



การควบคุมหุ่นยนต์อุตสาหกรรมด้วยไมโครคอมพิวเตอร์
INDUSTRIAL ROBOT BASE ON MICROCOMPUTER



โดย

นาย กลยุทธ ชัยลอม

นาย จักรกริช ชัยเทวรักษ์

นาย ชูพงศ์ ช่วยเพ็ญ

วัน เดือน ปี..... ๓๑ ๗๐๑ ๒๕๒๐

เลขทะเบียน..... ๐๓๗๐๐๑

เลขเรียกหนังสือ..... T 38094 ก ๒๘๗ ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมเครื่องกล
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง
037001

การควบคุมหุ่นยนต์อุตสาหกรรมด้วยไมโครคอมพิวเตอร์
INDUSTRIAL ROBOT BASE ON MICROCOMPUTER



ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2538

ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การควบคุมหุ่นยนต์อุตสาหกรรมด้วยไมโครคอมพิวเตอร์

ผู้จัดทำ

1. นาย กลยุทธ ชัยลอม 35104016
2. นาย จักรกริช ชัยเทวรักษ์ 35104060
3. นาย ชูพงศ์ ชวยเพ็ญ 35104123



.....อาจารย์ที่ปรึกษา

(ผศ.พรศักดิ์ อรรถวานิช)



.....อาจารย์ที่ปรึกษา

(อ.กวิน สนิธิเพิ่มพูน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมหุ่นยนต์อุตสาหกรรมด้วยไมโครคอมพิวเตอร์

นายกฤษฎ ชัยลอม*

นายจักรกริช ชัยเทวรักษ์*

นายชูพงศ์ ช่วยเพ็ญ*

อาจารย์ที่ปรึกษา

นายพรศักดิ์ อรรถวานิช**

นายกวิน สนิธิเพิ่มพูน***

ปีการศึกษา 2538

บทคัดย่อ

ปริญญาานิพนธ์นี้มีจุดประสงค์เพื่อศึกษาการควบคุมหุ่นยนต์อุตสาหกรรม 5 แกน รุ่นYasukawa Motoman L-3 ด้วยไมโครคอมพิวเตอร์ ในปริญญาานิพนธ์นี้จะประกอบไปด้วย คณิตศาสตร์ของหุ่นยนต์ ,forward kinematics, inverse kinematics และ robot simulation โดยเฉพาะอย่างยิ่งการแสดงลักษณะของหุ่นยนต์บน ไมโครคอมพิวเตอร์ก่อน เนื่องจากอุปกรณ์สำคัญของหุ่นยนต์ยังไม่พร้อมที่จะให้นำมาใช้งานได้ โปรแกรมที่แสดง การเคลื่อนที่ของหุ่นยนต์นี้ชื่อ Robot Control ซึ่งถูกออกแบบมาเพื่อหุ่นยนต์ Yasukawa Motoman L-3 เท่านั้น ดังนั้นเราจะสามารถเห็นภาพของหุ่นยนต์โดยคร่าว ๆ บนไมโครคอมพิวเตอร์ได้ และในอนาคตข้างหน้าเราจะมีการพัฒนาปรับปรุง หุ่นยนต์และโปรแกรมควบคุมให้สามารถทำงานที่ซับซ้อนมากยิ่งขึ้นได้

* นักศึกษา ระดับปริญญาตรี ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

** ผู้ช่วยศาสตราจารย์ ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

*** อาจารย์ ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDUSTRIAL ROBOT BASE ON MICROCOMPUTER

Mr. Konlayuth Chailorm*

Mr. Jakkrit Chaitewaruk*

Mr. Choopong Chuaypen*

Advisor

Assist. Prof. Pornsak Attavanich**

Mr. Kawin Sonthipermpon***

Academic Year 1995

ABSTRACT

This thesis aims to study about 5-axis robot model Yasukawa Motoman L-3 that is controlled by microcomputer. We represent about this robot about mathematical of robot manipulation, forward kinematics, inverse kinematics, and robot simulation of Yasukawa Motoman L-3. Especially, our robot simulation, we simulate characteristic of robot in microcomputer early since our robot is not moving because we have not some equipment and we can not operate it. Robot Control Program is designed for Yasukawa Motoman L-3 only. So that, we can see all link of robot in microcomputer. In future, we will develop this robot and control program to operate for more complicate work.

* Bachelor Degree Student, Mechanical Engineering Department, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang.

** Assistant Professor, Mechanical Engineering Department, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang.

*** Lecturer, Mechanical Engineering Department, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 Mathematics of Robot Manipulation	3
บทที่ 3 Coordinate System in Robot Application	17
บทที่ 4 การคำนวณสำหรับ Motoman L-3	26
บทที่ 5 Robot Simulation	40
บทที่ 6 การปฏิบัติงานของโปรแกรม Robot Control	81
บทที่ 7 บทสรุปและข้อเสนอแนะ	95
เอกสารอ้างอิง	109

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากภาวะเศรษฐกิจในปัจจุบัน มีการแข่งขันทางด้านเทคโนโลยีการผลิตสูงขึ้น จะต้องทำการผลิตให้มีชิ้นส่วนเสียน้อยที่สุด เพื่อลดต้นทุนที่สูญเสียไป และเพื่อความเที่ยงตรงรวมทั้งความเร็วในการผลิต ซึ่งจะได้ผลตามความต้องการดังกล่าวข้างต้น และยังสามารถสร้างความเชื่อถือในผลิตภัณฑ์ของตนได้ด้วย

ระบบอัตโนมัติที่นำมาใช้นอกจากระบบ CAD/CAM , CNC แล้วยังรวมถึงการใช้หุ่นยนต์ในการผลิตด้วย เนื่องจากหุ่นยนต์อุตสาหกรรมมีต้นทุนสูงจึงไม่เหมาะสมที่จะผลิตเพื่อใช้งานเฉพาะอย่าง ดังนั้นเราจึงทำการศึกษาถึงการควบคุมการทำงานของหุ่นยนต์อุตสาหกรรม เพื่อจะได้นำมาใช้กับหุ่นยนต์อุตสาหกรรมให้สามารถทำงานได้หลากหลายขึ้นโดยเพียงแต่เปลี่ยนโปรแกรมควบคุมเท่านั้น

1.2 วัตถุประสงค์ของโครงการ

- เพื่อศึกษาถึงความสัมพันธ์ของระบบพิกัด ที่เกิดการเปลี่ยนแปลงเมื่อเทียบกับระบบพิกัดใดพิกัดหนึ่ง
- เพื่อศึกษาการนำความสัมพันธ์ของระบบพิกัด มาประยุกต์ใช้กับหุ่นยนต์ อุตสาหกรรม ประเภท manipulation
- เพื่อศึกษาถึงวิธีทำรูปแบบลักษณะของหุ่นยนต์เมื่อทราบความสัมพันธ์ระหว่างระบบพิกัดที่จุดปลายของหุ่นยนต์และระบบพิกัดฐาน
- เพื่อศึกษาถึงการควบคุมการวางแผนการเคลื่อนที่ของหุ่นยนต์
- เพื่อการศึกษการพัฒนาโปรแกรมการควบคุมหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถสร้างแบบจำลองของการเคลื่อนไหวของหุ่นยนต์ได้
- เพื่อเป็นแนวทางในการพัฒนาการควบคุมหุ่นยนต์อื่น ในการใช้งานต่างๆ

1.4 แผนการดำเนินงานในการทำโครงการ

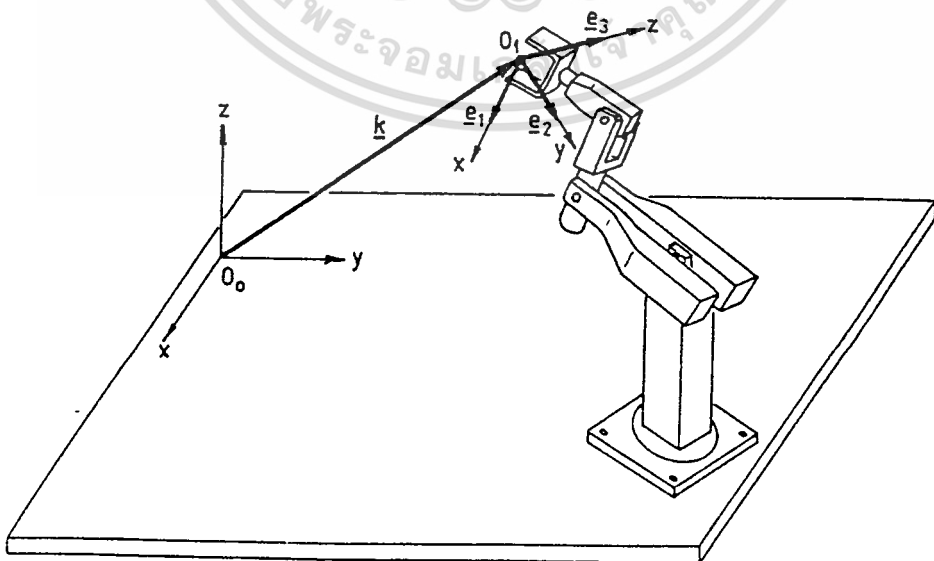
1. ศึกษาทฤษฎีพื้นฐานการสร้าง Kinematics ของหุ่นยนต์และสร้าง Kinematics ของหุ่นยนต์ต้นแบบ Yasukawa Motoman L-3
2. ศึกษาลักษณะของหุ่นยนต์เมื่อทราบ Kinematics (คือการหา inverse kinematics)
3. ทำการทดสอบ kinematics และ inverse kinematics
4. ทำการสร้างโปรแกรมการคำนวณ kinematics และ inverse kinematics
5. ศึกษาถึงการ Simulation ของหุ่นยนต์ที่แสดงผลทางจอคอมพิวเตอร์
6. สร้างโปรแกรม Simulation เพื่อใช้กับการจำลองสถานการณ์การเคลื่อนที่ของ Motoman L-3
7. ทดสอบความถูกต้องของการควบคุม

Mathematics of Robot Manipulation

2.1 การแปลงพิกัดร่วม (Homogeneous Coordinate Transformer)

ในการที่จะบอกพิกัดใดๆ ของวัตถุ จะใช้ตำแหน่งและมุมภายในพิกัด เพื่อแสดงตำแหน่งที่วัตถุนั้นอยู่ ทั้งนี้มีความจำเป็น เพื่อที่จะนิยามตำแหน่งและมุมที่สัมพันธ์กับระบบพิกัด รวดเร็วและเห็นได้ชัด และการใช้ homogeneous transformer เป็นวิธีที่มีเป็นประโยชน์มากที่สุดวิธีหนึ่ง [4]

ให้จุด O_0 คือ ฐานของระบบพิกัด (base coordinate system) กำหนดให้เวกเตอร์ k บอกตำแหน่งของระบบพิกัดอื่นอีกระบบพิกัดที่มีจุดกำเนิด O_1 ในฐานของระบบพิกัด O_0 เวกเตอร์ k บอกเพียงตำแหน่งของ O_1 เท่านั้นโดยไม่ได้บอกว่า มุมมีความสัมพันธ์อย่างไรกับระบบพิกัด O_0 แต่มุมสัมพันธ์สามารถบอกได้โดยเมตริกซ์ R ซึ่งเป็นเวกเตอร์ สำหรับการหมุนเป็นเมตริกซ์ 3×3 element ในเมตริกซ์การหมุน R ประกอบด้วย เวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ของระบบพิกัด O_1 ตามแนวแกนของระบบพิกัด ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 แสดงการอ้างอิงพิกัดฐานและการแปลงระบบพิกัด (translation coordinate system)

เวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ของระบบพิกัด ใช้ในการหาการแปลงระบบพิกัด การหมุน จากความสัมพันธ์ของเวกเตอร์และสเกลลา

$$e_1 = e_2 \times e_3 \quad \text{คือการคูณกันทางเวกเตอร์ (Cross product)}$$

$$e_2 \cdot e_3 = 0 \quad \text{คือการคูณกันทางสเกลลา (Inner product)}$$

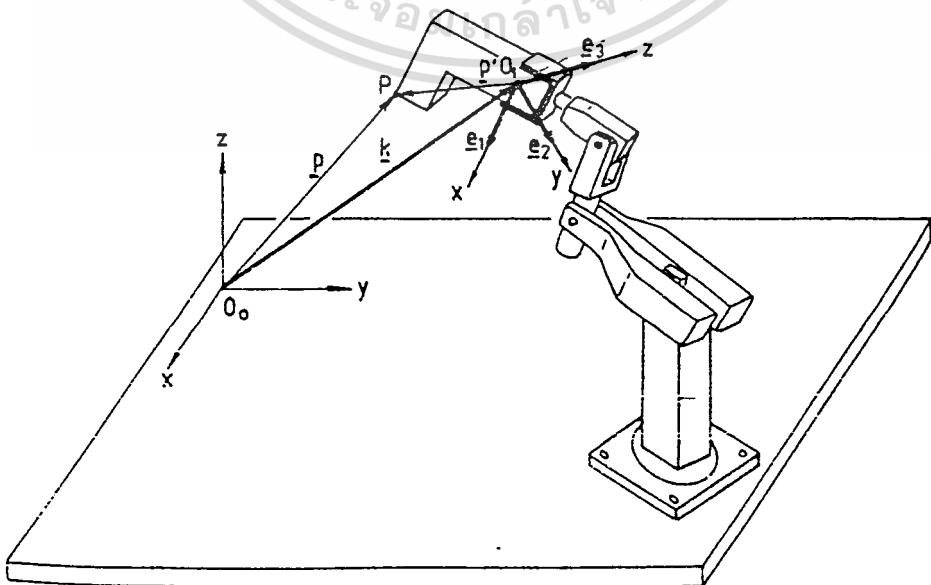
สรุปว่าผลของการคูณกันทางเวกเตอร์ (Cross product) ระหว่างเวกเตอร์ 1 หน่วย จะได้เวกเตอร์ที่ 3 และผลของการคูณกันทางสเกลลา (Inner product) จะได้ภาพฉายเวกเตอร์บนอีกเวกเตอร์หนึ่ง จะมี nonzero vector ที่ตั้งฉากซึ่งกันและกัน คือ

$$e_1 \cdot e_2 = e_2 \times e_3 = e_2 \cdot e_3 = 0$$

ดังนั้น ในชุดพิกัดใดๆ ในเวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ที่ฉายบนพิกัดของระบบพื้นฐาน จะมีคุณสมบัติข้างต้นก็ต่อเมื่อเป็นระบบพิกัดมุมฉาก [3]

Rotation matrix (R) คือ เมตริกซ์มุมฉาก 3×3 มีตัวประกอบ x, y, z ของเวกเตอร์ 1 หน่วย (e_1, e_2, e_3) ที่ฉายบนพิกัดของระบบพิกัด

$$R = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix}$$



รูปที่ 2.2 แสดงพิกัดสัมพันธ์และพิกัดสัมพันธ์มุมฉาก. k คือเวกเตอร์ตำแหน่งจากระบบพิกัดฐาน O_0 ถึงจุดกำเนิด O_1 . p' คือเวกเตอร์ตำแหน่งจากจุดกำเนิด O_1 ถึงจุด P . p คือเวกเตอร์ตำแหน่งจากระบบพิกัดฐานถึงจุด P ของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.2 พิจารณาพิกัดของจุด P บนโครงพื้นฐาน(base frame)เมื่อตำแหน่งของมันมีส่วนเกี่ยวข้องกับระบบพิกัด ตำแหน่งของจุด P บนระบบพิกัดฐานจะถูกคำนวณโดย premultiplying the vector p' คูณกับเวกเตอร์หมุน R บวกด้วยเวกเตอร์ ตำแหน่ง k

$$p = k + R * p'$$

แต่มีเมตริกซ์อีกอันที่แสดงผลลัพธ์ออกมาเหมือนกัน เมื่อเวกเตอร์ตำแหน่ง k ถูกรวมกันในตำแหน่งที่ 1 และ 4 กลายเป็น element ที่ 4 ของเวกเตอร์ p' ดังสมการ

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \end{bmatrix} * \begin{bmatrix} p'_x \\ p'_y \\ p'_z \end{bmatrix}$$

ที่แถวที่ 4 ของเมตริกซ์ 3x4 เมื่อค่า 0 3ตัว และค่า 1 อีกหนึ่งค่าทำให้เป็นเมตริกซ์ 4x4 เราสามารถสร้าง homogeneous transformer matrix ได้ดังสมการต่อไปนี้

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix}$$

A homogeneous transformer matrix ได้รวมเอาเวกเตอร์ตำแหน่ง k กับเวกเตอร์การหมุน R เพื่อที่จะได้ ตำแหน่งและมุมที่สมบูรณ์ของระบบพิกัดอีกระบบหนึ่งที่มีส่วนเกี่ยวข้องกับโครงสร้างพื้นฐาน (base frame) เมื่อนำการแปลงพิกัดของจุด P จากระบบพิกัดหนึ่งไปยังระบบพิกัดหนึ่ง [3]

แสดงในรูป 2.2 และมีสมการ homogeneous transformer matrix ให้เป็น T ที่ประกอบด้วยการหมุน R และการแปลง(transformer)ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T = \left[\begin{array}{ccc|c} & R & & k \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{array} \right]$$

R คือ เมทริกซ์ย่อย 3x3 ในส่วนของการหมุน (Rotation part)

k คือ 3x1 เมทริกซ์ ในส่วนของ transformer matrix

ส่วนของการแปลงคือ คอลัมน์เวกเตอร์ที่บอกตำแหน่งของ O_0 ไปยัง O_1 นอกจากนี้เรายังแปลง 4x4 matrix ที่ใช้ในการคำนวณ แชนกอลออกเป็น 2 แบบ

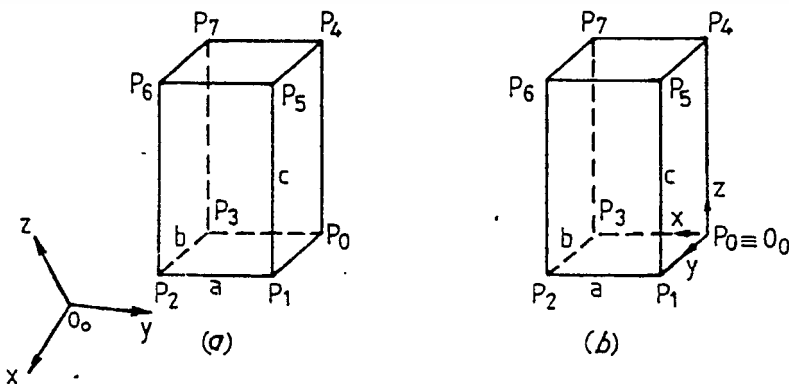
แบบที่ 1 บอกระบบพิกัดโดยสัมพันธ์กับโครงสร้างพื้นฐาน ตามที่ได้เสนอมมาแล้ว

แบบที่ 2 บอกความสัมพันธ์ระหว่าง 2 ระบบพิกัด โดยที่เชื่อมต่อกันอยู่

2.2 การบอกลักษณะของวัตถุในทางคณิตศาสตร์(Mathematics Description of Objects)

จุดมุ่งหมายในหัวข้อนี้คือ การแสดงการใช้ homogeneous transformer ในการบอกลักษณะของวัตถุ และตำแหน่งของมันในรูปแบบที่เข้าใจได้เพื่อใช้ในการคำนวณ วัตถุล้อมรอบด้วย planar surface ซึ่งบอกได้ด้วยเมทริกซ์ 4xN โดย N คือ จำนวนจุดปลายของวัตถุ ที่แต่ละจุดจะแสดงโดย position vector มีรูปแบบดังนี้

$$[x \ y \ z \ 1]^T ; T \text{ คือ transpose}$$



รูปที่ 2.3 แสดงจุดปลายลูกบาศก์ $P_0 - P_7$ (a) ระบบพิกัดวัตถุที่ไม่ขึ้นอยู่กับลักษณะของลูกบาศก์ (b) ระบบพิกัดที่กำหนดตามลักษณะของลูกบาศก์. โดยให้จุดกำเนิดคือ P_0

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบอกลักษณะของวัตถุ โดย planar surface มี 2 รูปแบบคือ [9]

1. จุดกำเนิดของระบบพิกัดวัตถุ สามารถมีตำแหน่งอิสระทุกๆลักษณะและมุมโคจรพิกัดของมันเป็นมุมปิด ดังรูป 2.3(a) รูปแบบทั่วไปของเมตริกซ์ที่ใช้แสดงวัตถุ N จุดปลายคือ

$$[\text{Object}] = \begin{bmatrix} x_0 & x_1 & - & - & - & x_{N-1} \\ y_0 & y_1 & - & - & - & y_{N-1} \\ z_0 & z_1 & - & - & - & z_{N-1} \\ 1 & 1 & - & - & - & 1 \end{bmatrix}$$

2. จุดกำเนิดของระบบพิกัดวัตถุ กำหนดให้อยู่กับที่ ดังรูป 2.3(b) ยกตัวอย่างของจุดปลายเช่น จุดศูนย์กลางโน้มถ่วง สามารถเลือกจุดกำเนิดได้

ให้วัตถุเป็นลูกบาศก์ที่มี 8 จุดปลาย $P_0 - P_7$ ในพิกัดคาตาเซียน ดังรูปที่ 2.3(b) จุดกำเนิดอยู่กับที่ ณ จุด P_0 แกนของระบบพิกัดวางแนวกับของจุดตัดที่ผ่านจุดกำเนิด มีปริมาตร $a \times b \times c$ มีทิศทาง x, y, z ตามลำดับคอลัมน์เมตริกซ์ของลักษณะวัตถุสำหรับจุด P_0 คือ

$$P_0 = [0 \ 0 \ 0 \ 1]^T$$

$$P_1 = [0 \ b \ 0 \ 1]^T$$

$$P_2 = [a \ b \ c \ 1]^T$$

และเรานำมาทำเป็นเมตริกซ์ของลูกบาศก์ก็ได้

$$[\text{cuboid}] = \begin{bmatrix} 0 & 0 & a & a & 0 & 0 & a & a \\ 0 & b & b & 0 & 0 & b & b & 0 \\ 0 & 0 & 0 & 0 & c & c & c & c \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

| | | | | | | |

$P_0 \ P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6 \ P_7$

พิจารณา transformer and rotation บนวัตถุ รูปแบบจะอธิบายโดยเมตริกซ์ 4x4 transformer ด้านการคำนวณ matrix T ในหัวข้อที่ผ่านมา ความสัมพันธ์ของตำแหน่งใหม่และตำแหน่งเริ่มแรกของวัตถุ คือ นำไปใช้

$$[\text{Object}]_n = T * [\text{Object}]_s$$

n = ตำแหน่งใหม่ (new position)

s = ตำแหน่งเริ่มแรก (starting position)

ในกรณีนี้ ตำแหน่งใหม่ หมายถึงโครงพิกัดพื้นฐานและ ตำแหน่งเริ่มแรก คือ ระบบพิกัดที่แสดงในการแสดงการแตกสมการดังนี้

$$\begin{bmatrix} x_{0,n} & x_{1,n} & - & - & - & x_{N-1,n} \\ y_{0,n} & y_{1,n} & - & - & - & y_{N-1,n} \\ z_{0,n} & z_{1,n} & - & - & - & z_{N-1,n} \\ 1 & 1 & - & - & - & 1 \end{bmatrix} \begin{bmatrix} R & | & k \\ - & - & - \\ 0 & 0 & 0 & | & 1 \end{bmatrix} * \begin{bmatrix} x_{0,s} & x_{1,s} & - & - & - & x_{N-1,s} \\ y_{0,s} & y_{1,s} & - & - & - & y_{N-1,s} \\ z_{0,s} & z_{1,s} & - & - & - & z_{N-1,s} \\ 1 & 1 & - & - & - & 1 \end{bmatrix}$$

เมตริกซ์ด้านซ้ายมือของสมการ แสดงจุดปลายของลูกบาศก์ตำแหน่งใหม่หลังจากการแปลงเมตริกซ์ทางด้านขวามือของสมการ แสดง transformer matrix และจุดปลายในโครงพิกัดเริ่มแรกตามลำดับ

2.3 ความสัมพันธ์ของตัวแปลงในเนื้อที่ว่างของแขนกล (Relative Transformer In the Robot Workspace)

ทางคณิตศาสตร์ชี้ให้เห็นถึงการแปลงของส่วนต่างๆของแขนกล พิจารณาเป็นลำดับของ homogeneous coordinate transformations เป้าหมายของการเคลื่อนที่มักจะถูกกำหนดให้อยู่ในเทอมของที่ตั้งในเนื้อที่ว่าง สิ่งสำคัญในส่วนนี้คือ manipulated ที่สัมพันธ์กับส่วนต่างๆ

จะเป็นประโยชน์อย่างมากในการใช้การแปลงสัมพันธ์นั้นคือ ถ้าใช้ในการทดลองความสัมพันธ์ของการแปลงพิกัดจะต้องทราบตำแหน่ง, มุม, ทิศทาง ของส่วนต่างๆของโครงสร้างถ้าใช้ในการเขียนโปรแกรมจะต้องจำเป็นที่จะต้องรู้ทุกส่วนประกอบ

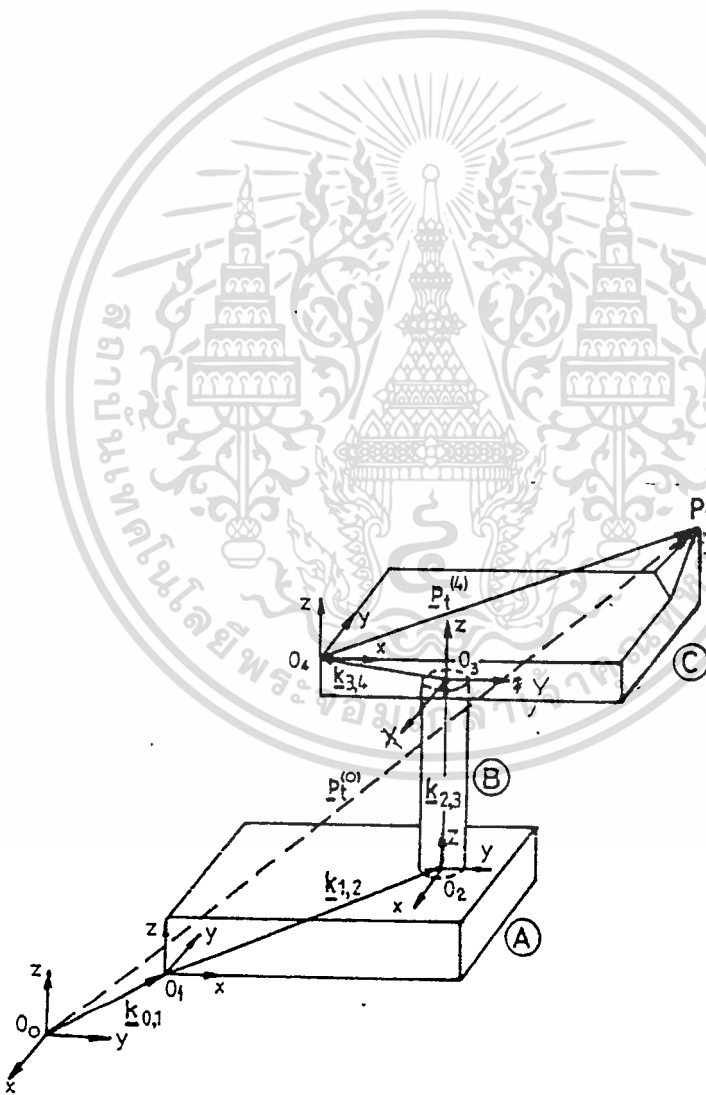
ตำแหน่งของจุดเป้าหมาย P_i ในระบบพิกัดฐานของจุดกำเนิด O_0 ถูกกำหนดโดยส่วนประกอบทางเรขาคณิต และตำแหน่งโดยสมบรูณ์ของจุดกำเนิด O_i ที่มีความสัมพันธ์ทั้งตำแหน่งและรูปลมมุติว่า เราต้องการที่จะเคลื่อนที่ส่วนประกอบของเครื่องมือไปยังเป้าหมาย P_i ในส่วนของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนูญาดเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C โดยที่เราทราบทิศทางทั้งหมดของ A,B,และC และที่ตั้งรวมทั้งมุมในเนื้อที่ว่าง เราสามารถหาตำแหน่งของจุด P_i เราสามารถหาพิกัดที่ได้โดยใช้ Homogeneous Transformation [4]

เพื่อที่จะแสดงวิธีการนี้เราจึงจะเข้าถึงจุด P_i ทีละขั้นจากจุดกำเนิด O_0 ก่อนที่จะทำ เรากำหนดระบบพิกัดย่อยให้แก่แต่ละส่วนของชิ้นส่วน จากข้อมูลที่เข้ามา เราใช้พิกัด P_i ในระบบพิกัดย่อย O_4 ซึ่งกำหนดให้เป็น

$$x_i^{(4)}, y_i^{(4)}, z_i^{(4)}$$



จุดมุ่งหมาย คือ อธิบายจุดของเป้าหมาย P_i ในระบบพิกัดฐานนั้นคือ เราต้องการกลุ่มของเวกเตอร์ $P_i^{(0)}$ พิกัดนั้นคือ

$$x_i^{(0)}, y_i^{(0)}, z_i^{(0)}$$

เมตริกซ์การแปลง (Transformation Matrix) ${}^i_n T$ อธิบายตำแหน่งและมุมของระบบพิกัดที่ n สัมพันธ์กับ i ซึ่งมีสูตรดังนี้ [9]

$${}^i_n T = \prod_{j=i+1}^{n-1} {}^j_n T = {}^i_{i+1} T * {}^{i+1}_{i+2} T * {}^{i+2}_{i+3} T * \dots * {}^{n-1}_n T$$

ที่ซึ่ง $i = 0, 1, 2, \dots, n-1$ เริ่มที่เลขใดๆ สำหรับ $i < n$ และ

$${}^i_n T = \left[\begin{array}{ccc|c} R_{i,i+1} & & & k_{i,i+1} \\ \hline & & & \\ 0 & 0 & 0 & 1 \end{array} \right]$$

ผลของการแปลงเมตริกซ์ ${}^i_n T$ อธิบายระบบพิกัดที่ n สัมพันธ์กับค่า i ใดๆที่เป็นส่วนประกอบของโครงพิกัด ที่คูณกันเป็นลำดับสำหรับ $i < n$ ดังนั้นพิกัดของเป้าหมาย P_i ในระบบพิกัดนั้นกับค่าส่วนประกอบโครงพิกัดใดๆ อธิบายได้เป็น

$$P_i^{(0)} = {}^i_n T * P_i^{(n)}$$

หรือ

$$\begin{bmatrix} x_i^{(i)} \\ y_i^{(i)} \\ z_i^{(i)} \\ 1 \end{bmatrix} = {}^i_n T * \begin{bmatrix} x_i^{(n)} \\ y_i^{(n)} \\ z_i^{(n)} \\ 1 \end{bmatrix}$$

จากตัวอย่างในรูปที่ 2.5 เวกเตอร์ $P_i^{(0)}$ และพิกัดถึงจุดเป้าหมาย P_i ให้ $i = 0$ และ $n = 4$ ในระบบพิกัดฐานได้ ${}^0_4 T$ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_t^{(0)} = {}^0_4T * P_t^{(4)}$$

หรือ

$$\begin{bmatrix} x_t^{(0)} \\ y_t^{(0)} \\ z_t^{(0)} \\ 1 \end{bmatrix} = {}^0_1T * {}^1_2T * {}^2_3T * {}^3_4T * \begin{bmatrix} x_t^{(4)} \\ y_t^{(4)} \\ z_t^{(4)} \\ 1 \end{bmatrix}$$

2.4 การแปลงในแขนกลหลายแกน

(Transformation Along The kinematics Chain)

แขนกลหลายแกนประกอบด้วยตัวเชื่อม (Link) ที่เชื่อมต่อกันด้วยจุดต่อ (Joint) แขนกลอาจจะมีส่วนของปลาย 2 ข้างเป็นรูปปริซึม (Prismatic joint) แขนกลหลายแกน n degree of freedom พิจารณาโดยที่มี m จุดต่อที่ต่อเชื่อมกัน n ตัวเชื่อม เมื่อนำมาประกอบกันเข้าก็จะกลายเป็นโซ่กล รูปแบบของตัวเชื่อมจะมีความสัมพันธ์กันระหว่างจุดต่อและจุดปลายแต่ละจุดปลายของตัวเชื่อม ในการที่จะอธิบายความสัมพันธ์ระหว่างส่วนต่างๆ ของแขนกล เราจะกำหนดระบบพิกัดขึ้นมาสำหรับแต่ละจุดต่อ ทุกๆตัวเชื่อมสามารถมีลักษณะของมันได้ 2 ลักษณะ [3]

1. ระยะ a_i คือระยะห่างของแกน z ที่ปลายทั้ง 2 ข้างของตัวเชื่อมตาม common normal (CN)

2. มุม α_i คือมุมที่เหลื่อจากการตั้งฉากกับเส้น a_i กับแนวแกน z ที่ลู่อเข้า

a_i มักจะเรียกว่า link length

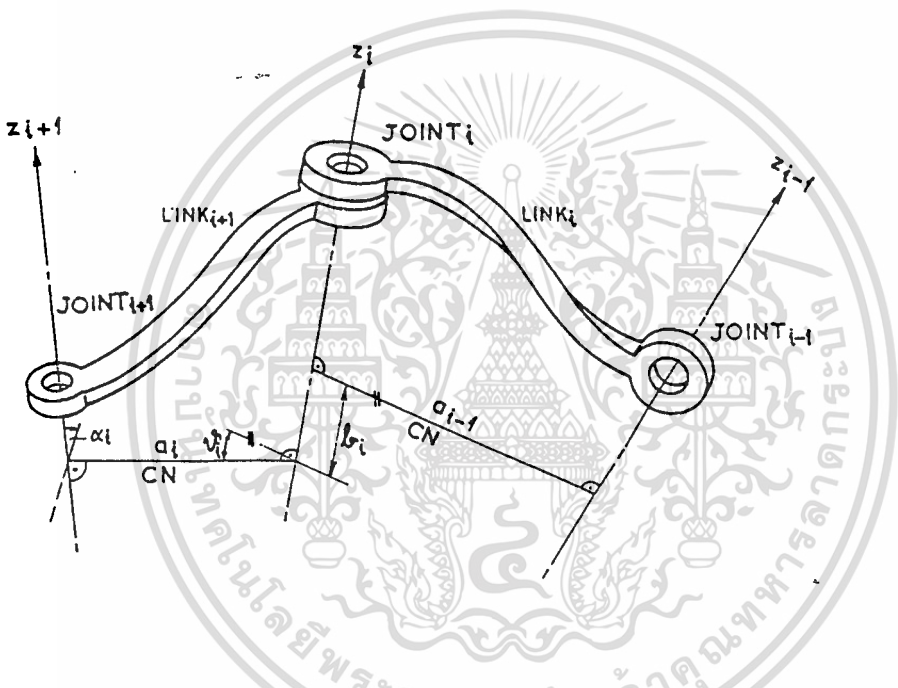
α_i เรียกว่า link twist angle

ในกรณีที่ไม่มีการลู่อเข้าของแนวแกน z, $\alpha_i = 0$ ทำให้เกิด common normals มากมาย

หลายเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดวางเป็นลำดับของตัวเชื่อม 2 ตัวเชื่อม จะต่อกันด้วยจุดต่อ ดังรูปที่ 2.5 ทุกๆแกนของจุดต่อ จะมี 2 common normal ก็จะเป็นเส้นของแต่ละตัวเชื่อมไป ความสัมพันธ์ทางตำแหน่งของตัวเชื่อม 2 ตัวนี้จะให้เป็น b_i คือระยะทางจากเส้น common normal ทั้งสองนั่นเอง และให้ θ_i คือมุมระหว่างเส้น common normal ทั้ง 2 ดังนั้น b_i และ θ_i คือตัวแปรของจุดต่อแต่ละจุดนั่นเอง



รูปที่ 2.5 แสดงจุดต่อของแขนกลและตัวแปรต่าง b_i = ระยะระหว่างจุดต่อ, θ_i = มุมจุดต่อ

ในโครงสร้างของแขนกลทำให้เราแยกได้ว่าระหว่าง revolute หรือ prismatic , revolute คือการหมุนของตัวเชื่อมระบบแกน และ θ_i , α_i คือตัวแปรของจุดต่อ prismatic คือการเคลื่อนที่ตามแนวแกน ดังนั้น a_i , b_i คือตัวแปร

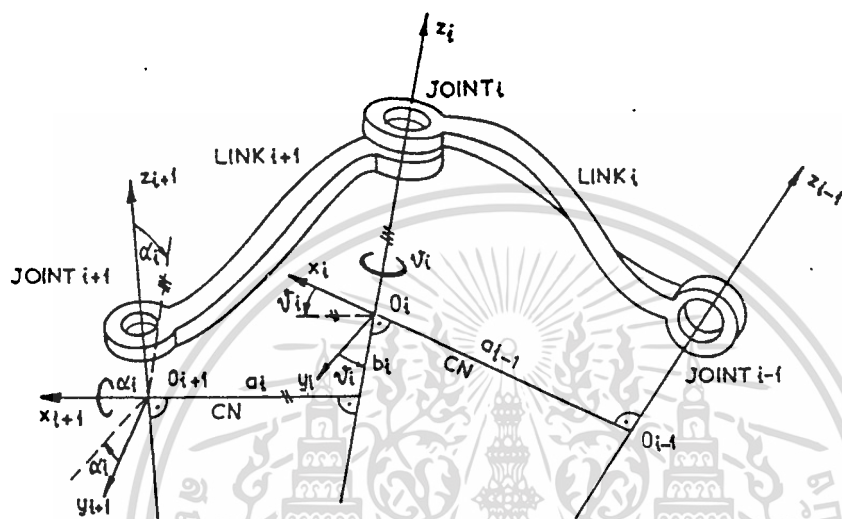
สำหรับการกำหนดระบบพิกัดของชิ้นส่วนแต่ละชิ้น เราจะใช้วิธีทางเรขาคณิตของ Denavit - Hartenberg วิธีนี้ระบบพิกัดจะถูกกำหนดให้เป็นลำดับของการหมุน 2 การหมุนและการเปลี่ยนตำแหน่ง 2 ตำแหน่ง เพื่อที่จะให้ระบบพิกัด i เข้ามาอยู่ตรงกับระบบพิกัด $i + 1$ ซึ่งดังแสดงในรูป 2.9 สำหรับจุดต่อ $i + 1$ และจุดต่อ i

พิจารณารูปที่ 2.6

- มุม θ_i หมุนรอบแกน z_i , $T(\theta_i)$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่แนะนำให้ไปทำตาม b_i เคลื่อนที่ไปตามแกน z_{i+1} , $T(b_i)$ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มุม a_i เคลื่อนที่ไปตามแกน x_{i+1} , $T(a_i)$
- มุม α_i หมุนรอบแกน x_{i+1} , $T(\alpha_i)$



รูปที่ 2.6 แสดงระบบพิกัดจุดต่อของแขนกล แต่ละจุดต่อจะมีกลุ่มของพิกัด คัดแปรจะกำหนดตามวิธีของ Denavit - Hartenberg

ให้ p_i และ p_{i+1} แสดงจุด p บน l และ $l+1$ ในระบบพิกัดจะได้

$$p_i = \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}_i$$

$$p_{i+1} = \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}_{i+1}$$

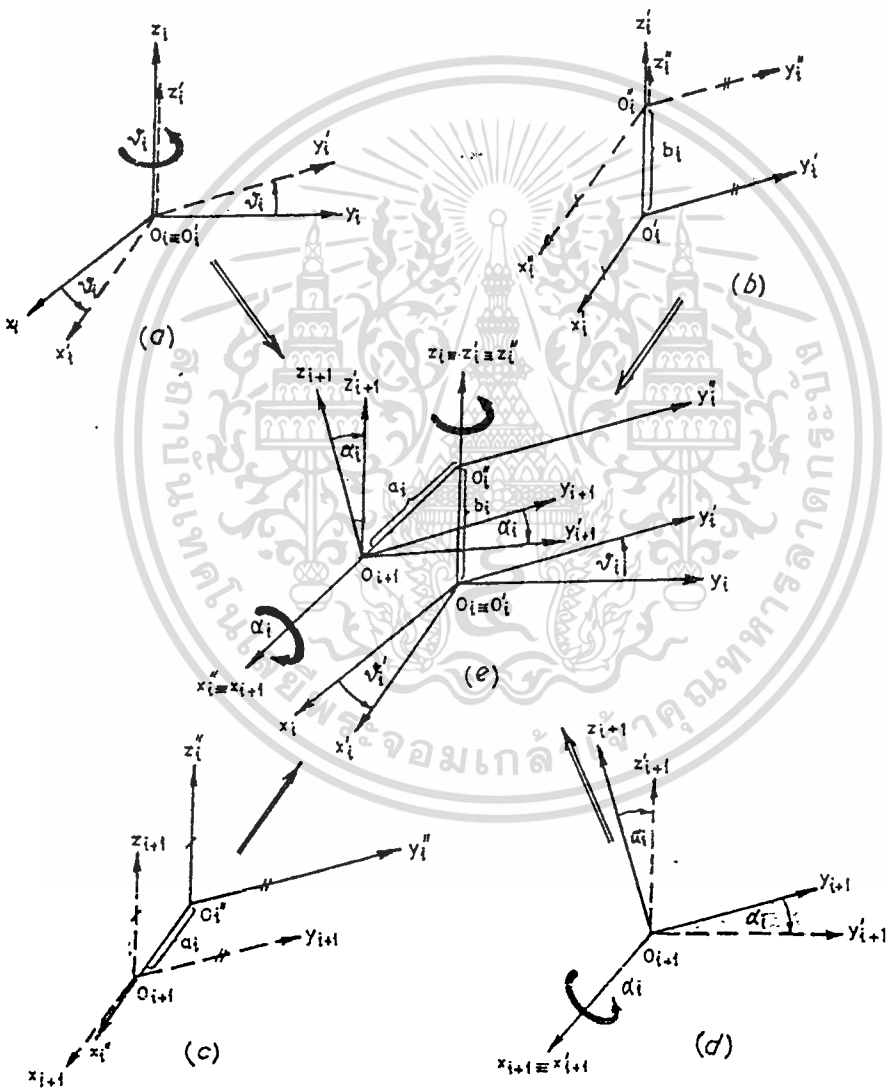
ความสัมพันธ์ระหว่างจุดต่อทั้ง 2 นั่นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_i = {}^0T_4 * P_{i+1}$$

${}^1T_{i+1}$ คือ Coordinate Transformation Matrix 4x4 เรามาพิจารณาทีละขั้นไปโดยใช้รูป

ที่ 2.7



รูปที่ 2.7 แสดงการแปลงทิกัด(Homogeneous coordinate transformation) (a) การหมุนของ θ_i รอบแกน z_i (b) คือระยะตาม แกนแกน z_i (c) คือระยะตามแนวแกน x_{i+1} (d) การหมุน α_i รอบแกน x_{i+1} (e) ผลของการแปลงระหว่าง joint i และ joint $i+1$ ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. The transformation matrix ของการหมุน θ_i รอบแกน z_i จะได้ว่า

$$T(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. The transformation matrix สำหรับการเคลื่อนที่ b_i ไปตามแกน z_i

$$T(b_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. The transformation matrix สำหรับการเคลื่อนที่ a_i ตามแกน x_{i+1}

$$T(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

4. The transformation matrix สำหรับการหมุน α_i รอบแกน x_{i+1}

$$T(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ผลของ Transformation matrix ระหว่างจุดต่อ 2 จุด ตามรูปที่ 2.12(e)

$${}_{i+1}^i T = T(\theta_i) * T(b_i) * T(a_i) * T(\alpha_i)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการคูณเราจะได้

$${}^{i+1}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & \cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

สมการ matrix ที่ได้คือ Overall transformation matrix ในรูปแบบทั่วไป ระหว่างจุดต่อ 2 จุดใดๆ ที่อยู่ข้างเคียงกัน i และ $i+1$ การหมุนเกิดขึ้นเพียงแกนเดียว จะเกิด θ_i หมุนรอบ z_i' สมมูลกับ z_i และ α_i รอบแกน x_i'' สมมูล x_{i+1} สำหรับคอลัมน์ที่ 4 ของ คือตำแหน่งของเวกเตอร์ k ระหว่างจุดกำเนิด O_i และ O_{i+1} [4]



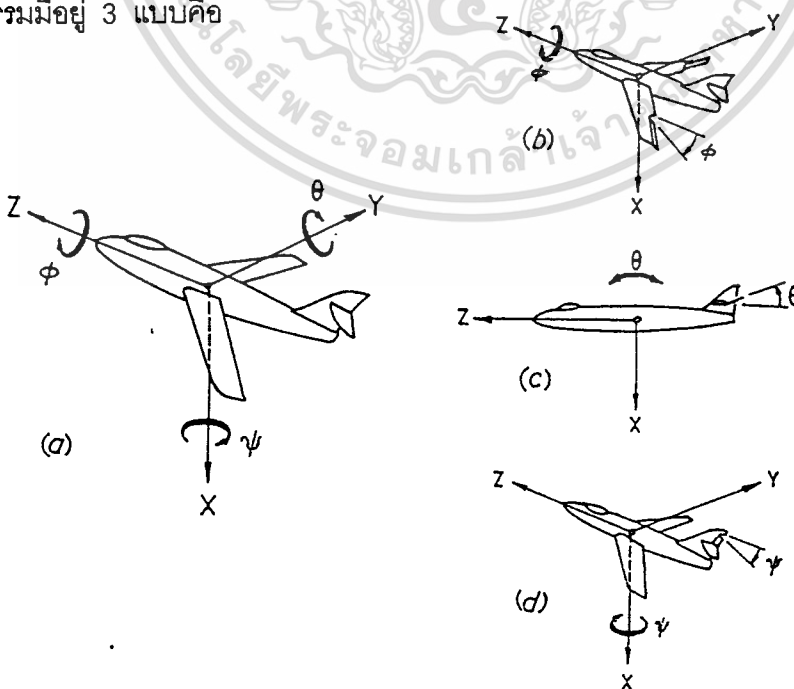
บทที่ 3

Coordinate System in Robot Application

3.1 การกำหนดมุมโดยอาศัยมุมของออยเลอร์ [3]

(Euler angles for specifying orientation)

การหมุนในระบบพิกัด สามารถอธิบายได้โดยเมตริกซ์ R คือ Submatrix 3 คูณ 3 ที่อยู่ด้านซ้ายมือบนของ homogeneous transformation matrix 4 คูณ 4 ในการประยุกต์ใช้กับแขนกลจะกำหนดมุมของวัตถุให้อยู่ในรูปของมุมออยเลอร์ (Euler Angles) มุมสัมพัทธ์ของระบบพิกัดมีส่วนเกี่ยวข้องกับมุมอื่นๆ ซึ่งสามารถอธิบายได้เป็นลำดับของการหมุนรอบแกน x,y,z ของระบบพิกัดอ้างอิง ในการหมุนนี้จะอยู่ในรูปของตัวแปรของมุม 3 ตัวแปร นั่นก็คือ Euler Angle นั่นเอง การเลือกใช้ขึ้นอยู่กับ การประยุกต์ใช้ตามปัญหาที่เกิดขึ้น ในปัญหาของเรา Euler Angle อธิบายอย่างแน่นอนในเทอมของการหมุนรอบแกน 3 แกน ภายใต้การแปลงนี้จะได้จำนวนรูปแบบของกลุ่มการหมุนที่มีตัวแปรค่อนข้างมาก แต่ลำดับการหมุนที่นิยมในการใช้กับแขนกลในอุตสาหกรรมมีอยู่ 3 แบบคือ



รูปที่ 3.1 แสดงการควบคุมเครื่องบิน (a) แสดงแกนหลัก x,y,z ของเครื่องบิน (b) ผลของมุม roll, ϕ ทำให้เกิด Aileron (c) ผลของมุม pitch, θ ทำให้เกิด Elevator (d) ผลของมุม yaw, ψ ทำให้เกิด Rudder

ไม่ว่าใครเห็นแต่ๆ ฟังสั้น อีกทีห้ามมีเหตุที่เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 roll-yaw-roll

3.1.2. roll-pitch-yaw

3.1.3. roll-pitch-roll

มีการหมุนอยู่ 3 ลักษณะเราจะอธิบายการหมุนให้โดยสมมุติกับเครื่องบิน หลังจากนั้นก็นำมาอธิบายโดยให้อยู่ในรูปของคณิตศาสตร์ การเคลื่อนที่ของเครื่องบินสามารถแสดงโดยการเคลื่อนที่รอบจุดศูนย์กลางโน้มถ่วง และการหมุนรอบจุดใดๆ กลุ่มของแกนคาเตเรียนและแกนหลัก แสดงตามรูป 3.1(a) แกนหลัก (Principle Axes) x, y, z และความเร็วเชิงมุมรอบแกน roll คือ ϕ แกน pitch คือ θ และแกน yaw คือ φ

มี 3 ระนาบหลักในการควบคุมการวางตัวของเครื่องบิน มี elevator, rudders และ ailerons ตามรูปที่ 3.1(b), (c), (d) ซึ่งลักษณะการเคลื่อนที่นี้คล้ายกับการเคลื่อนที่ของแขนของแขนกล ทำให้เข้าใจการเคลื่อนที่ของแขนกลได้ง่ายขึ้น

3.1.1 Roll - yaw - roll geometry

รูปแบบของ roll-yaw-roll

roll คือ การหมุนรอบแกน z

yaw คือ การหมุนรอบแกน x'

roll คือ การหมุนรอบแกน z''

พิจารณารูปที่ 3.2 เมตริกซ์ของ overall rotation $R_{RRR}(\alpha, \beta, \gamma)$ คือผลคูณของเมตริกซ์ของการหมุนของแต่ละมุม

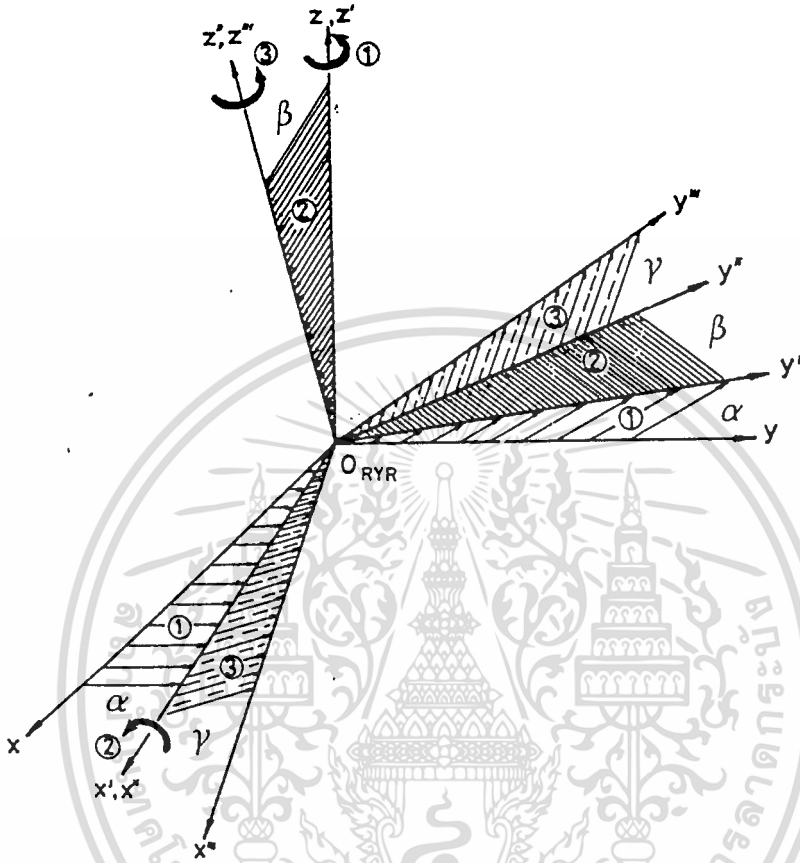
$$R_{RRR}(\alpha, \beta, \gamma) = R_z(\alpha) \cdot R_{x'}(\beta) \cdot R_{z''}(\gamma)$$

ที่ซึ่ง

α = หมุนรอบแกน z - roll

β = หมุนรอบแกน x' - yaw

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งาน = หมุนรอบแกน z'' - roll กรุณาอย่าให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 Euler angles (α, β, γ) สำหรับ roll - yaw - roll

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}_{roll}$$

$$R_x(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}_{yaw}$$

$$R_z(\chi\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}_{roll}$$

เพราะเป็นผลของการคูณจะได้เมตริกซ์ R จะได้

$$R_{RYR}(\alpha, \beta, \gamma) =$$

$$\begin{bmatrix} \cos\alpha \cos\gamma - \sin\alpha \cos\beta \sin\gamma & -\cos\alpha \sin\gamma - \sin\alpha \cos\beta \cos\gamma & \sin\alpha \sin\beta \\ \sin\alpha \cos\gamma + \cos\alpha \cos\beta \sin\gamma & -\sin\alpha \sin\gamma + \cos\alpha \cos\beta \cos\gamma & -\cos\alpha \sin\beta \\ \sin\beta \sin\gamma & \sin\beta \cos\gamma & \cos\beta \end{bmatrix}$$

ในการที่จะใช้ roll - yaw - roll geometry เมื่อ

1. เมื่อ $\beta = 0^\circ$ แกน z และ z'' วางตัวในแนวเดียวกันและขนานกันทั้งมุม α, γ ยิ่งวางตัวเหมือนเดิมดังนั้นสมการข้างต้นจะได้

$$R_{RYR}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos(\alpha + \gamma) & -\sin(\alpha + \gamma) & 0 \\ \sin(\alpha + \gamma) & \cos(\alpha + \gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\beta=0}$$

2. เมื่อ $\beta = 180^\circ$ แกน z และ z'' วางตัวอยู่ในแนวเดียวกันแต่อยู่ตรงกันข้ามกันดังนั้นสมการจะเป็น

$$R_{RYR}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos(\alpha - \gamma) & \sin(\alpha - \gamma) & 0 \\ \sin(\alpha - \gamma) & -\cos(\alpha - \gamma) & 0 \\ 0 & 0 & -1 \end{bmatrix}_{\beta=180}$$

เราสามารถที่จะจำกัดช่วง (0,180) เพราะว่าถ้าใช้ช่วงที่เกินพวกนี้แล้วก็ได้เมตริกซ์รูปแบบเดิม คือถ้าอยู่ในช่วง $(\alpha + 180, -\beta, \gamma + 180)$ จะได้เมตริกซ์ที่เหมือน $R_{RPY}(\alpha, \beta, \gamma)$ สำหรับ $-\beta$

3.1.2 Roll - pitch - yaw geometry

The roll - pitch - yaw geometry เป็นรูปแบบที่ใช้กันอย่างกว้างขวางมากในแขนงกล เพราะว่าการหมุนทั้ง 3 แกนในตำแหน่งอ้างอิงเป็นสิ่งที่ซับซ้อน roll คือการหมุนรอบแกน z , pitch คือการหมุนรอบแกน y' , yaw หมุนรอบแกน x'' ดังรูปที่ 3.3 $R_{RPY}(\phi, \theta, \varphi)$ คือ overall rotation matrix คือผลคูณของเมตริกซ์การหมุนของแต่ละหมุนตามแกนที่กำหนด

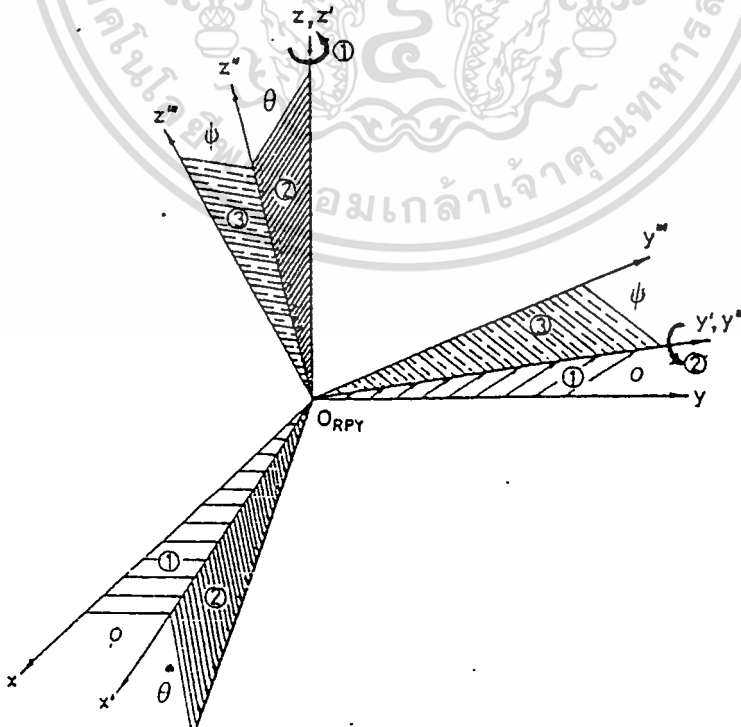
$$R_{RPY}(\phi, \theta, \varphi) = R_z(\phi) \cdot R_{y'}(\theta) \cdot R_{x''}(\varphi)$$

ที่ซึ่ง

ϕ = หมุนรอบแกน z - roll

θ = หมุนรอบแกน y' - pitch

φ = หมุนรอบแกน x'' - yaw



เมตริกซ์การหมุนของแต่ละมุมรอบแกนแสดงดังนี้

$$R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}_{roll}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}_{pitch}$$

$$R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}_{yaw}$$

ผลคูณของเมตริกซ์คือ

$$R_{RPY}(\phi, \theta, \varphi) = \begin{bmatrix} \cos\phi \cos\theta & \cos\phi \sin\theta \sin\varphi - \sin\phi \cos\varphi & \cos\phi \sin\theta \cos\varphi + \sin\phi \sin\varphi \\ \sin\phi \cos\theta & \sin\phi \sin\theta \sin\varphi + \cos\phi \cos\varphi & \sin\phi \sin\theta \cos\varphi - \cos\phi \sin\varphi \\ -\sin\theta & \cos\theta \sin\varphi & \cos\theta \cos\varphi \end{bmatrix}$$

ลักษณะพิเศษของ roll - pitch - yaw geometry ทำให้เมตริกซ์การหมุน R ใช้การไม่ได้เมื่อ $\theta = +90^\circ$ หรือ $\theta = -90^\circ$

1. เมื่อ $\theta = +90^\circ$ แกน x'' วางตัวในแนวเดียวกับ แกน z แต่ตรงกันข้ามกับมุม ϕ, φ จะเป็นมุมเดียวกัน

$$R_{RPY}(\phi, \theta, \varphi) = \begin{bmatrix} 0 & \sin(\varphi - \phi) & \cos(\varphi - \phi) \\ 0 & \cos(\varphi - \phi) & -\sin(\varphi - \phi) \\ -1 & 0 & 0 \end{bmatrix}_{\theta = 90^\circ}$$

2. เมื่อ $\theta = -90$ แกน x'' และแกน z วางตัวแนวเดียวกันและขนานกัน

$$R_{RPY}(\phi, \theta, \varphi) = \begin{bmatrix} 0 & -\sin(\varphi + \phi) & -\cos(\varphi + \phi) \\ 0 & \cos(\varphi + \phi) & -\sin(\varphi + \phi) \\ 1 & 0 & 0 \end{bmatrix}_{\theta = -90^\circ}$$

ดังนั้นเราจึงจำกัดช่วง $(-90, 90)$ เพราะว่าถ้าเกินช่วงนี้ไปเมตริกซ์การหมุนที่ได้ก็จะมีรูปแบบเดิม คือถ้าอยู่ในช่วง $(\phi - 90, -\theta, \varphi + 90)$ ก็เหมือนกับ R_{RPY} สำหรับ $-\theta$

3.1.3 Roll - pitch - roll geometric

รูปแบบของ roll - pitch - roll , roll จะหมุนรอบแกน z , pitch หมุนรอบแกน y' roll หมุนรอบแกน z'' อีกครั้ง ดังในรูปที่ 3.4

เมตริกซ์ $R_{RPR}(\partial, \lambda, \sigma)$ คือผลคูณของเมตริกซ์การหมุนของมุมในแต่ละแกน

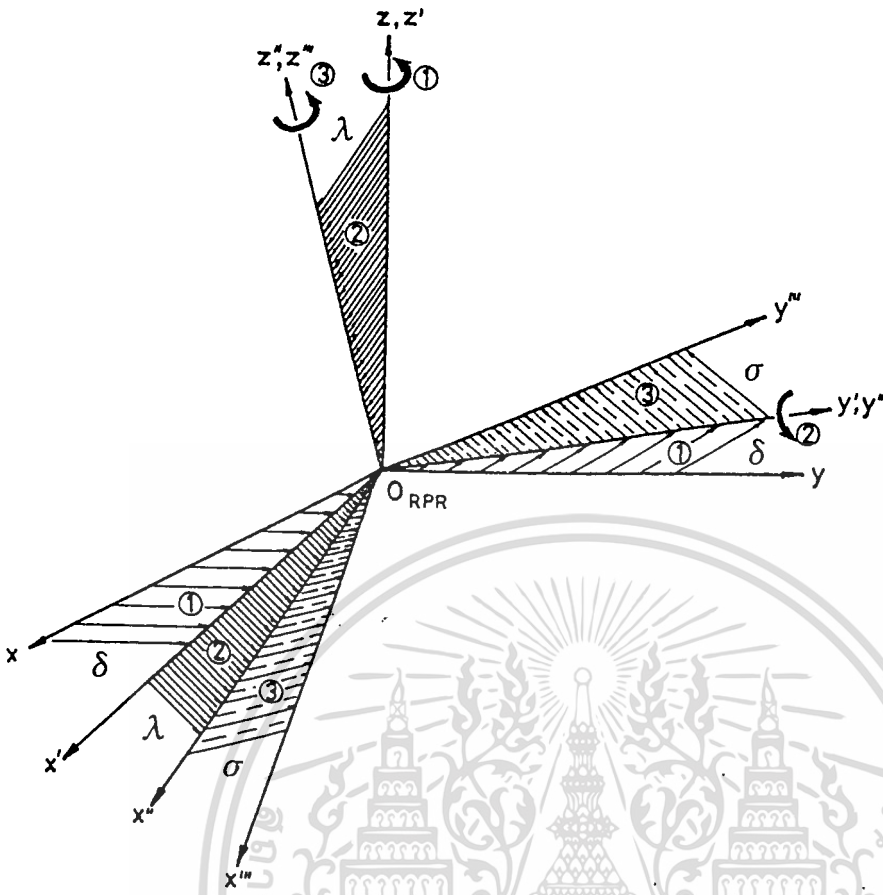
$$R_{RPR}(\partial, \lambda, \sigma) = R_z(\partial) \cdot R_{y'}(\lambda) \cdot R_{z''}(\sigma)$$

เมื่อ ∂ = หมุนรอบแกน z - roll
 λ = หมุนรอบแกน y' - pitch
 σ = หมุนรอบแกน z'' - roll

สำหรับเมตริกซ์การหมุนของแต่ละมุมคือ

$$R_z(\partial) = \begin{bmatrix} \cos \partial & -\sin \partial & 0 \\ \sin \partial & \cos \partial & 0 \\ 0 & 0 & 1 \end{bmatrix}_{roll}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 Euler angle (δ, λ, σ) สำหรับ roll - pitch - roll

$$R_y(\lambda) = \begin{bmatrix} \cos \lambda & 0 & \sin \lambda \\ 0 & 1 & 0 \\ -\sin \lambda & 0 & \cos \lambda \end{bmatrix}_{pitch}$$

$$R_z(\sigma) = \begin{bmatrix} \cos \sigma & -\sin \sigma & 0 \\ \sin \sigma & \cos \sigma & 0 \\ 0 & 0 & 1 \end{bmatrix}_{roll}$$

ผลของการคูณจะได้

$$R_{RPR}(\delta, \lambda, \sigma) =$$

$$\begin{bmatrix} \cos \delta \cos \lambda \cos \sigma - \sin \delta \sin \sigma & -\cos \delta \cos \lambda \sin \sigma - \sin \delta \cos \sigma & \cos \delta \sin \lambda \\ \sin \delta \cos \lambda \cos \sigma + \cos \delta \sin \sigma & -\sin \delta \cos \lambda \sin \sigma + \cos \delta \cos \sigma & \sin \delta \sin \lambda \\ -\sin \lambda \cos \sigma & \sin \lambda \sin \sigma & \cos \lambda \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะเฉพาะของ roll-pitch-roll geometric ที่ทำให้เมตริกซ์การหมุน $R_{RPR}(\vartheta, \lambda, \sigma)$ ใช้การไม่ได้เมื่อ $\lambda = 0^\circ$ หรือ $\lambda = 180^\circ$

1. เมื่อ $\lambda = 0^\circ$ และ σ และ ϑ หมุนอยู่ในแกนเดียวกัน ได้เมตริกซ์ออกมา

$$R_{RPR}(\vartheta, \lambda, \sigma) = \begin{bmatrix} \cos(\vartheta + \sigma) & -\sin(\vartheta + \sigma) & 0 \\ \sin(\vartheta + \sigma) & \cos(\vartheta + \sigma) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\lambda=0^\circ}$$

2. เมื่อ $\lambda = 180^\circ$ ทำให้ σ และ ϑ หมุนอยู่ในแกนเดียวกัน

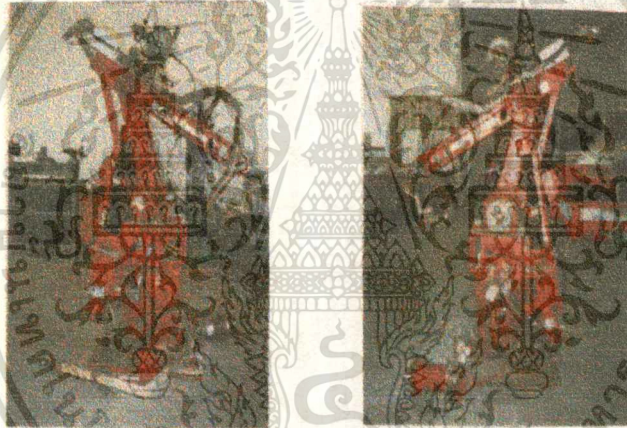
$$R_{RPR}(\vartheta, \lambda, \sigma) = \begin{bmatrix} -\cos(\vartheta - \sigma) & -\sin(\vartheta - \sigma) & 0 \\ -\sin(\vartheta - \sigma) & \cos(\vartheta - \sigma) & 0 \\ 0 & 0 & -1 \end{bmatrix}_{\lambda=180^\circ}$$

บทที่ 4

การคำนวณสำหรับ Motoman L-3

4.1 Forward Kinematics [4]

The Yasukawa Motoman L-3 เป็นแขนกล 5 แกน ที่ได้รับความนิยมในการใช้ในอุตสาหกรรมแต่เป็นแขนกลที่ไม่ได้ใช้โซ่กลแบบเปิดเหมือนแขนกลทั่วไปแต่ใช้ตัวขับเคลื่อนต่อโดยตรงกับตัวเชื่อม 2, 3 และยังใช้โซ่ขับเคลื่อนต่อ 4, 5 ด้วย

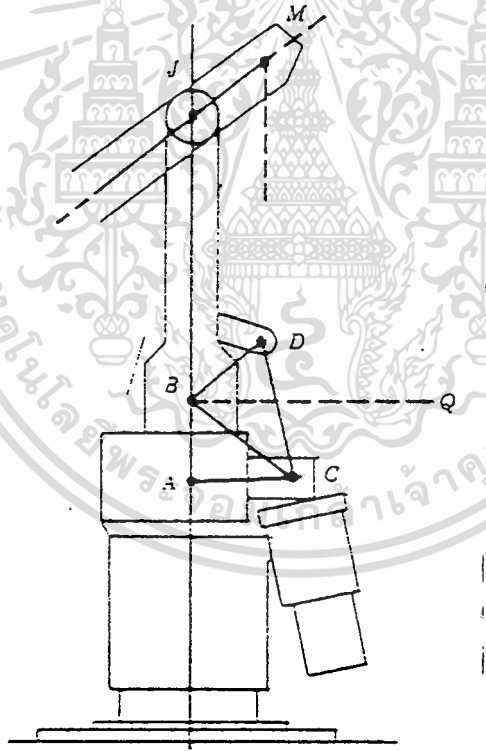


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับปีที่ 4.1 แสดง The Yasukawa Motoman L-3 อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแก้ปัญหาเราจะทำเป็น 2 ขั้นตอน

- ขั้นตอนแรก แก้ปัญหาสำหรับมุมของจุดต่อจากตำแหน่งกระตุ้น
- ขั้นตอนที่สอง เราจะแก้ปัญหาสำหรับตำแหน่งทางคาเตเซียน และ โอเรียนเตชัน ของตัวเชื่อมตัวสุดท้ายจากมุมของจุดต่อที่หาได้จากขั้นตอนแรก

จากรูปที่ 4.2 แสดงตัวเชื่อมทาง mechanism ซึ่งติดต่อกับตัวขับที่ 2 ซึ่งเชื่อมกับตัวเชื่อม 2,3 ของแขนกล เป็นตัวขับทางเส้นตรงซึ่งควบคุมความยาวของเส้นตรง DC โดยตรง สามเหลี่ยม ABC อยู่กับที่ ขณะที่เส้นตรง BD มีความยาวเท่าเดิม จุดต่อ 2 หมุนรอบจุด B และตัวเลื่อนกระตุ้นหมุนรอบจุด C ในขณะที่ตัวเชื่อมเคลื่อนที่ ดังนั้นเราก็สามารถกำหนดสัญลักษณ์ขึ้นมาสำหรับตัวขับ 2



รูปที่ 4.2 แสดงรายละเอียดทางด้าน Kinematics ของตัวเชื่อมที่ 2 และตัวกระตุ้นที่ 2 ของ Yasukawa

สำหรับค่าคงที่

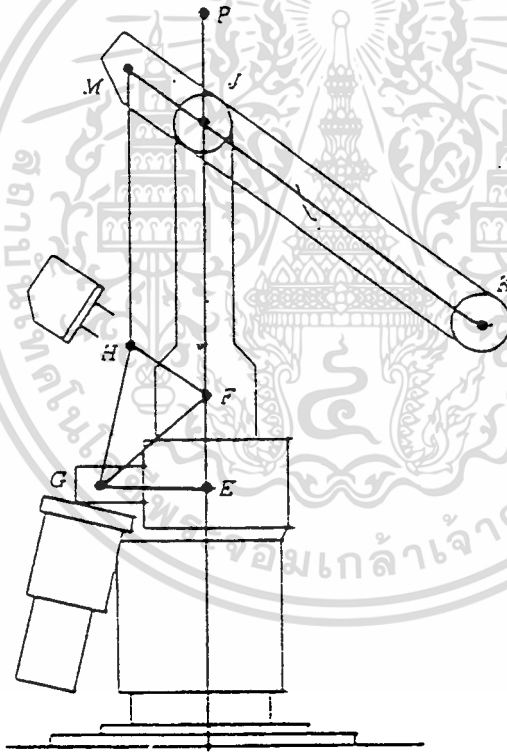
$$\gamma_2 = AB \quad \phi_2 = AC \quad \alpha_2 = BC$$

เอกสารนี้เป็นเอกสาร $\beta_2 = BD$ หารับการใช้อย่าง $\Omega_2 = \hat{JBD}$ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับค่าตัวแปร

$$\theta_2 = -\hat{J}\hat{B}Q \quad \phi_2 = \hat{C}\hat{B}D \quad g_2 = DC$$

จากรูปที่ 4.3 แสดงตัวขับซึ่งติดต่อกับตัวขับที่ 3 ซึ่งเชื่อมกับตัวเชื่อม 2, 3 ของแขนกล ตัวขับเป็นตัวขับทางเส้นตรงซึ่งควบคุมความยาวของส่วนของเส้นตรง HG โดยตรง สามเหลี่ยม EFG อยู่กับที่ ขณะที่เส้นตรง FH มีความยาวเท่าเดิม จุดต่อ 3 หมุนรอบจุด J และ ตัวเลื่อนหมุนรอบจุด G ขณะที่ตัวเชื่อมเคลื่อนที่ดังนั้นเราก็กำหนดสัญลักษณ์ต่างๆขึ้นมาสำหรับ ตัวขับ 3



รูปที่ 4.3 แสดงรายละเอียดทาง kinematics ของตัวเชื่อมที่ 3 และตัวกระตุ้นที่ 3 ของ Yasukawa

สำหรับค่าคงที่

$$\begin{aligned} \gamma_3 &= EF & \phi_3 &= EG & \alpha_3 &= GF \\ \beta_3 &= HF & l_3 &= JK & & \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับค่าตัวแปร

$$\theta_3 = PJK \quad \varphi_3 = GFH \quad g_3 = GH$$

การจัดวางตัวของตัวขับ และตัวเชื่อมมีเงื่อนไขตามมา ผลก็คือ ตัวขับที่ 2 ใช้ในตำแหน่งจุดต่อ 2 ขณะที่เคลื่อนที่ไป ตัวเชื่อม 3 ก็ยังคงมีมุมที่สัมพันธ์กับฐานของแขนกล เหมือนเดิมตัวขับที่ 3 ใช้สำหรับปรับมุมของตัวเชื่อม 3 สัมพันธ์กับฐานของแขนกล เหตุผลหนึ่งที่มีการจัดวางตัวเชื่อมแบบนี้ก็เพื่อเพิ่มความแข็งแรงทางโครงสร้างของตัวเชื่อมหลักของแขนกลและเพื่อนำมาใช้ในการเพิ่มความสามารถในการเข้าถึงตำแหน่งได้แม่นยำขึ้น

ตัวขับสำหรับตัวเชื่อม 4 และ 5 จะอยู่ที่ตัวเชื่อม 1 ของแขนกลซึ่งวางตัวในแนวแกนของจุดต่อ 2 (จุด B และ F) และการขับจะส่งผ่านโซ่ซึ่งมีอยู่ 2 ส่วน

- ส่วนแรก โซ่จะวางอยู่ในตัวเชื่อม 2
- ส่วนที่สอง วางตัวในตัวเชื่อม 3

ผลของการส่งผ่านด้วยระบบนี้มีเงื่อนไขตามมา : ตัวขับ 4 ซึ่งใช้กำหนดตำแหน่งจุดต่อ 4 จะสัมพันธ์กับฐานของแขนกล นอกจากจะสัมพันธ์กับตัวเชื่อมก่อนหน้านี้คือตัวเชื่อม 3 อีกด้วยหมายความว่าถ้าให้ตัวขับ 4 คงที่ก็จะรักษาตัวเชื่อมที่ 4 ให้คงที่ เมื่อเทียบกับฐานของแขนกล โดยไม่ต้องคำนึงถึงจุดต่อที่ 2,3 และตัวขับ 5 ก็เหมือนกันก็จะเชื่อมโยงจุดต่อ 5 โดยตรง

เราสามารถที่จะกำหนดสมการซึ่งใส่ค่าของตัวขับ (A_i) เพื่อจะได้สมการสมมูลของ ค่าที่จุดต่อ (θ_i) ในกรณีนี้ สมการที่ได้ส่วนใหญ่มาจากกฎของ โคไซน์ ตัวแปร K_i และค่าออฟเซต π_i ที่ปรากฏในสมการมีค่าคงที่สำหรับแต่ละตัวขับ ยกตัวอย่างเช่น ตัวขับ 1 ต่อโดยตรงกับจุดต่อที่แกน 1 ดังนั้นรูปสมการจึงอยู่ในรูปร่างง่าย คือมีเพียงค่าของ K_1 คูณกับตัวขับ A_1 บวกกับค่าออฟเซต π_1 ดังนั้น

$$\theta_1 = k_1 A_1 + \lambda_1$$

$$\theta_2 = \cos^{-1} \left(\frac{(k_2 A_2 + \lambda_2)^2 - \alpha_2^2 - \beta_2^2}{-2\alpha_2 \beta_2} \right) + \tan^{-1} \left(\frac{\phi_2}{\gamma_2} \right) + \Omega_2 - 270^\circ$$

$$\theta_3 = \cos^{-1} \left(\frac{(k_3 A_3 + \lambda_3)^2 - \alpha_3^2 - \beta_3^2}{-2\alpha_3 \beta_3} \right) - \theta_2 + \tan^{-1} \left(\frac{\phi_3}{\gamma_3} \right) - 90^\circ$$

$$\theta_4 = -k_4 A_4 - \theta_2 - \theta_3 + \lambda_4 + 180^\circ$$

$$\theta_5 = -k_5 A_5 + \lambda_5$$

(ถ้า มุมของสามเหลี่ยมมีตัวอักษร a,b,c, ที่ซึ่งมุม a ตรงกันข้ามกับความยาว A เพราะฉะนั้น $A^2 = B^2 + C^2 - 2BC \cos a$)



รูปที่ 4.4 แสดงการกำหนดระบบพิกัดสำหรับตัวเชื่อม ของ Yasukawa

รูปที่ 4.4 แสดงขอบเขตการเข้าถึงของตัวเชื่อม ในรูปแสดงการตอบสนองตำแหน่งไปยัง joint vector $\theta = (0, -90, 90, 90, 0)$ รูปที่ 4.5 แสดงค่าของตัวเชื่อมสำหรับ manipulator

Matrix Transformation ของตัวเชื่อม

$${}^0_1T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$${}^1_2T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3_4T = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & l_3 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4_5T = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90°	0	0	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	θ_4
5	90°	0	0	θ_5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.5 แสดง Link parameters ของ Yasukawa manipulator
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะได้ว่า

$${}^0_5T = {}^0_1T * {}^1_2T * {}^2_3T * {}^3_4T * {}^4_5T$$

หรือ

$${}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ที่ซึ่ง

$$r_{11} = \cos\theta_1 \cos(\theta_2 + \theta_3 + \theta_4) \cos\theta_5 - \sin\theta_1 \sin\theta_5$$

$$r_{21} = \sin\theta_1 \cos(\theta_2 + \theta_3 + \theta_4) \cos\theta_5 + \cos\theta_1 \sin\theta_5$$

$$r_{31} = -\sin(\theta_2 + \theta_3 + \theta_4) \cos\theta_5$$

$$r_{12} = -\cos\theta_1 \cos(\theta_2 + \theta_3 + \theta_4) \sin\theta_5 - \sin\theta_1 \cos\theta_5$$

$$r_{22} = -\sin\theta_1 \cos(\theta_2 + \theta_3 + \theta_4) \sin\theta_5 + \cos\theta_1 \sin\theta_5$$

$$r_{32} = \sin(\theta_2 + \theta_3 + \theta_4) \sin\theta_5$$

$$r_{13} = \cos\theta_1 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$r_{23} = \sin\theta_1 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$r_{33} = \cos(\theta_2 + \theta_3 + \theta_4)$$

$$p_x = \cos\theta_1 (l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3))$$

$$p_y = \sin\theta_1 (l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3))$$

$$p_z = -l_2 \sin\theta_2 - l_3 \sin(\theta_2 + \theta_3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 Inverse Kinematics [4]

ในการหาคำตอบนี้สามารถแบ่งได้เป็น 2 ส่วนด้วยกัน

-ในส่วนของพีชคณิต (Algebra)

-ในส่วนของเรขาคณิต (Geometry)

Motoman L-3 มีลักษณะที่แตกต่างจากแขนกลทั่วไป ในการหา inverse kinematics อยู่ 3 ประการคือ

1. เพราะว่ามีจุดต่อเพียง 5 จุดจึงไม่สามารถที่จะเอา จุดปลายไปยังตำแหน่ง และ มุมต่างๆ ไปได้ทั้งหมด (General Goal)
2. เพราะว่าเป็น the four-bar type ของตัวเชื่อมและการขับเคลื่อนที่ของ ตัวขับเพียง 1 ตัว สามารถทำให้จุดต่อ 2-3 จุดเคลื่อนที่ไปได้
3. ขีดจำกัดของตำแหน่งตัวขับจะไม่คงที่ แต่จะขึ้นอยู่กับตำแหน่งของตัวขับ ตัวอื่นๆ ในการที่จะตัดสินใจว่ากลุ่มการคำนวณของค่าตัวขับอยู่ในพิสัยหรือไม่จึง เป็นการที่ไม่ควรละเลย

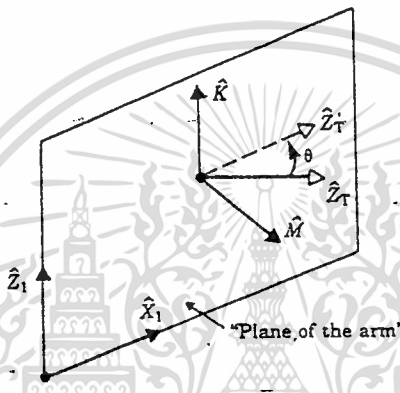
ถ้าเราพิจารณาถึงธรรมชาติของ เนื้อที่ย่อย (Subspace) ของ Motoman manipulator เราสามารถเห็นได้อย่างรวดเร็วว่า เนื้อที่ย่อยนี้สามารถที่จะบอกโดยว่ามีเพียงมุมที่จะทำได้เพียง 1 มุมบังคับ การบอกทิศทางของเครื่องมือ นั่นก็คือแกน Z_T ซึ่งจะวางตัวอยู่ใน "ระนาบของแขน" (Plane of the arm) ระนาบนี้คือ ระนาบในแนวตั้งซึ่งมีแกนของจุดต่อที่ 1 อยู่ และจุดที่ซึ่งแกน 4 และ 5 ตัดกัน การหมุนที่ใกล้ที่สุดจากจุดเดิมหมุนไปยังจุดใดๆ มีเพียง 1 ทาง เพื่อที่จะให้เกิดการ หมุนบนระนาบมีน้อยที่สุด ถ้าไม่มีการเปลี่ยนแปลงอย่างแน่นอนในการอ้างอิง ถึงเนื้อที่ย่อยแล้ว เราสามารถที่จะหาวิธีการแตกระนาบที่ต้องการ (General goal frame) เข้าไปในเนื้อที่ย่อยได้ แต่ที่ กล่าวมาทั้งหมดนี้ใช้ได้เฉพาะกรณีที่ wrist frame และ tool frame แตกต่างกันเพียงแค่การแค้มใน แนวแกน Z_w เท่านั้น

รูปที่ 4.6 แสดงให้เห็นระนาบของแขนตามปกติ \hat{M} และทิศทางของจุดปลายของ

อุปกรณ์คือ \hat{Z}_T ทิศทางของจุดปลายนี้ (pointing direction) จะต้องหมุนไปเป็นมุม θ รอบ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกขาดให้เป็นลิขสิทธิ์ของตนเอง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวกเตอร์ \hat{K} เป็นเหตุให้เกิดทิศทางของจุดปลายอันใหม่คือ Z_T' วางตัวอยู่ในระนาบ นั่นคือ
 เวกเตอร์ \hat{K} ซึ่งจะทำให้เกิดมุม θ น้อยสุดที่วางตัวอยู่ในระนาบแล้วยังต้องตั้งฉากกับแกน Z_T
 และ Z_T' ด้วย



รูปที่ 4.6 แสดงการหมุนของระนาบเป้าหมายในเนื้อที่ย่อย ของ Yasukawa

สำหรับขอบเขตเป้าหมาย (goal frame) ที่ต้องการ, เวกเตอร์ \hat{M} กำหนดเป็น

$$\hat{M} = \frac{1}{\sqrt{p_x^2 + p_y^2}} \begin{bmatrix} -p_x \\ p_y \\ 0 \end{bmatrix} \quad (1)$$

ที่ซึ่ง p_x และ p_y คือตำแหน่งของอุปกรณ์ที่ต้องการ

เมื่อ K กำหนดให้เป็น

$$K = \hat{M} \times \hat{Z}_T \quad (2)$$

และ Z_T' คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ \hat{A} การ, \hat{C} \hat{A} การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 $Z_T' = K \times \hat{M}$ (3)
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดของมุม θ คือ

$$\cos\theta = \hat{Z}_T \circ \hat{Z}_T' \quad (4)$$

จาก Rodrigues 's formula เราจะได้

$$\hat{Y}_T' = \cos\theta \hat{Y}_T + \sin\theta \left(\hat{K} \times \hat{Y}_T \right) + (1 - \cos\theta) \left(\hat{K} \cdot \hat{Y}_T \right) \hat{K} \quad (5)$$

ในที่สุดเราสามารถที่จะคำนวณหาคอสมน์ไม่ทราบค่าที่เหลือของ rotation matrix ใหม่ในอุปกรณ์ คือ

$$\hat{X}_T' = \hat{Y}_T' \times \hat{Z}_T' \quad (6)$$

สมการที่ (1) ถึงสมการ (6) อธิบายวิธีการแตกแกนให้อยู่ในมุมที่ต้องการใดๆ (General Goal Orientation) ในเนื้อที่ย่อย (Subspace) ของ Motoman robot

เราสมมุติให้ว่า B_T คือ wrist frame วางตัวอยู่ในเนื้อที่ย่อย เราสามารถหาสมการ kinematics ได้ซึ่งสมการที่จะอยู่ในรูปของผลคูณของ link Transformations

$${}^0_5T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T$$

แต่เรามี

$${}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

เอา ${}^0_1T^{-1}$ คูณทั้งสองข้างของสมการ

$${}^0_1T^{-1} {}^0_5T = {}^1_2T {}^2_3T {}^3_4T {}^4_5T$$

ทางด้านซ้ายของสมการคือ

$$\begin{bmatrix} c\theta_1 \cdot r_{11} + s\theta_1 \cdot r_{21} & c\theta_1 \cdot r_{12} + s\theta_1 \cdot r_{22} & c\theta_1 \cdot r_{13} + s\theta_1 \cdot r_{23} & c\theta_1 \cdot p_x + s\theta_1 \cdot p_y \\ -s\theta_1 \cdot r_{11} + c\theta_1 \cdot r_{21} & -s\theta_1 \cdot r_{12} + c\theta_1 \cdot r_{22} & -s\theta_1 \cdot r_{13} + c\theta_1 \cdot r_{23} & -s\theta_1 \cdot p_x + c\theta_1 \cdot p_y \\ -r_{31} & -r_{32} & -r_{33} & -p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ทางด้านขวามือของสมการคือ

$$\begin{bmatrix} * & * & \sin(\theta_2 + \theta_3 + \theta_4) & * \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ * & * & -\cos(\theta_2 + \theta_3 + \theta_4) & * \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(* คือจำนวนตัวแปร ซึ่งไม่ได้แสดงในที่นี้)

พิจารณาตำแหน่ง (3,4) เราจะได้

$$-\sin\theta_1 \cdot p_x + \cos\theta_1 \cdot p_y = 0$$

เราจะได้

$$\theta_1 = \tan^{-1} 2 \left(\frac{p_y}{p_x} \right)$$

พิจารณาตำแหน่ง (3,1) และ (3,2) จะได้

$$\sin\theta_5 = -\sin\theta_1 \cdot r_{11} + \cos\theta_1 \cdot r_{21}$$

$$\cos\theta_5 = -\sin\theta_1 \cdot r_{12} + \cos\theta_1 \cdot r_{22}$$

เพราะฉะนั้นเราจะคำนวณได้

$$\theta_5 = \tan^{-1} 2 \left(\frac{r_{21} \cos\theta_1 - r_{11} \sin\theta_1}{r_{22} \cos\theta_1 - r_{12} \sin\theta_1} \right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาตำแหน่ง (2,3) และ (1,3) จะได้

$$\cos(\theta_2 + \theta_3 + \theta_4) = r_{33}$$

$$\sin(\theta_2 + \theta_3 + \theta_4) = \cos\theta_1 \cdot r_{13} + \sin\theta_1 \cdot r_{23}$$

ทำให้เราได้

$$(\theta_2 + \theta_3 + \theta_4) = \tan^{-1} 2 \left(\frac{r_{13} \cos\theta_1 + r_{23} \sin\theta_1}{r_{33}} \right)$$

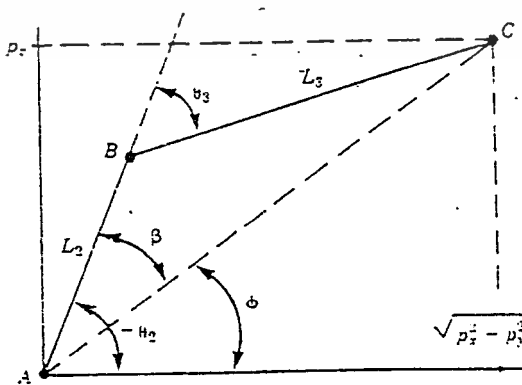
ในการหาค่ามุม $\theta_2, \theta_3, \theta_4$ แต่ละค่าที่เราจะหาได้ทางเรขาคณิต รูปที่ 4.7 แสดงระนาบของแขนให้จุด A คือจุดต่อของแกน 2, จุด B คือจุดต่อของแกน 3 และจุด C เป็นจุดต่อของแกนที่ 4

จากกฎของโคไซน์ เมื่อเราพิจารณาสามเหลี่ยม ABC จะได้

$$\cos(\theta_3) = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2l_2 l_3}$$

และ

$$\theta_3 = \tan^{-1} 2 \left(\frac{\sqrt{1 - \cos^2 \theta_3}}{\cos \theta_3} \right)$$



เอกสารนี้เป็นเอกสารที่สงวนรูปที่ 4.7 แสดงรูปสมมูลของแขนกล ที่ใช้คำนวณมุม ของ link 2, 3 ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.10 เราเห็นได้ชัดว่า $\theta_2 = -\phi - \beta$ หรือ

$$\theta_2 = \tan^{-1} 2 \left(\frac{p_z}{\sqrt{p_x^2 + p_y^2}} \right) - \tan^{-1} 2 \left(\frac{l_3 \sin \theta_3}{l_2 + l_3 \cos \theta_3} \right)$$

ในสุดท้ายเราจะได้

$$\theta_4 = (\theta_2 + \theta_3 + \theta_4) - \theta_2 - \theta_3$$

ในการหามุมของจุดต่อแต่ละจุดเราจำเป็นต้องแสดงค่าตัวกระตุ้น (Actuator Value) ชุดต่อไปในการหาค่า forward kinematics เราจะคำนวณหาค่า A_i ได้ดังนี้

$$A_1 = \frac{1}{k_1} (\theta_1 - \lambda_1)$$

$$A_2 = \frac{1}{k_2} \left(\sqrt{-2\alpha_2 \beta_2 \cos \left(\theta_2 - \Omega_2 - \tan^{-1} \left(\frac{\phi_2}{\gamma_2} \right) + 270^\circ \right) + \alpha_2^2 + \beta_2^2} - \lambda_2 \right)$$

$$A_3 = \frac{1}{k_3} \left(\sqrt{-2\alpha_3 \beta_3 \cos \left(\theta_2 + \theta_3 - \tan^{-1} \left(\frac{\phi_3}{\gamma_3} \right) + 90^\circ \right) + \alpha_3^2 + \beta_3^2} - \lambda_3 \right)$$

$$A_4 = \frac{1}{k_4} (180^\circ + \lambda_4 - \theta_2 - \theta_3 - \theta_4)$$

$$A_5 = \frac{1}{k_5} (\lambda_5 - \theta_5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่าตัวกระตุ้นมีขีดจำกัดในการเคลื่อนที่ เราจะต้องตรวจสอบค่าที่ได้เหล่านี้ว่าอยู่ใน ช่วงที่เป็นไปได้หรือไม่ ในการตรวจสอบนี้ในความเป็นจริงทำได้ยากมากในทาง mechanical ตัวกระตุ้นจะสัมพันธ์กันและมีผลซึ่งกันและกันในขณะที่เคลื่อนที่ สำหรับ Motoman ตัวกระตุ้น 2 และ 3 มีความสัมพันธ์ซึ่งกันและกัน ดังนั้นสามารถแสดงความสัมพันธ์นี้ได้เป็น

$$A_2 - 10000 > A_3 > A_2 + 3000$$

นั่นคือขีดจำกัดของตัวกระตุ้นที่ 2 เป็นฟังก์ชันของตำแหน่งตัวกระตุ้นที่ 3

และ

$$32000 - A_4 < A_5 < 55000$$

เพราะการหมุนของจุดหมุน 5 รอบหนึ่งจะหมุนได้ 25600 รอบ เมื่อ $A_4 > 2600$ ซึ่งมีอยู่สองสมการที่เป็นไปได้สำหรับ A_5



Robot Simulation

5.1 บทนำ

Simulation เป็นกระบวนการสร้างและทดลองอุปกรณ์ทำงาน(Manipulator) โดยใช้ระบบคอมพิวเตอร์สร้างเป็นรูปภาพขึ้นมา เพื่อจุดประสงค์ในการสังเกตและศึกษาการทำงานของอุปกรณ์ทำงาน(Manipulator) เมื่อเราสมมุติเหตุการณ์หรือ Parameter ขึ้นมา โดยกระบวนการเหล่านี้จะทำงานบนระบบคอมพิวเตอร์ทั้งสิ้น ซึ่งเราไม่จำเป็นที่จะต้องไปทดลองกับอุปกรณ์จริงเลยทำให้เป็นการลดเวลาและค่าใช้จ่ายต่าง ๆ ลงไปได้มาก [8]

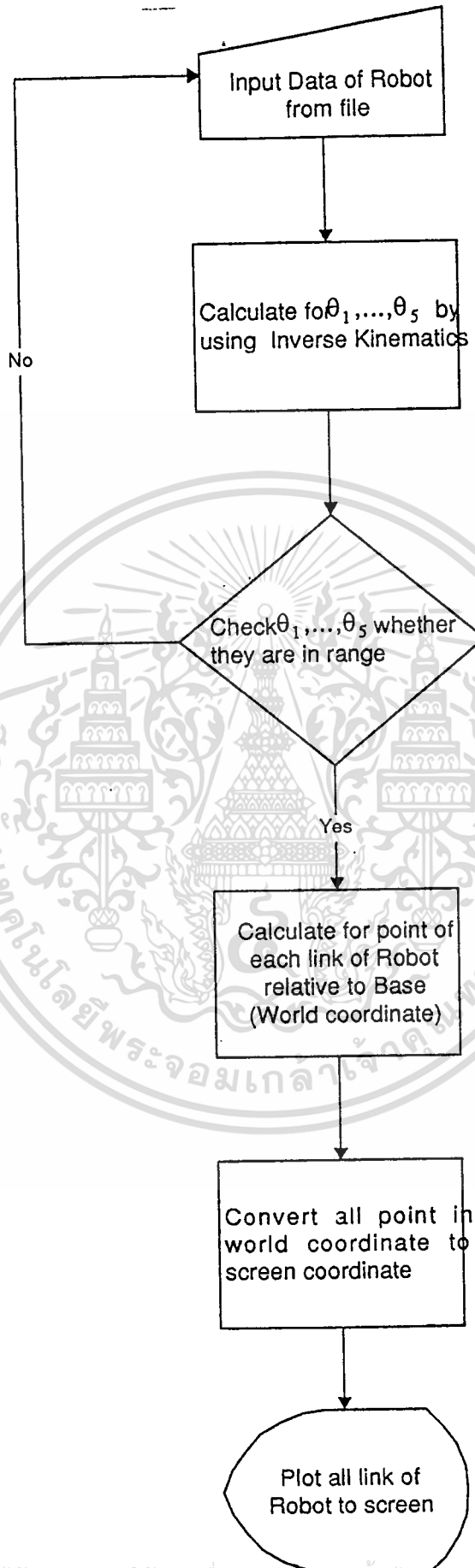
สำหรับใน Project นี้เรามีความจำเป็นที่จะต้องทำการ Simulation Robot ในคอมพิวเตอร์ เนื่องจากอุปกรณ์ต่าง ๆ ของตัว Robot ยังไม่พร้อมที่จะให้ทำการทดลองได้ ดังนั้นเราจึงจำเป็นที่จะต้องทำการ Simulation ขึ้นมาแทน เพื่อสังเกตการทำงานของ Robot โดยรูปภาพที่เราสร้างขึ้นในคอมพิวเตอร์จะจำลองลักษณะโดยคร่าว ๆ ของ Robot ขึ้นมา เพื่อให้เห็นการทำงานโดยง่าย ๆ ได้

Algorithm ในการเขียน Program มีดังนี้

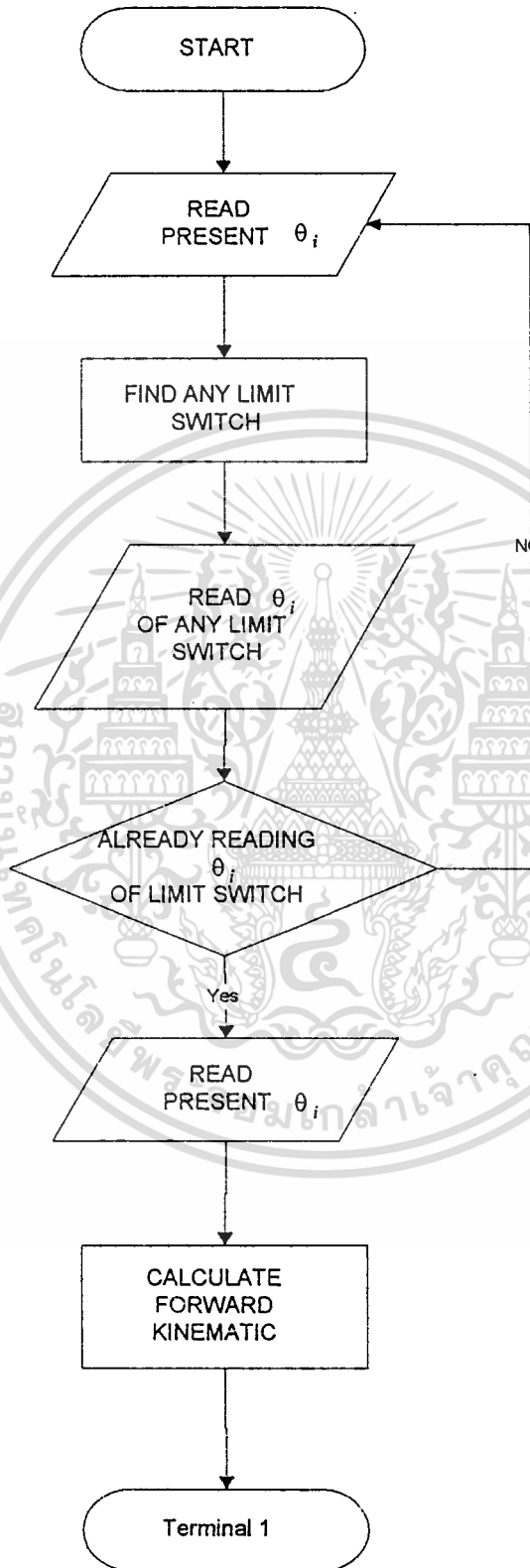
1. รับข้อมูลต่าง ๆ ของตัว Robot โดยใช้วิธีการอ่านเป็น File เข้ามา
2. คำนวณหาค่า $\theta_1, \dots, \theta_5$ โดยใช้ Inverse Kinematics
3. ตรวจสอบค่า $\theta_1, \dots, \theta_5$ ว่าเกินช่วงที่กำหนดหรือไม่ ถ้าเกินกลับไปทำข้อ 1. ใหม่
4. คำนวณหาค่าจุดของ link ของ Robot ทุก joint เทียบกับ Base (World coordinate*)
5. เปลี่ยนจาก World coordinate ให้เป็น Screen coordinate**
6. plot รูป Robot รวมทั้ง path การเคลื่อนที่โดยคร่าว ๆ บน Screen

* World coordinate เป็นระบบ coordinate ที่เป็นพิกัดโลกจะมีค่า x,y และ z โดยมีค่าไม่จำกัด [2]

** Screen coordinate เป็นระบบ coordinate ของหน้าจอภาพจะมีค่า x และ y เท่านั้น โดยวัดจากมุมบนซ้ายของจอภาพเป็นจุด (0,0) ค่ามากที่สุดของ coordinate นี้จะอยู่ที่มุมล่างขวาของจอภาพจะมีค่าเท่าใดขึ้นกับขนาด,ชนิด,ความละเอียดของจอภาพ หรือการ์ดควบคุม ซึ่งมีหน่วยเป็น pixel เช่น จอภาพปกติที่เราใช้กันอยู่จะมีความละเอียด 640 x 480 pixel เพราะฉะนั้นค่าที่มากที่สุดของ coordinate จะเป็น (639,479) เป็นต้น [2]

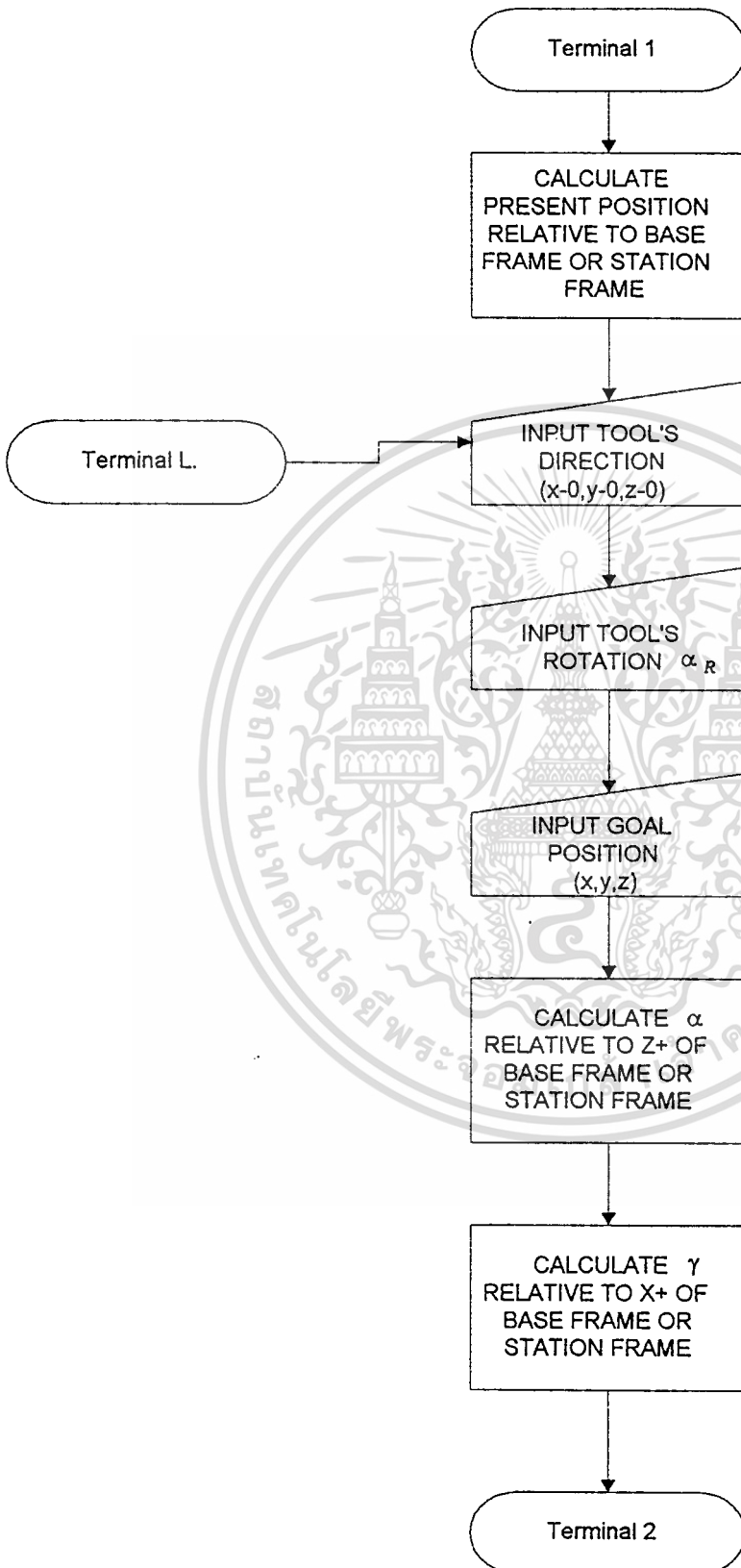


รูปที่ 5.1 Flow chart แสดง Algorithm ในการเขียน Program



แสดงขั้นตอนในการพัฒนาโปรแกรม (1)

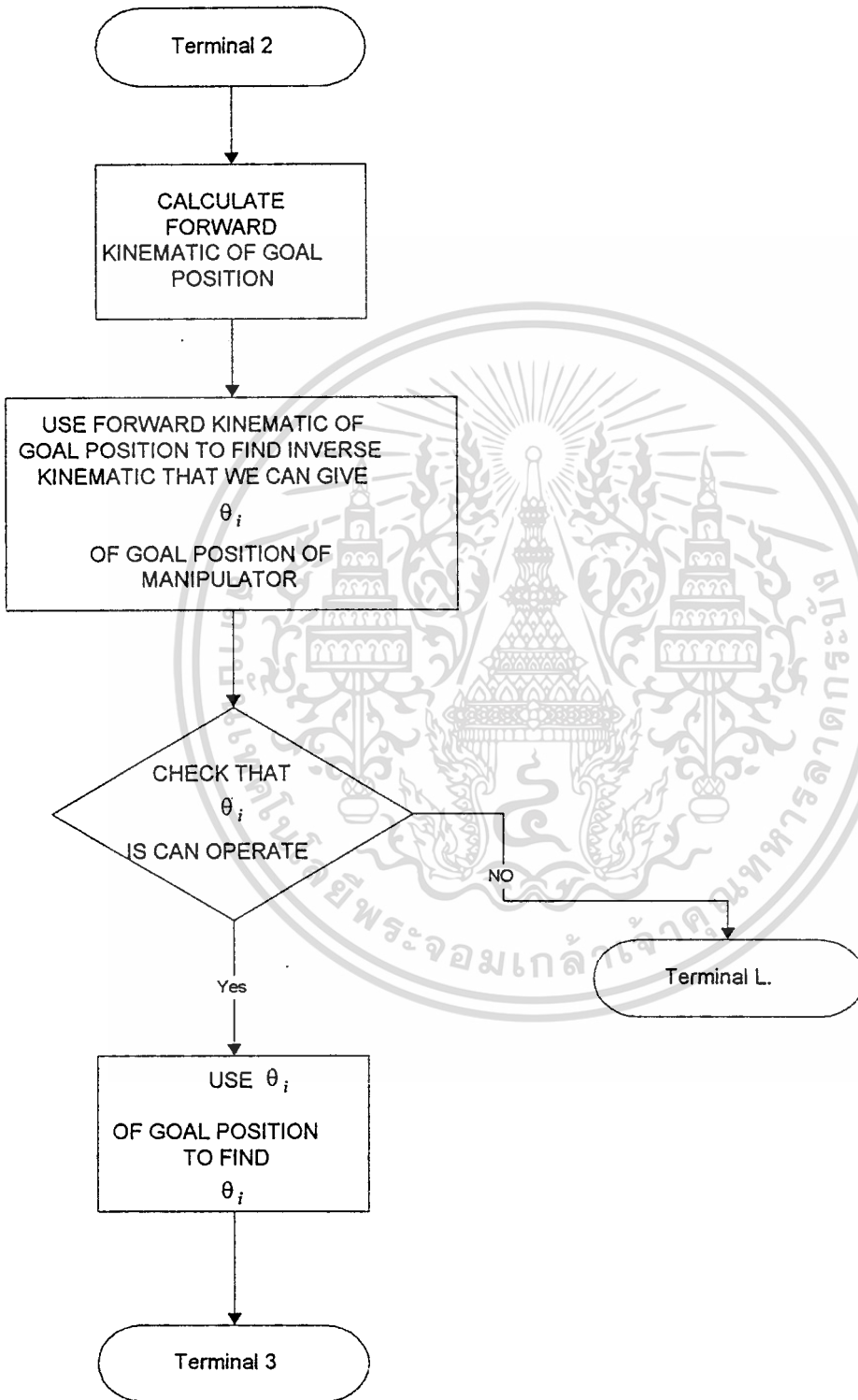
เอกรูปที่ 5.2.1 แสดง Flow chart อธิบายการเขียนโปรแกรมสำหรับการคำนวณ Forward และ Inverse Kinematic ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงขั้นตอนในการพัฒนาโปรแกรม (2)

รูปที่ 5.2.2 แสดง Flow chart อธิบายการเขียนโปรแกรมสำหรับการคำนวณ Forward และ Inverse Kinematic

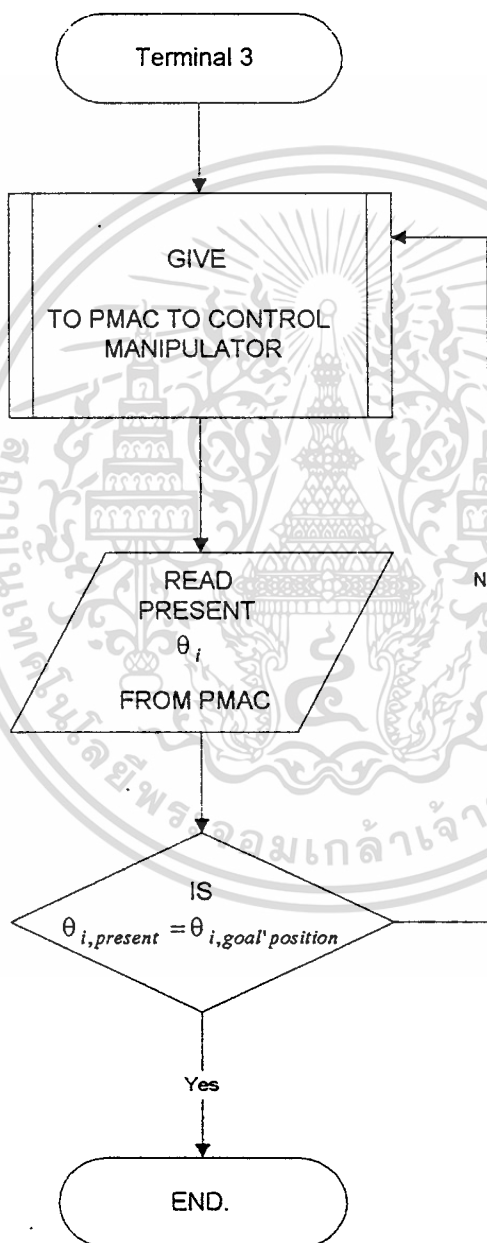
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงขั้นตอนในการพัฒนาโปรแกรม (3)

รูปที่ 5.2.3 แสดง Flow chart อธิบายการเขียนโปรแกรมสำหรับการคำนวณ Forward และ Inverse Kinematic

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงขั้นตอนในการพัฒนาโปรแกรม (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 5.2.4 แสดง Flow chart อธิบายการเขียนโปรแกรมสำหรับการคำนวณ Forward และ Inverse Kinematic
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 แสดงโปรแกรม Robot Control

robot2.cpp

```

//*****

// Main program robot2.cpp
// include file gpopup.cpp, gtext.cpp, 2d-7.cpp, trans-6.cpp
// Compiler " Turbo C++ version 3.00 "

// robot2.cpp - environment of program
// gpopup.cpp - create popup window and unpopup window
// gtext.cpp - generate text in graphics mode read and write to screen
// 2d-7.cpp - convert world to screen coordinate draw link, and path of robot
// trans-6.cpp - calculate forward, inverse kinematic and calculate all point
//           of all link of robot in world coordinate

//*****

#include<graphics.h>
#include<alloc.h>
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<process.h>
#include"gpopup.cpp"
#include"2d-7.cpp"

#define CHOICE 7

struct heading{
    char *choice; };

```

```

int row[CHOICE];
int col;
struct heading item[CHOICE];
int last_choice;
};

struct menu_struct menus;
char file_n[20];
void select_menu(int choice_no);

// setup environment
void setup_screen()
{
    setcolor(15);
    if(getmaxcolor()==1) setfillstyle(SOLID_FILL,0);
    else setfillstyle(SOLID_FILL,0);
    bar3d(0,0,getmaxx(),getmaxy(),10,0);
    setfillstyle(SOLID_FILL,MAGENTA);
    bar3d(getmaxx()*1/4-50,getmaxy()*1/4,getmaxx()*3/4+50,getmaxy()*3/4,0,1);
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,2);
    setcolor(getmaxcolor());
    setviewport(getmaxx()*1/4-50,getmaxy()*1/4,getmaxx()*3/4+50,getmaxy()*3/4,1);
    outtextxy(40,40,"ROBOT CONTROL PROGRAM");
    outtextxy(40,40+2*textheight("R")," Release 14.1 ");
    outtextxy(40,40+4*textheight("R")," by ");
    outtextxy(40,40+6*textheight("R")," Jakkrit Chaitewaruk");
    outtextxy(40,40+8*textheight("R")," Konlayuth Chailorm ");
    outtextxy(40,40+10*textheight("R")," Choopong Chuaypen ");
    setviewport(0,0,getmaxx(),getmaxy(),1);
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setfillstyle(SOLID_FILL,BLUE);
    bar3d(0,getmaxy()*19/20,getmaxx(),getmaxy(),0,0);

    setcolor(15);
    outtextxy(10,465,"F10-MENU, ALT-X to QUIT ESC-to Unpopup Window");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

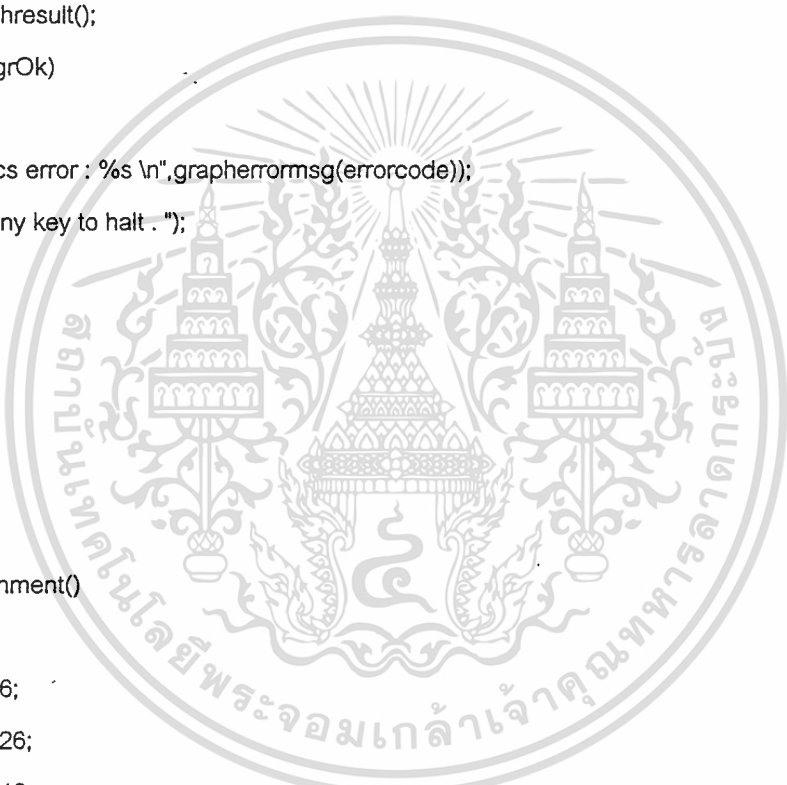
}

// initialize graphics mode
void init()
{
    int gdriver=DETECT,gmode,errorcode;
    initgraph(&gdriver,&gmode,"");
    errorcode=graphresult();
    if(errorcode != grOk)
    {
        printf("Graphics error : %s \n",grapherrormsg(errorcode));
        printf("Press any key to halt . ");
        getchar();
        closegraph();
        exit(1);
    }
}

void menu_assignment()
{
    menus.row[0]=6;
    menus.row[1]=26;
    menus.row[2]=46;
    menus.row[3]=66;
    menus.row[4]=86;
    menus.row[5]=106;
    menus.row[6]=126;

    menus.item[0].choice=" Main Memu ";
    menus.item[1].choice=" View all link ";
    menus.item[2].choice=" View some link ";
    menus.item[3].choice=" View goal position ";

```



```

menus.item[6].choice=" Quit      ";

menus.last_choice=6;
}

void display_menu()
{ int i;
  for(i=0;i<CHOICE;i++)
    outtextxy(menus.col,menus.row[i],menus.item[i].choice);
}

void draw_fill_rec(int lt,int tp,int rt,int bt)
{
  setfillstyle(1,GREEN);
  bar(lt,tp,rt,bt);
}

void erase_fill_rec(int lt,int tp,int rt,int bt)
{
  setfillstyle(SOLID_FILL,LIGHTGRAY);
  bar(lt,tp,rt,bt);
}

inverse(int choice_no)
{
  draw_fill_rec(menus.col+2,menus.row[choice_no]-4,menus.col+155,menus.row[choice_no]+10);
  setcolor(0);
  outtextxy(menus.col,menus.row[choice_no],menus.item[choice_no].choice);
  setcolor(15);
}

normal(int choice_no)
{

```

```

outtextxy(menus.col,menus.row[choice_no],rmenus.item[choice_no].choice);
}

// popup menu
open_menu()
{
    int i;
    setcolor(15);
    do{
        i=bioskey(0);
        switch(i) {
            case F10 : gpopup(5,5,getmaxx()*1/4+7,getmaxy()*2/4,THREED,getmaxcolor(),
                SOLID_FILL,LIGHTGRAY,BLACK);
                select_menu(1);
                break;
            case ALT_X : gunpop();closegraph();exit(0);break;
        };
    } while (1);
}

void help()
{
    gprintfxy(1,30,"View all link -view link at every goal position");
    gprintfxy(1,60,"View some link -view link at first & last goal position");
    gprintfxy(1,90,"View goal position -view goal position from select file");
    gprintfxy(1,120,"Edit data file -edit data files by using Quick Editor");
    gprintfxy(1,150,"Help -view this help message");
    gprintfxy(1,180,"Quit -quit from ROBOT CONTROL PROGRAM");
    gprintfxy(100,220,"Press any key ...");
    getch();
    setcolor(15);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
edit_file()
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  setcolor(15);
  rectangle(0,0,getmaxx()/2,80);
  gprintfxy(20,15,"Input file name : ... ");
  gscanfxy(20,40,"%s",file_n);
}

```

```

void select_menu(int choice_no)
{ int new_choice_no=0,i;
  display_menu();
  inverse(choice_no);
  for(;;){
    i=bioskey(0);
    switch(i) {
      case UP :if(choice_no==1) new_choice_no=menus.last_choice;
              else new_choice_no=choice_no-1;
              normal(choice_no);
              inverse(new_choice_no);
              choice_no=new_choice_no;
              break;
      case DOWN :if(choice_no==menus.last_choice) new_choice_no=1;
                else new_choice_no=choice_no+1;
                normal(choice_no);
                inverse(new_choice_no);
                choice_no=new_choice_no;
                break;
      case ENTER : gunpop();
                  switch(choice_no){
                    case 1 : cleardevice();plot_all_link();
                              setup_screen();open_menu();
                              break;
                    case 2 : cleardevice();plot_some_link();
                              setup_screen();open_menu();
                              break;

```

```

        case 3 : cleardevice();viewgoal();
                setup_screen();open_menu();
                break;

        case 4 : cleardevice();edit_file();closegraph();clrscr();
                spawnlp(P_WAIT,"q.exe","q.exe",file_n,NULL);
                init();setup_screen();open_menu();
                break;

        case 5 : gpopup(getmaxx()*1/4-
63,getmaxy()*1/4,getmaxx()*3/4+64,getmaxy()*3/4,
                THREEED,getmaxcolor(),SOLID_FILL,LIGHTGRAY,RED);
                help();gunpop();
                open_menu();
                break;
        case 6 : closegraph();exit(0);
                break;
};
break;

case ESCAPE : gunpop();open_menu();break;
}
}
}

remark()
{
    setcolor(15);
    gpopup(getmaxx()*1/4-50,getmaxy()*1/4,getmaxx()*3/4+50,getmaxy()*3/4,
            THREEED,getmaxcolor(),SOLID_FILL,LIGHTGRAY,RED);
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,2);
    gprintfxy(10,70,"This program use for only");
    gprintfxy(10,140,"YASUKAWA MOTOMAN L-3 !!!");
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
    gprintfxy(100,220,"Press any key ...");

    getch();
    gunpop();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(15);  
}  
  
//***** FUNCTION MAIN *****/  
  
main()  
{  
    init();  
    setup_screen();  
    remark();  
    menu_assignment();  
    open_menu();  
}  
  
//***** END PROGRAM *****/
```




```

line(right-i,top+i,right-i,bottom-i);
line(left+i,bottom-i,right-i,bottom-i);
setcolor(WHITE);
line(left+i,top+i,right-i,top+i);
line(left+i,top+i,left+i,bottom-i);
}
setcolor(bfc);
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
setfillstyle(bft,bfc);
bar3d(left+2,top+2,right-2,bottom-2,0,0);
}
else {
setlinestyle(bt,0,NORM_WIDTH);
setcolor(bc);
setfillstyle(bft,bfc);
bar3d(left,top,right,bottom,0,0);
}
}

int savewindow(int left,int top,int right,int bottom)
{
int halfpoint;
halfpoint=(top+bottom)/2;
wstack[wptr]->savearea1=malloc(imagesize(left,top,right,halfpoint));
wstack[wptr]->savearea2=malloc(imagesize(left,halfpoint+1,right,bottom));
if (wstack[wptr]->savearea1==NULL || wstack[wptr]->savearea2==NULL) return 0;
getimage(left,top,right,halfpoint,wstack[wptr]->savearea1);
getimage(left,halfpoint+1,right,bottom,wstack[wptr]->savearea2);
return 1;
}

```

//popup window ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สงวนลิขสิทธิ์ในสิ่งที่ปรากฏบนเว็บไซต์นี้ หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายบริการลูกค้าของเรา

```

int backfill,int fillcolor,int drawcolor)
{
struct viewporttype oldview;
int oldx,oldy,savecolor;
struct linesettingstype saveline;
struct fillsettingstype savefill;
if (wptr>=NUMWINDOWS) return 0;
wstack[wptr]=(struct GRAPHICSWINDOW *)malloc(sizeof(struct GRAPHICSWINDOW));
if(wstack[wptr]==0) return 0;
getviewsettings(&oldview);
oldx=getx();
oldy=gety();
setviewport(0,0,getmaxx(),getmaxy(),1);
getlinesettings(&saveline);
savecolor=getcolor();
getfillsettings(&savefill);
if(savewindow(left,top,right,bottom)==0) {
setviewport(oldview.left,oldview.top,oldview.right,oldview.bottom,1);
moveto(oldx,oldy);
free(wstack[wptr]);
return 0;
}
wstack[wptr]->bc=bordercolor;
wstack[wptr]->bt=bordertype;
wstack[wptr]->bft=backfill;
wstack[wptr]->bfc=fillcolor;
wstack[wptr]->dc=drawcolor;
paintwindow(left,top,right,bottom,bordertype,fillcolor,backfill,bordercolor);
if(bordertype==THREED)
setviewport(left+2,top+2,right-2,bottom-2,1);
else
setviewport(left+1,top+1,right-1,bottom-1,1);

```

เอกภพนี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สามารถให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

wstack[wptr]->fbottom=bottom;
wstack[wptr]->fright=right;
wstack[wptr]->vleft=oldview.left;
wstack[wptr]->vtop=oldview.top;
wstack[wptr]->vright=oldview.right;
wstack[wptr]->vbottom=oldview.bottom;
wstack[wptr]->cpx=oldx;
wstack[wptr]->cpy=oldy;
wstack[wptr]->drawcolor=savecolor;
wstack[wptr]->origbfc=settextbkcolor(fillcolor);
wstack[wptr]->origdc=settextcolor(drawcolor);
setlinestyle(saveline.linestyle,saveline.upattern,saveline.thickness);
setcolor(savecolor);
setfillstyle(savefill.pattern,savefill.color);
wptr++;
return 1;
}

//unpopup window
int gunpop()
{
    struct viewporttype curview;
    struct GRAPHICSWINDOW *w;
    if(wptr<=0) return 0;
    wptr--;
    w=wstack[wptr];
    setviewport(w->fleft,w->ftop,w->fright,w->fbottom,1);
    getviewsettings(&curview);
    putimage(0,0,w->savearea1,COPY_PUT);
    putimage(0,(curview.bottom-curview.top)/2+1,w->savearea2,COPY_PUT);
    setviewport(w->vleft,w->vtop,w->vright,w->vbottom,1);
    moveto(w->cpx,w->cpy);

    setcolor(w->drawcolor);
    settextbkcolor(w->origbfc);

```

```

settextcolor(w->origdc);
free(w->savearea1);
free(w->savearea2);
free(wstack[wptr]);
return 1;
}

```

```

void unpopallwindows()
{
while (gunpop());
}

```

```

int printfxy(int xloc,int yloc,char *fmt,...)
{
int oldbc=settextbkcolor(wstack[wptr-1]->bc);
int olddc=settextcolor(wstack[wptr-1]->dc);
int cnt=gprintfxy(xloc,yloc,fmt);
settextbkcolor(oldbc);
settextcolor(olddc);
return cnt;
}

```

gtext.cpp

```

#include<graphics.h>
#include<stdarg.h>
#include<stdio.h>
#include<string.h>
#include<math.h>
#include<conio.h>
#include<bios.h>
#include<ctype.h>

#define TEXTSIZE 8
#define BUFSIZE 140

#define TAB 0x0f09
#define BS 8 /*0x0e08*/
#define CR 13 /*0x1c0d*/
#define ESC 27 /*0x011b*/

#define BSPACE 0x0e08
#define ENTER 0x1c0d
#define ESCAPE 0x011b

#define DEL 0x5300
#define UP 0x4800
#define DOWN 0x5000
#define LEFT 0x4B00
#define RIGHT 0x4D00
#define F1 0x3b00
#define F10 0x4400
#define ALT_X 0x2d00

int grtextbkcolor=0;
int grtextcolor=13;

```



```
int settextcolor(int color)
{
    int oldcolor=grtextcolor;
    grtextcolor=color;
    return oldcolor;
}
```

```
int settextbkcolor(int color)
{
    int oldcolor=grtextbkcolor;
    grtextbkcolor=color;
    return oldcolor;
}
```

```
// graphics base printf function
int gprintfxy(int xloc,int yloc,char *fmt,...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt;
    struct fillsettingstype oldfill;
    char userfillpattern[8];
    va_start(argptr,fmt);
    cnt=vsprintf(str,fmt,argptr);
    if(str[0]==NULL) return 0;
    getfillsettings(&oldfill);
    if (oldfill.pattern==USER_FILL) getfillpattern(userfillpattern);
    setfillstyle(SOLID_FILL,grtextbkcolor);
    setcolor(grtextcolor);
    bar(xloc,yloc,xloc+textwidth(str),yloc+textheight("H")*5/4);
    if (oldfill.pattern==USER_FILL)
        setfillpattern(userfillpattern,oldfill.color);
```

```

    setfillstyle(oldfill.pattern,oldfill.color);
    outtextxy(xloc,yloc,str);
    va_end(argptr);
    return cnt;
}

// graphics base printf function
int gprintf(char *fmt,...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt,xloc,yloc;
    struct fillsettingstype oldfill;
    char userfillpattern[8];
    va_start(argptr,fmt);
    cnt=vsprintf(str,fmt,argptr);
    if(str[0]==NULL) return 0;
    xloc=getx();yloc=gety();
    getfillsettings(&oldfill);
    if (oldfill.pattern==USER_FILL) getfillpattern(userfillpattern);
    setfillstyle(SOLID_FILL,grtextbkcolor);
    setcolor(grtextcolor);
    bar(xloc,yloc,xloc+textwidth(str),yloc+textheight("H")*5/4);
    if (oldfill.pattern==USER_FILL)
        setfillpattern(userfillpattern,oldfill.color);
    else
        setfillstyle(oldfill.pattern,oldfill.color);
    outtext(str);
    va_end(argptr);
    return cnt;
}

```

```
int ggetche()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char ch;
ch=getch();
gprintf("%c",ch);
return ch;
}

```

```

int gputch(int c)
{
char buffer[2];
sprintf(buffer,"%c",c);
gprintf(buffer);
return c;
}

```

```

char *ggets(char *buffer)
{
int curloc,maxchars,oldcolor,ch;
struct viewporttype view;
char charbuff[3];
buffer[0]='\0';
curloc=0;
getviewsettings(&view);
maxchars=(view.right-gebx())/textwidth("M")-1;
if (maxchars<=0) return(NULL);
gprintfxy(getx(),gety(),"_");
while((ch=getche())!=CR && ch!=ESC) {
if (ch==BS) {
if (curloc >0) {
curloc--;
if (curloc<=maxchars) {
oldcolor=settextcolor(grtextbkcolor);
sprintf(charbuff,"%c",buffer[curloc]);
gprintfxy(getx()-textwidth(charbuff),gety(),"%c_",buffer[curloc]);
settextcolor(oldcolor);

```

```

        moveto(getx()-textwidth(charbuff),gety());
    }
}
else {
    if(currloc<maxchars) {
        oldcolor=settextcolor(grtextbkcolor);
        gprintfxy(getx(),gety(),"_");
        settextcolor(oldcolor);
        buffer[currloc]=ch;
        gputch(ch);
        currloc++;
    }
    else
        putch(0x07);
}
if (currloc<maxchars) gprintfxy(getx(),gety(),"_");
}
if (currloc<=maxchars) {
    oldcolor=settextcolor(grtextbkcolor);
    gprintfxy(getx(),gety(),"_");
    settextcolor(oldcolor);
}
if (ch==ESC) buffer[0]='\0';
    else buffer[currloc]='\0';
return buffer;
}

```

// graphics base scanf function

```
int gscanf(char *fmt,...)
```

```
{
    va_list argptr;
```

char str[BUFSIZE]; สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

int cnt; ไม่ควรละเลยทุกสิ่ง อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

va_start(argptr,fmt);
ggets(str);
cnt=vsscanf(str,fmt,argptr);
va_end(argptr);
return cnt;
}

```

// graphics base scanf function

```
int gscanfxy(int xloc,int yloc,char *fmt,...)
```

```

{
va_list argptr;
char str[BUFSIZE];
int cnt,oldx,oldy;
oldx=xloc;
oldy=yloc;
moveto(xloc,yloc);
va_start(argptr,fmt);
ggets(str);
cnt=vsscanf(str,fmt,argptr);
va_end(argptr);
moveto(oldx,oldy);
return cnt;
}

```

```
void drawcurs(int locx,int locy,unsigned char erase)
```

```

{
static int lastx=0,lasty=0;
if (erase) {
setcolor(grtextbkcolor);
moveto(lastx*TEXTSIZE,(lasty+1)*TEXTSIZE+1);
lineto(locx*TEXTSIZE+7,(locy+1)*TEXTSIZE+1);
lastx=locx;lasty = locy;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void displaystr(int locx,int locy,int dlen,char str[])
{
    int i;
    for (i=0;i<dlen && str[i] !='\0';i++)
        gprintfxy((locx+i)*TEXTSIZE,locy*TEXTSIZE,"%c",str[i]);
    for (;i<dlen;i++)
        gprintfxy((locx+i)*TEXTSIZE,locy*TEXTSIZE," ");
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2d-7.cpp

```

#include<graphics.h>
#include<math.h>
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<dos.h>
#include<trans-6.cpp>

typedef struct {
    double x,y,z;
} point;

point p[10];

double xwl,xwr,ywt,ywb,a,b,c,d;
int xvl,xvr,yvt,yvb,x1,y1,z1,x2,y2,z2,x3,y3,z3;
int color;

extern POSITION P,p_org[6],T;
FILE *file;
char filename[20];
int num;
open_menu();
//void setup_screen();

void set_window(double xmin,double xmax,double ymin,double ymax)
{
    xwl=xmin;xwr=xmax;
    ywb=ymin;ywt=ymax;
}

void set_viewport(int xmin,int xmax,int ymin,int ymax)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ควรละเมิดลิขสิทธิ์ของผู้อื่น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

yvb=ymin;yvt=ymax;
a=(xvr-xvl)/(xwr-xwl); b=xvl-a*xwl;
c=(yvt-yvb)/(ywt-ywb); d=yvb-c*ywb;
rectangle(xvl,yvb,xvr,yvt);

}

// convert world coordinate to screen coordinate
void WORLDtoPC(double xw,double yw,int &xpc,int &ypc)
{ int ypc_temp;
  xpc=(int)(a*xw+b+0.5);ypc_temp=(int)(c*yw+d+0.5);
  if (yvb>=240) ypc=479-(ypc_temp-yvb);
  else ypc=yvt-yvb-ypc_temp;
}

//draw xy plane
void xy_plane()
{
  set_window(-150,150,-150,150);
  set_viewport(0,319,0,239);
  WORLDtoPC(p_org[0].x,p_org[0].y,x1,y1);
  WORLDtoPC(p_org[3].x,p_org[3].y,x2,y2);
  WORLDtoPC(p_org[5].x,p_org[5].y,x3,y3);
  color=getcolor();
  setcolor(3);
  line(x1,y1,x1+30,y1); outtextxy(x1+30,y1,"x");
  line(x1,y1,x1,y1-30); outtextxy(x1,y1-30,"y");
  setcolor(14);
  circle(x1,y1,8);
  setcolor(2);
  circle(x1,y1,5);setfillstyle(SOLID_FILL,2);floodfill(x1,y1,2);
  setcolor(color);
  line(x1,y1,x2,y2);
  setcolor(13);

```

```

circle(x2,y2,5);setfillstyle(SOLID_FILL,13);floodfill(x2,y2,13);
setcolor(color);
line(x2,y2,x3,y3);
setcolor(color);
setcolor(4);
circle(x3,y3,3);setfillstyle(SOLID_FILL,4);floodfill(x3,y3,4);
setcolor(2);
setcolor(color);
outtextxy(10,220,"xy plane");

```

```

}

//draw xz plane
void xz_plane()
{
set_window(-150,150,-50,150);
set_viewport(0,319,240,479);
WORLDtoPC(p_org[0].x,p_org[0].z,x1,z1);
WORLDtoPC(p_org[3].x,p_org[3].z,x2,z2);
WORLDtoPC(p_org[5].x,p_org[5].z,x3,z3);
color=getcolor();
setcolor(3);
line(x1,z1,x1+30,z1); outtextxy(x1+30,z1,"x");
line(x1,z1,x1,z1-30); outtextxy(x1,z1-30,"z");
setcolor(14);
rectangle(x1-7,z1,x1+7,z1+20);
setcolor(color);
line(x1,z1,x2,z2);
setcolor(13);
circle(x2,z2,5);setfillstyle(SOLID_FILL,13);floodfill(x2,z2,13);
setcolor(color);
line(x2,z2,x3,z3);
setcolor(2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(color);
setcolor(4);
circle(x3,z3,3); setfillstyle(SOLID_FILL,4);floodfill(x3,z3,4);
setcolor(color);
outtextxy(10,460,"xz plane");
}

```

```
//draw yz plane
```

```
void yz_plane()
```

```

{
set_window(-150,150,-50,150);
set_viewport(320,639,240,479);
WORLDtoPC(p_org[0].y,p_org[0].z,y1,z1);
WORLDtoPC(p_org[3].y,p_org[3].z,y2,z2);
WORLDtoPC(p_org[5].y,p_org[5].z,y3,z3);
color=getcolor();
setcolor(3);
line(y1,z1,y1+30,z1); outtextxy(y1+30,z1,"y");
line(y1,z1,y1,z1-30); outtextxy(y1,z1-30,"z");
setcolor(14);
rectangle(y1-7,z1,y1+7,z1+20);
setcolor(color);
line(y1,z1,y2,z2);
setcolor(13);
circle(y2,z2,5);setfillstyle(SOLID_FILL,13);floodfill(y2,z2,13);
setcolor(color);
line(y2,z2,y3,z3);
setcolor(2);
circle(y1,z1,5);setfillstyle(SOLID_FILL,2);floodfill(y1,z1,2);
setcolor(color);
setcolor(4);
circle(y3,z3,3);setfillstyle(SOLID_FILL,4);floodfill(y3,z3,4);
setcolor(color);
outtextxy(330,460,"yz plane");
}

```

```

}

//read file
infile()
{ int i;
  setcolor(15);
  if ((file=fopen(filename,"r"))==NULL)
  {
    gprintfxy(330,40,"No file %s ",filename);
    getch(); setcolor(15);
    setup_screen();
    open_menu();
  }
  fscanf(file,"%d",&num);
  fscanf(file,"%lf %lf %lf",&T.x,&T.y,&T.z);
  for (i=0;i<num;i++)
    fscanf(file,"%lf %lf %lf",&p[i].x,&p[i].y,&p[i].z);
  fclose(file);
}

```

```

//read file
infile2()
{ int i;
  setcolor(15);
  if ((file=fopen(filename,"r"))==NULL)
  {
    gprintfxy(20,15,"No file %s ",filename);
    getch(); setcolor(15);
    setup_screen();
    open_menu();
  }

```

```
fscanf(file,"%d",&num);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ fscanf(file,"%lf %lf %lf",&T.x,&T.y,&T.z); เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i=0;i<num;i++)
    fscanf(file,"%lf %lf %lf",&p[i].x,&p[i].y,&p[i].z);
fclose(file);
}

```

```
//read file
```

```
void ginput()
```

```
{ int i,j,height;
```

```
rectangle(0,0,getmaxx(),getmaxy());
```

```
set_window(-150,150,-150,150);
```

```
set_viewport(320,639,0,239);
```

```
setcolor(15);
```

```
gprintfxy(330,40,"Input file name : ... ");
```

```
height=textheight("I")+30;
```

```
gscanfxy(330,40+height,"%s",filename);
```

```
for (i=321;i<639;i++)
```

```
    for (j=38;j<100;j++) putpixel(i,j,0);
```

```
setcolor(15);
```

```
infile();
```

```
setcolor(15);
```

```
}
```

```
void present()
```

```
{
```

```
xy_plane();delay(500);
```

```
xz_plane();delay(500);
```

```
yz_plane();delay(500);
```

```
}
```

```
//draw path of goal position
```

```
void drawpath(int k)
```

```
{
```

```
set_window(-150,150,-150,150); /* xy */
```

```
set_viewport(0,319,0,239);
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WORLDtoPC(p[k-1].x,p[k-1].y,x1,y1);
WORLDtoPC(p[k].x,p[k].y,x2,y2);
color=getcolor();
setcolor(10);
line(x1,y1,x2,y2);
setcolor(color);
delay(500);

```

```

set_window(-150,150,-50,150); /* xz */
set_viewport(0,319,240,479);
WORLDtoPC(p[k-1].x,p[k-1].z,x1,z1);
WORLDtoPC(p[k].x,p[k].z,x2,z2);
color=getcolor();
setcolor(10);
line(x1,z1,x2,z2);
setcolor(color);
delay(500);

```

```

set_window(-150,150,-50,150); /* yz */
set_viewport(320,639,240,479);
WORLDtoPC(p[k-1].y,p[k-1].z,y1,z1);
WORLDtoPC(p[k].y,p[k].z,y2,z2);
color=getcolor();
setcolor(10);
line(y1,z1,y2,z2);
setcolor(color);
delay(500);

```

```

}

```

```

plot_all_link()

```

```

{ int k;
  ginput();
  for (k=0;k<num;k++)

```

```

    { P.x=p[k].x; P.y=p[k].y; P.z=p[k].z;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    trans();
    present();
    if((k>0) && (k<num)) drawpath(k);
}
setcolor(15);
gprintfxy(330,40,"Press any key ...");
getch();
}

```

```

plot_some_link()
{ int k;
  ginput();
  for (k=0;k<num;k++)
    { P.x=p[k].x; P.y=p[k].y; P.z=p[k].z;
      trans();
      if((k==num-1) || (k==0)) present();
      if((k>0) && (k<num)) drawpath(k);
    }
  setcolor(15);
  gprintfxy(330,40,"Press any key ...");
  getch();
}

```

```

viewgoal()
{ int i,j,height,ypos=15;
  setcolor(15);
  rectangle(0,0,getmaxx(),getmaxy());
  gprintfxy(20,15,"Input file name : ... ");
  gscanfxy(20,40,"%s",filename);
  for (i=10;i<300;i++)
    for (j=10;j<100;j++) putpixel(i,j,0);
  setcolor(15);

```

```
infile2();
```

```
setcolor(15);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i=0;i<num;i++) {
    height=textheight("H");
    ypos+=height*2;
    if (ypos>=450) ypos=15;
    gprintfxy(20,ypos,"%d. x %.2f y %.2f z %.2f",i+1,p[i].x,p[i].y,p[i].z);
}
setcolor(15);
gprintfxy(20,ypos+50,"Press any key ...");
getch();
}

```



trans-6.cpp

```

#include <string.h>
#include <stdarg.h>
#include <stdio.h>
#include <math.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <graphics.h>

#define Pi 3.141592654

typedef struct {
    double x,y,z;
}POSITION;

/* Operating Function *****/
double sqr(double);

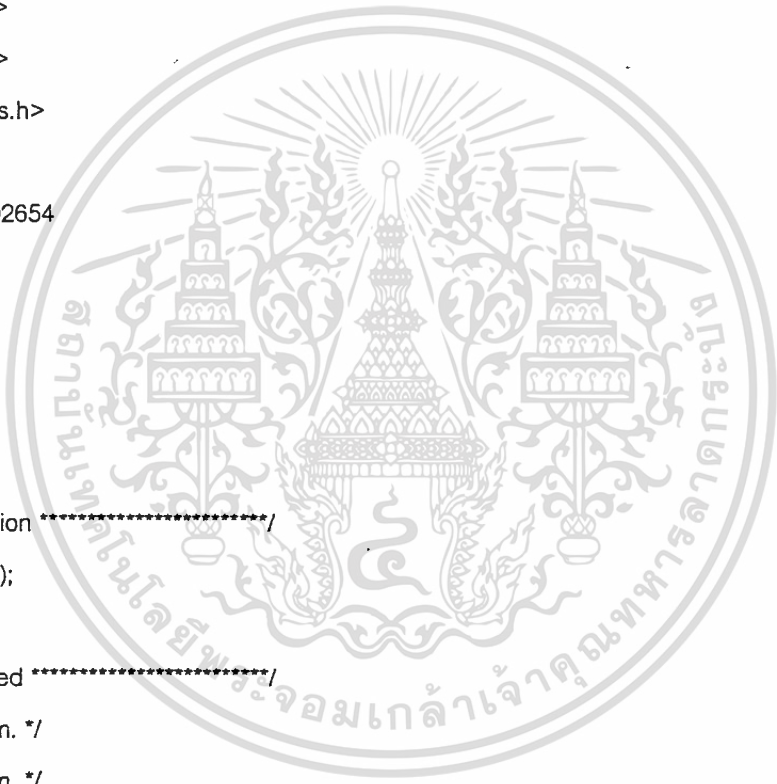
/* Operating Defined *****/
#define l2 65 /* cm. */
#define l3 80 /* cm. */

double grm,alp,bta;
double Q[6],T05[5][5];
POSITION P,p_org[6],T;

void init();
void setup_screen();
open_menu();

//calculate all point in world coordinate
void mcs()

```



```

{
    p_org[0].x=0.0; p_org[0].y=0.0; p_org[0].z=0.0;
    p_org[1].x=0.0; p_org[1].y=0.0; p_org[1].z=0.0;
    p_org[2].x=0.0; p_org[2].y=0.0; p_org[2].z=0.0;

    p_org[3].x=l2*cos(Q[1])*cos(Q[2]);
    p_org[3].y=l2*sin(Q[1])*cos(Q[2]);
    p_org[3].z=-l2*sin(Q[2]);

    p_org[4].x=P.x;
    p_org[4].y=P.y;
    p_org[4].z=P.z;

    p_org[5].x=P.x;
    p_org[5].y=P.y;
    p_org[5].z=P.z;
}

double sqr(double x) {
    return pow(x,2.00);
}

/*
double f(double q) {
    return sqrt(sqrt(T05[1][4])+sqrt(T05[2][4]))
        +l2*sin(Q[3])*sin(q)-(l2+l3*cos(Q[3]))*cos(q);
}

double FINE_Q2(void) {
    double a,Q,i;

    Q=0.0; a=0.0; i=0.0;

    while ( (f(Q)!=0.0)&&(sqr(f(Q))>0.00001)&&(i<10.0) )

```



```

if (f(a)*f(Q)<0.0) { Q=a-f(a)*(a-Q)/(f(a)-f(Q)); }
else { a=Q;i=i+1.0;Q=-i*Pi/10.0; }
}

return Q;

}

unsigned short int inverse_link_kinematic(void) {
    double Q234,temp1,temp2,temp3;

    Q234=acos(T05[3][3]);

    Q[5]=asin(T05[3][2]/sin(Q234));

    Q[1]=asin(T05[2][3]/sin(Q234));

    Q[3]=acos( (sqr(T05[1][4])+sqr(T05[2][4])+sqr(T05[3][4])
                -sqr(l2)-sqr(l3) )/(2*l2*l3) );

    Q[2]=FINE_Q2();

    Q[4]=Q234-Q[2]-Q[3];

    return 0;

}
*/

unsigned short int inverse_link_kinematic(void) {
    double temp1,temp2,Q234,c3;

```

```

Q[1]=atan2(temp1,temp2);

temp1=T05[2][1]*cos(Q[1])-T05[1][1]*sin(Q[1]);
temp2=T05[2][2]*cos(Q[1])-T05[1][2]*sin(Q[1]);
Q[5]=atan2(temp1,temp2);

temp1=T05[1][3]*cos(Q[1])+T05[2][3]*sin(Q[1]);
temp2=T05[3][3];
Q234=atan2(temp1,temp2);

c3=(sqr(T05[1][4])+sqr(T05[2][4])+sqr(T05[3][4])
    -sqr(l2)-sqr(l3))/(2*l2*l3);
if (sqr(c3)>1.0) {
    closegraph();
    printf("\n\n\n\t\tError in data file input !!!\n\n\n");
    printf("\t\tPlease check your data again.\n\n\n");
    printf("\t\tYour data may be out of range of robot to operate.");
    getch();
    init();
    setup_screen();
    open_menu();
}

temp1=sqrt(1.0-sqr(c3));
temp2=c3;
Q[3]=atan2(temp1,temp2);

temp1=T05[3][4];
temp2=sqrt(sqr(T05[1][4])+sqr(T05[2][4]));
Q[2]=-atan2(temp1,temp2)-atan2(l2*sin(Q[3]),l2+l3*cos(Q[3]));
Q[4]=Q234-Q[2]-Q[3];

```

```

return 0;
}

unsigned short int tool_input(void) {

    if ( T.x==0.0 && T.y==0.0 ) { grm=0.0; alp=0.0; }
    else {   grm = acos( T.z / sqrt(sqrt(T.x)+sqrt(T.y)+sqrt(T.z)) );
            alp = acos( T.y / sqrt(sqrt(T.x)+sqrt(T.y)) ); }

    bta=bta*Pi/180.0;
    if(! ( ( sqrt(l2+l3)<=(sqrt(P.x)+sqrt(P.y)+sqrt(P.z)) ) &&
            ( sqrt(l2+l3)>=(sqrt(P.x)+sqrt(P.y)+sqrt(P.z)) ) ) )
    { closegraph();
      printf("\n\n\n\t\tError in data file input !!!\n\n\n");
      printf("\t\tPlease check your data again.\n\n\n");
      printf("\t\tYour data may be out of range of robot to operate.");
      getch();
      init();
      setup_screen();
      open_menu();
    }

    return 0;
}

```

```

unsigned short int forward_kinematic(void)

```

```

{
    T05[1][1]= cos(alp)*cos(bta)-sin(alp)*cos(grm)*sin(bta);
    T05[2][1]= sin(alp)*cos(bta)+cos(alp)*sin(grm)*sin(bta);
    T05[3][1]= sin(grm)*sin(bta);
    T05[4][1]= 0.0;

```

```
T05[3][2]= sin(grm)*cos(bta);
```

```
T05[4][2]= 0.0;
```

```
T05[1][3]= sin(alp)*sin(grm);
```

```
T05[2][3]=-cos(alp)*sin(grm);
```

```
T05[3][3]= cos(grm);
```

```
T05[4][3]= 0.0;
```

```
T05[1][4]= P.x;
```

```
T05[2][4]= P.y;
```

```
T05[3][4]= P.z;
```

```
T05[4][4]= 1.0;
```

```
return 0;
```

```
}
```

```
void trans(void)
```

```
{
```

```
tool_input();
```

```
forward_kinematic();
```

```
inverse_link_kinematic();
```

```
mcs();
```

```
}
```



การปฏิบัติงานของโปรแกรม Robot Control

6.1 วิธีการใช้โปรแกรม Robot Control

1. เรียกโปรแกรมที่ prompt C:\> robot2
2. เมื่อเข้าโปรแกรมแล้วจะเห็น Comment แสดงขึ้นมา หลังจากนั้นให้กดปุ่มใด ๆ หลังจากนั้นจะมี Back ground ของโปรแกรมแสดงขึ้นมา ซึ่งจะมีข้อความบอกหน้าที่ของปุ่มบางปุ่ม
3. หากต้องการใช้โปรแกรม ให้กดปุ่ม F10 จะมี popup menu แสดงขึ้นมาหากต้องการทำงานใด ให้ใช้ปุ่มลูกศรเลื่อนแถบแสงบน menu ขึ้นลงแล้วกดปุ่ม Enter
4. ใน menu ประกอบด้วย function หลักอยู่ 6 function ประกอบด้วย

View all link

View some link

View goal position

Edit data file

Help

Quit

View all link จะเป็น function เกี่ยวกับการแสดง link ของ Robot ทุก link เมื่อ Robot เคลื่อนที่ไปตามแต่ละ goal position โดยผู้ใช้จะต้องใส่ชื่อ data file ของ Robot เข้าไปเมื่อมีการเลือกใช้ function นี้

View some link เป็น function ที่แสดง link ของ Robot บาง link ซึ่งได้แก่ link ของ goal position ที่ 1 และ link ของ goal position สุดท้าย โดยการทำงานจะเหมือนกับ function view all link

View goal position เป็น function ที่จะใช้ดู goal position จาก data file ของ Robot ซึ่งเมื่อเลือกใช้ function นี้ผู้ใช้จะต้องใส่ชื่อ data file เข้าไป

Edit data file เป็น function ที่ใช้แก้ไขหรือดูข้อมูลของ data file ของ Robot ซึ่งเมื่อเลือกใช้ function นี้ผู้ใช้จะต้องใส่ชื่อ data file เข้าไป

Help เป็น function แสดงวิธีการใช้โปรแกรม โดยแสดงวิธีการใช้ทุกคำสั่งที่มีอยู่ใน menu

Quit เป็น function ที่ใช้เมื่อต้องการออกจากโปรแกรม Robot Control

5. Format ของ data file ซึ่งสามารถใช้โปรแกรม Editor ได้ ๆ สร้างขึ้นมาก็ได้

5 → คือจำนวนของ goal position

1 2 1 → คือ Vector แสดง Rotation ปลาย link ของ Robot

15 25 60 → คือ goal position ที่ 1

20 30 50 → คือ goal position ที่ 2

-15 25 70 → คือ goal position ที่ 3

35 -25 77 → คือ goal position ที่ 4

40 45 55. → คือ goal position ที่ 5

6.2 รูปภาพแสดงการทำงานของโปรแกรม Robot Control

6.2.1 แสดงวิธีการเข้าสู่โปรแกรม Robot Control

```

C:\ROBOT>dir

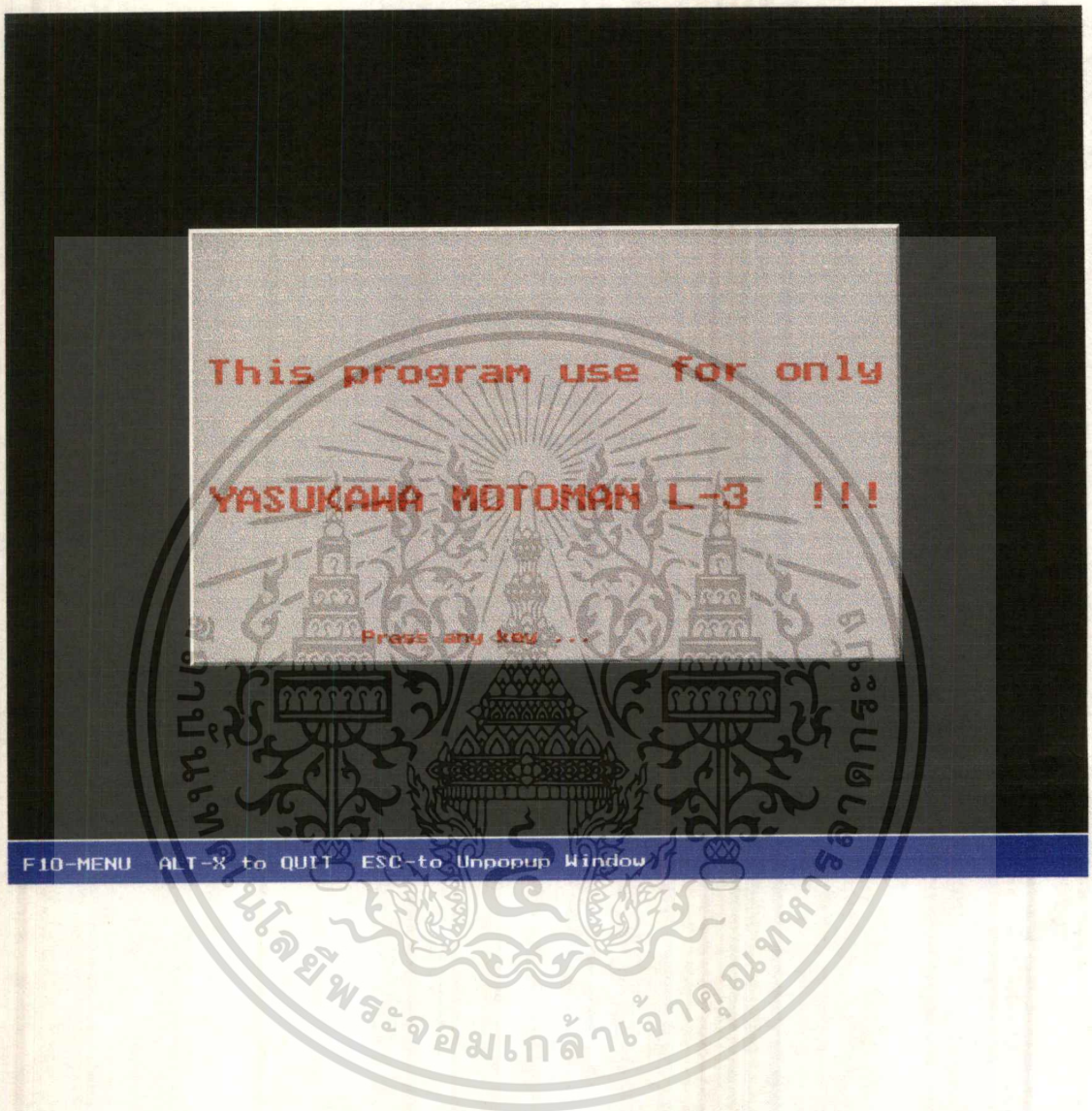
Volume in drive C is MECHANICAL
Volume Serial Number is 3C0D-09F6
Directory of C:\ROBOT

<DIR>          03-07-96   7:49p
<DIR>          03-07-96   7:49p
MOTO1         41 02-29-96   3:06a
MOTOMAN       44 02-28-96  11:33p
MEGAUGA   BGI   5,554 02-18-92   3:00a
DATZ         DAT   42 02-24-96   8:37p
DATA         DAT  113 02-24-96   8:37p
ROBOTZ       EXE  91,071 02-29-96   4:50a
8 file(s)
96,865 bytes
20,959,232 bytes free

C:\ROBOT>robotZ

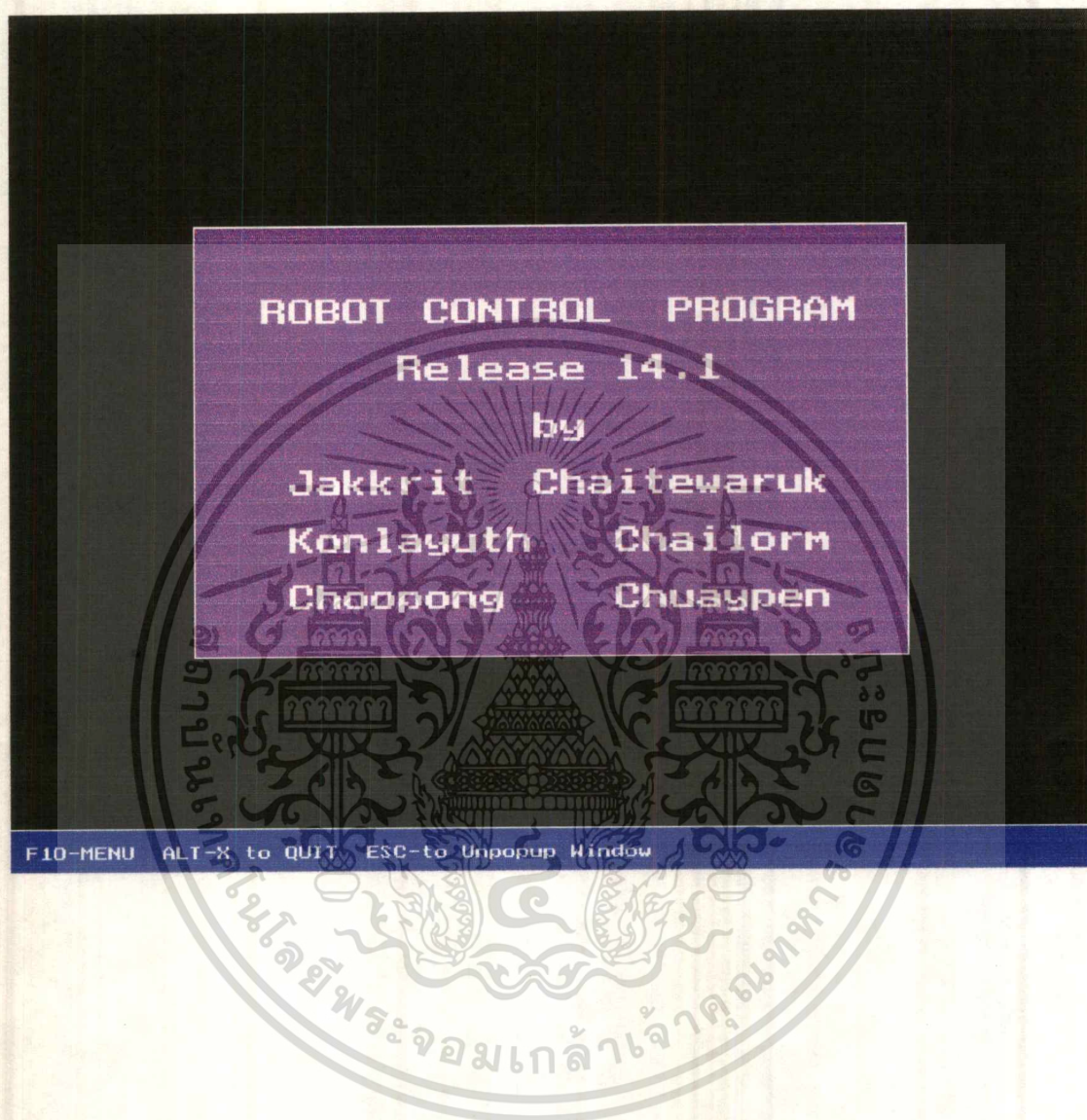
```

6.2.2 เมื่อเข้ามาในโปรแกรม Robot Control จะมีลักษณะดังข้างล่าง



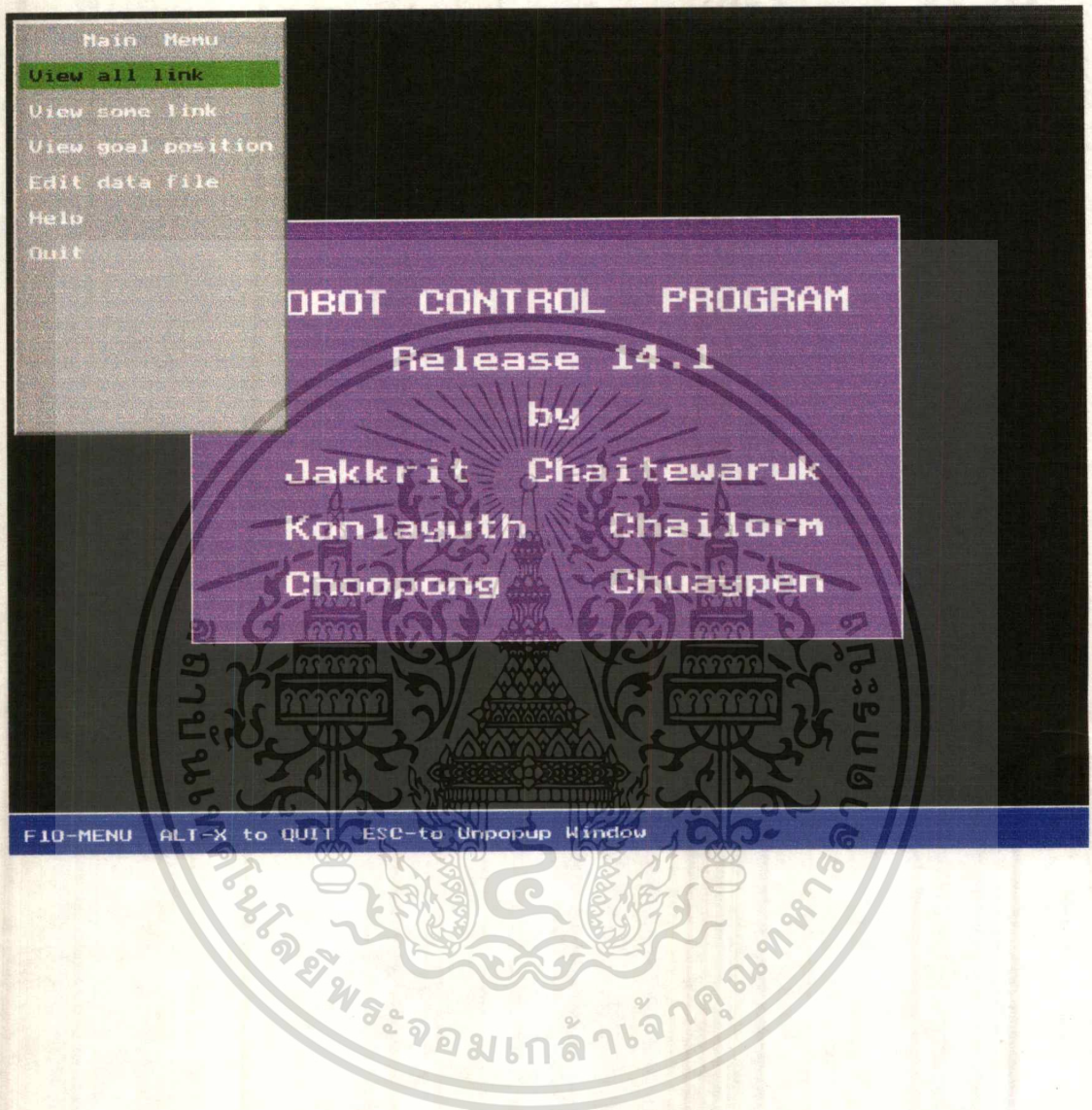
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3 แสดง Environment ของโปรแกรม Robot Control



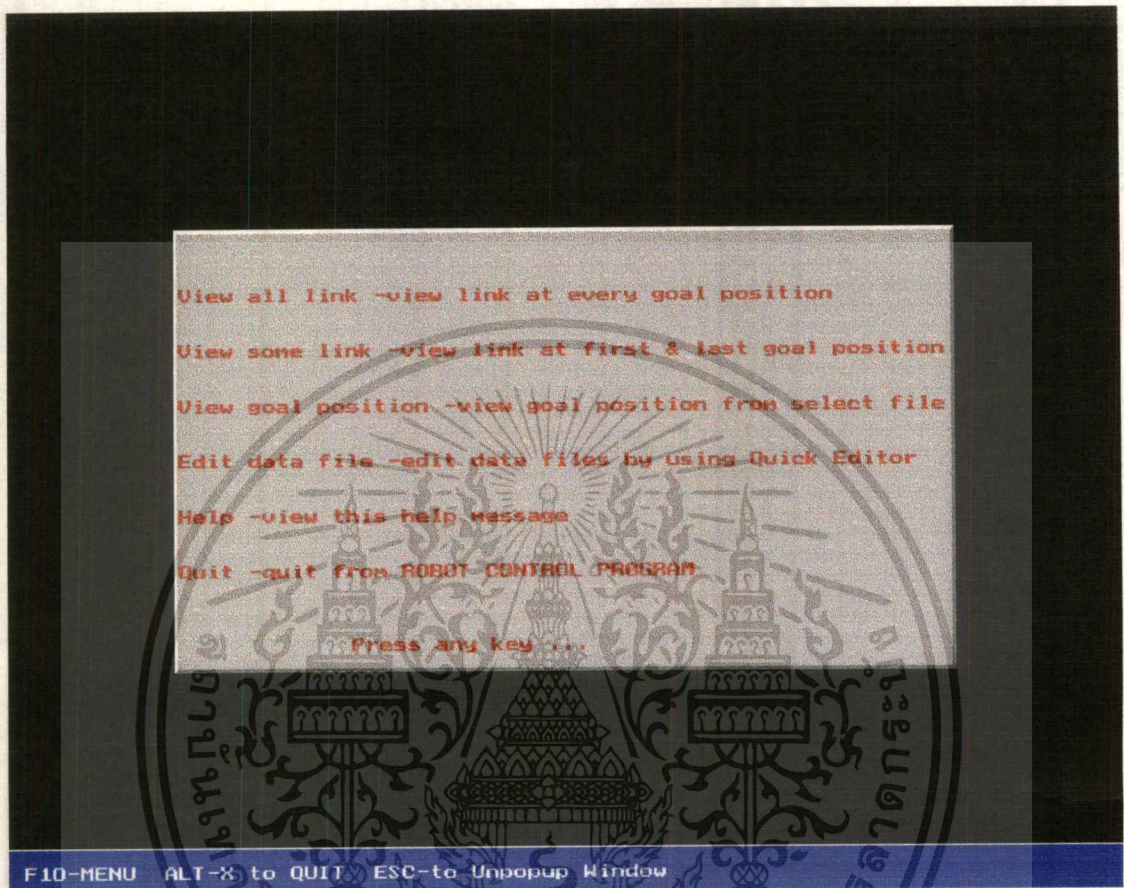
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 แสดง menu function ของโปรแกรม Robot Control



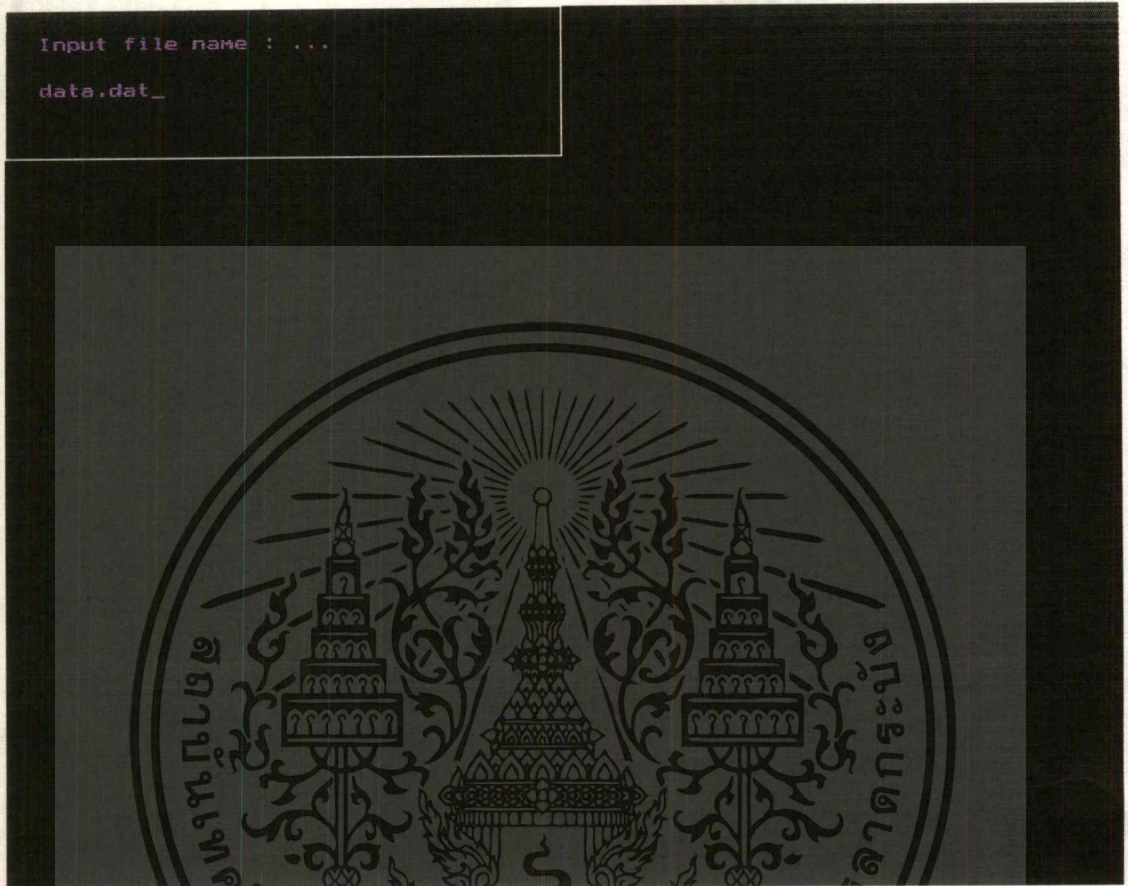
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.6 แสดงข้อความช่วยเหลือจากการเลือก menu Help



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.6 แสดงผลจากการเลือก menu Edit data file



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

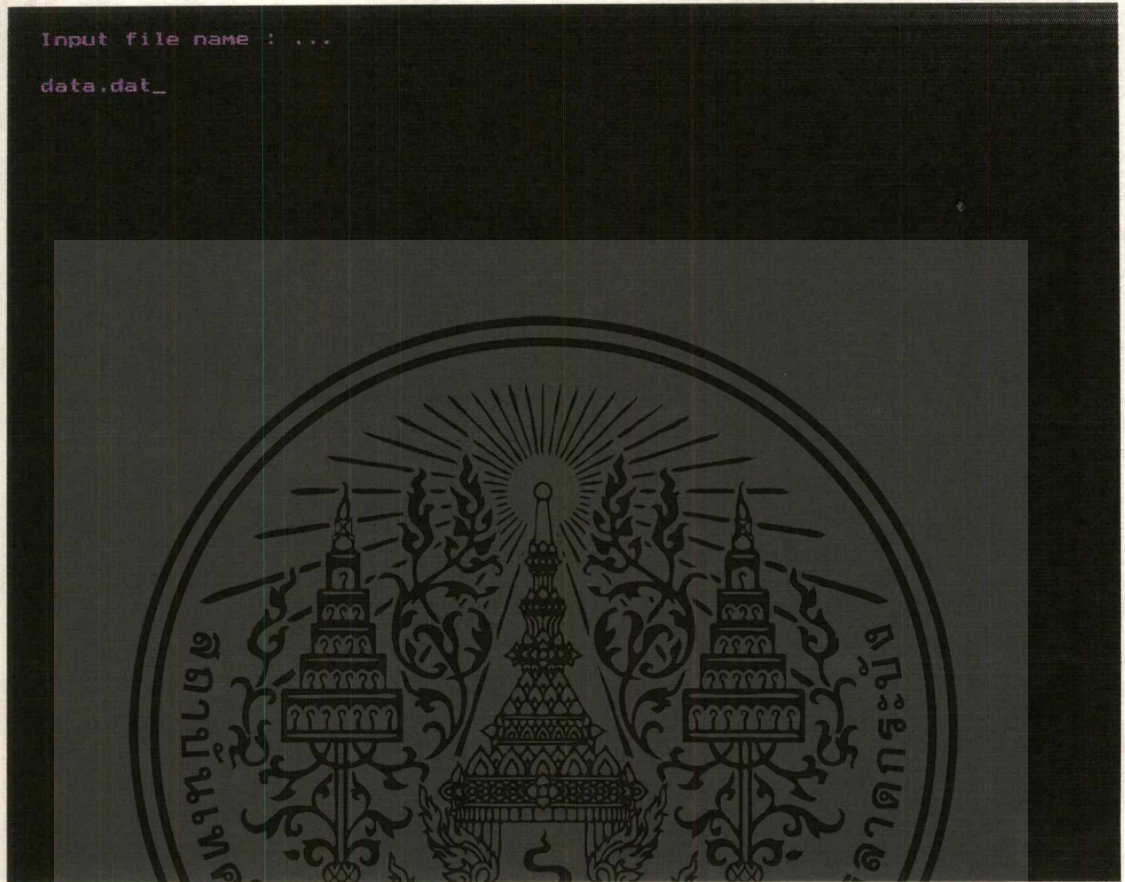
6.2.7 แสดงลักษณะของ data file และโปรแกรม Editor จากการเลือก menu Edit data file

```
L 1      C 1      IA      439k      c:\robot\data.dat
10
1 1 1
10 20 50
40 50 60
70 80 90
35 45 65
40 50 80
66 55 80
50 30 60
44 55 90
56 35 66
45 53 88
```

```
<*** End of File ***>
```

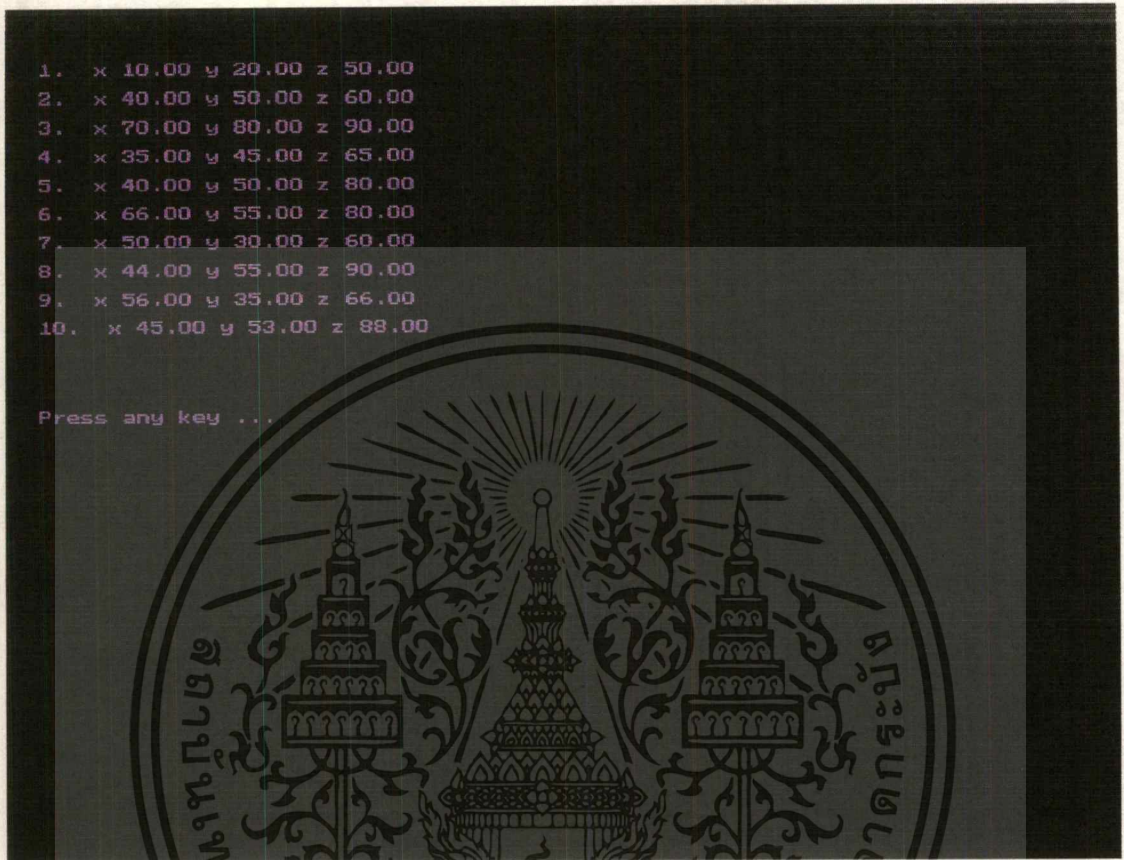


6.2.8 แสดงผลจากการเลือก menu View goal position



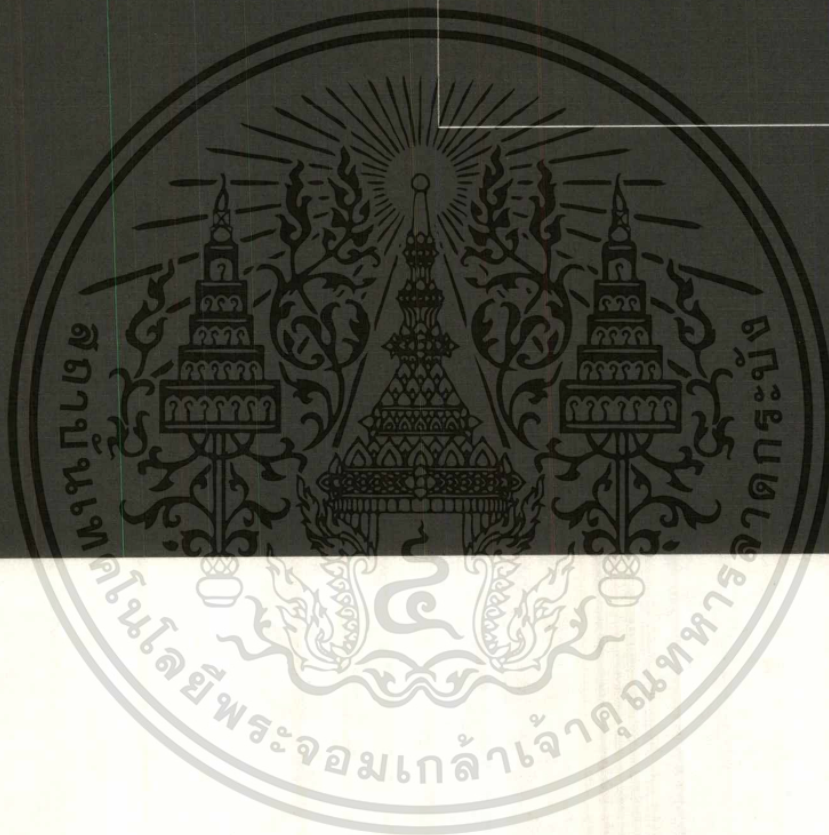
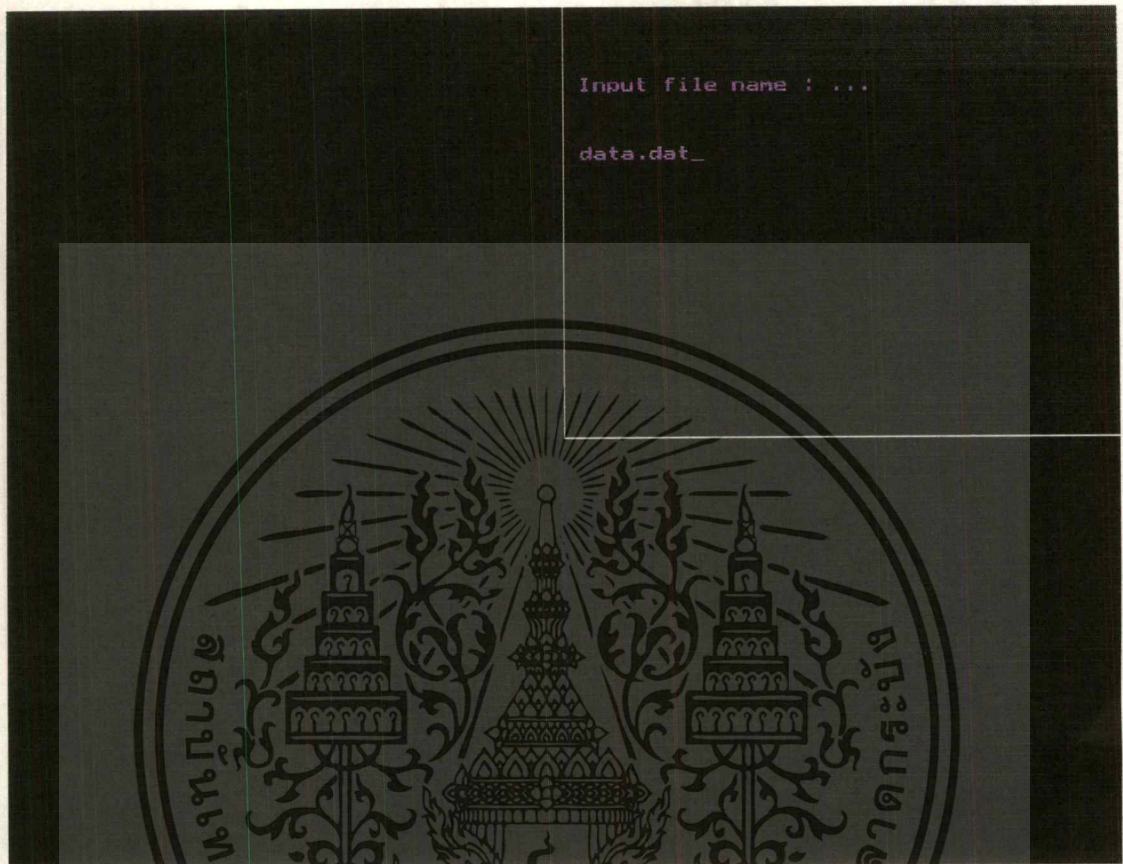
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.9 แสดงผลจากการเลือกและใช้งาน menu View goal position

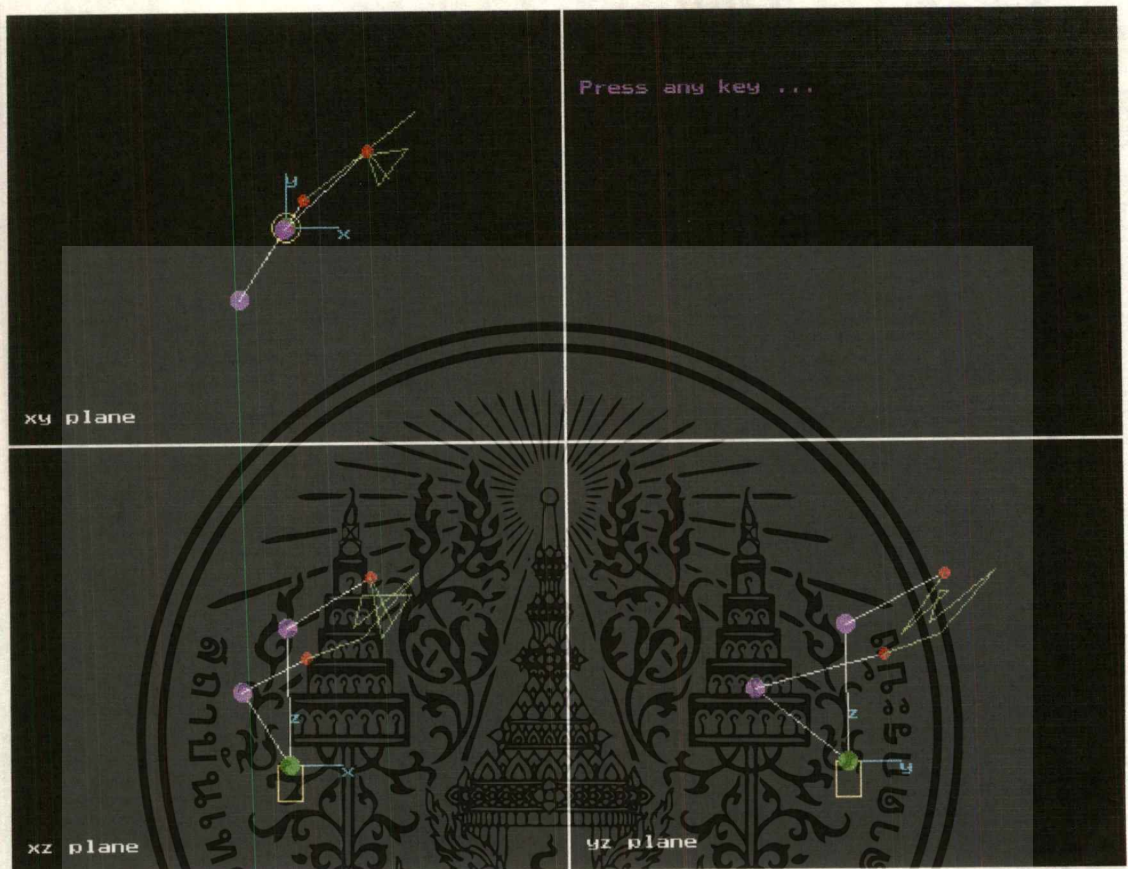


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.10 แสดงผลจากการเลือกใช้ menu View all link และ menu View some link

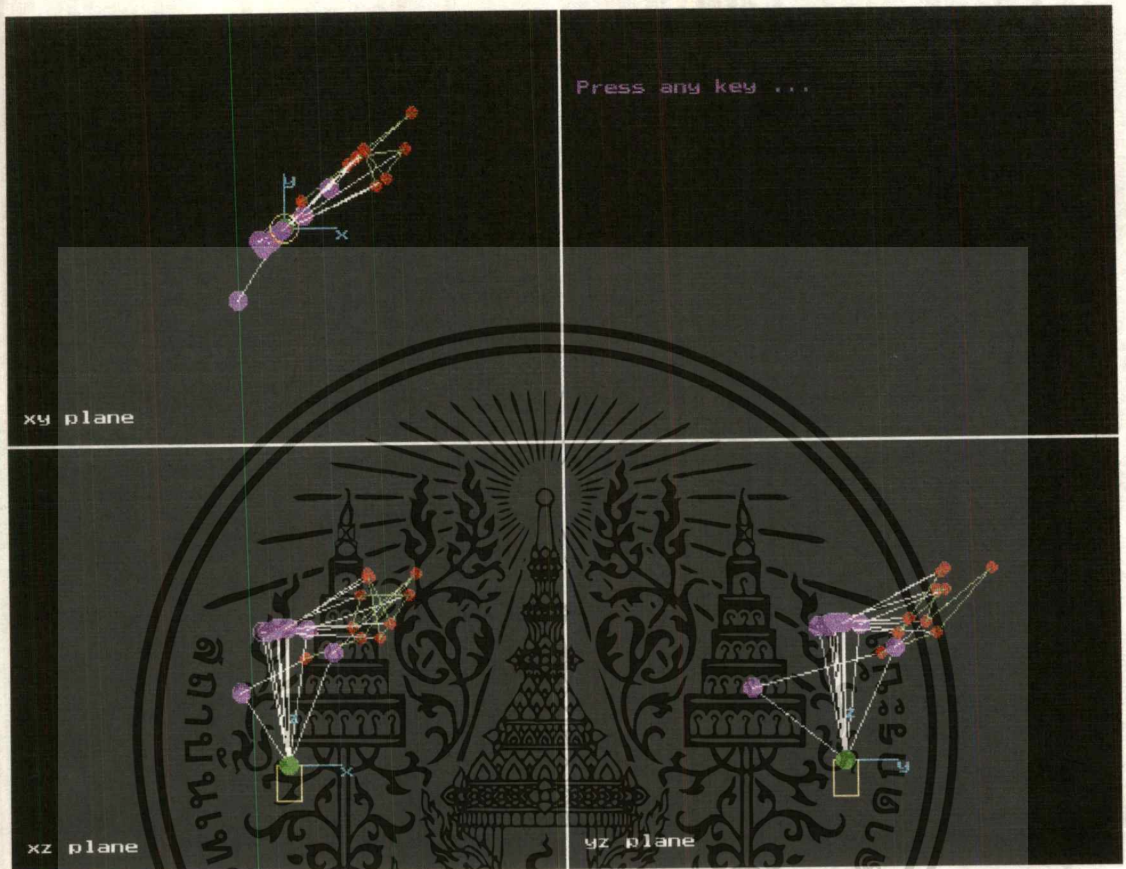


6.2.11 แสดงผลจากการทำงานตาม menu View some link



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.12 แสดงผลจากการทำงานตาม menu View all link



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุปและข้อเสนอแนะ

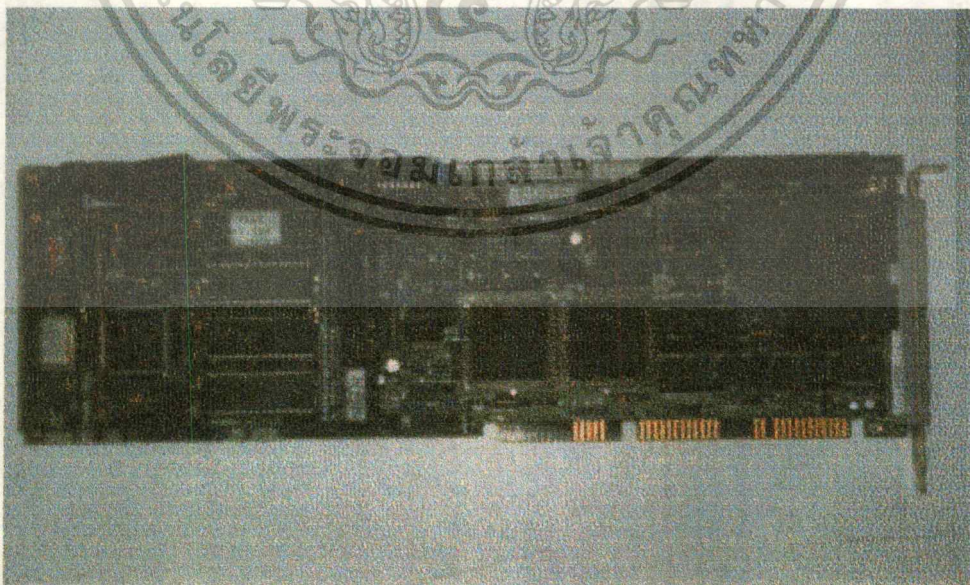
7.1 บทสรุป

ในการทำปริญญานิพนธ์ฉบับนี้ทำให้เราได้ทราบถึงรายละเอียดต่าง ๆ ของ Robot รุ่น Yasukawa Motoman L-3 ซึ่งวิธีการคำนวณต่าง ๆ นั้นยังสามารถนำไปใช้ในการคำนวณในตัว Robot อื่น ๆ ได้อีกด้วย ในขั้นแรกนี้งานหลักที่เราทำคือการสร้างโปรแกรมแสดงการเคลื่อนที่ของ Robot (Robot Simulation) ซึ่งมีชื่อว่า Robot Control ซึ่งการเคลื่อนที่ของ Robot ที่แสดงในโปรแกรมนั้นเราสามารถบอกได้ทันทีว่าถูกต้องแน่นอนเพราะว่าพิกัดต่าง ๆ ของ link ของ Robot ที่แสดงในโปรแกรมนั้นเราคำนวณได้จากสมการ Inverse Kinematics ของ Robot แล้วนำมาแสดงโดยตรงที่จอภาพคอมพิวเตอร์ เราสามารถสังเกตเห็นว่าปลาย link ของ Robot เคลื่อนที่ไปตาม goal position ที่มีอยู่ใน data file ได้จริง ๆ สำหรับงานขั้นต่อไปนั้นเราจะต้องควบคุมตัว Robot จริงพร้อมกับแสดงภาพในจอคอมพิวเตอร์ไปด้วย ซึ่งในการควบคุมจริงนั้นเรายังต้องใช้สมการ Inverse Kinematics ชุดเดิมอยู่ แต่ว่าการเคลื่อนที่ของตัว Robot จริง ๆ นั้นอาจจะเคลื่อนที่ไม่ได้เท่ากับที่แสดงในจอคอมพิวเตอร์หรือไม่ได้เท่ากับที่เราคำนวณจากสมการ Inverse Kinematics ซึ่งอาจจะมีสาเหตุมาจากอุปกรณ์ต่าง ๆ ในระบบการควบคุม Robot ดังนั้นเราจำเป็นที่จะต้องทำการแก้ไขเหตุเหล่านี้ให้ได้เมื่อทุกอย่างพร้อม แล้วจึงจะสามารถพัฒนางานขั้นต่อไปได้

7.2 ข้อเสนอแนะ

7.2.1 เป้าหมายในอนาคต

1. เมื่อได้อุปกรณ์ที่จำเป็น (Amplifier Driver) ครบทั้งหมดแล้วจึงนำมาประกอบเข้ากับตัว Robot ให้สมบูรณ์
2. ทดลองเดินเครื่องเพื่อตรวจสอบว่าอุปกรณ์ทุกส่วนสมบูรณ์หรือไม่
3. เชื่อมต่อ Robot กับ Computer แบบ Close loop (ใช้ Card PMAC Controller) เพื่อควบคุม Robot โดยใช้คอมพิวเตอร์โดยตรง [5]
4. ควบคุม Robot ให้เคลื่อนที่ไปแบบ point to point และแสดงภาพการเคลื่อนที่บนจอภาพคอมพิวเตอร์ ในเวลาเดียวกัน โดยพยายามควบคุมให้เป็นการเคลื่อนที่พร้อมกัน(Simultaneous) ทุกแกนของ Robot
5. ควบคุม Robot ให้เคลื่อนที่ไปตาม path ที่กำหนด(จำนวน point มากกว่า 2) โดยมีลักษณะที่เป็น Smooth curve (Trajectory) [4] ซึ่ง Smooth curve นี้จะ generate จาก Input path ของผู้ใช้ (ยังไม่ใช้ Smooth curve) โดยเงื่อนไขอื่น ๆ เหมือนข้อ 4. ทุกประการ
6. ควบคุม Dynamic force [4] ของ Robot โดยเงื่อนไขอื่น ๆ เหมือนข้อ 5. ทุกประการ



รูปที่ 7.1 แสดง PMAC card version 1.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.2 Algorithm สำหรับ generate trajectory

1. เปิด data file
2. อ่านจำนวนจุด มาเก็บไว้ในตัวแปร n
3. อ่านค่าเวลาที่ใช้ทั้งหมด มาเก็บไว้ในตัวแปร T
4. อ่านค่าตำแหน่ง via point มาไว้ในตัวแปร x_i, y_i, z_i จำนวน n points
5. คำนวณหาค่า $T_i = T/n$

6. นำจุด via point ที่ได้ไปหา θ_i ของ $\theta_1, \dots, \theta_5$

$$\dot{\theta}_0 = 0; \dot{\theta}_n = 0 \quad i=1,2,\dots,n-1$$

$$A = \theta_{i+1} - \theta_i; B = \theta_i - \theta_{i-1}$$

if $(A*B) < 0$ then $\dot{\theta}_i = 0$ else $\dot{\theta}_i = (A+B)/(2T_i)$

7. นำ θ_i มาหา a_1, a_2, a_0, a_3 ของ $\theta_1, \theta_2, \dots, \theta_5$

$$a_0 = \theta_i$$

$$a_1 = \dot{\theta}_i$$

$$a_2 = 3(\theta_{i+1} - \theta_i)/T_i^2 - 2\dot{\theta}_i/T_i - \dot{\theta}_{i+1}/T_i$$

$$a_3 = -2(\theta_{i+1} - \theta_i)/T_i^3 + (\dot{\theta}_{i+1} + \dot{\theta}_i)/T_i^2$$

หลักการ- หาค่ามุมบิดไปของแต่ละแกน x,y,z ทหารด้วยจำนวนจุด n จะได้ orientated ของแต่ละ

Via point

8. เริ่มการเคลื่อนที่

for loop จำนวน n loop

for loop t=0 to T, step ทีละ clock cycle

เพราะฉะนั้น จำนวนรอบ=จำนวน Period จาก t=0 to t=T,

คำนวณหาค่าที่จะป้อนแก่การเคลื่อนที่(Trajectory)

$$\theta_i = a_1 + 2a_2t + 3a_3t^2$$

$$\ddot{\theta}_i = 2a_2 + 6a_3t$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้อน θ_i และ $\bar{\theta}_i$ ของแต่ละ $\theta_1, \theta_2, \dots, \theta_n$ ถ้าเป็นไปได้ควรเป็นแบบขนาน คือป้อนพร้อมกัน

9. จบการเคลื่อนที่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kinematic			Q1	Q2	Q3	Q4	Q5	Tf=10	
1	0	0	10	0.785398	-0.295727	0.578008	1.288516	2.356194	rad.
0	1	0	10	44.98189	-16.93711	33.10409	73.79682	134.9457	deg.
0	0	1	10	0	0	0	0	0	angle velocity
0	0	0	1						
1	0	0	11	0.832981	-0.271274	0.580027	1.262043	2.403778	rad.
0	1	0	10	47.70711	-15.53658	33.21973	72.28064	137.6709	deg.
0	0	1	10	4.757806	-1.555142	3.322524	7.228997	13.75419	angle velocity
0	0	0	1						
1	0	0	12	0.876058	-0.247338	0.582239	1.235896	2.446854	rad.
0	1	0	10	50.17423	-14.16573	33.34639	70.78313	140.138	deg.
0	0	1	10	5.005871	-1.41853	3.33519	7.079719	14.00225	angle velocity
0	0	0	1						
1	0	0	13	0.915101	-0.224086	0.584642	1.21024	2.485897	rad.
0	1	0	10	52.41031	-12.83402	33.48407	69.31374	142.3741	deg.
0	0	1	10	5.230732	-1.285659	3.348958	6.933081	14.22711	angle velocity
0	0	0	1						
1	0	0	14	0.950547	-0.201622	0.587239	1.18518	2.521343	rad.
0	1	0	10	54.44041	-11.54745	33.63276	67.87848	144.4042	deg.
0	0	1	10	5.434879	-1.157171	3.363827	6.789724	14.43126	angle velocity
0	0	0	1						
1	0	0	15	0.982794	-0.180005	0.590027	1.160775	2.55359	rad.
0	1	0	10	56.28728	-10.3094	33.79246	66.48073	146.2511	deg.
0	0	1	10	5.620585	-1.033435	3.379797	6.650017	14.61696	angle velocity
0	0	0	1						
1	0	0	16	1.012197	-0.15926	0.593008	1.137048	2.582993	rad.
0	1	0	10	57.97128	-9.121247	33.96318	65.12186	147.9351	deg.
0	0	1	10	5.789889	-0.914616	3.396868	6.514126	14.78627	angle velocity
0	0	0	1						
1	0	0	17	1.039072	-0.139384	0.596181	1.114	2.609869	rad.
0	1	0	10	59.5105	-7.982911	34.14491	63.80179	149.4743	deg.
0	0	1	10	5.944608	-0.800728	3.415042	6.382065	14.94099	angle velocity
0	0	0	1						

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกที่รูปที่ 7.2.1 แสดงผลลัพธ์จากการ Generate Trajectory ทุกครั้งที่มีการนำไปใช้

Kinematic			Q1	Q2	Q3	Q4	Q5	Tf=10
1	0	0	18	1.063698	-0.120359	0.599546	1.091609	2.634494 rad.
0	1	0	10	60.92088	-6.893303	34.33765	62.51944	150.8847 deg.
0	0	1	10	6.086346	-0.691677	3.434316	6.25374	15.08272 angle velocity
0	0	0	1					
1	0	0	19	1.086318	-0.102154	0.603104	1.069846	2.657115 rad.
0	1	0	10	62.21642	-5.850639	34.54141	61.27302	152.1802 deg.
0	0	1	10	6.216515	-0.5873	3.454692	6.128987	15.21289 angle velocity
0	0	0	1					
1	0	0	20	1.107149	-0.084729	0.606854	1.048672	2.677945 rad.
0	1	0	10	63.40943	-4.852689	34.75618	60.06029	153.3732 deg.
0	0	1	10	6.336355	-0.48738	3.476169	6.00759	15.33273 angle velocity
0	0	0	1					
1	0	0	21	1.126377	-0.068042	0.610796	1.028042	2.697173 rad.
0	1	0	10	64.51069	-3.896962	34.98197	58.87878	154.4745 deg.
0	0	1	10	-3.168178	0.24369	-1.738085	-3.003795	-7.666367 angle velocity
0	0	0	1					
1	0	0	22	1.144169	-0.052046	0.614931	1.016911	2.717314 rad.
0	1	0	10	65.52967	-2.98084	35.21877	57.72586	155.4935 deg.
0	0	1	10	0	0	0	0	0 angle velocity
0	0	0	1					

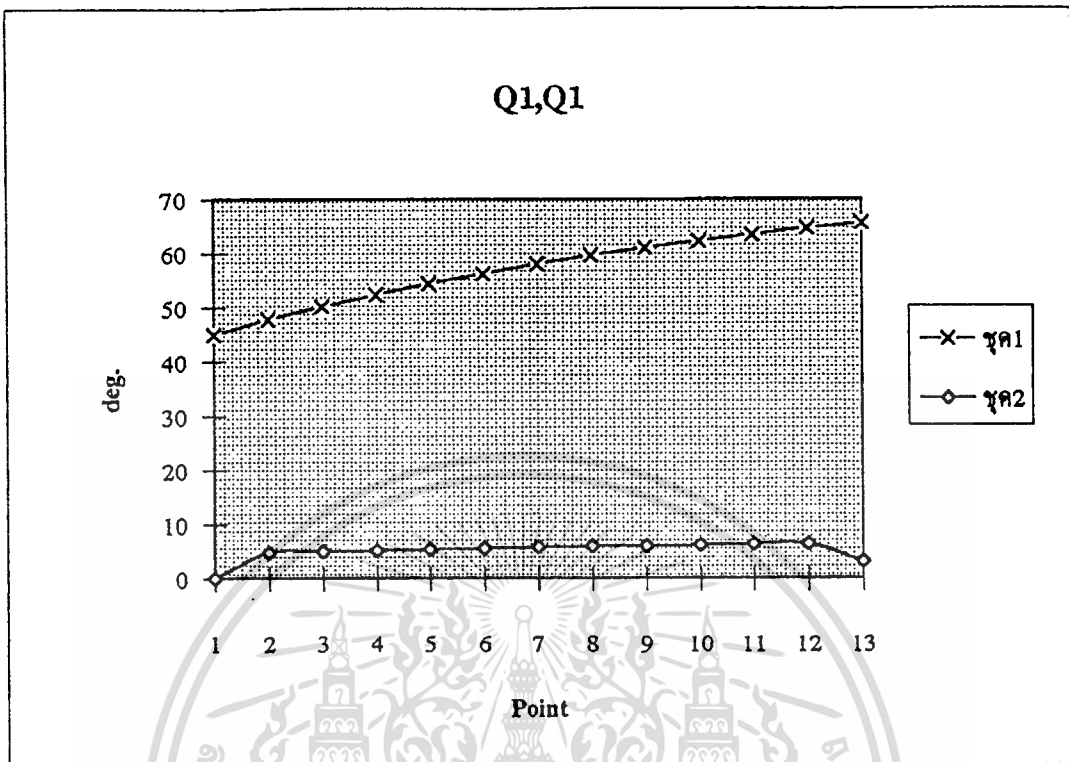
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกรูปที่ 7.2.2 แสดงผลลัพธ์จากการ Generate Trajectory การทุกครั้งที่มีการนำไปใช้

Q1	Q2	Q3	Q4	Q5	
0.785398	-0.295727	0.578008	1.288516	2.356194	a0
0	0	0	0	0	a1
-0.394024	0.19753	-0.328783	-0.768385	-1.293662	a2
0.042128	-0.018352	0.032994	0.075322	0.132091	a3
0.832981	-0.271274	0.580027	1.262043	2.403778	a0
4.757806	-1.555142	3.322524	7.228997	13.75419	a1
-1.378135	0.494007	-0.994224	-2.198697	-4.077048	a2
-0.002454	-0.001376	-0.000127	0.001502	-0.002454	a3
0.876058	-0.247338	0.582239	1.235896	2.446854	a0
5.005871	-1.41853	3.33519	7.079719	14.00225	a1
-1.457165	0.452223	-0.997804	-2.153333	-4.156079	a2
-0.002224	-0.001335	-0.000138	0.001472	-0.002224	a3
0.915101	-0.224086	0.584642	1.21024	2.485897	a0
5.230732	-1.285659	3.348958	6.933081	14.22711	a1
-1.528731	0.411446	-1.001714	-2.108646	-4.227645	a2
-0.002019	-0.001288	-0.000149	0.001437	-0.002019	a3
0.950547	-0.201622	0.587239	1.18518	2.521343	a0
5.434879	-1.157171	3.363827	6.789724	14.43126	a1
-1.593628	0.371919	-1.005954	-2.064879	-4.292542	a2
-0.001837	-0.001239	-0.00016	0.001398	-0.001837	a3
0.982794	-0.180005	0.590027	1.160775	2.55359	a0
5.620585	-1.033435	3.379797	6.650017	14.61696	a1
-1.652586	0.333793	-1.010525	-2.022182	-4.351499	a2
-0.001675	-0.001188	-0.000171	0.001359	-0.001675	a3
1.012197	-0.15926	0.593008	1.137048	2.582993	a0
5.789889	-0.914616	3.396868	6.514126	14.78627	a1
-1.706262	0.297146	-1.015426	-1.980634	-4.405176	a2
-0.001531	-0.001138	-0.000182	0.00132	-0.001531	a3
1.039072	-0.139384	0.596181	1.114	2.609869	a0
5.944608	-0.800728	3.415042	6.382065	14.94099	a1
-1.755245	0.262001	-1.020658	-1.940258	-4.454159	a2
-0.001403	-0.001089	-0.000193	0.001281	-0.001403	a3

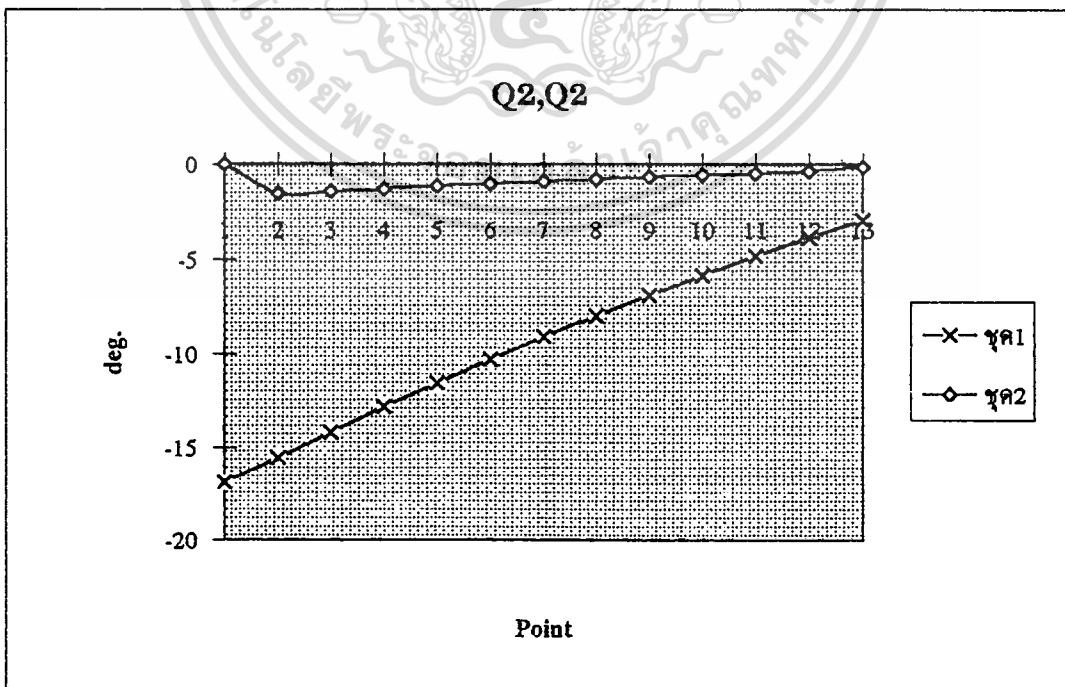
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 7.2.3 แสดงผลลัพธ์จากการ Generate Trajectory
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Q1	Q2	Q3	Q4	Q5	
1.063698	-0.120359	0.599546	1.091609	2.634494	a0
6.086346	-0.691677	3.434316	6.25374	15.08272	a1
-1.800054	0.228345	-1.02622	-1.901039	-4.498968	a2
-0.001289	-0.001042	-0.000204	0.001245	-0.001289	a3
1.086318	-0.102154	0.603104	1.069846	2.657115	a0
6.216515	-0.5873	3.454692	6.128987	15.21289	a1
-1.841148	0.196136	-1.032112	-1.862938	-4.540062	a2
-0.001188	-0.000997	-0.000215	0.001211	-0.001188	a3
1.107149	-0.084729	0.606854	1.048672	2.677945	a0
6.336355	-0.48738	3.476169	6.00759	15.33273	a1
-0.917415	0.101779	-0.514652	-0.936584	-2.266872	a2
-0.097248	0.005399	-0.052594	-0.087751	-0.232194	a3
1.126377	-0.068042	0.610796	1.028042	2.697173	a0
6.446955	-0.391676	3.498748	5.889308	15.44333	a1
0.664205	-0.021254	0.354721	0.566171	1.563843	a2
0.029644	-0.004269	0.016907	0.032344	0.074626	a3
1.144169	-0.052046	0.614931	1.007912	2.714965	a0
3.225534	-0.194848	1.749099	2.943939	7.723724	a1
-1.96589	0.089425	-1.056563	-1.731776	-4.664804	a2
0.131059	-0.005962	0.070438	0.115452	0.310987	a3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกที่ 7.2.4 แสดงผลลัพธ์จากการ Generate Trajectory การทุกครั้งที่มีการนำไปใช้

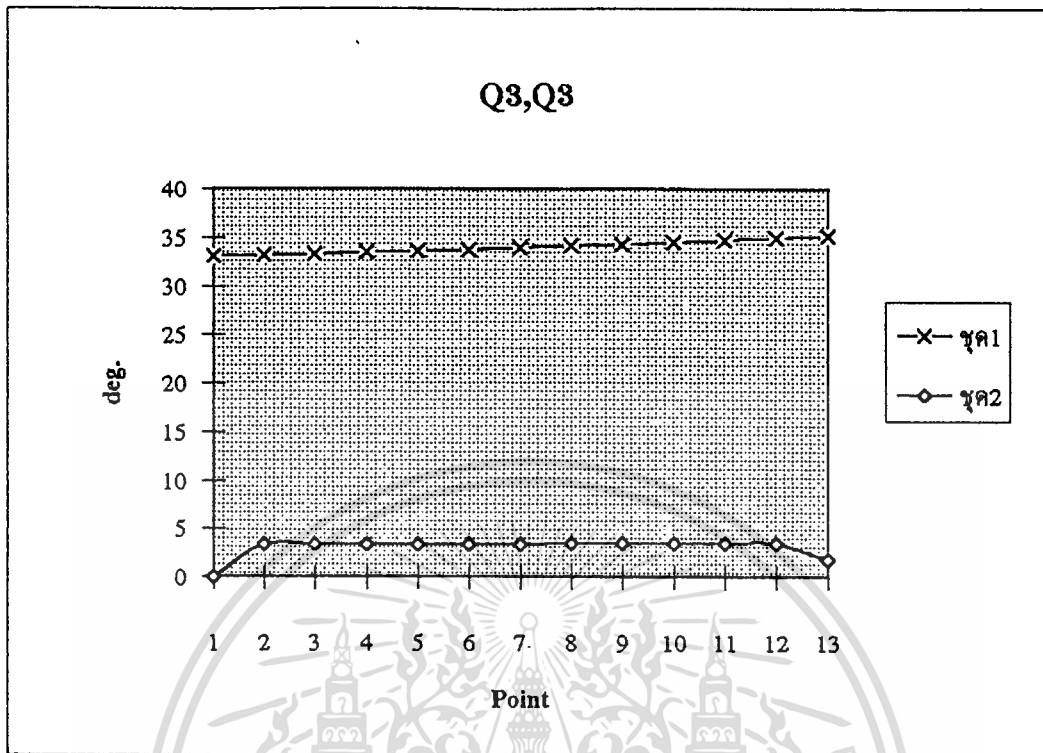


รูปที่ 7.2.5 แสดงผลลัพธ์จากการ Generate Trajectory

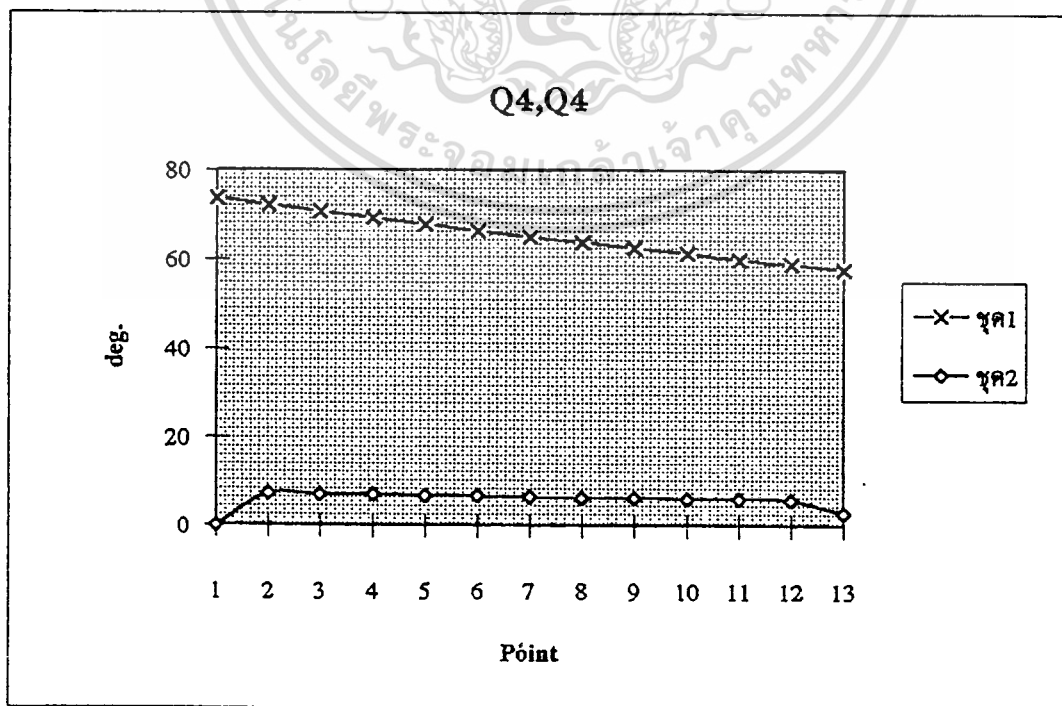


รูปที่ 7.2.6 แสดงผลลัพธ์จากการ Generate Trajectory

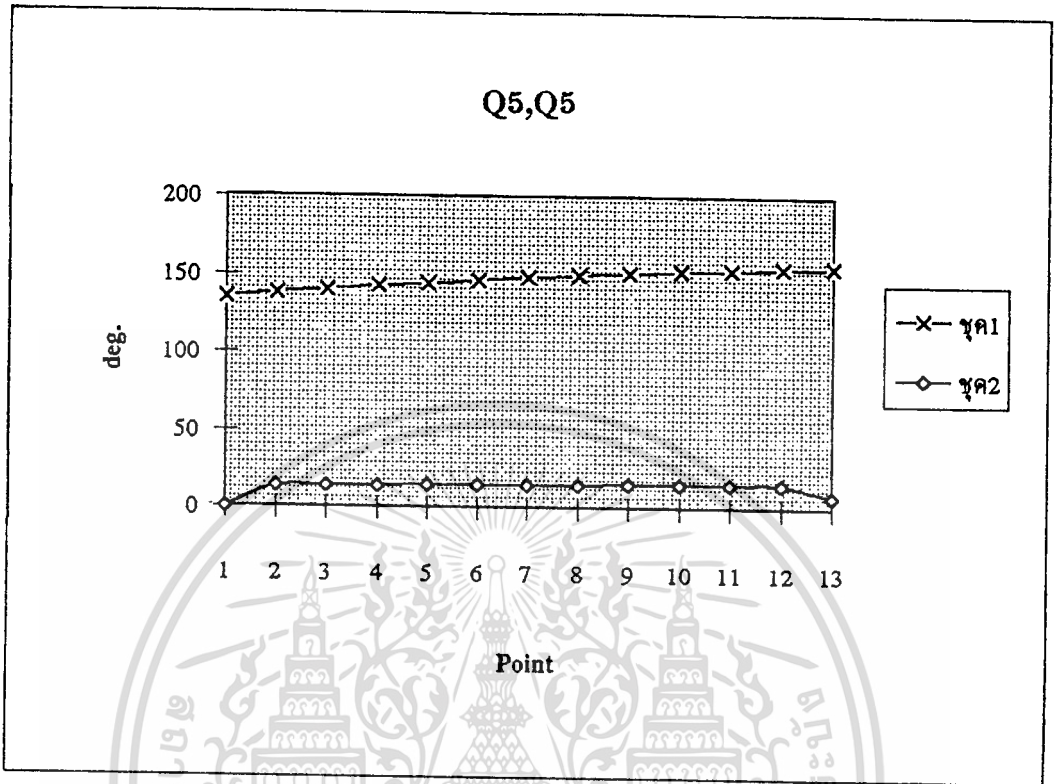
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.2.7 แสดงผลลัพธ์จากการ Generate Trajectory



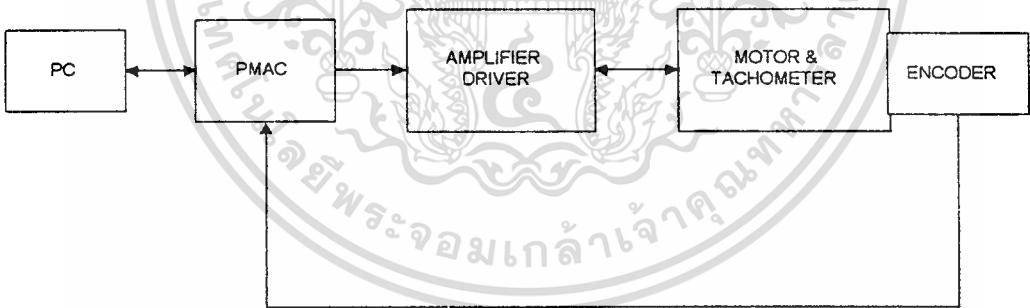
รูปที่ 7.2.8 แสดงผลลัพธ์จากการ Generate Trajectory



รูปที่ 7.2.9 แสดงผลลัพธ์จากการ Generate Trajectory

7.2.3 อุปกรณ์ควบคุมการเคลื่อนที่ของ Robot

ในการควบคุม Robot เราจะใช้การควบคุมที่มีชื่อว่า PMAC (Programmable Multi-Axis Controller) ซึ่งเป็นผลิตภัณฑ์ของ DELTA TAU DATA SYSTEM, INC. PMAC เป็น Controller ที่ยืดหยุ่นมากสามารถใช้ในงานด้านต่าง ๆ ได้มากมาย การ Setup สามารถทำการ Setup ได้ทั้งที่ตัวการ์ดเองและใช้ Software Setup การติดตั้งตัวการ์ด PMAC จะติดตั้งในตัว PC (Personal Computer) โดยใช้ระบบ BUS ซึ่ง PMAC จะติดต่อกับ PC ที่ Address 528 (210H) และ PMAC จะติดต่อกับระบบ (Robot) โดยใช้อุปกรณ์ที่เรียกว่า Terminal Block Board (ACC - 8D) ซึ่งเป็นผลิตภัณฑ์ที่ช่วยเสริมการทำงานของ PMAC ให้ยืดหยุ่นมากยิ่งขึ้น โดย PMAC จะสามารถควบคุมมอเตอร์พร้อม ๆ กันได้ถึง 8 ตัว หรือควบคุม Robot ได้พร้อม ๆ กันถึง 8 แกน สำหรับวิธีการควบคุมเราสามารถเขียนเป็น Block Diagram ได้ดังนี้



รูปที่ 7.3 แสดงระบบควบคุมมอเตอร์โดยใช้ PMAC

PMAC จะรับ Supply voltage $\pm 12 - \pm 15$ volts โดยจะรับจาก Servo Amplifier หรือ PC BUS ก็ได้ สำหรับ PMAC นี้จะใช้ CPU ของ Motorola รุ่น DSP 56001 (Digital Signal Processor) ในการคำนวณการเคลื่อนที่ของ Robot ทั้ง 8 แกน PMAC สามารถที่จะนำไปใช้งานด้านต่าง ๆ ตามต้องการได้อย่างง่ายดาย โดยการ Set up ที่ตัวการ์ดได้ ,การจัด Parameter ต่าง ๆ โดยทาง Software และ Hardware Setup,การเขียนโปรแกรมการเคลื่อนที่ และ การเขียน PLC Program โดยงานแต่ละอย่างที่กล่าวมาสามารถควบคุมมอเตอร์ได้ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้

พร้อมกันถึง 8 ตัว สำหรับสัญญาณที่ออกจาก PMAC จะเป็นสัญญาณ Analog โดยอัตโนมัติ ซึ่งในตัว PMAC จะมี DAC (Digital to Analog Converter) สำหรับแปลงสัญญาณ Digital ให้เป็นสัญญาณ Analog เพื่อนำไปใช้งานได้ในทันที โดยสัญญาณจะอยู่ในช่วง $-32,768$ ถึง $+32,767$ (16 bit สำหรับ ± 10 volts)

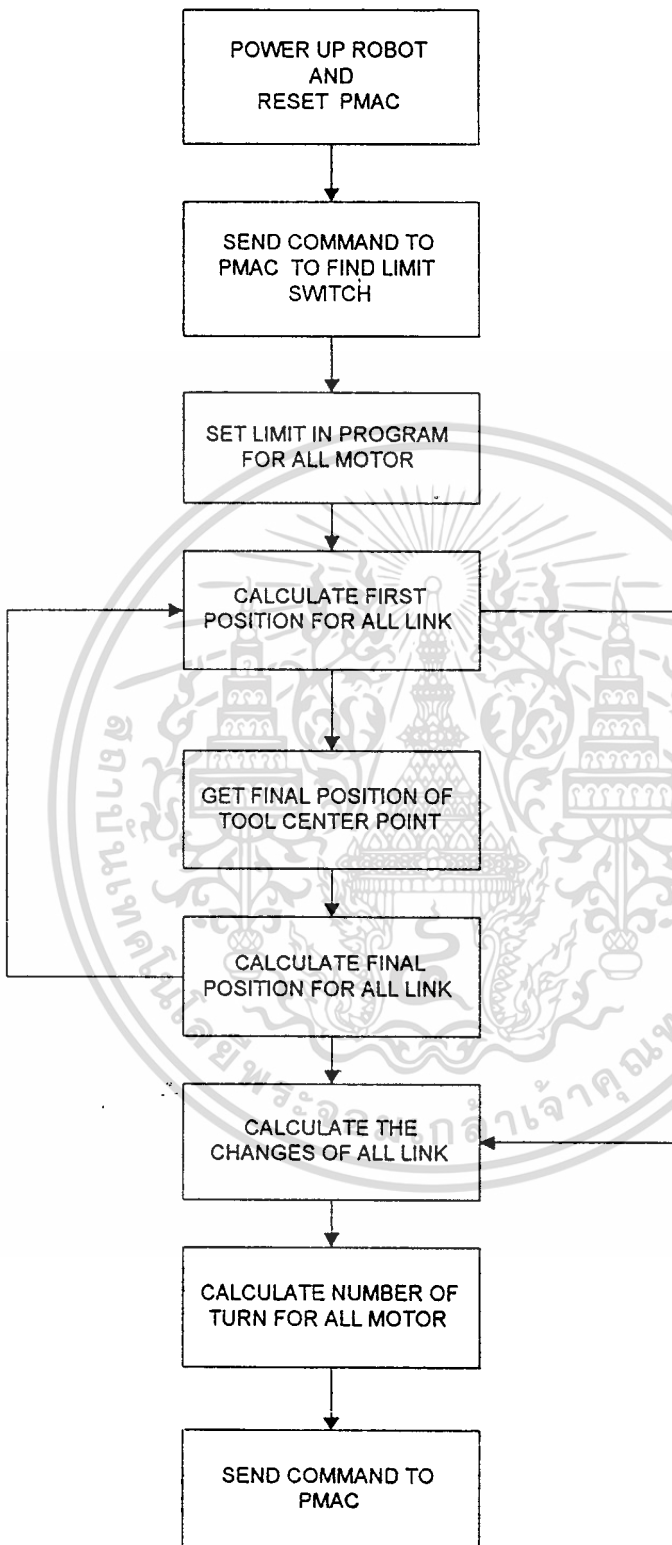
7.2.4 การส่งคำสั่งติดต่อกับ PMAC

PMAC สามารถทำงานต่าง ๆ ได้โดยการส่งคำสั่งเป็น ASCII Code โดยจะเป็นลักษณะของ Text Strings ได้ และเราสามารถรับการตอบสนองต่อคำสั่งของ PMAC ในรูปของ ASCII Text Strings ได้เช่นกัน การใช้ PMAC ควบคุมมอเตอร์จะทำในลักษณะ Closing Servo Loop โดยอัตโนมัติ ซึ่งมีจุดประสงค์เพื่อที่จะควบคุมให้ได้ตามตำแหน่งที่ต้องการ (ตำแหน่งจริงของมอเตอร์ต้องเท่ากับตำแหน่งที่เราสั่งเข้าไป) การควบคุมของ PMAC จะเป็นแบบ PID Control (Proportional Integral Derivatives Control) โดยที่ตัว PMAC เองจะสามารถทำการ "Auto-Tuning" โดยการคำนวณหาค่า Gain ที่เหมาะสมให้แก่การควบคุมมอเตอร์ได้โดยอัตโนมัติ

7.2.5 การเขียนโปรแกรมควบคุม Robot โดยใช้ PMAC

มีหลักการควบคุมที่สำคัญคือ การควบคุมการหมุนของตัวมอเตอร์เพื่อขับ Load ให้ได้ตามตำแหน่งที่ต้องการ โดยอาจจะเป็นการขับ Ball Screw, โซ่ หรือสายพาน ที่จะบังคับ Link ต่าง ๆ ของ Robot ให้เคลื่อนที่ไปตามตำแหน่งต่าง ๆ ที่ต้องการ โดยตำแหน่งที่ Link เหล่านี้เราสามารถคำนวณได้จาก Forward Kinematic และ Inverse Kinematic ของ Robot ตัวนั้น ๆ เมื่อได้ตำแหน่งที่จะเคลื่อนที่ไปแล้ว ก็จะนำมาแปลงเป็นคำสั่งควบคุมของ PMAC แล้วทำการส่งคำสั่งควบคุมนี้เข้าไปในตัว PMAC เพื่อให้ PMAC ควบคุมมอเตอร์ให้หมุนตามจำนวนรอบที่เราคำนวณได้ และเมื่อต้องการให้ Robot เคลื่อนที่ไปจุดอื่น ๆ อีก ก็จะทำตามวิธีการที่กล่าวมาข้างต้นนี้อีกครั้ง ซึ่งเขียนเป็น Block Diagram ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 แสดงวิธีการควบคุมตำแหน่งของ Robotic โดยใช้ PMAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] ถันวา ศรีประโมง, "การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม", มหาวิทยาลัยเทคโนโลยีมหานคร, กรุงเทพฯ, 2537.
- [2] สมพัฒน์ รุ่งตะวันเรืองศรี, "เรียนรู้คอมพิวเตอร์กราฟิกส์ 2 มิติ ด้วยภาษา C", ซีเอ็ดยูเคชั่น, กรุงเทพฯ, 2535.
- [3] Bottema, O. , "Theoretical Kinematics" , North Holland , Amsterdam, 1979.
- [4] Craig, John J. " Introduction to Robotics" , Addison - Wesley , USA , 1986 .
- [5] DELTA TAU DATA SYSTEM , INC . " PMAC 's USER MANUAL AND SOFTWARE REFERENCE" , USA , 1993 .
- [6] Fu, K.S. "Robotics Control, Sensing, Vision, and Intelligence" , McGraw-Hill, USA, 1987.
- [7] Heiny, Loren. "Power Graphics Using Turbo C++" , Colioris, USA, 1994.
- [8] Kheir, Naim A., "System Modeling and Computer Simulation" , Marcel Dekker, Inc. , USA, 1996.
- [9] Murray, Richard M., "A Mathematical Introduction to Robotic Manipulation" , CRC Press, Inc. , USA, 1994.