



ปีการศึกษา 2534

ปริณิษานินพนธ์ เรื่อง

แขนกล

(ROBOT ARM)

ผู้จัดทำ

นายธิตติ	ตันอารีย์	รหัส 81.1115
นางสาวธิตารัตน์	นรภาณุजनันท์	รหัส 81.1112
นายธีธี	เจริญศรี	รหัส 81.1130

อาจารย์ที่ปรึกษา

อ. โศศล

ชวนชน

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ 008497 ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริิณญาณ์
เรื่อง

ปีการศึกษา 2534
แขนง (ROBOT ARM)

ผู้จัดทำ

นาย ชิตี	ตันอารีย์	รหัส 31.1115
นางสาว ชิตารัตน์	พรภาณูจนานันท์	รหัส 31.1112
นาย นิธิ	เจริญศรี	รหัส 31.1130



(อ. โทศล ขวนขยัน)
อาจารย์ที่ปรึกษา

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แขนกล

นางสาว นิตารัตน์ พรภาบุญนานันท์ 31.1112
นาย อธิติ ตันอารีย์ 31.1115
นาย นิธิ เจริญศรี 31.1130

อาจารย์ที่ปรึกษา
อาจารย์ โกศล ชวนขยัน
ภาคการศึกษาที่ 2 ปีการศึกษา 2534

บทคัดย่อ

ในโครงงานนี้ เป็นการศึกษา การควบคุม Stepping motor ซึ่งเป็น motor ที่ต้องการการควบคุมแบบ open loop เพื่อให้ไปยังตำแหน่งต่างๆ ตามต้องการ โดยใช้ Microcomputer เป็นตัวควบคุม และนำ Stepping motor นี้ไปเป็นตัวขับเคลื่อนแขนกล โดยแขนกลนี้แตกออกแบบเป็นลักษณะที่ติดตั้งกลม ซึ่งมีลักษณะคล้ายแขนของมนุษย์ ทำให้เคลื่อนที่ได้รอบตัวและบริเวณที่การทำงานกว้าง เพื่อนำไปใช้งานในการเจาะแผ่น PCB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROBOT ARM

Thidaral Pornkarnchananun 31.1112

Thiti Tanaree 31.1115

Nithi Chareonsri 31.1130

Adviser

MR.Kosal Chuankayan

For 2nd semester 1992

ABSTRACT

In this project is used in studying the controlling of stepping motor which must be controlled in open loop system. The objective is to reach the required positions. We use the personal computer to be the controller and use stepping motors to manipulate the robot arm.

This robot arm is designed in cartesian co-ordinate for the wide area operating. Its mechanical and operating design simulate the figure and operating of human arm. The application for our robot arm is drilling the print circuit board.

บทที่	หน้า
1. จุดประสงค์	1
2. ทฤษฎีเบื้องต้นของ stopping motor	3
2.1 ความแม่นยำของตำแหน่ง	5
2.2 อัตราแรงเฉื่อย	5
2.3 อัตรา stepping และ ความถี่ pulse	6
2.4 ลักษณะการทำงานของ stepping motor	19
2.5 ลักษณะเฉพาะของแรงบิดสถิตย์	22
2.6 ลักษณะเฉพาะของแรงบิดจลน์	28
3. Drive circuit สำหรับ stepping motor	28
3.1 หลักการ	30
3.2 การออกแบบวงจร drive	34
4. การออกแบบโปรแกรม	34
4.1 การคำนวณหาค่ามุมของมอเตอร์แต่ละตัว	34
4.2 วิธีคำนวณหามุม	35
4.3 การคำนวณหาจำนวน clock ของมอเตอร์แต่ละตัว	38
4.4 การออกแบบสัญญาณ sequential	38
5. การออกแบบแขนกล	42
5.1 ลักษณะของแขนกล	42
5.2 ระบบส่งกำลัง	48
6. ผลการทดลอง	50
6.1 ผลการทดลอง	50
6.2 รูปผลการทดลอง	50
7. วิธีการใช้โปรแกรม	52

-Flow chart -program -กิตติกรรมประกาศ -บรรณานุกรม

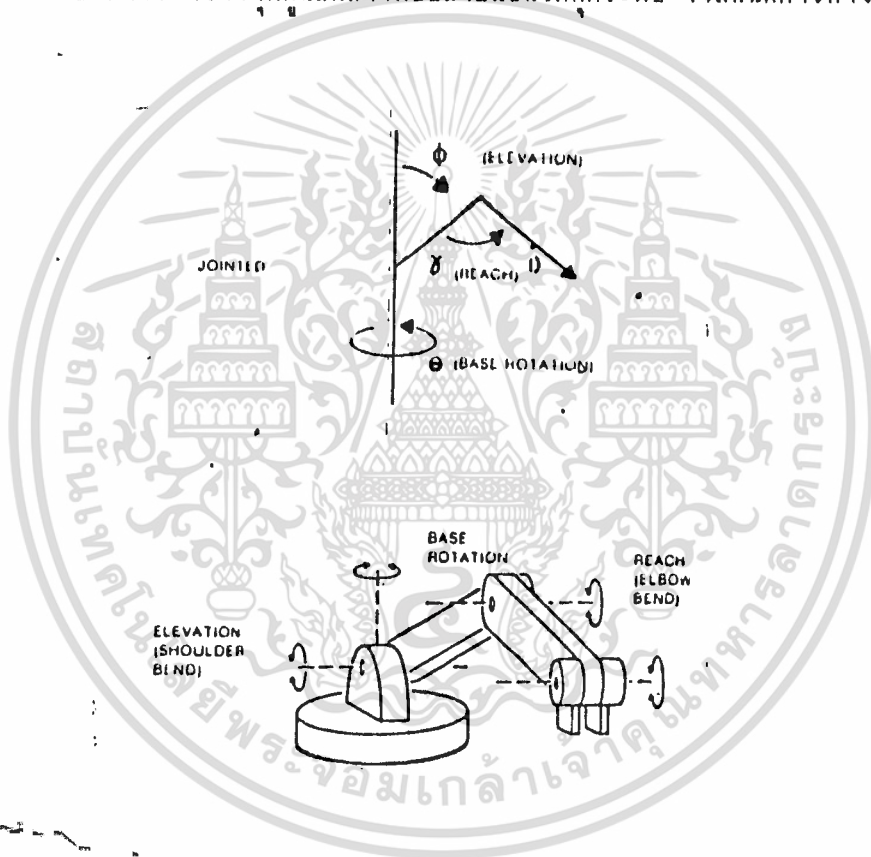
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

จุดประสงค์

การออกแบบแขนกลและโปรแกรม จะเน้นหนักในการให้ความสะดวกแก่ผู้ใช้และง่ายต่อการทำงาน แขนกลที่ออกแบบนี้จะอยู่ในลักษณะพิกัดทรงกลม ซึ่งทำให้เคลื่อนที่ได้รอบตัวและสามารถปรับระยะห่างระหว่างจุดศูนย์กลางกับปลายแขนได้ทุกระดับ จึงมีพื้นที่การทำงานมาก



จากแบบที่กำหนดไว้ทำให้เราต้องใช้ stepping motor 4 ตัว

การควบคุมแขนกลนั้นเราจำเป็นต้องทราบว่า stepping motor แต่ละตัวอยู่ที่ตำแหน่งใด และต้องสามารถควบคุมการเคลื่อนที่ของมอเตอร์แต่ละตัวไปตำแหน่งต่างๆ ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถเลือกใช้ Motor ได้ทั้ง AC motor ,DC motor และ stepping motor แต่ AC motor และ DC motor มีส่วนควบคุมที่ค่อนข้างยุ่งยาก เช่น การ on หรือ off Motor อีกทั้งยังต้องมี feedback เพื่อควบคุมความเร็วและตำแหน่งซึ่งจำเป็นต้องมีอุปกรณ์มากทำให้มีน้ำหนักมากด้วย ในโครงการนี้จึงเลือก stepping motor ซึ่งมีการควบคุมที่ง่ายกว่า และได้ผลใกล้เคียงกัน ส่วนคุณลักษณะต่าง ๆ ของ stepping motor จะกล่าวถึงต่อไป

จากลักษณะของแขนกลที่ออกแบบ การควบคุมตำแหน่งของปลายแขน จำเป็นต้องป้อนค่ามุมให้แก่ stepping motor แต่ละตัว ทำให้ไม่สะดวกต่อผู้ใช้จึงได้ออกแบบโปรแกรม ที่ทำให้เกิดความสะดวก คือ ผู้ใช้จะสั่งแขนกลได้โดยการป้อนค่า co-ordinate (x,y) ของแผ่น PCB ลงไปโปรแกรมแล้ว โปรแกรมจะคำนวณหามุมของ stepping motor แต่ละตัว และควบคุมการทำงานของ Motor แต่ละตัวไปยังมุมที่ต้องการได้

และในส่วนของโปรแกรม ยังให้ความสะดวกในการเก็บข้อมูล เขียนข้อมูลใหม่ อ่านข้อมูลเก่า ในกรณีที่ต้องการทำการเจาะซ้ำ ๆ กัน และการควบคุมการเจาะแบบครั้งเดียว เพื่อแก้ไขจุดผิดพลาด และในโปรแกรมจะมีการแยก function การทำงานแบบต่าง ๆ ใน menu ทำให้สามารถเลือกการทำงานได้อย่างสะดวก

การ Interface ระหว่าง Robot driver กับ Microcomputer เราจะ Interface ผ่าน printer port ของเครื่องคอมพิวเตอร์ ซึ่งมีข้อดีคือ สามารถต่อได้ง่าย และสะดวกในการสับเปลี่ยนเครื่องคอมพิวเตอร์ในการควบคุม

สรุปจุดประสงค์

1. ผู้ใช้สามารถควบคุมได้โดยการป้อนค่า co-ordinate ของแผ่น PCB
2. สามารถเก็บข้อมูลใหม่ และอ่านข้อมูลของตำแหน่งการเจาะแผ่น PCB เก่าได้
3. ต่อกับ Microcomputer ผ่านทาง Printer port

บทที่ 2

ทฤษฎีเบื้องต้นของ Stepping motor

stepping motor เป็น DC motor ชนิดหนึ่ง ซึ่งแตกต่างจาก DC motor ทั่วไปคือไม่ต้องอาศัยแปรงถ่านในการทำงาน โดยการทำงานจะขึ้นกับวงจรรีเลย์ทรอนิกส์ การควบคุมความเร็วและตำแหน่งของ Motor สามารถทำได้โดยไม่ต้องมีการ feedback จึงเรียกการทำงานแบบนี้ว่า open loop

แม้ว่าการควบคุมแบบ open loop จะเป็นการขับมอเตอร์ที่สะดวก ไม่ยุ่งยาก แต่จะมีปัญหาบางประการ เช่น การหมุนของ rotor อาจ oscillate และไม่คงที่ในบางช่วงความถี่ ดังนั้นจึงทำให้การควบคุมความเร็วและความเร่งของ stepping motor ที่มีการควบคุมแบบ open loop จึงไม่สามารถทำให้มีความเร็วได้เท่ากับ DC motor ที่มีการควบคุมแบบ feedback (close loop) ได้

ลักษณะสำคัญที่สุดของ stepping motor คือ มันจะหมุนด้วยองศาคงที่ ในแต่ละ pulse ที่ถูกป้อนเข้ามาที่ระดับคลวด ซึ่งได้จาก logic sequencer

การสร้าง step pulse logic sequencer จะทำให้ทราบว่าเฟสใดจะถูกกระตุ้นและเฟสใดจะถูกปลดการกระตุ้น

step	ϕ_1	ϕ_2	ϕ_3	ϕ_4
1	x	x		
2		x	x	
3			x	x
4	x			x

ดังรูป ใน step 1 ϕ_1 และ ϕ_2 ถูกกระตุ้น ส่วน ϕ_3 และ ϕ_4 ลดการกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสัญญาณเหล่านี้จะส่งไปภาค Driver เพื่อทำการร่ายกำลังเพื่อป้อนกระแสเข้าสู่เฟสต่างๆ ของ stepping motor

logic sequencer เหล่านี้จะถูกสร้างจาก IC TTL , CMOS หรือ Microprocessor ก็ได้ โดย logic high แสดงถึงการกระตุ้นขดลวดของเฟสนั้น

เนื่องจาก stepping motor จะหมุนด้วยมุมคงที่ในแต่ละ pulse ที่ป้อนเข้ามา ขนาดของมุมนี้ จะถูกเรียกว่า step angle มีหน่วยเป็นองศา step angle ยิ่งเล็ก ตำแหน่งการหมุนก็จะยิ่งละเอียด เราสามารถคำนวณหา step number ได้จากสูตร

$$s = 360/\theta \quad \dots(2.1)$$

เมื่อ s = step number (step / รอบ)

θ = step angle

stepping motor ที่ใช้ทั่วไป ส่วนใหญ่จะมี $s = 96, 128, 136, 200$

ถ้าเป็น stepping motor ที่มีความแม่นยำสูงอาจมี $s = 500$ หรือ 1000 steps

2.1 ความแม่นยำของตำแหน่ง

การเคลื่อนไปตามตำแหน่งอย่างแม่นยำจะเป็นตัวตัดสินคุณภาพของ stepping motor ซึ่งจะถูกออกแบบให้หมุนทีละ step angle ตามสัญญาณ pulse ที่ป้อนเข้ามา และหยุดในตำแหน่งที่ถูกต้อง ดังนั้นความแม่นยำในขณะที่เคลื่อนที่แบบ no load จะขึ้นอยู่กับประสิทธิภาพของตัว stepping motor เอง แต่ stepping motor ในขณะที่เคลื่อนที่แบบมี load จะทำให้ air gap ระหว่างตัว rotor และ pole ต่างจากขณะที่ไม่มี load หรือช่วง air gap จะกว้างขึ้นนั่นเอง ซึ่งช่วงกว้างของ air gap จะขึ้นอยู่กับค่า maximum static torque - ซึ่งยังแบ่งได้เป็น

- holding torque หมายถึง maximum static torque ที่ป้อนเข้าสู่เฟสของ stepping motor ที่ถูกกระตุ้นโดยจะทำให้เกิดการหมุน
- detent torque หมายถึง maximum static torque ที่เกิดในเฟสของ stepping motor ที่ไม่ถูกกระตุ้น ซึ่งเป็น torque ที่เกิดจากแม่เหล็กถาวรใน rotor

ของ Motor แบบ แม่เหล็กถาวร ซึ่งจะไม่ทำให้เกิดการหมุนตัวของ stepping motor

ซึ่งโดยทั่วไปแล้ว holding torque ยิ่งมากเท่าใด ความผิดพลาดในตำแหน่งอันเนื่องมาจาก load ก็ยิ่งน้อย

ตำแหน่งขณะ stepping motor หยุดหมุน

- rest position or equilibrium position หมายถึง ตำแหน่งที่ stepping motor มาหยุด เกิดจาก sequential logic ที่ไม่มีการเปลี่ยนแปลง step อีก

- detent position หมายถึงตำแหน่งที่ stepping motor แบบแม่เหล็กถาวรใน rotor หรือ stator หยุด เมื่อไม่มีการกระตุ้น

2.2 อัตราแรงเฉื่อย

จากจุดหมายในการทำงานของ stepping motor อย่างหนึ่ง คือ สามารถหมุนได้เร็วเท่าที่จะเป็นไปได้ โดยจะตอบสนอง input pulse ที่ป้อนเข้ามา จะเห็นว่า stepping motor จำเป็นต้องมีการเริ่มต้นหมุนอย่างรวดเร็วและต้องหยุดทันทีเมื่อมีขบวน pulse เข้ามาหรือหยุดตามลำดับ ซึ่งถ้าขบวน pulse ถูกขัดจังหวะขณะที่หมุนด้วยความเร็วค่าหนึ่ง stepping motor ก็ควรจะหยุดในตำแหน่งที่ pulse สุดท้ายเข้ามา ซึ่งจะเห็นได้ว่าอัตราส่วนของแรงบิดเฉื่อยของ rotor จะต้องมีขนาดใหญ่ เมื่อเปรียบเทียบกับ motor ไฟฟ้าทั่วไป ทำให้เกิดการเปลี่ยนแปลงของกระแสมากในขณะเริ่มและหยุดหมุน จึงทำให้การออกแบบวงจร

Driver circuit และ Power supply ต้องมีค่านิ่งถึงด้วย

2.3 อัตรา stepping และความถี่ pulse

ความเร็วในการหมุนของ stepping motor จะอยู่ในเทอมของจำนวน step ต่อวินาที หรือ stepping rate, ซึ่งมักจะนิยมใช้บอกความเร็วของ stepping motor เพราะใน stepping motor จำนวน pulse ที่ป้อนให้ logic sequencer จะเท่ากับ จำนวน step ที่ stepping motor เคลื่อนไป ดังนั้นเราสามารถกำหนดความเร็วได้จากความถี่ของ pulse

ที่ป้อนเข้า motor ได้

การเปรียบเทียบความเร็วของ stepping motor จากแบบ stepping rate ซึ่งให้ค่าที่ไม่มาตรฐาน (ขึ้นอยู่กับจำนวน step ของ stepping motor แต่ละตัว) มาเป็นแบบจำนวนรอบต่อวินาที (rotational speed) จะได้ความสัมพันธ์คือ

$$n = 60f/s \quad \dots (2.2)$$

โดยที่ $n = \text{rotational speed (r.p.m.)}$

$f = \text{stepping rate}$

$s = \text{step number}$

2.4 ลักษณะการทำงานของ stepping motor

ตามปกติ stepping motor แบ่งออกเป็นหลายประเภท เช่น

1. Variable reluctance stepping motor (VRSM)
2. Permanent magnet stepping motor (PMSM)
3. Hybrid stepping motor (HSM)
4. Hybrid stepping motor having a permanent magnet in the stator
5. Permanent magnet motor with claw-poles
6. Linear stepping motor

แต่ stepping motor ที่นิยมใช้กันมากและนำมาใช้ใน project นี้คือ Variable-reluctance stepping motor และ Hybrid stepping motor

2.4.1 Variable-reluctance stepping motor (VR stepping motor)

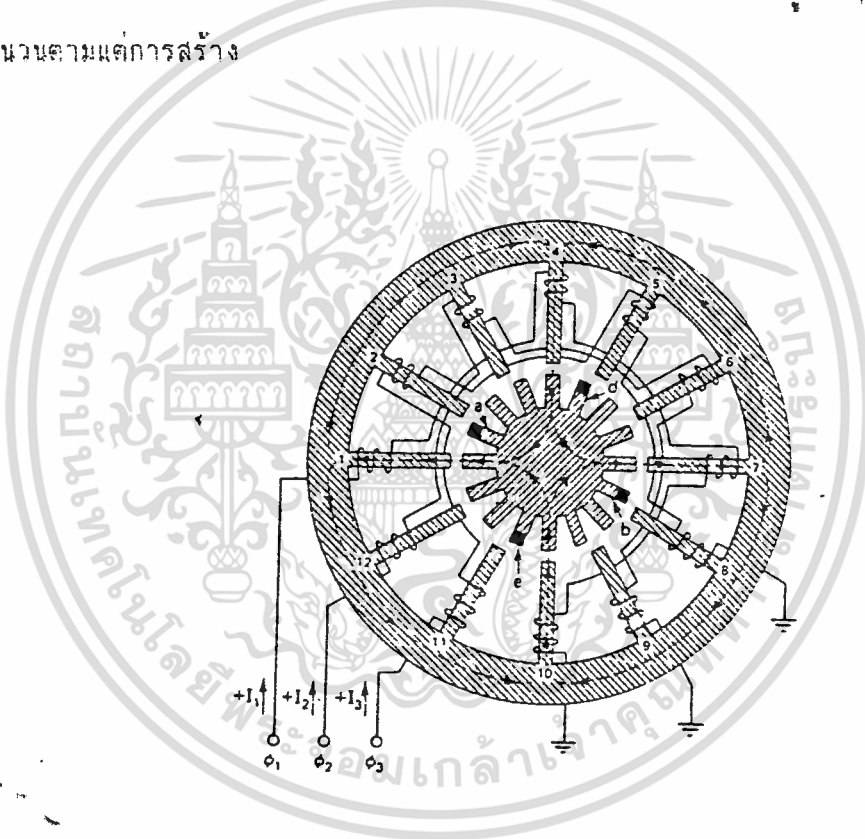
โครงสร้างของ stepping motor ชนิดนี้ แบ่งเป็น single-stack และ multi-stack โดยมีความแตกต่างกันเล็กน้อย แต่มีหลักการเหมือนกันคือ ทั้ง rotor และ stator จะถูกสร้างจากโลหะที่มีคุณสมบัติทางแม่เหล็ก (magnetic material) และไม่มีแม่เหล็กเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถาวรอยู่ใน stepping motor คุณสมบัติของโลหะที่มีคุณสมบัติแม่เหล็ก คือ จะพยายามปรับตัวเองให้อยู่แนวที่มีเส้นแรงแม่เหล็กผ่านมากที่สุด จากคุณสมบัตินี้เราสามารถนำมาทำ stepping motor ได้คือ เปลี่ยนแนวของเส้นแรงแม่เหล็กเป็นขั้น ๆ รอบวงกลมจะทำให้ rotor ปรับตัวเคลื่อนที่ตาม ทำให้เคลื่อนที่เป็นวงรอบได้

VR stepping motor แบ่งออกเป็น

1. single stack VR stepping motor

มอเตอร์ประเภทนี้จะมี rotor เพียงตัวเดียว และมี pole อยู่รอบๆ มอเตอร์ ซึ่งจะมีจำนวนตามแต่การสร้าง



รูป 2.2 โครงสร้างภายใน VRSM แบบ single stack

โดยที่ขั้วลวดแต่ละเฟสจะพัน pole 4 ตัว เช่น ในเฟส 1 จะพัน pole 1,4,7,10 ของ stator ทำให้มี pole 12 ที่ และมี rotor pole 16 ที่ ในการทำงานจะต้องใช้ rotor 4 ที่ต่อ 1 เฟส pole ที่อยู่ตรงข้ามกันจะพันขลวดกลับกัน เพื่อให้เกิดความสมดุลย์ของเส้นแรงแม่เหล็ก เมื่อเกิดการไหลเข้าและออกจาก rotor เช่น เมื่อเฟส 1 ถูกกระตุ้นจะเกิดเส้นแรงแม่เหล็กไหลออกจาก pole 10 และ pole 4 ผ่านไปยัง rotor ก่อนที่จะออกจาก

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rotor ที่ pole ที่ 1,7 ทำให้เกิดการ วง loop โดยผ่าน stator flame จึงทำให้ฟันของ rotor จะอยู่ตรงกับ pole ที่เกิดเส้นแรงแม่เหล็กมากที่สุด เพื่อให้เกิดช่องว่างระหว่างฟันของ pole และฟันของ rotor น้อยที่สุด ทำให้ค่า reluctance น้อยที่สุด และเกิด flux ผ่านตัว rotor มากที่สุด ซึ่งถ้าเราต้องการให้ rotor หมุนไปในทิศทางตามเข็มนาฬิกา เราจำเป็นต้องกระตุ้นเฟสที่ 3 หลังจาก that 1, หยุดป้อน ทำให้ pole ที่ 2,5,8,11 เกิดการไหลเวียนของเส้นแรงแม่เหล็ก ทำให้ฟัน a ของ rotor ซึ่งอยู่ใกล้ pole 2 มากที่สุดเคลื่อนที่เข้ามา ในขณะที่ฟัน b,d,c ก็จะเคลื่อนที่เข้า pole ที่ 5,8,11 ตามลำดับซึ่งมีเส้นแรงแม่เหล็กหนาแน่นที่สุด จึงทำให้ rotor มีการเคลื่อนที่ตามเข็มนาฬิกา การบังคับ stepping motor ตัวนี้ให้เคลื่อนที่ตามเข็มนาฬิกาจำเป็นต้องป้อนกระแสไปเรียงตามลำดับดังนี้ 1-3-2-1 และถ้าต้องการให้เคลื่อนที่ทวนเข็มนาฬิกาจำเป็นต้องป้อน 1-2-3-1

เมื่อทราบการทำงาน เราจะสามารถคำนวณมุมและความเร็วของ stepping motor ได้จากสูตรต่อไปนี้

กำหนด

- N_r = จำนวนของซี่ rotor
- N_s = จำนวนของซี่ stator
- N_p = จำนวน เฟส
- e_s = มุมของแต่ละ step
- R_s = stepping rate (step/revolution)

จะเห็นว่าถ้าเฟสต่าง ๆ ของ VRSM นี้ถูกกระตุ้นเป็น sequence จำเป็นต้องใช้ N_p step เพื่อให้ฟันของ rotor มีการเคลื่อนจาก pole เดิมไปสู่ pole ใหม่ ดังรูป 3.

จะได้ว่า

$$e_s = P_r / N_p = 360 / N_r N_p \quad \text{องศา/step} \quad \dots (2.3)$$

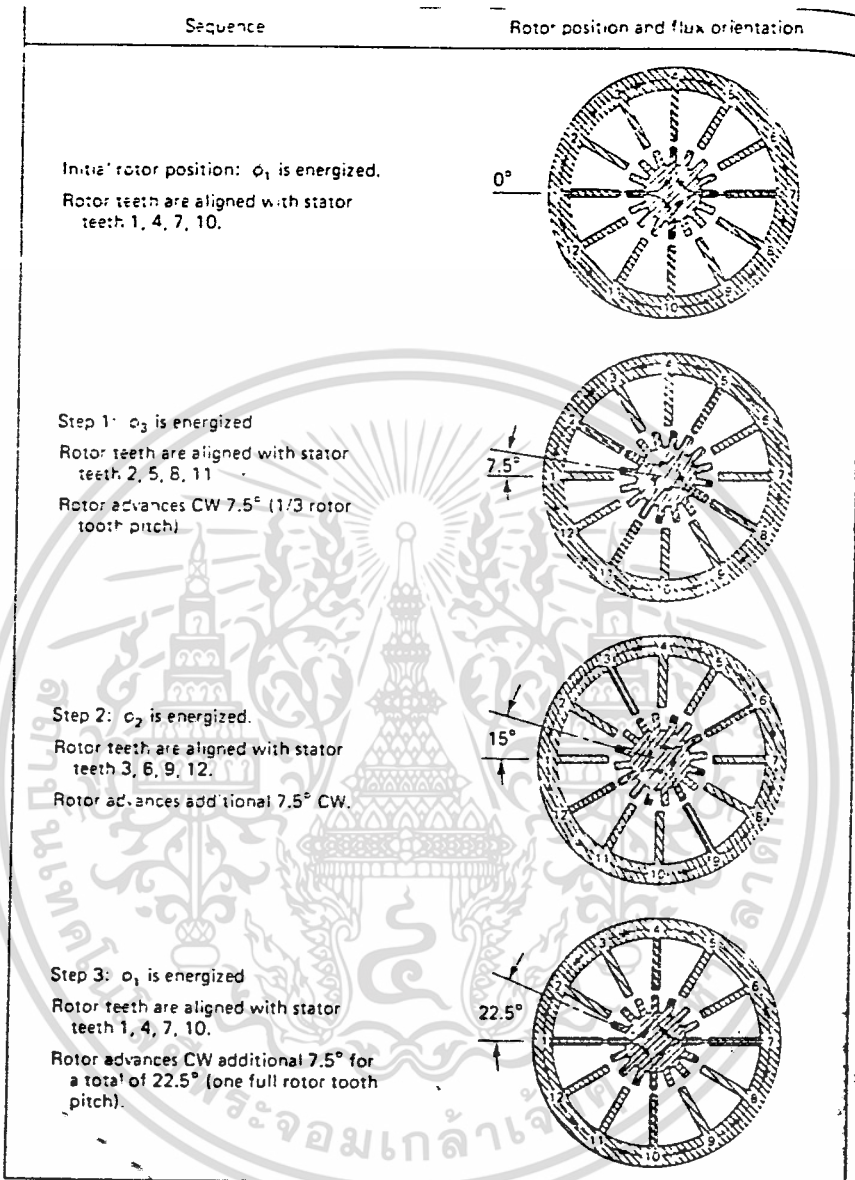
จากที่ 1 รอบ เท่ากับ 360 องศา

$$R_s = 360 / e_s = N_r N_p \quad \text{step/รอบ} \quad \dots (2.4)$$

ถ้ามีการป้อนความถี่ของ pulse = f และให้ stepping motor เคลื่อนที่ 1 step per 1 pulse

$$\text{เราจะได้ speed}(\omega) = 60f / R_s = 60f / N_r N_p \quad \text{rpm} \quad \dots (2.5)$$

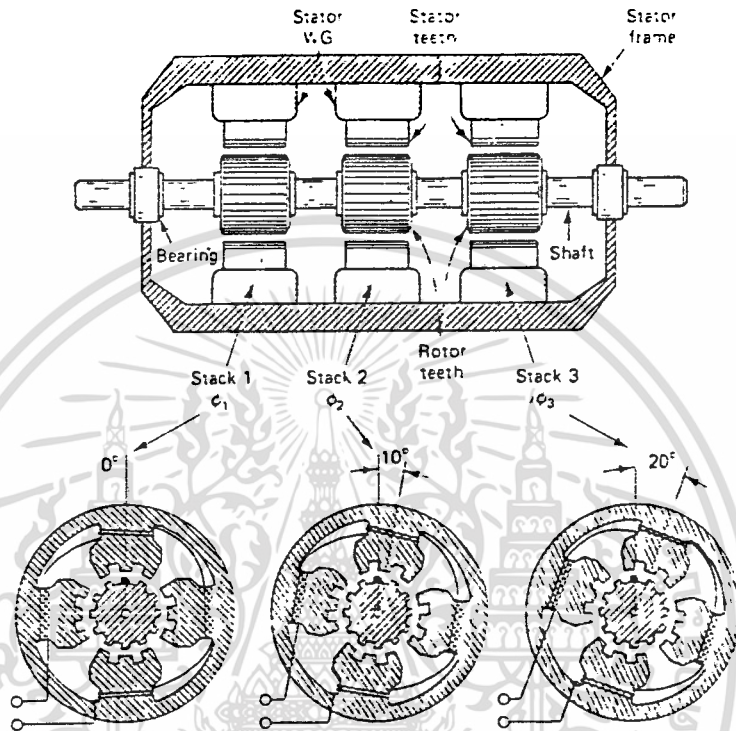
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.3 แสดงการเคลื่อนที่ของ rotor และทิศทางของ flux ในแต่ละ step

2. Multistack variable-reluctance stepping motor

stepping motor ในลักษณะนี้จะมี stack หลายตัว โดยที่จำนวน stack จะขึ้นอยู่กับจำนวน phase ดังได้แสดงในรูปที่ 2.4



รูป 2.4 โครงสร้างภายในของ multi stack VRSM

จากภาพตัดของ stepping motor จะเห็นได้ว่า โครงสร้างของ stepping motor จะประกอบด้วยชั้น 3 ชั้น โดยที่แต่ละชั้นจะประกอบด้วย stator และ rotor ของตัวมันเอง stator แต่ละชั้นจะถูก winding จากขดลวดในเฟสเดียวกัน ดังรูป จำนวนชั้นจะเท่ากับจำนวนเฟสของ stepping motor และใน stator ยังประกอบด้วย pole 4 ตัว แต่ละตัวประกอบด้วยซี่ฟันอีก ซึ่งต่างจาก single stack ที่แต่ละ pole จะมีฟันอยู่ที่เดียว ข้อแตกต่างอีกอย่างหนึ่งคือ จำนวนซี่ของ stator และของ rotor จะเท่ากันได้ ซึ่งใน single stack จะไม่สามารถเท่ากันได้ มิฉะนั้น step angle จะเท่ากับ 0

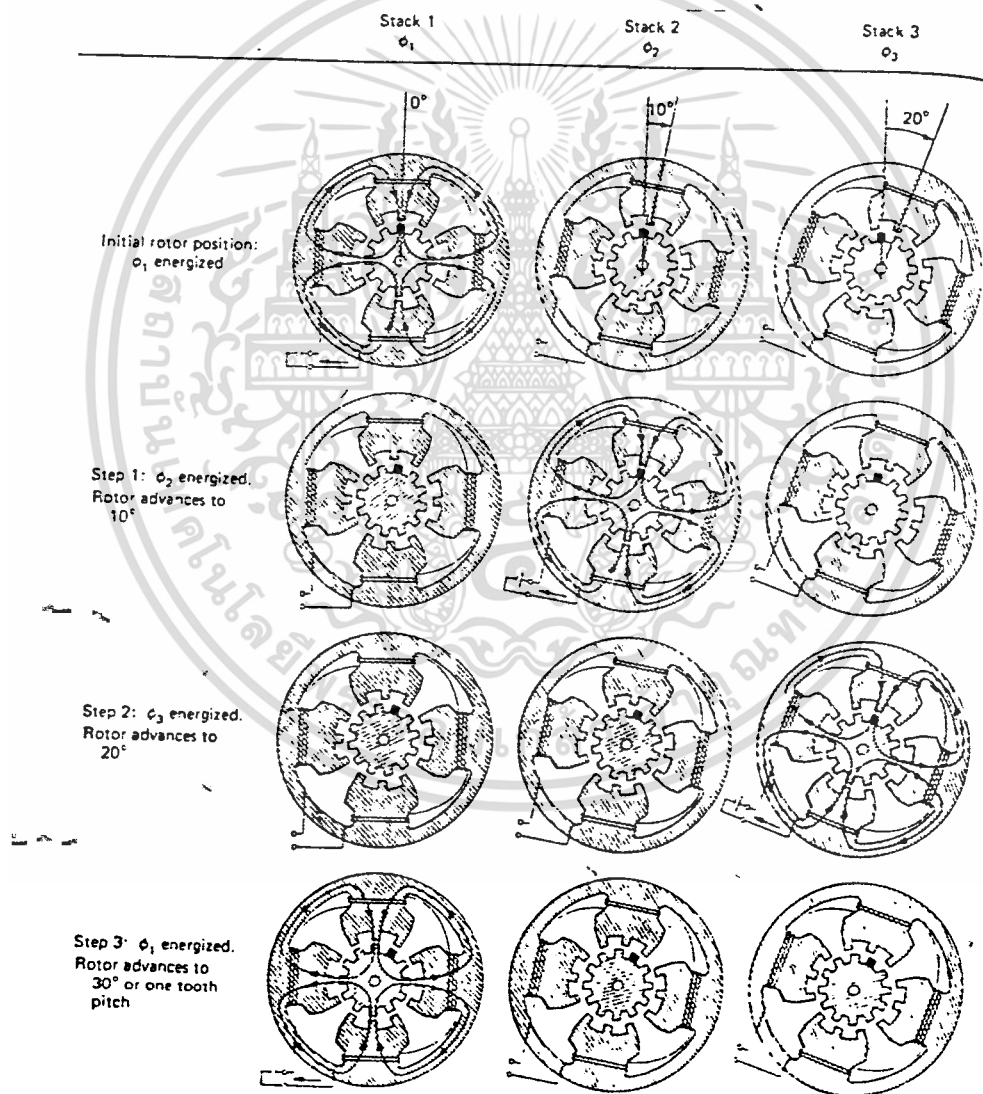
ดังนั้น วิธีการที่จะทำให้ step angle ไม่เท่ากับ 0 คือ ในส่วนของ stator แต่ละตัวจะมีซี่ที่เบี่ยงเบนต่างกันไป ดังรูปบน สีค้ำล่างนี้ ใน stack นี้แรกซี่ของ rotor ที่ค้ำเอกสารนี้จะเบี่ยงเบนไปจากตำแหน่งเดิมของ rotor ที่ค้ำเอกสารนี้ไป 10 องศา ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกระบายสัปดาห์จะตรงกัน ใน stack ที่สอง ซึ่งของ rotor จะเบี่ยงเบนออกจากซี่ของ rotor ประมาณ 10 องศา เช่นเดียวกัน ใน stack ที่ 3 ที่เบี่ยงเบนไป 20 องศา เป็นผลให้เกิด step angle ขึ้น

$$\theta_s = P_r / N_p \dots (2.6)$$

โดยที่ P_r คือ ระยะห่างระหว่างซี่ของ rotor

N_p คือ จำนวนซี่ของ pole หรือ rotor (จะเท่ากัน)



รูป 2.5 แสดงการเคลื่อนที่ของ rotor และทิศทางของ flux ใน multi-stack VRSM เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ multi- stack VRSM จะคล้ายกับ single stack VRSM ดังรูปที่ 2.5 นั่นคือใน step 0 phase 1 จะถูกกระตุ้น ทำให้ air gap ของซี่ pole ต่าง ๆ ของ stack 1 จะตรงกับซี่ของ rotor เช่นเดียวกับใน single stack และใน step ที่ 2 สี่ของ rotor จะเคลื่อนที่ไปตามทิศทางตามเข็มนาฬิกา เพราะใน stack ที่ 2 pole ต่าง ๆ จะถูกกระตุ้นให้เกิดสนามแม่เหล็ก ทำให้ซี่ของ rotor ต้องเคลื่อนตัวเพื่อให้เกิด min-reluctance และเกิด flux สูงสุด จากนั้นก็จะเคลื่อนที่ไปใน step ที่ 3 ซึ่งทำงานเช่นเดียวกับ step ที่ 2 ซึ่งการควบคุมการเคลื่อนที่จะป้อนกระแสเข้าขดลวดเรียงลำดับจาก 1-2-3-1 ในทิศตามเข็มนาฬิกา และ 1-3-2-1 ในทิศทวนเข็มนาฬิกา

และจากคุณสมบัติของ stepping motor ที่นำมาใช้งานใน project นี้ เป็น stepping motor 4 phase ที่มี rotor teeth = 50 แบบ VRSM multi-stack

เพราะฉะนั้นเราจะได้

$$R = 200 \text{ step/revolution}$$

$$e = 1.8 \text{ องศา}$$

และ

$$w = 60f/200 = 0.3f \text{ rpm.}$$

2.4.2 Permanent - magnet stepping motor

จากรูปที่ 2.6

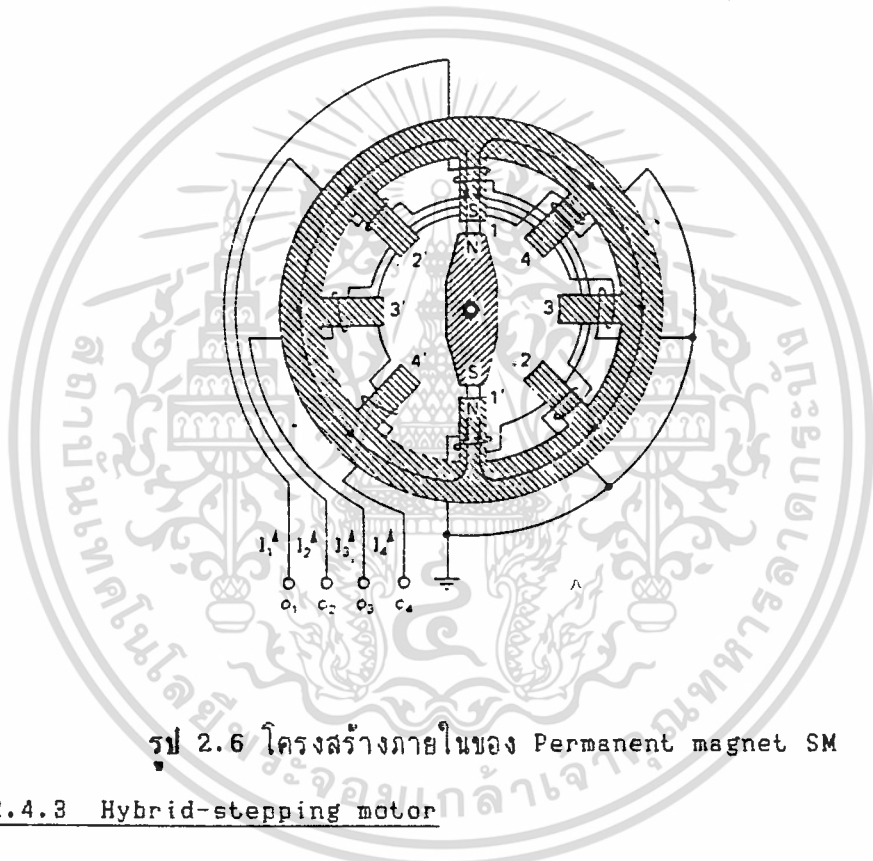
แสดงถึง โครงสร้างภายในของ PMSM ซึ่งเป็น stepping motor 4 เฟส และแต่ละเฟสจะไปพันรอบ pole 2 pole ทำให้เกิด pole ใน stator 8 pole และ rotor จะทำจากสารแม่เหล็กถาวรซึ่งตรงกับ pole 1 และ 1' ซึ่งถูกพันด้วยขดลวดที่มาจากเฟสเดียวกัน โดยขั้วของ pole 1 จะถูกตรึงจากสนามแม่เหล็กของ pole ทั้งสองและ rotor จะเคลื่อนที่จาก pole 1 ไปตามแต่ลักษณะการกระตุ้น เช่น 1-4-3-2 จะทำให้หมุนตามนาฬิกา แต่หลังจากเคลื่อนที่ไป 180 องศาแล้ว จำเป็นต้องกระตุ้นเฟส 1-4-3-2 ในทางตรงข้าม เพื่อให้ stepping motor เคลื่อนที่ได้ครบรอบ

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้เพื่อใช้ในการศึกษาวิจัยเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร หากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำได้ยากเพราะ step angle จะขึ้นกับจำนวน pole เพียงอย่างเดียว

$$\theta_s = 360/N_p \quad \text{step/rev.} \quad \dots(2.7)$$

แต่ประโยชน์ของ stepping motor ชนิดนี้ คือใช้ในการทำนาฬิกา หรืองานที่ไม่ต้องการมุมระหว่าง step ละเอียตมากนัก



รูป 2.6 โครงสร้างภายในของ Permanent magnet SM

2.4.3 Hybrid-stepping motor

Hybrid stepping motor เป็นการผสมผสานรูปแบบของ PM และ VR stepping motor ซึ่งโครงสร้างของ hybrid stepping motor นี้จะประกอบด้วย section ซึ่งต้องมี 2 section เป็นอย่างน้อย โดยแต่ละ section จะประกอบด้วย rotor และ stator และในระหว่าง rotor จะมีแกนเชื่อมต่อกัน ซึ่งทำจากสารแม่เหล็กถาวร

จำนวนฟันของ rotor และ stator จะไม่เท่ากัน ซึ่งต่างจาก multi-stack VRSM ซึ่งฟันของ stator และ rotor เท่ากัน ส่วนโครงสร้างของ a และ b จะเหมือน

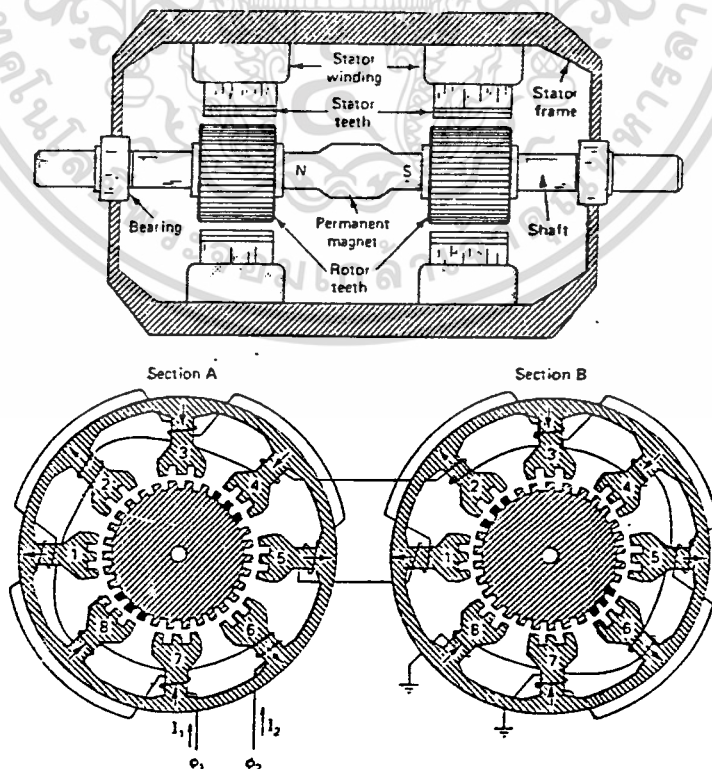
กัน แต่จะแตกต่างที่ตำแหน่งการวางของฟันใน rotor ทั้งสอง โดยที่ฟันของ rotor a จะ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบี่ยงเบนจากฟันของ rotor b อยู่ $(1/2)*P_r$ (จะแตกต่างจาก multi stack ซึ่งฟันของ rotor ทุกตัวจะตรงกัน แต่ตำแหน่งฟันของ pole จะเบี่ยงเบนกัน) เช่น $P_r = 12$ องศา ฟันของ rotor จะเบี่ยงเบน 6 องศา

การฟันขดลวดของ stepping motor จะมีลักษณะกระจายไปทั้งสอง section โดยเฟส 1 จะฟันกับ pole ที่ 1,3,5,7 ของ section a และบน pole ที่ 1,3,5,7 ของ section b และเฟสที่สองจะฟันใน pole ที่ 2,4,6,8 ของแต่ละ section

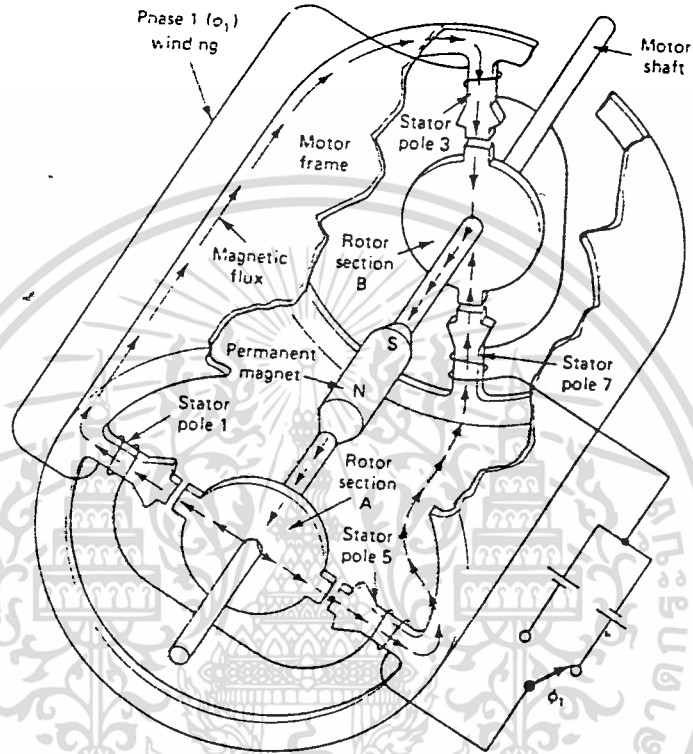
แกนแม่เหล็กที่ rotor ของ section a จะเป็นขั้วเหนือ และที่ section b เป็นขั้วใต้ จากการที่ทำการกระตุ้นทั้ง 2 section พร้อมกัน และ rotor แต่ละ section จะถูกเคลื่อนที่ภายในสนามแม่เหล็กคนละส่วนจึงจำเป็นต้องตัดส่วนของ rotor ให้แยกกัน แม่เหล็กถาวรจะเป็นตัวกำเนิด flux ในส่วนของ rotor ให้มีการไหลในทิศทางเดียวและจะไหลออกจาก rotor ที่ตำแหน่งของ pole ที่ถูกกระตุ้นให้มี reluctance ต่ำสุด

ปริมาณของ flux ที่ผ่าน stator pole จะเกิดจากการเหนี่ยวนำกระแสเข้าเฟสของ pole เหล่านี้ โดยทิศทางของ flux ของ stator pole จะขึ้นกับทิศทางของกระแสที่เข้าขดลวดของเฟส (ใช้กฎมือขวา) ดังรูปที่ 2.7



รูป 2.7 โครงสร้างภายในของ Hybrid stepping motor
เอกสารนี้เป็นเอกสารลับ ไม่ว่ากรณิใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาภาวะ เมื่อ ϕ_1 ถูกกระตุ้นโดย I_1 ที่ถูกป้อนในทิศทางตามรูป ฟันของ rotor ใน section a จะตรงกับฟันของ stator ของ pole 1 และ 5 ใน section b ฟันของ rotor จะตรงกับฟันของ stator ที่ pole 3 และ 7 ดังแสดงในรูปที่ 2.7



รูป 2.8 แสดงทิศทาง flux แม่เหล็กเมื่อ Phase 1 ถูกกระตุ้น

flux จากขั้วเหนือของแม่เหล็กถาวรจะเข้าสู่ rotor ของ section a และออกจากรotor ผ่านทาง pole 1 และ 5 (pole 1 และ 5 ถูกกระตุ้นจาก I_1 ในทิศทางที่ก่อให้เกิด reluctance น้อยที่สุด ในการให้ flux ไหลออกจาก section a) flux จะเคลื่อนที่ผ่านตัวตั้งของ stator และเข้าสู่ rotor ของ section b ผ่าน pole 3 และ 7 (I_1 จะกระตุ้น pole 1 และ 5 ให้มีค่า reluctance น้อยที่สุด เพื่อให้ flux ผ่านเข้า rotor ของ section b) เป็นผลให้เกิดเป็น loop ที่สมบูรณ์ที่ขั้วใต้ของแม่เหล็กถาวร โดยที่ flux ของแม่เหล็กถาวรจะเสริมกำลังกับ flux ที่มาจากขดลวด

เพื่อจะให้ stepping motor เคลื่อนไปทิศทางตามเข็มนาฬิกา (cw) I_1 จะหยุดป้อน และ I_2 จะป้อนเข้าเฟส 2 เหตุที่ ϕ_2 เมื่อถูกกระตุ้นหลังทำให้ stepping

motor เคลื่อนไปทาง cw ได้ เราจะสังเกตที่ตำแหน่งฟันของ stator และ rotor ทั้งสองส่วน ในรูป 2.7

สังเกตฟันที่ขยายสีดำเป็นหลัก ฟันสีดำจะอยู่ใกล้จะไปตรงกับฟันของ stator pole 4 และ 8 ใน section a และ pole 2 และ 6 ใน section b แต่จะเกิดปัญหาหากเราป้อน I_1 ต่อจากเลิกป้อน I_2 แล้วในทิศทางเดิม จะทำให้ stepping motor หมุนผิดทาง เพราะเกิดการกระตุ้นที่ pole 1 และ 5 อีก ทำให้ rotor ย้อนกลับ ทำให้เราจำเป็นต้องป้อน I_1^- (เป็นการกลับทิศของ I_1^+ ทำให้การกระตุ้นจะเกิดใน pole ที่ 3 และ 7 แทน ทำให้ reluctance ที่ pole ทั้งสองมีค่าน้อย rotor จะเคลื่อนเข้าหา) ดังนั้นการ switch เพื่อให้ stepping motor ทำงานอย่างถูกต้องในทิศทางเข็มนาฬิกา จะแสดงดังรูปที่ 2.9

Step	I_1	I_2	Flux out sec. A pole nos.	Flux in sec. B pole nos.	Section A	Section B
1	+		1, 5	3, 7		
2		-	4, 8	2, 6		
3	-		3, 7	1, 5		
4		+	2, 6	4, 8		

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าเราจำเป็นต้องป้อน sequence เป็น $I_1^+, I_2^-, I_1^-, I_2^+, I_1^+$ ตามลำดับ หากป้อนกลับกันจะได้การเคลื่อนที่ทวนเข็มนาฬิกา

จะเห็นได้ว่า การที่ฟันเคลื่อนไป 1 ซี่ ต้องให้ pulse 4 steps

$$\text{step angle} = (1/4) * P_r$$

$$\text{ดังนั้น } \theta = P_r / 4 = 360 / 4N_r = 90 / N_r \quad \dots(2.8)$$

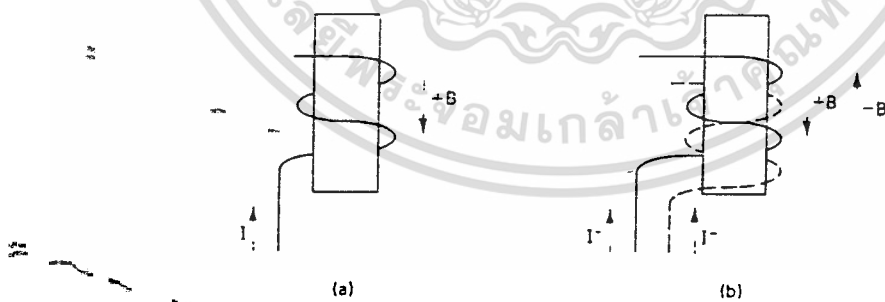
$$= \left| P_r - P_r \right| \quad \dots(2.9)$$

stepping motor ที่ใช้งานอยู่ มี $N_r = 50$

เพราะฉะนั้น

$$\theta = 1.8 \text{ องศา}$$

จากที่กล่าวมาการทำงานของ hybrid stepping motor จำเป็นต้องใช้กระแสที่ป้อนให้เฟสทั้งสองทิศทาง ทำให้ต้องมี power supply 2 ชุด หรือใช้ bipolar driver ทำให้สิ้นเปลืองค่าใช้จ่าย ดังนั้นนิยมจะใช้ power supply ชุดเดียวมากกว่า ซึ่งมีความแตกต่างกันในการนำขดลวดนั้นคือ เมื่อใช้ power supply แบบ bipolar driver เราใช้มอเตอร์แบบ unipolar ใช้การกลับทิศกระแสไปเป็นตัวเปลี่ยนทิศของ flux ในแกนเหล็ก



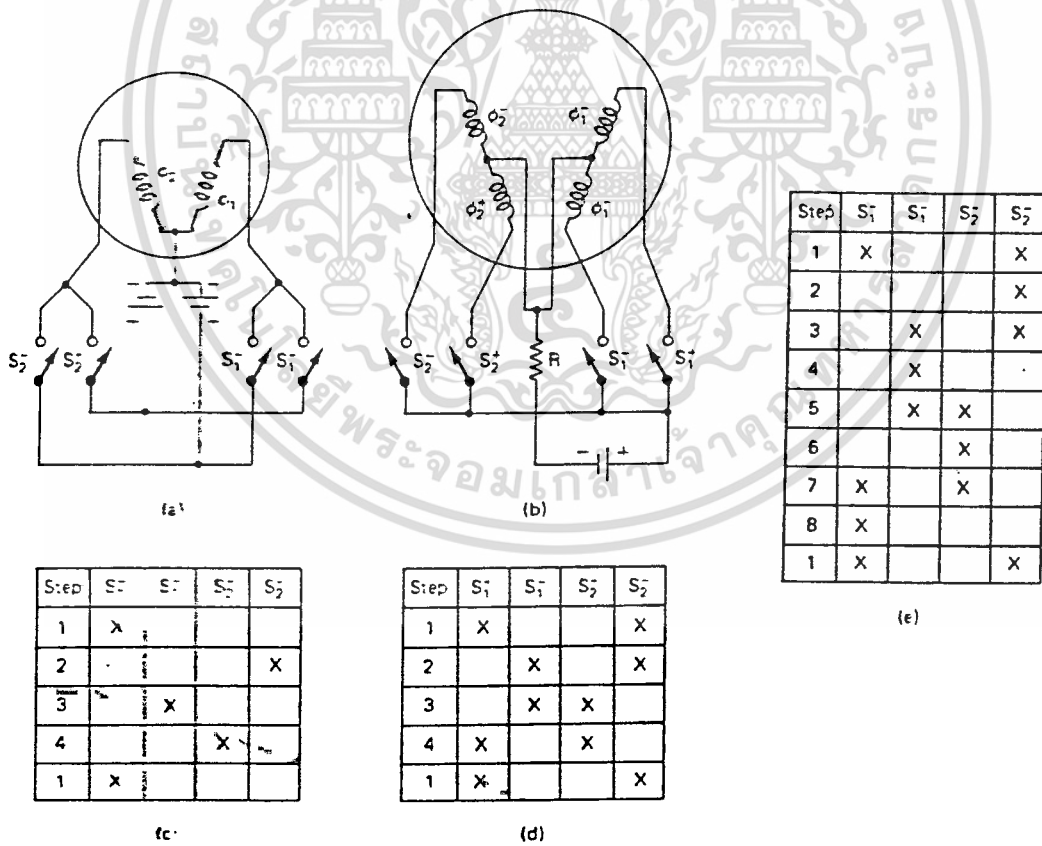
รูป 2.10 a) การพันแบบ unifilar b) การพันแบบ bifilar

การพันแบบ bipolar จะประกอบด้วยขดลวด 2 ชุด พันทับกันในทิศทางตรงกันข้าม โดยการป้อนกระแส I^+ เหนือหน้าขดลวดให้เกิดขั้วเหนือที่ด้านล่าง และเกิดขั้วแม่เหล็ก B^+ ในทิศทางลง แต่เมื่อมีการป้อนกระแส I^- ไปตามเส้นประ ผลที่ได้จะทำให้ทิศ

ทางการเพิ่มขั้วนำและทิศของสนามแม่เหล็ก B ถูกกลับทิศ

ถ้าสมมติว่า hybrid stepping motor ในรูปที่ 1 เป็นการพันแบบ bifilar ขดลวด ϕ_1 จริงๆ จะถูกแทนด้วยขดลวด 2 ขด คือ ϕ_1^+ และ ϕ_1^- ซึ่งพันในทิศตรงข้าม และ ϕ_2^+ และ ϕ_2^- ที่พันในทิศทางตรงข้ามเช่นกัน ซึ่งทำให้เราได้ stepping motor ที่มี 4 เฟส โขลแต่ละเฟสจะถูกควบคุมด้วยกระแสขั้วเดียวกัน

จากรูปที่ 1 จะแสดงวงจรและการ switch ของ stepping motor วงจร a) เป็น 2 เฟส hybrid stepping motor ที่ใช้ bipolar drive และ วงจร b) เป็น 4 phase hybrid stepping motor ที่ใช้ unipolar drive ลำดับการ switch 4 ขั้นก็จะทำให้ stepping motor หมุนตามเข็มนาฬิกาด้วย 1 phase-on, 2 phase-on หรือการ switch แบบ half stepping eight step sequence ซึ่งจะใช้ step angle ที่เล็กกว่า

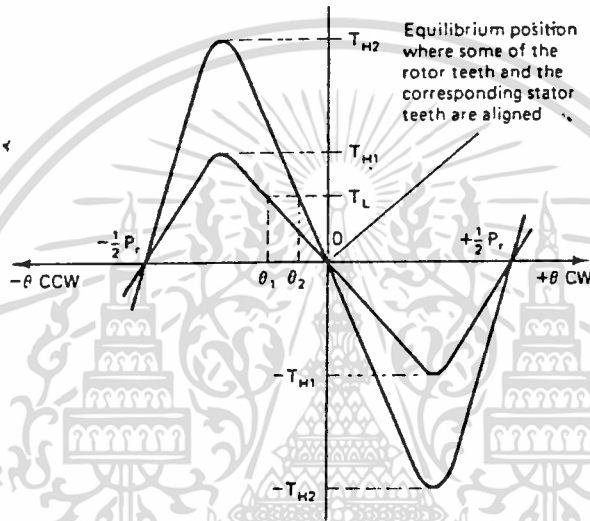


รูป 2.11 a) วงจรการ switch ของ unipolar b) วงจรการ switch ของ bipolar

เอกสารนี้เป็นเอกสารเพื่อการ switch การใช้งาน 1 phase on d) 2 phase on e) half step ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ลักษณะเฉพาะของแรงบิดสถิตย์ (static torque) ใน stepping motor

เมื่อมีฟันบางซี่ของ rotor อยู่ในแนวเดียวกับ stator จุดนั้นจะเป็นจุดสมมูลของ rotor มันจะอยู่ในตำแหน่งนั้นจนกว่ามันจะได้รับสัญญาณกระตุ้นเฟสอีก จุดที่ rotor สมมูลคือจุด 0 ที่อยู่ในรูปที่ 2.12



รูป 2.12 ลักษณะเฉพาะของแรงบิดสถิตย์ของ stepping motor

ที่จุดนี้ stepping motor จะไม่มีแรงบิดเกิดขึ้น ถ้า rotor ถูกหมุนในทิศทางทวนเข็มนาฬิกาด้วยค่ามุม θ_1 โดย load ภายนอก (T_L) stepping motor จะสร้างแรงบิดที่มีค่าเป็นบวกในทิศทางทวนเข็มนาฬิกาเพื่อปรับจุดสมมูลกับ load แต่ถ้า rotor ถูกหมุนในทิศทางตามเข็มนาฬิกา stepping motor ก็ จะสร้างแรงบิดย้อนกลับในทิศทางตรงข้าม ด้วยเหตุนี้แรงบิดที่ stepping motor นี้จะถูกเรียกว่า แรงบิดส่งคืน (restoring torque) ซึ่งจะอยู่ในทิศทางเข้าหาจุดสมมูล ความสัมพันธ์ระหว่างผลของแรงบิดกับการเคลื่อนที่ของ rotor จะมีลักษณะเป็น sine-wave (รูปร่างที่แท้จริงจะขึ้นอยู่กับโครงสร้างของ rotor กับ stator) ซึ่งมีความยาวคลื่นเท่ากับความห่าง (pitch) ของฟัน rotor ซึ่งมีค่าเท่ากับ P_r

แรงบิดสูงสุดคือแรงบิดสูงสุดของ stepping motor ถูกสร้างขึ้นจากกระแสที่ป้อนให้เฟสคลวด แรงบิดที่ว่านี้คือ T_H ซึ่งก็คือแรงบิดสถิตย์ (static torque) หรือแรงบิดที่

ทำให้หยุด (holding torque) ค่าของแรงบิดสูงสุดนี้จะหาได้จาก data sheet โดยดูตามระดับกระแสคงที่ แม้ว่าในบางกรณีแรงบิดที่ stepping motor สร้างขึ้นนี้จะเพิ่มขึ้นอย่าง linear กับกระแสเฟส แต่แรงบิดก็จะถึงจุดอ้อมตัวที่ที่ระดับกระแสคงที่ที่ทำให้เส้นแรงแม่เหล็กของ rotor กับ stator อ้อมตัว แต่หากกระแสเพิ่มขึ้น จะไม่เกิดแรงบิดเพิ่มขึ้น

แรงบิดที่ stepping motor สร้างขึ้นเป็นศูนย์เมื่อ rotor เคลื่อนที่ออกไป $(1/2)P_r$ และถ้าหากเคลื่อนที่ต่อไป rotor จะเข้าสู่จุดสมมูลย์ที่อยู่ถัดไป หรือเข้าสู่จุดอื่นที่มีระยะเท่ากับจำนวนเท่าของความห่างของฟัน rotor เมื่อเหตุการณ์นี้เกิดขึ้น ความคล่องจองกัน ระหว่าง pulses ที่ป้อนกับ steps ของ stepping motor (ซึ่งมีความสำคัญในด้านความแม่นยำของตำแหน่ง) จะสูญเสียไป

เพื่อให้แน่ใจว่าสิ่งนี้จะไม่เกิดขึ้น ควรเลือก stepping motor ที่มีแรงบิดสถิตย์มากกว่าแรงบิดของ load

เมื่อเอา load ภายนอกออก rotor จะเคลื่อนเข้าสู่จุดสมมูลย์เดิม (ในทิศทางที่แสดงคือตามเข็มนาฬิกา) ซึ่งคือที่ 0 stepping motor จะสร้างแรงบิดที่เพิ่มขึ้นทีละน้อย จนถึงจุดที่แรงเสียดทานภายในของ stepping motor จะมากกว่าแรงบิดที่ stepping motor สร้างขึ้น rotor จะหยุดที่จุดนี้ซึ่งจะทำให้มันไม่ถึงจุดสมมูลย์ที่แท้จริง ความไม่แน่นอนในตำแหน่งของ rotor ที่ใกล้สมมูลย์ โดยทั่วไป จะถูกเรียกว่า ความคลาดเคลื่อนของตำแหน่ง ขณะไม่มี load ซึ่งปกติแล้วจะน้อยกว่า 1 องศา ความคลาดเคลื่อนนี้จะไม่เป็นค่าสะสม มันจะเท่ากันหมดไม่ว่าจะเคลื่อน 1 step หรือ 1000 steps

ตามรูป stepping motor ที่มีแรงบิดสถิตย์ T_{H1} จะทำให้ตำแหน่งคลาดเคลื่อนด้วยค่า θ_1 และ stepping motor ที่มีค่า T_{H2} จะสร้างความคลาดเคลื่อน θ_2 ถ้า $\theta_1 > \theta_2$ ความคลาดเคลื่อนจะน้อยลงถ้า load มีค่าน้อยลงหรือ ใช้ stepping motor ที่มีค่าแรงบิดสถิตย์มากขึ้น

ความคลาดเคลื่อนในตำแหน่งโดยที่มี load ภายนอกจะสามารถประมาณได้จากการสมมุติว่ารูปร่างของแรงบิดเป็นรูป sine เมื่อความยาวคลื่นของคลื่น sine คือ 2π และความยาวคลื่นของแรงบิดที่เกิดขึ้นคือ $P_r \cdot \theta$ จะถูกแทนด้วยค่า $(2\pi/P_r)\theta = N_r \cdot \theta$ เมื่อ θ คือ การเคลื่อนที่เชิงมุมของ rotor จากจุดสมมูลย์ และ N_r คือจำนวนฟันของ rotor การเท่ากันของ T กับ T และ θ กับ θ ความคลาดเคลื่อนของตำแหน่งจะคำนวณได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_L = -T_H \sin N_r \theta_r \quad \dots(2.10)$$

สมมติว่า $T_H = 40 \text{ N/m}$ สำหรับ hybrid stepping motor 4 phase ที่มี $N_r = 30, T_L = 12 \text{ N/m}$ ที่ถูกป้อนให้เคลื่อนที่ตามเข็มนาฬิกา ความคลาดเคลื่อนด้านตำแหน่ง อาจถูกคำนวณได้ดังนี้

$$\theta_r = (\sin^{-1}(-12/40))/30 = 0.58^\circ \quad \dots(2.12)$$

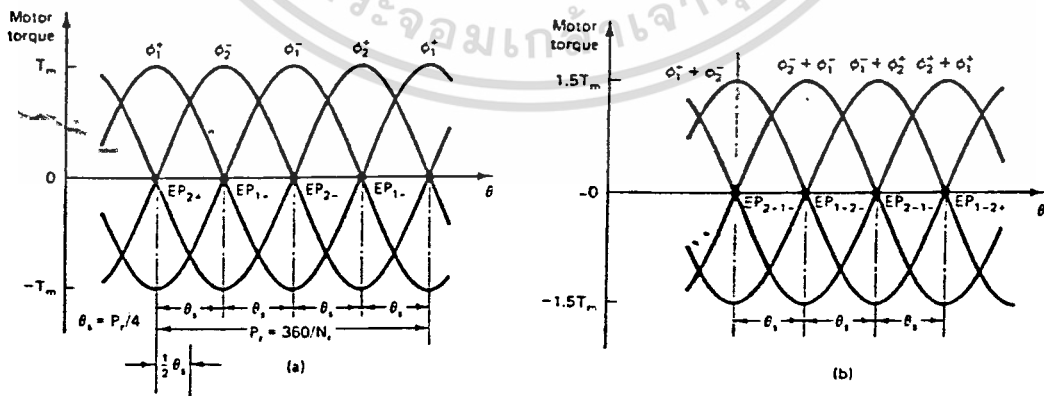
ความคลาดเคลื่อนของตำแหน่งโดยประมาณคือ 19 % ของ step angle ซึ่งมีค่า 3 องศา และ 7% ของความห่างของฟัน rotor ซึ่งเท่ากับ 12 องศา ความคลาดเคลื่อนของตำแหน่งเพิ่มเป็น 1.77 องศาสำหรับ stepping motor ที่มีค่า $T_H = 15 \text{ N/m}$

ความคลาดเคลื่อนของตำแหน่งอาจหาได้จากการประมาณเส้นตรงที่เป็นส่วนของรูปคลื่นของแรงบิดซึ่งอยู่ระหว่างจุดสูงสุด 2 จุด ดังนี้

$$T_L = mT_H \quad \dots(2.13)$$

ซึ่งความชัน m หมายถึงความผิด เมื่อมี load ความคลาดเคลื่อนของตำแหน่งจะน้อยลงเมื่อ stepping motor มีความผิดมากขึ้น

การเลือกวิธีกระตุ้นเฟสสามารถมีผลกับความสามารถในการสร้างแรงบิดของ stepping motor อย่างมาก ดังที่กล่าวมาแล้วว่า stepping motor อาจถูกกระตุ้นที่ละเฟสหรือทีละ 2 เฟส ดังรูป



รูป 2.13 แสดงความสัมพันธ์ระหว่างแรงบิดกับตำแหน่ง

เอกสารนี้เป็นเอกสารที่สแกนใส่สำหรับเพื่อการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.13 แรงบิดของ stepping motor ซึ่งเปลี่ยนแปลงตามตำแหน่งของ rotor โดย

- การกระตุ้นเฟสเดียว
- การกระตุ้นทีละสองเฟสของ HSM แบบ 4 เฟส

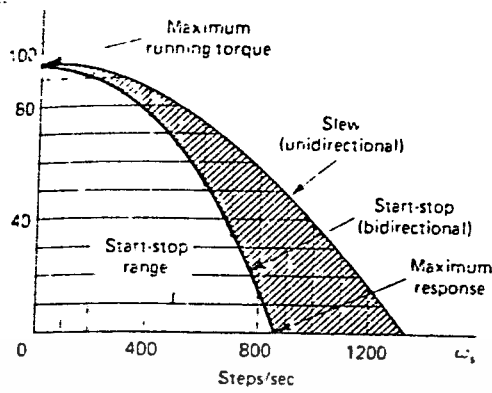
เมื่อแรงบิดของ HSM แบบ 4 เฟส แปรเปลี่ยนตามตำแหน่งของ rotor แม้ว่า การกระตุ้นเฟสแบบคู่จะให้แรงบิดมากกว่า 50% เมื่อเทียบกับการกระตุ้นทีละเฟส แต่กำลังที่ใช้ใน stepping motor จากแหล่งจ่ายไฟก็จะมากเป็นสองเท่าเช่นกัน กรณีนี้โดยทั่วไปเราจะเอาแรงบิดที่เพิ่มขึ้นแม้ว่าจะต้องจ่ายกำลังเพิ่มขึ้นก็ตาม อย่างไรก็ตาม การกระตุ้นเฟสแบบคู่จะไม่เปลี่ยน step angle แต่ จุดสมดุลย์ของ rotor จะเคลื่อนไปครึ่ง step angle

2.6 ลักษณะเฉพาะของแรงบิดจลน์ (dynamic torque) ของ stepping motor

เมื่อหัวข้อที่แล้ว เราพิจารณาแต่แรงบิดสถิตย์ซึ่งเมื่อ rotor อยู่ที่จุดสมดุลย์ rotor จะไม่หมุน แต่ความจริงแล้ว rotor ต้องหมุนเพื่อให้การเคลื่อนไหวที่เหมาะสมกับ load เมื่อมี load stepping motor จะเริ่มหมุนและเพิ่มความเร็วในการหมุน load เพื่อให้เข้าสู่ตำแหน่งที่ต้องการ ซึ่งทำให้ stepping motor ต้องสร้างแรงบิดที่มีค่าเพียงพอที่จะเอาชนะแรงเสียดทานและแรงเฉื่อยทั้งหมด

ในการเร่งแรงเฉื่อย stepping motor อาจต้องสร้างแรงบิดที่มีค่ามากหากต้องเร่งให้ load ถึงตำแหน่งที่ต้องการในเวลาอันสั้น การที่ stepping motor ไม่สามารถจะสร้างแรงบิดได้มากขนาดนั้นอาจทำให้ stepping motor ถูกหน่วง และทำให้สูญเสียความคล่องจอร์ระหว่าง steps ของ stepping motor ที่ได้กับการกระตุ้นเฟสและในที่สุดก็ทำให้ load เข้าสู่ตำแหน่งที่ไม่ถูกต้อง

ความสามารถในการสร้างแรงบิดของ stepping motor เป็นฟังก์ชันของการสร้างความเร็วซึ่งอาจหาได้จากรูปที่ 2.14



รูป 2.14 ลักษณะเฉพาะของ ความเร็วกับแรงบิดของ stepping motor

จากกราฟจะเห็นว่ากราฟจะมีการทำงานอยู่ 2 บริเวณคือช่วงเริ่ม-หยุดกับช่วง slew

ช่วงเริ่ม-หยุด อยู่ระหว่างแกน co-ordinate กับส่วนโค้ง เริ่ม-หยุด เมื่อมีการทำงานในช่วงนี้ stepping motor อาจหยุดและเริ่มหมุนใหม่หรือกลับทิศทางการหมุนโดยไม่เสีย step. coordinate ของแรงบิดสูงสุดที่ stepping motor สร้างขึ้นที่ความเร็วปกติภายในช่วงนี้จะเป็นเส้นโค้งเริ่ม-หยุด จุดตัดระหว่างเส้นโค้งนี้กับแกนตั้งจะถูกเรียกว่าแรงบิดสูงสุดขณะหมุน (the maximum running torque) ซึ่งที่เห็นในรูปนี้จะเท่ากับ 95 oz-in. และการตอบรับสูงสุดจะเป็นจุดที่เส้นโค้งเริ่ม-หยุดตัดกับแกนนอน (850 steps/s) ด้วยเหตุนี้ stepping motor ที่ไม่มี load สามารถทำงานที่ความเร็วสูงสุด 850 steps/s ถ้าหากแรงบิดที่ load คือ 50 oz-in. ดังนั้น stepping motor จะไม่สามารถมีความเร็วเกิน 600 oz-in. ถ้าหากมันยังทำงานอยู่ในช่วงนี้

ช่วง slew แสดงด้วยบริเวณที่เป็นเงาระหว่างเส้นโค้งเริ่ม-หยุดกับเส้นโค้ง slew ถ้าหากstepping motor ทำงานในช่วงนี้ มันต้องไม่หยุดหรือเริ่มหมุนหรือเปลี่ยนทิศ นี่เป็นช่วงการทำงานที่มีทิศทางเดียวซึ่งจะไม่มี การสูญเสีย steps ระหว่างหมุน ไม่ควรให้การทำงานอยู่นอกช่วง slew ออกไป

การสร้างการทำงานภายในช่วง slew ก่อนอื่น stepping motor ต้องทำงานในช่วงเริ่ม-หยุด แล้วจึงค่อยใช้ acceleration ramp ที่ควบคุมได้ดังรูป ทำให้มีการผ่านช่วง slew ในการนี้ stepping motor ต้องสร้างแรงบิดเพิ่มเพื่อเร่งแรงเฉื่อยทั้งหมดรวมทั้ง rotor และ load จาก w_{1} สู่ w_{2} ในช่วงเวลา Δt เราจะใช้สมการต่อไป

นี้คำนวณหา แรงบิดเฉื่อย (inertia torque) ซึ่ง $w_{\text{ท่า}} = dw/dt$ อาจแทนด้วย $\Delta w / \Delta t$ การคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น

$$T_s = J_T (\Delta \omega_s / \Delta t) = J_T (\Delta \omega_s / \Delta t) (1/R_s) 2\pi \text{ oz-in.} \dots (2.14)$$

ซึ่ง J_T = แรงเฉื่อยทั้งหมดรวมทั้ง rotor และ load มีหน่วย oz-in.^2

$$\Delta \omega_s = \omega_{s2} - \omega_{s1} \text{ มีหน่วย steps/s}$$

R_s = stepping rate = $360/e_s = N_r N_s$ มีหน่วย steps/s และ 2π rad/rev เปลี่ยน ω_s จาก steps/s เป็น rad/s

การผ่านช่วงนี้ต้องถูกสร้างภายใต้ภาวะความเร่งที่ถูกควบคุมจากช่วง slew ช่วงเริ่มหยุด ถ้าหาก stepping motor ต้องถูกหยุดโดยปราศจากการเสีย steps อาจสมมุติได้ว่ามัน ramp ขึ้นและลงในเวลาเท่ากัน ระหว่างช่วงเวลา ramp ลง แรงบิดที่สนองแรงเฉื่อยเป็นลบ ดังนั้นมันจึงช่วย stepping motor สร้างแรงบิด

จากกราฟลักษณะเฉพาะความเร็ว-แรงบิด แรงบิดที่ stepping motor สร้างขึ้นถูกเรียกว่า pull-out torque ซึ่ง rotor ต้องสร้าง 1 step สำหรับแต่ละ excitation pulse แต่ขดลวดเฟส pull-out torque เป็นแรงบิดสูงสุดที่ stepping motor สร้างขึ้น เมื่อแรงบิดที่ load มากกว่า pull-out torque ความสัมพันธ์ระหว่าง pulses ที่ป้อนกับ steps ของ rotor จะเสียไป

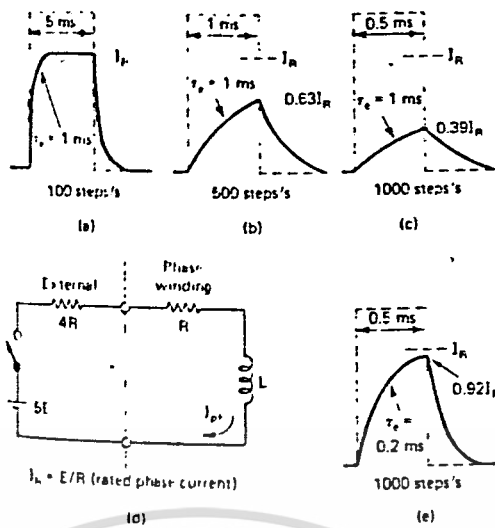
pull-out torque จะลดลงตามความเร็วที่เพิ่มขึ้น แรงบิดที่ลดลงนี้เป็นเพราะการลดลงของกระแสที่ป้อนให้กับขดลวดที่ความเร็วในการหมุนที่สูง เมื่อขดลวดถูกแทนด้วยวงจรที่มี R กับ L ที่อนุกรมกัน ผลของวงจรที่จะไปสร้าง step input ด้วย amplitude E คือ

$$I_{ph} = (E/R)(1 - e^{-t/\lambda})$$

ซึ่ง E/R เป็นกระแสเฟสค่าสุดท้ายซึ่งเป็นกระแสเฟสขณะปกติ

และ $\lambda = L/R$ ซึ่งเป็น time constant ของขดลวด

การกระตุ้นเฟสอาจใช้ 50 % ของ duty cycle ของ square wave ที่มีความถี่ ω_s pulses/s ดังนั้นความยาวของ pulse หนึ่งจะเท่ากับ $T = 2\pi/\omega_s$ ถ้าหาก $T/\lambda = 5$ I_{ph} จะประมาณเท่ากับอัตรา E/R ที่การกระตุ้น ณ จุดสุดท้าย ดังรูป a) ซึ่งเท่ากับ 100 steps/s



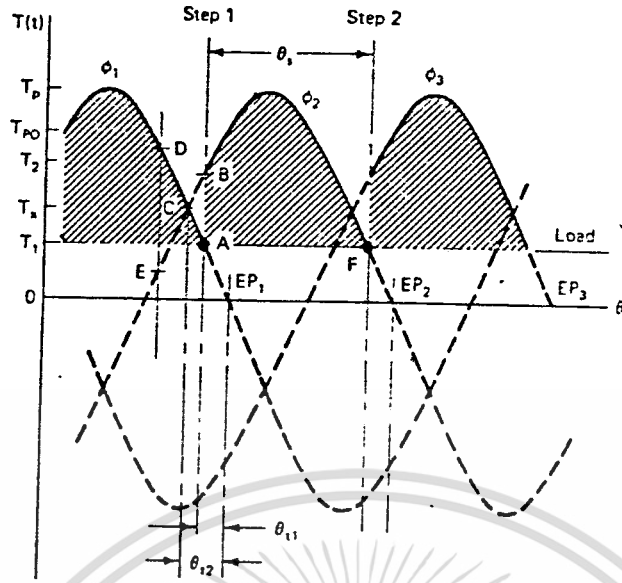
รูป 2.15 ผลของอัตรา stepping ของกระแสเฟส

จากรูป ที่อัตรา stepping สูงๆ จะไม่มีเวลาอย่างเพียงพอที่จะให้กระแสเฟสเพิ่มขึ้นจนเท่ากับอัตรากระแสปกติ รูป e) แสดงให้เห็นว่าตัวต้านทางภายนอกจะลด time constant ทำให้กระแสเฟสและแรงบิดเพิ่มขึ้น เส้นประแสดงการกระตุ้น เส้นทึบแสดงกระแสเฟส

จากการที่กระแสเฟสที่ไม่สามารถเพิ่มกระแสถึงอัตรากระแสที่แท้จริงได้ เมื่อความเร็วในการทำงานเพิ่มขึ้น และทำให้แรงบิดลดลง ปัญหานี้สามารถแก้ได้ด้วยการใส่ตัวต้านทางภายนอกอนุกรมกับขดลวดดังรูป d) เมื่อตัวต้านทางมีค่าเป็น 4 เท่าของค่าความต้านทานของขดลวดก็จะสามารถปรับปรุงกระแสเฟสซึ่งสามารถทำให้ที่ input 1000 steps/s สามารถสร้างกระแสได้ 92 % ของอัตรากระแสปกติที่จุดสุดท้ายของช่วงที่ถูกกระตุ้นซึ่งมีค่าเพิ่มขึ้นมาก อย่างไรก็ตามแรงบิดจะไม่มีเปลี่ยนแปลงมากถ้าความเร็วต่ำแต่ถ้าความเร็วสูงดังรูป และใช้วงจรที่ใส่ตัวต้านทางภายนอก ดังรูป e) ราคาที่ต้องจ่ายสำหรับแหล่งจ่ายไฟที่ใหญ่ขึ้น ในกรณีของตัวต้านทาง 4R ที่ใส่เข้าไป แหล่งจ่ายไฟต้องจ่าย output ที่มี voltage และกำลัง 5 เท่า จากขอกเดิมที่เคยจ่าย

พิจารณารูปที่ 2.16

แรงบิดที่ถูกสร้างโดย stepping motor ขณะพริบตาเดียว จะมีค่าเกินแรงบิดที่ load เพื่อให้เร่ง stepping motor ให้เข้าสู่ตำแหน่งต่อไป พื้นที่ที่แรงเงาไว้แสดงแรงบิดโดยประมาณที่ใช้ในการเร่ง load



รูป 2.16 แสดงแรงบิดของ motor ขณะที่มี load

จากรูป แรงบิดที่ load T_L ถูกขับโดย stepping motor 3 เฟส กราฟที่ได้แสดงแรงบิดที่ stepping motor สร้างขึ้นชั่วขณะซึ่งขึ้นอยู่กับตำแหน่งของ rotor

ในการขับ load จะต้องมียกกำหนดที่สำคัญอย่างหนึ่งคือ แรงบิดที่ stepping motor สร้างขึ้นชั่วขณะนี้จะต้องมากกว่าแรงบิดที่ load เพื่อให้แรงบิดทั้งหมดโดยประมาณที่ใช้เร่ง rotor เพื่อเข้าสู่ตำแหน่งต่อไป สมมติว่า ϕ_1 ถูกกระตุ้นและ rotor เคลื่อนเข้าสู่จุดสมดุล EP_1 อย่างไม่รู้ตัวมาก่อนที่จะถึง ϕ_2 ก็ถูกกระตุ้นที่จุด a ทำให้แรงบิดกระโดดสู่จุด b (แรงบิดที่ b เป็นแรงบิดที่เกิดขึ้นเพราะ ϕ_2 ในเวลาที่เกิดการ switching จาก ϕ_1 สู่ ϕ_2) ในเวลานี้ ϕ_2 กำลังทำงานและเร่ง load สู่ตำแหน่งต่อไปที่จุด f เมื่อ ϕ_3 ถูกกระตุ้น รูปกราฟก็จะเข้าไปเรื่อย ๆ

มุมของแรงบิด ϕ_1 สามารถใช้แสดงตำแหน่งของ rotor ที่การกระตุ้นเฟสเกิดขึ้นสัมพันธ์กับจุดสมดุลของเฟสก่อนหน้า มุมของแรงบิดจะระบุมุมโดยที่จะมีลำดับการกระตุ้นเฟสสัมพันธ์กับจุดสมดุลสถิตย์ของ rotor ดังรูป ϕ_1 เป็นมุมของแรงบิด T_1 ที่เกิดขึ้นจาก load การกระตุ้น ϕ_2 จะทำที่จุด c เกิด crossover torque T_x ด้วยมุมของแรงบิด ϕ_2 โดยปกติแล้วจุดกระตุ้นที่มากกว่า ϕ_2 จะถูกใช้บ่อยเพราะถ้าหาก ϕ_2 ถูกกระตุ้น สมมุติที่จุด d แรงบิดที่รวมจุด e ซึ่งมีค่าต่ำกว่าแรงบิดที่ load เป็นผลให้เกิดการหน่วงของ stepping motor

แรงบิดที่ load ต้องน้อยกว่า pull-out torque T_{po} ซึ่งค่าที่ได้ขึ้นกับจำนวน
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ใดเห็นไปเชิงพาณิชย์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟสและอยู่ระหว่าง T_x กับแรงบิดสูงสุด T_u ถ้าหากแรงเฉื่อยของ load ละทิ้งแรงบิดสูงสุดต้องไม่มากกว่า T_x ตรงกันข้าม แรงบิดที่ load T_2 ที่ load มีแรงเฉื่อยมากอาจยอมรับได้ เหตุผลคือแม้ว่าแรงบิดของ stepping motor ซึ่งขณะมีค่าต่ำกว่า T_2 จะเกิดพลังงานจลน์ในแรงเฉื่อยเพียงพอที่จะขับ load เข้าสู่จุดกระตุ้นต่อไป T_x เพิ่มขึ้นตามจำนวนเฟสและการสร้างแรงบิดและดังนั้นความสามารถที่จะขับ load ของ stepping motor ก็เพิ่มขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Drive circuit สำหรับ stepping motor

3.1 หลักการ

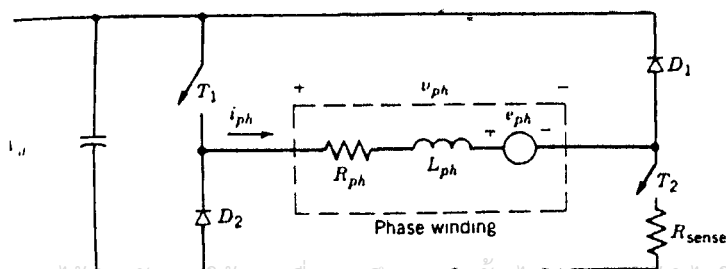
เพื่อให้ได้แรงบิดมากที่สุด วงจร drive ซึ่งทำหน้าที่ switching voltage ที่ป้อนให้ stepping motor ในขณะที่ turn-on จะมีกระแสไหลผ่านขดลวดเพิ่มจาก 0 ไปเป็นระดับที่ต้องการ และในขณะที่ turn-off จะมีกระแสลดลงเป็น 0 ซึ่งการเปลี่ยนระดับ voltage จะต้องมียุ่ช่วงเวลาสั้นที่สุด เพราะว่ายช่วงเวลาระหว่างการ turn-on และ turn-off อาจจะตรงกัน ถ้าววงจร drive มีการเปลี่ยนระดับช้า ซึ่งมีผลทำให้แรงบิดลดลง

สิ่งสำคัญอีกอย่างหนึ่งคือ ในขณะที่ turn-off จะเกิดกระแสไหลย้อนกลับในขดลวดซึ่งเกิดจากขั้วของเส้นแรงแม่เหล็ก ซึ่งวงจร drive ควรจะมีการออกแบบให้กระแสส่วนนี้ไหลกลับเข้าสู่แหล่งจ่ายไฟ

วงจร drive จะแบ่งออกเป็น bipolar และ unipolar ซึ่งจะใช้กับ stepping motor แบบต่าง ๆ ดังนี้

3.1.1 Unipolar voltage drive

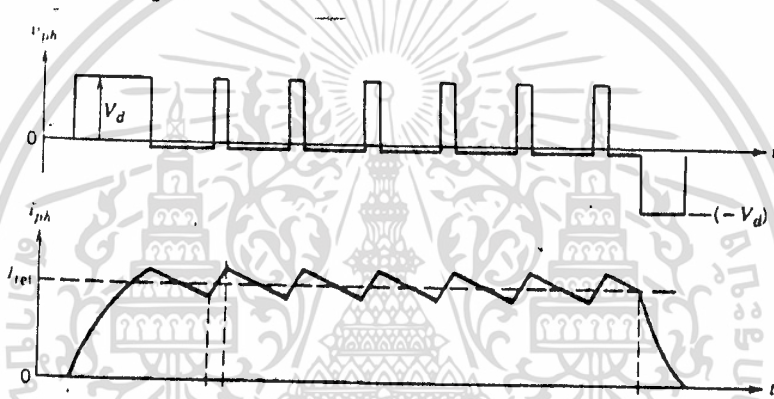
วงจรมีจะใช้กับ stepping motor ที่มีการป้อนกระแสเพียงทิศทางเดียว เช่น Variable-reluctance stepping motor และ hybrid ที่มีการนำขดลวดแบบ bipolar-wound phase



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูป 3.1 วงจร unipolar voltage drive
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป V_d จะเข้าสู่ขดลวดโดยผ่าน switch 2 ตัว โดยขดลวดจะแทนด้วย equivalent circuit และ V_d ที่ใช้จะต้องดึงกระแสไหลย้อนกลับในขณะ turn-off ได้

เพื่อให้เกิดการกระตุ้นขดลวด T_1 และ T_2 จะ turn-on มีผลให้ขดลวดมีศักดาตกคร่อม V_d ซึ่งจะเหนี่ยวนำให้เกิด i_{ph} ขึ้น T_2 จะปิดตลอดเพื่อต้องการให้ขดลวดพร้อมจะถูกกระตุ้น T_1 จะ switch ตาม PWM มีผลให้ขดลวดถูกกระตุ้นเป็นจังหวะ โดยกระแสไหลผ่านขดลวดจะถูกควบคุมจาก I_{ref} ซึ่งจะมีผลต่อจังหวะของ PWM ในขณะที่ turn-on, $V_{ph} = V_d$ และ i_{ph} เพิ่มขึ้น เมื่อ turn-off i_{ph} จะไหลผ่าน T_2 , R_{ph} และ D_2 แล้วจะลดลงดังรูป



รูป 3.2 กราฟความสัมพันธ์ กระแสเฟสกับการ switch

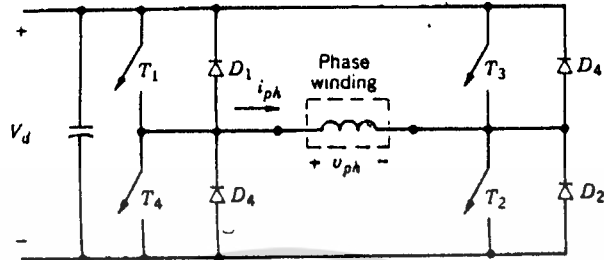
ทำให้ $i_{ref} = I_{ref}$ เป็นกระแสไหลผ่านขดลวด

เมื่อไม่กระตุ้นขดลวด T_2 จะ turn-off กระแสในขดลวดจะไหลผ่าน ไดโอด D_1 และ D_2 และ $V_{ph} = -V_d$ เป็นผลทำให้กระแสลดลงเป็น 0 ทันที และ inductive energy จะไหลกลับสู่แหล่งจ่ายไฟตรง

3.1.2 Bipolar voltage drive

วงจร drive แบบ bipolar จะใช้กับ stepping motor แบบ permanent-magnet และ hybrid stepping motor แต่ละ phase ของ stepping motor เหล่านี้ ต้องป้อน voltage จากแหล่งจ่ายไฟตรงให้กับขดลวดต้องกลับขั้วไปมาได้ ซึ่งใช้วงจร full-bridge dc-dc converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.3 วงจร bipolar voltage drive

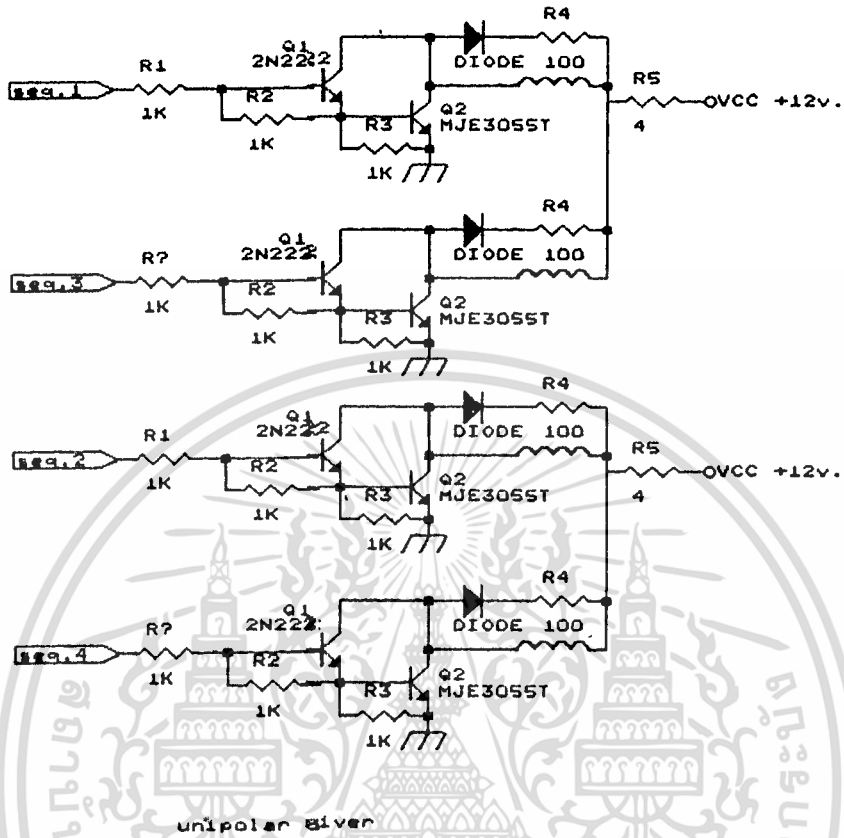
จากรูป T_1 และ T_2 จะทำงานพร้อมกัน ส่วน T_3 และ T_4 ก็จะทำงานพร้อมกันด้วย เมื่อต้องการจะกระตุ้นขดลวดเป็นบวก T_1 และ T_2 จะ turn-on ทำให้ i_{ph} เพิ่มขึ้น เมื่อต้องการเปลี่ยนแปลงทิศทางกระแส ก็ให้ T_1 และ T_2 turn-off ในช่วงนี้จะเกิด blanking เป็นช่วงเวลาสั้นก่อน เพื่อให้ i_{ph} ไหลผ่าน D_4 และ D_3 เข้าสู่แหล่งจ่ายไฟ ทำให้ i_{ph} ลดลง แล้ว T_3 และ T_4 จึงจะ turn-on

3.2 การออกแบบวงจร drive

เนื่องจากใน project ชั้นนี้มีการใช้ stepping motor หลายชนิด คือ hybrid-stepping motor ซึ่งมีการพันแบบ bifilar, Variable reluctance และ 2 phase hybrid stepping motor ซึ่งมีการใช้วงจร drive ทั้ง 2 แบบ

3.2.1 วงจร drive circuit สำหรับ hybrid stepping motor และ VRSM

เนื่องจาก hybrid stepping motor แบบ bifilar ใช้ driver แบบ unipolar วงจรที่ใช้จึงประกอบด้วยแหล่งจ่ายไฟให้ขดลวดและส่วน switching เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.4 วงจร driver สำหรับ HSM และ VRSM

การทำงาน สัญญาณจาก sequencer จะผ่านเข้าขาเบสของ Q_1 ทำให้ Q_1 นำกระแสไหลเข้าสู่ขาเบสของ Q_2 ทำให้ Q_2 นำกระแสด้วย ทำให้ Q_1 และ Q_2 ทำหน้าที่เป็น switch ความคุมการเปิดปิดกระแสผ่านขดลวด กระแสเข้าขาเบสของ Q_1 จะถูกควบคุมจาก R_1 ซึ่งจะหาได้จาก

$$I_{input} = (I_{out} / \beta_1 \beta_2) + (I_o / (R_2 + R_3)) \quad (1)$$

$$= (5 - (2 \times 0.6)) / 6 \quad (2)$$

จาก (1) และ (2) $R_1 = 3.8 \beta_1 \beta_2 / I_o + 3.16 (R_2 + R_3)$

R_2 และ R_3 ทำหน้าที่ถ่ายประจุที่ charge ให้ส่วนเบสของ Q_1 และ Q_2 ออกเพื่อให้ในขณะที่ turn off จะ off ได้เร็วขึ้น D_1 และ R_4 จะทำหน้าที่นำกระแสย้อนกลับขณะที่ turn off ขดลวด กลับเข้าสู่แหล่งจ่ายไฟ และ R_5 เป็นตัวจำกัดกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ในขณะ turn on ไม่ให้ขดลวดมีการเหนี่ยวนำกระแสมากเกินไป ป้องกันขดลวดเสียหาย ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

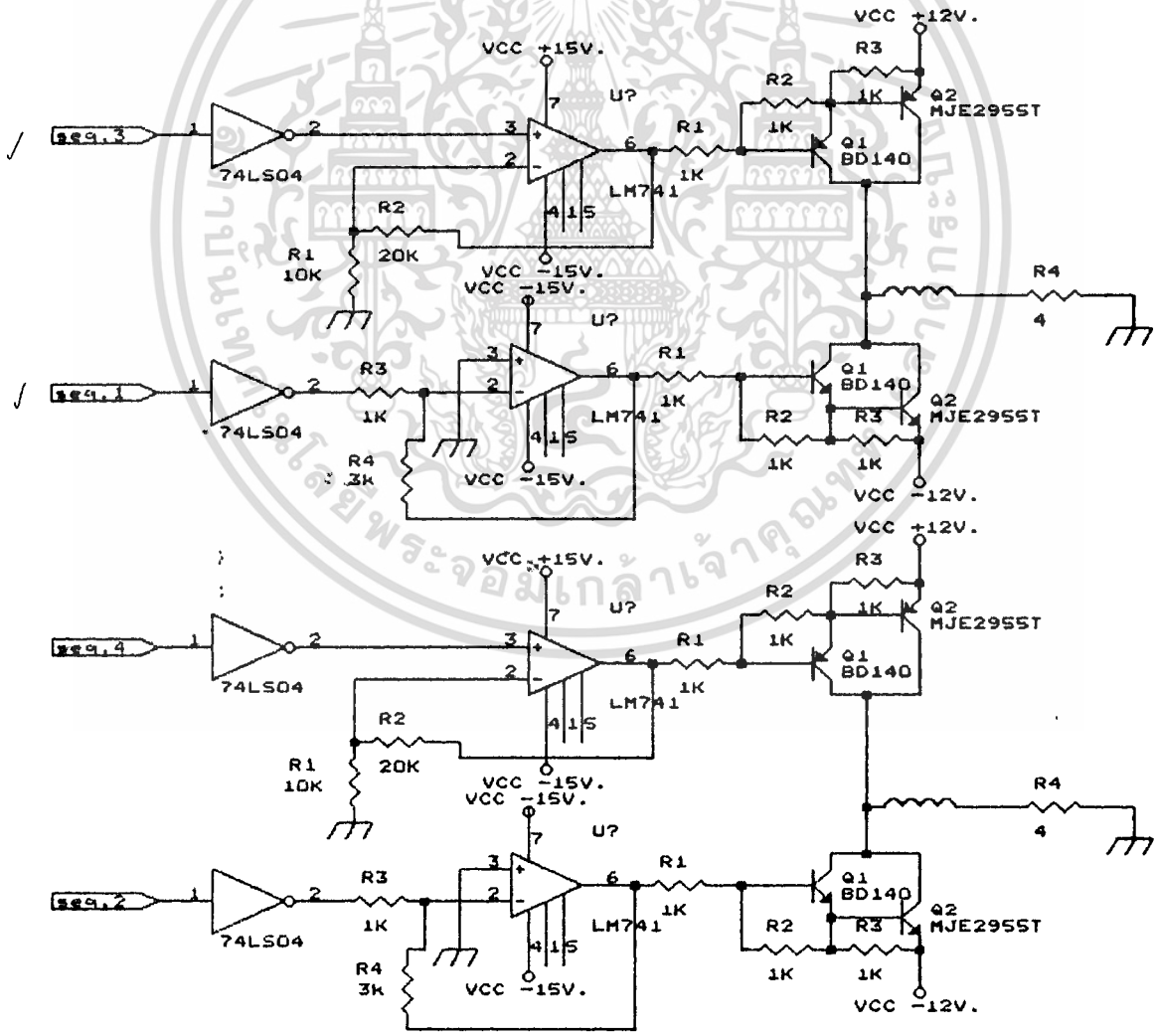
R_2 และ R_3 เลือกใช้ตามความเหมาะสม แต่ไม่ควรมีค่าน้อยเกินไป R_4 เป็นตัวจำกัดกระแสผ่าน D_4

$$R_4 = V_L / I_{Dmax}$$

R_6 ใช้ $4R_L$ เมื่อ R_L เป็นความต้านทานของขดลวด

3.2.2 Drive circuit ของ hybrid stepping motor แบบ 2 เฟส

จากที่ stepping motor แบบ 2 เฟส hybrid ใช้วงจร drive แบบ bipolar จึงมีการใช้แหล่งจ่ายไฟ 2 ชุด เป็นบวกและลบ



bipolar driver for 2 phase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูป 3.5 วงจร driver สำหรับ 2 phase HSM
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

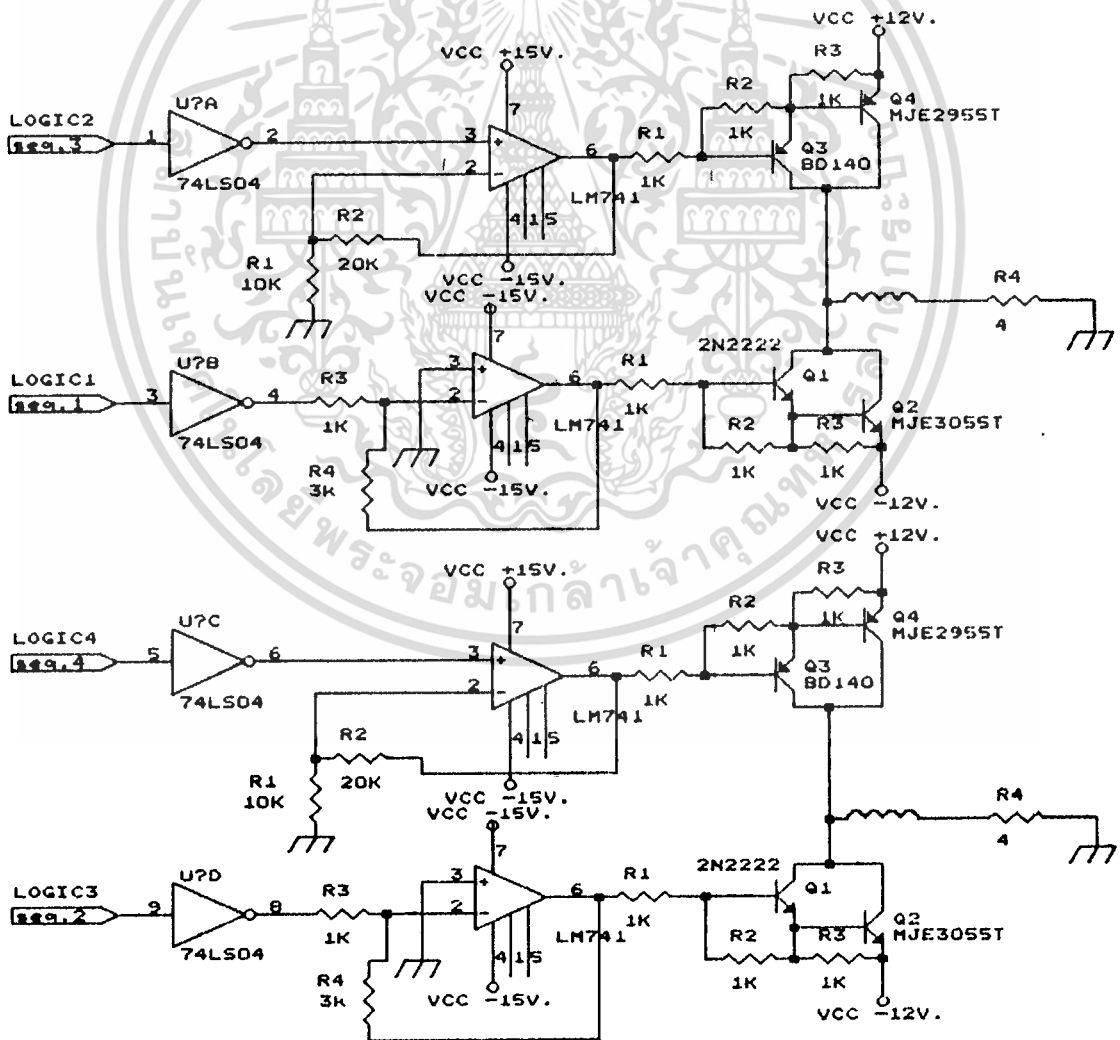
R_2 และ R_3 เลือกใช้ตามความเหมาะสม แต่ไม่ควรมีค่าน้อยเกินไป R_4 เป็นตัวจำกัดกระแสผ่าน D_4

$$R_4 = V_L / I_{Dmax}$$

R_5 ใช้ $4R_L$ เมื่อ R_L เป็นความต้านทานของขดลวด

3.2.2 Drive circuit ของ hybrid stepping motor แบบ 2 เฟส

จากที่ stepping motor แบบ 2 เฟส hybrid ใช้วงจร drive แบบ bipolar จึงมีการใช้แหล่งจ่ายไฟ 2 ชุด เป็นบวกและลบ



bipolar driver for 2 phase

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานที่ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูป 3.5 วงจร driver สำหรับ 2 phase HSM
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ logic 1 เป็น high เมื่อผ่าน inverter gate ทำให้เป็น low แล้วจึงผ่าน inverting amp. ซึ่งทำให้ output เป็น 0 ทำให้ Q_1 และ Q_2 on นำกระแสออกจากขดลวดของ ϕ_1 ซึ่งถูกกระตุ้นให้ไหลกลับสู่แหล่งจ่ายไฟ

เมื่อ logic 1 เป็น low เมื่อผ่าน inverter gate จะทำให้เป็น high เมื่อผ่าน inverting amp. จะถูกขยายให้ output เป็น -12 โวลต์ ทำให้ ϕ_1 และ ϕ_2 off

ในลักษณะคล้ายกัน เมื่อ logic 2 เป็น high เมื่อผ่าน inverter gate จะเป็น low จากนั้นจะผ่าน non-inverting amp. ซึ่ง output จะเป็น low เช่นเดิม ทำให้ Q_3 และ Q_4 on นำกระแสจากแหล่งจ่ายไฟ +12 โวลต์ไหลเข้าสู่ขดลวด และเมื่อ logic 2 เป็น low เมื่อผ่าน inverter เป็น high แล้วจะถูกขยายโดยวงจร non-inverting amp. ให้เป็น +12 โวลต์ ซึ่งมีผลให้ Q_3 และ Q_4 off

การทำงานของ logic 3 และ logic 4 ก็ทำงานเหมือนกับ logic 1 และ logic 2 จะเห็นได้ว่าเราควบคุมทิศทางกระแสในขดลวดทำได้โดยการ switch logic ตามต้องการ

ส่วนการคำนวณหาค่า R_1, R_2, R_3, R_4 ทำเช่นเดียวกับ hybrid stepping motor

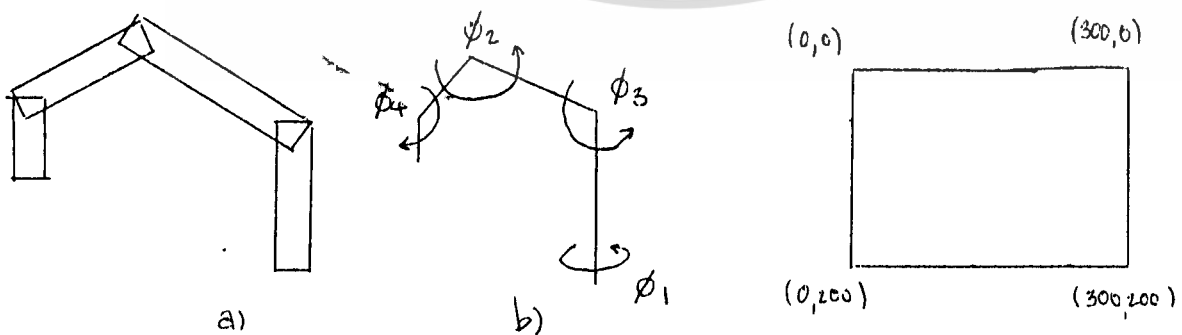
บทที่ 4

การออกแบบโปรแกรม

จุดมุ่งหมายของโปรแกรมนี้คือ

1. เป็นวงจร sequential ให้กับมอเตอร์แต่ละตัว
2. เป็นวงจร counter
3. เป็นตัวคำนวณแปรค่าจากค่า x,y ในแนวระนาบให้เป็นค่ามุมสำหรับมอเตอร์ตัวต่าง ๆ
4. เป็นโปรแกรมที่สามารถรักษาค่าของข้อมูล x,y ของหลายวงจรพิมพ์แบบต่าง ๆ ไว้ได้
5. ควบคุมมอเตอร์ทั้งหมดโดยใช้ข้อมูลที่ส่งออกทาง printer port ของเครื่อง microcomputer
6. มี menu เลือกการทำงาน

4.1 การคำนวณหาค่ามุมของมอเตอร์แต่ละตัว



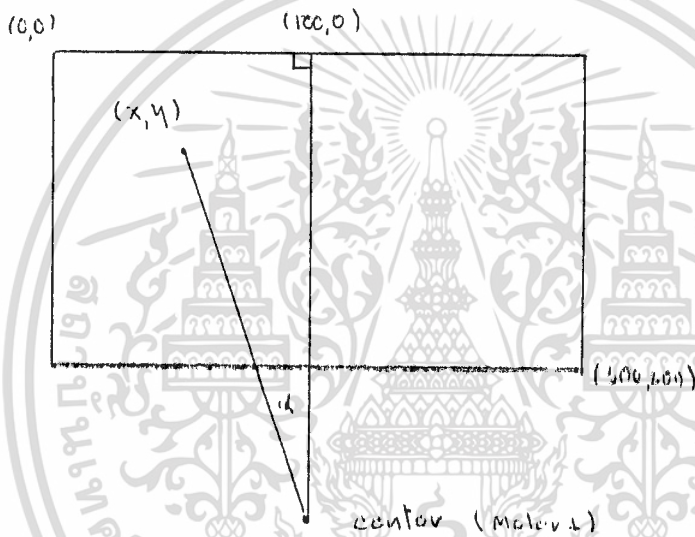
รูป 4.1 a) ลักษณะของแขนกล b) มุมของมอเตอร์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.1 ค) เป็นการแสดงพื้นที่การเจาะของแขนกลซึ่งมีหน่วยเป็น มม. โดยแนวแกน x มีระยะทางการเจาะเท่ากับ 300 มม. และแนวแกน y มีระยะทางการเจาะเท่ากับ 200 มม.

4.2 วิธีคำนวณหามุม

จาก จุด x, y สามารถหามุมต่างๆ ได้ดังนี้



รูป 4.2 พื้นที่การทำงาน

4.2.1 มุมของ motor 1

หาได้จากสูตร

$$\alpha = \arctan(150 - x)/y$$

จากรูป 4.2 เราจะแยกการหา $\text{angle}1$ ได้ 2 กรณี คือ

1. $x < 150$

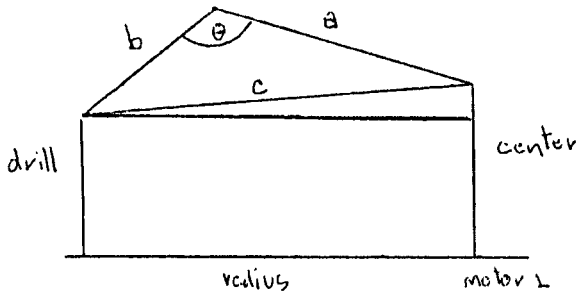
$$\text{angle}1 = 90^\circ - \alpha = 90^\circ - \arctan(150-x)/(200-y+y_0)$$

2. $x > 150$

$$\text{angle}1 = 90^\circ + \alpha = 90^\circ + \arctan(150-x)/(200-y+y_0)$$

4.2.2 มุมของ motor 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.3 แสดงการหามุมของ motor 2

จากรูป 4.2

ถ้า $drill = center$

$$radius = ((150-x)^2 + (200-y-y_0)^2)^{1/2}$$

จะได้ว่า ระยะ $c = \sqrt{(radius)^2 - (center-drill)^2}$

$$จาก \quad c^2 = a^2 + b^2 + 2ab \cos \theta$$

เป็นสูตรหาตัวที่เหลือของสามเหลี่ยมด้านใด ๆ

$$\theta = \arccos (a^2 + b^2 - c^2) / 2ab$$

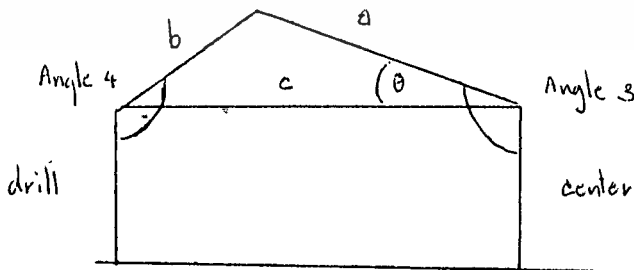
$$ดังนั้น \quad angle2 = \arccos (a^2 + b^2 - c^2) / 2ab$$

4.3.3 มุมของ motor 3 และ motor 4

สามารถแบ่งเป็นกรณีต่างๆ ดังนี้

1. กรณี $drill = center$

จาก รูป 4.4



รูป 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

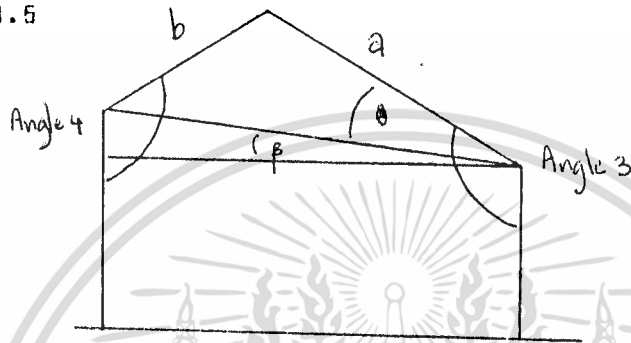
$$\theta = \arccos (a^2 + c^2 - b^2) / 2ac$$

$$\text{angle3} = \theta + 90^\circ$$

$$\text{angle4} = 90^\circ + (180 - \text{ang2} - \theta)$$

2.กรณี drill > center

จาก รูป 4.5



รูป 4.5

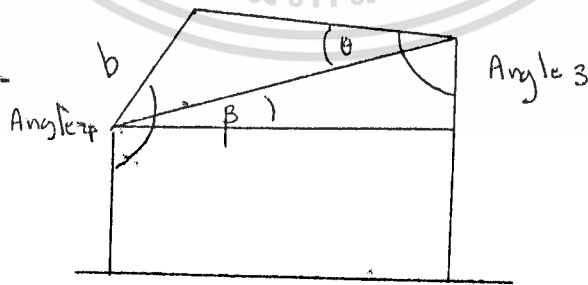
จะได้ว่า $\beta = \arctan(\text{axis} - \text{center}) / \text{radius}$

$$\text{angle3} = 90^\circ + \beta + \theta$$

$$\text{angle4} = 90^\circ - \beta + (180^\circ - \text{ang2} - \theta)$$

3.กรณี drill < center

จาก รูป 4.6



รูป 4.6

จะได้ว่า

$$\text{angle3} = 90^\circ - \beta + \theta$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

$$\text{angle4} = 90^\circ + \beta + (180^\circ - \text{ang2} - \theta)$$

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การคำนวณหาจำนวน clock ของ motor แต่ละตัว

จะใช้สูตร $clock = (\text{จำนวน clock ต่อรอบ} \times \text{มุมที่เปลี่ยนไป} \times \text{อัตราทด}) / 360^\circ$
 ส่วนที่ศทางการเคลื่อนที่ของมอเตอร์หาจากการนำมุมเดิม มาเปรียบเทียบกับมุมใหม่ แล้ว
 กำหนดทิศทางโดยกำหนดเป็น mm

(motor direction) โดย

mm = 1 คือ การเคลื่อนที่ตามเข็มนาฬิกา

mm = 0 คือ การเคลื่อนที่ทวนเข็มนาฬิกา

4.4 การออกแบบสัญญาณ sequential

เนื่องจาก เราออกแบบให้ใช้ต่อกับเครื่อง PC โดยผ่าน printer port ซึ่งเป็น
 parallel output ซึ่งมีด้วยกัน 8 บิต
 จาก data ของ printer port



เราจะแบ่งข้อมูลเป็น 2 ชุด โดย

D₀ - D₃ เป็น sequential ควบคุมมอเตอร์

ซึ่งการควบคุม hybrid motor และ 4 phase motor ต้องใช้ D₀ - D₃ ที่มี state ดังนี้

ψ_1	ψ_2	ψ_3	ψ_4
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1

ψ_1	ψ_2	ψ_3	ψ_4
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1

การควบคุม motor 2 phase ต้องใช้ state ดังนี้

ϕ_1	ϕ_2	ϕ_3	ϕ_4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

ϕ_1	ϕ_2	ϕ_3	ϕ_4
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0

cw

ccw

ส่วน $D_4 - D_7$ จะเป็นตัว selector ที่จะเลือกตัวใด

ถ้าเลือก มอเตอร์ตัวที่ 1 $D_4 - D_7$ จะเท่ากับ 1 0 0 0

เลือก มอเตอร์ตัวที่ 2 $D_4 - D_7$ จะเท่ากับ 0 1 0 0

เลือก มอเตอร์ตัวที่ 3 $D_4 - D_7$ จะเท่ากับ 0 0 1 0

เลือก มอเตอร์ตัวที่ 4 $D_4 - D_7$ จะเท่ากับ 0 0 0 1

นำข้อมูล $D_0 - D_7$ มาต่อกัน จะได้ output ไปควบคุมการเลือกมอเตอร์และควบคุมให้มอเตอร์หมุนเพื่อให้ได้ตำแหน่งที่ต้องการ เช่น ต้องการให้ motor1 เคลื่อนตามเข็มนาฬิกา :

ข้อมูล $D_0 - D_7$ ก็จะเป็น

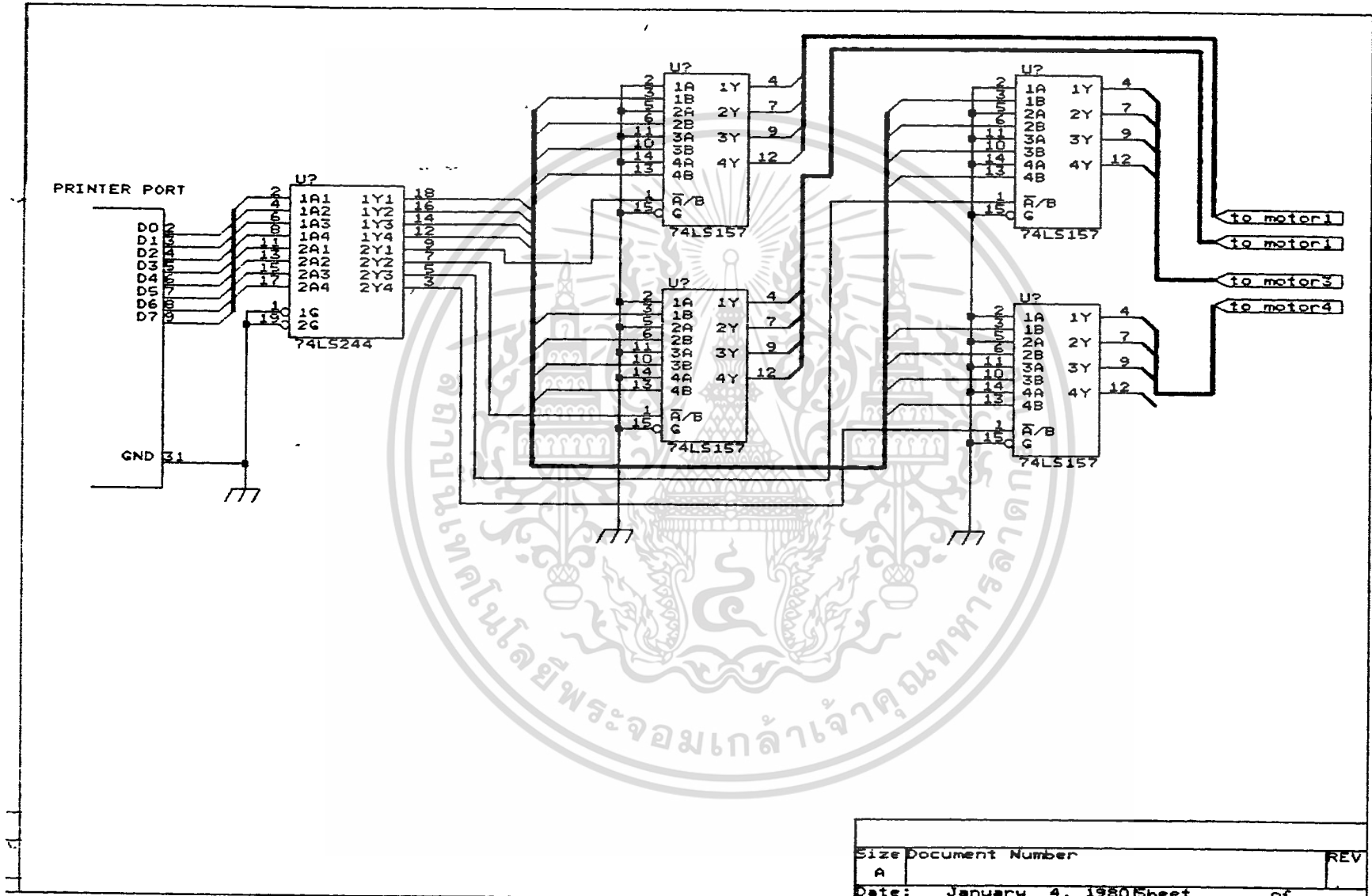
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	1	0	0
1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	1
1	0	0	0	1	0	0	1

ต้องการให้ มอเตอร์ 2 เคลื่อนที่ทวนเข็มนาฬิกา

ข้อมูล $D_0 - D_7$ ก็จะเป็น

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	0	1	1	0	0
0	1	0	0	1	0	0	1
0	1	0	0	0	0	1	1
0	1	0	0	0	1	1	0

ดังนั้น การบังคับ motor จึงใช้ data ดังกล่าวป้อนออกทาง printer port และ
 จับไปที่แผงมีวงจร collector เพื่อเลือกสัญญาณ sequent bit ไปยังตัว motor ที่
 ต้องการ ดังรูป 4.7



Size	Document Number	REV
A		
Date:	January 4, 1980	Sheet of

บทที่ 5

การออกแบบแขนกล

การออกแบบโครงสร้างทางเครื่องกล (Mechanical structure) แบ่งออกเป็น 2 ส่วนซึ่งจะประกอบด้วย

1. แหล่งกำลัง (Power source) เป็นหน่วยจ่ายกำลังให้กับโครงสร้างทางเครื่องกล เป็นตัวผลักดันให้โครงสร้างทางเครื่องกลเคลื่อนที่ แหล่งกำลังอาจจะเป็นมอเตอร์ไฟฟ้า, ไฮดรอลิค, นิวเมตริก และอื่นๆ
2. ระบบส่งกำลัง (Power transmission system) เป็นตัวส่งกำลังจากแหล่งกำลังไปยังจุดที่โครงสร้างทางเครื่องกลเคลื่อนที่

5.1 ลักษณะของแขนกล

การเลือกลักษณะของแขนกล จะขึ้นอยู่กับลักษณะการนำไปใช้งานและพื้นที่ใช้งาน

Principle	Axes		Wrist		
	Kinematic chain	Workspace	1	2	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ ครู ใช้ใน ลักษณะแขนกลแบบต่างๆ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากในโครงการนี้ต้องการการเคลื่อนที่ที่เป็นทรงกลม จึงเลือกลักษณะของแขนกลแบบสุดท้าย ซึ่งมีพื้นที่การทำงานดังรูป 5.1

5.2 ระบบส่งกำลัง

การเลือกระบบส่งกำลังจะขึ้นอยู่กับลักษณะการเคลื่อนที่ของโครงสร้างทางเครื่องกล ลักษณะของแหล่งกำลัง และการติดตั้งแหล่งกำลัง

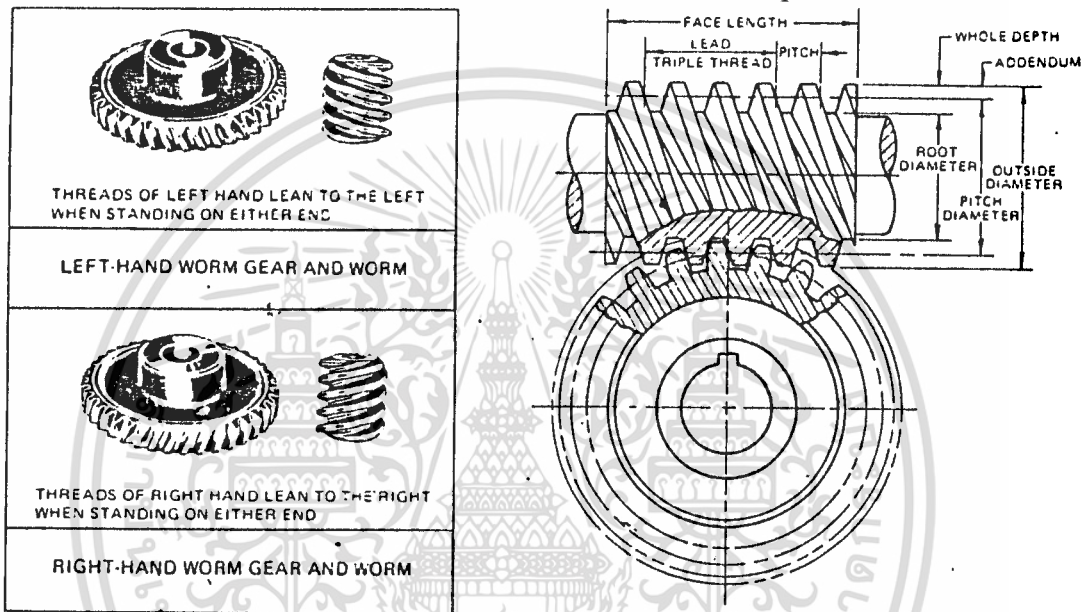
System	Principle	Characteristics	Transmission of Movement	Speed Reduction	Transmission at Distance	Application
Spur gearing		First rotatory arm axis High moments	$\frac{R}{R}$	Yes	No	Arm
Bevel gearing		Special case for flange installation	$\frac{R}{R}$	Yes	No	Arm
Worm gear		High transmission ratio Heavy in weight Heat problems	$\frac{R}{R}$	Yes, high	No	Arm Wrist Gripper
Planetary gear		Expensive Heavy in weight	$\frac{R}{R}$	Yes, high	No	Arm
Harmonic drive		Very high transmission ratio Small dimensions Light in weight	$\frac{R}{R}$	Yes, very high	No	Arm Wrist
Chain drive		No backlash Heavy in weight No vibration	$\frac{R}{R} \frac{T}{R}$ $\frac{R}{R} \frac{T}{T}$	Possible	Yes	Slide module Wrist
Toothed belt drive		Backlash and vibration problematic Very light in weight	$\frac{R}{R} \frac{T}{R}$ $\frac{R}{R} \frac{T}{T}$	Possible	Yes	Wrist Gripper
Four bar linkage		Very good for precise power transmission on distance	$\frac{R}{R}$	Possible	Yes	Arm
Slider crank chain		Special case application	$\frac{R}{T} \frac{T}{R}$	No	Yes	Arm Wrist
Screw drive spindle		High transmission Friction (lubrication) problem	$\frac{R}{T}$	Yes, high	Yes	Wrist
Recirculating ball nut and screw		High transmission Very high precision and reliability Expensive	$\frac{R}{T}$	Yes, very high	Yes	Arm Wrist
Toothed rack		Precise Inexpensive	$\frac{R}{T} \frac{T}{R}$	Yes	Yes	Arm Wrist Gripper
Cylinder		A lot of variations possible for hydraulic and pneumatic systems	$\frac{T}{T}$	Yes	Yes	Arm Wrist Gripper

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทวิศวกรรมเครื่องกลฯ ไม่อนุญาตให้ทำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกไปใช้หา และนำค่าข้างต้นไปอ้างอิงของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 5.2 ลักษณะการส่งกำลังแบบต่างๆ

จากรูป 5.2 ถ้าเราต้องการการส่งกำลังจากมอเตอร์ไปยังโครงสร้างทางเครื่องกลที่มีลักษณะการเคลื่อนที่แบบเลื่อนเข้าออก เราจำเป็นต้องเลือกแบบ ball nut and screw แต่ในโครงการนี้ ซึ่งใช้แหล่งกำลังแบบมอเตอร์ มีการเคลื่อนที่เอาโครงสร้างทางเครื่องกลเป็นแบบวงกลม แต่การติดตั้งมอเตอร์จะเป็นมุมฉากกับการเคลื่อนที่ จึงเลือกใช้ระบบสร้างกำลังแบบ worm gear เป็นส่วนสำคัญ

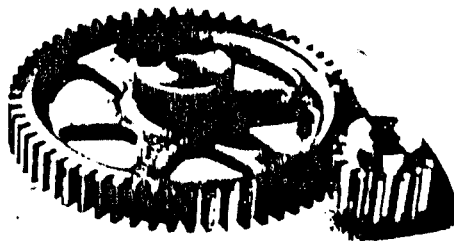


รูป 5.3 Worm gear

การตัดอัตราส่งกำลัง

$$\text{worm gear} = \text{จำนวนร่องของเฟือง}$$

ระบบส่งกำลังอีกลักษณะหนึ่งที่น่าสนใจ คือ เฟือง (gear) ซึ่งเป็นการส่งกำลังแบบวงกลม



รูป 5.4 gear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราส่งกำลัง

$$\text{gearing} = R_1/R_2$$

โดยที่ R = รัศมีของเฟืองแต่ละตัว

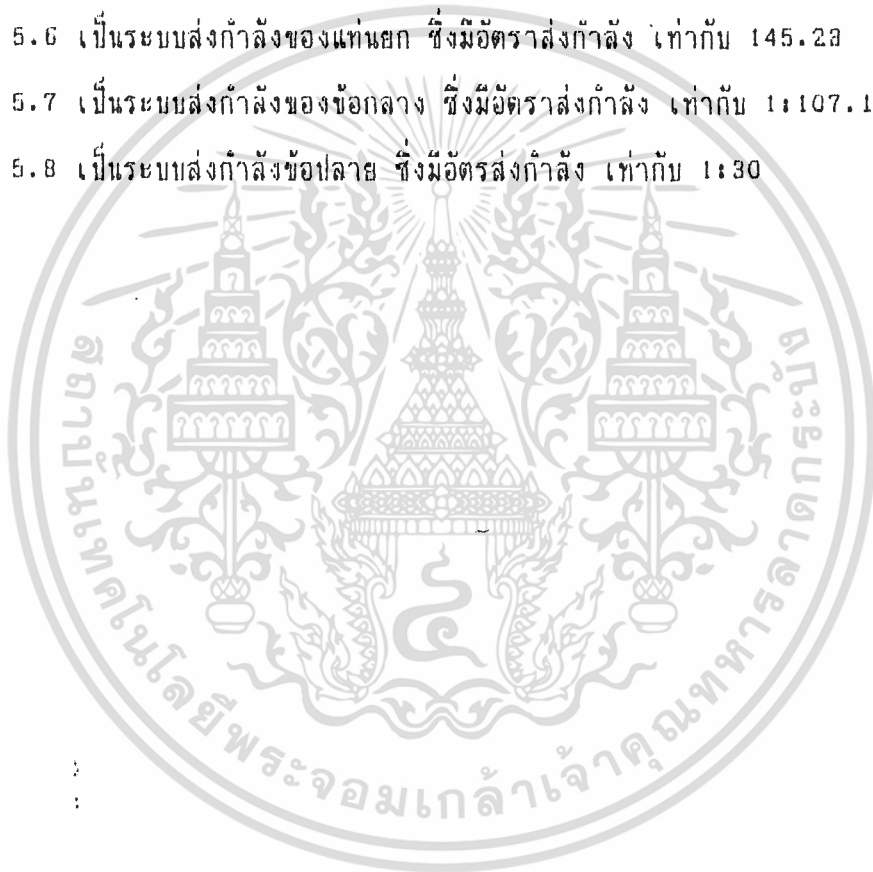
จากหลักการดังกล่าว ทำให้สามารถออกแบบลักษณะของแกนกลและระบบส่งกำลังของมอเตอร์แต่ละตัวได้ ดังรูปต่อไปนี้

รูป 5.5 เป็นระบบส่งกำลังที่ฐาน ซึ่งมีอัตราส่งกำลัง เท่ากับ 145.23

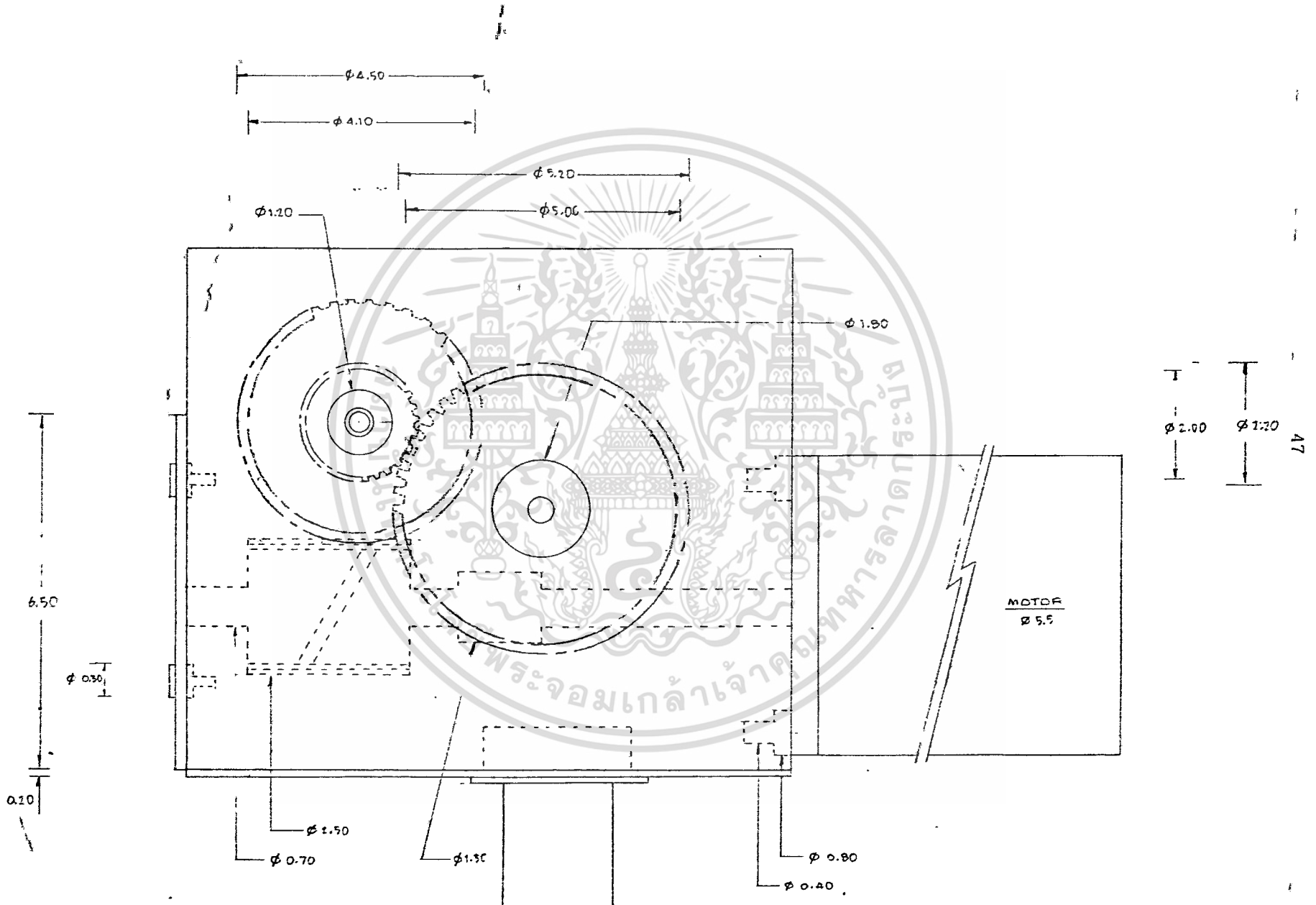
รูป 5.6 เป็นระบบส่งกำลังของแท่นยก ซึ่งมีอัตราส่งกำลัง เท่ากับ 145.23

รูป 5.7 เป็นระบบส่งกำลังของขั้วกลาง ซึ่งมีอัตราส่งกำลัง เท่ากับ 1:107.14285

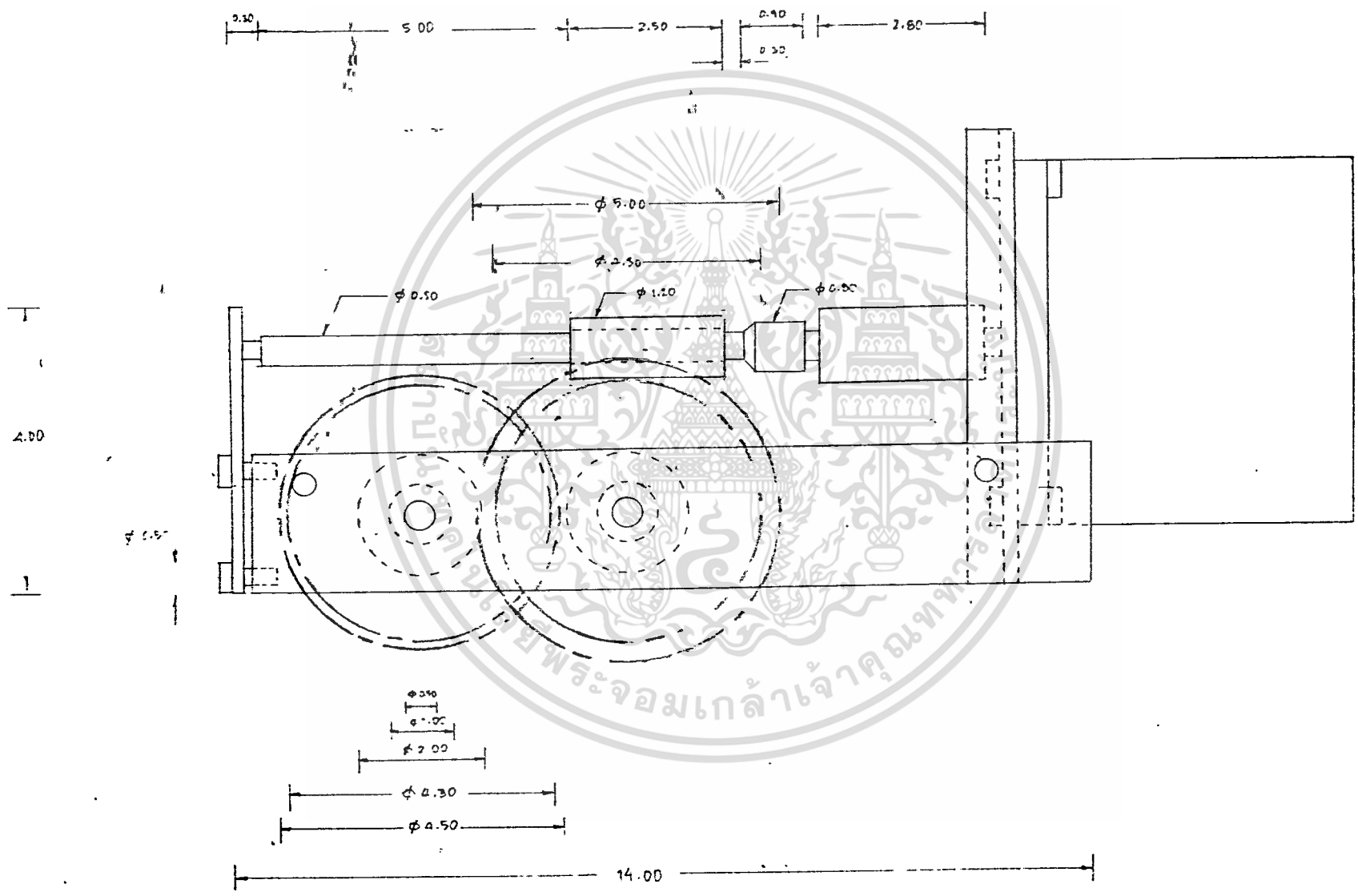
รูป 5.8 เป็นระบบส่งกำลังข้อปลาย ซึ่งมีอัตราส่งกำลัง เท่ากับ 1:30



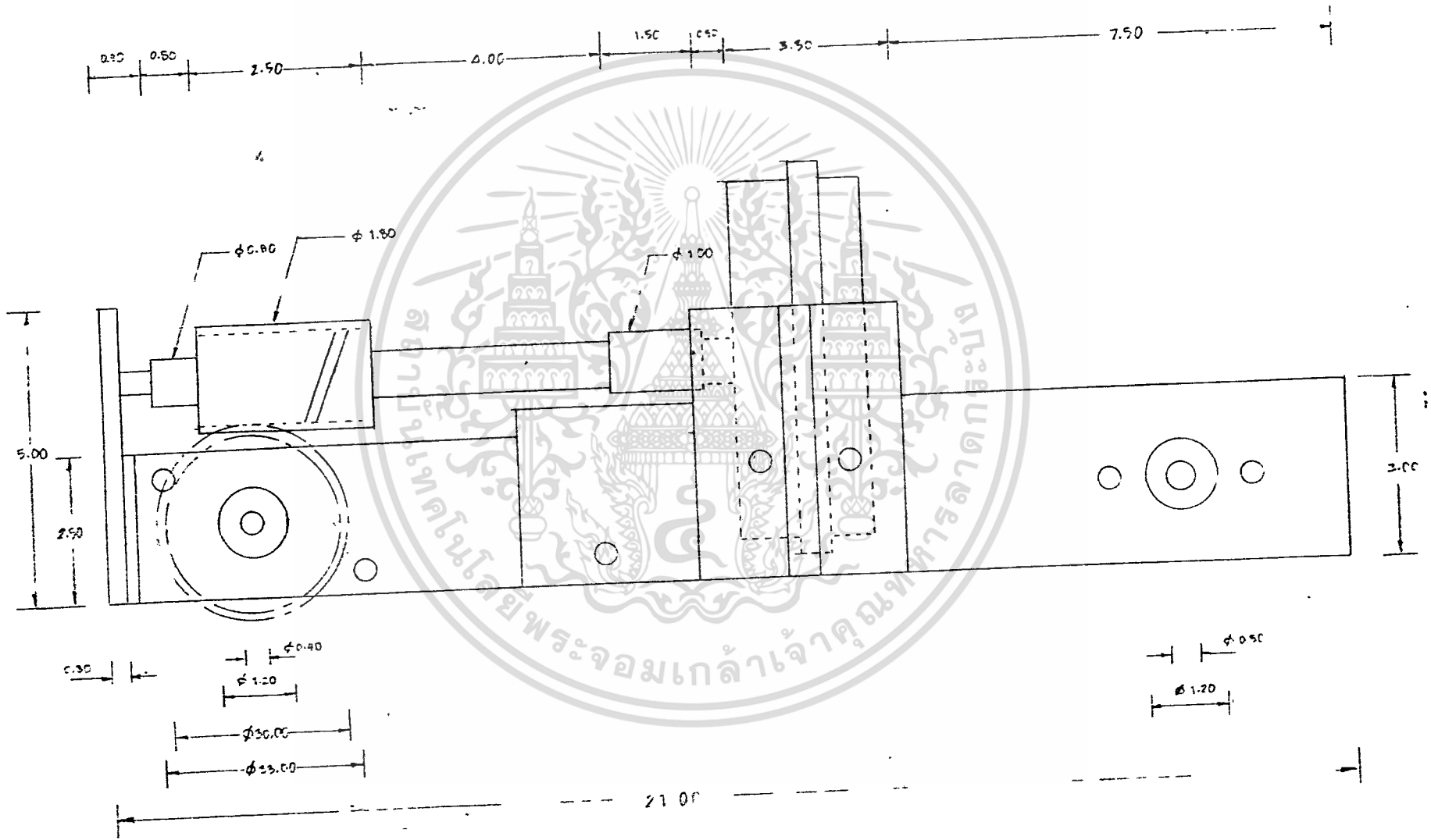
รูป 5.6



รูป 5.7



รูป 5.8



บทที่ 6

ผลการทดลอง

6.1 ผลการทดลอง

จากการป้อนข้อมูลจำนวน 9 ค่า โดยผ่านฟังก์ชัน single step ผลที่ได้ปรากฏดังนี้

ตำแหน่งที่ต้องการ		ตำแหน่งจริงที่เจาะ		
แกน x	แกน y	แกน x	แกน y	
0	0	-10	-3	
0	100	-10	97	*
0	200	-10	197	*
150	0	150	0	
150	100	150	102	*
150	200	150	193	*
300	0	307	5	
300	100	308	112	
300	200	308	215	

หมายเหตุ * เป็นตำแหน่งที่ตอกสว่านเจาะไม่ถึงแผ่น pcb

6.2 สรุปผลการทดลอง

จากการทดลองพบว่า ความแม่นยำในการเจาะยังไม่สมบูรณ์มาก ทั้งนี้เนื่องจากปัญหาเอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่าทาง *mechanical* ของตัวแกนกลางปรุขักร เช่น การที่ระยะห่างของฟันเฟือง 2 อันที่ต้องสลับกัน

มีขนาดไม่พอดีกันจำนวนฟันของเฟืองที่หาได้ยังไม่ละเอียดเท่าที่ต้องการ เพราะการที่จะหาซื้อเฟืองที่มีคุณลักษณะที่ต้องการจากการคำนวณนั้นหาได้ยากมาก

นอกจากนี้ ความโค้งตัวเล็กน้อยของอลูมิเนียมที่ใช้ทำแขนเนื่องมาจากการรับ load ก็มีผลทำให้ตำแหน่งที่ได้คลาดเคลื่อนเช่นกัน อีกทั้ง การตั้งองศาเริ่มแรกของแต่ละส่วนของแขน(ที่ต้องทาบมต่อกัน) หากได้ไม่เท่ากับที่ตกลงไว้ในโปรแกรม การเจาะก็จะไม่ได้ตำแหน่งที่ต้องการจริง ๆ

จากการลุ่มป้อนตำแหน่งต่าง ๆ โดยให้ตำแหน่ง x มีค่าสูงสุดที่ 300 ตำแหน่ง y สูงสุดที่ 200 (ขนาดแผ่น pcb ที่ใช้เจาะจะมีขนาดสูงสุดเท่ากับ 20×30 cm.) ผลของตำแหน่งที่ได้รวมแล้วพบว่ามีความคลาดเคลื่อนที่ยอมรับได้ แต่ถ้าต้องการความแม่นยำมากกว่านี้คงจะต้องมีการปรับปรุงหลายอย่าง เช่น การเพิ่มข้อต่อ (joint) อีกหนึ่งข้อที่ตรงกลางระหว่างข้อต่อสุดท้ายกับหัวเจาะ เพื่อจะได้เป็นตัวรับให้หัวเจาะสามารถเบนทิศทางให้เข้าสู่ตำแหน่งที่ต้องการจริงได้มากขึ้น การเพิ่มวงจรถ่วงเซอร์ให้กับมอเตอร์แต่ละตัวเพื่อป้องกันการที่สัญญาณนาฬิกาที่ป้อนเข้ามอเตอร์ไม่สัมพันธ์กับการหมุนของมอเตอร์ เนื่องจากกรที่มอเตอร์ถูก take load มาก

การเปลี่ยนเฟืองให้มีจำนวนฟันที่ละเอียดพอ และเปลี่ยนเฟืองที่ต้องสลับกันให้มีระยะห่างของฟันที่ใกล้เคียงกันมากขึ้น การเปลี่ยนอลูมิเนียมเป็นโลหะอื่นที่มีน้ำหนักไม่มากแต่มีความคงทนกว่า

และสุดท้ายเมื่อเสร็จจากการเจาะจุดสุดท้ายแล้ว ควรจะมีวิธีที่ทำให้แต่ละมุมของข้อต่อมีค่าเท่ากับมุมเริ่มแรกของแต่ละข้อต่อที่กำหนดในโปรแกรมได้อย่างพอดี โดยไม่ต้องพึ่งการปรับมุมด้วยสายตาอย่างเดียว

การใช้โปรแกรม

เมื่อเรียกโปรแกรม "ROBOT" แล้ว จะพบว่า มี menu ให้เลือก 6 อย่าง คือ

- open file
- new file
- run
- single step
- manual
- quit

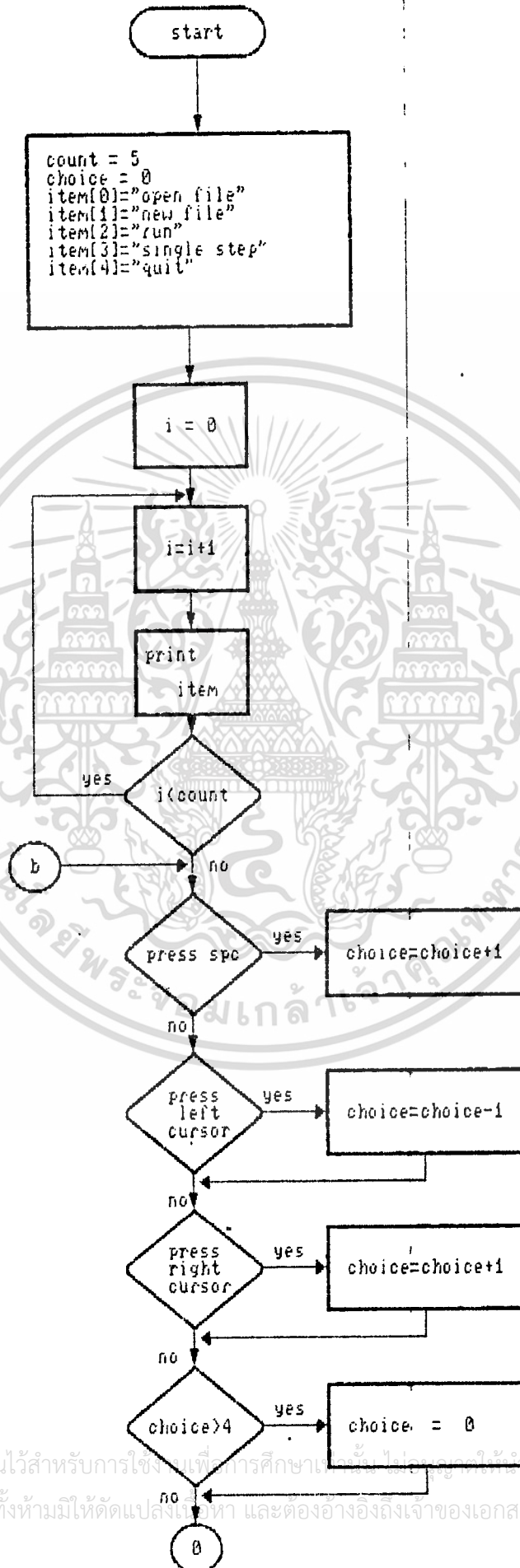
new file เป็นการสร้าง file ข้อมูลใหม่ขึ้นมา โดยที่เราจะป้อนตำแหน่ง x,y และจำนวนจุดที่เราต้องการเก็บไว้เป็น file หนึ่ง เมื่อเราต้องการใช้ชุดข้อมูลนี้ก็ให้เรียก open file ด้วยชื่อ file ที่ตั้งไว้ แล้วจึง run ฟังก์ชันนี้เหมาะกับการเจาะหลายๆตำแหน่งหลายรอบ

หากต้องการเจาะเพียงรอบเดียว ก็ใช้ฟังก์ชัน single step แล้วป้อนตำแหน่งและจำนวนที่ต้องการลงไป แล้วจึง run

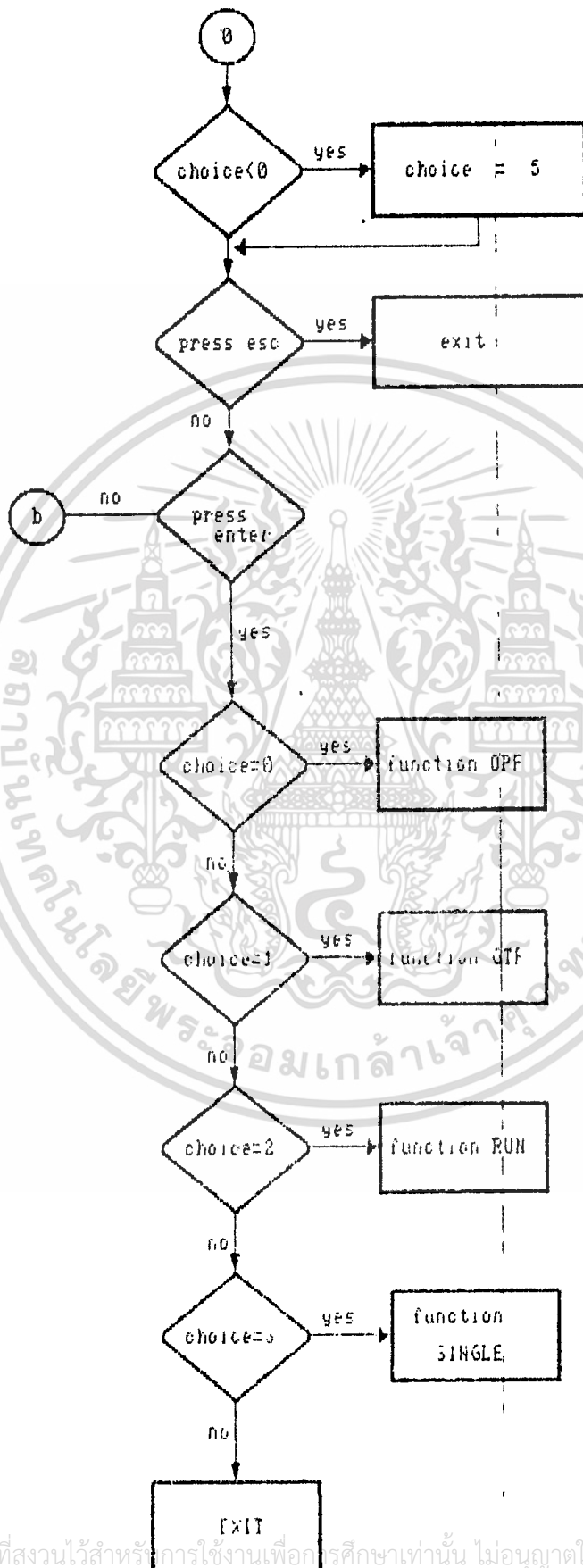
ฟังก์ชัน manual มีไว้สำหรับการเลื่อนตำแหน่งหัวเจาะไปยังตำแหน่งของแผ่น pcb ที่ต้องการด้วยตัวเอง โดยการใช้ ปุ่ม cursor เลื่อนขึ้นลงซ้ายขวาตามต้องการ

ฟังก์ชัน quit เป็นการออกจากโปรแกรม .

Menu

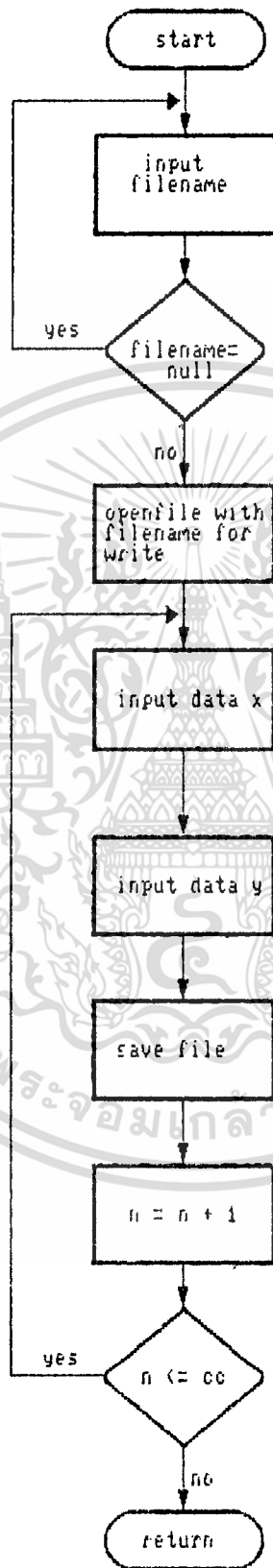


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



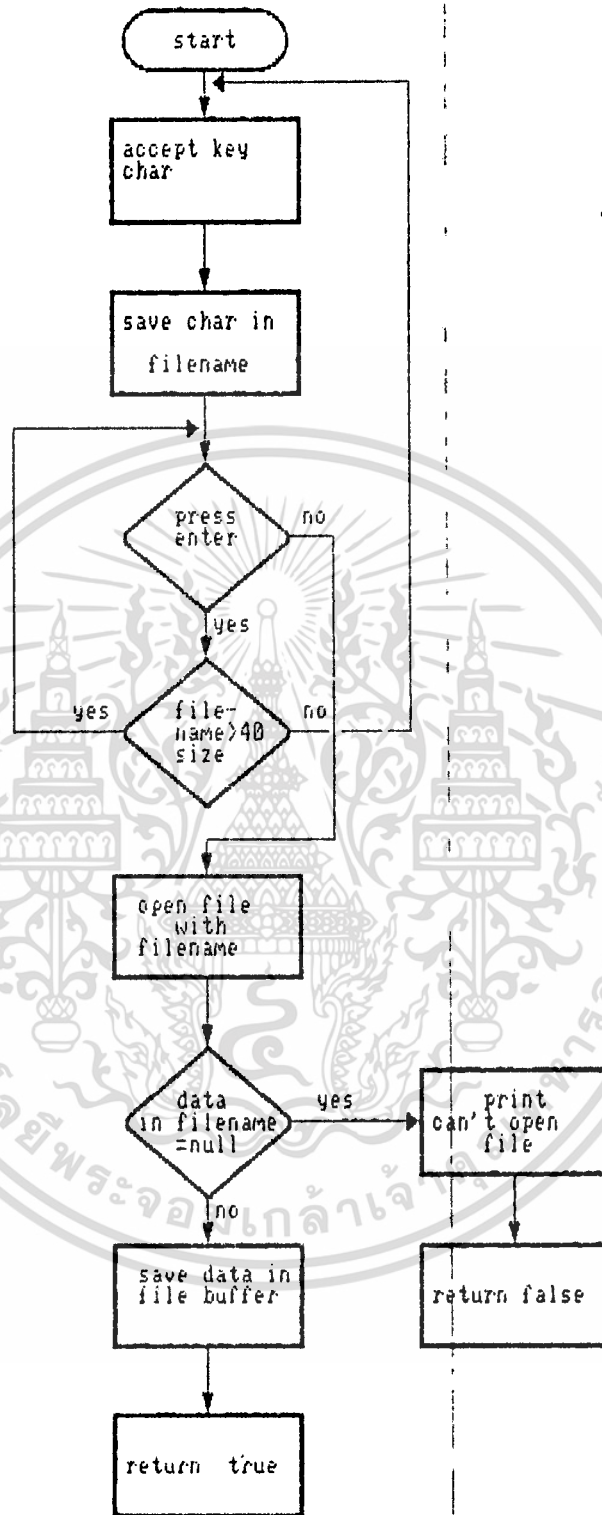
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

gtf



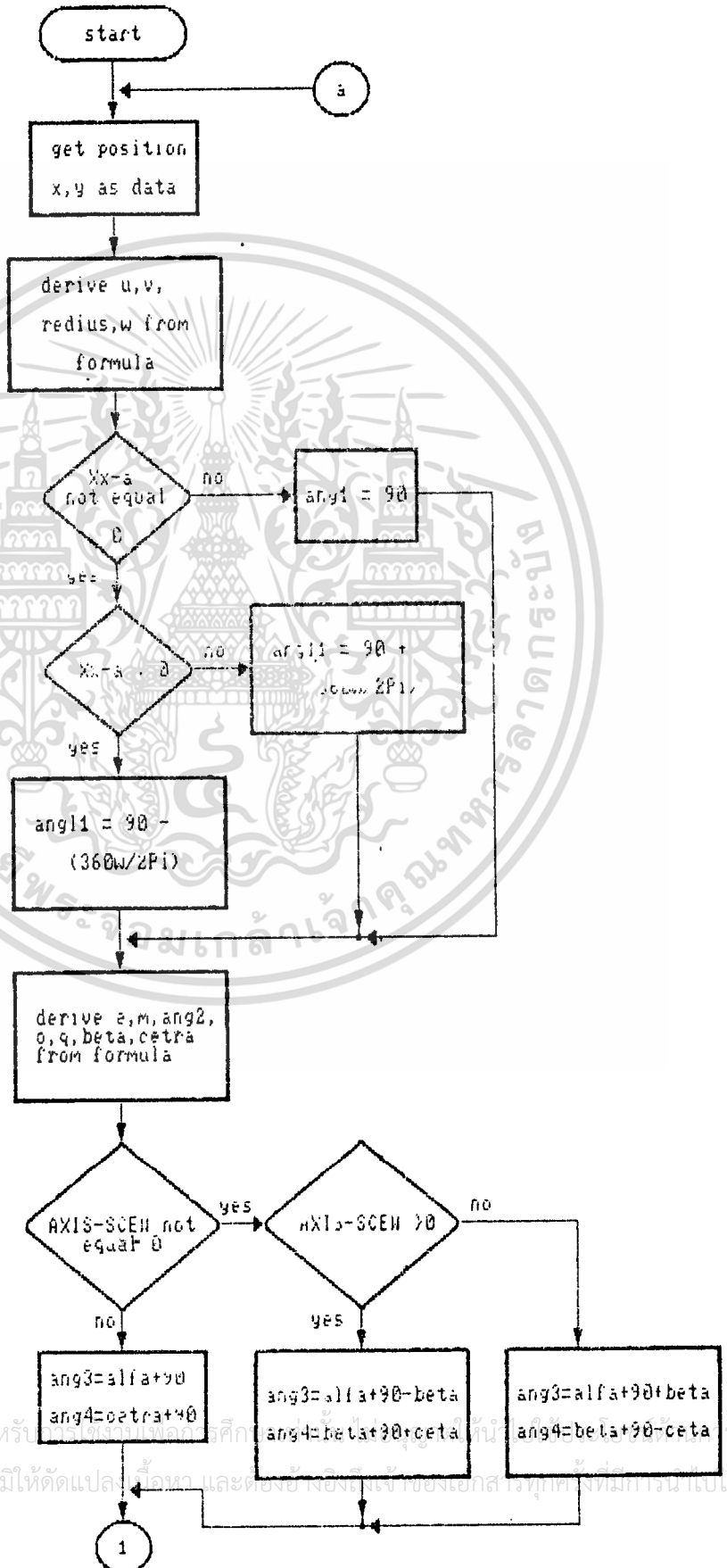
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

opf

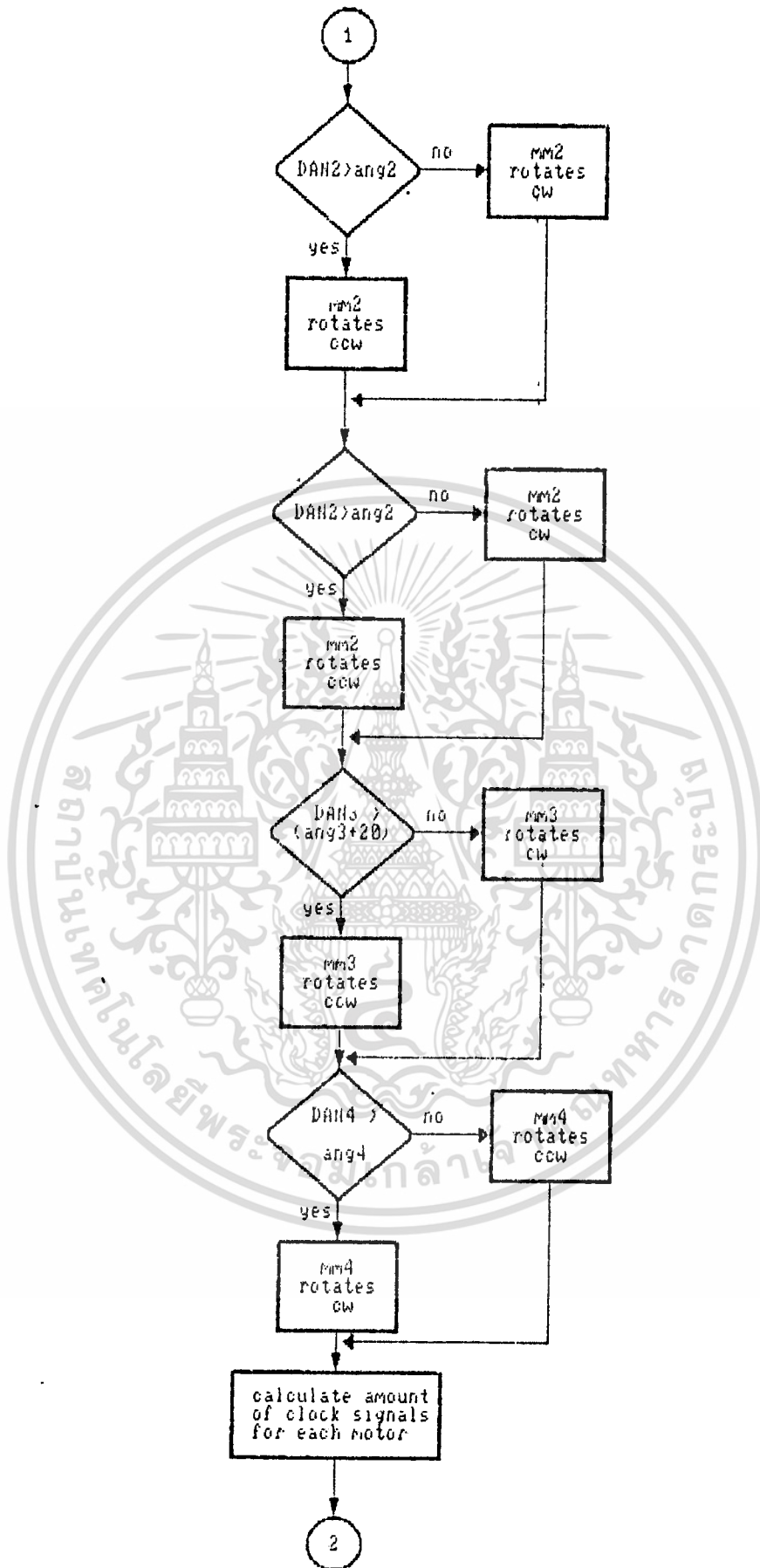


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

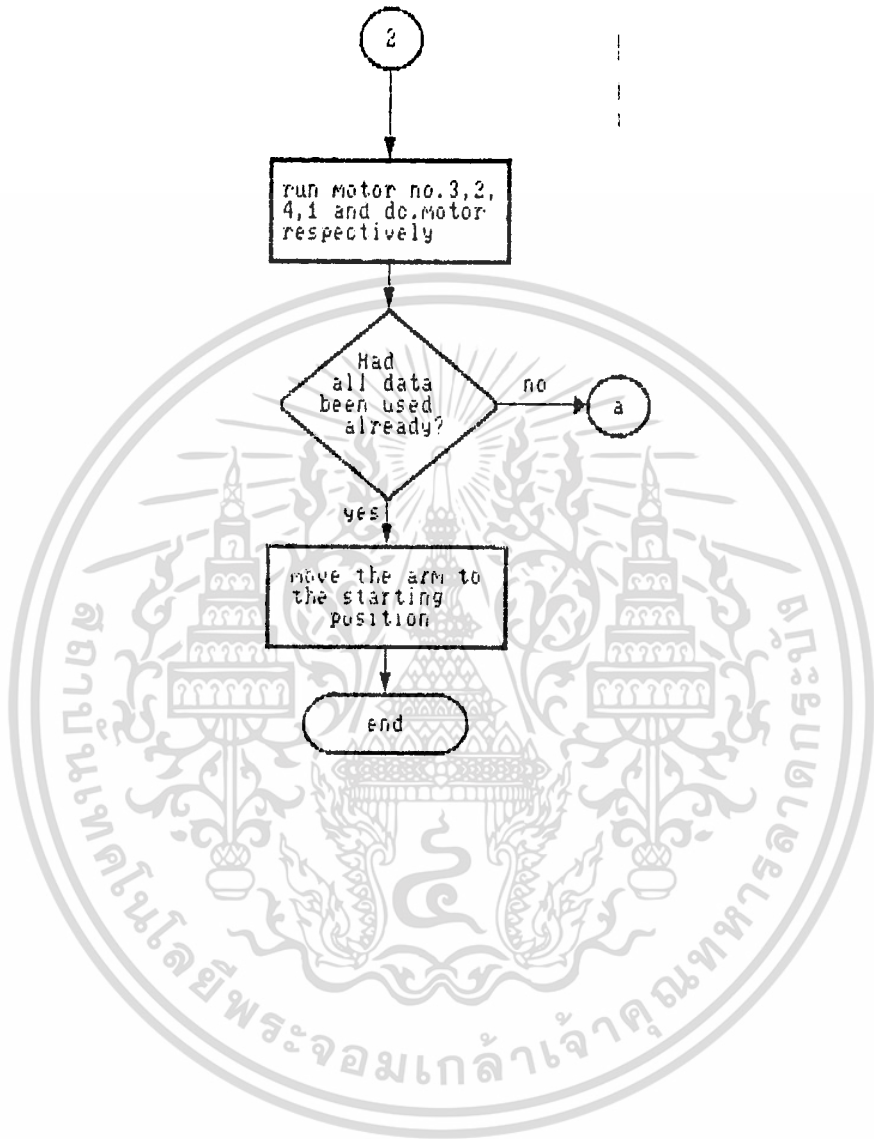
run



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรรมไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/*      Robotic controler program      */
/*-----*/
/*      write by      */
/*      Thiti Tanaree      31.1115      */
/*      Thidaral Pornkarnchananun 31.1112 */
/*      Nithi chareonsri      31.1130      */
/*      KMITL      */
/*-----*/

#include <bios.h>
#include <conio.h>
#include <ctype.h>
#include <dos.h>
#include <fcntl.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define ONE 888#define TWO 890
#define TRUE 1
#define FALSE 0
#define COM 0
#define BSPC 8
#define ENTER 13
#define ESC 27
#define SPC 32
#define SIZE 13
#define NULL 0
#define Pi 3.1415
#define Xx 150 /*----- range of work -----*/
#define Yy 200
#define Yo 150
#define ARM1 251 /*----- legth of arm 1 -----*/
#define ARM2 178 /*----- legth of arm 2 -----*/
#define AXIS 179 /*----- legth of axis -----*/
#define SCEW 152 /*----- legth of screw -----*/
#define ANI 90 /*----- angle of axis -----*/

```

```

#define AN2 90 /*----- angle of motor 2 ---*/
#define AN3 150 /*----- angle of motor 3 ---*/
#define AN4 150 /*----- angle of motor 4 ---*/

```

```

/*----- Define type of struct & union -----*/

```

```

struct style

```

```

    { char left_up;

```

```

char right_up;

```

```

char left_down;

```

```

char right_down;

```

```

char ver_line;

```

```

char hor_line;

```

```

};

```

```

union inkey

```

```

    { char ch[2];

```

```

int i;

```

```

};

```

```

struct instuction

```

```

    { char ins[15];

```

```

char n;

```

```

};

```

```

union byteword

```

```

    { char byte[2];

```

```

unsigned word;

```

```

};

```

```

/*----- Local function -----*/

```

```

;

```

```

;

```

```

void drawblock(char x, char y, char ex, char ey, char style);

```

```

char respmain(char count, char *item[], char choice);

```

```

int getline(char *line, int max);

```

```

int optf(char *arg);

```

```

void menu(void);

```

```

void setcursor(char start, char end);

```

```

int getf ();

```

```

void manual ();

```

```

void run();

```

```

void single();

```

```

/*----- External variable -----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int *filebuff;

```

```

int chkfile;
union byteword fsize;
int, add1 = 12, add2 = 8, add3 = 12, add4 = 8; /*----- Adder number -----*/
double DAN1=AN1, DAN2=AN2, DAN3=AN3, DAN4=AN4;

```

```

/*----- Main Program -----*/

```

```

void main(int argc, char *argv[])
{
    int i; clrscr ();
    if (argc>1)
        chkfile =opf(argv[1]);
    menu();
}

```

```

/*----- Menu Program -----*/

```

```

void menu(void)
{
    char *item[] = {" Open file ",
    " New file ",
    " RUN ",
    " Single step ",
    " Manual ",
    " Quit "};
    int j;
    char count = 6;
    char i, newchoice=0;
    textmode('g');
    textattr(7);
}

```

```

drawblock(2,2,79,24,0);

```

```

for (j=2;j<79;j++)

```

```

{

```

```

    gotoxy(j,2);

```

```

    putchar(205);

```

```

}

```

```

gotoxy(2,2);putchar(213);

```

```

gotoxy(79,2);putchar(184);

```

```

drawblock(20,8,60,16,1);

```

```

for (j=9;j<15;j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(21,j);
printf("                ");
    }

gotoxy(31,10);printf("Robotic Controller");
gotoxy(34,12);printf("Menu Program");
gotoxy(32,13);printf("Press  spacebar");
    for(;;)
    {
setcursor (1,0);
    gotoxy (1,1);textattr(0x07);
for (i=0;i<count;i++)
    puts(item[i]);
newchoice = respmain(count,item,newchoice);
switch(newchoice)
    {
    case 0 : if(opf (NULL))chkfile=TRUE;break;
    case 1 : glf();break;
    case 2 : run();break;
    case 3 : single();break;
    case 4 : manual();break;
    case 5 : clrscr();gotoxy (1,3);
        printf("wait a minute ");
        moveback(DAN1,DAN2,DAN3,DAN4);
        gotoxy (1,5);exit(1);
    }
}
}
}
}
char respmain (char count,char #item[],char choice)
{
    union
    { char ch[2];
int i;
    }lc;
    char x=1 , y=1;
    char arrowchoice = choice;
    gotoxy ((x+choice*SIZE),y);
    textattr (0x07);
    puts(item[arrowchoice]);
    for(;;)
    {

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้ดำเนินการค้า หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(!bioskey(1));
c.i = bioskey(0);
golaxy((x+arrowchoice*SIZE),y);
textattr(0x07);
cputs(item[arrowchoice]);
if (c.ch[0])
switch(c.ch[0])                                /*-- Normal key --*/
{
case ENTER : return arrowchoice;
case SPC : arrowchoice++;break;
case ESC : exit(1);
}
else
switch (c.ch[1])
{
case 0x4B: arrowchoice--;break;
case 0x4D: arrowchoice++;break;
}
if (arrowchoice == count) arrowchoice = 0;
if (arrowchoice < 0) arrowchoice = count-1;
golaxy((x+arrowchoice*SIZE),y);
textattr(0x70);
cputs(item[arrowchoice]);
}
}
/*-----*/
/*          Black making program          */
/*-----*/
void drawblock(char x,char y,char endx,char endy,char n)
{
register char i;
struct style b[2];
b[0].left_up = 218;
b[0].right_up = 191;
b[0].ver_line = 179;
b[1].left_up = 201;
b[1].right_up = 187;
b[1].ver_line = 186;
b[0].left_down = 192;
b[0].right_down = 217;

```

```

    b[0].hor_line    = 196;
    b[1].left_down  = 200;
    b[1].right_down = 188;
    b[1].hor_line   = 205;

    for (i=y+1;i<endy;i++)
    {
gotoxy(x,i);
putch(b[n].ver_line);
gotoxy(endx,i);
putch(b[n].ver_line);
    }
    for (i=x+1;i<endx;i++)
    {
gotoxy(i,y);
putch(b[n].hor_line);
gotoxy(i,endy);
putch(b[n].hor_line);
    }
gotoxy(x,y);putch(b[n].left_up);
gotoxy(x,endy);putch(b[n].left_down);
gotoxy(endx,y);putch(b[n].right_up);
gotoxy(endx,endy);putch(b[n].right_down);
}

/* --      - set cursor -----*/
;
;
void setcursor(char start,char end)
{
    union REGS r;
    r.h.ak = 1;
    r.h.ch = start;
    r.h.cl = end;
    int86(0x10,&r,&r);
}

/*----- Open file -----*/
int  opt (char *arg)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char fname[80];
```

```

    register int i;
    drawblock(4,3,42,9,0);
for (i=0;i<5;i++)
    {
        gotoxy(5,4+i);
        cprintf("                ");
    }
gotoxy(5,7);cprintf("                ");
    if (!strcmp(arg,NULL))
    {
        gotoxy(1,2);
        textattr(7);
        setcursor(6,7);
        gotoxy(21,4);cprintf("                ");
        gotoxy(5,4);cputs("Input file name : ");
        getline(fname,40);
    }
    else strcpy(fname,arg);
if ((fp = fopen(fname,"r"))==NULL)
    {
        gotoxy(5,6);
        cprintf("can't open file");
        return FALSE;
    }
    fsize.word = filesize(fp);
    gotoxy(0,6);cprintf("Data ready");
    filebuff = malloc(fsize.word);
    fread(filebuff,sizeof(filebuff),fsize.word,fp);
    fclose(fp);
    return TRUE;
}

/* ----- File size fn. -----*/

int filesize(FILE*fp)
{
    unsigned int i;
    i=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    getch(fp);
    i++;
}
while (!feof(fp));
    rewind (fp);
    return i-1;
}
/*----- Getline function -----*/

int getline(char *startptr,int max)
{
    register int c;
    register char *ptr = startptr;
    register char *endptr = startptr +max -1;
for (;;)
    {
        c = getch();
        if (c==13) break;
if (c==8)
    {
        putchar(8);
        putchar(' ');
        putchar(8);
        ptr--;
    }
if (ptr < endptr)
    {
        if (isprint(c))
        {
            putchar(c);
            *ptr++=c;
        }
    }
    *ptr = '\0';
    return (c==EOF)? -1:ptr - startptr;
}

```

```

int gtf ()
{
    FILE *fp;
    int n=1;
    int i,q,*pq,w,*pw,c,cc;
    char fname[80];
    gotoxy (17,2);textattr (7);
    setcursor (6,7);
    drawblock(16,4,50,11,0);
    for(i=0;i<6;i++)
    {
        gotoxy(17,5+i);cprintf("
    do
    {
        gotoxy(17,5);
        cputs("Input file name : ");
        getline (fname,22);
    }
    while (!strcmp(fname,NULL));
    gotoxy(19,6);
    cprintf("Number of data : ");
    scanf("%d",&q);
    pq = &q;cc = *pq;
    fp = fopen(fname,"w");
    do
    {
        gotoxy(21,8);
        cprintf("
        gotoxy(17,8);
        cprintf("No.%d Data x : ",n);
        scanf("%d",&w);
        pw = &w;c=*pw;
        putw(c,fp);
        gotoxy(21,9);
        cprintf("
        gotoxy(23,9);
        cprintf("Data y : ",n);
        scanf("%d",&w);
        pw = &w;c=*pw;
        putw(c,fp);
    }
}; ]

```

```

n++;
}
while (n <=cc);
fclose(fp);
gotoxy(40,5);cprintf("Ok.");
return;
}
/*----- Run program -----*/
void run()
{
int a,b,m1,m2,m3,m4,ck1,ck2,ck3,ck4,s,z,i,j;
double m,u,v,e,uu,redius;
double w,o,ang1,ang2,ang3,ang4,dng1,dng2,dng3,dng4;
double q,alfa,beta,cetra;
drawblock(25,5,56,21,0);
for (i=6;i<20;i++)
{
gotoxy(26,i);
cprintf(" ");
}
gotoxy(40,6);
cprintf("RUN");
for (i=25;i<56;i++)
{
gotoxy(i,18);
putch(196);
}
gotoxy(25,18);putch(195);
gotoxy(58,18);putch(180);
gotoxy(27,19);
cprintf("Data size= %d",(fsize.word)/4);
DAN1 = AN1;DAN2 = AN2;DAN3 = AN3;DAN4 = AN4;
for (i=0;i<<(fsize.word)/4;i++)
{
a = *(filehuff++);
b = *(filehuff++);
gotoxy(26,7);
cprintf("No.%d",i+1);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ควรเผยแพร่ หวังสัน อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(27,8);
cprintf("Co_ordinate : x = %d y = %d ",a,b);
for (j=0;j<4;j++)
{
gotoxy(43,j+9);
cprintf("          ");
}
u = fabs(Xx-a);
v = fabs((Yy-b)+Yo);
redius = sqrt((u*u)+(v*v)); /*---- redius ----*/
w = atan(u/v);
if (Xx-a == 0)
ang1 = 90.0;
else if ((Xx-a) >0)
ang1 = 90.0 - ((w*360.0)/(2.0*Pi));
else if ((Xx-a)<0)
ang1 = 90.0 + ((w*360.0)/(2.0*Pi));
else printf("error");
e = sqrt(pow(redius,2)+pow((AXIS-SCEW),2));
m = ((pow(ARM1,2)+pow(ARM2,2))-pow(e,2))/(2.0*ARM1*ARM2);
ang2 = ((360.0*acos(m))/(2.0*Pi));
o = ((pow(ARM1,2)+pow(e,2))-pow(ARM2,2))/(2.0*ARM1*e);
alfa = ((360.0*acos(o))/(2.0*Pi));
q = fabs(AXIS - SCEW);
beta = atan(q/redius);
cetra = 180.0-(eng2+alfa);
switch(AXIS-SCEW != 0)
{
case (AXIS-SCEW > 0 ):ang3 = (alfa + 90.0)-beta;break;
case (AXIS-SCEW < 0):ang3 = beta + alfa + 90.0;break;
default:ang3 = alfa + 90.0;break;
}
switch(AXIS-SCEW != 0)
{
case ((AXIS-SCEW) > 0):ang4 = beta + 90.0 + cetra;break;
case ((AXIS-SCEW) < 0):ang4 = (cetra + 90.0)-beta;break;
default:ang4 = cetra + 90.0;break;
}
}

```

เอกสารนี้ (if (DAN1>ang1)) สำหรับการสั่งการศึกษาค้นคว้า (No. motor) ในกรณีที่มีเงื่อนไขด้านค่า
 ไม่ว่างกรณีที่มีค่าเป็น 0; ทั้งห้ามมิให้ตัดเย็บลงเนื้อหา และต้องยื่นซองถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else          /*-----  0 = ccw , 1 = cw  -----*/

    mm1 = 1;
if (DAN2>ang2)
    mm2 = 0;
else
    mm2 = 1;
if (DAN3>(ang3+20.0))
    mm3 = 0;
else
    mm3 = 1;
if (DAN4>ang4)
    mm4 = 1;
else
    mm4 = 0;
/*---- diff of angle ----*/
dng1 = fabs(DAN1 - ang1);
dng2 = fabs(DAN2 - ang2);
dng3 = fabs(DAN3 - (ang3+20));
dng4 = fabs(DAN4 - ang4);
/*----- clock , rate gear -----*/
ck1 = floor((dng1*200*145)/360);
ck2 = floor((dng2*200*107)/360);
ck3 = floor((dng3*200*145)/360);
ck4 = floor((dng4*200*30)/360);
gotoxy(34,13);cprintf("          ");
gotoxy(27,9);
cprintf("Step 1.  Angle 3 : %6.2f",ang3);
    move3(mm3,ck3);
gotoxy(27,10);
cprintf("Step 2.  Angle 2 : %6.2f",ang2);
    move2(mm2,ck2);
gotoxy(27,11);
cprintf("Step 3.  Angle 4 : %6.2f",ang4);
    move4(mm4,ck4);
gotoxy(27,12);
cprintf("Step 4.  Angle 1 : %6.2f",ang1);
    move1(mm1,ck1);
    gotoxy(27,13);
    cprintf("Step 5.  Screw down"); movedown ();

```



```

    delay(7);
    }
    while (acc <= c);
}

move2(mm,c) /*---- sequentail of motor 2 ----*/
int mm,c;
{
    int base,acc,ot;
    base = 32 ;acc = 0;ot = 0;
do
{
    if (mm == 1)
switch(add2)
{
    case 12:add2 = 6;break;
    case 9:add2 = 12;break;
    case 3:add2 = 9;break;
    case 6:add2 = 3;break;
    default:gotoxy(1,23);cprintf("error");
}
else if (mm == 0)
switch(add2)
{
    case 12:add2 = 9;break;
    case 9:add2 = 3;break;
    case 3:add2 = 6;break;
    case 6:add2 = 12;break;
    default:gotoxy(1,23);cprintf("error");
}
acc++;
ot = base + add2;
outport(ONE,ot);
outp(TWO,1);
outp(TWO,0);
delay (5);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

move3(mm,c)                                     /*---- sequential of motor 3 ----*/
int mm,c;
{
    int base , acc , ot ;
    base = 64;acc = 0;ot = 0;
do
{
    if (mm == 1)
switch (add3)
{
    case 12: add3 = 6;break;
    case 9 : add3 = 12;break;
    case 6 : add3 = 3;break;
    case 3 : add3 = 9;break;
}
else if (mm == 0)
switch (add3)
{
    case 12: add3 = 9;break;
    case 9 : add3 = 3;break;
    case 3 : add3 = 6;break;
    case 6 : add3 = 12;break;
}
    acc++;
    ot = base + add3;
    output(ONE,ot);
    outp(TWO,1);
    outp(TWO,0);
    delay(5);
}
while (acc < c);
}

move4(mm,c)                                     /*---- sequentail of motor 4 ----*/
int mm,c;
{
    int base,acc,ot;
    base = 16;acc = 0;ot = 0;
do
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (mm == 0)
switch (add4)
{
    case 8:add4 = 4;break;
    case 4:add4 = 2;break;
    case 2:add4 = 1;break;
    case 1:add4 = 0;break;
    default:gotoxy(1,23);cprintf("error");
}

    else if (mm == 1)
    switch (add4)
{
    case 8:add4 = 1;break;
    case 4:add4 = 8;break;
    case 2:add4 = 4;break;
    case 1:add4 = 2;break;
    default:gotoxy(1,23);cprintf("error");
}

    acc++;
    ot = base + add4;
    outport(ONE,ot);
    outp(TWO,1);
    outp(TWO,0);
    delay (7);
}
while (acc < c);
}

moveback(angl1,angl2,angl3,angl4) /* move to normally position */
float angl1,angl2,angl3,angl4;
{
    int mm1,mm2,mm3,mm4,ck1,ck2,ck3,ck4;
    float ING1,ING2,ING3,ING4;
    if (AN1 > angl1)
        mm1 = 1;
else mm1 = 0;

    if (AN2 > angl2)
        mm2 = 1; else mm2 = 0;

    if (AN3 > angl3)
        mm3 = 1; else mm3 = 0;
}

```

```

        mm3 = 1;
else mm3 = 0;
    if (AN4 > angl4)
        mm4 = 0;
else mm4 = 1;
    ING1 = fabs(angl1 - AN1);
    ING2 = fabs(angl2 - AN2);
    ING3 = fabs(angl3 - AN3);
    ING4 = fabs(angl4 - AN4);
ck1 = floor((ING1*200*145)/360);
ck2 = floor((ING2*200*107)/360);
ck3 = floor((ING3*200*145)/360);
ck4 = floor((ING4*200*30)/360);
    move3(mm3,ck3);
    move1(mm1,ck1);
    move2(mm2,ck2);
    move4(mm4,ck4);
    DAN1 = AN1; DAN2 = AN2; DAN3 = AN3; DAN4 = AN4;
}
movedown() /*----- move down -----*/
{
    int c,base,acc,ot;
    base = 64;acc = 0;
    c = 1610;
    do
    {
        switch (add3)
        {
            case 12: add3 = 9;break;
            case 9 : add3 = 3;break;
            case 6 : add3 = 12;break;
            case 3 : add3 = .6;break;
        }
        acc++;
        ot = base + add3;
        outpart(ONE,ot);
        outp(TWO,1);
        outp(TWO,0);
        delay(10);
    }
}

```

```

}
while (acc < c);
}

/*-----*/
/*      Single step program      */
/*-----*/

void single()
{
    int a,b;
    int mm1,mm2,mm3,mm4,ck1,ck2,ck3,ck4,w,*pw,q,*pq,r,*pr,c,i,j;
    double s,e,m,o,ang1,ang2,ang3,ang4,dng1,dng2,dng3,dng4;
    double radius,beta,alfa,cetra;
    int u,v;
    drawblock(49,3,74,5,0);
    gotoxy(50,4);
    cprintf(" How many 's' data : "); /*--- put number of data ---*/
    scanf("%d",&w);
    pw=&w; c=*pw;
    for (i=0;i<c;i++)
    {
        for (j=0;j<11;j++)
        { gotoxy(31,6+j);
          cprintf("
          ");
        }
        gotoxy(32,7);cprintf("Max value x=300 : y=200");
        gotoxy(35,8);cprintf("Enter data No. %d",i+1);
        drawblock(30,6,70,17,0);
        gotoxy(40,10);cprintf("
        ");
        gotoxy(32,10);cprintf("Data x : "); /*-- input value x --*/
        scanf("%d",&q);
        pq=&q; s=*pq;
        gotoxy(40,12);cprintf("
        ");
        gotoxy(32,12);cprintf("Data y : "); /*-- input value y --*/
        scanf("%d",&r);
        pr=&r; b=*pr;
        for (j=0;j<6;j++)
        { gotoxy(26,12+j);
          cprintf("

```

```

}
gotoxy(5,21);cprintf("          ");
u = fabs(Xx-a);
v = fabs((Yy-b)+Yo);
s = atan((float)u/v);
if ((Xx-a) == 0)
ang1 = 90.0;
if ((Xx-a)>0)
ang1 = 90.0 - ((s*360)/(2*Pi));
if ((Xx-a)<0)
ang1 = 90.0 + ((s*360)/(2*Pi));
radius = (sqrt(pow((Xx-a),2)+pow(((Yy-b)+Yo),2)));
e = sqrt(pow(radius,2)+pow((AXIS-SCEW),2));
m = ((pow(ARM1,2)+pow(ARM2,2))-pow(e,2))/(2.0*ARM1*ARM2);
ang2 = (acos(m)*360.0)/(2.0*Pi);      /***/
o = ((pow(ARM1,2)+pow(e,2))-pow(ARM2,2))/(2.0*ARM1*e);
alfa = (acos(o)*360.0)/(2.0*Pi);    /***/
q = fabs(AXIS-SCEW);
beta = atan(q/radius);              /***/
cetra = 180.0 - (ang2+alfa);        /***/
switch ((AXIS-SCEW)!=0)
{
case ((AXIS-SCEW)>0):ang3 = (90.0+alfa)-beta;
ang4 = (cetra+beta)+90.0;break;
case ((AXIS-SCEW)<0):ang3 = (90.0+alfa)+beta;
ang4 = (cetra+90.0)-beta;break;
default:ang3 = alfa+90.0;ang4 = 90.0+cetra;break;
}
if (DAN1>ang1)
mm1 = 0;
else
mm1 = 1;
if (DAN2>ang2)
mm2 = 0;
else
mm2 = 1;
if (DAN3>(ang3+20.0))
mm3 = 0;
else
mm3 = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น; อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (DAN4>ang4)
mm4 = 1;
else
    mm4 = 0;
    dng1 = fabs(DAN1-ang1);
    dng2 = fabs(DAN2-ang2);
    dng3 = fabs(DAN3-(ang3+20.0));
    dng4 = fabs(DAN4-ang4);
ck1 = floor((dng1*200*145)/360);
ck2 = floor((dng2*200*107)/360);
ck3 = floor((dng3*200*145)/360);
ck4 = floor((dng4*200*30)/360);
drawblock(25,11,51,18,0); /*-- move --*/
    gotoxy(26,12);
    cprintf("Step.1 Angle 3 :%6.2f",ang3);
move3(mm3,ck3);
    gotoxy(26,13);
    cprintf("Step.2 Angle 2 :%6.2f",ang2);
move2(mm2,ck2);
    gotoxy(26,14);
    cprintf("Step.3 Angle 4 :%6.2f",ang4);
move4(mm4,ck4);
    gotoxy(26,15);
    cprintf("Step.4 Angle 1 :%6.2f",ang1);
move1(mm1,ck1);
    gotoxy(28,16);
    cprintf("Movedown");
movedown();
DAN1 = ang1;DAN2 = ang2;DAN3 =ang3;DAN4 = ang4;
    }
    gotoxy(28,17);
    cprintf("Moveback");
moveback(ang1,ang2,ang3,ang4);
    menu ();
}
manual()
{ char key=0;
int i; drawblock (8,4,27,18,0);
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
for (i=5;i<18;i++)

```

```

    { gotoxy(9,i);cprintf("
"); }
for (i=8;i<27;i++)
    { gotoxy(i,16);putch(196); }
    gotoxy(8,16);putch(195);
    gotoxy(27,16);putch(180);
gotoxy(15,17);cprintf(" Stop ");
gotoxy(10,5);cprintf("Manual control");
gotoxy(12,6);cprintf("use key");
gotoxy(15,8);cprintf("%c :Front",30);
gotoxy(15,9);cprintf("%c :Back",31);
gotoxy(15,10);cprintf("%c :Right",16);
gotoxy(15,11);cprintf("%c :Left",17);
gotoxy(15,12);cprintf("End :Stop");
gotoxy(15,13);cprintf("SPC :Up");
gotoxy(15,14);cprintf("PgDn :Drill");
gotoxy(15,15);cprintf("ESC :Menu");gotoxy(2,15);
while ((key=getch())!=27)
    {
        switch(key)
    { case 32:up();break; /* spc */
      case 61:down();break; /* PgDn */
      case 72:front();break;
      case 60:back();break;
      case 77:right();break;
      case 75:left();break;
    }
        }
menu();
}
left()
{ char key=0;
  int c=0;
  int ct;
  float dng;
  gotoxy(15,17);cprintf(" Left ");
  while ((key=getch())!=79) /* end */
  {
      switch (add1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ `case 12: add1 = 6; break;` เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case 9:add1 = 12;break;
        case 3:add1 = 9;break;
        case 6:add1 = 3;break;
    }
    c++;
    ot=128+add1;
    outport(ONE,ot);
    outp(TWO,1);
    outp(TWO,0);
    delay(10);
    }
    dng = (c*360)/(200*145);
    DAN1 = DAN1-dng;
    manaul();
}
right()
{ char key=0;
  int c=0;
  int ot;
  float dng;
  gotoxy(15,17);printf(" Right ");
  while ((key=getch())!=79)
  {
switch (add1)
{
    case 12:add1 = 9;break;
    case 9 :add1 = 3;break;
    case 3 :add1 = 6;break;
    case 6 :add1 = 12;break;
}
    c++;
    ot = 128+add1;
    outp(ONE,ot);
    outp(TWO,1);
    outp(TWO,0);
    delay(10);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 dng = (c*360)/(200*145);
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 DAN1 = DAN1+dng;

```

        manual();
    }
up()
{ char key=0;
  int c,ot;
  float dng;c=0;
  gotoxy(15,17);cprintf(" Up ");
  while ((key=getch())!=79)
  {
      switch(add3)
{ case 12:add3 = 6;break;
  case 9 :add3 = 12;break;
  case 3 :add3 = 9;break;
  case 6 :add3 = 3;break;
}
      c++;
      ot = 64+add3;
      outp(ONE,ot);
      outp(TWO,1);
      outp(TWO,0);
      delay(10);
  }
      dng =(c*360)/(200*145);
      DAN3 = DAN3+dng;
      manual();
}
down()
{ char key=0;
  int ot,c,mm4,ck4;
  float dng,dng4;c=0;
  gotoxy(15,17);cprintf(" Down ");
  dng = 360.0-(DAN3+DAN2);
  if (dng>DAN4)
      mm4 = 0;
      else mm4 = 1;
      dng4 = fabs(DAN4-dng);
      ck4 = floor((dng4*200*30)/360);
      move4(mm4,ck4);

```

```

{
    switch(add3)
    {
        case 12: add3 = 9; break;
        case 9 : add3 = 3; break;
        case 3 : add3 = 6; break;
        case 6 : add3 = 12; break;
    }

    c++;
    ot = 64+add3;
    outp(ONE,ot);
    outp(TWO,1);
    outp(TWO,0);
    delay(10);
}

dng = (c*360)/(200*145);
DAN3 = DAN3-dng;
manual();
}

front()
{ char key=0;
  int ot,c;
  float dng;c=0;
  gotoxy(15,17);cprintf(" Front ");
  while ((key=getch())!=79)
  {
switch (add2)
{
    case 12: add2=6; break;
    case 9 : add2=12; break;
    case 3 : add2=9; break;
    case 6 : add2=3; break;
}

c++;
ot = 32+add2;
outp(ONE,ot);
outp(TWO,1);
outp(TWO,0);
delay(10);
}
}

```

```

    }
    dng = (c*360)/(200*107);
    DAN2 = DAN2+dng;
    manual();
}
back()
{ char key=0;
  int ot,c;
  float dng;c=0;
  gotoxy(15,17);printf(" Back ");
  while ((key=getch())!=79)
  {
    switch (add2)
    {
case 12:add2 = 9;break;
case 9 :add2 = 3;break;
case 3 :add2 = 6;break;
case 6 :add2 = 12;break;
    }
    c++;
    ot = 32+add2;
    outp(ONE,ot);
    outp(TWO,1);
    outp(TWO,0);
    delay(10);
  }
  dng = (c*360)/(200*107);
  DAN2 = DAN2-dng;
  manual();
}
/*----- END -----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณอาจารย์และเจ้าหน้าที่ใน shop เครื่องกล ที่ช่วยให้อุปกรณ์และอำนวยความสะดวกในการทำงานและให้คำแนะนำต่าง ๆ

สุดท้ายนี้ขอกราบขอบพระคุณอาจารย์ โทศล ขวนขยัน อาจารย์ที่ปรึกษาของเราที่ช่วยให้แนวความรู้เกี่ยวกับการสร้างแขนกลและให้คำแนะนำในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้น ตลอดจนช่วยเอาใจใส่และให้กำลังใจในการทำงานแก่พวกเราเสมอ จนทำให้โครงการของเราสำเร็จลุล่วงไปด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Analog and digital control systems

Ramakant Gayakwad

Leonard Sokoloff

2. Robot Design handbook

Gerry E. Andeen

3. Engineering Drawing and Design

Cecil Jensen

4. Handbook of industrial robotics

Shimon Y. Nof

