMCL(2) = -1  
 TMCR(2) = 1  
 TMRL(2) = 1  
 TMRR(2) = -1

TRFL(2) = 0  
 TRCL(2) = 0  
 TRRL(2) = 0  
 TRFR(2) = 0  
 TRCR(2) = 0  
 TRRR(2) = 0

TMFL(3) = 0  
 TMFR(3) = 0  
 TMCL(3) = 0  
 TMCR(3) = 0  
 TMRL(3) = 0  
 TMRR(3) = 0

TRFL(3) = 1  
 TRCL(3) = -1  
 TRRL(3) = 1  
 TRFR(3) = -1  
 TRCR(3) = 1  
 TRRR(3) = -1

**End Sub**

**Public Sub RunState()**

CmdStop.Enabled = True  
 CmdPause.Enabled = True  
 CmdTable.Enabled = False  
 CmdCNN.Enabled = False  
 CmdManual.Enabled = False

‘Set object status at run state times.

```

OptCom1.Enabled = False
OptCom2.Enabled = False
ChkMoveFrontLeft.Enabled = False
ChkMoveFrontRight.Enabled = False
ChkMoveCentreLeft.Enabled = False
ChkMoveCentreRight.Enabled = False
ChkMoveRearLeft.Enabled = False
ChkMoveRearRight.Enabled = False
ChkRotateFrontLeft.Enabled = False
ChkRotateFrontRight.Enabled = False
ChkRotateCentreLeft.Enabled = False
ChkRotateCentreRight.Enabled = False
ChkRotateRearLeft.Enabled = False
ChkRotateRearRight.Enabled = False
TxtMoveFrontLeft.Enabled = False
TxtMoveFrontRight.Enabled = False
TxtMoveCentreLeft.Enabled = False
TxtMoveCentreRight.Enabled = False
TxtMoveRearLeft.Enabled = False
TxtMoveRearRight.Enabled = False
TxtRotateFrontLeft.Enabled = False
TxtRotateFrontRight.Enabled = False
TxtRotateCentreLeft.Enabled = False
TxtRotateCentreRight.Enabled = False
TxtRotateRearLeft.Enabled = False
TxtRotateRearRight.Enabled = False
CmdStop.SetFocus
TempMFL = TxtMoveFrontLeft.Text
TempRFL = TxtRotateFrontLeft.Text
TempMCL = TxtMoveCentreLeft.Text
TempRCL = TxtRotateCentreLeft.Text
TempMRL = TxtMoveRearLeft.Text
TempRRL = TxtRotateRearLeft.Text
TempMFR = TxtMoveFrontRight.Text
TempRFR = TxtRotateFrontRight.Text
TempMCR = TxtMoveCentreRight.Text
TempRCR = TxtRotateCentreRight.Text
TempMRR = TxtMoveRearRight.Text
TempRRR = TxtRotateRearRight.Text
Pause = False
TmrDrive.Enabled = True

```

**End Sub**

```

Public Sub drive()                                'Send data to com port
    Dim Buffer As Variant
    MyCom.PortOpen = True
    Buffer = Chr(240)
    MyCom.Output = Buffer
    Buffer = Chr(Int(100 - ((MFR + 0.9) * 100) / 2))

```

```

MyCom.Output = Buffer
Buffer = Chr(241)
MyCom.Output = Buffer
Buffer = Chr(Int(130 - ((RFR + 1.05) * 90) / 2))
MyCom.Output = Buffer
Buffer = Chr(242)
MyCom.Output = Buffer
Buffer = Chr(Int(100 - ((MCR + 1) * 100) / 2))
MyCom.Output = Buffer
Buffer = Chr(243)
MyCom.Output = Buffer
Buffer = Chr(Int(130 - ((RCR + 0.95) * 90) / 2))
MyCom.Output = Buffer
Buffer = Chr(244)
MyCom.Output = Buffer
Buffer = Chr(Int((((MRR + 0.9) * 115) / 2) + 50))
MyCom.Output = Buffer
Buffer = Chr(245)
MyCom.Output = Buffer
Buffer = Chr(Int(130 - ((RRR + 1.05) * 90) / 2))
MyCom.Output = Buffer
Buffer = Chr(250)
MyCom.Output = Buffer
Buffer = Chr(Int((((RRL + 1.18) * 90) / 2) + 40))
MyCom.Output = Buffer
Buffer = Chr(251)
MyCom.Output = Buffer
Buffer = Chr(Int(100 - ((MRL + 1) * 100) / 2))
MyCom.Output = Buffer
Buffer = Chr(252)
MyCom.Output = Buffer
Buffer = Chr(Int((((RCL + 1.25) * 90) / 2) + 40))
MyCom.Output = Buffer
Buffer = Chr(253)
MyCom.Output = Buffer
Buffer = Chr(Int((((MCL + 0.975) * 115) / 2) + 75))
MyCom.Output = Buffer
Buffer = Chr(254)
MyCom.Output = Buffer
Buffer = Chr(Int((((RFL + 1.225) * 90) / 2) + 40))
MyCom.Output = Buffer
Buffer = Chr(255)
MyCom.Output = Buffer
Buffer = Chr(Int((((MFL + 1) * 115) / 2) + 75))
MyCom.Output = Buffer
MyCom.PortOpen = False
End Sub

Private Sub CmdCNN_Click()
    CNN = True

```

```

Umov = 4
Urot = 1
RunState
For n = 0 To 5
    Xrot(n) = 0.2
    Yrot(n) = 0.2
    Xmov(n) = 0.2
Next n
I = 0
TmrCNN.Enabled = True
End Sub

Public Sub RunTable()
    If TMFL(State) <> 0 Then
        MFL = TMFL(State)
    End If
    If TRFL(State) <> 0 Then
        RFL = TRFL(State)
    End If
    If TMCL(State) <> 0 Then
        MCL = TMCL(State)
    End If
    If TRCL(State) <> 0 Then
        RCL = TRCL(State)
    End If
    If TMRL(State) <> 0 Then
        MRL = TMRL(State)
    End If
    If TRRL(State) <> 0 Then
        RRL = TRRL(State)
    End If
    If TMFR(State) <> 0 Then
        MFR = TMFR(State)
    End If
    If TRFR(State) <> 0 Then
        RFR = TRFR(State)
    End If
    If TMCR(State) <> 0 Then
        MCR = TMCR(State)
    End If
    If TRCR(State) <> 0 Then
        RCR = TRCR(State)
    End If
    If TMRR(State) <> 0 Then
        MRR = TMRR(State)
    End If
    If TRRR(State) <> 0 Then
        RRR = TRRR(State)
    End If
    TxtMoveFrontLeft.Text = MFL

```

'lookup table for table mode

```

TxtMoveFrontRight.Text = MFR
TxtMoveCentreLeft.Text = MCL
TxtMoveCentreRight.Text = MCR
TxtMoveRearLeft.Text = MRL
TxtMoveRearRight.Text = MRR
TxtRotateFrontLeft.Text = RFL
TxtRotateFrontRight.Text = RFR
TxtRotateCentreLeft.Text = RCL
TxtRotateCentreRight.Text = RCR
TxtRotateRearLeft.Text = RRL
TxtRotateRearRight.Text = RRR

```

**End Sub**

**Public Sub RunCNN()**

'Put CNN output before send to port.

```

RFL = Yrot(0)
RCL = Yrot(1)
RRL = Yrot(2)
RFR = Yrot(3)
RCR = Yrot(4)
RRR = Yrot(5)
MFL = Ymov(2)
MCL = Ymov(3)
MRL = Ymov(4)
MFR = Ymov(5)
MCR = -Ymov(0)
MRR = -Ymov(1)
TxtMoveFrontLeft.Text = MFL
TxtMoveFrontRight.Text = MFR
TxtMoveCentreLeft.Text = MCL
TxtMoveCentreRight.Text = MCR
TxtMoveRearLeft.Text = MRL
TxtMoveRearRight.Text = MRR
TxtRotateFrontLeft.Text = RFL
TxtRotateFrontRight.Text = RFR
TxtRotateCentreLeft.Text = RCL
TxtRotateCentreRight.Text = RCR
TxtRotateRearLeft.Text = RRL
TxtRotateRearRight.Text = RRR

```

**End Sub**

**Private Sub CmdManual\_Click()**

```

Dim Buffer As Variant
MyCom.PortOpen = True
Dim MoveFrontRight As Integer
If ChkMoveFrontRight.Value = Checked Then
    Buffer = Chr(240)

```

```

MyCom.Output = Buffer
If TxtMoveFrontRight.Text > 1 Then TxtMoveFrontRight.Text = 1
If TxtMoveFrontRight.Text < -1 Then TxtMoveFrontRight.Text = -1
MoveFrontRight = Int(100 - ((TxtMoveFrontRight.Text + 0.9) * 100) / 2)
Buffer = Chr(MoveFrontRight)
MyCom.Output = Buffer
End If
Dim RotateFrontRight As Integer
If ChkRotateFrontRight.Value = Checked Then
    Buffer = Chr(241)
    MyCom.Output = Buffer
    If TxtRotateFrontRight.Text > 1 Then TxtRotateFrontRight.Text = 1
    If TxtRotateFrontRight.Text < -1 Then TxtRotateFrontRight.Text = -1
    RotateFrontRight = Int(130 - ((TxtRotateFrontRight.Text + 1.05) * 90) / 2)
    Buffer = Chr(RotateFrontRight)
    MyCom.Output = Buffer
End If
Dim MoveCentreRight As Integer
If ChkMoveCentreRight.Value = Checked Then
    Buffer = Chr(242)
    MyCom.Output = Buffer
    If TxtMoveCentreRight.Text > 1 Then TxtMoveCentreRight.Text = 1
    If TxtMoveCentreRight.Text < -1 Then TxtMoveCentreRight.Text = -1
    MoveCentreRight = Int(100 - ((TxtMoveCentreRight.Text + 1) * 100) / 2)
    Buffer = Chr(MoveCentreRight)
    MyCom.Output = Buffer
End If
Dim RotateCentreRight As Integer
If ChkRotateCentreRight.Value = Checked Then
    Buffer = Chr(243)
    MyCom.Output = Buffer
    If TxtRotateCentreRight.Text > 1 Then TxtRotateCentreRight.Text = 1
    If TxtRotateCentreRight.Text < -1 Then TxtRotateCentreRight.Text = -1
    RotateCentreRight = Int(130 - ((TxtRotateCentreRight.Text + 0.95) * 90) / 2)
    Buffer = Chr(RotateCentreRight)
    MyCom.Output = Buffer
End If
Dim MoveRearRight As Integer
If ChkMoveRearRight.Value = Checked Then
    Buffer = Chr(244)
    MyCom.Output = Buffer
    If TxtMoveRearRight.Text > 1 Then TxtMoveRearRight.Text = 1
    If TxtMoveRearRight.Text < -1 Then TxtMoveRearRight.Text = -1
    MoveRearRight = Int((((TxtMoveRearRight.Text + 0.9) * 115) / 2) + 50)
    Buffer = Chr(MoveRearRight)
    MyCom.Output = Buffer
End If
Dim RotateRearRight As Integer
If ChkRotateRearRight.Value = Checked Then
    Buffer = Chr(245)

```

```

MyCom.Output = Buffer
If TxtRotateRearRight.Text > 1 Then TxtRotateRearRight.Text = 1
If TxtRotateRearRight.Text < -1 Then TxtRotateRearRight.Text = -1
RotateRearRight = Int(130 - ((TxtRotateRearRight.Text + 1.05) * 90) / 2)
Buffer = Chr(RotateRearRight)
MyCom.Output = Buffer
End If
Dim RotateRearLeft As Integer
If ChkRotateRearLeft.Value = Checked Then
    Buffer = Chr(250)
    MyCom.Output = Buffer
    If TxtRotateRearLeft.Text > 1 Then TxtRotateRearLeft.Text = 1
    If TxtRotateRearLeft.Text < -1 Then TxtRotateRearLeft.Text = -1
    RotateRearLeft = Int(((((TxtRotateRearLeft.Text + 1.18) * 90) / 2) + 40)
    Buffer = Chr(RotateRearLeft)
    MyCom.Output = Buffer
End If
Dim MoveRearLeft As Integer
If ChkMoveRearLeft.Value = Checked Then
    Buffer = Chr(251)
    MyCom.Output = Buffer
    If TxtMoveRearLeft.Text > 1 Then TxtMoveRearLeft.Text = 1
    If TxtMoveRearLeft.Text < -1 Then TxtMoveRearLeft.Text = -1
    MoveRearLeft = Int(100 - ((TxtMoveRearLeft.Text + 1) * 100) / 2)
    Buffer = Chr(MoveRearLeft)
    MyCom.Output = Buffer
End If
Dim RotateCentreLeft As Integer
If ChkRotateCentreLeft.Value = Checked Then
    Buffer = Chr(252)
    MyCom.Output = Buffer
    If TxtRotateCentreLeft.Text > 1 Then TxtRotateCentreLeft.Text = 1
    If TxtRotateCentreLeft.Text < -1 Then TxtRotateCentreLeft.Text = -1
    RotateCentreLeft = Int(((((TxtRotateCentreLeft.Text + 1.25) * 90) / 2) + 40)
    Buffer = Chr(RotateCentreLeft)
    MyCom.Output = Buffer
End If
Dim MoveCentreLeft As Integer
If ChkMoveCentreLeft.Value = Checked Then
    Buffer = Chr(253)
    MyCom.Output = Buffer
    If TxtMoveCentreLeft.Text > 1 Then TxtMoveCentreLeft.Text = 1
    If TxtMoveCentreLeft.Text < -1 Then TxtMoveCentreLeft.Text = -1
    MoveCentreLeft = Int(((((TxtMoveCentreLeft.Text + 0.975) * 115) / 2) + 75)
    Buffer = Chr(MoveCentreLeft)
    MyCom.Output = Buffer
End If
Dim RotateFrontLeft As Integer
If ChkRotateFrontLeft.Value = Checked Then
    Buffer = Chr(254)

```

```

    MyCom.Output = Buffer
    If TxtRotateFrontLeft.Text > 1 Then TxtRotateFrontLeft.Text = 1
    If TxtRotateFrontLeft.Text < -1 Then TxtRotateFrontLeft.Text = -1
    RotateFrontLeft = Int((((TxtRotateFrontLeft.Text + 1.225) * 90) / 2) + 40)
    Buffer = Chr(RotateFrontLeft)
    MyCom.Output = Buffer
End If
Dim MoveFrontLeft As Integer
If ChkMoveFrontLeft.Value = Checked Then
    Buffer = Chr(255)
    MyCom.Output = Buffer
    If TxtMoveFrontLeft.Text > 1 Then TxtMoveFrontLeft.Text = 1
    If TxtMoveFrontLeft.Text < -1 Then TxtMoveFrontLeft.Text = -1
    MoveFrontLeft = Int((((TxtMoveFrontLeft.Text + 1) * 115) / 2) + 75)
    Buffer = Chr(MoveFrontLeft)
    MyCom.Output = Buffer
End If
MyCom.PortOpen = False
End Sub

Private Sub CmdPause_Click()
    If Pause Then
        Pause = False
        TmrDrive = True
        If CNN Then TmrCNN = True
        CmdPause.Caption = "Pause"
        CmdStop.SetFocus
    Else
        Pause = True
        TmrDrive = False
        TmrCNN = False
        CmdPause.Caption = "Play"
        CmdPause.SetFocus
    End If
End Sub

Private Sub CmdStop_Click()
    TxtMoveFrontLeft.Text = TempMFL
    TxtRotateFrontLeft.Text = TempRFL
    TxtMoveCentreLeft.Text = TempMCL
    TxtRotateCentreLeft.Text = TempRCL
    TxtMoveRearLeft.Text = TempMRL
    TxtRotateRearLeft.Text = TempRRL
    TxtMoveFrontRight.Text = TempMFR
    TxtRotateFrontRight.Text = TempRFR
    TxtMoveCentreRight.Text = TempMCR
    TxtRotateCentreRight.Text = TempRCR
    TxtMoveRearRight.Text = TempMRR
    TxtRotateRearRight.Text = TempRRR
    CmdStop.Enabled = False

```

CmdPause.Enabled = False  
 CmdTable.Enabled = True  
 CmdCNN.Enabled = True  
 CmdManual.Enabled = True  
 OptCom1.Enabled = True  
 OptCom2.Enabled = True  
 ChkMoveFrontLeft.Enabled = True  
 ChkMoveFrontRight.Enabled = True  
 ChkMoveCentreLeft.Enabled = True  
 ChkMoveCentreRight.Enabled = True  
 ChkMoveRearLeft.Enabled = True  
 ChkMoveRearRight.Enabled = True  
 ChkRotateFrontLeft.Enabled = True  
 ChkRotateFrontRight.Enabled = True  
 ChkRotateCentreLeft.Enabled = True  
 ChkRotateCentreRight.Enabled = True  
 ChkRotateRearLeft.Enabled = True  
 ChkRotateRearRight.Enabled = True  
 TxtMoveFrontLeft.Enabled = True  
 TxtMoveFrontRight.Enabled = True  
 TxtMoveCentreLeft.Enabled = True  
 TxtMoveCentreRight.Enabled = True  
 TxtMoveRearLeft.Enabled = True  
 TxtMoveRearRight.Enabled = True  
 TxtRotateFrontLeft.Enabled = True  
 TxtRotateFrontRight.Enabled = True  
 TxtRotateCentreLeft.Enabled = True  
 TxtRotateCentreRight.Enabled = True  
 TxtRotateRearLeft.Enabled = True  
 TxtRotateRearRight.Enabled = True  
 TmrDrive.Enabled = False

**If table Then**

    table = False

    CmdTable.SetFocus

**End If**

**If CNN Then**

    CNN = False

    CmdCNN.SetFocus

    TmrCNN.Enabled = False

**End If**

MFL = 0

RFL = 0

MCL = 0

RCL = 0

MRL = 0

RRL = 0

MFR = 0

RFR = 0

MCR = 0

RCR = 0

```

MRR = 0
RRR = 0
drive
End Sub

```

```

Private Sub CmdTable_Click()
    table = True
    State = 0
    RunState
    CmdStop.SetFocus
    MFL = -1
    RFL = 0
    MCL = -1
    RCL = 0
    MRL = -1
    RRL = 0
    MFR = -1
    RFR = 0
    MCR = -1
    RCR = 0
    MRR = -1
    RRR = 0
    drive
    TxtMoveFrontLeft.Text = TxtMoveFrontLeft.Text + 1
End Sub

```

```

Private Sub Form_Load()
    InitTable
    table = False
    CNN = False
End Sub

```

```

Private Sub OptCom1_Click()
    MyCom.CommPort = 1
End Sub

```

```

Private Sub OptCom2_Click()
    MyCom.CommPort = 2
End Sub

```

```

Private Sub TmrCNN_Timer()           ‘CNN calculate timer overflow.
    For n = 0 To 4
        Xrot(n) = (Urot * Yrot(n)) - (Urot * Yrot(n + 1)) + I
        Yrot(n) = 0.5 * (Abs(Xrot(n) + 1) - Abs(Xrot(n) - 1))
    Next n
    Xrot(5) = (Urot * Yrot(5)) + (Urot * Yrot(0)) + I

```

```

Yrot(5) = 0.5 * (Abs(Xrot(5) + 1) - Abs(Xrot(5) - 1))
For n = 0 To 5
    Ymov(n) = 0.5 * (Abs(Yrot(n) + 1) - Abs(Yrot(n) - 1))
Next n
End Sub

Private Sub TmrDrive_Timer()           'Drive step timer overflow.
    If table Then RunTable
    If CNN Then RunCNN
    drive
    If State < 3 Then
        State = State + 1
    Else
        State = 0
    End If
End Sub

'***** Object Property in application. *****

VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0";
"MSCOMM32.OCX"
Begin VB.Form CNN
    Caption           = "CNN Simulator V 1.0"
    ClientHeight      = 2775
    ClientLeft        = 3300
    ClientTop         = 3900
    ClientWidth       = 7575
    Icon              = (None)
    ScaleHeight       = 2775
    ScaleWidth        = 7575
    Begin VB.CommandButton CmdPause
        Caption       = "Pause"
        Enabled       = 0           'False
        Height        = 375
        Left          = 5400
        TabIndex      = 32
        Top           = 2400
        Width         = 975
    End
End

Begin VB.CheckBox ChkMoveRearRight
    Caption           = "MoveRearRight"
    Height            = 255
    Left              = 5760
    TabIndex          = 14
    Top               = 1320
    Width             = 1695
End
Begin VB.CheckBox ChkRotateRearRight

```

```
    Caption      = "RotateRearRight"
    Height       = 255
    Left         = 3960
    TabIndex     = 13
    Top          = 1320
    Width        = 1695
End
Begin VB.CheckBox ChkMoveCentreRight
    Caption      = "MoveCentreRight"
    Height       = 255
    Left         = 5760
    TabIndex     = 10
    Top          = 720
    Width        = 1695
End
Begin VB.CheckBox ChkRotateCentreRight
    Caption      = "RotateCentreRight"
    Height       = 255
    Left         = 3960
    TabIndex     = 9
    Top          = 720
    Width        = 1695
End
Begin VB.CheckBox ChkMoveFrontRight
    Caption      = "MoveFrontRight"
    Height       = 255
    Left         = 5760
    TabIndex     = 6
    Top          = 120
    Width        = 1695
End
Begin VB.CheckBox ChkRotateFrontRight
    Caption      = "RotateFrontRight"
    Height       = 255
    Left         = 3960
    TabIndex     = 5
    Top          = 120
    Width        = 1695
End

Begin VB.CheckBox ChkRotateRearLeft
    Caption      = "RotateRearLeft"
    Height       = 255
    Left         = 1920
    TabIndex     = 12
    Top          = 1320
    Width        = 1695
End
Begin VB.CheckBox ChkMoveRearLeft
```

```
    Caption      = "MoveRearLeft"
    Height       = 255
    Left         = 120
    TabIndex     = 11
    Top         = 1320
    Width        = 1695
End
Begin VB.CheckBox ChkRotateCentreLeft
    Caption      = "RotateCentreLeft"
    Height       = 255
    Left         = 1920
    TabIndex     = 8
    Top         = 720
    Width        = 1695
End
Begin VB.CheckBox ChkMoveCentreLeft
    Caption      = "MoveCentreLeft"
    Height       = 255
    Left         = 120
    TabIndex     = 7
    Top         = 720
    Width        = 1695
End
Begin VB.CheckBox ChkRotateFrontLeft
    Caption      = "RotateFrontLeft"
    Height       = 255
    Left         = 1920
    TabIndex     = 4
    Top         = 120
    Width        = 1695
End
Begin VB.CheckBox ChkMoveFrontLeft
    Caption      = "MoveFrontLeft"
    Height       = 195
    Left         = 120
    TabIndex     = 3
    Top         = 120
    Width        = 1575
End

Begin VB.OptionButton OptCom2
    Caption      = "Com2"
    Height       = 255
    Left         = 1080
    TabIndex     = 28
    Top         = 2400
    Width        = 735
End
Begin VB.OptionButton OptCom1
```

```

Caption      = "Com1"
Height       = 255
Left         = 120
TabIndex    = 27
Top          = 2400
Value        = -1          'True
Width        = 735
End
Begin VB.Frame FamCom
Height       = 615
Left         = 0
TabIndex    = 29
Top          = 2160
Width        = 2055
End
Begin VB.TextBox TxtMoveRearRight
Alignment    = 1          'Right Justify
BeginProperty DataFormat
Type         = 1
Format       = "#,##0.0000000"
HaveTrueFalseNull = 0
FirstDayOfWeek = 0
FirstWeekOfYear = 0
LCID         = 1054
SubFormatType = 1
EndProperty
Height       = 285
Left         = 5760
TabIndex    = 26
Text         = "0"
Top          = 1560
Width        = 1695
End
Begin VB.TextBox TxtRotateRearRight
Alignment    = 1          'Right Justify
BeginProperty DataFormat
Type         = 1
Format       = "#,##0.0000000"
HaveTrueFalseNull = 0
FirstDayOfWeek = 0
FirstWeekOfYear = 0
LCID         = 1054
SubFormatType = 1
EndProperty
Height       = 285
Left         = 3960
TabIndex    = 25
Text         = "0"
Top          = 1560
Width        = 1695

```

```

End
Begin VB.TextBox TxtMoveCentreRight
    Alignment    = 1           'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"
        HaveTrueFalseNull = 0
        FirstDayOfWeek = 0
        FirstWeekOfYear = 0
        LCID      = 1054
        SubFormatType = 1
    EndProperty
    Height      = 285
    Left        = 5760
    TabIndex    = 22
    Text        = "0"
    Top         = 960
    Width       = 1695
End
Begin VB.TextBox TxtRotateCentreRight
    Alignment    = 1           'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"
        HaveTrueFalseNull = 0
        FirstDayOfWeek = 0
        FirstWeekOfYear = 0
        LCID      = 1054
        SubFormatType = 1
    EndProperty
    Height      = 285
    Left        = 3960
    TabIndex    = 21
    Text        = "0"
    Top         = 960
    Width       = 1695
End

```

```

Begin VB.TextBox TxtMoveFrontRight
    Alignment    = 1           'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"
        HaveTrueFalseNull = 0
        FirstDayOfWeek = 0
        FirstWeekOfYear = 0
        LCID      = 1054
    EndProperty

```

```

        SubFormatType      = 1
    EndProperty
    Height      = 285
    Left        = 5760
    TabIndex    = 18
    Text        = "0"
    Top         = 360
    Width       = 1695
End
Begin VB.TextBox TxtRotateFrontRight
    Alignment    = 1          'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"
        HaveTrueFalseNull = 0
        FirstDayOfWeek = 0
        FirstWeekOfYear = 0
        LCID      = 1054
        SubFormatType = 1
    EndProperty
    Height      = 285
    Left        = 3960
    TabIndex    = 17
    Text        = "0"
    Top         = 360
    Width       = 1695
End
Begin VB.TextBox TxtRotateRearLeft
    Alignment    = 1          'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"
        HaveTrueFalseNull = 0
        FirstDayOfWeek = 0
        FirstWeekOfYear = 0
        LCID      = 1054
        SubFormatType = 1
    EndProperty
    Height      = 285
    Left        = 1920
    TabIndex    = 24
    Text        = "0"
    Top         = 1560
    Width       = 1695
End
Begin VB.TextBox TxtMoveRearLeft
    Alignment    = 1          'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"

```

```

        HaveTrueFalseNull    = 0
        FirstDayOfWeek       = 0
        FirstWeekOfYear      = 0
        LCID                  = 1054
        SubFormatType        = 1
    EndProperty
    Height    = 285
    Left      = 120
    TabIndex  = 23
    Text      = "0"
    Top       = 1560
    Width     = 1695
End
Begin VB.TextBox TxtRotateCentreLeft
    Alignment    = 1          'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"
        HaveTrueFalseNull = 0
        FirstDayOfWeek = 0
        FirstWeekOfYear = 0
        LCID      = 1054
        SubFormatType = 1
    EndProperty
    Height      = 285
    Left        = 1920
    TabIndex    = 20
    Text        = "0"
    Top         = 960
    Width       = 1695
End
Begin VB.TextBox TxtMoveCentreLeft
    Alignment    = 1          'Right Justify
    BeginProperty DataFormat
        Type      = 1
        Format     = "#,##0.0000000"
        HaveTrueFalseNull = 0
        FirstDayOfWeek = 0
        FirstWeekOfYear = 0
        LCID      = 1054
        SubFormatType = 1
    EndProperty
    Height      = 285
    Left        = 120
    TabIndex    = 19
    Text        = "0"
    Top         = 960
    Width       = 1695
End
Begin VB.TextBox TxtRotateFrontLeft

```

```

Alignment      = 1           'Right Justify
BeginProperty DataFormat
  Type         = 1
  Format        = "#,##0.0000000"
  HaveTrueFalseNull = 0
  FirstDayOfWeek = 0
  FirstWeekOfYear = 0
  LCID         = 1054
  SubFormatType = 1
EndProperty
Height        = 285
Left         = 1920
TabIndex     = 16
Text        = "0"
Top         = 360
Width      = 1695
End
Begin VB.TextBox TxtMoveFrontLeft
Alignment      = 1           'Right Justify
BeginProperty DataFormat
  Type         = 1
  Format        = "#,##0.0000000"
  HaveTrueFalseNull = 0
  FirstDayOfWeek = 0
  FirstWeekOfYear = 0
  LCID         = 1054
  SubFormatType = 1
EndProperty
Height        = 285
Left         = 120
TabIndex     = 15
Text        = "0"
Top         = 360
Width      = 1695
End
Begin VB.CommandButton CmdManual
Caption        = "Manual"
Height        = 375
Left         = 6480
TabIndex     = 2
Top         = 2400
Width      = 975
End
Begin VB.CommandButton CmdStop
Caption        = "Stop"
Enabled       = 0           'False
Height        = 375
Left         = 4320
TabIndex     = 30
TabStop      = 0           'False

```

```

        Top          = 2400
        Width        = 975
    End
    Begin VB.CommandButton CmdCNN
        Caption       = "CNN"
        Height        = 375
        Left          = 3240
        TabIndex      = 0
        Top          = 2400
        Width        = 975
    End
    Begin VB.CommandButton CmdTable
        Caption       = "Table"
        Height        = 375
        Left          = 2160
        TabIndex      = 1
        Top          = 2400
        Width        = 975
    End
    Begin MSCommLib.MSComm MyCom
        Left          = 1800
        Top          = 2160
        _ExtentX      = 1005
        _ExtentY      = 1005
        _Version      = 393216
        DTREnable    = -1           'True
        BaudRate      = 2400
        ParitySetting = 4
        StopBits      = 2
    End
    Begin VB.Timer TmrDrive
        Enabled       = 0           'False
        Interval      = 200
        Left          = 2400
        Top          = 2160
    End
    Begin VB.Timer TmrCNN
        Enabled       = 0           'False
        Interval      = 50
        Left          = 2880
        Top          = 2160
    End

    Begin VB.Label LblComment

```



```

stop_h      equ    00h
stop_1      equ    00h
cnt_max     equ    0E0h

temp_S0     equ    7Eh
temp_S1     equ    7Fh
servo_0     equ    p0
servo_1     equ    p2
err_out     equ    p1
persent     equ    80h
next        equ    90h

s1A         equ    95h
s2A         equ    93h
s3A         equ    91h
s1B         equ    94h
s2B         equ    92h
s3B         equ    90h
s1C         equ    9Ah
s2C         equ    9Ch
s3C         equ    9Eh
s1D         equ    9Bh
s2D         equ    9Dh
s3D         equ    9Fh

                org    0000h
                ajmp   Init

                org    0023h                ;Com  int_vector
                ajmp   com_serv

Int_serv:      org    002Bh                ;T2  int_vector
                clr    TF2
                setb   RS0                ;set  Registor_bank1,Active
                mov    R7,A
                movx   A,@DPTR            ;mov  A,@DPTR0
                xrl   A,temp_S0
                mov    temp_S0,A
                mov    servo_0,A
                inc   DPTR                ; +2MC
                mov    DPTR_NO,#01h       ;set  DPTR1,Active
                movx  A,@DPTR            ;mov  A,@DPTR1
                xrl   A,temp_S1
                mov    temp_S1,A
                mov    servo_1,A
                inc   DPTR                ; +2MC
                mov    DPTR_NO,#00h       ;set  DPTR0,Active

begin_select: cjne   R6,#01h,no_step_state ;step state (state 1)
                djnz  R5,non_over
                mov   RCP2H,#stop_h
                mov   RCP2L,#stop_1
                mov   R6,#02h            ;set  state_no,2
                mov   A,R7
                clr   RS0                ;set  Registor_bank0,Active
                reti

non_over:      mov   RCP2H,#step_h
                mov   RCP2L,#step_1
                mov   A,R7

```

```

        clr     RS0                ;set   Registor_bank0,Active
        reti

no_step_state:  cjne   R6,#00h,no_start_state ;start state (state 0)
                mov    R5,#0DFh          ;clr   count
                mov    RCP2H,#step_h
                mov    RCP2L,#step_l
                mov    R6,#01h           ;set   state_no,1
                mov    A,R7
                clr    RS0                ;set   Registor_bank0,Active
                reti

no_start_state: cjne   R6,#02h,no_stop_state ;stop state (state 2)

                push   DPH                ;save  @SP,DPTR0
                push   DPL
                mov    DPTR_NO,#01h       ;set   DPTR1,Active
                push   DPH                ;save  @SP,DPTR1
                push   DPL
                mov    DPTR_NO,#00h       ;set   DPTR0,Active

                mov    R0,#80h
                mov    R1,#90h
                mov    R3,#80h
find_loop_S0:   mov    A,R3
                rl     A
                mov    R3,A
                mov    A,@R0
                mov    _R2,@R1
                cjne   A,_R2,is_new_S0
is_new_S0:     ajmp   not_new_S0
                mov    DPL,@R0
                inc    DPTR
                movx   A,@DPTR
                XRL   A,R3
                movx   @DPTR,A
                mov    DPL,@R1
                inc    DPTR
                movx   A,@DPTR
                XRL   A,R3
                movx   @DPTR,A
                mov    @R0,_R2
not_new_S0:    inc    R0
                inc    R1
                cjne   R0,#88h,find_loop_S0

                mov    DPTR_NO,#01h       ;set   DPTR1,Active
find_loop_S1:   mov    A,R3
                rl     A
                mov    R3,A
                mov    A,@R0
                mov    _R2,@R1
                cjne   A,_R2,is_new_S1
is_new_S1:     ajmp   not_new_S1
                mov    DPL,@R0
                inc    DPTR
                movx   A,@DPTR
                XRL   A,R3

```

```

movX  @DPTR,A
mov   DPL,@R1
inc   DPTR
movx  A,@DPTR
XRL  A,R3
movX  @DPTR,A
mov   @R0,_R2
not_new_S1: inc   R0
inc   R1
cjne  R0,#90h,find_loop_S1

mov   RCP2H,#step_h
mov   RCP2L,#step_l

pop   DPL           ;get  DPTR1,@SP
pop   DPH
mov   DPTR_NO,#00h ;set  DPTR0,Active
pop   DPL           ;get  DPTR1,@SP
pop   DPH
mov   R6,#03h      ;set  state_no,0

mov   A,R7
clr   RS0           ;set  Registor_bank0,Active
setb  ES
reti

no_stop_state: cjne  R6,#03h,state_error ;disable com state (state 3)
clr   ES
mov   RCP2H,#start_h
mov   RCP2L,#start_l
mov   DPTR_NO,#01h ;set  DPTR1,Active
mov   DPTR,#DPTR1 ;mov  DPTR1,#DPTR1
mov   DPTR_NO,#00h ;set  DPTR0,Active
mov   DPTR,#DPTR0 ;mov  DPTR0,#DPTR0
mov   R6,#00h      ;set  state_no,0
mov   A,R7
clr   RS0           ;set  Registor_bank0,Active
reti

state_error:  mov   err_out,#01h ;out  error01
mov   A,R7
clr   RS0           ;set  Registor_bank0,Active
reti

com_serv:    clr   RI
setb  RS0           ;set  Registor_bank1,Active
mov   R7,A
cjne  R4,#00h,srl_state2
mov   A,SBUF
subb  A,#0F0h
jc    data_err1
add   A,#90h
mov   R4,A
sjmp  end_srl
data_err1:  mov   err_out,#02h
sjmp  end_srl
srl_state2: mov   A,SBUF
subb  A,#0E0h
jnc   data_err2
mov   R0,_R4

```



```

        mov     A,R2
        movx   @DPTR,A           ;mov   DPTR0,A
        inc    DPTR             ;inc   DPTR0
        mov    0A2h,#01h       ;set   DPTR1,Active

LoopServo_8_15:
        mov    R2,#00h
        mov    A,R2
        RR    A
        mov    R2,A
        mov    A,@R0
        cjne  A,_R1,NoEqu_8_15 ;cjne  A,R1,NoEqu_8-15
        orl   _R2,#80h
NoEqu_8_15:
        inc    R0
        cjne  R0,#90h,LoopServo_8_15
        mov    A,R2
        movx   @DPTR,A           ;mov   DPTR1,A
        inc    DPTR             ;inc   DPTR1
        mov    0A2h,#00h       ;set   DPTR0,Active

        inc    R1
        cjne  R1,#cnt_max,set_table_loop

        mov    A,#00h
        movx   @DPTR,A           ;mov   DPTR0,#00h
        inc    DPTR
        movx   @DPTR,A
        mov    DPTR,#DPTR0       ;mov   DPTR0,#0000h
        mov    DPTR_NO,#01h     ;set   DPTR1,Active
        movx   @DPTR,A           ;mov   DPTR1,#00h
        inc    DPTR
        movx   @DPTR,A
        mov    DPTR,#DPTR1       ;mov   DPTR1,#0100h
        mov    DPTR_NO,#00h     ;set   DPTR0,Active
        mov    R6,#00h           ;set   state_no,0

;          ***** serial Init *****

        mov    SCON,#0D0h
        mov    TMOD,#20h
        mov    TH1,#0CCh
        mov    TL1,#0CCh
        setb   TR1
        mov    R4,#00h
;          ***** Return Def *****

        mov    A,R7           ;mov   A,temp_A
        clr    RS0           ;set   Registor_bank0,Active
        ajmp   start_program

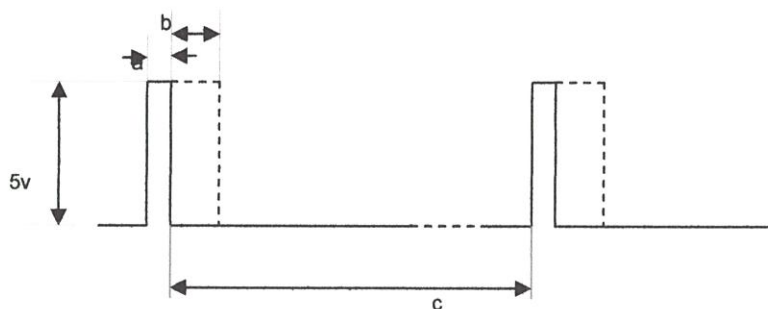
;-----
start_program:  sjmp $
end

```

## ภาคผนวก ง

### การแปลงสัญญาณแอนะล็อกเป็นสัญญาณควบคุมเซอร์โวมอเตอร์

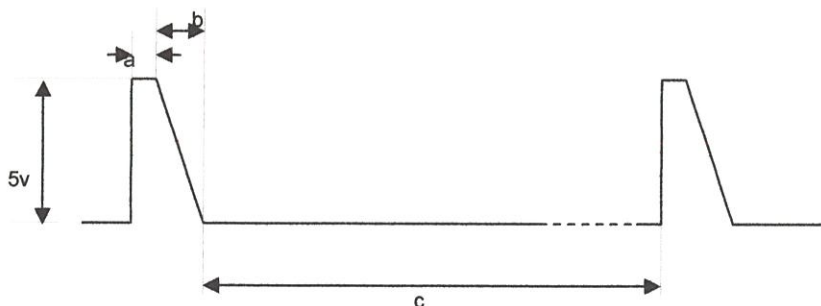
การควบคุมเซอร์โวมอเตอร์ที่ใช้กับอุปกรณ์เครื่องเล่นวิทยุบังคับ ซึ่งอาศัยความกว้างของพัลส์มาเป็นตัวควบคุมองศาในการหมุนของมอเตอร์ โดยมีข้อกำหนดดังนี้



รูปที่ ง.1 สัญญาณที่ใช้ควบคุมเซอร์โวมอเตอร์ ในอุปกรณ์เครื่องเล่นวิทยุบังคับ โดยค่าของ  $a = 0.5 \text{ ms}$ ,  $b = 0-2 \text{ ms}$  ขึ้นอยู่กับมุมที่ต้องการและ  $c > 20 \text{ ms}$

จากรูปที่ ง.1 แสดงข้อกำหนดต่างๆของการควบคุมเซอร์โวมอเตอร์สำหรับอุปกรณ์เครื่องเล่นวิทยุบังคับ ซึ่งนำมาใช้ควบคุมหุ่นยนต์ 6 ขา จากข้อกำหนด พัลส์ควบคุมจะต้องมีความกว้างอย่างน้อยที่สุด  $0.5 \text{ ms}$  และจะมีความกว้างมากขึ้นเมื่อต้องการให้มุมของมอเตอร์มีมากขึ้น ซึ่งค่าสูงสุดที่ค่ากำหนดไว้คือ  $2.5 \text{ ms}$  จะได้มุมมอเตอร์ที่  $180^\circ$

เนื่องจาก สัญญาณที่ได้จากโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์เป็นแบบแอนะล็อก ดังนั้นการควบคุมเซอร์โวมอเตอร์ จึงต้องแปลงสัญญาณดังกล่าวให้เป็นสัญญาณควบคุมแบบความกว้างพัลส์เสียก่อน โดยการสร้างสัญญาณมาเปรียบเทียบเพื่อให้ได้สัญญาณควบคุมเซอร์โวมอเตอร์ตามที่ต้องการ สัญญาณที่สร้างมาเปรียบเทียบมีรูปแบบดังนี้



รูปที่ ง.2 สัญญาณที่สร้างขึ้นมาเปรียบเทียบกับสัญญาณ CNN โดยมีค่าของ  $a = 0.5 \text{ ms}$ ,  $b = 2.5 \text{ ms}$  และ  $c = 20 \text{ ms}$

การสร้างสัญญาณดังรูปที่ ง.2 นั้น เพื่อให้ง่ายและตัดปัญหาทางด้านรูปร่างของสัญญาณ จึงได้อาศัย MCS51 เป็นตัวสร้างสัญญาณ โดยจะนำข้อมูลแบบ 8 บิตจาก MCS51 มาผ่าน วงจร ADC แบบ R2R เพื่อให้ได้สัญญาณที่ต้องการ ซึ่งวงจรดังกล่าวได้แสดงรวมอยู่ในรูปที่ ง.3 ในหัวข้อต่อไปแล้ว ส่วนโปรแกรมที่ใช้ในการสร้างสัญญาณดังกล่าวแสดงดังต่อไปนี้

```

start_h equ    0FCh
start_l  equ    18h
step_h   equ    0FFh
step_l   equ    0FBh
stop_h   equ    60h
stop_l   equ    00h

org      0000h
sjmp    Init

org      000Bh
clr     TF0
step_s: cjne   r2,#00h,start_s
        djnz  r3,stop_s
        mov  TL0,#stop_l
        mov  TH0,#stop_h
        mov  p1,r3
        mov  r2,#01h
        reti

stop_s:  mov  TL0,#step_l
        mov  TH0,#step_h
        mov  p1,r3
        reti

start_s: cjne   r2,#01h,err_s
        mov  r3,#0FFh
        mov  p1,r3
        mov  TL0,#start_l
        mov  TH0,#start_h
        mov  r2,#00h
        reti

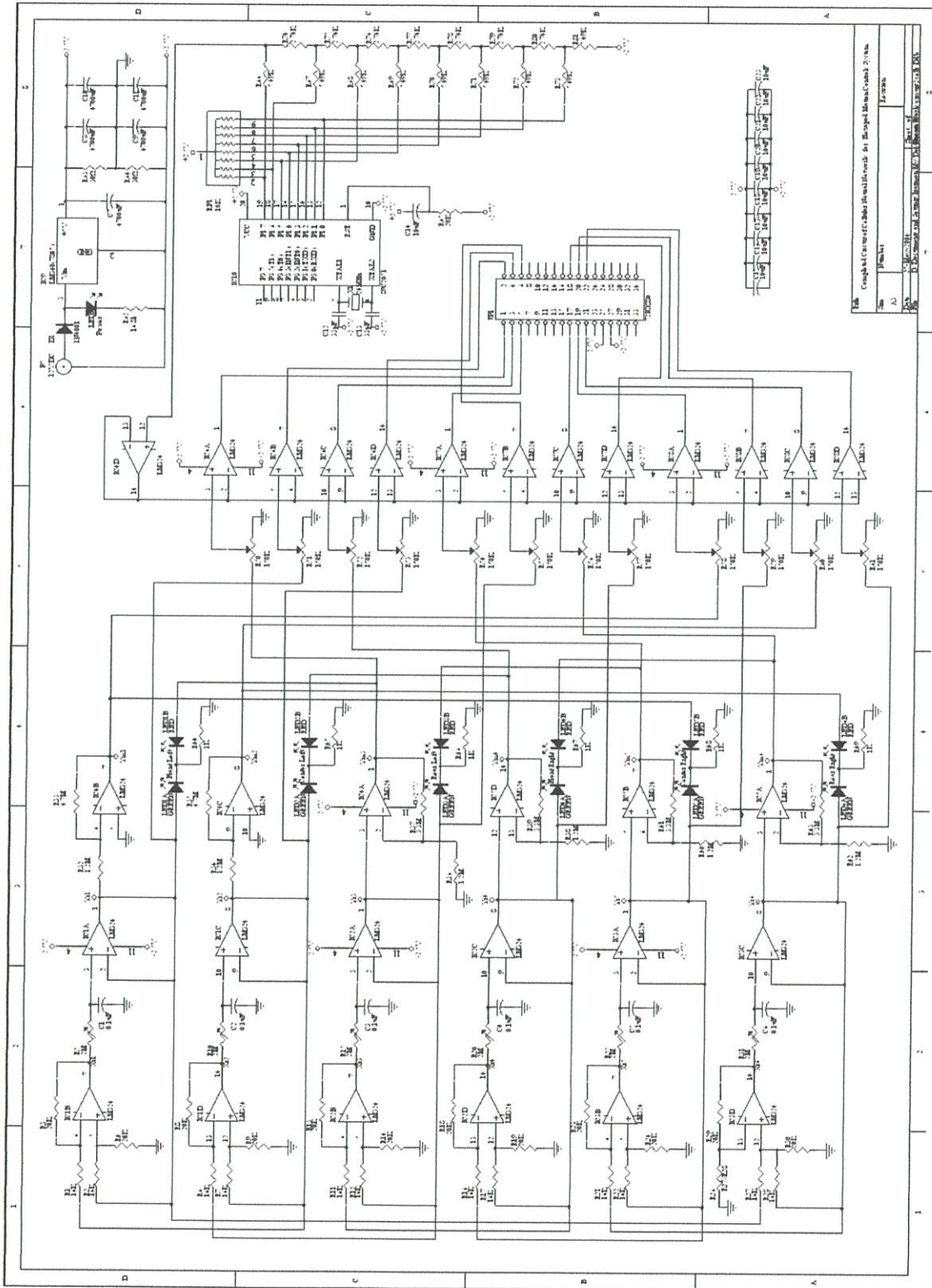
err_s:   mov  p3,#0F0h

Init:   mov  TMOD,#01h
        mov  IE,#82h
        mov  r3,#00h
        mov  TL0,#stop_l
        mov  TH0,#stop_h
        mov  p1,r3
        mov  r2,#01h
        setb TR0
        sjmp $

```

# วงจรจำลองโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์

วงจรรวมทั้งหมดในการควบคุมการเคลื่อนที่ด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์แสดงได้ดังนี้



รูปที่ ง.3 วงจรรวมทั้งหมดของชุดควบคุมการเคลื่อนที่หุ่นยนต์ 6 ขาด้วยโครงข่ายประสาทเทียมแบบเซลล์ลูลาร์

## ประวัติผู้เขียน

นายไกรมน มณีศิลป์ เกิดเมื่อวันที่ 30 พฤศจิกายน 2515 ที่จังหวัด กรุงเทพมหานคร สำเร็จ  
การศึกษาอุตสาหกรรมศาสตรบัณฑิต สาขาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม  
(วิศวกรรมสารสนเทศ) คณะวิศวกรรมศาสตร์ เมื่อปี 2544