

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมมอเตอร์แบบป้อนกลับ (แขนกล)

MOTOR FEEDBACK CONTROL (ROBOT ARM)



โดย
นาย เกรียงศักดิ์ อิมศรี 39013150
นาย ทัศนยา คารคิด 39013180
นาย สุขประชัย ผิวฝ้าย 39013181

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ก - ก

ปีการศึกษา 2541

ก

เลขหมู่.....
เลขทะเบียน.....34040
วัน, เดือน, ปี.....1 ต.ค. 2542

เอกสารนี้สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อเจ้าหน้าที่หอสมุดฯ

การควบคุมมอเตอร์แบบป้อนกลับ (แขนกล)
MOTOR FEEDBACK CONTROL (ROBOT ARM)

โดย

นาย เกรียงศักดิ์ อิ่มศรี 39013150
นาย สัญญา ควรคิด 39013180
นาย สุขประชัย มิวฝ้าย 39013181

อาจารย์ที่ปรึกษา

พลศาสตร์ เลิศประเสริฐ

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

ปริญญาโท ปีการศึกษา 2541

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง

เรื่อง การควบคุมมอเตอร์แบบป้อนกลับ (แขนกล)

ผู้จัดทำ

- | | | |
|------------------|---------|----------|
| นาย เกรียงศักดิ์ | อิมศรี | 39013150 |
| นาย ตัญญา | ควรรคิด | 39013180 |
| นาย สุขประชัย | ผิวฝ้าย | 39013181 |



P.L.อาจารย์ที่ปรึกษา
(อ.พลศาสตร์ เลิศประเสริฐ)

การควบคุมมอเตอร์แบบป้อนกลับ (แขนกล)

MOTOR FEEDBACK CONTROL (ROBOT ARM)

นาย เกียรติศักดิ์ อิ่มศรี 39013150

นาย สัตยญา ทวีตติ 39013180

นาย สุขประชัย ผิวฝ้าย 39013181

อ.พลศาสตร์ เลิศประเสริฐ(อาจารย์ที่ปรึกษา)

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการตอบได้



(อ.พลศาสตร์ เลิศประเสริฐ)
อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบป้อนกลับ (แขนกล)

นาย เกรียงศักดิ์ อิ่มศรี 39013150
นาย สัญญา วรรณคิด 39013180
นาย สุขประชัย ผิวฝ้าย 39013181
อ.พลศาสตร์ เลิศประเสริฐ(อาจารย์ที่ปรึกษา)
ปีการศึกษา 2541

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ กล่าวถึงการออกแบบเพื่อการสร้างแขนกลที่ใช้ในงานอุตสาหกรรม โดยเป็นแขนกล 6 แกนทำงาน โดยใช้ดีซีมอเตอร์ และควบคุมการทำงานโดยใช้ ไมโครคอมพิวเตอร์ ตั้งงาน โดยผ่านการอินเตอร์เฟส

จุดเด่นของแขนกลนี้คือ การใช้สัญญาณป้อนกลับเป็นตัวควบคุมการเคลื่อนที่ตามตำแหน่งที่กำหนด และการใช้ ดีซีมอเตอร์จะช่วยให้มีแรงบิดมากกว่าและช่วยให้อ้างถึงตำแหน่งได้ละเอียดกว่าการใช้สเตปป์มอเตอร์

FEEDBACK CONTROL (ROBOT ARM)

Mr. KRIENGSACK IMSRI 39013150

Mr. SUNYA KUANKID 39013180

Mr. SUKPRACHAI PIEWFAII 39013181

Mr. POLASAS LERSPRASERT(ADVISOR)

ABSTRACT

This thesis presents the control of a Robot Arm that designed for industrial use, which is 6 axis robot arm. This robot arm is driven by DC MOTOR using a microcomputer as the control system.

The outstanding part of Robot Arm is position control using feedback signal to position Control. DC MOTOR has driving more torque and resolution than stepping motor.

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
สารบัญ	III
บทที่ 1 บทนำ	1
บทที่ 2 แขนกล	2
2.1 ชนิดของหุ่นยนต์	2
2.2 มือของหุ่นยนต์ (Robot end effector)	3
2.3 ความแม่นยำในการเคลื่อนที่	5
2.4 การเคลื่อนที่ของหุ่นยนต์	6
บทที่ 3 ดีไซน์มอเตอร์	8
3.1 หลักการทำงานของดีไซน์มอเตอร์	8
3.2 การแยกประเภทของดีไซน์มอเตอร์	9
บทที่ 4 การอินเตอร์เฟส	17
4.1 การ์ดขยาย (Expansion Card)	18
บทที่ 5 โครงสร้างและการออกแบบ	23
5.1 โครงสร้างของโครงการ	24
5.2 ส่วนประกอบโครงการ	25
5.3 การออกแบบส่วนไคร์มอเตอร์	28
5.4 การออกแบบชุดอินเตอร์เฟส	30
5.5 การออกแบบโปรแกรม	33
บทที่ 6 ผลการทดลอง	38
บทที่ 7 สรุปและวิเคราะห์	39
ภาคผนวก	40
กิตติกรรมประกาศ	64
หนังสืออ้างอิง	65

บทที่ 1

บทนำ

โครงการนี้เป็นโครงการควบคุมดิซิมอเตอร์แบบป้อนกลับโดยจะใช้ในการควบคุมแขนกลเพื่อให้แขนกลที่มีแรงบิดสูงและมีค่าความละเอียดในการเคลื่อนที่สูงกว่าการใช้สเตปป์มอเตอร์ ซึ่งในอดีตต้นกำเนิดจริง ๆ ของแขนกลเกิดขึ้นจากนิยายวิทยาศาสตร์แบบเพื่อฝันหรือที่เราเรียกว่าแฟนตาซี(fantasy)ของชาวเชคโกสโลวาเกีย(CZECHOSLOVAKIA) ในปี 1920 โดยทั่วไปแล้วพอพูดถึงคำว่า "หุ่นยนต์" คนทั่วไปมักจะคิดถึงเครื่องยนต์ที่มีรูปร่างเหมือนมนุษย์สามารถที่จะคิดและทำงานได้ด้วยตัวเอง แต่ที่จริงแล้วตามคำนิยาม ROBOTICS หมายถึงเครื่องยนต์ที่ประกอบกันขึ้นมาเพื่อทำงานบางอย่างแทนมนุษย์ สามารถควบคุมได้โดยใช้โปรแกรมทางคอมพิวเตอร์ และสามารถที่จะเปลี่ยนการทำงานให้ทำงานอย่างอื่น ๆ ได้ (ตามคำนิยามของ)สมาคมหุ่นยนต์โรงงาน (Robotics Instrial Association,RIA)และสถาบันเทคโนโลยีเกี่ยวกับหุ่นยนต์ของอเมริกา (Robotics Institute of Amarica, RIA)

จากคำนิยามนี้จะเห็นว่าหุ่นยนต์ก็คือเครื่องยนต์อัตโนมัติชนิดหนึ่ง ที่สามารถเปลี่ยนโปรแกรมการทำงานได้เท่านั้นเองเครื่องยนต์อัตโนมัติ (Automatics)นี้แบ่งออกได้เป็น 3 ชนิด คือ

- 1.Fixed Automation คือเครื่องยนต์อัตโนมัติที่ทำงานอยู่กับที่และทำงานอย่างเดิมตลอด
- 2.Programmable Automation คือเครื่องยนต์อัตโนมัติที่สามารถเปลี่ยนการทำงานได้หลายอย่างแต่ต้องรองานที่ทำในตอนนั้นให้เสร็จก่อนจึงสามารถเปลี่ยนได้
- 3.Flexible Automation คือเครื่องยนต์อัตโนมัติที่สามารถเปลี่ยนการทำงานได้ และสามารถทำงานหลาย ๆ อย่างได้ในวงจรการทำงานเดียวกัน

ในรายงานฉบับนี้จะเน้นที่หุ่นยนต์ในโรงงานอุตสาหกรรม (Industrial Robotics) ดังนั้นคำว่า หุ่นยนต์หรือRobotics ที่กล่าวถึงจะหมายถึงหุ่นยนต์ในโรงงานอุตสาหกรรมเท่านั้น

บทที่ 2

แขนกล

2.1 ชนิดของหุ่นยนต์

หุ่นยนต์แบ่งตามลักษณะการเคลื่อนที่และลักษณะของแขน แบ่งได้เป็น 4 ชนิดใหญ่ๆ คือ

1. Polar Configuration.
2. Cylindrical Configuration.
3. Cartesian Configuration.
4. Articulated Configuration.

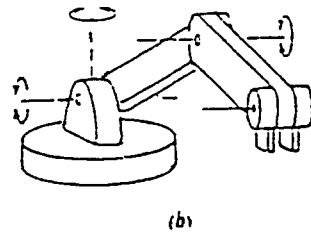
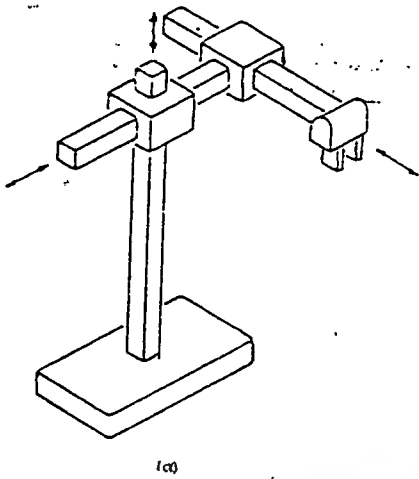
หุ่นยนต์ที่มีลักษณะเป็นแบบ โพลาร์ (Polar) นั้นลักษณะการเคลื่อนที่ของแขนจะสามารถยกขึ้นลงได้ในแนวตั้งโดยยกท่ามุมกับฐาน สามารถหมุนได้รอบตัว (รูป 2.1 a) พื้นที่การทำงานจะเป็นแบบทรงกลมดังนั้นในบางครั้งจึงเรียกว่า "Spherical coordinate"

หุ่นยนต์ที่มีลักษณะเป็นแบบ ไซเลนดริคัล (Cylindrical) (รูป 2.1 b) เคลื่อนที่ขึ้นข้างบนได้ตามแกนตั้งที่เป็นแกนหลัก และสามารถเคลื่อนที่ไปมาได้โดยใช้แกนนอน แกนตั้งสามารถที่จะหมุนได้ พื้นที่การทำงานจะเป็นแบบทรงกระบอก

หุ่นยนต์ที่มีลักษณะเป็นแบบ อาร์ติคูลเลต (Articulated) (รูป 2.1 c) ลักษณะการเคลื่อนที่จะมีแกน 3 แกน เป็นเหมือนแกน X,Y,Z ดังนั้นในบางครั้งจึงเรียกว่า หุ่นยนต์เรคติลิเนียร์ (Rectilinear Robot) พื้นที่การทำงานสามารถที่จะทำงานได้ในส่วนที่เป็นด้านของมือเพียงด้านเดียวเท่านั้นเพราะไม่มีการหมุนของฐาน

หุ่นยนต์ที่มีลักษณะเป็น อาร์ติคูลเลต หุ่นยนต์แบบนี้จะมีลักษณะใกล้เคียงกับแขนของมนุษย์มีข้อหมุนต่างๆเหมือนกัน (รูป 2.1 d) ดังนั้นพื้นที่การทำงานจึงสามารถที่จะทำงานได้ในทุกตำแหน่งในระยะความยาวของแขน

ในโครงการนี้ได้เลือกเอาแบบ อาร์ติคูลเลต เป็นต้นแบบในการทำทั้งนี้เพราะว่ามีลักษณะเหมือนแขนของมนุษย์ ซึ่งคนส่วนใหญ่ก็มีความคิดอยู่แล้วว่า หุ่นยนต์ต้องมีลักษณะเหมือนมนุษย์และอีกอย่างในการพัฒนาหุ่นยนต์ชนิด อาร์ติคูลเลตนี้สามารถที่จะพัฒนาไปสู่การทำงานได้ดีกว่าชนิดอื่น ๆ



รูปที่ 2.1 ชนิดของหุ่นยนต์ (a) คาร์ทีเซียน (b) อาร์ติคูลेट

ข้อดีข้อเสียของแต่ละชนิดนี้แตกต่างกันออกไป เพราะว่าลักษณะทางกายภาพแตกต่างกันแต่ถ้ามองในแง่ของการทำงานที่เป็นแบบซ้ำๆ ที่เดิมตลอด ชนิด คาร์ทีเซียน จะสามารถทำงานได้ดีกว่า (สามารถเคลื่อนที่ไปหาเป้าหมายโดยมีความผิดพลาดน้อยที่สุด แต่ถ้ามองในแง่การเข้าถึงวัตถุชนิดที่เป็นแบบ โพลาร์ และอาร์ติคูลेट จะสามารถเข้าถึงวัตถุได้ดีกว่าชนิดอื่นและชนิดที่เป็นแบบไซเลนดริคัล จะมีข้อดีในแง่ที่สามารถยกวัตถุได้มากกว่า ในงานทั่วไปแล้วใช้แบบ โพลาร์และ ไซเลนดริคัล เพราะ 2ชนิดนี้สามารถที่จะทำงานเป็นแบบ Load และ Unload โดยมีการเคลื่อนที่ของแขนออกไปในด้านข้างได้ดีกว่าชนิดอื่นๆ

2.2 มือของหุ่นยนต์ (Robot end effector)

มือจับของหุ่นยนต์คือตัวที่ทำหน้าที่ขั้นสุดท้ายของหุ่นยนต์มือจับของหุ่นยนต์สามารถแบ่งตามลักษณะของการทำงานได้ดังนี้

1. ทำงานเป็นตัวจับ (Gripper)
2. ทำหน้าที่เป็นเครื่องมือ (Tools)

มือของหุ่นยนต์ชนิดที่เป็นตัวจับมีหน้าที่โดยทั่วไปคือ การจับเบะการยกวัตถุ สามารถแบ่งออกอีกเป็น

2 ชนิดคือ Single gripper และ Double Gripper. Single gripper หมายถึง มือจับที่มีการเคลื่อนที่ของนิ้วเพียงข้างเดียวอีกข้างหนึ่งอยู่กับที่ และ Double gripper มีการเคลื่อนที่ของนิ้วทั้ง 2 ข้างเข้าหากัน

มือของหุ่นยนต์ชนิดที่เป็นตัวจับ สามารถแบ่งออกเป็นหลายๆ ชนิดตามลักษณะของตัวควบคุมได้ดังนี้

2.2.1 เมคคานิกกริปเปอร์ (Mechanical gripper)

เมคคานิกกริปเปอร์ คือมือจับที่มีกลไกทางเมคคานิกเป็นตัวควบคุมนิ้วบางครั้งเรียกว่าจอร์ส

(jaws) หน้าที่พื้นฐานของมือจับแบบเมคคานิก คือการส่งถ่ายกำลังจากมือ ไปสู่วัตถุต้นกำลังอาจใช้ไฟ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟ้า, นิวเมติกส์, เริงกล หรือ ไฮโดรลิก ลักษณะการจับของมือจับนี้ สามารถที่จะออกแบบการจับได้ 2 อย่างคือ

1. ใช้ลักษณะทางกายภาพ เช่น ต้องการจับวัตถุเป็นทรงกลมตรงปลายมือก็ทำเป็นส่วนโค้งที่มีขนาดพอเหมาะกับวัตถุนั้น (รูป 2.2)

2. ใช้ความเสียดทาน (รูป 2.3) นี้ไม่ได้ออกแบบให้มีลักษณะเหมือนกับรูปทรงของวัตถุแต่อาศัยแรงเสียดทานของนิ้วกับวัตถุ โดยมีสมการของแรงเสียดทานคือ

$$\mu n F_g = W \quad \text{-----(2.1)}$$

เมื่อ μ = สัมประสิทธิ์แรงเสียดทานนิ้วกับวัตถุ

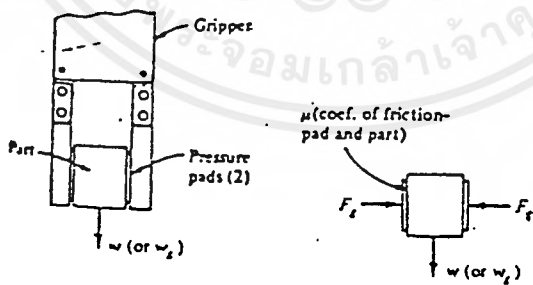
n = จำนวนนิ้วของมือจับ

F_g = แรงจับ

W = น้ำหนัก

สมการ 2.1 นี้ใช้เมื่อแรงโน้มถ่วงมีทิศทางขนานกันกับพื้นที่สัมผัส ถ้าทิศทางของแรงโน้มถ่วงเปลี่ยนไปจะใช้สมการ 2.2 คือ

$$\mu n F_g = mg \quad \text{-----(2.2)}$$



รูปที่ 2.3 มือจับที่อาศัยแรงเสียดทานของนิ้วกับวัตถุ

2.3 ความแม่นยำในการเคลื่อนที่

ความสำคัญอย่างหนึ่งของการสร้างหุ่นยนต์คือต้องการความแม่นยำในการทำงาน ความแม่นยำของหุ่นยนต์แต่ละตัวกับ ตัวแปร 3 ประการคือ

1. สเปเชียลเรซอลูชัน (Spatial resolution)
2. แอควเรซี (Accuracy)
3. รีพีทibility (Repeatability)

2.3.1 สเปเชียลเรซอลูชัน

คือช่วงการเคลื่อนที่ที่มีระยะทางสั้นที่สุดที่หุ่นยนต์แต่ละตัวสามารถที่จะทำได้ สเปเชียลเรซอลูชัน ขึ้นอยู่กับองค์ประกอบสำคัญ 2 ประการคือ

2.3.1.1 ระบบการควบคุม (Control system) ระบบการควบคุมนี้จะรวมถึงการวัดสัญญาณป้อนกลับของหุ่นยนต์ด้วย ช่วงของการเคลื่อนที่ที่ระบบควบคุมสามารถที่จะทำได้ขึ้นอยู่กับหน่วยความจำหลักของเครื่องคอมพิวเตอร์ที่มีหน่วยความจำ 8 bit จะสามารถที่จะแบ่งช่วงการเคลื่อนที่ได้เป็น 256 ช่วงคือช่วงของการเคลื่อนที่ที่คอมพิวเตอร์แบ่ง ได้มีค่าเท่ากับ 2 เมื่อ n คือหน่วยความจำหลักของเครื่อง

2.3.1.2 ความคลาดเคลื่อนเชิงกล (Mechanical inaccuracy) ความคลาดเคลื่อนเชิงกลของหุ่นยนต์แต่ละตัวขึ้นอยู่กับลักษณะของ ข้อหมุน (joint) และข้อต่อ (link) และระบบต้นกำลังของหุ่นยนต์นั้นด้วย

2.3.2 แอควเรซี

แอควเรซี คือตัวที่แสดงถึงความสามารถของหุ่นยนต์ในการที่เคลื่อนที่เข้าใกล้จุดเป้าหมายตามที่เรตั้ง แอควเรซี สามารถที่จะกำหนดให้อยู่ในเทอมของ สเปเชียลเรซอลูชัน ได้ทั้งนี้เพราะว่าการเคลื่อนที่ให้เข้าใกล้จุดเป้าหมาย ก็ต้องขึ้นอยู่กับช่วงของการเคลื่อนที่ที่มีความละเอียดมากน้อยเพียงใด ในการทำงานเราต้องวางจุดที่เราต้องการให้หุ่นยนต์ทำงานอยู่ระหว่างกลางของตำแหน่งการเคลื่อนที่ ของหุ่นยนต์ทั้งนี้เพราะว่าความคลาดเคลื่อนเชิงกลมีผลต่อความแม่นยำของหุ่นยนต์

ความแม่นยำของหุ่นยนต์กำหนดให้เท่ากับครึ่งหนึ่งของระยะทางการเคลื่อนที่ที่สั้นที่สุดของหุ่นยนต์ที่สามารถทำได้ความแม่นยำของหุ่นยนต์ ขึ้นอยู่กับองค์ประกอบเหล่านี้คือ

1. พื้นที่การทำงานของหุ่นยนต์ถ้าแขนทำงานในพื้นที่การทำงานจะมีความแม่นยำมากกว่าเมื่อแขนออกนอกพื้นที่การทำงาน
2. วงรอบการทำงาน ถ้าวงรอบการทำงานเป็นวงรอบที่แน่นอนความแม่นยำจะมีมากขึ้น
3. น้ำหนักที่ได้รับ ถ้าหุ่นยนต์ทำงาน โดยการรับน้ำหนักมากๆ ความแม่นยำจะลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 รีพีททีบิลิตี

รีพีททีบิลิตี คือความสามารถของหุ่นยนต์ในการกลับมาทำงานซ้ำที่เดิมจากรูป 2.1 จุดเป้าหมายคือจุด T แต่เนื่องจากข้อจำกัดของหุ่นยนต์ หุ่นยนต์ไม่สามารถที่จะทำงานที่จุด T ได้ ดังนั้นจึงทำงานที่จุด P กับจุด T คือตัวแสดงถึงความแม่นยำของหุ่นยนต์ แต่เมื่อสั่งให้หุ่นยนต์กลับมาทำงานที่จุด P อีกครั้งหนึ่งหุ่นยนต์ไม่สามารถกลับมาทำงานที่จุด P ได้ที่จุด R แทนระยะห่างระหว่างจุด P และจุด R คือค่าแสดงถึงรีพีททีบิลิตี



รูปที่ 2.4 ความแม่นยำและความสามารถในการทำซ้ำของหุ่นยนต์

2.4 การเคลื่อนที่ของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์สามารถแบ่งการเคลื่อนที่ออกได้ 2 ส่วนคือ

2.4.1 การเคลื่อนที่ของตัวและแขน (body and arm)

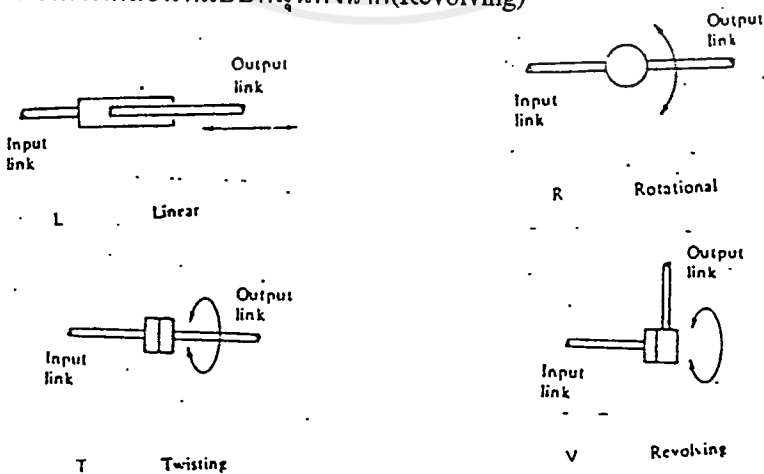
การเคลื่อนที่ของตัวและแขนแบ่งตามลักษณะของข้อหมุนได้ดังนี้คือ

2.4.1.1 การเคลื่อนที่แบบเส้นตรง (Linear)

2.4.1.2 การเคลื่อนที่แบบหมุนรอบจุดหมุน (Rotational)

2.4.1.3 การเคลื่อนที่แบบบิดรอบจุดหมุน (Twisting)

2.4.1.4 การเคลื่อนที่แบบหมุนตั้งฉาก (Revolving)



รูปที่ 2.5 ลักษณะการเคลื่อนที่ของแขนแบบต่างๆ

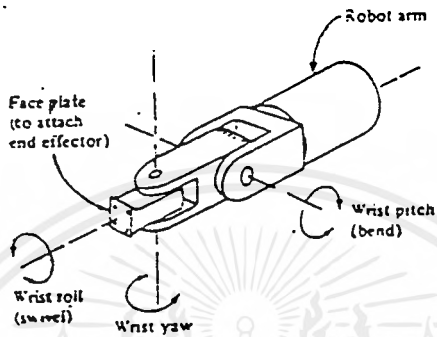
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การเคลื่อนที่ของมือ

2.4.2.1 หมุน(Roll)บางครั้งเรียกว่า Swivelข้อมือหมุนรอบแกนของแขน

2.4.2.2 บิด(Pitch)บางครั้งเรียกว่า Bend ข้อมือจะยกขึ้นลงในแนวตั้ง

2.4.2.3 สาย(Yaw) หมายถึงการบิดไปมาทางซ้ายและขวาของแกนมือ



รูปที่ 2.3 การเคลื่อนที่ของมือ

บทที่ 3

ดีซีมอเตอร์

3.1 หลักการทำงานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานสดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษคือแรงบิดของเพลลาของมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ แรงบิดของเพลลาดีซีมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ แรงบิดของเพลลาของดีซีมอเตอร์จะได้ผลระหว่างสนามแม่เหล็กกับขดลวดตัวนำ หลักการนี้แสดงได้ในรูป 3.1 ในที่นี้กระแสที่ไหลในขดลวดตัวนำจะสร้างฟิลด์ที่ประกอบด้วยเส้นแรงแม่เหล็ก และขดลวดตัวนำเหล่านั้นอยู่ห่างจากศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสเท่ากับ

$$T = K\phi I \quad (3-1)$$

เมื่อ T คือแรงบิดของเพลลา มีหน่วยเป็นนิวตัน-เมตร

ϕ คือเส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์

I คือกระแสเป็นแอมแปร์

K คือตัวคงที่

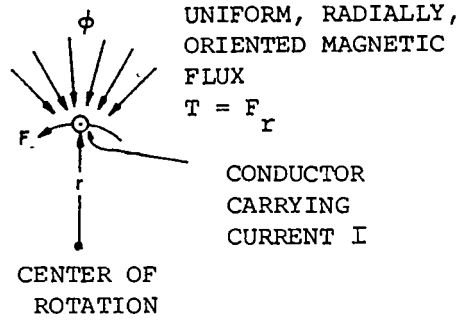
ดังนั้นแรงบิดของเพลลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแสและขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กจะทำให้เกิดโวลต์เด็งตกคร่อมตัวมันเอง โวลต์เด็งนี้จะเป็นส่วนสัดส่วนกับความเร็วของเพลลาของมอเตอร์และค่านการไหลของกระแส ความสัมพันธ์ระหว่างโวลต์เด็งย้อนกลับนี้และความเร็วของเพลลาของมอเตอร์คือ

$$E = K\phi\omega \quad (3-2)$$

เมื่อ E คือโวลต์เด็งย้อนกลับ emf มีหน่วยเป็นโวลต์

ϕ คือเส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์

ω คือความเร็วของมอเตอร์ มีหน่วยเป็นเรเดียน/วินาที



รูปที่ 3.2 แสดงถึงการเกิดแรงบิดในมอเตอร์

3.2 การแยกประเภทของดีซีมอเตอร์

ดีซีมอเตอร์สามารถแบ่งออกได้เป็นหลายประเภทขึ้นอยู่กับลักษณะวิธีการสร้างสนามแม่เหล็กของตัวมอเตอร์และขึ้นอยู่กับพื้นฐานการออกแบบ โครงสร้างของอาร์มาเจอร์การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกออกได้เป็น 2 แบบคือ

1. ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้
2. ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กมีค่าคงที่

ถ้าเราจะพิจารณาแยกประเภทตามลักษณะการออกแบบ โครงสร้างอาร์มาเจอร์สามารถแยกออกได้เป็น 3 แบบคือ

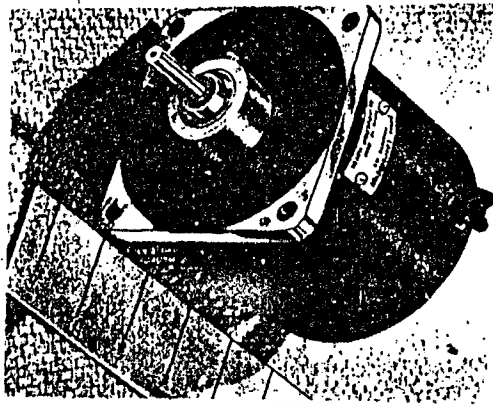
1. ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นแกนเหล็ก
2. ดีซีมอเตอร์แบบอาร์มาเจอร์ที่มีขดลวดพันอยู่บนผิว
3. ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นขดลวดหมุน

นอกจากนี้ยังมีดีซีมอเตอร์ชนิดพิเศษอีกแบบหนึ่งคือ แบบไม่มีแปรงถ่าน (brushless dc motor) ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับดีซีมอเตอร์ชนิดมีแปรงถ่านยกเว้น การคอมมิวเทชั่นกระทำโดยเทคนิคทางอิเล็กทรอนิกส์แทนที่จะกระทำโดยวิธีการทางเชิงกล

ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้

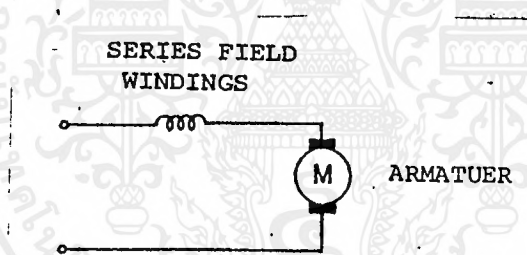
ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้ยังแบ่งออกได้เป็น 2 แบบคือ

- ก) แบบขดลวดสนามแม่เหล็กต่ออนุกรมกับขดลวดอาร์มาเจอร์
- ข) แบบขดลวดสนามแม่เหล็กแยกกระตุ่น

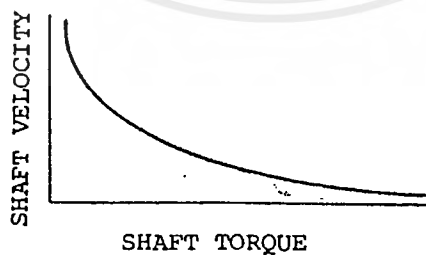


รูปที่ 3.4 เซอร์ โวมอเตอร์แบบ Drag cup

ตัวอย่างของมอเตอร์แบบขดลวดสนามแม่เหล็กต่ออนุกรมแสดงดังในรูป 3.5 มอเตอร์แบบนี้จะมีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแสที่ผ่านเส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ และเราจะ ได้ความสัมพันธ์ระหว่างความเร็วและแรงบิดเป็นนอนลิเนียร์ดังแสดงในรูป 3.6



รูปที่ 3.5 ซีซีมอเตอร์แบบอนุกรม

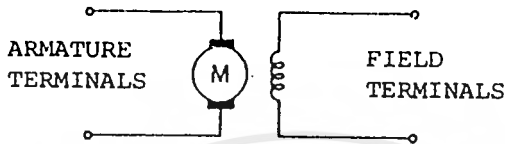


รูปที่ 3.6 คุณสมบัติระหว่างความเร็วและแรงบิด

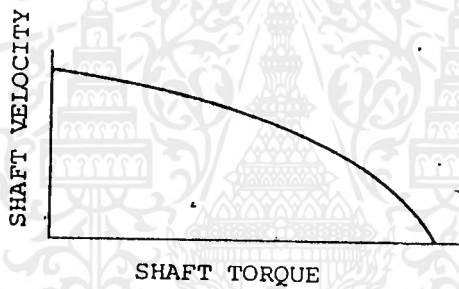
มอเตอร์ดังกล่าวจะใช้งานในภาวะเฉพาะเมื่อต้องการแรงบิดสูงที่ความเร็วต่ำ และแรงบิดที่มีความเร็วสูงเช่นระบบการขับเคลื่อนของรถลาก

ตัวอย่างของมอเตอร์แบบขดลวดสนามแม่เหล็กแยกกระตุ้นแสดงในรูป 3.7 ซีซีมอเตอร์แบบนี้มักนิยมเรียกกันว่ามอเตอร์ชาน (shunt motor) มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดที่เห็นเอกสารนี้โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างอิสระต่อกระแสของอาร์มาเจอร์ยังผลให้สามารถควบคุมพารามิเตอร์ของมอเตอร์ให้มีค่าคงที่ได้ตลอดช่วงพิสัยที่กว้าง มอเตอร์นี้มักจะใช้งานในกรณีระบบบังคับการเคลื่อนที่ที่ต้องการแรงบิดสูง ในรูป 2.8 แสดงถึงคุณสมบัติระหว่างแรงบิดกับความเร็วของชั้นท์มอเตอร์ภายใต้ภาวะการกระตุ้นสนามแม่เหล็กคงที่และอาร์มาเจอร์โวลต์เต็งคงที่



รูปที่ 3.7 ดีไซน์มอเตอร์แบบปรับสนามแม่เหล็กได้



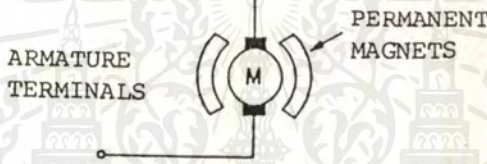
รูปที่ 3.8 คุณสมบัติระหว่างความเร็วและแรงบิด

ดีไซน์มอเตอร์แบบเส้นแรงแม่เหล็กคงที่

ระบบการกระตุ้นฟิลาท์ของมอเตอร์โดยทั่วไปในปัจจุบันมักใช้เป็นแบบ แม่เหล็กถาวร ดังแสดงในรูป 3.10 ในระบบนี้เส้นแรงของฟิลาท์มีค่าคงที่ดังนั้น อัตราส่วนระหว่างกระแสอาร์มาเจอร์และแรงบิดจะมีค่าคงที่ ดังนั้นสมการ (3-1) เขียนให้ง่ายขึ้นได้เป็น



รูปที่ 3.9 ฐานมอเตอร์แบบแม่เหล็กถาวร



รูปที่ 3.10 คีชีมอเตอร์แบบฟิลด์เป็นแม่เหล็กถาวร

$$T = K_f I \quad \text{-----(3-3)}$$

และสมการ (3-2) ได้เป็น

$$E = K_w \omega \quad \text{-----(3-4)}$$

ระบบนี้จะให้ความสัมพันธ์ระหว่าง กระแสอาร์มาเจอร์ แรงบิดและความเร็วอยู่ในลักษณะ
 ลิเนียร์ สมการทางไฟฟ้าของคีชีมอเตอร์แบบนี้เขียนได้เป็น

$$V = K_w \omega + L(di/dt) + R_i \quad \text{-----(3-5)}$$

เมื่อ V คือ โวลต์เตจที่ป้อนให้กับมอเตอร์

K_w คือค่าคงที่ของ โวลต์เตจย้อนกลับ

L คืออินดักเตนซ์ของอาร์มาเจอร์

R คือความต้านทานที่ขั้วของมอเตอร์

สมการไดนามิกของมอเตอร์คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_g = J(dw/dt) + Bw + T_r + T_L \quad \text{-----}(3-6)$$

เมื่อ T_g คือแรงบิดที่กำเนิดโดยมอเตอร์

G คือผลรวมของโมเมนต์ของแรงเฉื่อยของมอเตอร์และโหลด

B คือสัมประสิทธิ์ของวิสตอสแคมป์ฟิง

T_r คือแรงบิดเสียดทานภายใน

T_L คือแรงบิดโหลด

สมการต่าง ๆ ของมอเตอร์แบบแยกกระตุ่นฟิลด์จะเหมือนกับแบบกระตุ่นฟิลด์คงที่ อย่างไรก็ตาม ความข้อเสียของมอเตอร์แบบฟิลด์แม่เหล็กถาวรซึ่งเหนือกว่ามอเตอร์แบบมีโครงสร้างของฟิลด์ด้วยการพันของขดลวดคือ ไม่มีกำลังสูญเสียในฟิลด์ ประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเทียบกับมอเตอร์ที่มีขนาดของกำลังเท่ากัน

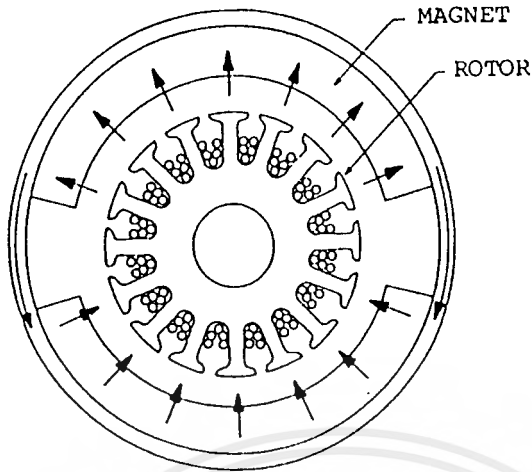
นอกจากนั้นความสัมพันธ์เชิงเส้นในสมการ (3-1) ยังให้ค่าของกระแสอาร์มาเจอร์ที่สูงกว่าดีซีมอเตอร์แบบฟิลด์เป็นขดลวด การประยุกต์ใช้งานเหมาะกับระบบที่ต้องการแรงบิดของโหลดสูง

ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นแกนเหล็ก

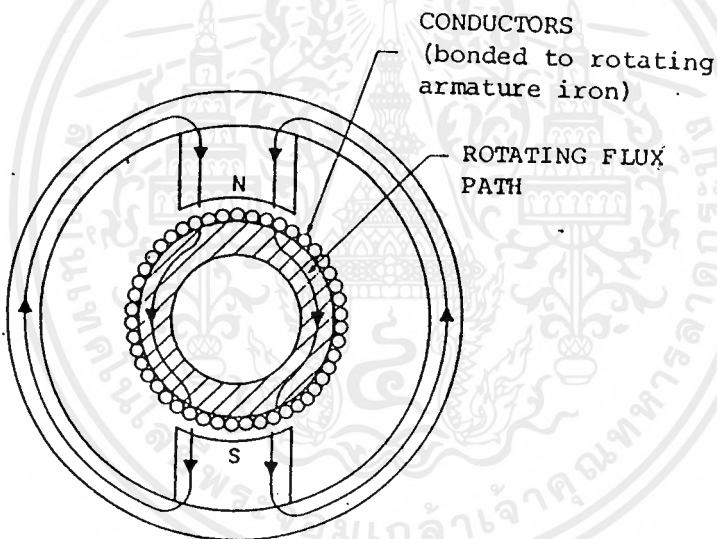
โครงสร้างของโรเตอร์และสเตเตอร์ของมอเตอร์แบบแกนเหล็กแสดงดังในรูป 3.11 โครงสร้างของมอเตอร์แบบนี้มีโมเมนต์ของแรงเฉื่อยสูงที่สุดและมีค่าอินดักแตนซ์ของโรเตอร์สูงที่สุดด้วย ดังนั้นมอเตอร์นี้จึงมีปริมาณการจุความร้อนได้สูง และสามารถชะทอนโอเวอร์โหลดได้ในระยะเวลาที่ยาวนานโดยไม่ทำให้มอเตอร์เสียหาย

ดีซีมอเตอร์แบบอาร์มาเจอร์มีขดลวดพันอยู่บนพื้นผิว

ในรูป 3.12 แสดงถึงการออกแบบของโรเตอร์ที่มีขดลวดพันอยู่บนพื้นผิว โดยไม่มีสลักทำให้ได้อินดักแตนซ์ของโรเตอร์ต่ำกว่าแบบแกนเหล็ก ข้อเสียคือ ทำให้ขนาดของมอเตอร์แบบนี้ใหญ่ขึ้นและราคาแพงกว่าแบบแกนเหล็กด้วย



รูปที่ 3.11 แสดงรูปหน้าตัดของคีมอเตอร์แบบอาร์เมเจอร์เป็นแกนเหล็กและฟิลด์เป็นแม่เหล็ก



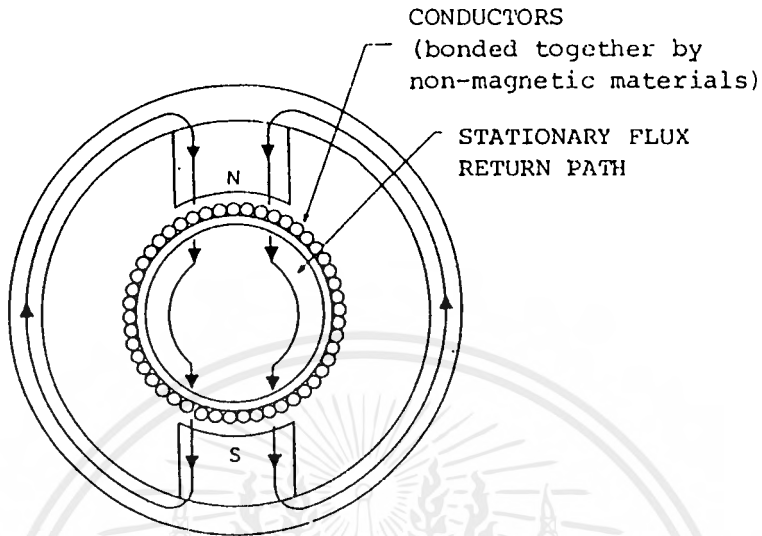
รูปที่ 3.12 แสดงรูปหน้าตัดของคีมอเตอร์แบบมีขลวดบนพื้นผิวและฟิลด์เป็นแม่เหล็กถาวร

คีมอเตอร์แบบอาร์มาเจอร์เป็นขลวดคหมุม

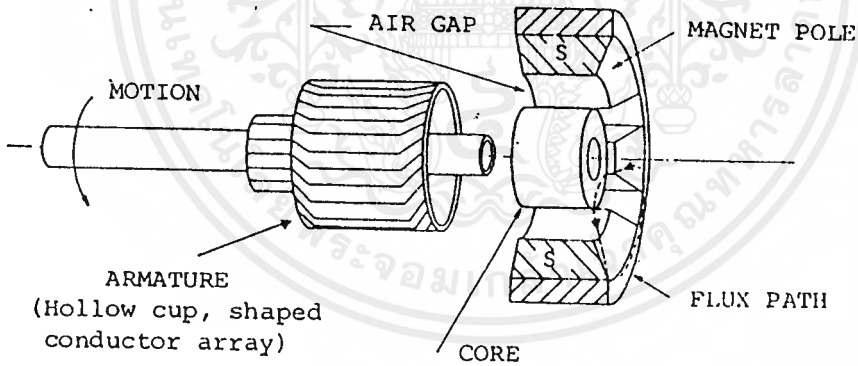
มอเตอร์แบบขลวดคหมุมนี้ได้รับการออกแบบเพื่อให้มีโมเมนต์ของแรงเฉื่อยน้อยมากดังแสดงในรูป 3.13 และรูปหน้าตัดซึ่งมองด้านข้างของมอเตอร์ดังกล่าวแสดงในรูป 3.14 รูปลักษณะโครงสร้างภายนอกและโครงสร้างของแม่เหล็ก มอเตอร์แบบนี้มีช่องว่างอากาศ (air - gap) ระหว่างแม่เหล็กมากกว่ามอเตอร์ทั้งสองแบบที่กล่าวมาแล้ว ดังนั้นจำเป็นต้องออกแบบให้โครงสร้างของแม่เหล็กให้ใหญ่ขึ้นเพื่อให้ได้ช่องว่างของอากาศระหว่างเส้นแรงแม่เหล็กที่เท่ากับของมอเตอร์ทั้งสองแบบดังกล่าว ดังนั้นราคาของมอเตอร์แบบนี้จึงมีราคาแพง นอกจากนั้น โครงสร้างของโรเตอร์มีความจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความร้อนต่ำมากถ้าหากเกิดโอเวอร์โหลดก็จะทำให้มอเตอร์เสียได้ง่ายและโรเตอร์ลักษณะนี้จะมีค่าอินดักแตนซ์ต่ำมากคือน้อยกว่า 10 ไมโครเฮนรี



รูปที่ 3.13 หน้าตัดแบบมีโรเตอร์เป็นขดลวดเคลื่อนที่และฟิลด์เป็นแม่เหล็กถาวร

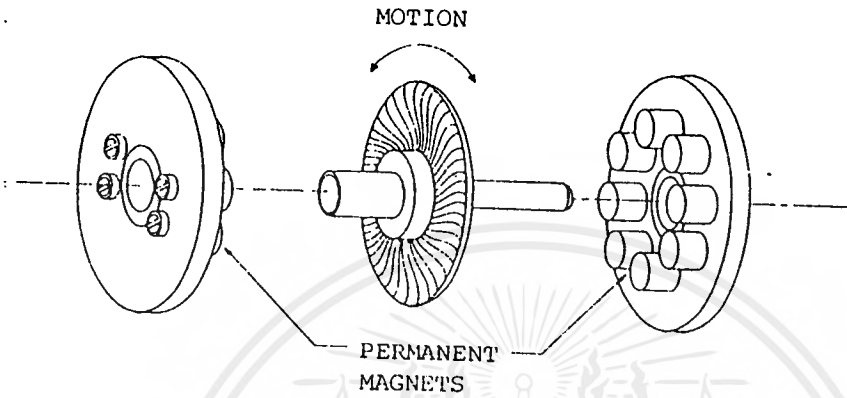


รูปที่ 3.14 รูปด้านข้าง

ในรูป 3.15 แสดงถึงมอเตอร์แบบขดลวดเคลื่อนที่อีกลักษณะหนึ่งซึ่งมีโครงสร้างของอาร์มาเจอร์เป็นรูปร่างงานซึ่งทำขึ้นจากขดลวดตัวนำซ้อนกันหลาย ๆ ชั้น ซึ่งเรามักจะเรียกกันว่า “printed motor” ตัวอย่างอีกอันหนึ่งของมอเตอร์แบบขดลวดเคลื่อนที่ (หมุน) แสดงในรูป 3.10 ซึ่งเป็นมอเตอร์ขนาด 7 แรงม้า 800 แอมป์ ในปัจจุบันคิซิมอเตอร์แบบโรเตอร์เป็นขดลวดหมุนนี้ให้คุณสมบัติการทำงานที่ดีเยี่ยมเหมาะสำหรับเป็นตัวขับเคลื่อนในระบบการบังคับตำแหน่ง และยังให้อัตรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนระหว่างแรงบิดและแรงเฉื่อยได้สูงและมีค่าอินดักเต้นซ์ต่ำที่สุดเมื่อเทียบกับมอเตอร์แบบอื่น ๆ นอกจากนั้นความสามารถในการเพิ่มอัตราเร่งยังกระทำได้สูง 10^6 เรเดียน/วินาที²



รูปที่ 3.15 แสดงถึงคิซิมอเตอร์แบบ โรเตอร์เป็นขดขวดหมุนมีรูปร่างเป็นจาน

บทที่ 4

การ อินเทอร์เน็ต

IBM PC กับการอินเทอร์เน็ต

ในปัจจุบัน คอมพิวเตอร์มีราคาถูกลงอย่างมาก มีการนำไปใช้งานกันอย่างกว้างขวาง ไม่ว่าจะเป็นในด้านที่เกี่ยวข้องกับเอกสารหรือ งานทั่ว ๆ ไป เช่น การทำเสียงเอฟเฟคใช้ประกอบภาพยนตร์หรือ การทำรูปในด้านการโฆษณา ฯลฯ การใช้งานในด้านต่าง ๆ เหล่านี้ มักจะมีอุปกรณ์ที่ใช้ประกอบการทำงาน ซึ่งเป็นตัวเชื่อมต่อระหว่าง และคอมพิวเตอร์ทำให้การใช้งานเป็นไปอย่างง่ายคดียิ่งขึ้น เช่น จอยสติค เม้าส์ปากเขียนจอย (Light Pen) เป็นต้น อุปกรณ์เหล่านี้จะต่อกับคอมพิวเตอร์โดยมีรูปแบบการติดต่อที่เป็นมาตรฐานที่นิยมใช้กันอยู่ทั่วไป คือ มาตรฐานการสื่อสารแบบอนุกรม (RS-232) และมาตรฐานการสื่อสารแบบขนาน (GPIB) ที่จำเป็นจะต้องมีมาตรฐานเดียวกันก็เพื่อให้อุปกรณ์นั้นสามารถใช้งานได้อย่างกว้างขวาง ไม่เฉพาะเจาะจงใช้กับเครื่องยี่ห้อใดยี่ห้อหนึ่งเท่านั้น ดังนั้นคอมพิวเตอร์ไม่ว่าจะเป็นเครื่องรุ่นใดก็ตามคนละยี่ห้อก็ตาม สามารถจะใช้อุปกรณ์เหล่านี้ได้ ขอให้มึรูปแบบการติดต่อที่เหมือนกันก็พอ

แต่อย่างไรก็ตามมาตรฐานเหล่านี้ก็จะมีข้อจำกัดอยู่บ้าง ดังนั้นสำหรับในบทความนี้ เราจะแนะนำการเชื่อมต่อ (การอินเทอร์เน็ต) กับ IBM PC โดย อุปกรณ์ I/O ที่ต้องการติดต่อโดยตรงกับระบบบัสของ IBM PC ซึ่งการเชื่อมต่ออุปกรณ์โดยตรงกับระบบบัสของคอมพิวเตอร์ เป็นการเพิ่มขีดความสามารถในการติดต่อและเพิ่มประสิทธิภาพในการทำงานของอุปกรณ์ 1/10 ที่ต่ออยู่กับระบบได้ดียิ่งขึ้น

AT สล็อต VS XT สล็อต

เนื่องจาก AT ใช้กับ CPU เบอร์ 80286 ซึ่งเป็น CPU ขนาด 16 บิตแท้คือ มีการประมวลผลภายในเป็น 16 บิต ในขณะที่ XT ใช้ CPU เบอร์ 8088 ซึ่งเป็น 16 บิต เทียม คือ มีการประมวลผลภายในเป็นแบบ 16 บิต แต่คำคำสั่งมีเพียงแค่ 8 บิตเท่านั้น ดังนั้นการจัดการเกี่ยวกับข้อมูลขนาด 16 บิต จึงต้องการทำสองครั้ง ครั้งละ 8 บิต จะเห็นได้ว่าระบบบัสของ XT เป็นขีดเขตของระบบบัสของ AT

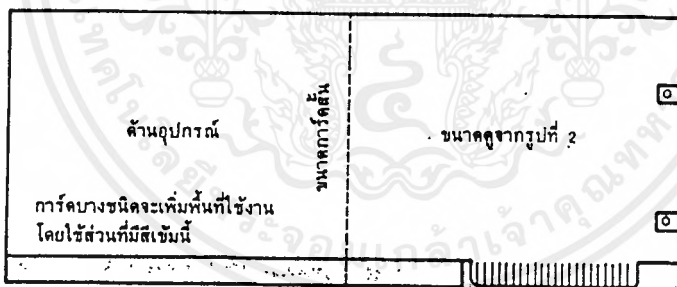
บนเมนบอร์ดของ AT จะมีสล็อตอยู่ 2 ชนิด คือ สล็อตสั้น และสล็อตยาว ซึ่งสล็อตยาวจะมีจำนวนขาสัญญาณ และตำแหน่งของขาสัญญาณบนสล็อตเหมือนกับ XT ส่วนขาสัญญาณที่อยู่บนสล็อตสั้นที่เพิ่มขึ้นจะเป็นสัญญาณที่มีแค่เฉพาะบน AT เท่านั้น ประกอบด้วย ข้อมูลครึ่งบน 8 บิต แอ็คเครสที่เพิ่มขึ้นมาอีก 4 บิต และ ขาสัญญาณควบคุมที่เพิ่มขึ้นจาก XT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

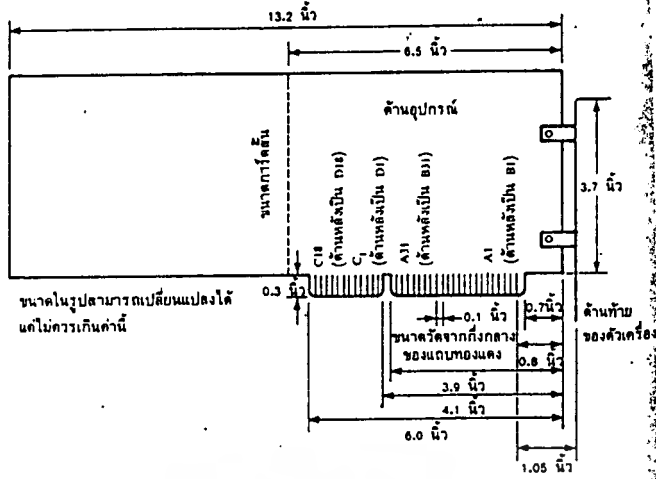
ขนาดและรูปร่างของการ์ดที่เสียบลงบนสล็อตของ XT จะคล้ายกับของ AT และเนื่องจากสัญญาณบนสล็อตจะมีตำแหน่งตรงกันด้วย การ์ดที่ใช้บน XT ทุกการ์ดสามารถนำมาใช้ได้บน AT โดยเสียบไว้บนสล็อตยาว อย่างไรก็ตามเราจะต้องคำนึงถึงอัตราการรับส่งข้อมูลของการ์ดนั้นด้วย เพราะความเร็วในการทำงานของ AT จะเร็วกว่า XT ดังนั้น เมื่อนำการ์ดของ XT มาใช้กับ AT การทำงานของ AT อาจจะช้าลงก็ได้ แต่ถ้าการทำงานของการ์ดนั้นไม่เกี่ยวข้องกับหน่วยความจำ หรือการทำ DMA เราก็ไม่จำเป็นต้องสนใจกับอัตราการรับส่งข้อมูลของมัน

4.1 การขยาย (Expansion Card)

รูปที่ 4.1 แสดงขนาดของการ์ดขยายแบบยาวสำหรับ XT และรูปที่ 2 แสดงขนาดของการ์ดขยายแบบยาวสำหรับ AT ส่วนขอบของการ์ดขยาย XT บริเวณที่มีสีเข้มในรูปที่ 1 เป็นพื้นที่ที่สามารถใช้งานได้ แต่ไม่นิยมทั้งนี้เพราะเมื่อนำไปใช้กับรุ่น AT จะเสียบไม่ได้ (ติดสล็อตต้นของ AT) เส้นประที่อยู่ในรูปแสดงขนาดของการ์ดขยายแบบสั้น ซึ่งการ์ดแบบนี้นิยมใช้กับวงจรรินเตอร์เฟสที่มีจำนวนอุปกรณ์น้อย ๆ



รูปที่ 4.1 ลักษณะของการ์ดขยายรุ่น XT



รูปที่ 4.2 ลักษณะของการ์ดขยายรุ่น AT

สล็อตแต่ละอันบนเมนบอร์ดระยะห่างกัน 0.8 นิ้ว ดังนั้นความหนาของแผ่นการ์ดรวมทั้งตัวอุปกรณ์จึงไม่ควรเกินค่านี้ ตามหนังสือ Technical Reference ของ IBM แนะนำว่าความหนาของแผ่นการ์ดรวมทั้งแบบอุปกรณ์บนแผ่นการ์ดไม่ควรเกิน 12.7 มิลลิเมตร หรือ 0.5 นิ้ว

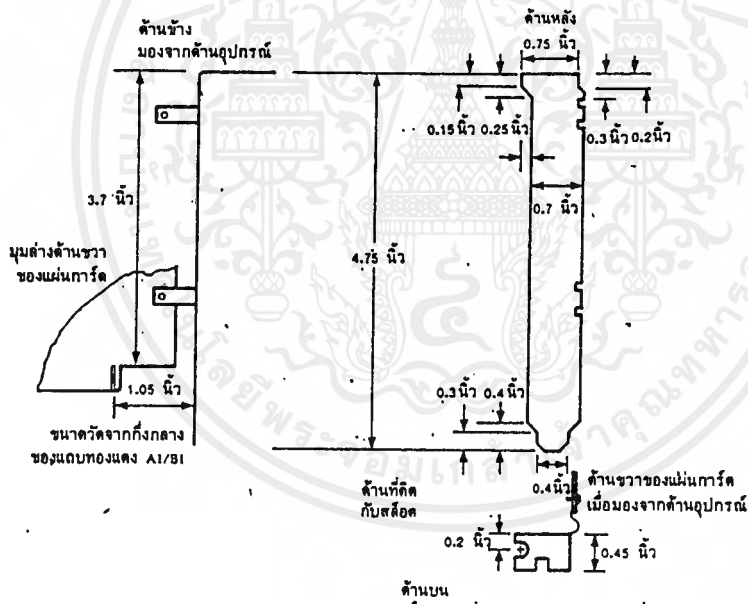
ด้านปลายสุดของการ์ดขยายจะต่ออยู่กับแผ่นเหล็ก มีลักษณะดังในรูปที่ 3 ถ้าการ์ดขยายมีคอนเน็กเตอร์ที่ใช้ต่อกับอุปกรณ์ภายนอก ก็จะติดตัวคอนเน็กเตอร์นั้นไว้บนแผ่นเหล็กนี้ เช่นพรีนเทอ์การ์ด หรือ การ์ด RS-232 ก่อนเสียบการ์ดบนสล็อต เราควรถอดแผ่นเหล็กกันฝุ่นที่อยู่ด้านหลังของตัวเครื่อง ที่ตรงกับตำแหน่งของสล็อตออกเสียก่อน เมื่อเสียบการ์ดแล้ว แผ่นเหล็กนี้จะทำหน้าที่เป็นที่ยึดของตัวคอนเน็กเตอร์ และกันฝุ่นด้วย แต่ถ้าแผ่นการ์ดไม่มีส่วนที่ต่อกับอุปกรณ์ภายนอก เช่น การ์ดขยายหน่วยความจำไม่จำเป็นต้องมีแผ่นเหล็กนี้ก็ได้ รูปที่ 3 แสดงขนาด และรูปร่างลักษณะของแผ่นเหล็กนี้

การ์ดขยายทำหน้าที่เป็นคอนเน็กเตอร์ตัวผู้ในตัว ขณะที่สล็อตเป็นคอนเน็กเตอร์ตัวเมีย เมื่อเสียบการ์ดขยายลงในสล็อต ตัวหนีบในสล็อตจะสัมผัสกับสายทองแดงที่สัมผัสกับตัวหนีบควรจะขูบทองเพื่อให้หน้าไฟฟ้าได้ดีขึ้นและไม่สกปรกง่าย แต่เพื่อความประหยัดถ้าการ์ดที่ใช้ไม่จำเป็นต้องถอดเข้าถอดออกบ่อย ๆ ใช้คิบูขบก็พอ ปัญหาอีกอย่างในการอินเตอร์เฟสผ่านสล็อตก็คือ การ์ดที่ใช้จะต้องเป็นแผ่น PCB แบบสองหน้าเสมอ ไม่ว่าจะจริงจะมีขนาดเล็กหรือใหญ่เท่าใด แต่เราแก้ปัญหาได้โดยใช้แผ่น PCB สองหน้า ทำเป็นคอนเน็กเตอร์ ถ้าผู้ใช้เสียบในสล็อต และเดินสายทองแดงเหมือนกับคอนเน็กเตอร์ของการ์ดด้วย ดังแสดงในรูปที่ 4 สายทองแดงด้านอุปกรณ์จำนวน 31 เส้น (A1-A31) ก็เดินออกมาที่ปลายด้านตรงข้ามกับด้านที่เป็นคอนเน็กเตอร์ ส่วนสายทองแดงด้านบัคกรี (B1-B31) ทำการทูลงมาด้านอุปกรณ์ แล้วเดินสายไปในทางเดียวกับสายทองแดงด้านอุปกรณ์ โดยให้แนวของสายด้านอุปกรณ์อยู่สูงกว่าแนวของสายด้านบัคกรี ระยะห่างระหว่างสายทองแดงบริเวณคอนเน็กเตอร์ ให้มีระยะห่างเท่ากับ 0.1 นิ้ว ตรงปลายของสายทองแดงที่เดินออกมาจากบริเวณ คอน

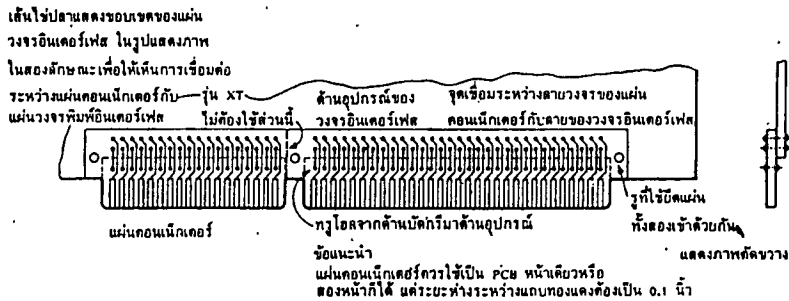
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใช้ได้เห็นประโยชน์ของการนำเอกสารนี้ไปใช้ ไม่ว่าจะวิธีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เน็ทเคอร์ทั้งหมดเจาะรูไว้เพื่อใช้เป็นตัวเชื่อมกับแผ่นวงจรที่ต้องการอินเทอร์เฟส การเดินลายทองแดงของแผ่นวงจรที่ต้องการอินเทอร์เฟส ก็ให้เดิน ในลักษณะเดียวกัน คือ ให้แนวของลายด้านอุปกรณอยู่สูงกว่า แนวลายด้านบัดกรี การต่อกันทำโดยนำแผ่นวงจรทั้งสองวางมาทับกัน โดยรูของแนวแต่ละด้านตรงกัน แล้วใช้ลวดทองแดงหรือขาอุปกรณ์เสียบสอดรู แล้วบัดกรีหัวท้ายให้เรียบร้อย วิธีนี้จะช่วยลดความยุ่งยากในการทำแผ่นวงจรพิมพ์ของวงจรอินเทอร์เฟสลงได้มาก

การควบคุมอุปกรณ์ I/O ที่ต่ออยู่กับ IBM PC จะกระทำผ่านพอร์ต โดยการอ้างถึงแอดเดรสของพอร์ตที่อุปกรณ์นั้นต่ออยู่โดยตรง ดังนั้น การที่จะใช้งานหรือควบคุมอุปกรณ์เหล่านี้ จึงจำเป็นคือ วงศีกษาถึงวิธีการควบคุมพอร์ต ใน IBM PC พอร์ตและหน่วยความจำจะแยกจากกันโดยเด็ดขาด ถึงแม้ว่าการอ้างถึงจะใช้สัญญาณจากแอดเดรสบัสเหมือนกันก็ตาม แต่สัญญาณที่ใช้ในการอินเบิ้ลในการอ่านและเขียนข้อมูลจะต่างกันก็จะต่างกัน ดังนั้นการติดต่อกับพอร์ตจึงมีคำสั่งแยกต่างหากออกจากคำสั่งที่ใช้ติดต่อกับหน่วยความจำ คือ IN และ OUT ด้วยเหตุนี้ การจัดแอดเดรสของหน่วยความจำและแอดเดรสของพอร์ต I/O จึงแยกออกจากกัน ใน IBM PC



รูปที่ 4.3 ขนาดของแผ่นเหล็กด้านหลังของการ์ดขยาย



รูปที่ 4.4 ลักษณะของแผ่น PCB ใช้แทนคอนเนกเตอร์ของการ์ดขยายและการเดินสายวงจร

ตารางที่ 4.1 แสดงการจัดแอดเดรสของหน่วยความจำบน IBM

Block 0	00000-0FFFF	RAM to 64K
Block 1	10000-1FFFF	RAM to 128K
Block 2	20000-3FFFF	RAM to 192K
Block 3	30000-3FFFF	RAM to 256K
Block 4	40000-4FFFF	RAM to 320K
Block 5	50000-5FFFF	RAM to 384K
Block 6	60000-6FFFF	RAM to 448K
Block 7	70000-7FFFF	RAM to 512K
Block 8	80000-8FFFF	RAM to 576K
Block 9	90000-9FFFF	RAM to 640K
Block A	A0000-AFFFF	Extended video memory
Block B	B0000-BFFFF	Standard video memory
Block C	C0000-CFFFF	BIOS extension (eg EGA)
Block D	D0000-DFFFF	Other use
Block E	E0000-EFFFF	Other use
Block F	F0000-FFFFF	BIOS EPROM

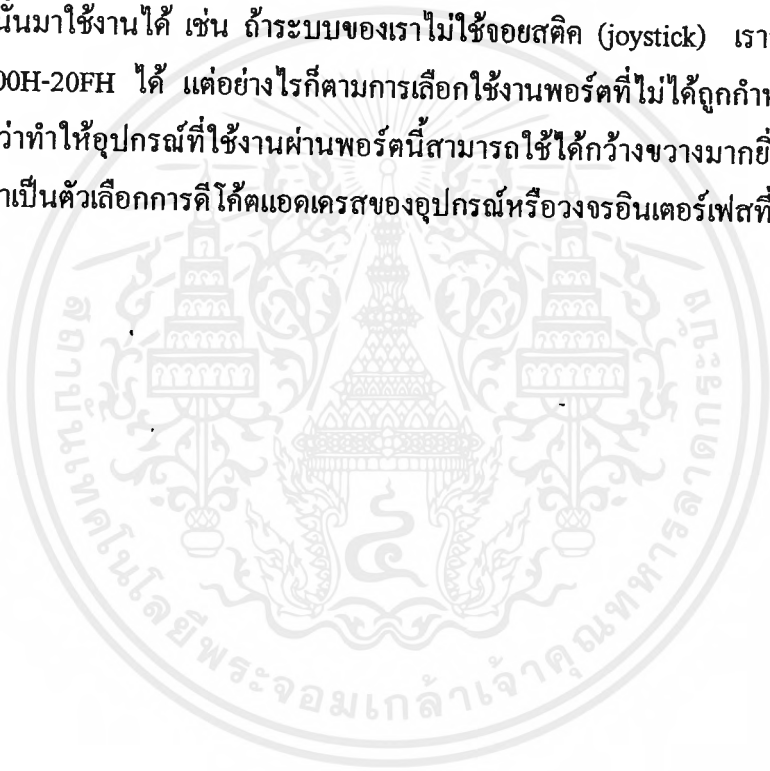
การจัดแอดเดรสของหน่วยความจำแสดงได้ในตารางที่ 1 และในตารางที่ 2 แสดงการจัดแอดเดรสของพอร์ต I/O

การอ้างแอดเดรสของหน่วยความจำจะใช้แอดเดรสทั้งหมด 20 เส้น คือ AO-A19 ในรุ่น XT (8088) แต่ในรุ่น AT (80286) ใช้แอดเดรส 24 เส้น คือ AO-A19 และ A20-A23 การอ้างแอดเดรสของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตสำหรับ CPU เบอร์ 8088 และ 80286 สามารถเข้าถึงได้ถึง 64K พอร์ตแต่ใน IBM PC ทั้งในรุ่น AT และ XT ออกแบบให้ใช้แอดเดรสเพียง 10 เส้น เท่านั้น คือ A0-A9 ดังนั้นจำนวนพอร์ตสูงสุดที่สามารถเข้าถึงได้คือ 1024 พอร์ตและในจำนวนทั้งหมดนี้ยังแบ่งออกเป็น 2 กลุ่ม คือกลุ่มพอร์ตที่มีแอดเดรสอยู่ในช่วง 000H-0FFH จะใช้งานบนเมนบอร์ดสำหรับชิพพอร์ตนั่น เช่น 8259 (Interrupt Controller), 8237 (DMA Controller), 8253 (Timer & Counter) และกลุ่มที่มีแอดเดรสอยู่ในช่วง 100H-3FFH จะใช้งานกับการ์ดขยายต่าง ๆ ที่เสียบในสล็อต

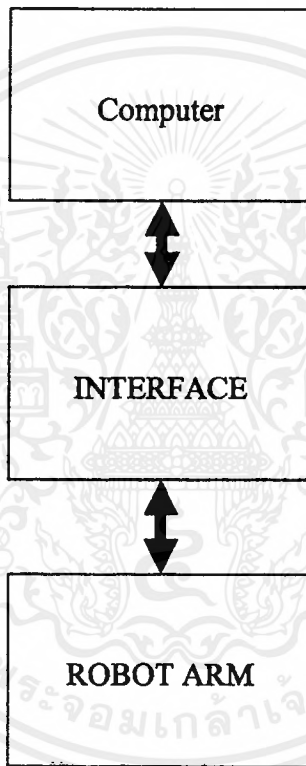
จากตารางที่ 2 จะเห็นว่าแอดเดรสของพอร์ตถูกแบ่งออกเป็นช่วงย่อย ๆ ซึ่งจะกำหนดไว้ให้ใช้กับอุปกรณ์ I/O เฉพาะอย่างถ้าในระบบของเราไม่ได้ใช้งานอุปกรณ์นั้นเราสามารถนำแอดเดรสของพอร์ตในช่วงนั้นมาใช้งานได้ เช่น ถ้าระบบของเราไม่ใช่จอยสติค (joystick) เราสามารถใช้งานพอร์ตในช่วง 200H-20FH ได้ แต่อย่างไรก็ตามการเลือกใช้งานพอร์ตที่ไม่ได้ถูกกำหนด ให้ใช้กับอุปกรณ์อื่นจะคิดว่าจะทำให้อุปกรณ์ที่ใช้งานผ่านพอร์ตนี้สามารถใช้ได้กว้างขวางมากยิ่งขึ้นถ้าจะให้คิดควรใช้คิปสวิทช์ทำเป็นตัวเลือกการดีโค้ดแอดเดรสของอุปกรณ์หรือวงจรถอนอินเทอร์เฟสที่เราสร้างขึ้น



บทที่ 5

โครงสร้างและการออกแบบ

5.1 โครงสร้างของโครงการ



รูปที่ 5.1 บล็อกไดอะแกรมของโครงการ

คอมพิวเตอร์ใช้สำหรับส่งค่าข้อมูลที่ใช้ในการควบคุมแขนกลไปยังระบบอินเตอร์เฟส

และรับค่าป้อนกลับจากระบบอินเตอร์เฟส กลับเข้ามาทำการแสดงผลการทำงานของแขนกล

อินเตอร์เฟส ทำหน้าที่รับค่าข้อมูลจากคอมพิวเตอร์ แล้วส่ง ไปยังมอเตอร์ต่าง ๆ ของแขนกล

เพื่อให้แขนกลเคลื่อนที่ไปยังตำแหน่งที่ต้องการ และรับค่าป้อนกลับจากแขนกลมาส่งให้คอมพิวเตอร์

เพื่อทำการประมวลผล

แขนกลจะเคลื่อนที่ตามคำสั่งของคอมพิวเตอร์ โดยใช้ ดีซีมอเตอร์ควบคุมการเคลื่อนไหวทั้งหมด 5 ตัว และจะใช้เซ็นเซอร์ส่งค่ากลับไปยังคอมพิวเตอร์ เพื่อให้ได้ตำแหน่งที่ต้องการ

5.2 ส่วนประกอบของโครงการ

โครงการจะประกอบด้วย

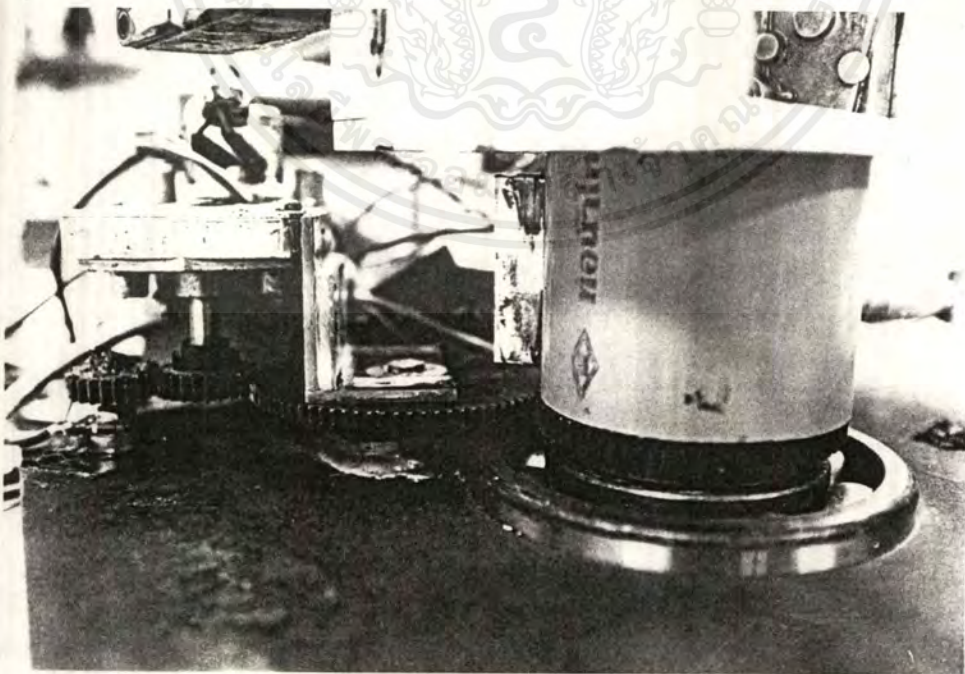
5.2.1 ส่วนคอมพิวเตอร์ ซึ่งจะมีหน้าที่ประมวลผลในการรับส่งข้อมูลจากอินพุท โดยจะส่งข้อมูลผ่านการ์ดอินเตอร์เฟสไปสู่แขนกล และรับจากแขนกลมาสู่คอมพิวเตอร์

5.2.2 ส่วนไดร์มอเตอร์ ซึ่งจะมีหน้าที่ควบคุมมอเตอร์ให้ทำงานตามคำสั่งของคอมพิวเตอร์ ในการควบคุมจะใช้พัลส์วิทมอดแอมป์ไฟต์ควบคุมการจ่ายกระแสเพื่อควบคุมความเร็วมอเตอร์

5.2.3 ส่วนเซ็นเซอร์ ซึ่งจะใช้โพเทนทิโอมิเตอร์ติดตั้งตามส่วนต่าง ๆ ของแขนกลเพื่อเป็นค่าบอกตำแหน่งของแขนกลและส่งให้คอมพิวเตอร์นำไปประมวลผล

5.2.4 ส่วนแขนกล ซึ่งใช้ในโครงการนี้จะเป็นแขนกลแบบอาร์ติคูลेट ซึ่งจะควบคุมการเคลื่อนที่จากดีซีมอเตอร์ 4 ตัว โดยแต่ละตัวจะทำหน้าที่ควบคุมการเคลื่อนที่ในแต่ละส่วน คือ

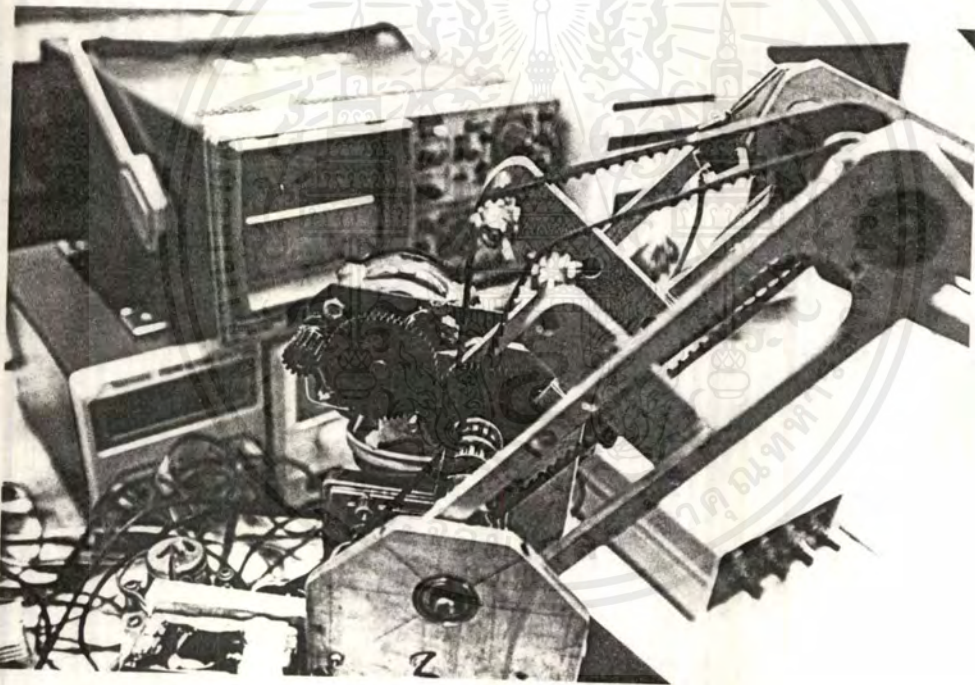
1. ส่วนฐาน (Base Part) ประกอบด้วยฟันเฟืองซึ่งมีระยะพิท 0.2 มิลลิเมตร เป็นตัวส่งผ่านกำลังจากมอเตอร์ไปหมุนแขนกลและจะติดตั้งเซ็นเซอร์เพื่อบอกตำแหน่งที่หมุนไปดังรูปที่ 5.2



รูปที่ 5.2 แสดงส่วนฐานของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

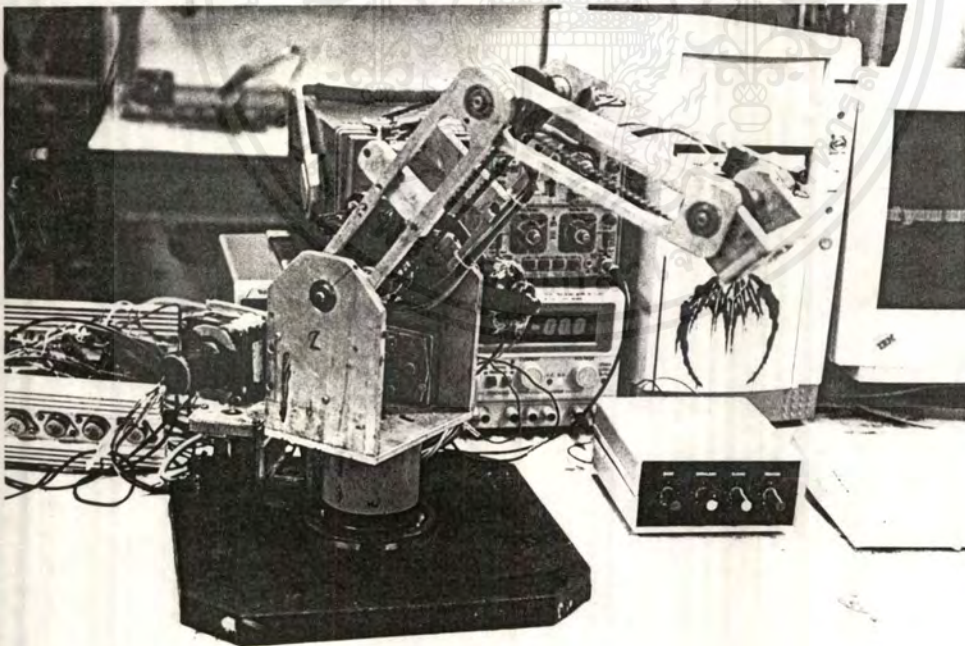
2. ส่วนหัวไหล่ (Shoulder Part) ประกอบด้วยฟีนเฟืองซึ่งมีระยะพิท 0.2 มิลลิเมตร เป็นตัวส่งผ่านกำลังจากมอเตอร์ไปยกแขนกลขึ้นลง โดยจะติดตั้งเซนเซอร์เพื่อบอกตำแหน่งดังรูปที่ 5.3
3. ส่วนข้อศอก (Elbow Part) จะประกอบด้วยสายพานแบบซิงโครนัส (synchronous belt) และพูลเลย์ ซึ่งจะช่วยทดแรงจากมอเตอร์ส่งไปยังแขนส่วนบน โดยจะติดตั้งเซนเซอร์เพื่อบอกตำแหน่งของข้อศอก ดังรูปที่ 5.4
4. ส่วนข้อมือ (Hand - Connector Part) จะประกอบด้วยสายพานแบบซิงโครนัส (synchronous belt) และพูลเลย์ ซึ่งจะช่วยทดแรงจากมอเตอร์ส่งไปยังแขนส่วนบน โดยจะติดตั้งเซนเซอร์เพื่อบอกตำแหน่งของข้อศอก ดังรูปที่ 5.5



รูปที่ 5.3 แสดงส่วนหัวไหล่



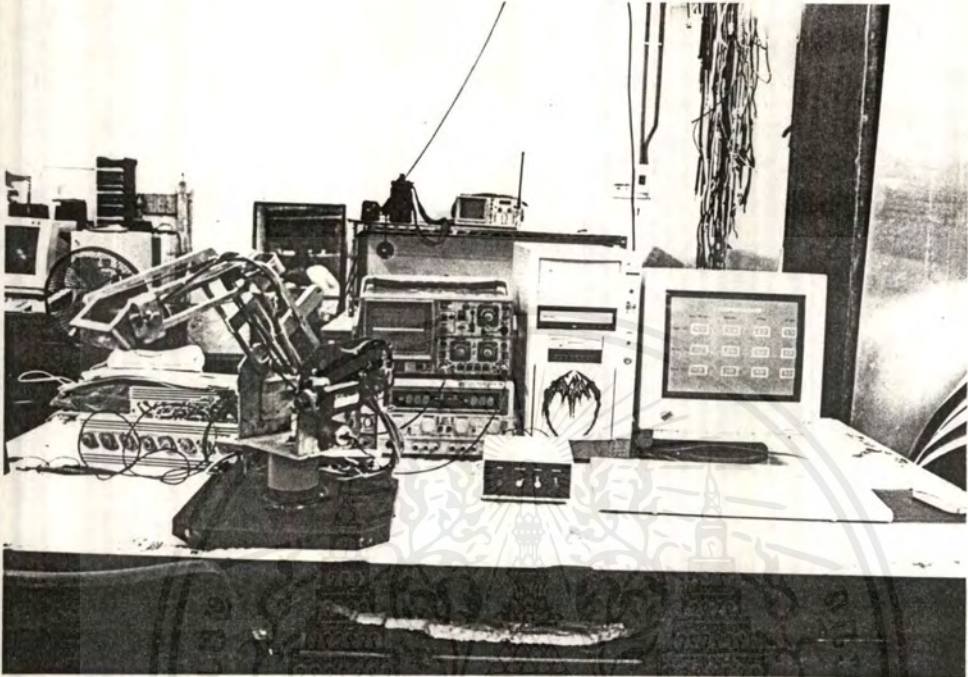
รูปที่ 5.4 แสดงส่วนข้อศอก



รูปที่ 5.5 แสดงส่วนข้อมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำส่วนต่าง ๆ มาประกอบกันจะทำให้แขนกลสามารถทำงานได้ดังรูปที่ 5.6



รูปที่ 5.6 แขนกล

5.3 การออกแบบส่วนไดร์มอเตอร์

เนื่องจากใช้ดีซีมอเตอร์ ทำให้ใช้กระแสมากโดยเฉพาะในช่วงการเริ่มต้นการทำงาน เพราะฉะนั้นจึงต้องมีวงจรที่ทำหน้าที่จ่ายกระแสให้กับมอเตอร์ โดยแต่ละส่วนจะแยกการทำงานของมอเตอร์แต่ละตัว เพื่อให้การทำงานของแขนกลจากคำสั่งของคอมพิวเตอร์สามารถทำได้โดยง่าย และในการเคลื่อนที่ของมอเตอร์ในช่วงที่ใช้กระแสต่างกันจะมีความเร็วในการเคลื่อนที่ต่างกันทำให้ต้องใช้พัลส์วิดท์มอดคูลเลชันแอมพลิไฟร์เข้ามาช่วยในการไดร์มอเตอร์เพื่อควบคุมความเร็วของมอเตอร์โดยจะมีวงจรดังรูป



5.4 การออกแบบชุดอินเตอร์เฟส

ในโครงการนี้จะทำการอินเตอร์เฟส โดยผ่านการ์คอินเตอร์เฟส ซึ่งฮาร์ดแวร์กำหนดตำแหน่ง โดยโฟโต้ไทป์ (Phototype) ไว้ที่ 300 H โดยใช้ IC 7430 nand gate 8 อินพุตเป็นตัวกำหนดตำแหน่ง โดยที่เอาท์พุตต่อกับ Chip Select ของ IC 8255 เพื่อเป็นตัวอ้างพอร์ตอินพุตเอาท์พุต โดยกำหนดให้ พอร์ต A เป็นเอาท์พุต เพื่อส่งข้อมูลไปยังชุดไคร้มอเตอร์ที่ตำแหน่งต่าง ๆ เพื่อควบคุมให้ทำงานตามที่กำหนด

พอร์ต B ทำหน้าที่เป็นอินพุตรับสัญญาณป้อนกลับที่มาจากเซนเซอร์ที่ตำแหน่งต่าง ๆ เข้ามา โดยแปลงสัญญาณอนาลอกให้เป็นดิจิทัลผ่าน IC ADC 0809 เข้ามายังพอร์ต B

พอร์ต C เป็นทั้งอินพุตและเอาท์พุตเพื่อควบคุมการเอกไซม์มอเตอร์แต่ละตัวและควบคุม สัญญาณ EOC ของ ADC 0809



อธิบายการทำงานของวงจร

จากวงจรจะกำหนด PA0-PA7 เป็น พอร์ตเอาต์พุต ส่งไปควบคุมจุด ไคร์ฟมอเตอร์ทั้ง 4 ตัว โดยมอเตอร์แต่ละตัว ควบคุมการทำงานโดยใช้บิตควบคุม 2 บิต คือ ควบคุมสวิทช์ เปิด-ปิด และการ พอร์เวอร์ค/รีเวอร์ส ของมอเตอร์ ซึ่งสามารถกำหนดบิต ควบคุมมอเตอร์แต่ละตัว ได้ดังนี้

Motor	Forward/Reverse				On/Off				Bit
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	Control
SHOULDER	0	0	0	1	0	0	0	1	11H
	0	0	0	0	0	0	0	1	01H
ELBOW	0	0	1	0	0	0	1	0	22H
	0	0	0	0	0	0	1	0	02H
BASE	0	1	0	0	0	1	0	0	44H
	0	0	0	0	0	1	0	0	04H
GRIPPER	1	0	0	0	1	0	0	0	88H
	0	0	0	0	1	0	0	0	08H

PB0-PB7 เป็นพอร์ตอินพุท โดยจะต่อกับ D0-D7 ของ ADC 0809 สำหรับ ADC 0809, อนุ ล็อก ทู คิจิตอล คอนเวอร์เตอร์ 8 อินพุท นั้นจะรับสัญญาณอินพุท ทั้ง 8 ซึ่งมาจาก อินพุท โวลท์เตจ 4 อินพุท (มาจากการปรับ โวลุ่ม) และอินพุท โวลท์เตจ ฟีดแบ็ค อีก 4 อินพุท ซึ่งมาจากมอเตอร์แต่ละตัว โดยจะกำหนดให้

INO	ของ	ADC	0809	เป็น	input voltage	ของ	Base Motor
IN1	ของ	ADC	0809	เป็น	input voltage feedback	ของ	Base Motor
IN2	ของ	ADC	0809	เป็น	input voltage	ของ	Gripper Motor
IN3	ของ	ADC	0809	เป็น	input voltage feedback	ของ	Gripper Motor
IN4	ของ	ADC	0809	เป็น	input voltage	ของ	Shoulder Motor
IN5	ของ	ADC	0809	เป็น	input voltage feedback	ของ	Shoulder Motor
IN6	ของ	ADC	0809	เป็น	input voltage	ของ	Elbow Motor
IN7	ของ	ADC	0809	เป็น	input voltage feedback	ของ	Elbow Motor

PC0-PC7 จะกำหนดให้ PC0-PC3 เป็นอินพุท

PC4-PC7 เป็นเอาต์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC0-PC3 เป็นพอร์ต อินพุทจะต่อกับสวิทช์ ทั้ง 4 ตัว ซึ่งจะไปต่อกับมอเตอร์แต่ละตัว โดยกำหนดให้มีสถานะเป็น 1 เพื่อเป็นตัวกำหนดค่าเริ่มต้นของมอเตอร์แต่ละตัวซึ่งค่าเริ่มต้นก็คือ ค่าที่อินพุท โวลต์เตจ ฟีดแบ็ค เป็น 0 นั่นเอง โดยเมื่อเริ่มต้น อินทิเกรตโปรแกรม มอเตอร์จะทำให้กลไกเคลื่อนที่ไปสัมผัส สวิทช์ ซึ่งจะทำให้สวิทช์ เบ็ดวงจรลงกราวด์มีสถานะเป็น 0 โปรแกรมก็จะรับรู้ค่าเริ่มต้น แล้วเคลื่อนที่ไปหาค่ากลาง (มิดเดิล แวลู) แล้วจึงเริ่มเข้าสู่ เมน โปรแกรม

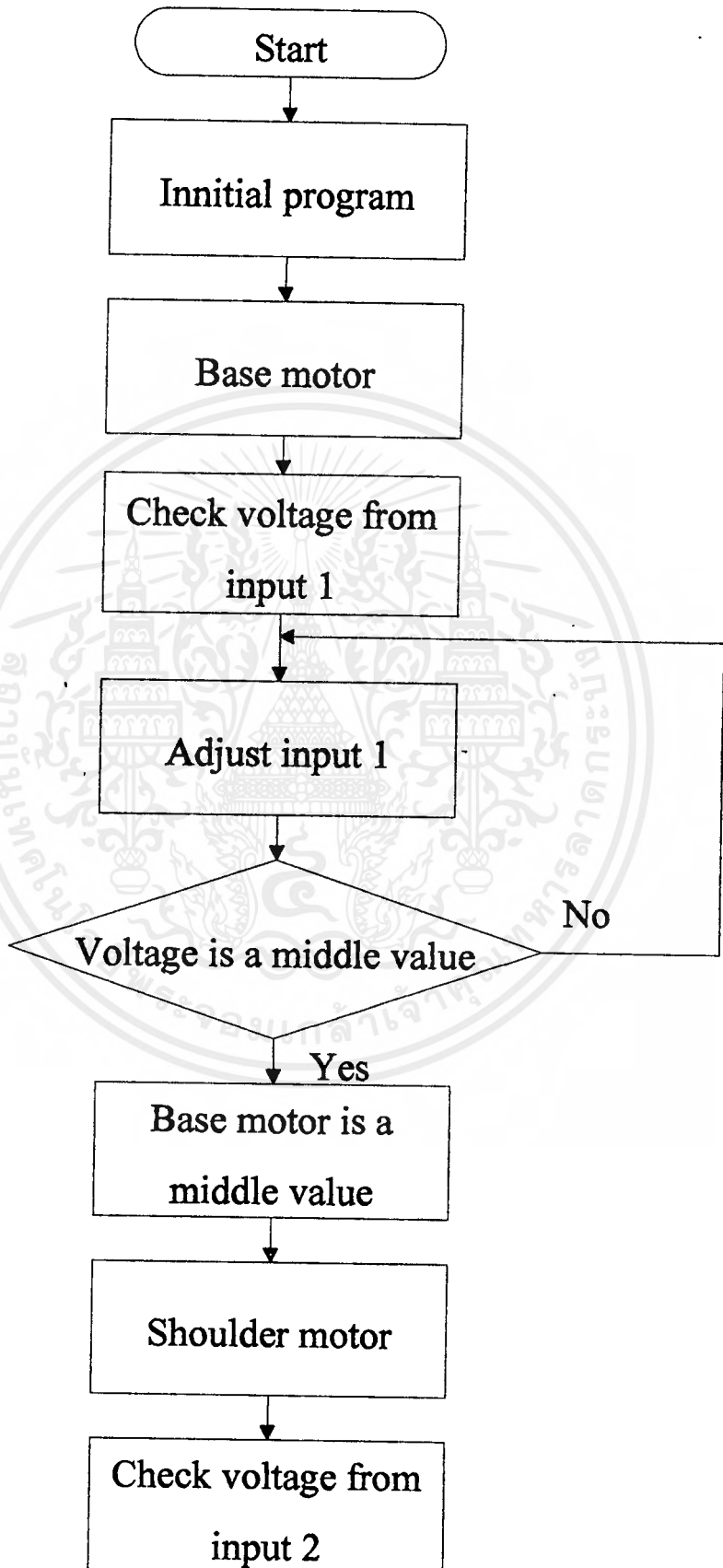
PC4-PC7 เป็นพอร์ต เอาท์พุท ซึ่งจะป็นสัญญาณควบคุม ADC 0809 โดย PC4 ต่อกับ ขา สตาร์ท, PC5 ต่อกับขา G , PC6 ต่อกับขา B, และ PC7 ต่อกับขา A ซึ่งเราสามารถกำหนดค่า PC4-PC7 เพื่อควบคุม IN0-IN7 ได้ดังนี้คือ

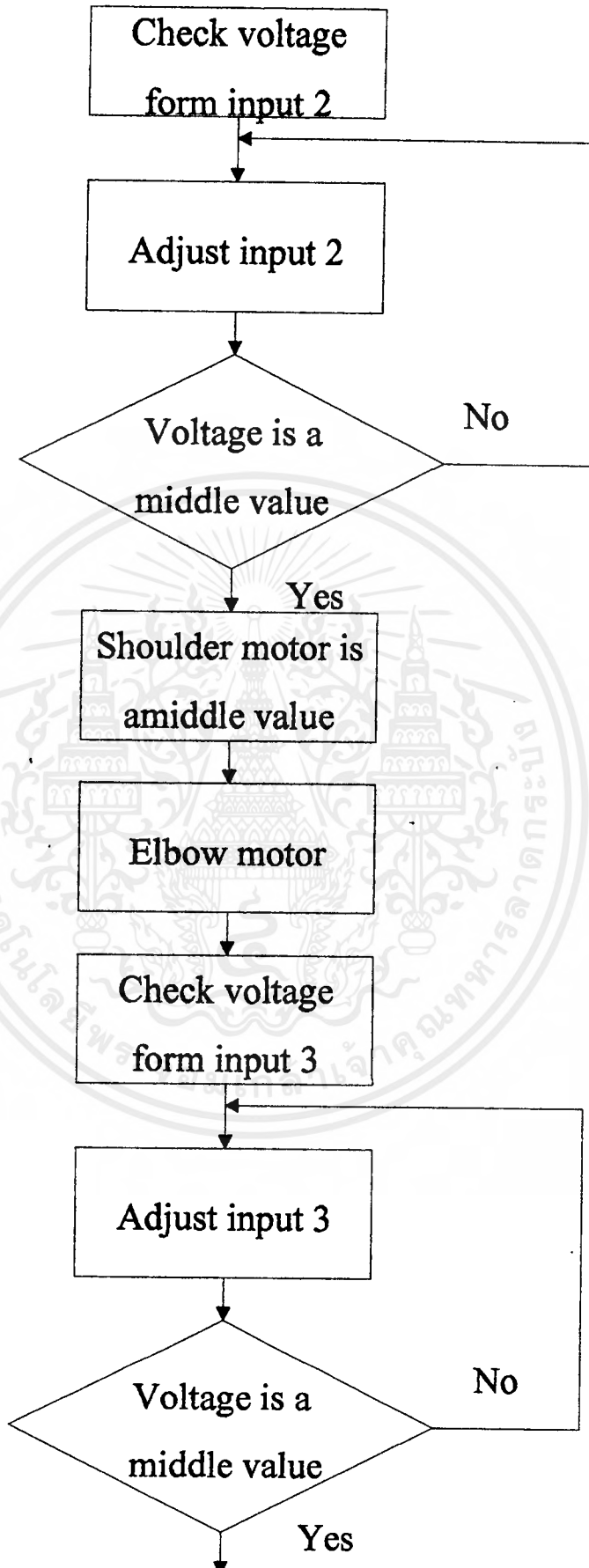
INPUT		PC7= A	PC6=B	PC5=C	PC4=START	Control bit
IN0	Input Voltage base motor	0	0	0	0	00H
IN1	Input Voltage feedback base motor	1	0	0	0	80H
IN2	Input Voltage gripper motor	0	1	0	0	40H
IN3	Input Voltage feedback gripper motor	1	1	0	0	C0H
IN4	Input Voltage shoulder motor	0	0	1	0	20H
IN5	Input Voltage feedback Shoulder motor	1	0	1	0	A0H
IN6	Input Voltage elbow motor	0	1	1	0	60H
IN7	Input Voltage feedback Elbow motor	1	1	1	0	E0H

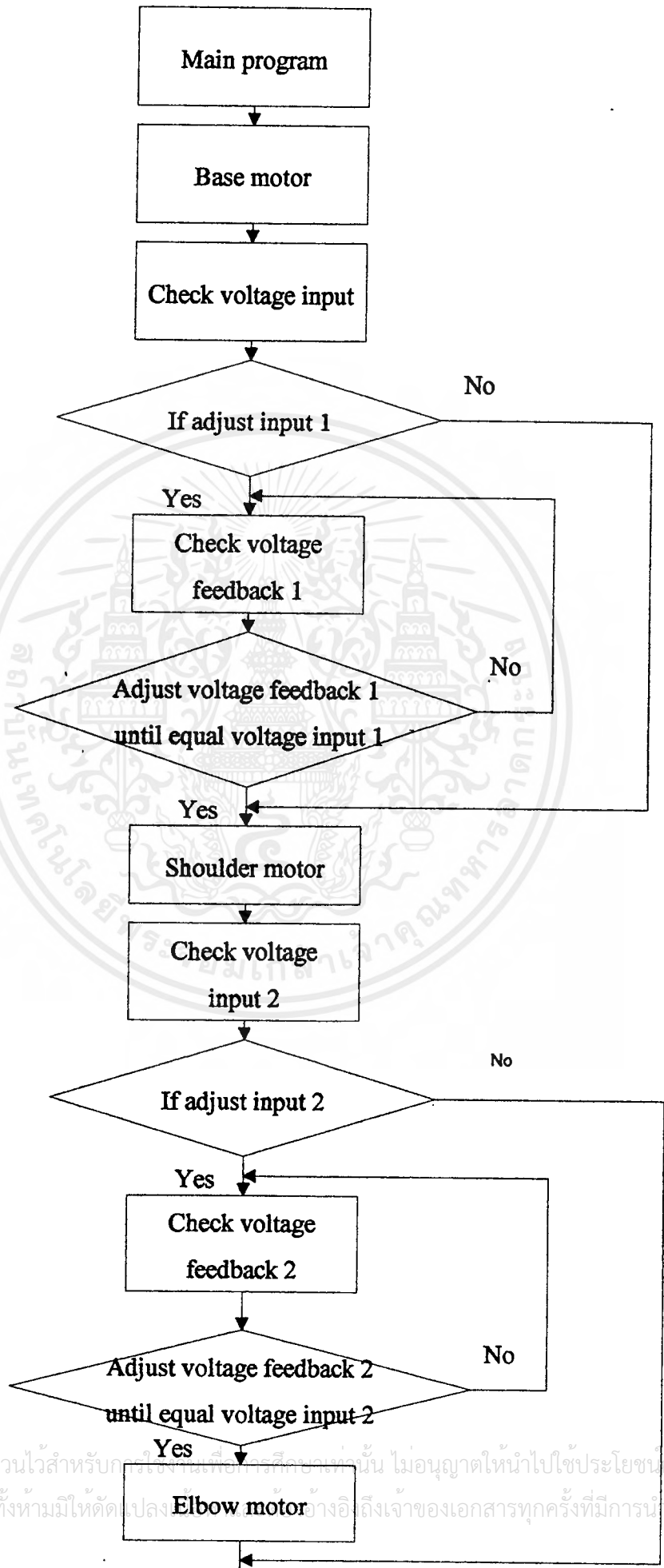
5.4 การออกแบบโปรแกรม

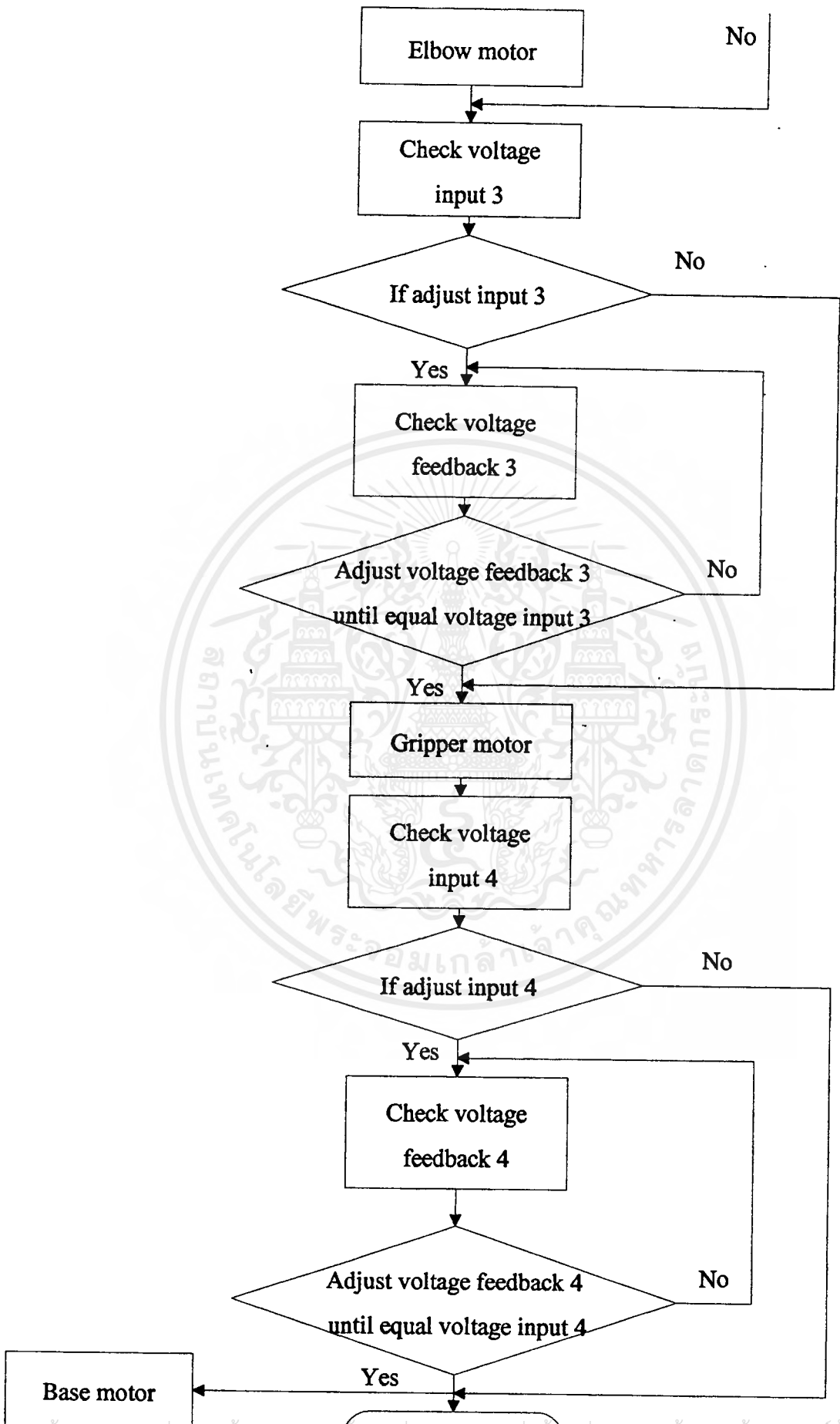
แขนกลจะทำงานโดยรับค่าจากคอมพิวเตอร์แล้วจะเคลื่อนที่ไปยังตำแหน่งที่ต้องการ แล้วจะส่งสัญญาณป้อนกลับไปยังคอมพิวเตอร์เมื่อถึงตำแหน่งที่ต้องการ ซึ่งในการควบคุมการทำงานจะเริ่มต้นการทำงานที่ตำแหน่ง 50% แล้วจะรับค่าอินพุทเพื่อควบคุมให้มอเตอร์เคลื่อนที่ตามตำแหน่งที่ผู้ใช้ต้องการ โดยจะออกแบบตามแสดงดังโพลซาร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้









บทที่ 6

ผลการทดลอง

ผลการทดลองจากการทดลองป้อนอินพุทให้แกนกลเคลื่อนที่ตาม แล้วนำผลอินพุทและเอาท์พุทมาแสดงผลบนหน้าจอกอมพิวเตอร์ โดยจะแยกส่วนต่าง ๆ ดังนี้

1. ส่วนฐาน เมื่อป้อนอินพุทส่วนฐานจะเคลื่อนที่ไปยังตำแหน่งที่อินพุทต้องการแล้วนำผลกลับมาแสดง จะเห็นได้ว่าเอาท์พุทที่นำมาแสดงจะมีค่าคลาดเคลื่อนกับอินพุทเล็กน้อยประมาณ 0.2 เปอเซนต์โดยนำผลมาแสดงในส่วนของ ค่า ผิดพลาด และในส่วนของการเคลื่อนที่ที่สามารถเคลื่อนที่ได้ ได้ดีมีความเร็วค่อนข้างคงที่

2. ส่วนที่หัวไหล่ เมื่อป้อนอินพุทส่วนฐานจะเคลื่อนที่ไปยังตำแหน่งที่อินพุทต้องการแล้วนำผลกลับมาแสดง จะเห็นได้ว่าเอาท์พุทที่นำมาแสดงจะมีค่าคลาดเคลื่อนกับอินพุทเล็กน้อยประมาณ 0.2 เปอเซนต์โดยนำผลมาแสดงในส่วนของค่าผิดพลาด และในส่วนของการเคลื่อนที่ที่สามารถเคลื่อนที่ได้ พอใช้มีความเร็วแตกต่างกันในบางช่วงเล็กน้อย

3. ส่วนข้อศอก เมื่อป้อนอินพุทส่วนฐานจะเคลื่อนที่ไปยังตำแหน่งที่อินพุทต้องการแล้วนำผลกลับมาแสดง จะเห็นได้ว่าเอาท์พุทที่นำมาแสดงจะมีค่าคลาดเคลื่อนกับอินพุทเล็กน้อยประมาณ 0.2 เปอเซนต์โดยนำผลมาแสดงในส่วนของค่าผิดพลาด และในส่วนของการเคลื่อนที่ที่สามารถเคลื่อนที่ได้ พอใช้มีความเร็วแตกต่างกันในบางช่วงเล็กน้อย

4. ส่วนข้อมือฐาน เมื่อป้อนอินพุทส่วนฐานจะเคลื่อนที่ไปยังตำแหน่งที่อินพุทต้องการแล้วนำผลกลับมาแสดง จะเห็นได้ว่าเอาท์พุทที่นำมาแสดงจะมีค่าคลาดเคลื่อนกับอินพุทเล็กน้อยประมาณ 0.2 เปอเซนต์โดยนำผลมาแสดงในส่วนของค่าผิดพลาด และในส่วนของการเคลื่อนที่ที่สามารถเคลื่อนที่ได้ ดีมีความเร็วค่อนข้างคงที่

บทที่ 7

สรุปและวิเคราะห์ผล

จากการทดลองจะเห็นว่ามีความคลาดเคลื่อนที่เกิดขึ้นจากการทำงาน สาเหตุสำคัญๆ ที่พบได้แก่

1. ความคลาดเคลื่อนที่เกิดจากกลไกของแขนกล ซึ่งเป็นสาเหตุสำคัญในการพิจารณาความคลาดเคลื่อนนี้ สามารถแยกเป็นต่างๆ ดังนี้คือ

1.1 ความคลาดเคลื่อนในการติดตั้งความคลาดเคลื่อนในการติดตั้งอุปกรณ์ โวลต์เดจพิคแบ็ค ซึ่งตั้งให้ค่าโวลต์เดจพิคแบ็ค เท่ากับ 0 พอดี เมื่อกลไกเคลื่อนที่มาสัมผัสสวิทช์ ซึ่งค่านี้ก็คือค่าเริ่มต้นนั่นเอง ซึ่งเมื่อเรามีการทำงานของกลไกมากขึ้นก็จะทำให้มีการเคลื่อนที่มากขึ้นด้วย ส่งผลให้มีการเปลี่ยนแปลงตำแหน่งคลาดเคลื่อนจากค่าเริ่มต้นไปเรื่อยๆ แต่ก็ยังเป็นปริมาณน้อยมาก เมื่อเทียบกับระยะทางในการเคลื่อนที่

1.2 เมื่อกลไกการทำงานมีการเคลื่อนที่ จนกระทั่งค่า โวลต์เดจอินพุทและค่าโวลต์เดจพิคแบ็คอินพุท มีค่าเท่ากันพอดี โปรแกรมก็จะสั่งให้กลไกหยุดการทำงาน ซึ่งมันจะต้องหยุดการเคลื่อนที่ทันที แต่ในการทำงานจริงๆ ก็ยังคงมีความคลาดเคลื่อนในจุดนี้เกิดขึ้นด้วย โดยเฉพาะในกรณีการเคลื่อนที่ลงซึ่งจะต้องมีค่าแรงโน้มถ่วงของโลกร่วมด้วย

2. ความคลาดเคลื่อนที่เกิดจาก โปรแกรมการทำงาน เพราะว่า โปรแกรมจะรับค่าอินพุท ซึ่งเป็นค่า ดิจิตอล 8 บิต ตั้งแต่ 0-225 (0-5 โวลท์) ซึ่งเป็นค่าจำนวนเต็มมาประมวลผล เมื่ออินพุททั้ง 8 ช่องซึ่งเป็นสัญญาณอนาล็อกจาก ADC 0809 มีการเปลี่ยนแปลงเล็กน้อย ก็จะทำให้ค่า ดิจิตอล ที่แปลงได้ มีการเปลี่ยนระดับตามไปด้วย ส่งผลให้การประมวลผลเกิดการคลาดเคลื่อนไปด้วย แต่ความคลาดเคลื่อนที่เกิดขึ้นจะมีค่าน้อยกว่าความคลาดเคลื่อนที่เกิดจากกลไกของแขนกลมาก

ภาคผนวก

program

```

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#define port_a 0x300
#define port_b 0x301
#define port_c 0x302
#define comp 0x303

void TextWindow(int x1,int y1,int x2,int y2,int fcolor,int bcolor);
void printxy(int x,int y,char line);
void windows(void);
void main()
{
    char ch;
    float a,e,f,g,h,i,j,k,l,s,t,u,v;
do
    {
        clrscr();
        TextWindow(1,1,80,26,BLUE,BLUE);
        TextWindow(3,2,78,24,CYAN,CYAN);
        textcolor(RED);
    }

        while(0);

windows();

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void printxy(int x,int y,char line)
{
    gotoxy(x,y);
    printf("%c",line);
}

void windows(void)
{
    float x=0,y=0,e,f,g,h,i,j,k,l,s,t,u,v,a;
    textcolor(LIGHTMAGENTA); gotoxy(30,4);
    cprintf("Intital Start program 50%");
    textcolor(BLUE);    gotoxy(14,6);
    cprintf("Base Motor");
    textcolor(BLUE);    gotoxy(33,6);
    cprintf("Shoulder");
    textcolor(BLUE);    gotoxy(53,6);
    cprintf("Elbow");
    textcolor(BLUE);    gotoxy(71,6);
    cprintf("Gripper");
    textcolor(BLUE);    gotoxy(4,9);
    cprintf("Input (%)");
    textcolor(BLUE);    gotoxy(4,14);
    cprintf("Output (%)");
    textcolor(BLUE);    gotoxy(4,19);
    cprintf("Error (%)");
    for(y=8;y<=22;y+=5)
    {
        for(x=15;x<80;x+=18)
        {
            textcolor(YELLOW);gotoxy(x,y);
            cprintf(".");
        }
    }
}

```

```

    }
}
for(y=8;y<=22;y+=5)
{
    for(x=24;x<80;x+=18)
    {
        textcolor(YELLOW);gotoxy(x,y);
        cprintf("ฟ");
    }
}
for(y=10;y<=20;y+=5)
{
    for(x=15;x<80;x+=18)
    {
        textcolor(YELLOW);gotoxy(x,y);
        cprintf("ภ");
    }
}
for(y=10;y<=20;y+=5)
{
    for(x=24;x<80;x+=18)
    {
        textcolor(YELLOW);gotoxy(x,y);
        cprintf("");
    }
}
for(i=0;i<3;i++)
{
    for(x=15;x<80;x+=18)
    {

```

```

        for(y=(i*5)+9;y<(i*5)+10;y++)

```

```

        {
            textcolor(YELLOW);gotoxy(x,y);
            cprintf("ณ");
        }
    }
    for(x=24;x<80;x+=18)
    {
        for(y=(i*5)+9;y<(i*5)+9;y++);
        {
            textcolor(YELLOW);gotoxy(x,y);
            cprintf("ณ");
        }
    }
}
for(i=0;i<=3;i++)
{
    for(y=8;y<20;y+=5)
    {
        for(x=(i*18)+16;x<(i*18)+24;x++)
        {
            textcolor(YELLOW);gotoxy(x,y);
            cprintf("-");
        }
    }
    for(y=10;y<24;y+=5)
    {
        for(x=(i*18)+16;x<(i*18)+24;x++)
        {
            textcolor(YELLOW);gotoxy(x,y);
            cprintf("-");
        }
    }
}

```

```

}
}

```

```

gotoxy(17,19);textcolor(RED);
printf("00.00");
gotoxy(35,19); textcolor(RED);
printf("00.00");
gotoxy(53,19); textcolor(RED);
printf("00.00");
gotoxy(71,19);
printf("00.00");
gotoxy(17,14);textcolor(RED);
printf("00.00");
gotoxy(35,9);textcolor(RED);
printf("00.00");
gotoxy(35,14);textcolor(RED);
printf("00.00");
gotoxy(53,9);textcolor(RED);
printf("00.00");
gotoxy(53,14);textcolor(RED);
printf("00.00");
gotoxy(71,9);textcolor(RED);
printf("00.00");
gotoxy(71,14);textcolor(RED);
printf("00.00");

outp (comp,0x83);

/* initial program */

/* base motor */

```

m1:

```
outp (port_a,0x00);
```

```

        outp (port_c,0x00);
        outp (port_c,0x10);
        outp (port_c,0x00);
        a=inportb (0x301);
        g=a;
        gotoxy(17,9);textcolor(RED);
        cprintf("%.2f",g);
    if (g==100)
        {
            outp (port_a,0x04);
        }
    else
        {
            goto m1;
        }
        /* middle value */
        outp (port_a,0x04);    /* left */
        outp (port_c,0x0f);
m1:    s=inportb (0x302);
        if (s==11)
            {
                outp (port_a,0x44);    /* right */
            }
        else
            {
                outp (port_a,0x04);
                goto m1;
            }

```

```

/* initial input from feedback base motor */
    outp (port_a,0x44);
m111:    outp (port_c,0x80);
        outp (port_c,0x90);
        outp (port_c,0x80);
        a = inportb (0x301);
        h=a;
        gotoxy(17,14);
        cprintf("%.2f",h);
if (h>g)
    {
        outp (port_a,0x44);
    }
else if (h<=g)
    {
        outp (port_a,0x44);
        gotoxy(17,14);
        cprintf ("h= %d\n",h);
        goto m111;
    }
m1111:  outp (port_c,0x80);
        outp (port_c,0x90);
        outp (port_c,0x80);
        a = inportb (0x301);
        h=a;
        gotoxy(17,14);
        cprintf("%.2f",h);
if (h>=g)
    {
        outp (port_a,0x00);
    }

```

```

else
    goto m1111;

    /* initial program */
    /* shoulder motor */

m2:    outp (port_a,0x00);
        outp (port_c,0x20);
        outp (port_c,0x30);
        outp (port_c,0x20);
        a=inportb (0x301);
        i=a;
        gotoxy(35,9);
        cprintf("%.2f",i);
    if (i>80 && i<83)
        {
            outp (port_a,0x22);
        }
    else
        {
            goto m2;
        }

        /* middle value */

        outp (port_a,0x22);    /* up */
        outp (port_c,0x0f);

m22:   t=inportb (0x302);
        if (t==14)
            {
                outp (port_a,0x02);    /* down */

```

```

    }
else
    {
        goto m22;
    }

/* initial input from feedback shoulder motor */
    outp (port_a,0x02);
m222: outp (port_c,0xa0);
        outp (port_c,0xb0);
        outp (port_c,0xa0);
        a = inportb (0x301);
        j=a;
        gotoxy(35,14);
        cprintf("%.2f",j);
        if (j>i)
        {
            outp (port_a,0x02);
        }
else if (j<=i)
    {
        outp (port_a,0x02);
        goto m222;
    }
m2222: outp (port_c,0xa0);
        outp (port_c,0xb0);
        outp (port_c,0xa0);
        a = inportb (0x301);
        j=a;
        gotoxy(35,14);
        cprintf("%.2f",j);

```

```

if (j>=i)
    {
        outp (port_a,0x00);
    }
else
    goto m2222;

/* initial program */
/* elbow motor */

m3:    outp (port_a,0x00);
        outp (port_c,0x60);
        outp (port_c,0x70);
        outp (port_c,0x60);
        a=inportb (0x301);
        k=a;
        gotoxy(53,9);
        cprintf("%.2f",k);
if (k==100)
    {
        outp (port_a,0x11);
    }
else
    {
        goto m3;
    }

/* middle value */

        outp (port_a,0x11);    /* up */
        outp (port_c,0x0f);

```

```

m33:      u=inportb (0x302);
          if (u==13)
              {
                  outp (port_a,0x01);    /* down */
              }
          else
              {
                  outp (port_a,0x11);
                  goto m33;
              }

/* initial input from feedback elbow motor */
          outp (port_a,0x01);
m333:     outp (port_c,0xe0);
          outp (port_c,0xf0);
          outp (port_c,0xe0);
          a = inportb (0x301);
          l=a;
          gotoxy(53,14);
          cprintf("%.2f",l);

          if (l>k)
              {
                  outp (port_a,0x01);
              }

          else if (l<=k)
              {
                  outp (port_a,0x01);
                  gotoxy(53,14);
                  printf ("l= %d\n",l);
                  goto m333;

```

```

    }
m3333:    outp (port_c,0xe0);
          outp (port_c,0xf0);
          outp (port_c,0xe0);
          a = inportb (0x301);
          l=a;
          gotoxy(53,14);
          cprintf("%.2f",l);

```

```
if (l>=k)
```

```

{
  outp (port_a,0x00);
}

```

```
else
```

```

  goto m3333;
  /* initial program */
  /* gripper motor */

```

```

m4:      outp (port_a,0x00);
          outp (port_c,0x40);
          outp (port_c,0x50);
          outp (port_c,0x40);
          a=inportb (0x301);
          e=a;
          gotoxy(71,9);
          cprintf("%.2f",e);

```

```
if (e>50 && e<53)
```

```

{
  outp (port_a,0x08);
}

```

```
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        goto m4;
    }

    /* middle value */

    outp (port_a,0x88);    /* up */
    outp (port_c,0x0f);

m44:  v=inportb (0x302);
    if (v==7)
    {
        outp (port_a,0x08);
    }
    else
    {
        goto m44;
    }

    /* initial input from feedback gripper motor */
    outp (port_a,0x08);
m444: outp (port_c,0xc0);
        outp (port_c,0xd0);
        outp (port_c,0xc0);
        a = inportb (0x301);
        f=a;
        gotoxy(71,14);
        cprintf("%.2f",f);
        if (f>e)
        {
            outp (port_a,0x08);
        }

```

```

else if (f<=e)
    {
        outp (port_a,0x08);
        goto m444;
    }
m4444:    outp (port_c,0xc0);
          outp (port_c,0xd0);
          outp (port_c,0xc0);
          a = inportb (0x301);
          f=a;
          gotoxy(71,14);
          cprintf("%.2f",f);
if (f>=e)
    {
        outp (port_a,0x00);
    }
else
    goto m4444;

```

```

/* main program base motor */

```

```

loop1:    clrscr();

          outp (port_c,0x00);
          outp (port_c,0x10);
          outp (port_c,0x00);
          a=inportb (0x301);
          g=a;

```

```

loop2:    clrscr();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        outp (port_c,0x00);
        outp (port_c,0x10);
        outp (port_c,0x00);
        a=inportb (0x301);
        g=a;
    if (h>g-3 && h<g+3)
        {
            gotoxy(17,9);textcolor(RED);
            cprintf("%.2f",g);
            gotoxy(17,19);
            cprintf("%.2f",((g-h)/g)*100);
            goto loop3;
        }
    else
        {
            goto input1;
        }

    /* in new value from base motor */

```

```

input1:  clrscr();
        outp (port_c,0x80);
        outp (port_c,0x90);
        outp (port_c,0x80);
        a=inportb (0x301);
        h=a;
        clrscr();
    outp (port_c,0x80);
        outp (port_c,0x90);
        outp (port_c,0x80);
        a=inportb (0x301);

```

```

        h=a;
        gotoxy(17,14);
        cprintf("%.2f",h);
    if (h>g-3)
        {
            outp (port_a,0x04);
            goto input12;
        }
    else if (h<g+3)
        {
            outp (port_a,0x44);
        }
input12: clrscr();
        outp (port_c,0x80);
        outp (port_c,0x90);
        outp (port_c,0x80);
        a=inportb (0x301);
        h=a;
        gotoxy(17,14);
        cprintf("%.2f",h);
    if (h>g-3 && h<g+3)
        {
            outp (port_a,00);
            goto loop2;
        }
    else
        {
            goto input11;
        }

```

/* main program shoulder motor */

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

loop3:   clrscr();

         outp (port_c,0x20);

         outp (port_c,0x30);

         outp (port_c,0x20);

         a=inportb (0x301);

         i=a;

loop4:   clrscr();

         outp (port_c,0x20);

         outp (port_c,0x30);

         outp (port_c,0x20);

         a=inportb (0x301);

         i=a;

if (j>i-3 && j<i+3)
{
  gotoxy(35,9);
  cprintf("%.2f",i);
  gotoxy(35,19);
  cprintf("%.2f",((i-j)/i)*100);
  goto loop5;
}

else

{

  goto input21;

}

/* new value from shoulder motor */

input21:  clrscr();

          outp (port_c,0xa0);

          outp (port_c,0xb0);

```

```

    outp (port_c,0xa0);
    a=inportb (0x301);
    j=a;
    clrscr();
    outp (port_c,0xa0);
    outp (port_c,0xb0);
    outp (port_c,0xa0);
    a=inportb (0x301);
    j=a;
    gotoxy(35,14);
    printf("%.2f",j);
if (j>i-3)
    {
        outp (port_a,0x22);    /* up */
        goto input22;
    }
else if (j<i+3)
    {
        outp (port_a,0x02);    /* down */
    }
input22:  clrscr();

    outp (port_c,0xa0);
    outp (port_c,0xb0);
    outp (port_c,0xa0);
    a=inportb (0x301);
    j=a;
    gotoxy(35,14);
    printf("%.2f",j);
    if (j>i-3 && j<i+3)
    {
        outp (port_a,0x00);

```

```

        goto loop4;
    }
else
    {
        goto input21;
    }

/* main program elbow motor */

loop5:  clrscr();
        outp (port_c,0x60);
        outp (port_c,0x70);
        outp (port_c,0x60);
        a=inportb (0x301);
        k=a;
loop6:  clrscr();
        outp (port_c,0x60);
        outp (port_c,0x70);
        outp (port_c,0x60);
        a=inportb (0x301);
        k=a;
if (l>k-3 && l<k+3)
    {
        gotoxy(53,9);
        cprintf("%%.2f",k);
        gotoxy(53,19);
        cprintf("%%.2f",((k-1)/k)*100);
        goto loop7;
    }
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        goto input31;
    }

    /* new value from elbow motor */

input31:    clrscr();

            outp (port_c,0xe0);
            outp (port_c,0xf0);
            outp (port_c,0xe0);
            a=inportb (0x301);
            l=a;
            clrscr();
            outp (port_c,0xe0);
            outp (port_c,0xf0);
            outp (port_c,0xe0);
            a=inportb (0x301);
            l=a;
            gotoxy(53,14);
            cprintf("%.2f",l);

    if (l>k-3)
    {
        outp (port_a,0x11);    /* up */
        goto input32;
    }

    else if (l<k+3)
    {
        outp (port_a,0x01);    /* down */
    }

input32:    clrscr();

            outp (port_c,0xe0);
            outp (port_c,0xf0);

```

```

        outp (port_c,0xe0);
        a=inportb (0x301);
        l=a;
        gotoxy(53,14);
        cprintf("%.2f",l);
if (l>k-3 && l<k+3)
    {
        outp (port_a,0x00);
        goto loop6;
    }
else
    {
        goto input31;
    }
/* main program gripper motor */
loop7: clrscr();
        outp (port_c,0x40);
        outp (port_c,0x50);
        outp (port_c,0x40);
        a=inportb (0x301);
        e=a;
loop8:  clrscr();
        outp (port_c,0x40);
        outp (port_c,0x50);
        outp (port_c,0x40);
        a=inportb (0x301);
        e=a;
if (f>e-3 && f<e+3)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        gotoxy(71,9);
        cprintf("%.2f",e);
        gotoxy(71,19);
        cprintf("%.2f",((e-f)/e)*100);
        goto loop1;
    }
else
    {
        goto input41;
    }

/* new value from gripper motor */
input41: clrscr();
        outp (port_c,0xc0);
        outp (port_c,0xd0);
        outp (port_c,0xc0);
        a=inportb (0x301);
        f=a;
        clrscr();
        outp (port_c,0xc0);
        outp (port_c,0xd0);
        outp (port_c,0xc0);
        a=inportb (0x301);
        f=a;
        gotoxy(71,14);
        cprintf("%.2f",f);
if (f>e-3)
    {
        outp (port_a,0x88);
        goto input42;

```

```

    }
    else if (f<e+3)
    {
        outp (port_a,0x08);
    }
input42:   clrscr();
           outp (port_c,0xc0);
           outp (port_c,0xd0);
           outp (port_c,0xc0);
           a=inportb (0x301);
           f=a;
           gotoxy(71,14);
           cprintf("%.2f",f);
if (f>e-3 && f<e+3)
{
    outp (port_a,00);
    goto loop8;
}
else
{
    goto input41;
}

}

void TextWindow(int x1,int y1,int x2,int y2,int fcolor,int bcolor)
{
    int x,y;
    textcolor(fcolor);
    textbackground(bcolor);

```

```

for(y=y1+1;y<y2;y++)
{
    for(x=x1+1;x<x2;x++)
    {
        gotoxy(x,y);
        putch(0x20);
    }
    gotoxy(x1,y);putch(0xb3);
    gotoxy(x2,y);putch(0xb3);
}
gotoxy(x1+1,y1);
for(x=x1+1;x<x2;x++)putch(0xc4);
gotoxy(x1+1,y2);
for(x=x1+1;x<x2;x++)putch(0xc4);
gotoxy(x1,y1);putch(0xda);
gotoxy(x2,y1);putch(0xbf);
gotoxy(x1,y2);putch(0xc0);
gotoxy(x2,y2);putch(0xd9);
}

```

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลงได้ด้วยความร่วมมือและความตั้งใจในการทำงานของเพื่อนร่วมงานในกลุ่ม PROJECT ทั้งได้รับคำแนะนำจากอาจารย์ พลศาสตร์ เทศประเสริฐ ซึ่งเป็นอาจารย์ที่ปรึกษาทางด้านเทคนิคและแนวคิดต่าง ๆ ในการปฏิบัติงาน นอกจากนี้ยังได้รับคำแนะนำทางด้าน Hardware จากรุ่นน้องภาคเครื่องกล

คณะผู้จัดทำจึงขอขอบคุณ บุคคลทุกท่านที่ได้ให้คำแนะนำ ไว้ ณ โอกาสนี้เป็นอย่างสูง



.....
(นาย เกรียงศักดิ์ อิ่มศรี)

.....
(นาย สัตยญา กวรคิด)

.....
(นาย ศุภประชัย พิ้วฝ้าย)

หนังสืออ้างอิง

1. Schilling, Robert J , “Fundamental of Robotics Analysis and Control”,Prentice-Hall,1990
2. Moshe Shoham ,”Robotic Structure”,Eshed Robotec, (1982)
3. ผศ. โยธิน เปรมปราณีรัชต์ “วิเคราะห์และออกแบบระบบควบคุมมอเตอร์”,พระจอมเกล้าลาดกระบัง,2536
4. นาย ไพศาล เศรษฐคนประเสริฐ “มูมพีซีฮาร์ดแวร์”,เซมิคอนดักเตอร์อิเล็กทรอนิกส์ ฉบับที่ 104 , 2534

