



ปีการศึกษา 2533

การควบคุมแขนกลเพื่องานเชื่อมโลหะ

WELDING ROBOT ARM

โดย

อเนศ สวรรณผ่อง 301394

สรเชษฐ์ รตโนภาส 301329

อาจารย์ที่ปรึกษา

รศ.ดร. โยธิน เปรมปราณีรัตน์

ผศ.ดร. จงกล งามวิวิทย์

ปริญญานิพนธ์ปีการศึกษา 2533

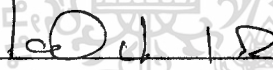
ภาควิชา ระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การควบคุมแบบกลเพื่องานเชื่อมโลหะ

ผู้จัดทำ

1. ชเนศ สุวรรณม่วง 301094

2. สุรเชษฐ์ รตโนภาส 301329



(รศ.ดร.โยชิน เปรมปราณีรัตน์)

อาจารย์ที่ปรึกษา



(ผศ.ดร.จงกล จามวิวิทย์)

อาจารย์ที่ปรึกษา

เลขหมู่ T 33095 ศ ๗
เลขทะเบียน 027928
วัน, เดือน, ปี ๑๙๙๓

การควบคุมแขนกลเพื่องานเชื่อมโลหะ

ชเน็ค สุวรรณผ่อง
สุรเชษฐ์ รัตนภาส

รศ.ดร.โยชิน เปรมปราณีรัตน์
ผศ.ดร.จงบกล งามวิวิทย์
อาจารย์ที่ปรึกษา
ปีการศึกษา 2533

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ จะอธิบายถึงการออกแบบวิธีการควบคุมแขนกลให้ทำงานแบบการควบคุมเส้นทางต่อเนื่อง (CONTINUOUS PATH CONTROL) ซึ่งพัฒนามาจากการควบคุมตำแหน่ง (POSITION CONTROL) โดยใช้เครื่องคอมพิวเตอร์ส่วนบุคคลเป็นหน่วยประมวลผลความเร็ว, เส้นทางเคลื่อนที่ และตำแหน่งที่ต้องการไปแล้วส่งสัญญาณไปควบคุมวงจรมอเตอร์ภายนอก เพื่อกระตุ้นให้สเตปปิ้งมอเตอร์ (STEPPING MOTOR) ในแต่ละข้อต่อ (JOINT) หมุนไปในมุมที่เหมาะสมเพื่อให้แขนกลเคลื่อนที่ไปยังตำแหน่งที่ต้องการได้อย่างถูกต้อง วิธีการใช้ประมวลผลในการหาค่ามุมในแต่ละช่วงแขน (LINK) ในปริญญานิพนธ์นี้ใช้วิธีทางเรขาคณิต ซึ่งวิธีนี้สามารถออกแบบการควบคุมได้ง่ายแต่วิธีทางเรขาคณิตนี้สามารถนำไปใช้ได้กับแขนกลชนิดระนาบเดียว (Planar) ทั่วๆไปได้เท่านั้น

WELDING ROBOT ARM

THANES SUWANPONG

SURACHET RATANOPAS

ASSOCIATE PROF. DR. YOTIN PREMFRANEERACH

ASSISTANT PROF. DR. JONGKOL NGAMVIVIT

ADVISORS

1990

ABSTRACT

This thesis describes the design method for control Robot Arm in form of point to - point control and continuous path control that develop from position control. The design uses PERSONAL COMPUTER to calculate speed, path and target position, and interface with MOTOR DRIVE CIRCUIT that send the control signal to stepping motor in each joint. The stepping motor in each joint is rotated in appropriate angle and make the link move, so the end effector can approach to the target position. This thesis uses Geomatic Method to solve the angle of each link. This method is easy to design for controlling but can only apply to general planar Robot Arm.

สารบัญ

หน้า

บทคัดย่อ

Abstract

บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการ	5
บทที่ 3	การคำนวณและการสร้าง	19
บทที่ 4	การทดลองและผลการทดลอง	39
บทที่ 5	บทวิจารณ์และสรุป	47

ภาคผนวก

ภาคผนวก A	คู่มือการใช้งานแขนกล ROBOT-1
ภาคผนวก B	โปรแกรมการควบคุมแขนกลโดยวิธีทางเรขาคณิต
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

สารบัญรูปภาพ

	หน้า
รูป 1.1	จำนวนประชากรหุ่นยนต์คิดเป็นเปอร์เซ็นต์ของประเทศสหรัฐอเมริกา . 1
รูป 1.2	จำนวนประชากรหุ่นยนต์ในแต่ละปีของประเทศสหรัฐอเมริกา 2
รูป 1.3	หุ่นยนต์ที่ใช้ในการวาดภาพ 3
รูป 1.4	หุ่นยนต์ที่ใช้ในงานเชื่อมเป็นจุด ๆ 3
รูป 2.1	ข้อต่อชนิดต่าง ๆ 5
รูป 2.2	ส่วนประกอบต่าง ๆ ของแขนกล 6
รูป 2.3	ลักษณะการเคลื่อนที่แบบต่าง ๆ 7
รูป 2.4	แขนกลที่ใช้ทดลอง 9
รูป 2.5	การกำหนดระนาบแกนต่าง ๆ 9
รูป 2.6	การเลื่อนระนาบ 13
รูป 2.7	ช่วงแขนในสองมิติ 14
รูป 2.8	แขนกลในระนาบสองมิติ 16
รูป 3.1	สถานะของแขนกลบนจอภาพคอมพิวเตอร์ 19
รูป 3.2	ความหมายของส่วนต่าง ๆ ของภาพบนจอคอมพิวเตอร์ 20
รูป 3.3	โปรแกรมควบคุมแขนกล 21
รูป 3.4	แสดงมุมปลายแขน (WRIST ANGLE) 22
รูป 3.5	ผังการทำงานของโปรแกรมย่อย GEO_CAL 23
รูป 3.6	ผังขั้นตอนการทำงานของโปรแกรมย่อย GEO_CAL 25
รูป 3.7	ผังการทำงานของโปรแกรมย่อย AMUST_DRV 27
รูป 3.8	ผังลำดับการทำงานของโปรแกรมย่อย AMUST_DRV 28
รูป 3.9	ผังการทำงานของโปรแกรมย่อย POSITION 29
รูป 3.10	ผังลำดับการทำงานของโปรแกรมย่อย POSITION 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		หน้า
รูป 3.11	ผังการทำงานของโปรแกรมย่อย CONTINUOUS	31
รูป 3.12	ผังลำดับการทำงานของโปรแกรมย่อย CONTINUOUS	32
รูป 3.13	ผังลำดับการทำงานของโปรแกรมย่อย RUN COMMAND FILE	34
รูป 3.14	ผังลำดับการทำงานของโปรแกรมหลัก	36
รูป 3.15	ผังลำดับการทำงานของโปรแกรมย่อย RTOKEN	37
รูป 3.16	ผังลำดับการทำงานของโปรแกรมย่อย REXE	38
รูป 4.1	เส้นตรงวาดโดยใช้ความยาวส่วนย่อยแตกต่างกัน	39
รูป 4.2	เส้นตรงแนวราบความยาวแตกต่างกัน	42
รูป 4.3	เส้นตรงแนวตั้งความยาวแตกต่างกัน	43
รูป 4.4	ตารางช่องห่าง 1 ซม. วาดจากแขนกล	44
รูป 4.5	ผลลัพธ์การเคลื่อนที่แบบควบคุมตำแหน่ง	45

สารบัญตาราง

	หน้า
ตาราง 2.1 พารามิเตอร์ต่าง ๆ ของแบบกล	10
ตาราง 2.2 เปรียบเทียบข้อดีข้อเสียของการคำนวณแต่ละแบบ	17
ตาราง 4.1 แสดงค่าผิดพลาดทางตำแหน่งแนวราบเมื่อใช้ช่วงย่อยต่าง ๆ กัน..	40
ตาราง 4.2 แสดงค่าผิดพลาดทางตำแหน่งแนวตั้งเมื่อใช้ช่วงย่อยต่าง ๆ กัน...	41

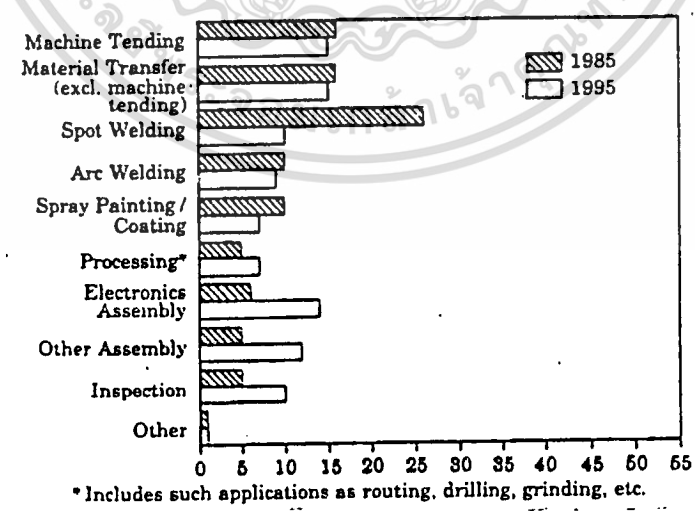


บทที่ 1

บทนำ

ในปัจจุบันนี้โรงงานอุตสาหกรรมหลายแห่งในโลก ได้มีการนำเทคโนโลยีทางด้านหุ่นยนต์เข้ามาใช้ทดแทนมนุษย์ในการผลิตสินค้า เช่น อุตสาหกรรมผลิตรถยนต์ ซึ่งหุ่นยนต์นั้นสามารถจะทำงานได้อย่างถูกต้อง , แน่นนอนและมีความรวดเร็วกว่ามนุษย์ และสามารถที่จะทำงานในสภาพแวดล้อมที่เป็นอันตรายต่อมนุษย์ เช่นงานในห้องอบสารเคมี บางอย่าง , งานในห้องที่มีอุณหภูมิสูง เป็นต้น วัตถุประสงค์ที่ใช้หุ่นยนต์ในแต่ละงานก็ต่างกัน เช่นใช้ หุ่นยนต์ในงานเชื่อมโลหะ (welding Robot) , งานพ่นสี (Spray Painting /Coating Robot) ปริมาณของหุ่นยนต์ที่นำไปใช้ในโรงงานแต่ละปีก็มีปริมาณที่สูงขึ้นเรื่อยๆ

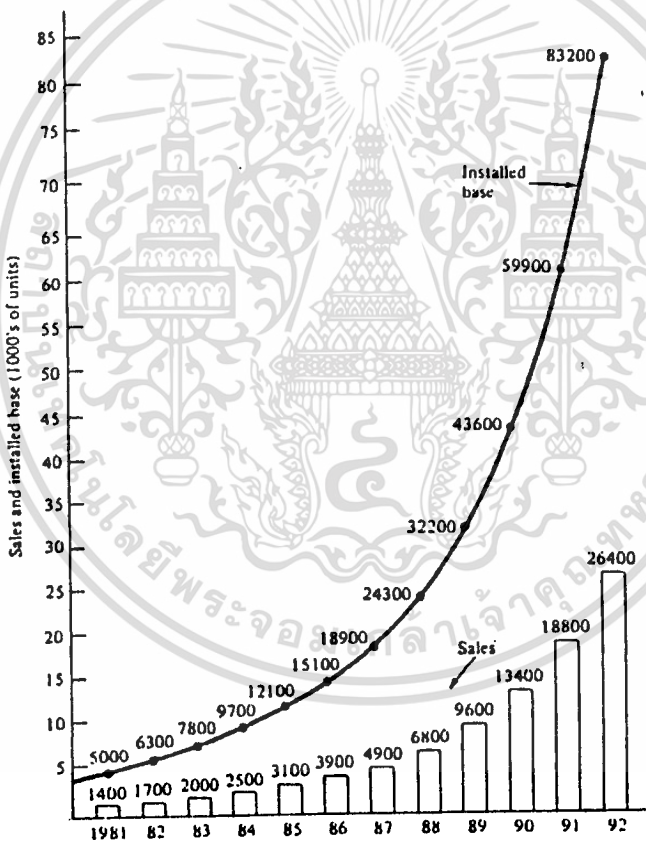
ในประเทศสหรัฐอเมริกาได้มีการสำรวจปริมาณคิดเป็นเปอร์เซ็นต์ของประชากรหุ่นยนต์ในแต่ละปีแยกเป็นชนิดๆ ดังรูป 1.1



รูป 1.1 จำนวนประชากรหุ่นยนต์คิดเป็นเปอร์เซ็นต์ของประเทศสหรัฐอเมริกา

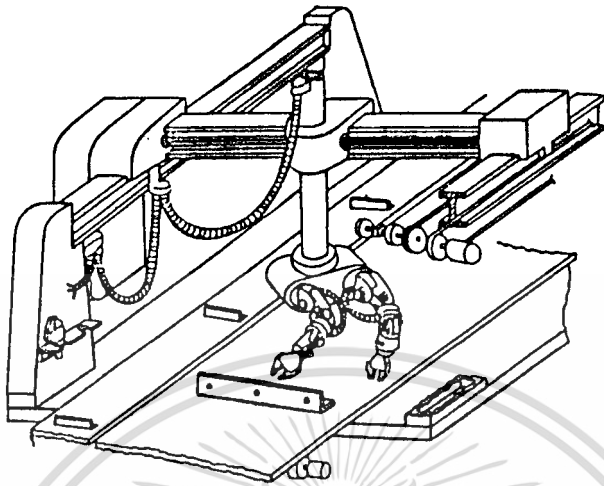
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังมีการทำนายได้อีกว่า ในอนาคตนั้นปริมาณของ หุ่นยนต์ที่นำ มาใช้ในโรงงานอุตสาหกรรมจะมีมากขึ้นทุกๆปี ดังรูป 1.2

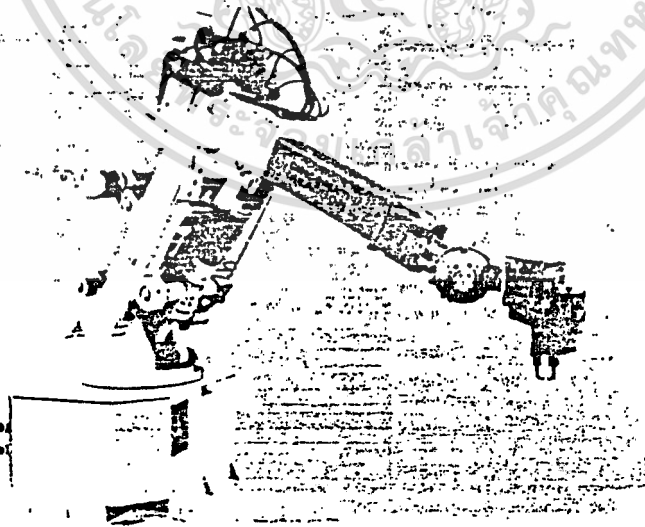


รูป 1.2 จำนวนประชากรหุ่นยนต์ในแต่ละปีของประเทศสหรัฐอเมริกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 หุ่นยนต์ที่ใช้ในการวาดภาพ



รูปที่ 1.4 หุ่นยนต์ที่ใช้ในงานเชื่อมเป็นจุดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนในอุตสาหกรรมของประเทศไทยนั้น บทบาทของเทคโนโลยีทางด้านหุ่นยนต์ยังมีน้อยมากเทคโนโลยีทางด้านนี้ยังคงเป็นเพียงการศึกษา, วิจัย ซึ่งหลายๆ สถาบันการศึกษา กำลังกระทำอยู่

ปริญญาณพนธ์ฉบับนี้เป็นการศึกษาออกแบบส่วนของการควบคุมการเคลื่อนที่ โดยมีวัตถุประสงค์ที่จะสามารถทำให้แขนกลเคลื่อนที่แบบต่อเนื่อง (Continuous path control) ซึ่ง การเคลื่อนที่แบบนี้สามารถนำไปใช้กับการเคลื่อนที่ของ หุ่นยนต์เชื่อม (Welding Robot), หุ่นยนต์พ่นสี (Spray Painting/Coating Robot) เป็นต้น



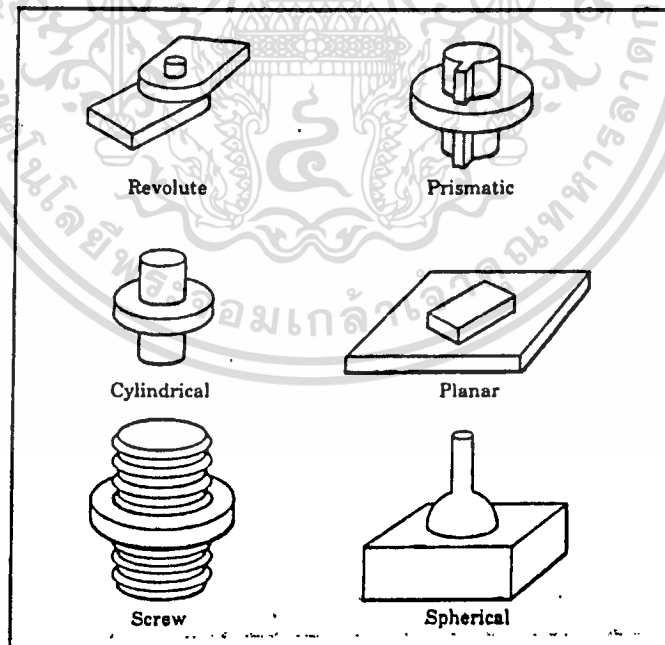
บทที่ 2

ทฤษฎีและหลักการ

ทฤษฎีเบื้องต้น

ข้อต่อ (JOINT) คือ จุดต่อระหว่างแกนของแขนกลซึ่งมีอยู่หลายชนิด แต่แบ่งตามลักษณะการทำงานได้ 6 ชนิดคือ

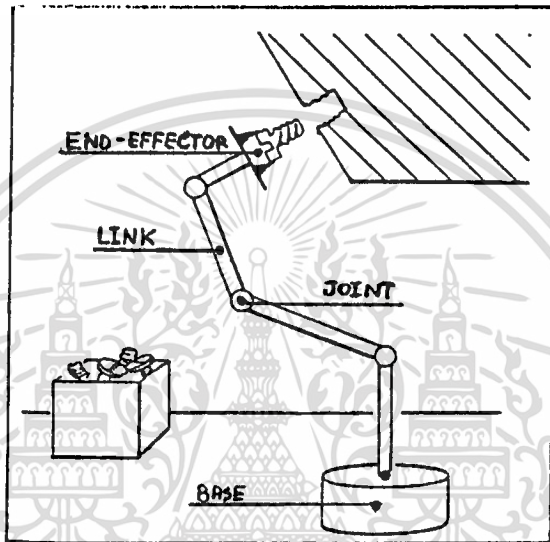
- ข้อต่อแบบหมุน (Revolute joint)
- ข้อต่อแบบเลื่อนเข้า-ออก (Prismatic joint)
- ข้อต่อแบบทรงกระบอก (Cylindrical)
- ข้อต่อแบบสกรู (Screw)
- ข้อต่อแบบทรงกลม (Spherical)
- ข้อต่อแบบระนาบ (Planar)



รูปที่ 2.1 ข้อต่อชนิดต่างๆ

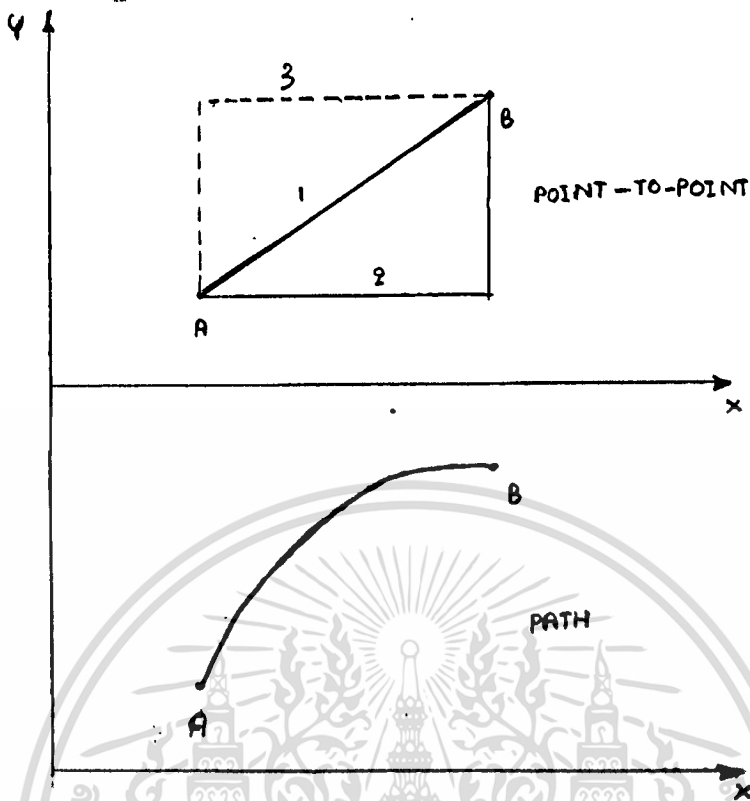
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- # ช่วงแขน (LINK) คือ ส่วนที่เป็นแกนอยู่ระหว่างข้อต่อ หรืออยู่ระหว่าง JOINT กับมือหนีบ (END-EFFECTOR)
- # มือหนีบ (END-EFFECTOR) คือ ส่วนปลายสุดของแขนกลที่จะถูกนำไปใช้ในการทำงาน
- # ฐาน (BASE) ส่วนฐาน เป็นส่วนที่ต้องรับแรงทั้งหมดที่ถ่ายทอดผ่าน LINK



รูป 2.2 ส่วนประกอบต่าง ๆ ของแขนกล

- # การควบคุมตำแหน่ง (Position Control) คือ การเคลื่อนที่จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งโดยไม่คำนึงถึงเส้นทางในการเคลื่อนที่ไป เรียกอีกอย่างหนึ่งว่าการเคลื่อนที่แบบจุดไปจุด (point to point)
- # การควบคุมเส้นทาง (Path Control) คือ การเคลื่อนที่จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งโดยคำนึงถึงเส้นทางในการเคลื่อนที่



รูป 2.3 ลักษณะการเคลื่อนที่แบบต่าง ๆ

ข้อกำหนดของ Denavit และ Hartenberge เป็นข้อกำหนดที่ใช้ในการกำหนดแกน x, y, z โดยมีข้อกำหนดดังนี้

1. การกำหนดแกน z
 - ในกรณีที่ เป็นข้อต่อแบบหมุน แกน z จะตั้งฉากกับการหมุน
 - ในกรณีที่ เป็นข้อต่อแบบเลื่อนเข้า-ออก แกน z จะวางตัวตามการเคลื่อนที่
2. การกำหนดแกน x แกน x จะอยู่ในทิศทางเดียวกันทุก ๆ ระนาบ

วิธีการออกแบบส่วนควบคุมการเคลื่อนที่

ในการออกแบบส่วนควบคุมแขนกลนั้น เราต้องวิเคราะห์ลักษณะของแขนกลตามข้อต่อต่างๆ เสียก่อนโดยพิจารณาสิ่งต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ต้องพิจารณาถึง จำนวนช่วงแขน, จำนวนข้อต่อ, ลักษณะของช่วงแขน และข้อต่อ เช่น ความยาวของช่วงแขน, ชนิดของข้อต่อ, ลักษณะการเคลื่อนที่ของข้อต่อ, ความสามารถในการเคลื่อนที่ของข้อต่อ เป็นต้น

2. ลักษณะของมือหนีบ

เมื่อรู้ถึงลักษณะข้อมูลของช่วงแขน, ข้อต่อ และ มือหนีบแล้ว ปัญหาต่อไปก็คือ เมื่อกำหนดตำแหน่งที่จะให้แขนกลเคลื่อนที่ไป, เส้นทางการเคลื่อนที่, ความเร็ว, ทำอย่างไร ถึงจะสามารถไปถึงสิ่งที่กำหนดเหล่านั้นได้...

สิ่งที่ต้องคำนวณเพื่อแก้ปัญหานี้ก็คือ "มุมระหว่างช่วงแขนแต่ละช่วงแขน ที่กระทำต่อกัน สำหรับการเคลื่อนที่จากจุดหนึ่งไปอีกจุดหนึ่ง "

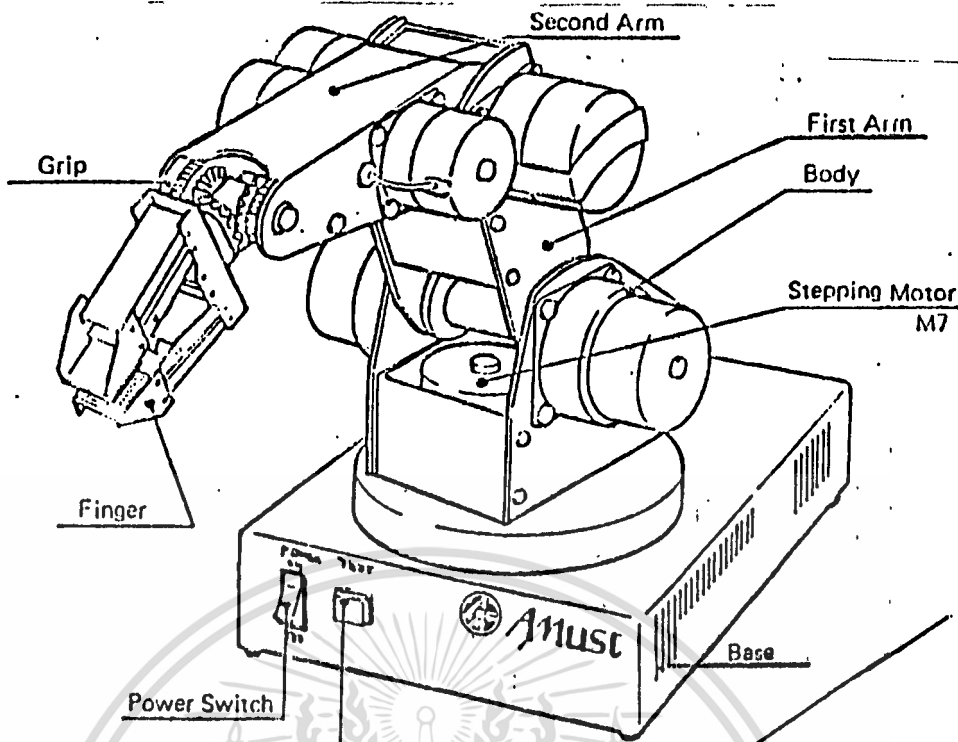
การคำนวณ

วิธีที่ใช้ในการคำนวณหามุมระหว่างช่วงแขนแต่ละช่วงแขนที่กระทำต่อกัน มีอยู่

3 วิธี คือ

1. การคำนวณแบบไปข้างหน้า (Forward Kinematic Method)
2. การคำนวณแบบย้อนกลับ (Inverse Kinematic Method)
3. การคำนวณโดยใช้วิธีทางเรขาคณิต (Geometric Method)

จากรูป 2.4 เป็นหุ่นยนต์ที่ใช้ในการทดลอง เป็นหุ่นยนต์ที่มี 5 ระบุาบอิสระ (Degree of Freedom) จากข้อกำหนดของ Denavit และ Hartenberg สามารถกำหนดโคออดิเนตเฟรมที่ข้อต่อต่างๆ ได้ดังรูปที่ 2.5 และมีพารามิเตอร์ต่างๆ ดังแสดงในตารางที่ 2.1



รูป 2.4 แสดงการทดลอง

2.1. DIRECT KINEMATICS IN ROBOTICS

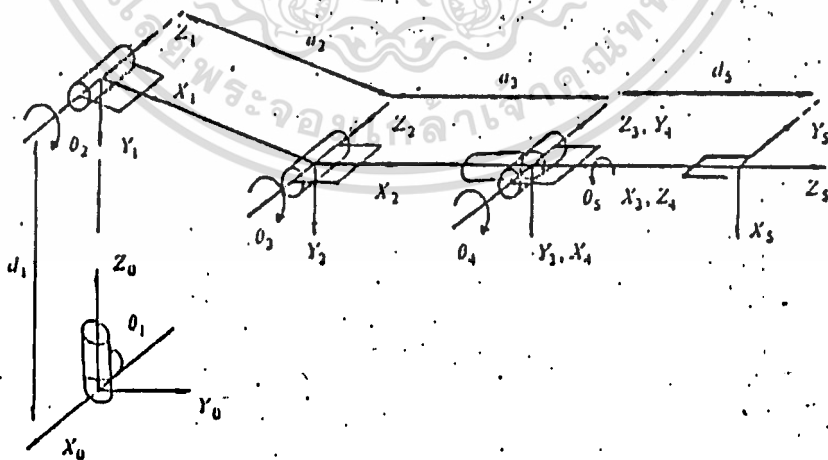


Figure 2.11.5 Structure of coordinate frames.

รูป 2.5 การกำหนดระนาบแกนต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Joint	θ_i	α_i	a_i	d_i
1	θ_1	-90	0	d_1
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4	90	0	0
5	θ_5	0	0	d_5

ตารางที่ 2.1 พารามิเตอร์ต่าง ๆ

จากตารางเราสามารถเขียนเมตริกในการเปลี่ยนระนาบ (Transformation Matrix) จากข้อต่อที่ $i-1$ ไปยังข้อต่อที่ i ได้เป็น

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S_i = \sin\theta_i \quad C_i = \cos\theta_i \quad S_{i,j} = \sin(\theta_i + \theta_j) \quad C_{i,j} = \cos(\theta_i + \theta_j)$$

การเปลี่ยนระนาบจากปลายมือลงมาที่ฐานคือ 0T_5 จะหาได้จาก

$${}^0T_5 = {}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4 * {}^4A_5$$

ซึ่งมีส่วนประกอบต่างๆ เป็นดังนี้

$$({}^0T_5)_{1,1} = C_1 C_2 C_3 C_4 C_5 - C_1 S_2 S_3 C_4 C_5 - S_1 S_5$$

$$({}^0T_5)_{1,2} = -C_1 C_2 C_3 C_4 S_5 + C_1 S_2 S_3 S_4 S_5 - S_1 C_5$$

$$\begin{aligned}
({}^0T_5)_{13} &= C_1 C_{23} S_4 + C_1 S_{23} C_4 \\
({}^0T_5)_{14} &= d_5 C_1 C_{23} S_4 + d_5 C_1 S_{23} C_4 + a C_1 (C_{23} + C_2) \\
({}^0T_5)_{21} &= S_1 C_{23} C_4 C_5 - S_1 S_{23} S_4 S_5 + C_1 S_5 \\
({}^0T_5)_{22} &= -S_1 C_{23} C_4 S_5 + S_1 S_{23} S_4 S_5 + C_1 C_5 \\
({}^0T_5)_{23} &= S_1 C_{23} S_4 + S_1 S_{23} C_4 \\
({}^0T_5)_{24} &= d_5 S_1 C_{23} S_4 + d_5 S_1 S_{23} C_4 + a S_1 (C_{23} + C_2) \\
({}^0T_5)_{31} &= -S_{23} C_4 C_5 - C_{23} S_4 C_5 \\
({}^0T_5)_{32} &= S_{23} C_4 S_5 + C_{23} S_4 S_5 \\
({}^0T_5)_{33} &= -S_{23} S_4 + C_{23} C_4 \\
({}^0T_5)_{34} &= -d_5 S_{23} S_4 + d_5 C_{23} C_4 + d_1 - a(S_{23} + S_2) \\
({}^0T_5)_{41} &= ({}^0T_5)_{42} = ({}^0T_5)_{43} = 0 \\
({}^0T_5)_{44} &= 1
\end{aligned}$$

เนื่องจากเมตริกในการเปลี่ยนระนาบ T จะเป็น เมตริก 4*4 ซึ่งมีส่วนประกอบ

ดังนี้

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- a: เวกเตอร์ในการเข้าถึง (Approach vector)
- o: เวกเตอร์ในการหมุน (Orientation vector)
- n: เวกเตอร์ตั้งฉากกับการเข้าถึง (Normal vector และ $n=oxa$)
- p: เวกเตอร์ตำแหน่ง (Position vector)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธี การคำนวณแบบไปข้างหน้า

จากลักษณะของแขนกล จะรู้ถึงเมตริกในการเปลี่ยนระนาบเมื่อนำไปเปรียบเทียบกับในแต่ละส่วนประกอบ จะได้ว่า

$$p_x = ({}^0T_5)_{14}$$

$$p_y = ({}^0T_5)_{24}$$

$$p_z = ({}^0T_5)_{34}$$

$$a_x = ({}^0T_5)_{13}$$

$$a_y = ({}^0T_5)_{23}$$

$$a_z = ({}^0T_5)_{33}$$

$$o_x = ({}^0T_5)_{12}$$

$$o_y = ({}^0T_5)_{22}$$

$$o_z = ({}^0T_5)_{32}$$

$$n_x = ({}^0T_5)_{11}$$

$$n_y = ({}^0T_5)_{21}$$

$$n_z = ({}^0T_5)_{31}$$

ดังนั้นถ้าเราทราบตำแหน่งที่ต้องการและลักษณะการเข้าหาจุดนั้นของปลายแขน ก็จะหาค่ามุมต่างได้โดยการแก้สมการทั้งหมดพร้อมๆกัน

วิธี การคำนวณแบบย้อนกลับ

จะหาย้อนกลับกับวิธีการคำนวณแบบไปข้างหน้า วิธีทำก็คือเราทราบว่าเมตริกในการเปลี่ยนระนาบจากปลายมือมาที่ฐานนั้นเกิดจากการเปลี่ยนระนาบต่อกันมาเป็นทอดๆ

$${}^0T_5 = {}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4 * {}^4A_5$$

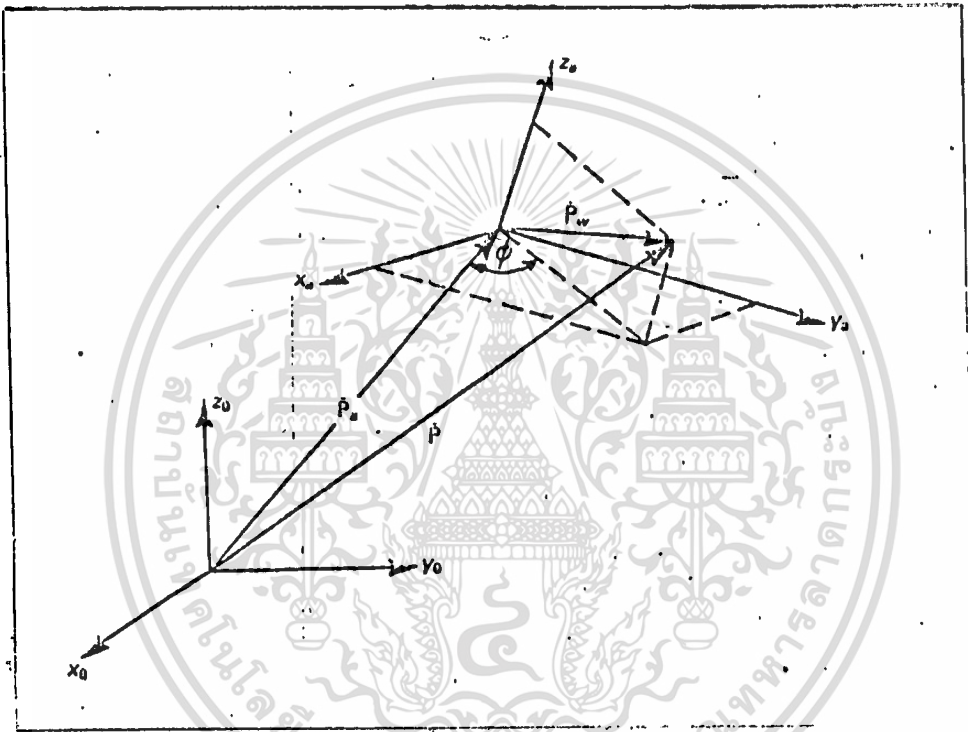
และแต่ละขั้นตอนจะเกิดจากมุม θ_i หรือ α_i เท่านั้น เมื่อเราเอาเมตริกอินเวอร์สของ ${}^{-1}A_1$ คูณทั้งสองด้าน แล้วนำส่วนประกอบในเมตริกที่ได้มาใหม่มาตั้งสมการแล้วแก้สมการก็จะสามารถหาค่า θ_i ออกมาได้

$${}^0A_1^{-1} * {}^0T_5 = {}^0A_1^{-1} * {}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4 * {}^4A_5 = {}^1T_5$$

หลังจากนั้นก็นำเมตริกอินเวอร์สต่างๆเข้าไปคูณก็จะได้ค่ามุมต่างๆออกมา

วิธี ทางเรขาคณิต

เป็นวิธีที่จะสามารถใช้กับแกนกลที่มีมือหนีบวางตัวอยู่ในระนาบเดียวกันกับ
ช่วงแขน ต่างๆซึ่งแกนกลที่ใช้ในการวิจัยก็จัดอยู่ในกลุ่มนี้ด้วย ไม่ว่าจะฐานจะหมุนไปเป็นมุม
เท่าใด มือหนีบก็จะยังคงอยู่ในระนาบเดียวกับช่วงแขน ดังนั้นเราจึงสามารถที่จะมองให้
เป็นสองมิติได้เสมอ โดยแยกการหมุนของฐานออกไปได้



รูป 2.6 การเลื่อนระนาบ

จากรูปจะได้ว่า

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} P_{ax} \\ P_{ay} \\ P_{az} \end{bmatrix} + \begin{bmatrix} P_{wx} \\ P_{wy} \\ P_{wz} \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\phi = \tan^{-1}(A_y / A_x)$$

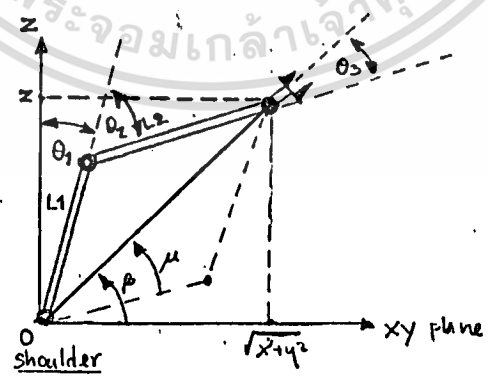
$$\theta = \tan^{-1}(\sqrt{(A_x^2 + A_y^2)} / A_z)$$

$$P_{wx} = |P_w| * \sin\theta * \cos\phi$$

$$P_{wy} = |P_w| * \sin\theta * \sin\phi$$

$$P_{wz} = |P_w| * \cos\theta$$

หลังจากนั้นจะพิจารณาเฉพาะค่า P_w ที่อยู่ในระนาบ Z-XY plane



รูป 2.7 ช่วงแขนในสองมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีนี้จุดหมุนที่หัวไหล่มีการเลื่อน (Translation) ขึ้นมาจากฐานฐาน ค่าความสูงต่างๆ ก็ต้องเลื่อนให้มาอยู่ในการพิจารณาที่ระนาบใหม่ด้วย หลังจากนั้นก็จะใช้กฎของ sin และ cos มาหามุมต่างได้

เพื่อความสะดวก กำหนดระยะบนระนาบ XY ให้เป็น A

$$A = \sqrt{(x^2 + y^2)}$$

จะได้ว่า

$$|P_{\bullet}|^2 = L1^2 + L2^2 + 2 * L1 * L2 * \cos \theta_2$$

$$A^2 + Z^2 = L1^2 + L2^2 + 2 * L1 * L2 * \cos \theta_2$$

$$\theta_2 = \cos^{-1} \left((A + Z - L1 - L2) / (2 * L1 * L2) \right)$$

$$\mu = \cos^{-1} \left((|P_{\bullet}| + L1^2 - L2^2) / (2 * |P_{\bullet}| * L1) \right)$$

$$\beta = \tan^{-1} (Z/A)$$

นั่นคือ

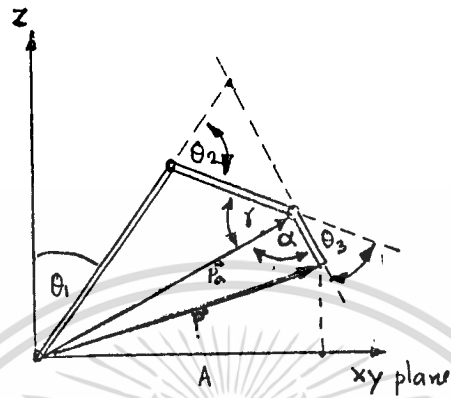
$$\theta_1 = \pi/2 - (\beta + \mu)$$

$$\delta = \cos^{-1} \left((|P_{\bullet}|^2 + L2^2 - L1^2) / (2 * L2 * |P_{\bullet}|) \right)$$

$$\alpha = \cos^{-1} \left((|P|^2 + L3^2 - |P_{\bullet}|^2) / (2 * L3 * |P|) \right)$$

$$\theta_2 = \pi - [\delta + \alpha]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.8 แขนกลในระนาบสองมิติ

- ϕ = มุมที่ฐานหมุน
- θ_1 = มุมต้นแขนทำกับฐาน
- θ_2 = มุมแขนส่วนบนทำกับแขนส่วนล่าง
- θ_3 = มุมแขนทำกับข้อมือ

ข้อเปรียบเทียบของวิธีคำนวณแบบต่างๆ แสดงได้ดังตารางที่ 2.2

ชนิดของการคำนวณ	ข้อดี	ข้อเสีย
1. การคำนวณแบบไปข้างหน้า	สามารถนำไปใช้กับ แขนงกลทุกชนิดได้ ความเร็วในการคำนวณ สูง	ต้องบอกค่ามุมต่างๆ ในขณะนั้นให้แขนงกลก่อน จึงจะรู้ว่าขณะนั้นปลาย แขนงอยู่ที่ตำแหน่งไหน วิธีนี้จึงเหมาะสำหรับ เส้นทางที่กำหนดไว้ล่วงหน้า และออกแบบการ ควบคุมได้ยาก
2. การคำนวณแบบย้อนกลับ	สามารถนำไปใช้กับ การควบคุมการเคลื่อนที่ ของแขนงได้ทุกชนิด และง่ายต่อการใช้งาน คือ บอกเพียงจุดที่ต้องการ จะไปเท่านั้น	ออกแบบการควบคุมการ เคลื่อนที่ได้ยาก เนื่องจาก จะต้องใช้เวลาในการคํ านวนมาก ต้องใช้กับเครื่อง คอมพิวเตอร์ที่มีความสามารถ สูง จึงจะได้การเคลื่อน ที่ที่เร็วเพียงพอต่อการใ้ งาน
3. การคำนวณโดยใช้วิธีทาง เรขาคณิต	ออกแบบการควบคุม การเคลื่อนที่ได้ง่าย และง่ายต่อการใช้งาน บอกเพียงจุดที่ต้องการ จะไปเท่านั้น ความเร็วในการคำนวณสูง	นำไปใช้กับการควบคุม การเคลื่อนที่ของแขนง ชนิดระนาบเดียวเท่านั้น

ตารางที่ 2.2 ตารางเปรียบเทียบข้อดีข้อเสียของการคำนวณแต่ละแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบกลที่ใช้ทดลองเป็นแบบระนาบเดี่ยว ดังนั้นจึงสามารถใช้วิธีการควบคุมแบบเรขาคณิตได้ โดยจะใช้เครื่องคอมพิวเตอร์ส่วนบุคคลเป็นหน่วยประมวลผลหาค่ามุมที่ช่วงแบนแต่ละช่วงกระทำต่อกันและส่งสัญญาณไปกระตุ้นให้สเตปปีงมอเตอร์ของแต่ละข้อต่อทำงาน ส่วนของโปรแกรมการควบคุมต่างๆ จะได้อธิบายในบทที่ 3 ต่อไป



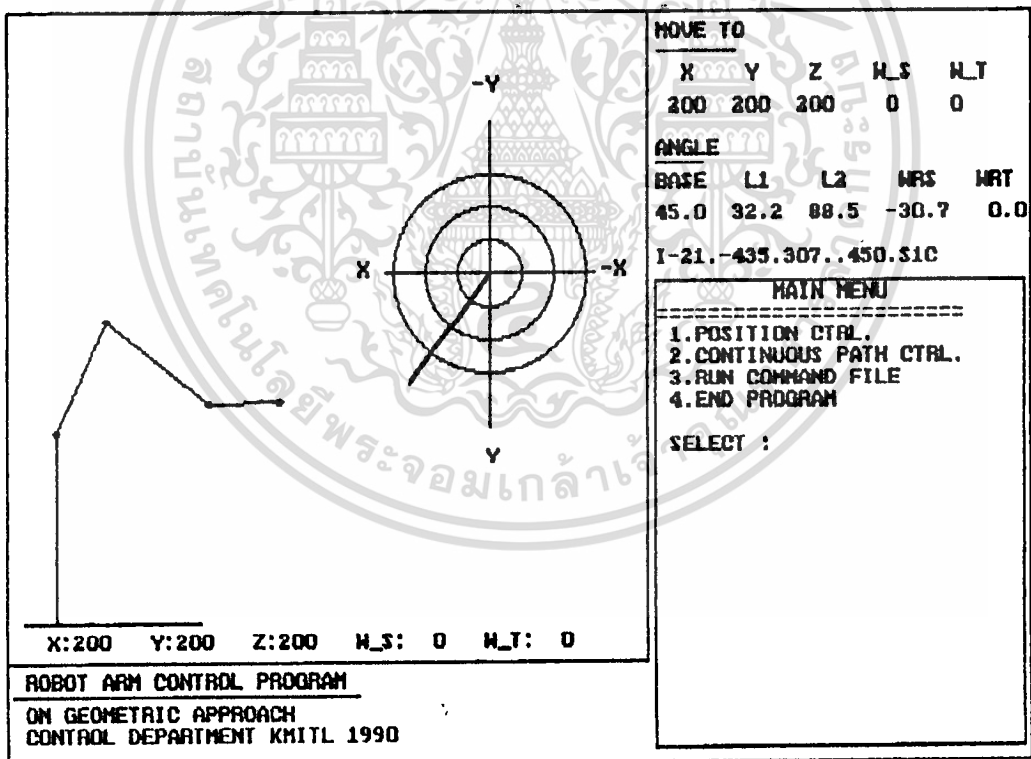
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

จากบทที่ 2 อธิบายถึงวิธีการคำนวณหาค่ามุมที่แขนแต่ละช่วงกระทำต่อกันซึ่งมีวิธีการคำนวณ 3 วิธี แต่วิธีที่ใช้ในปริศยานี้พจน์นี้คือ วิธีทางเรขาคณิตการคำนวณจะใช้เครื่องคอมพิวเตอร์ เป็นตัวประมวลผลดังนั้นจึงต้องสร้างโปรแกรมสำเร็จรูปขึ้นมาเพื่อใช้ทดลองควบคุม โดยหลักการคำนวณหาค่ามุมต่างๆนั้นจะได้มาจากบทที่ 2

เมื่อเรียกใช้โปรแกรมควบคุมแขนกล จอภาพจะแสดงสถานะต่างๆของแขนกลในขณะนั้น ดังรูปที่ 3.1



รูป 3.1 สถานะของแขนกลบนจอภาพคอมพิวเตอร์

ตำแหน่งต่างๆของจอภาพจะแสดงค่าสถานะดังรูปที่ 3.2

<p>แสดงภาพแขนกล ด้านข้าง(Side View) และด้านบน (Top View)</p>	<p>ตำแหน่งที่ต้องการไป มุมของแขนกลปัจจุบัน คำสั่งที่ส่งไปให้แขนกล</p>
<p>แสดงตำแหน่งปัจจุบันของ ปลายแขนกล</p>	<p>ส่วนโต้ตอบกับ ผู้ใช้โปรแกรม</p>

รูป 3.2 ความหมายของส่วนต่างๆของภาพบนจอคอมพิวเตอร์

การรับคำสั่งเพื่อไปควบคุมแขนกลสามารถทำได้สองวิธีคือ

1. รับคำสั่งโดยตรงจากแป้นพิมพ์

2. รับคำสั่งมาจากแฟ้มข้อมูลที่บรรจุชุดคำสั่งต่างๆไว้ซึ่งวิธีการนี้จะอำนวยความสะดวกในการพัฒนาแขนกลต่อไป โดยสามารถเขียนคำสั่งการทำงานของแขนกลไว้เป็นขั้นๆสำหรับงานแต่ละอย่าง และนำไปเก็บไว้ในแฟ้มข้อมูลซึ่งสามารถเรียกออกมาใช้ได้ทันทีที่ต้องการ

คำสั่งต่างๆเหล่านี้จะถูกเก็บไว้ในแฟ้มข้อมูลที่มีนามสกุลว่า RCF ย่อมาจาก Robot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command File ดังตัวอย่างในรูปที่ 3.3

SCRIBE.RCF

```

home
open
posi 200 200 200 250 200
close
cont -200 100 250 90 130
open
delay 1000
home
end

```

รูปที่ 3.3 โปรแกรมควบคุมแขนกล

คำสั่งต่างๆที่สามารถนำไปใช้สั่งงานแขนกลมีดังนี้

home - เป็นคำสั่งให้แขนกลเคลื่อนที่กลับสู่จุดเริ่มต้น (Home position)

open - เปิดมือจับ

close - ปิดมือจับ

posi - เคลื่อนที่แบบควบคุมตำแหน่งไปยังจุดที่กำหนด

มีรูปแบบการใช้งานเป็นดังนี้

```
posi <X> <Y> <Z> <Wrist swivel> <Wrist twist>
```

cont - เคลื่อนที่แบบควบคุมเส้นทางไปยังจุดที่กำหนด

มีรูปแบบการใช้งานดังนี้

```
cont <X> <Y> <Z> <Wrist swivel> <Wrist twist>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

delay - หน่วงเวลา มีรูปแบบการใช้งานดังนี้

delay <milli second>

swivel - หมุนข้อมือไปสู่มุมที่กำหนด มีรูปแบบการใช้งานดังนี้

swivel <degree>

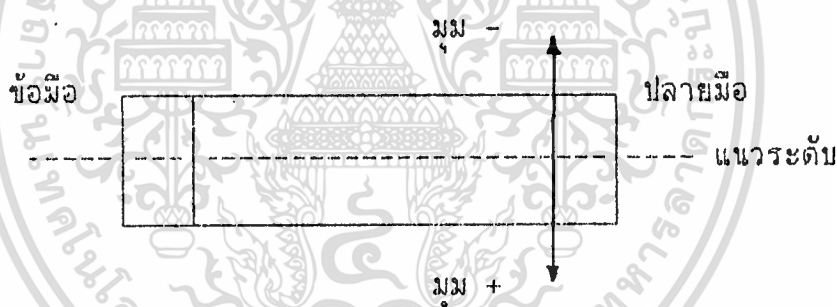
twist - ปรับมุมข้อมือที่กำกับแนวราบให้เป็นไปตามที่กำหนด

มีรูปแบบการใช้งานดังนี้

twist <degree>

end - คำสั่งสิ้นสุดการทำงาน

หมายเหตุ: มุม Wrist angle คือมุมที่ทำกับแนวระดับดังแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 แสดงมุมปลายแขน (Wrist angle)

โปรแกรมการทำงาน

ในโปรแกรมที่สร้างขึ้นมานั้นจะประกอบด้วยโปรแกรมย่อย 3 ส่วน คือ

1. โปรแกรมย่อย POSITION
2. โปรแกรมย่อย CONTINUOUS
3. โปรแกรมย่อย COMMAND FILE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในลํวณโปรแกรมย่อยทั้ง 3 จะเรียก โปรแกรมย่อยซึ่งเป็นจุดสำคัญ 2 โปรแกรมย่อย คือ

1. GEO_CAL

2. AMUST_DRV

ดังนั้นจะขออธิบาย การทำงานของโปรแกรมย่อยทั้ง 2 ก่อน

โปรแกรมย่อย GEO_CAL

โปรแกรม GEO_CAL เป็นโปรแกรมที่จะทำหน้าที่คำนวณหาค่ามุมที่ LINK แต่ละ LINK กระทบต่อกัน โดยจะมีลักษณะการทำงานตามรูป 3.5



รูปที่ 3.5 ผังการทำงานของโปรแกรมย่อย GEO_CAL

ข้อมูลเข้า

X : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน X

Y : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน Y

Z : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน Z

WT : มุมที่มีอิทธิพลกระทํอแนวระดับ

ข้อมูลออก

ANGLE (O) : มุมที่แขนที่ 1 กระทบกับแนวแกน Z

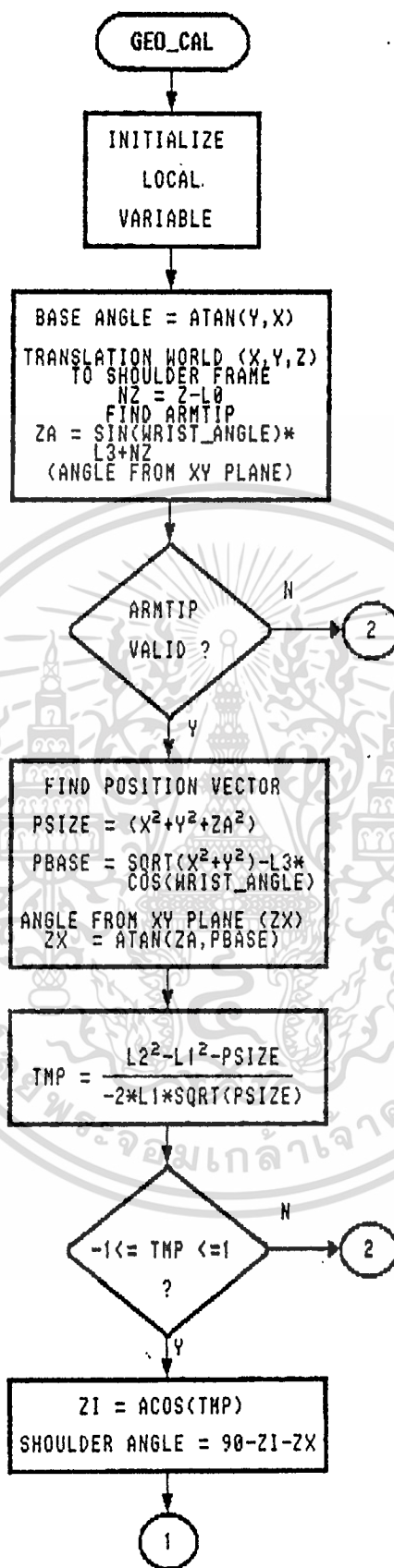
- ANGLE (1) : มุมที่แขนที่ 2 กระทบกับแนวแกนแขนที่ 1
 ANGLE (2) : มุมที่มือหนีบกระทำต่อแนวระดับ
 ANGLE (3) : มุมที่ต้องการให้มือหนีบหมุน
 ANGLE (4) : มุมที่ฐาน
 ANGLE (5) : ค่าความเร็วที่ใช้
 ANGLE (6) : ค่าที่ใช้ประมวลผลการ บิด-เปิด ของมือหนีบ

รายละเอียดการทำงานแสดงให้เห็นจากผังลำดับการทำงาน (Flow chart)

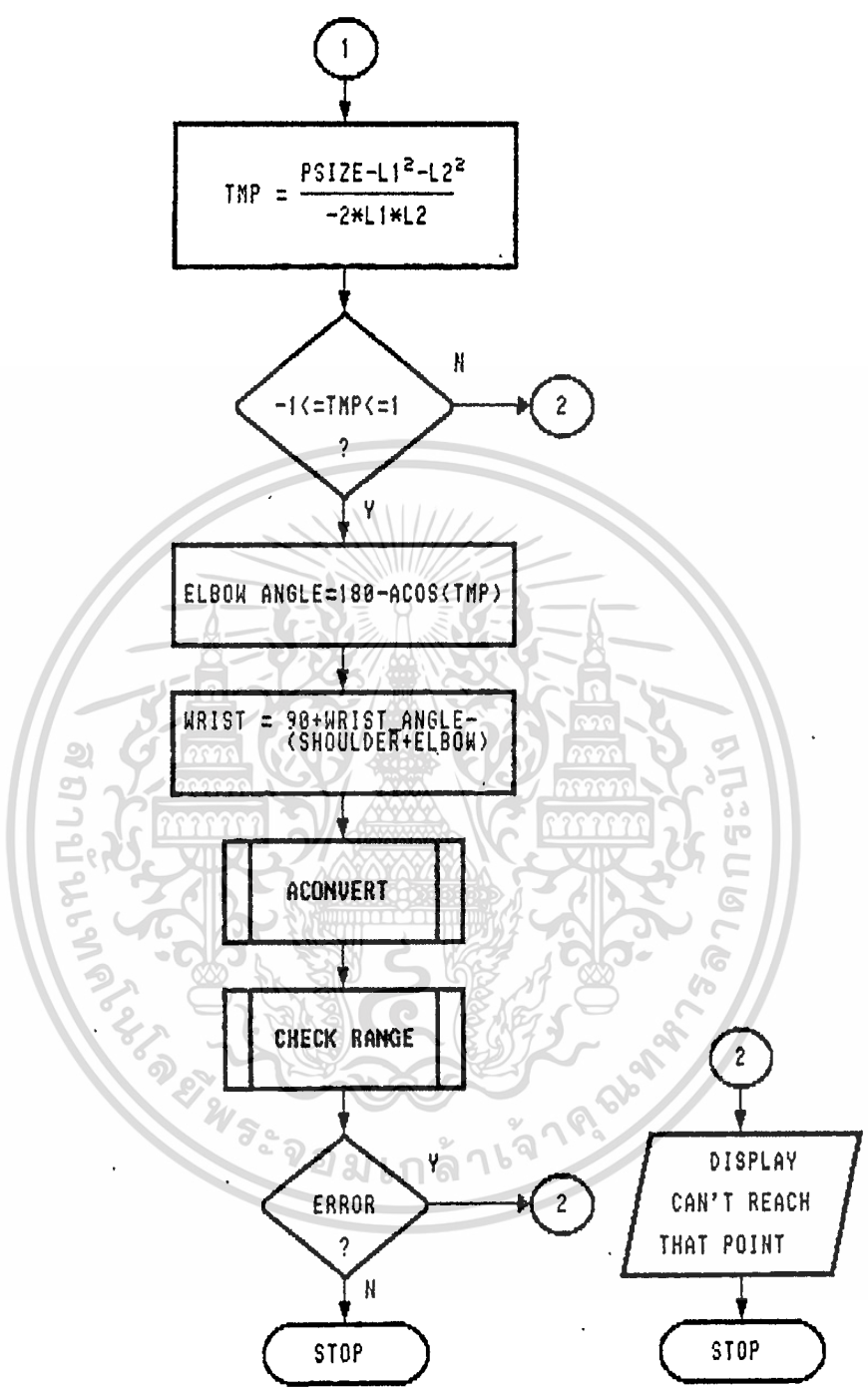
GEO_CAL ดังรูปที่ 3.6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

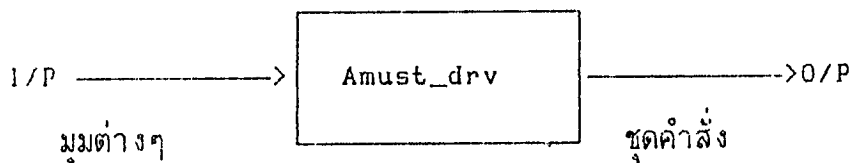


รูปที่ 3.6 ฟังก์ชันตอนการทำงานของโปรแกรมย่อย GEO_CAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมย่อย Amust_drv

โปรแกรมย่อยส่วนนี้จะทำหน้าที่แปลงค่ามุมต่างๆที่ได้จากการคำนวณออกมาเป็นชุดคำสั่งของแขนกลและส่งออกทางพอร์ตเครื่องพิมพ์ มีลักษณะการทำงานดังรูป 3.7



รูป 3.7 ผังการทำงานของโปรแกรมย่อย Amust_drv

ข้อมูลเข้า

Angle(0)..Angle(6) : มุมต่างๆของแขนกลที่ได้จากการคำนวณ

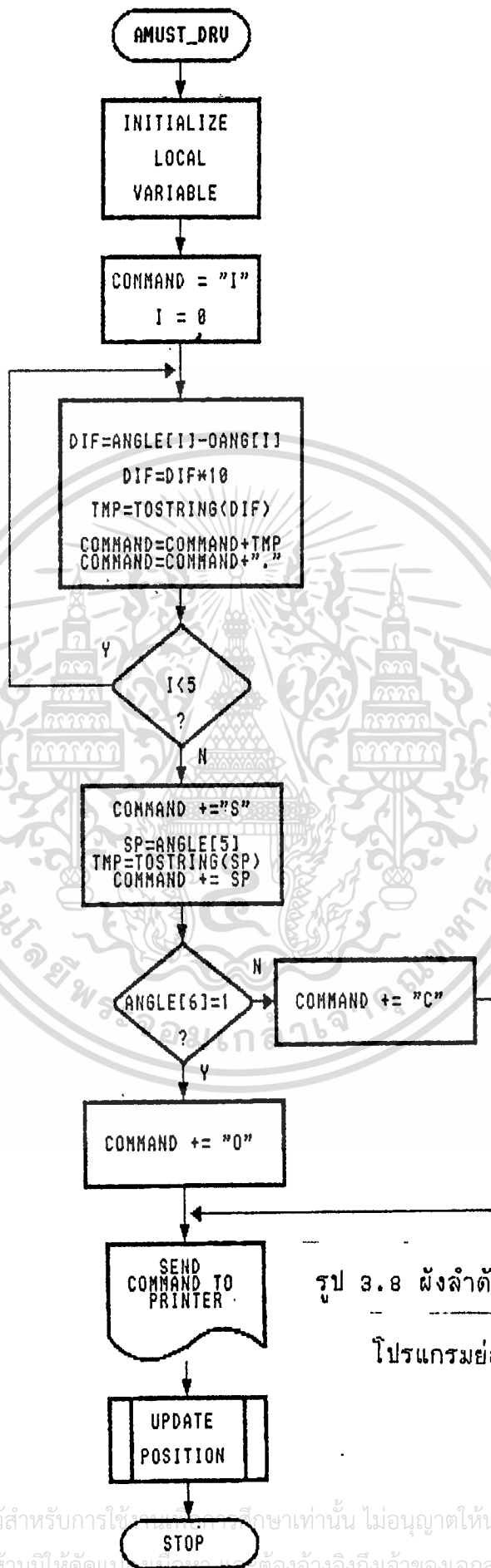
ข้อมูลออก

คำสั่งในรูปแบบ

"I<n1>.<n2>.<n3>.<n4>.<n5>.S<n6><n7>"

- n1: จำนวนลูกสัญญาณที่ต้องใช้ในการหมุนไหล่ให้ถึงจุดที่กำหนด
- n2: จำนวนลูกสัญญาณที่ต้องใช้ในการหมุน link ที่สองให้ถึงจุดที่กำหนด
- n3: จำนวนลูกสัญญาณที่ต้องใช้ในการหมุนระนาบข้อมือให้ถึงจุดที่กำหนด.
- n4: จำนวนลูกสัญญาณที่ต้องใช้ในการหมุนข้อมือให้ถึงมุมที่กำหนด
- n5: จำนวนลูกสัญญาณที่ต้องใช้ในการหมุนฐานให้แขนกลอยู่ในแนวระนาบ
- n6: ระดับความเร็วในการเคลื่อนที่(ตั้งแต่ระดับ 1- 5)
- n7: สถานะของมือหนีบ(ปิด - เปิด)

รายละเอียดการทำงานแสดงให้เห็นจากผังลำดับการทำงาน AMUST_DRV ดังรูป 3.8



รูป 3.8 ผังลำดับการทำงานของ
โปรแกรมย่อย Amust_drv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในมหาวิทยาลัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแบบหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมย่อย POSITION

โปรแกรม POSITION จะทำหน้าที่ควบคุมการเคลื่อนที่ให้เป็นไปตามการควบคุมตำแหน่ง โดยจะดึงโปรแกรมย่อย GEO_CAL , และ AMUST_DRV มาใช้งาน มีลักษณะการทำงานดังรูป 3.9



รูป 3.9 ผังการทำงานของโปรแกรมย่อย POSITION

ข้อมูลเข้า

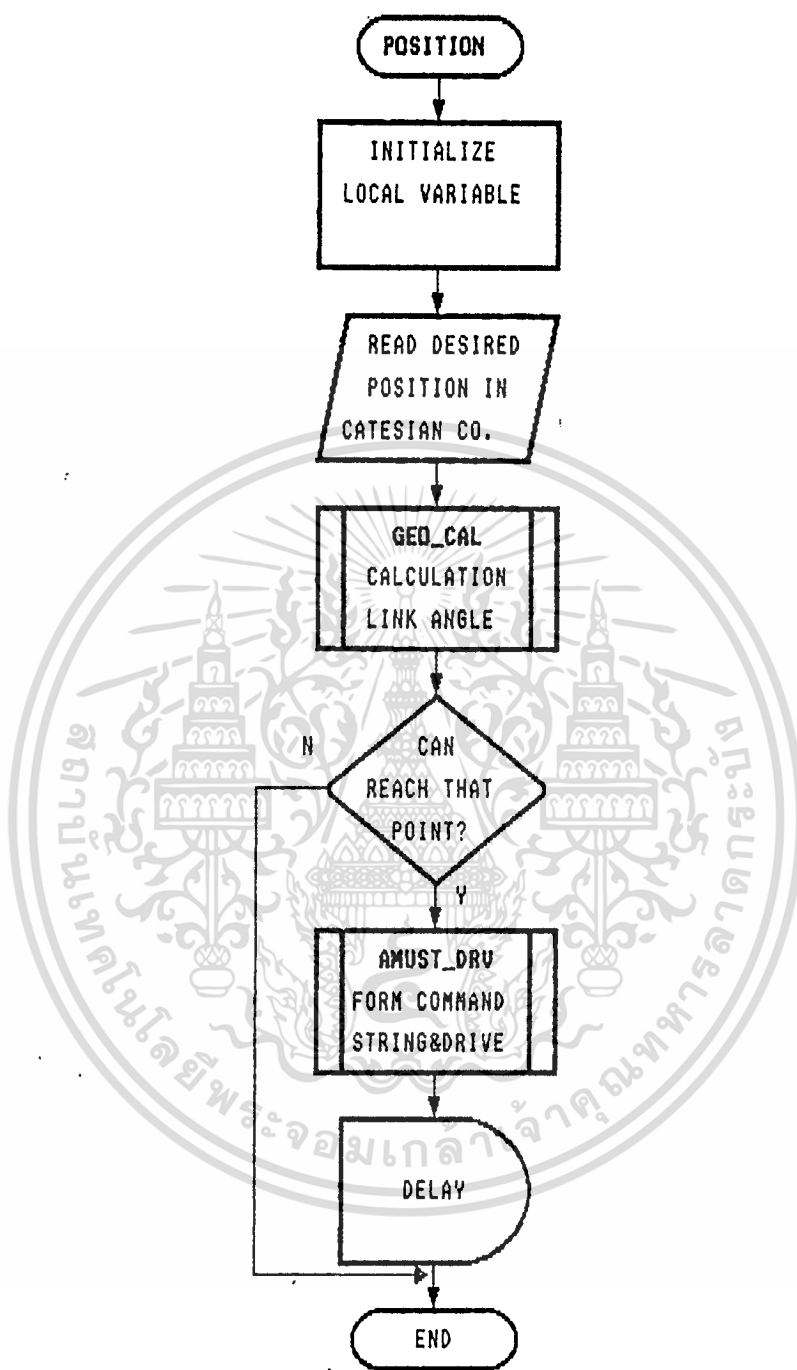
- X : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน X
 Y : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน Y
 Z : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน Z
 WT : มุมที่มีมือหนีบกระทำต่อแนวแกนระนาบ
 WS : มุมในการหมุนมือหนีบ

ข้อมูลออก

ชุดคำสั่งประกอบด้วย

- I : สั่งให้แขนกลเคลื่อนที่
 o, c : ปิด-เปิด มือหนีบ

รายละเอียดการทำงานแสดงให้เห็นจากผังลำดับการทำงาน POSITION ดังรูป 3.10

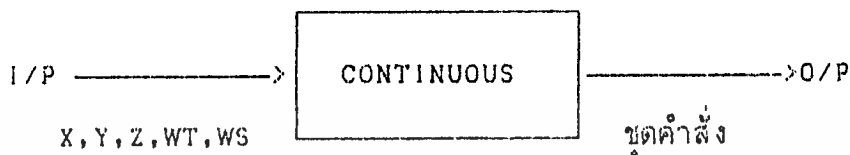


รูป 3.10 แผนผังการทำงานของโปรแกรมย่อย POSITION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมย่อย CONTINUOUS

เป็นโปรแกรมย่อยที่ทำหน้าที่ ควบคุมการเคลื่อนที่ให้เป็นแบบการควบคุมเส้นทาง โดยจะดึงโปรแกรมย่อย GEO_CAL และ AMUST_DRV มาทำงาน และมีลักษณะการทำงานดังรูป 3.11



รูป 3.11 ผังการทำงานของโปรแกรมย่อย CONTINUOUS

ข้อมูลเข้า

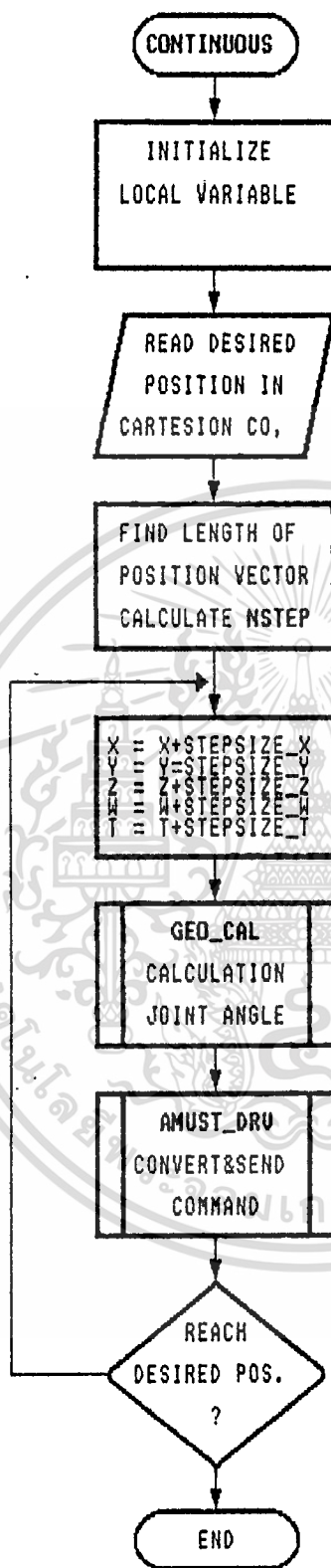
- X : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน X
 Y : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน Y
 Z : ตำแหน่งของจุดที่ต้องการเคลื่อนที่ไปในแนวแกน Z
 WT : มุมที่มือหีบกระทำต่อแนวแกนระนาบ
 WS : มุมในการหมุนมือหีบ

ข้อมูลออก

ชุดคำสั่งประกอบด้วย

- I : สั่งให้แขนกลเคลื่อนที่
 O, C : ปิด-เปิด มือหีบ

รายละเอียดการทำงานแสดงให้เห็นจากผังลำดับการทำงาน CONTINUOUS ดังรูป 3.12



รูป 3.12 พังลำดับการทำงานของโปรแกรมย่อย CONTINUOUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมย่อย RUN COMMAND FILE

โปรแกรมย่อย RUN COMMAND FILE จะประกอบด้วยโปรแกรมย่อย 2 โปรแกรม คือ

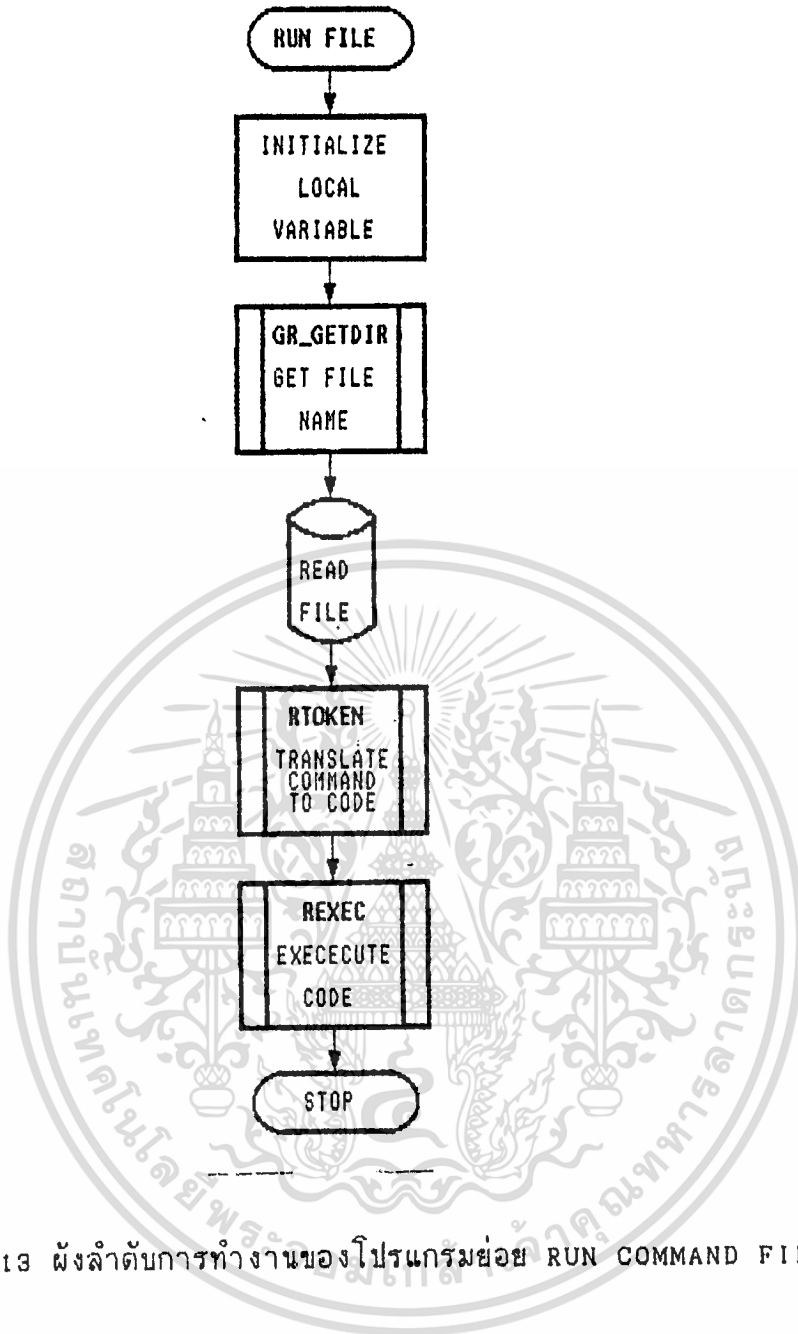
1. โปรแกรม RTOKEN จะทำหน้าที่รับชุดคำสั่งที่เขียนในแฟ้มข้อมูลมาเก็บในตัวแปร TOKEN

2. โปรแกรมย่อย REXE จะทำหน้าที่นำค่าตัวแปร TOKEN ไปประมวลผลเป็นชุดคำสั่งส่งออกทางพอร์ต (PORT) เครื่องพิมพ์

โปรแกรมย่อย RUN COMMAND FILE ทำหน้าที่ประมวลคำสั่งที่เขียนในแฟ้มคำสั่ง (COMMAND FILE) แล้วเปลี่ยนเป็นชุดคำสั่งส่งออกทางพอร์ตเครื่องพิมพ์ เพื่อควบคุมการเคลื่อนที่ของแขนกล ซึ่งคำสั่งที่เขียนในแฟ้มคำสั่งจะประกอบด้วยคำสั่งต่อไปนี้

- "HOME" : กลับสู่ตำแหน่งอ้างอิง
- "OPEN" : เปิดมือหนีบ
- "CLOSE" : ปิดมือหนีบ
- "POS 1" : ควบคุมแบบควบคุมตำแหน่ง
- "CONT" : ควบคุมแบบควบคุมเส้นทาง
- "DELAY" : หน่วยเวลา
- "SWIVEL" : สั่งให้มือหนีบหมุน
- "DIGIT" : ค่าตัวเลขที่จะใช้ประมวลผล
- "TWIST" : มุมที่มือหนีบกระทำต่อแนวแกนของแขนช่วงที่ 2
- "END" : จบแฟ้มคำสั่ง

การทำงานของโปรแกรมสามารถดูได้จากผังลำดับการทำงาน ในรูป 3.13



รูป 3, 13 ผังลำดับการทำงานของโปรแกรมย่อย RUN COMMAND FILE

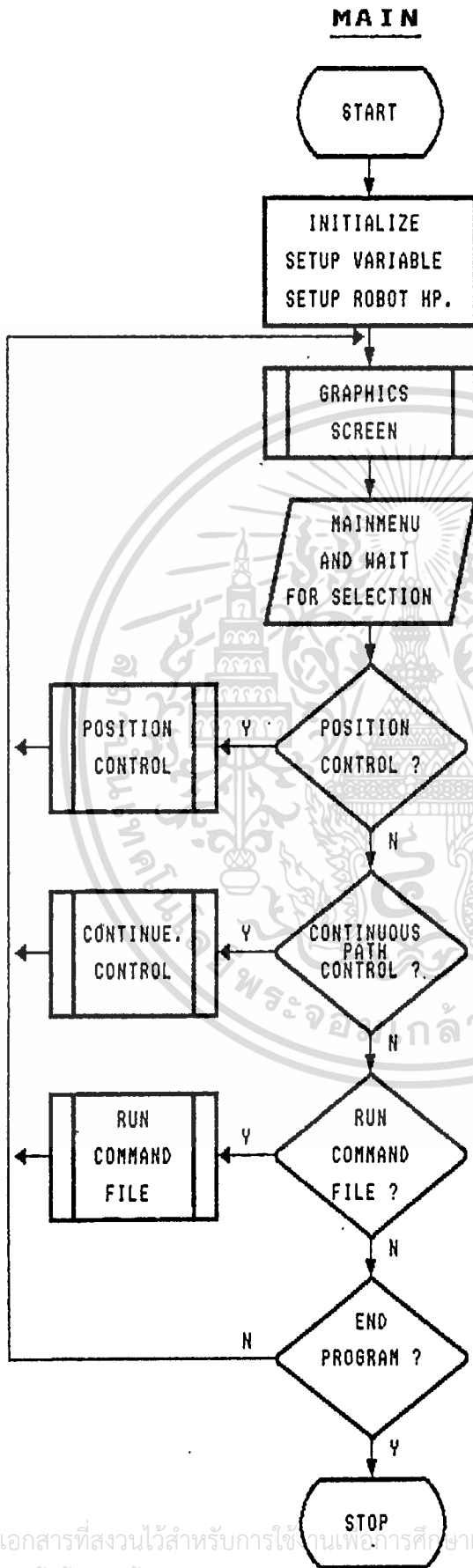
โปรแกรมหลัก MAIN

จะเป็นโปรแกรมส่วนที่เรียกโปรแกรมย่อยอื่นๆขึ้นมาทำงานตามแต่ผู้ใช้จะสั่งงาน
เข้าไปทางแป้นพิมพ์ และจะวนรอบรอรับคำสั่งจนกว่าจะมีคำสั่งให้จบการทำงาน โดยมี
ขั้นตอนการทำงานดังรูป 3.14

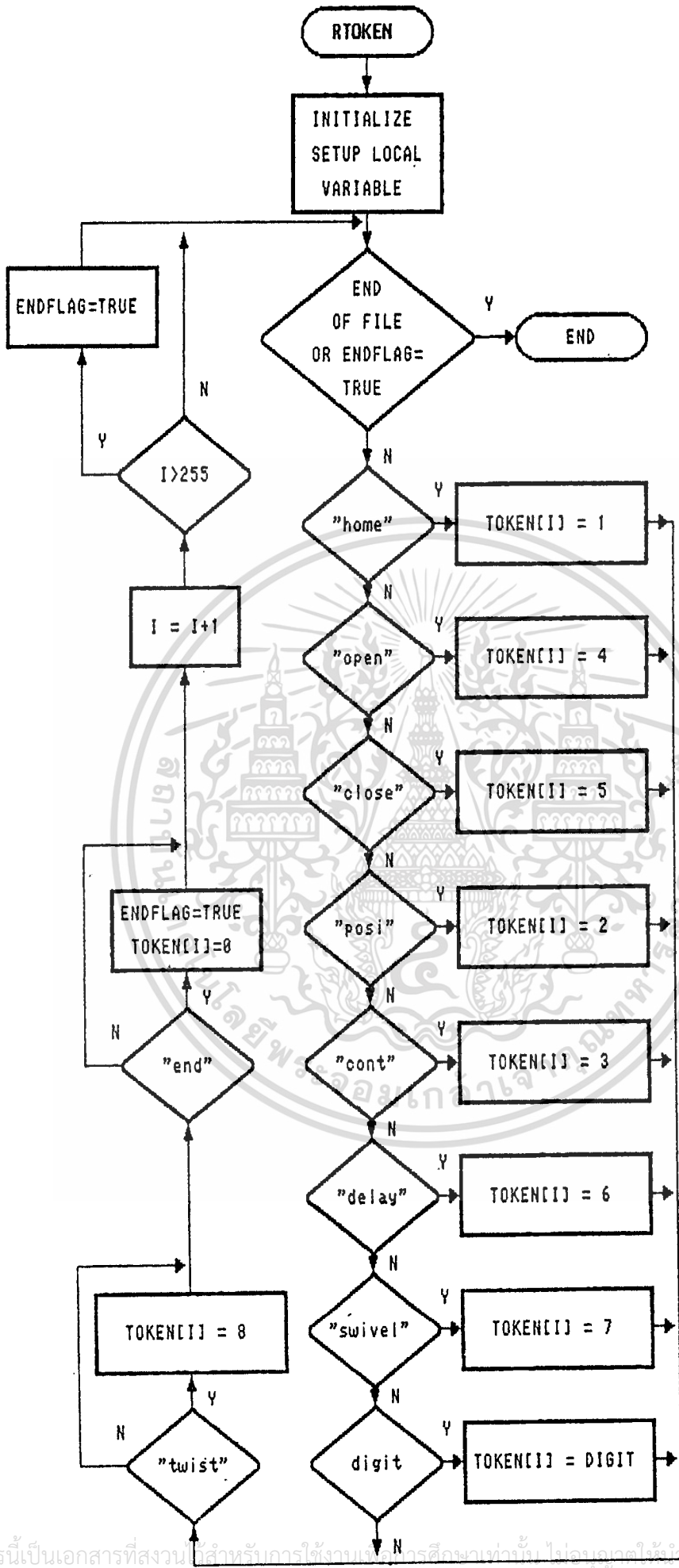
โปรแกรมการทำงานทั้งหมดมีอยู่ในภาคผนวก B ท้ายปริญญาโทนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

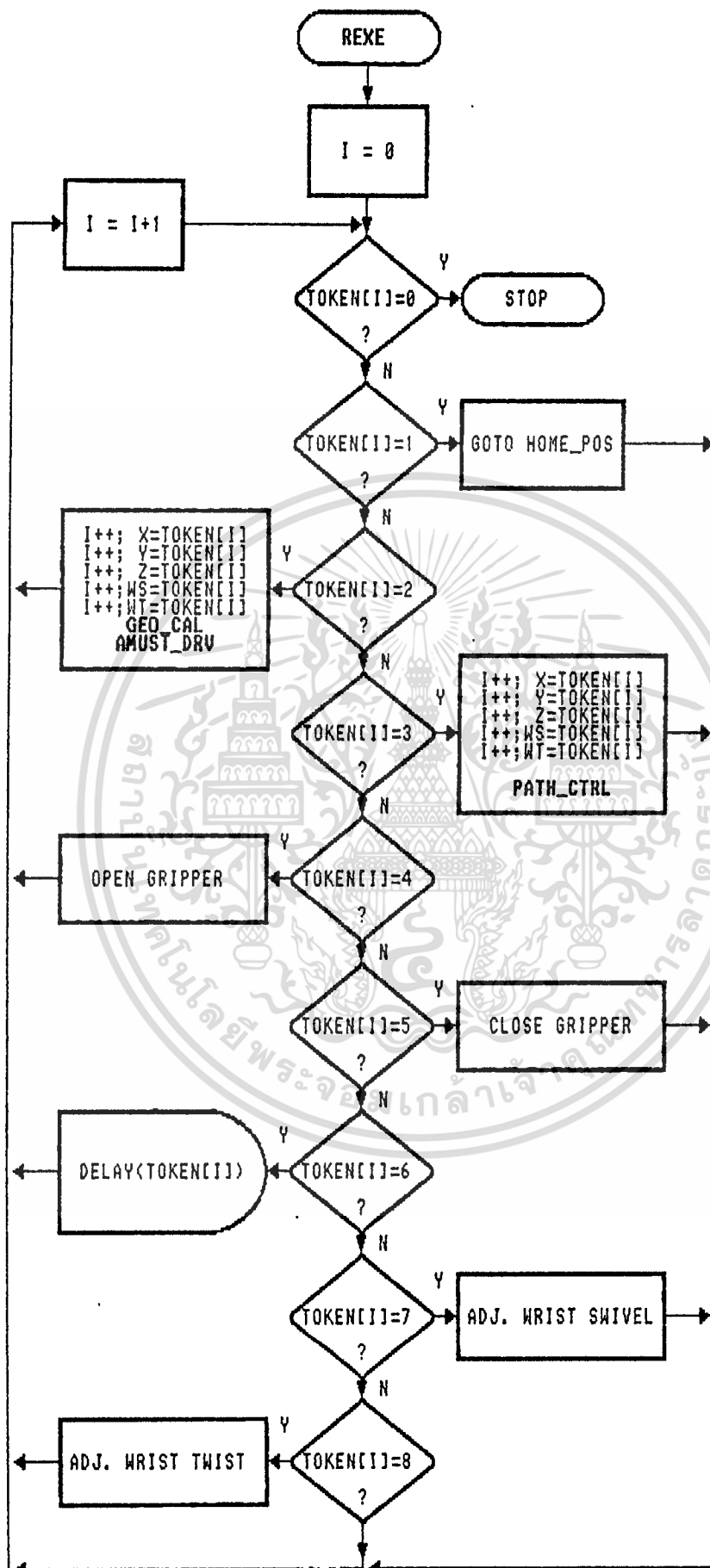


รูปที่ 3.14 ผังลำดับการทำงานของ
ตัวโปรแกรมหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขหรือดัดแปลงเนื้อหา

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 3.15 ฟังก์ชันการทำงานของโปรแกรมย่อย RTOKEN



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ รูปที่ 3.16 ผังลำดับการทำงานของโปรแกรมย่อย REXE ทุกครั้งที่มีการนำไปใช้

บทที่ 4

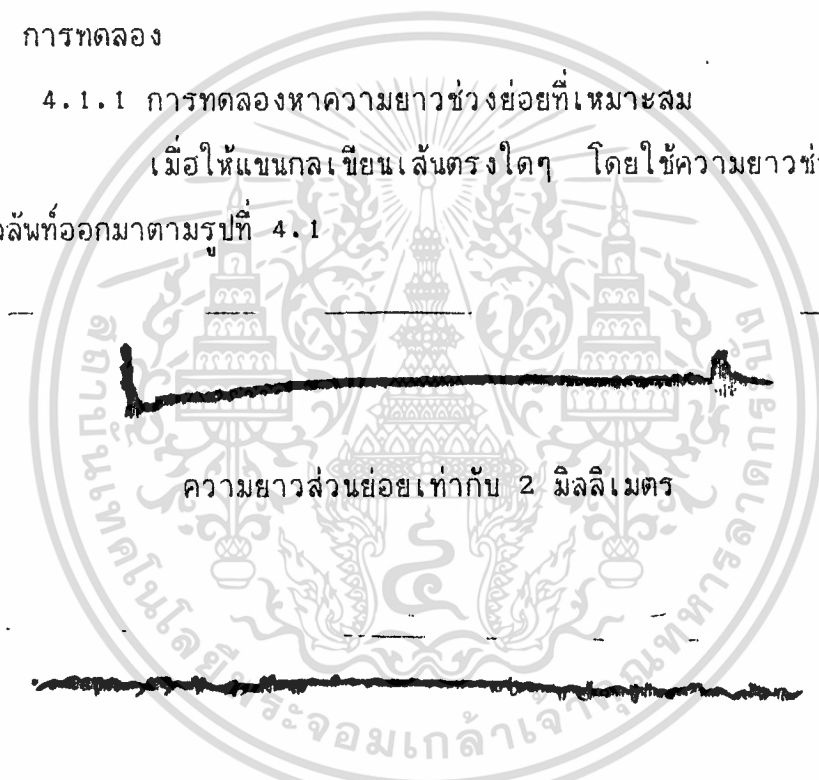
การทดลองและผลการทดลอง

การเคลื่อนที่แบบเส้นตรงต่อเนื่องนั้นมีหลักการทั่วไปคือ จะแบ่งเส้นทางระหว่างจุดเริ่มต้นการเคลื่อนที่และจุดสุดท้ายออกเป็นส่วนย่อยๆที่มีระยะทางสั้นๆเท่าๆกัน และทำการเคลื่อนที่ไปที่ละส่วนย่อยจนกว่าจะถึงจุดปลายทาง ยิ่งแบ่งออกเป็นส่วนย่อยมากเท่าใดก็จะได้เส้นทางที่เรียบมากขึ้นเท่านั้น

4.1 การทดลอง

4.1.1 การทดลองหาความยาวช่วงย่อยที่เหมาะสม

เมื่อให้แขนกลเขียนเส้นตรงใดๆ โดยใช้ความยาวช่วงย่อยขนาดต่างๆได้ผลลัพท์ออกมาตามรูปที่ 4.1



ความยาวส่วนย่อยเท่ากับ 5 มิลลิเมตร



ความยาวส่วนย่อยเท่ากับ 8 มิลลิเมตร.

รูปที่ 4.1 เส้นตรงวาดโดยใช้ความยาวส่วนย่อยแตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.1 จะเห็นว่าเมื่อใช้ความยาวช่วงย่อยน้อยๆจะให้เส้นตรงที่มีความเรียบมากขึ้น และการกระตุกของแขนกลในช่วงออกตัว และช่วงที่แขนกลพยายามจะหยุดนิ่ง จะน้อยกว่าเมื่อใช้ช่วงย่อยที่มีค่ามากๆ

เราต้องการจะหาค่าช่วงย่อยที่เหมาะสมที่สุดสำหรับแขนกลชุดนี้ โดยให้แขนกลวาดเส้นตรงแนวราบยาว 10 ซม. โดยใช้ค่าช่วงย่อยต่างๆกัน ได้ค่าความผิดพลาดของความยาวเส้นตรงที่วาดได้ดังตาราง 4.1

ความยาวช่วงย่อย มม.	ค่าความผิดพลาด %
2	12.50
3	8.44
4	8.05
5	6.25
6	6.67
7	7.14
8	7.54
9	10.72
10	12.31

ตาราง 4.1 ตารางแสดงค่าผิดพลาดทางตำแหน่งแนวราบเมื่อใช้ช่วงย่อยต่างๆกัน

และทดลอง เช่นเดียวกันเมื่อวาดเส้นตรงแนวตั้งได้ผลดังตารางที่ 4.2

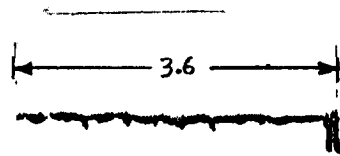
ความยาวช่วงย่อย มม.	ค่าความผิดพลาด %
2	9.56
3	9.22
4	7.35
5	6.57
6	6.83
7	7.04
8	7.23
9	8.42
10	9.46

ตาราง 4.2 ตารางแสดงค่าผิดพลาดทางตำแหน่งแนวตั้งเมื่อใช้ช่วงย่อยต่างๆกัน

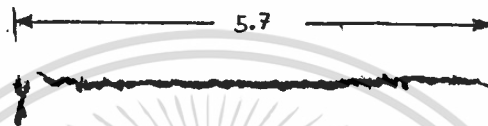
จะเห็นว่าค่าความผิดพลาดจะเกิดขึ้นน้อยที่สุดเมื่อใช้ช่วงย่อย 5 มม. ทั้งในแนวราบและแนวตั้ง ดังนั้นในการทดลองขั้นต่อไปจะใช้ค่าความยาวช่วงย่อย 5 มม. เป็นค่าทดสอบ เนื่องจากจะให้ค่าความผิดพลาดทางตำแหน่งแนวราบต่ำที่สุดและได้เส้นทางการเคลื่อนที่ที่เรียบกว่าค่าอื่นๆ

4.1.2 การทดลองหาค่าความผิดพลาดทางตำแหน่งในแนวราบ

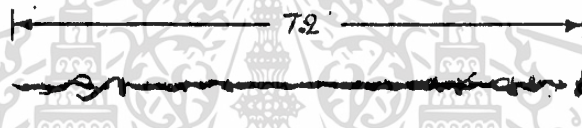
ให้แกนกลางวาดเส้นตรงแนวราบความยาวแตกต่างกันและให้ความยาวส่วนย่อย 5 มิลลิเมตร ได้ผลดังรูปที่ 4.2



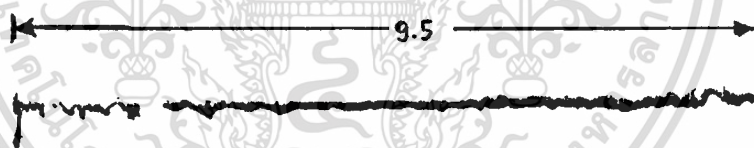
เส้นตรงยาว 4 ซม.



เส้นตรงยาว 6 ซม.



เส้นตรงยาว 8 ซม.



เส้นตรงยาว 10 ซม.

รูปที่ 4.2 เส้นตรงแนวราบความยาวแตกต่างกัน

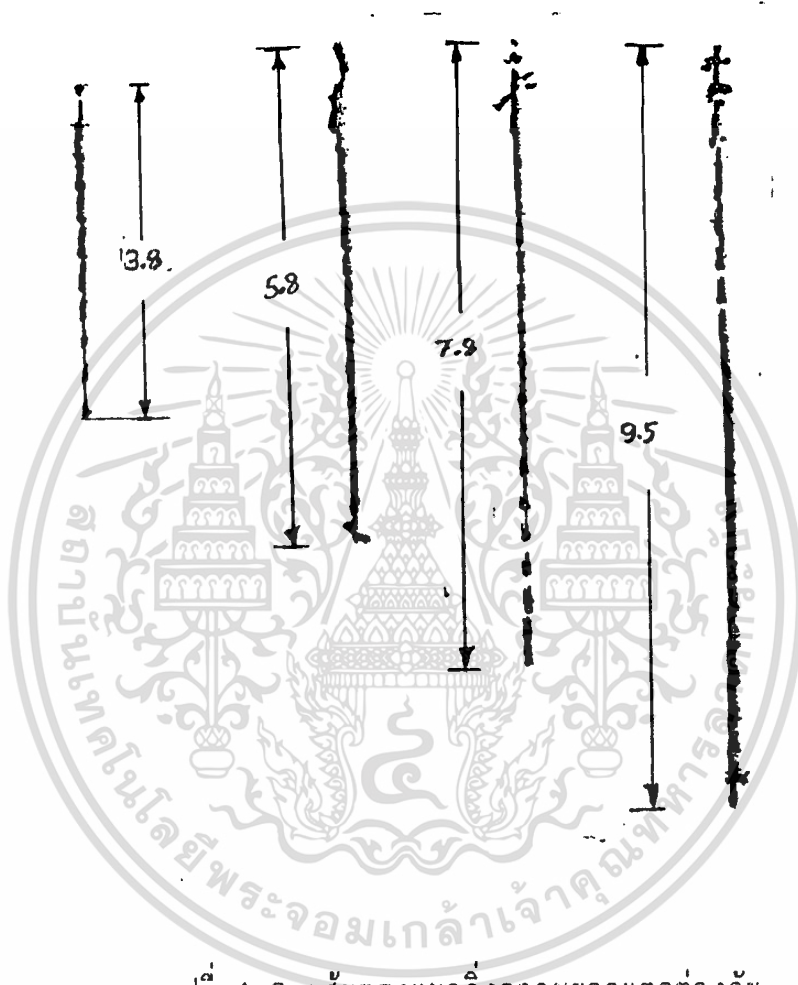
จากรูป 4.2 วัดความยาวเส้นตรงต่างๆได้ดังนี้ 3.8 ซม. 5.7 ซม. 7.2 ซม. และ 9.5 ซม. ตามลำดับ จึงสามารถกล่าวได้ว่าการเคลื่อนที่แบบควบคุมเส้นทางตามแนวราบมีค่าความผิดพลาดขนาด 6.25% เนื่องมาจากค่าแบ็กแลช (backlash) ของเฟืองส่งกำลังของแขนกล และจากค่าความผิดพลาดของเส้นตรงแต่ละเส้นมีขนาด 2 มม. 3 มม. 8 มม. และ 5 มม. ตามลำดับจึงไม่ให้ความผิดพลาดแบบสะสมตามความยาว

4.1.3 การทดลองหาค่าความผิดพลาดทางตำแหน่งในแนวตั้ง

วาดเส้นตรงแนวตั้งความยาวแตกต่างกันและใช้ความยาวส่วนย่อย

5 มิลลิเมตร ได้ผลลัพธ์ออกมาดังรูปที่ 4.3

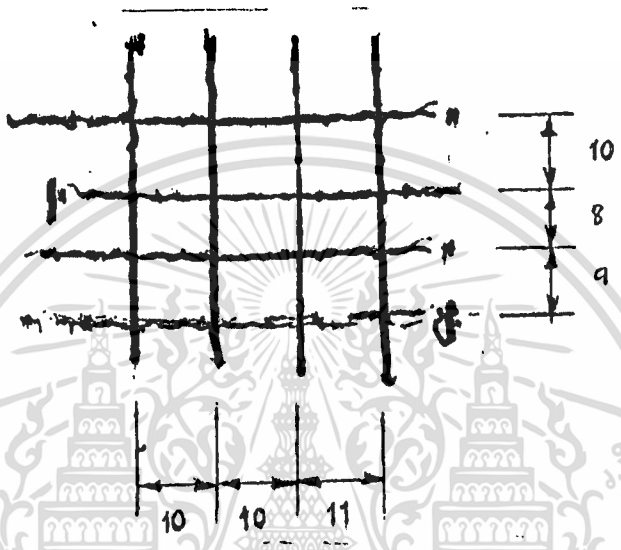
4 ซม. 6 ซม. 8 ซม. 10 ซม.



รูปที่ 4.3 เส้นตรงแนวตั้งความยาวแตกต่างกัน

จากรูปวัดความยาวเส้นตรงต่างๆ ได้ดังนี้ 3.8 ซม. 5.8 ซม. 7.8 ซม. และ 9.5 ซม. ตามลำดับ จึงสามารถกล่าวได้ว่าการเคลื่อนที่แบบควบคุมเส้นทางตามแนวตั้งมีค่าความผิดพลาด 3.95 % และเป็นความผิดพลาดชนิดไม่สะสมตามระยะทาง

4.1.4 การทดลองหาความแม่นยำโดยรวมเมื่อใช้วิธีการควบคุมแบบ เรขาคณิตโดยให้แขนกลวาดตารางที่มีช่องห่างในแนวดิ่งและแนวนอนเป็น 1 ซม. ได้ ผลลัพธ์ดังรูปที่ 4.4



รูปที่ 4.4 ตารางช่องห่าง 1 ซม. วาดจากแขนกล

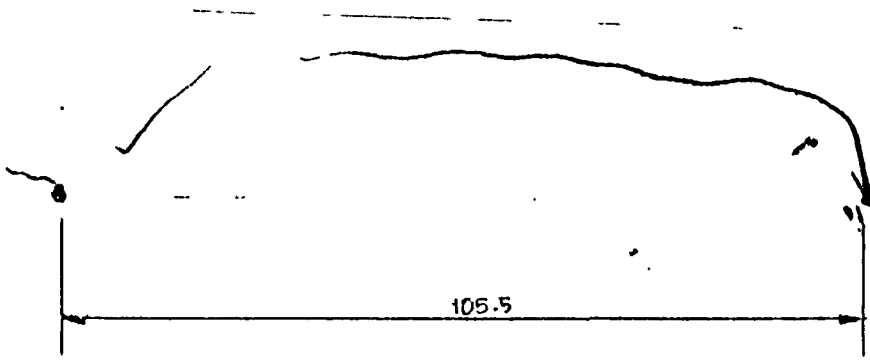
จากรูประยะห่างระหว่างเส้นแนวดิ่งที่วัดได้จากรูปที่ 4.4 มีค่าโดยเฉลี่ยเท่ากับ 10.33 มม. คิดเป็นความผิดพลาดได้ร้อยละ 3.3

และระยะห่างระหว่างเส้นแนวนอนที่วัดได้จากรูปที่ 4.4 มีค่าโดยเฉลี่ยเท่ากับ 9 มม. คิดเป็นความผิดพลาดได้ร้อยละ 10 คิดเป็นความผิดพลาดโดยรวม 6.65 %

4.1.5 การทดลองหาความผิดพลาดทางตำแหน่งของการควบคุมแบบควบคุมตำแหน่ง

การทดลองตอนนี้จะสั่งให้แขนกลเคลื่อนที่ไปยังจุดต่างๆดังนี้ เริ่มต้นจากตำแหน่ง (-50, 300, 320) ไปยังตำแหน่ง (50, 300, 280) ซึ่งมีระยะทางระหว่างจุดเท่ากับ 107.7 มม. ได้ผลลัพธ์ดังรูป 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูป 4.5 ผลลัพธ์การเคลื่อนที่แบบควบคุมตำแหน่ง

จากรูปที่ 4.5ก สามารถคำนวณค่าความผิดพลาดทางตำแหน่งได้ดังนี้

$$\begin{aligned} \text{ค่าความผิดพลาด (\%)} &= \frac{107.7 - 105.5}{107.7} * 100 \% \\ &= 2.04 \% \end{aligned}$$

พิจารณาเส้นตรงอีกเส้นหนึ่งคือเส้นตรงที่เริ่มต้นจากตำแหน่ง (-20, 300, 310) ไปยังตำแหน่ง (50, 300, 280) ซึ่งมีระยะทางระหว่างจุดเท่ากับ 76.15 มม. ได้ผลลัพธ์ดังรูป 4.5ข สามารถคำนวณค่าความผิดพลาดทางตำแหน่งได้ดังนี้

$$\begin{aligned} \text{ค่าความผิดพลาด (\%)} &= \frac{76.15 - 74.00}{76.15} * 100 \% \\ &= 2.88 \% \end{aligned}$$

= 2.82 %

จะเห็นว่าการเคลื่อนที่แบบควบคุมตำแหน่งมีค่าความผิดพลาดโดยเฉลี่ยเท่ากับ 2.43 %



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

จากการทดลองใช้งานโปรแกรมควบคุมแขนกลที่ได้พัฒนาขึ้นมาี้ ทำให้ทราบถึงอุปสรรคและข้อควรระวังต่างๆเป็นจำนวนมาก สิ่งที่ควรคำนึงถึงมากที่สุดคือค่าความผิดพลาดที่เกิดจากค่าแบคแลชของเฟืองที่ข้อต่อต่างๆของแขนกล เนื่องจากเป็นค่าที่จะก่อให้เกิดความผิดพลาดทางตำแหน่งได้มากที่สุด และนอกจากนั้นการควบคุมที่ใช้อยู่ในปริณญาณินธ์นี้ เป็นการควบคุมแบบวงจรเปิด (Open Loop) ซึ่งถ้าหากมีความผิดพลาดทางตำแหน่งเกิดขึ้นแขนกลก็จะไม่สามารถเข้าถึงจุดที่ต้องการได้นอกจากจะได้เพิ่มค่าชดเชยต่างๆเข้าไปในโปรแกรมการควบคุม สิ่งที่สำคัญอีกประการหนึ่งก็คือ แขนกลที่ใช้ทดสอบเป็นแขนกลที่ไม่สามารถควบคุมความเร็วของแขนช่วงต่างๆได้อย่างอิสระ จึงทำให้การควบคุมจะต้องใช้ค่าโดยประมาณเพื่อรอให้ส่วนต่างๆทำงานเสร็จพร้อมๆกันจึงทำให้เส้นทางการเคลื่อนที่ของจุดปลายมือไม่ตรงตามที่ต้องการนัก

สรุป การควบคุมแขนกลโดยใช้วิธีทางเรขาคณิตที่ได้พัฒนาขึ้นมาี้ มีข้อได้เปรียบกว่าการคำนวณโดยใช้วิธีการอื่นก็คือ การคำนวณจะใช้เวลาน้อยเนื่องจากมีฟังก์ชันการคำนวณน้อยกว่าวิธีอื่น ๆ มาก จึงสามารถนำไปใช้กับเครื่องคอมพิวเตอร์ขนาดเล็กๆที่มีขีดความสามารถที่ไม่สูงมากนักได้เป็นอย่างดี และสามารถควบคุมได้ทั้งตำแหน่งและเส้นทางการเคลื่อนที่โดยมีความผิดพลาดทางตำแหน่งบ้างเล็กน้อยซึ่งอยู่ในช่วงที่ยอมรับได้

ถ้าหากมีการพัฒนาแขนกลให้มีค่าแบคแลชน้อยลง และใช้มอเตอร์ไฟฟ้ากระแสตรง (D.C. Motor) ที่มีการควบคุมตำแหน่งเป็นระบบวงจรปิด และสามารถควบคุมความเร็วได้อย่างอิสระต่อกันเป็นหน่วยขับเคลื่อนและพัฒนาโปรแกรมการควบคุมให้สามารถปรับเปลี่ยนความเร็วมอเตอร์ให้สอดคล้องกับตำแหน่งที่ต้องการหยุดก็จะทำให้ระบบแขนกลมีความเที่ยงตรงมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MINI-ROBOT SIMULATOR ROBOT-1

OWNER'S MANUAL

 AMUSE COMPUTER CORP.

เอกสารนี้เป็นเอกสารที่สงวนเวลาไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่หรือใช้งานตามการก
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FOREWARD

Thank you for purchasing our Training Mini-Robot Simulator ROBOT-1. This Robot (ROBOT-1) was designed for speed and flexibility. It incorporates a variety of innovations. Its primary development objectives are for training and educational use and as an industrial simulator. To insure the most efficient use of this robot, please first read this operation manual thoroughly. Instructions contained herein are subject to change without prior notice.

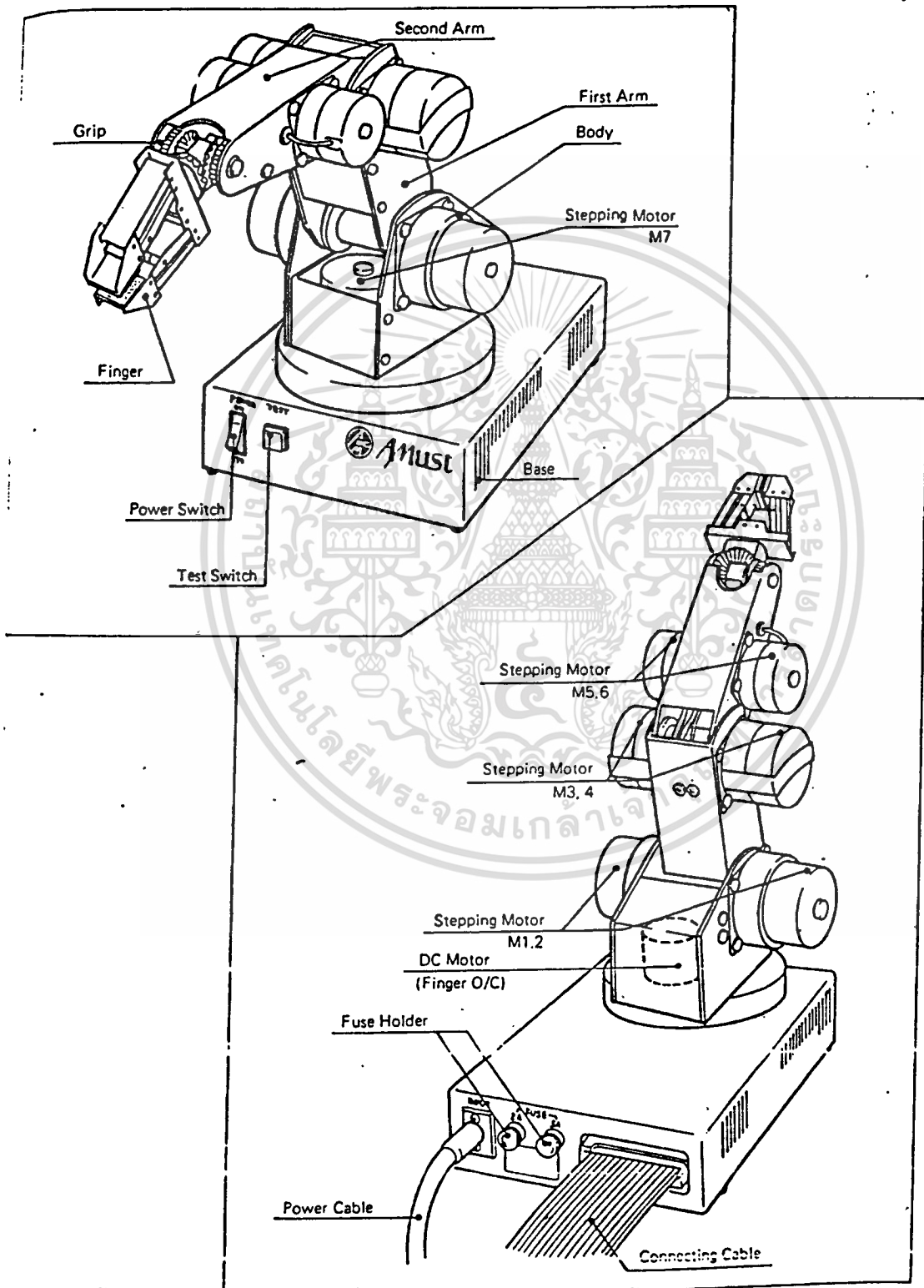
PRECAUTIONS

- This Robot (ROBOT-1) is not designed for actual industrial use, and attempted use as such should be avoided.
- Generally, the robot should be positioned on a level plane during operation. Avoid using it in direct sunlight or near a heat source.
- Be careful not to clog the air vent holes in the base or lower robot body. Avoid use of the robot where it may be affected by vibration or electrical interference.
- Attempting to move any axis by hand or forcibly stop the robot while it is in motion, could result in deviation of the program indication or damage to the mechanism.
- Do not leave the robot while it is in operation. Also, to avoid overheating of the motor, do not operate the robot continuously for more than an hour.
- After turning on electrical power, check that the cooling fan, located in the base, is operating. If not, it could result in damage to the circuits.

TABLE OF CONTENTS

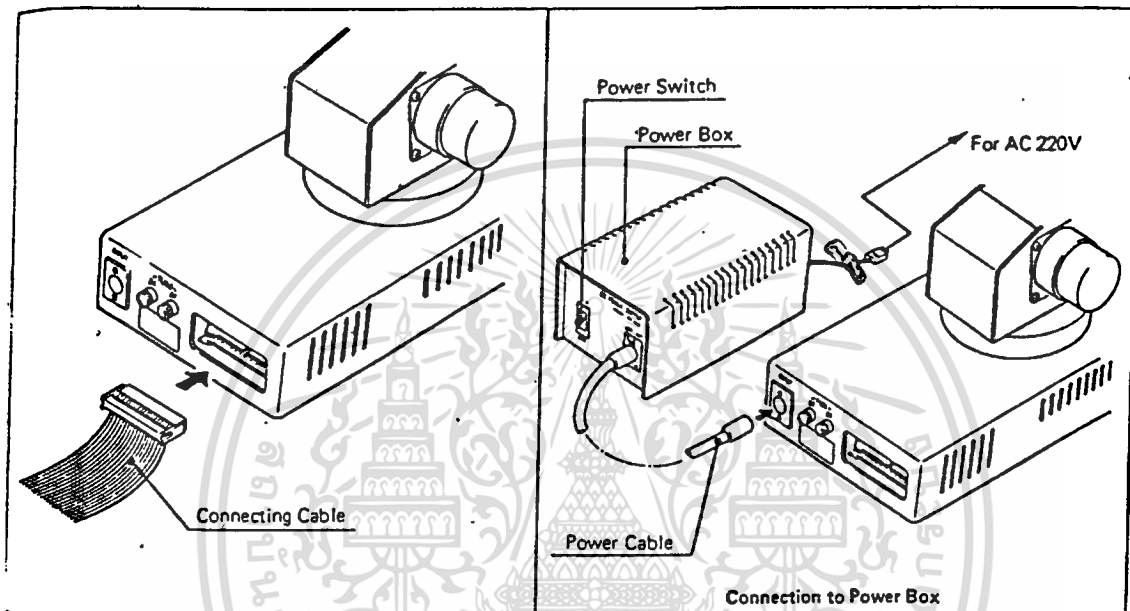
(1) Part Nomenclature	3
(2) Cable Connections	4
a. Connection of Personal Computer	
b. Connection of Power Box	
(3) Operation Test	5
a. Power Input Precaution	
b. Test Mode 1	
Note: Establishing Home Position	
c. Test Mode 2	
(4) Intelligence Command	8
a. Command Input	
b. Command and BASIC Language	
c. Command Input	
d. Explanation of Each Command	
(5) Specifications	17

Part Nomenclature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 Cable Connections



a. Connection to Personal Computer

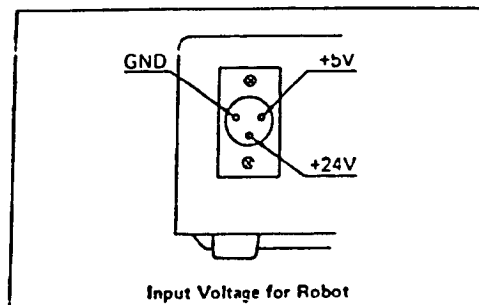
- 1) Insure that the connection between the robot and personal computer is secure so that the cable does not come loose during operation. A loose cable will prevent clear input of commands from the computer and result in inefficient operation, etc.
- 2) Use care in laying out the cable. Avoid pulling or twisting the cable.
- 3) Always use the cable provided by the computer manufacturer.

b. Connection to Power Box

First, confirm that the robot and power box switches are OFF, and then proceed as follows:

- (1) Connect the power cable between the robot and power box.
- (2) Insert the power box electrical cord into a 220V AC electrical outlet.

(NOTE: The electrical capacity of the power box is 220V AC. Do not attempt to use any other voltage source.)



3 Operation Test

a. Precautions for Electrical Power Input

First, turn On the power box switch and then the robot power switch. The grip will close when the robot power switch is turned ON. At this time, the following self-diagnoses will automatically take place. If there is an error, the grip will remain open.

(Self-Diagnosis Functions)

- a. ROM and SUM Check
- b. RAM battery Back-up Check
- c. RAM READ & WRITE Check

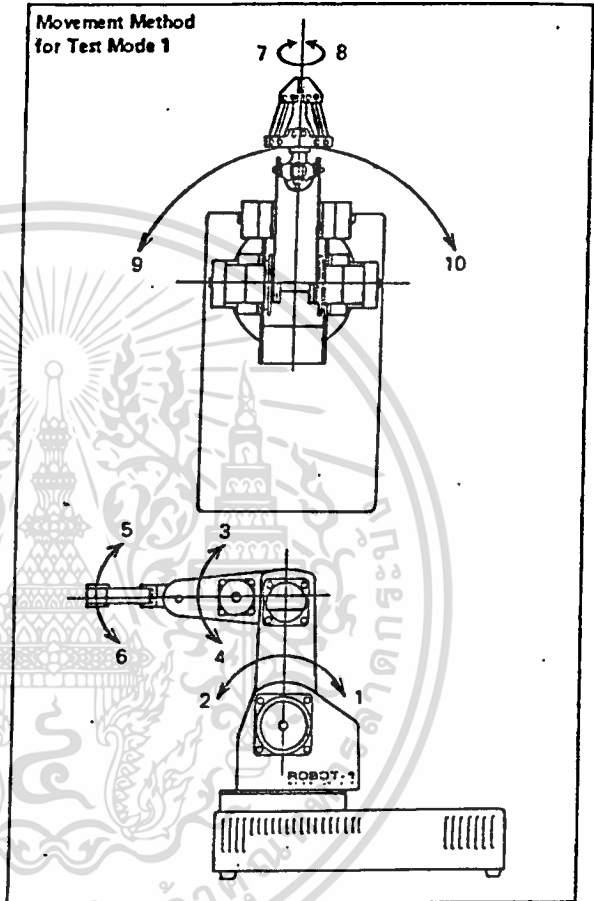
If there is an error, reinsert the power plug and commence operation after the grip closes.

b. Test Mode 1

With this mode, operation of each robot axis can be checked and the home position established.

While depressing the test switch, turn ON the power switch. Hold the test switch in this position for 1 to 6 seconds and then release it. Mode 1 will be activated. Subsequently, whenever the test switch is turned ON or OFF, each arm will move a little at a time.

To cancel Test Mode 1, turn OFF the power switch.

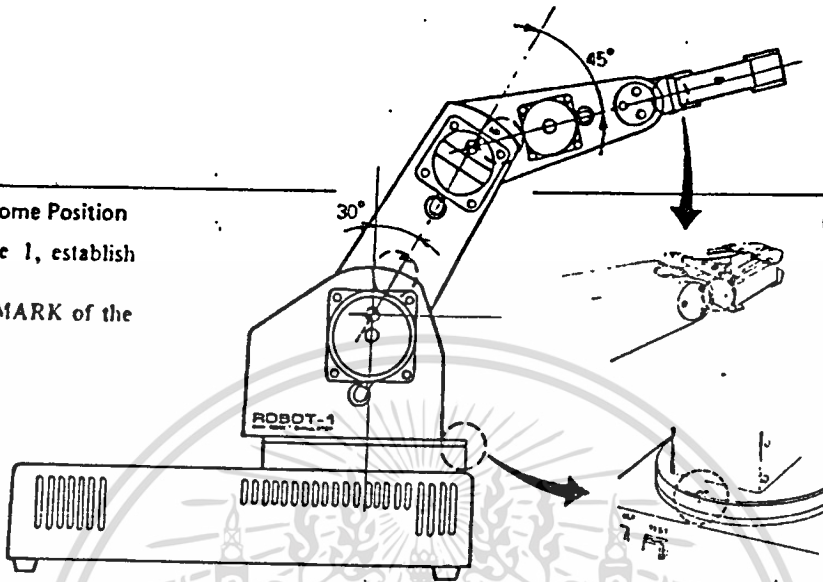


Operation will continue as shown as long as the test switch is depressed. When switch is released and then turned ON again, operation will continue from the subsequent sequence.

No.	Operation Component	Direction of Movement
1	1st Arm	Upward
2	1st Arm	Downward
3	2nd Arm	Upward
4	2nd Arm	Downward
5	Wrist Twist	Upward
6	Wrist Twist	Downward
7	Wrist Swivel	Clockwise (from front)
8	Wrist Swivel	Counter Clockwise (from front)
9	Body Rotation	Right (from front)
10	Body Rotation	Left (from front)

After Sequence 1, operation is repeated from No. 1

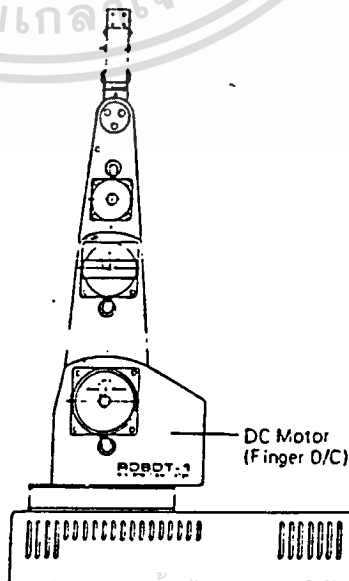
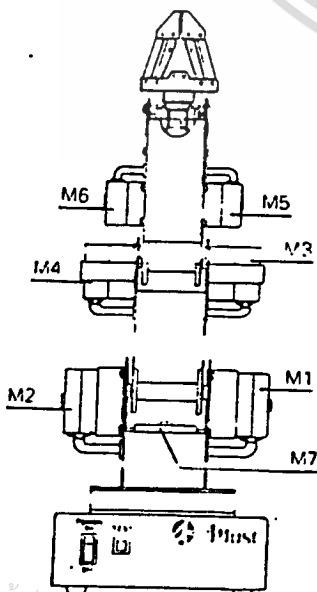
- Establishing Home Position
Using Test Mode 1, establish Home Position.
- Align the HP MARK of the robot.



Maximum Possible Movement Range from Home Position

Motor No.	Nomenclature	Movement Angle	Max. Move. Range from Home Pos.	
			+ Positive	- Negative
M1 & 2	1st Arm	0.1°/pulse	450 pulse	450 pulse
M3 & 4	2nd Arm	0.1°/pulse	450 pulse	450 pulse
M5 & 6	Wrist Twist	0.1°/pulse	900 pulse	900 pulse
M5 & 6	Wrist Swivel	0.1°/pulse	3600 pulse	3600 pulse
M7	Body Rotation	0.1°/pulse	1500 pulse	1500 pulse

Operation angle is 0.1° per step. Therefore, each angle (°) is 1/10 of the No. of steps.



C. Test Mode 2

After establishing home position of the robot with test mode 1, turn OFF the power. Then, while depressing the test switch, turn ON the power switch, and hold the test switch in this position for at least 6 seconds. When the robot begins to move, release the test switch, and execute as shown in the table below.

Operation No.	1st Arm (No. of Steps)	2nd Arm (No. of Steps)	Wrist Twist (No. of Steps)	Wrist Swivel (No. of Steps)	Rotation (No. of Steps)	Finger Position	Movement Speed	Waiting Time After Movement
1	-200	-200	0	0	900	OPEN	1	1 (Second)
2	-100	-200	0	0	0	OPEN	3	1
3	0	0	0	0	0	CLOSE	—	5
4	200	200	0	3600	-1800	CLOSE	1	1
5	-200	-200	0	0	0	CLOSE	3	1
6	0	0	0	0	0	OPEN	—	4
7	200	200	0	0	0	OPEN	3	1
8	100	200	0	0	900	CLOSE	1	1
9	-200	-200	0	0	-900	OPEN	1	1
10	-100	-200	0	0	0	OPEN	3	1
11	0	0	0	0	0	CLOSE	—	5
12	200	200	0	-3600	1800	CLOSE	1	1
13	-200	-200	0	0	0	CLOSE	3	1
14	0	0	0	0	0	OPEN	—	4
15	200	200	0	0	0	OPEN	3	1
16	100	200	0	0	-900	CLOSE	2	2
17	300	400	-800	0	0	CLOSE	3	1
18	-800	-100	1600	0	0	CLOSE	3	1
19	500	-300	-800	0	0	CLOSE	3	1

After No. 19, mode is repeated from No. 1

Operation is in sequence for each speed of Nos. 1 - 19 above.

If the test switch is depressed at any pause between each operation number, the robot will return to home position and stop, and test mode No. 2 will be terminated. From this point, commands from the computer will be accepted.

* Turning OFF the power switch will also

cancel test mode No. 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Intelligence Command

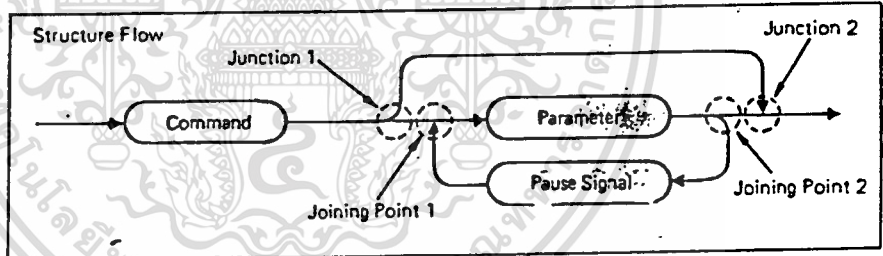
What is Intelligence Command ?

Complete control of robot operation would be beyond the processing capabilities of the computer. To overcome this problem, the robot is provided with a mechanism having, to some extent, an intelligence function so that the computer can relatively easily process robot movement.

Presently in interpreter language, most personal computers refer to this as BASIC, and this is pre-equipped in the robot. Therefore, even without including an assembler or machine language, resemblance to a high degree of automation process, etc. is possible.

In other words, Intelligence Command can be said to act as an interpreter for the robot.

Input of Commands



ITEM	FORM
Command	Alphabet Capitals
Parameter	Decimal (to 10th place), Integer (for negative, use a "-")
Pause Signal	Period (".") [ASC11 Code (2E) 16]

How to Read the Structure Flow Above

Flow is generally from left to right.

At a junction, advancement can be made along any course.

Advancement can be made in either left/right or up/down directions. At Junction No. 2, advance can be made in fore/aft and downward directions.

Flow can be made to follow along with the flow. In this case, at both Joining Points 1 & 2, flow can be in right/left directions.

Command and BASIC

A LPRINT (PRINT # n) is used for relaying intelligence commands to the robot. Commands to the robot are possible by the LPRINT arranged program shown at right.

```
10 LPRINT "H"  
20 LPRINT "I100.200....S10"  
30 LPRINT "I-100....."  
40 LPRINT "I..-200...S3C"  
50 LPRINT "N"  
60 END
```

Also, by use of DATA, the program above can be changed as shown at right.

```
100 DIM D$(10)  
110 DATA "H"  
120 DATA "I100.200....S10"  
130 DATA "I-100....."  
140 DATA "I..-200...S3C"  
150 DATA "N"  
160 REM --- DATA READ ---  
170 FOR I=1 TO 5  
180 READ D$(I)  
190 NEXT I  
200 REM --- COMMAND OUT ---  
210 FOR I=1 TO 5  
220 LPRINT D$(I)  
230 NEXT I  
240 END
```

The command explanation contains a usage sample which resembles the program below. This program can easily be monitored by TV and a command sent to the robot.

NOTE: Examples are shown for the BASIC language established by various computer manufacturers)

```
10 OPEN "C:" AS "LPT0:"  
20 INPUT A$  
30 PRINT #1, A$  
40 CLOSE #1  
50 GOTO 10
```

F-Basic (Fujitsu)

```
10 A$=""  
20 INPUT A$  
30 LPRINT A$  
40 GOTO 10
```

N-Basic (Nihon Denki)

Input Form for Each Command

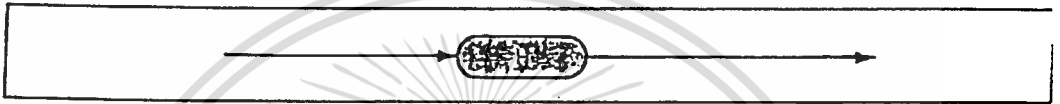
Command	No. of Parameters	Note
C	0	—
E	7	$-450 \leq a_1 \leq 450$ $-450 \leq a_2 \leq 450$ $-900 \leq a_3 \leq 900$ $-3600 \leq a_4 \leq 3600$ $-1500 \leq a_5 \leq 1500$
		Assured only within range at left
		$1 \leq \text{Speed Constant} \leq 5$
H	0	—
I	7	Same as Command E
K	7	Same as Command E
M	0	—
N	0	—
O	0	—
P	1	$1 \leq \text{Program No.} \leq 15$
R	1 ~ 15	$1 \leq \text{Program No.} \leq 15$
T	1	$1 \leq \text{Time Constant} \leq 99$
Z	0	—
NEW	1 ~ 15	$1 \leq \text{Program No.} \leq 15$

Definitions of each parameter, etc. are given on the

d. Explanation of Each Command

H HOME Command

Objective: To establish HOME position
Structure:



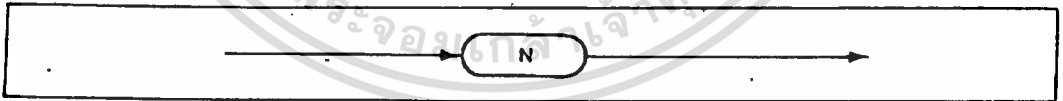
Explanation:

This is a command to establish basic starting point of robot movement.

- (1) When this command is given, the robot will return to home position.
- (2) As long as this command is not designated, home position will be that at the time electrical power was applied.
- (3) When "H" command is executed, internally the amount of all movement from the starting point will be zero.

N RESTORE HOME Command

Objective: To restore HOME Position
Structure:



Explanation:

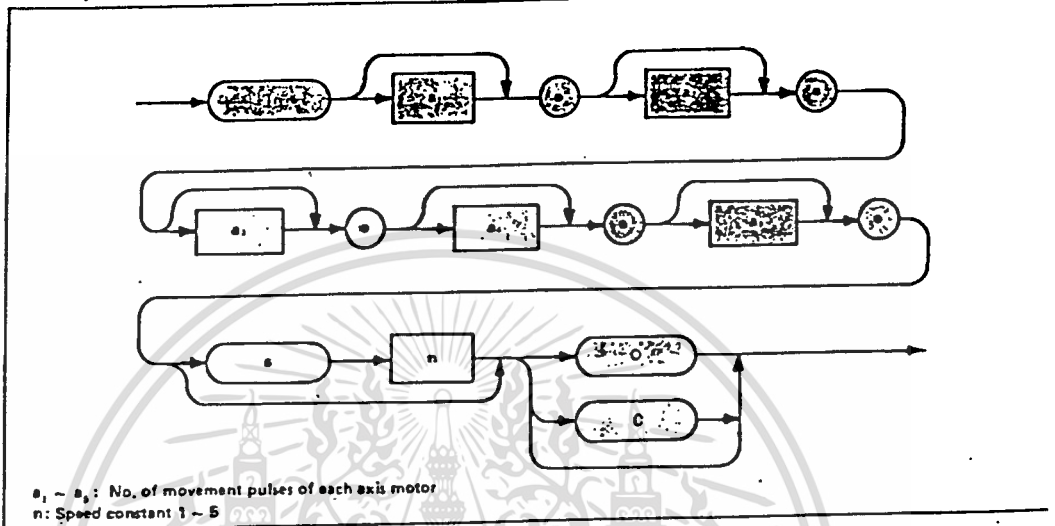
This is a command to cause the robot to move to the HOME position established by the HOME command.

- (1) Each axis motor causes movement to the HOME position established by the "H" command.
- (2) If HOME position has not been established ("H" Command not designated), it will return to the position it was in when electrical power was applied.

DIRECT MOVE Command

Objective: To move only the number of steps directly instructed.

Structure:



Explanation:

This is a command to cause each axis motor to move individually the number of steps stipulated by the parameter. However, finger positions will not change unless Finger OPEN/CLOSE instruction is given.

PROGRAM Command

Objective: To establish a teaching program No.

Structure:



Explanation:

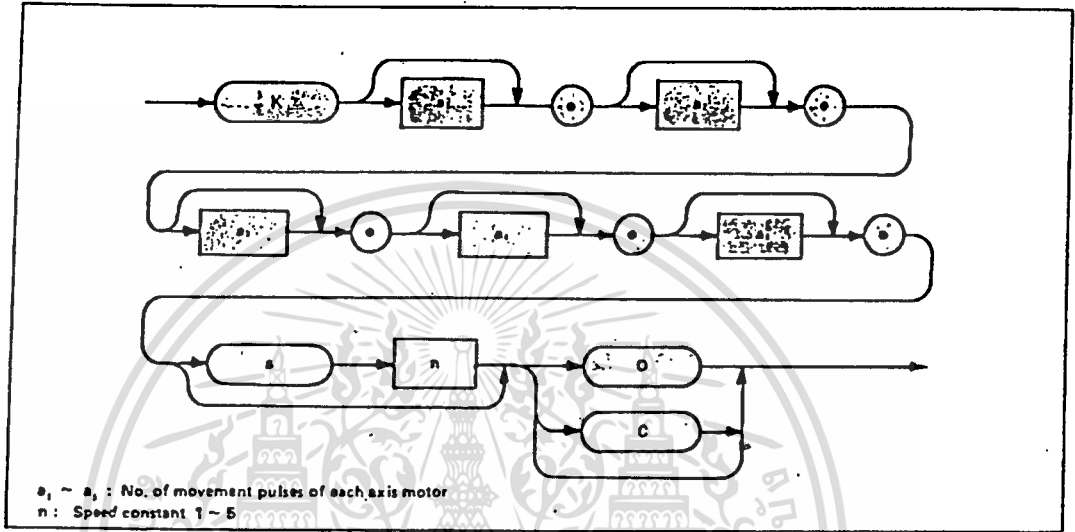
This is a command to establish a number which will cause the robot to remember a chain of movements.

- (1) A number already retained in the program cannot be stipulated.
- (2) This command should be stipulated before commands "P", "E" and "K".
- (3) The program number stipulated by "P" command is used by the "R" command.

KEYIN Command

Objective: To retain memory of movement

Structure:



Explanation:

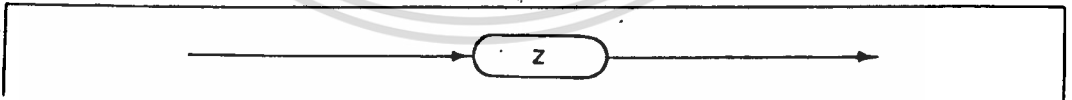
This is a command to cause the robot to remember certain movements.

- (1) Please refer to "I" command for parameter stipulation.
- (2) Use the "P" command.

END Command

Objective: To complete the program

Structure:



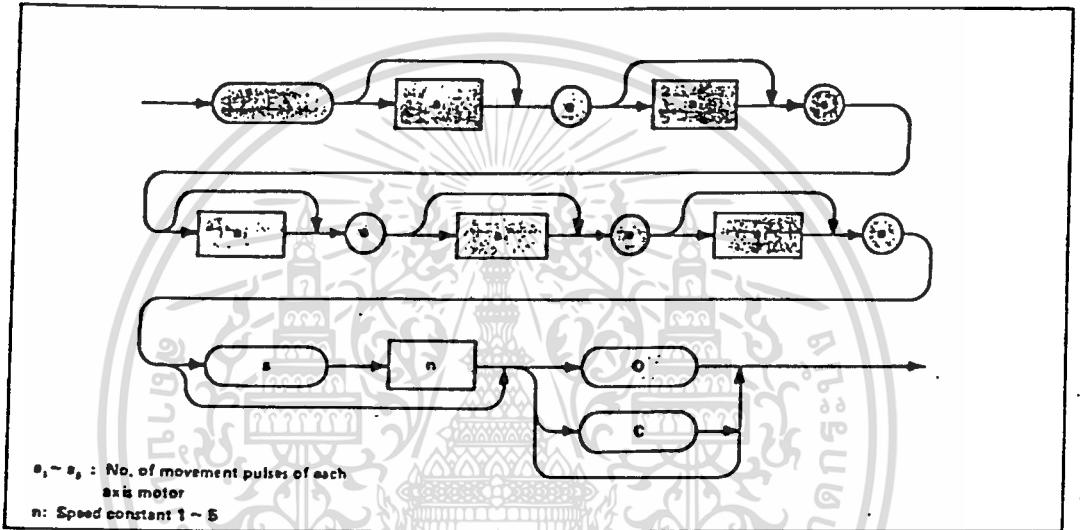
Explanation:

After teaching has been programmed, this command is used to cause termination during execution of a program.

TEACHING Command

Objective: To move to an established position

Structure:



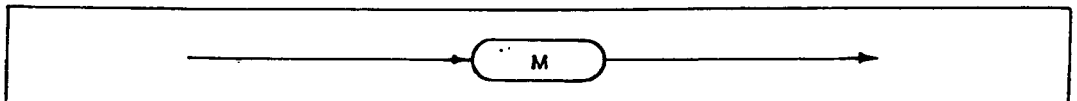
Explanation:

This command is used to move the robot, little by little, to a target position. The number of movements is voluntary. If a MEMORY command is input while the robot is moving to the target position, the position last completed will be retained in memory.

MEMORY Command

Objective: To cause memory of a chain of movements

Structure:



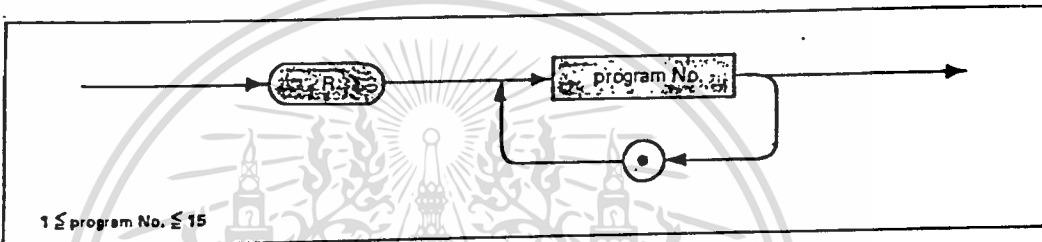
Explanation:

This is a command to cause the robot to remember an optimum position during teaching while the robot is moving. This command is used in contrast with "E".

RUN Command

Objective: To execute a teaching program

Structure:



Explanation:

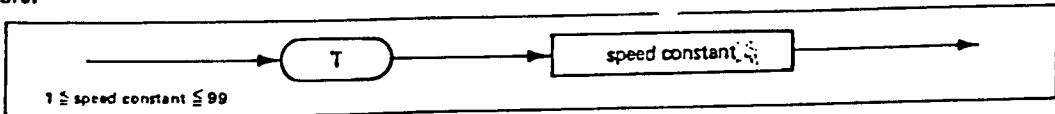
This is a command used to voluntarily designate a teaching program, and to execute, repeatedly, that program.

- (1) The program number can be designated 15 times.
- (2) Numbers not taught will not be executed.
- (3) There are two groups of numbers; 1 to 10 and 11 to 15. After execution of 1 to 10, the original point is restored. This is not true for 11 through 15.
- (4) When a teaching program is being executed during a RUN command, and "Z" command is designated, movement will be terminated after execution of one step in the program.

TIMER Command

Objective: To stop movement for a specified time

Structure:



Explanation:

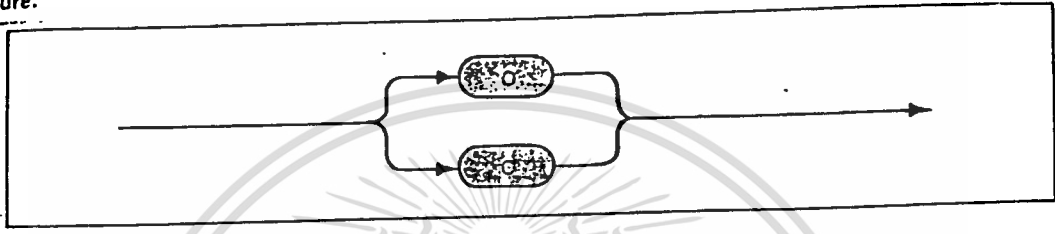
This command is used to stop the robot for a specified time.

- (1) During a teaching program, if this command is given after "K" command, one step of "K" command will be retained in memory. This is also true for "E" command.

- (2) Amount of movement retained by No. 8 is +20° for No. 1 Arm, +25° for No. Arm and +15° for Wrist Twist. Speed is 4.

O/C FINGER Command

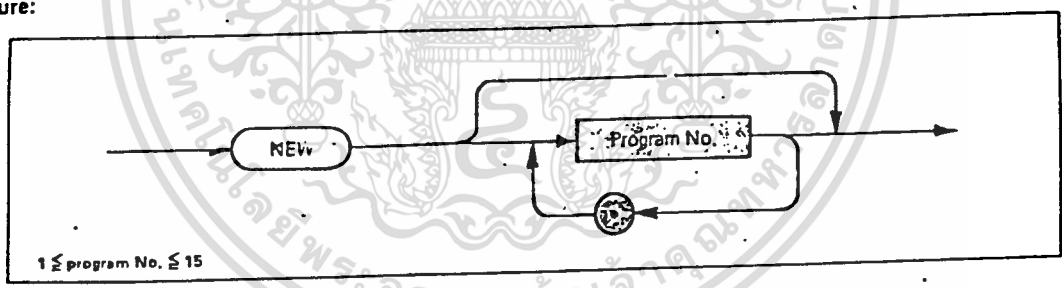
Objective: To open and close the hand.
Structure:



Explanation:
 This is a command to open and close the robot hand. O/C teaching is the same as for TIMER command.

NEW Program Command

Objective: To cancel a teaching program.
Structure:



- Explanation:**
- (1) This command is used to designate a program number for cancelling a teaching program.
 - (2) If there is no designation of a program number, all programs will be cancelled.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <cos.h>
#include <alloc.h>
#include <dir.h>
#include <bios.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <stdarg.h>
#include <graphics.h>

#define TO DEG 57.29578
#define TO RAD 0.0174532
#define TRUE 1
#define FALSE 0

/*=====
ROBOT PARAMETER
=====*/
#define L0 175
#define L1 120
#define L2 150
#define L3 90

/*=====
Max Angle supported by Mech.
structures
Operation step = 0.1 Degree
=====*/
int max_an[5]={ 75, 90, 90, 360, 150};
int min_an[5]={-15, 0,-70,-360,-150};

/*=====
FORMAT "l[shoulder].[elbow].[wrist_twist].[wrist_swivel].[body].[Sspeed][gripper]"
ANGLE 0 1 2 3 4 5 6
=====*/

/*=====
GLOBAL VARIABLE
=====*/
float angle[7],pangle[7]={30,45,0,0,90,0,0};
float range[7]={0.523596,0.785394,0,0,1.570788,0,0};
int px=0;
int py=292;
int pz=341;
int pnt=0;
int pws=0; /* last position in base frame */

int kx=0;
int ky=292;
int kz=341;
int kws=0;
int knt=0; /* destination */

```

```

int x,y,z,wt,ws,err;
int token[255];
int gm,gd,gx,gy;
FILE *fp;
char txt[10],command[30],fname[15];

struct ffb!k block;
struct viewporttype vinfo;

int gprintf(int xloc, int yloc, char %fmt, ... );

/*=====*/

main()
(
    /* draw in scale 1/2 */
    int ch;
    gd=DETECT;
    initgraph(&gd,&gm,"");
    fprintf(stdprn,"H\n");
    rectangle(1,1,400,300); /* viewport1 (2,2,399,299) */
    rectangle(1,1,639,347);
    outtextxy(5,305," ROBOT ARM CONTROL PROGRAM");
    line(5,315,220,315);
    outtextxy(5,320," ON GEOMETRIC APPROACH");
    outtextxy(5,330," CONTROL DEPARTMENT KMITL 1990");
    rectangle(405,125,635,340);
    x=px;y=py;z=pz;wt=pwt;
    aconvert();
    update_gr();

do
(
    setviewport(405,127,634,339,0);
    /* main screen start here */
    /* main viewport */
    ch=0;
    clearviewport();
    outtextxy(1, 1," MAIN MENU");
    outtextxy(1,10,"=====");
    outtextxy(1,20," 1.POSITION CTRL.");
    outtextxy(1,30," 2.CONTINUOUS PATH CTRL.");
    outtextxy(1,40," 3.RUN COMKAND FILE");
    outtextxy(1,50," 4.END PROGRAM");
    outtextxy(1,70," SELECT :");
    ch=getch();
    switch(ch)
    {
        case 0x31: position();break;
        case 0x32: conti(); break;
        case 0x33: rtoken();
                    rexec(); break;
        case 0x34: fprintf(stdprn,"N\n");break;
        default: break;
    }
}

```

เอกสารนี้เป็นเอกสารที่... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น... และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    while(ch!='\x34');
    closegraph();
}
/*****/

position()
{
    char tmp[10];
    setviewport(406,127,634,339,0);
    clearviewport();
    outtextxy(1,1," POSITION CONTROL ('q'uit?)");
    cuttextxy(1,10,"=====");
    outtextxy(1,20,"Move to X : ");ggets(100,20,tmp);
    if(!tmp=="q") return(0); else x=atoi(tmp);
    outtextxy(1,30,"Move to Y : ");ggets(100,30,tmp);y=atoi(tmp);
    outtextxy(1,40,"Move to Z : ");ggets(100,40,tmp);z=atoi(tmp);
    outtextxy(1,50,"Wrist W : ");ggets(100,50,tmp);wt=atoi(tmp);
    outtextxy(1,60,"Wrist Swi : ");ggets(100,60,tmp);ws=atoi(tmp);
        rangle[3]=TORAD*ws;
    outtextxy(1,70,"Gripper 0/1 ");ggets(100,70,tmp);rangle[6]=atoi(tmp);
    kx=x;ky=y;kz=z;kws=ws;kwt=wt;
    geo_cal(x,y,z,wt);
    if(!err) amust_dr();
    ggets(1,70,tmp);
}
/*****/

ccnti()
{
    char tmp[10];
    setviewport(406,127,634,339,0);
    clearviewport();
    outtext(" PATH CONTROL ('q'uit?)");
    outtextxy(1,10,"=====");
    outtextxy(1,20,"Move to X : ");ggets(100,20,tmp);
    if(!tmp=="q") return(0); else x=atoi(tmp);
    outtextxy(1,30,"Move to Y : ");ggets(100,30,tmp);y=atoi(tmp);
    outtextxy(1,40,"Move to Z : ");ggets(100,40,tmp);z=atoi(tmp);
    outtextxy(1,50,"Wrist W : ");ggets(100,50,tmp);wt=atoi(tmp);
    outtextxy(1,60,"Wrist Swi : ");ggets(100,60,tmp);ws=atoi(tmp);
        rangle[3]=TORAD*ws;
    kx=x;ky=y;kz=z;kws=ws;kwt=wt;
    path_ctrl(x,y,z,wt);
    gets(tmp);
}
/*****/

rtolen()
{
    int i.count=0;
    char command[10],flag,est=0;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ควรกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setviewport(406,127,634,339,0);
clearviewport();
```

```
for(i=0;i<255;i++,token[i]=0);
```

```
/*
```

Available command	token
=====	=====
end = End program	0
home = Home position	1
posi = Position control x y z swivel wrist	2
cont = Continuous path control x,y z swivel wrist	3
open = Open gripper	4
close= Close gripper	5
delay = wait (n3ec)	6
swivel = swivel wrist	7
twist = twist wrist	8

```
*/
```

```
grgetdir();
fp=fopen(fname,"r");
if(fp==NULL)
{
    outtextxy(1,40," ERROR in open file!");delay(1000);
    return(-1);
}
flag=1;
while((fscanf(fp,"%s",command)!=-1) && count<255 && flag) /* end of file */
{
    if(strcmp(command,"end")==0)
    {
        flag=0;est=1;
        token[count]=0;
    }
    else
    if(strcmp(command,"home")==0) token[count]=1;
    else
    if(strcmp(command,"open")==0) token[count]=4;
    else
    if(strcmp(command,"close")==0) token[count]=5;
    else
    if(strcmp(command,"posi")==0) token[count]=2;
    else
    if(strcmp(command,"cont")==0) token[count]=3;
    else
    if(strcmp(command,"delay")==0) token[count]=6;
    else
    if(strcmp(command,"swivel")==0) token[count]=7;
    else
    if(strcmp(command,"twist")==0) token[count]=8;
    else
    if(isdigit(*command) || *command=='-')
    {
        token[count]=atoi(command);
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
    gprintf(1,40,"Unrecognize [ %s ] \n",command);
    flag=0;delay(1000);
}

count++;
}
if(est==0) gprintf(1,50,"Missing END command");delay(1000);
fclose(fp);
}

/*=====*/
rEXEC()
{
    int i=0;
    while(token[i]!='0')
    {
        delay(1000);
        switch(token[...])
        {
            case 1: fprintf(stderr,"N\n");
                    x=px=kx=0; y=py=ky=292;
                    z=pz=kz=341; wt=pwt=kwt=0;
                    ws=pws=kws=0;
                    oangle[0]=angle[0]=30;
                    oangle[1]=angle[1]=45;
                    oangle[2]=angle[2]=0;
                    oangle[3]=angle[3]=0;
                    cangle[4]=angle[4]=90;
                    oangle[5]=angle[5]=0;
                    oangle[6]=angle[6]=0;
                    rangle[0]=0.523596;rangle[1]=0.785394;rangle[2]=0;
                    rangle[3]=0;rangle[4]=1.570788;rangle[5]=0;rangle[6]=0;
                    update_gr();
                    break;
            case 2:i++;
                    x=token[i]; i++;
                    y=token[i]; i++;
                    z=token[i]; i++;
                    angle[3]=token[i]*TORAD;i++;
                    wt=token[i];
                    geo_cal(x,y,z,wt);
                    if(!err)
                    {
                        kx=x;ky=y;kz=z;kwt=wt;kws=ws;
                        agust_drv();
                    }
                    break;
            case 3:i++;
                    x=token[i]; i++;
                    y=token[i]; i++;
                    z=token[i]; i++;
                    angle[3]=token[i]*TORAD;i++;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        kx=x;ky=y;kz=z;kwt=wt;kws=ws;
        path_ctrl(x,y,z,wt);
        break;
    case 4: fprintf(stdprn,"I....S10\n");
            delay(5000);
            break;
    case 5: fprintf(stdprn,"I....S1C\n");
            delay(5000);
            break;
    case 6:i++;
            delay(token[i]);
            break;
    case 7:i++;
            pws=token[i];
            angle[3]=token[i];
            sprintf(command,"I...%d...S1\n",angle[3]);
            fprintf(stdprn,"%s",command);
            break;
    case 8:i++;
            angle[2]=token[i];
            pwt=token[i];
            sprintf(command,"I...%d...S1\n",token[i]);
            fprintf(stdprn,"%s",command);
            break;
    }
    i++;
}
}

/*=====*/
geo_cal(x,y,z,wt)
int x,y,z,wt; /* (x,y,z) ,w=angle from XY plane */
{
    float za; /* arm's tip */
    float c,s,tmp,ww;
    float psize,pbase;
    float nz,nsize,zata;
    float zi;
    int i;

    err=FALSE;
    range[4]=atan2(y,t); /* base angle BODY */
    ww=wt*TORAD;

    /* now! translation along LB to shoulder joint */
    nz=z-L0;

    /* calculate for arm's tip */
    c=sin(ww);
    za=nz+L3*c;

    if(za<=-187) return(-1);
    /* vector P */
    tmp=pow(x,2)+pow(y,2);
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pbase=sqrt(tmp)-L1*cosiww);

psize=pow(za,2)+pow(pbase,2);
zetax=atan2(za,pbase);

tmp=(pow(L2,2)-pow(L1,2)-psize)/(-2*L1*sqrt(psize));
if(tmp>=(float)(-1.0) && tmp<=(float)(1.0))
    z1=acos(tmp);
else
{
    impossible();
    return(-1);
}
rangle[0]=asin(1)-z1-zetax; /* correct */

tmp=(psize-pow(L1,2)-pow(L2,2))/(-2.0*L1*L2);
if(tmp>=(float)(-1.0) && tmp<=(float)(1.0))
{
    rangle[1]=acos(tmp);
    rangle[1]=acos(-1)-rangle[1];
}
else
{
    putchar(0x07);
    impossible();
    return(-1);
}
rangle[2]=asin(1)+ww-(rangle[0]+rangle[1]);
aconvert();

/* check range */
for(i=0;i<5;i++)
    if((angle[i] > (float)max_an[i]) || (angle[i] < min_an[i]))
    {
        putchar(0x07);
        putchar(0x07);
        impossible();
        return(-1);
    }
px=x;py=y;pz=z;
pws=ws;pwt=wt;
}

/*=====*/

path_ctrl(xx,yy,zz,wt)
int xx,yy,zz,wt;
{
    float  dx,dy,dz,tx,ty,tz,tws,twt,dwt,dws;
    float  nstep,step=5;
    float  stx,sty,stz,swt,stws;
    float  v_size;
    int i;

    dx=dy=dz=0;      /* different of point */

    =====*/

update_gr()
{
    /* viewport 1 */
    /* viewport1 (2,2,399,299) */
    setviewport(2,2,399,299,0);
    clearviewport();
    printf(25,235,"X:%3d  Y:%3d  Z:%3d  W_S:%3d  W_T:%3d",px,py,pz,pws,pwt);

    outtextxy(210,115," X -X");
    outtextxy(290,30,"-Y");
    outtextxy(290,200," Y");
    line(300,50,300,190);
    circle(300,120,20);circle(300,120,40);
    circle(300,120,60);
    line(235,120,365,120);

```

```

diff=(angle[j]-cangle[j])*10;
if(diff>1 || diff<=-1)
:
    diff*=-1;
    itoa(diff,tmp,10);
    strcat(command,tmp);
}
strcat(command,".");
}
clr_str(&tmp,sizeof(tmp));
/* set to maximum speed angle[5]=1 */
angle[5]=1;
if(angle[5]<6 && angle[5]>0)
{
    itoa(angle[5],tmp,10);
    strcat(command,"S");
    strcat(command,tmp);
}

clr_str(&tmp,sizeof(tmp));
if(angle[6]==0) tmp='C'; else tmp='D';

strcat(command,tmp);
fprintf(stdout,"%s\n",command);
show();
for(j=0;j<7;j++)
    angle[j]=angle[j];
update_gr();
}

/*****
impossible()
{
    getviewsettings(&vinfo);
    setviewport(406,127,634,339,0);
    putch(7);
    err=TRUE;
    fprintf(1,102,"Position out of Range!");
    setviewport(vinfo.left,vinfo.top,vinfo.right,vinfo.bottom,vinfo.clip);
}

*****/

ggets(x,y,buf)
int x,y;
char *buf;
{
    int ch,index=0;
    do
    {
        ch=0;
        ch=getch();
        switch (ch)
        {
            case '\x23: if(index==0) break;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
    setcolor(BLACK);
    gprintf((x+((index-1)*8)),y,"%c",&(buf+(index-1)));
    index--;
    *(buf+index)=0;
    setcolor(WHITE);
}
case 0x00: break;
case 0x0d: *(buf+index)=(char) 0;
break;
default: gprintf((x+(index*8)),y,"%c",(char)ch);
*(buf+index)=(char) ch;
index++;
}
}
while(ch!=0x0d);
}
/*=====*/
int gprintf( int xloc, int yloc, char *fmt, ... )
{
    va_list argptr;
    char str[140];
    int cnt;

    va_start( argptr, format );
    cnt = vsprintf( str, fmt, argptr );
    outtextxy( xloc, yloc, str );
    yloc += textheight( 'H' ) + 2;
    va_end( argptr );
    return( cnt );
}
/*=====*/

clr_str(buff,size)
char *buff;
int size;
{
    int i;
    for(i=0;i<size;++i,++buff) *buff=0;
}

/*=====*/
aconvert()
{
    int i;
    for(i=0;i<7;++i)
        angle[i]=rangle[i]*TODEG;
}
/*=====*/
show()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 0x48: if(yy>0)
{
    if(((yy-1)*4+xx)<=count)
    {
        yy--;
        frame(xx,yy,xx,yy+1);
    }
}
break;
case 0x50: if(yy<(count/4))
{
    if(((yy+1)*4+xx)<=count)
    {
        yy++;
        frame(xx,yy,xx,yy-1);
    }
}
break;
case 0x4b: if(xx>0)
{
    if((yy*4+(xx-1))<=count)
    {
        xx--;
        frame(xx,yy,xx+1,yy);
    }
}
break;
case 0x40: if(xx<3)
{
    if((yy*4+(xx+1))<=count)
    {
        xx++;
        frame(xx,yy,xx-1,yy);
    }
}
break;
case 0x1c: if((yy*4+xx)>count) break;
else
    strcpy(fname,(ptr+(yy*4+xx)*13));
loop=0;
break;
}
}
}
putimage(0,0,gbuff,COPY_PUT);
setviewport(vinfo.left,vinfo.top,vinfo.right,vinfo.bottom,vinfo.clip);
free(gbuff);
}
else putchar(7);
}

frame(x,y,ox,oy)
int x,y,ox,cy;
{
    setcolor(BLACK);

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรนำออกนอกระบบ หรือทำซ้ำโดยไม่ได้รับอนุญาต การนำออกนอกระบบโดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย และต้องรับผิดชอบต่อเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 0x4E: if(yy>2)
{
    if((yy-1)*4-x)<=count)
    {
        yy--;
        frame(xx,yy,xx,yy+1);
    }
}
break;
case 0x50: if(y<(count/4))
{
    if((yy+1)*4-x)<=count)
    {
        yy++;
        frame(xx,yy,xx,yy-1);
    }
}
break;
case 0x4B: if(x>2)
{
    if((yy+4*(xx-1))<=count)
    {
        xx--;
        frame(xx,yy,xx+1,yy);
    }
}
break;
case 0x4D: if(x<2)
{
    if((yy+4*(xx+1))<=count)
    {
        xx++;
        frame(xx,yy,xx-1,yy);
    }
}
break;
case 0x5C: if((yy+4*xx)>count) break;
else
strcpy(fname,(tr+(yy+4*xx)+13));
loop=0;
break;
}
}
}

```

```

putimage(0,0,gbuff,COPY_PUT);
setviewport(vinfo.left,vinfo.top,vinfo.right,vinfo.bottom,vinfo.clip);
free(gbuff);
}
else putchar(7);
}

```

```

frame(x,y,ox,oy)

```

เอกสารนี้เป็นทรัพย์สินของสำนักงานเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 {
 ไม่ว่า `setcolor(BLACK)`; ทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
rectangle((cx#110)+13,(cy#15)+8,(cx#110)+115,(cy#15)+18);  
setcolor(WHITE);  
rectangle((x#110)+13,(y#15)+8,(x#110)+115,(y#15)+18);
```

]



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ รศ.ดร.โยธิน เปรมปราณีรัชต์ และ ผศ.ดร.จงกล งามวิวิทย์ เป็นอย่างสูง ที่ได้ให้คำปรึกษาแนะนำตลอดจนช่วยเหลือในด้านต่างๆเกี่ยวกับการทำปริญญาโทมาโดยตลอด ขอขอบคุณ อาจารย์ วรศักดิ์ จิตรภักดี อาจารย์ อังรงค์ศักดิ์ สุกใส และ อาจารย์ เกียรติวรรณ ทรงสัจย์ ที่ได้ให้ความช่วยเหลือด้านอุปกรณ์ที่ต้องใช้ในโครงการนี้เป็นอย่างดี ขอขอบคุณห้องสมุดคณะวิศวกรรมศาสตร์ลาดกระบัง และ ห้องสมุดกลางของสถาบัน ที่เป็นแหล่งสนับสนุนทางด้านตำราและเอกสารอ้างอิงต่างๆและต้องขอขอบคุณเพื่อนๆและน้องๆทุกท่านที่ได้ให้ความช่วยเหลือในด้านต่างๆจนปริญญาโทฉบับนี้สำเร็จลงได้ด้วยดี



ธเนศ สุวรรณผ่อง
สุรเชษฐ์ รัตนภาส
2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้