

เทคนิคอย่างง่ายในการสร้างโมเดลและจำลองสำหรับการควบคุม
แบบแขนกล 4 แกน

A SIMPLE TECHNIQUE IN THE SIMULATION AND MODELING OF
FOUR-AXIS ROBOT-ARM CONTROL.

เจตติยะ ศรีพทกษ
JETTIYA SRIPTUK

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2551

KMITL-2008-EN-M-060-024

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เทคนิคอย่างง่ายในการสร้างโมเดลและจำลองสำหรับการควบคุม
แบบแขนกล 4 แกน

A SIMPLE TECHNIQUE IN THE SIMULATION AND MODELING OF
FOUR-AXIS ROBOT-ARM CONTROL

เจตติยะ ศรีพิทักษ์
JETTIYA SRIPITUK

เลขหมู่.....
เลขทะเบียน..... 79820
วัน,เดือน,ปี..... 18 ส.ย. 2551

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2551
KMITL-2008-EN-M-060-024

**A SIMPLE TECHNIQUE IN THE SIMULATION AND MODELING OF
FOUR-AXIS ROBOT-ARM CONTROL**

JETTIYA SRIPITUK

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

2008

KMITL-2008-EN-M-060-024

COPYRIGHT 2008

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	เทคนิคอย่างง่ายในการสร้างโมเดลและจำลองสำหรับการควบคุมแบบแขนกล 4 แกน
นักศึกษา	นายเจตติยะ ศรีพิทักษ์
รหัสประจำตัว	46068002
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
พ.ศ.	2551
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ประภาส อุคคกิม่าพันธุ์

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้แสดงถึงการสร้างโมเดลและโปรแกรมการจำลองแบบเวลาจริงโดยได้ออกแบบและพัฒนาสำหรับชุดควบคุมแขนกล โดยเทคนิคการวิเคราะห์หึ่งถึงเส้นทางเดินที่ปลอดภัยของพื้นที่การทำงานและลักษณะการเคลื่อนที่แบบจุดถึงจุดของแขนกลด้วยหลักกลศาสตร์ การเคลื่อนที่โดยวิธีตรง และวิธีย้อนกลับซึ่งเป็นเทคนิคการเขียนเป็นชุดคำสั่งควบคุมในโปรแกรมเมทแลปมาช่วยอธิบายโดยแสดงผลออกเป็นภาพเคลื่อนไหวในเมทแลป ซึ่งทำให้ทราบถึงเส้นทางเดินก่อนการส่งคำสั่งให้แขนกล ทั้งในโหมดควบคุมด้วยมือและระบบอัตโนมัติจากคั่นบังคับควบคุมผ่านทางเครื่องคอมพิวเตอร์

Thesis Title A Simple Technique in the Simulation and Modeling of Four-Axis Robot-Arm Control
Student Mr. Jettiya Sripituk
Student ID. 46068002
Degree Master of Engineering
Program Instrumentation Engineering
Year 2008
Thesis Advisor Assoc.Prof Prapart Ukakimapurn

ABSTRACT

This thesis presents the modeling and real-time simulation programs that were developed for the robot arm control. The technique analyzes the safety path, working area and point to point motion of robot. The implementation of experimental forward and inverse kinematics control method as Matlab m-files are described. The visualization and animation capabilities of Matlab provide a realistic perception of behavior of the test beds without robot before send the command to robot. Both the manual and automatic control is to control by using the joystick and library file via PC respectively.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้เป็นอย่างดีด้วยความกรุณาจากอาจารย์ปรีक्षा รศ. ประภาส อุคคกิมพันธ์ ที่ให้คำชี้แนะช่วยแก้ไขปัญหาลดจนองค์ความรู้และประสบการณ์ที่ดี มาโดยตลอด ผู้วิจัยขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ธีรวัฒน์ เทพมณี ที่คอยให้การช่วยเหลือเกี่ยวกับแนวทางการ ออกแบบโปรแกรมแม่ทแยงที่ใช้ในการทำวิทยานิพนธ์ฉบับนี้ ขอขอบพระคุณ รศ.ไสว พงศ์ สวัสดิ์ ที่ให้คำแนะนำเกี่ยวกับการทำวิจัยมาโดยตลอดและกราบขอบพระคุณ คุณพ่อ คุณแม่ และผู้มีพระคุณทุกท่านที่ได้เป็นกำลังใจและสนับสนุนในการศึกษาโดยตลอด ขอขอบคุณ คุณ ชนิตา อินทวิเศษ ที่คอยช่วยเหลือ ดูแลและเป็นกำลังใจให้มาโดยตลอด

อาจารย์ในภาควิชาเพื่อนๆตลอดจนคณะกรรมการสอบที่ช่วยชี้แนะและตรวจทาน วิทยานิพนธ์นี้จนสำเร็จลุล่วงด้วยดี

เจตติยะ ศรีพิทักษ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	IX
สารบัญตาราง.....	XII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย.....	2
1.5 ขอบเขตการศึกษา.....	2
1.6 ขั้นตอนการศึกษา.....	2
บทที่ 2 ทฤษฎีพื้นฐานของการวิเคราะห์และควบคุมหุ่นยนต์.....	3
2.1 บทนำ.....	3
2.1.1 การแบ่งประเภทของแขนกล.....	4
2.1.1.1 Cartesian (Gantry) Robot.....	5
2.1.1.2 Cylindrical.....	6
2.1.1.3 Spherical Robot (Polar).....	7
2.1.1.4 SCARA Robot.....	8
2.1.1.5 Articulates Arm (Revolute).....	9
2.1.2 ลักษณะเฉพาะของหุ่นยนต์ (Robot Specification).....	10
2.1.3 ความแม่นยำของการเคลื่อนที่.....	14
2.1.3.1 Spatial Resolution.....	14
2.1.3.2 Accuracy.....	14
2.1.3.3 Repeatability.....	14

สารบัญ(ต่อ)

	หน้า
2.1.4 การเคลื่อนที่ของหุ่นยนต์.....	14
2.1.5 คุณสมบัติอื่น ๆ ของหุ่นยนต์อุตสาหกรรมทั่วไป.....	15
2.1.5.1 ระบบไฮดรอลิก.....	15
2.5.1.2 ระบบมอเตอร์ไฟฟ้า.....	15
2.5.1.3 ระบบนิวเมติกส์.....	15
2.1.6 End Effector.....	15
2.1.6.1 Gripper.....	15
2.1.6.2 มือที่เป็นอุปกรณ์ (Tool as End Effector).....	15
2.1.7 การควบคุมการทำงานด้วยหุ่นยนต์.....	16
2.1.8 การสั่งงาน.....	16
2.1.8.1 การสั่งงานแบบแมนนวล.....	16
2.1.8.2 การสั่งงานแบบลิตทูล.....	16
2.1.8.3 การสั่งงานด้วยโปรแกรม.....	17
2.2 คีเนมาติกส์ของแขนกล(Robot Arm Kinematics).....	17
2.2.1 Direct Kinematics :The Arm Equation.....	20
2.2.2 อินเวอร์สคีเนมาติก (The Inverse Kinematics).....	24
2.3 โครงสร้างของระบบควบคุมแขนกล.....	25
2.3.1 ส่วนตรวจจับ (Sensor & Encoder).....	25
2.3.2 หน่วยประมวลผลกลางและควบคุม.....	25
2.3.3 วงจรขับมอเตอร์.....	25
2.4 มอเตอร์ไฟฟ้ากระแสตรงแบบ ดีซีเซอร์โวมอเตอร์(DC Servo Motor).....	26
2.4.1 เซอร์โวมอเตอร์(DC ServoMotor).....	26
2.4.2 โครงสร้างระบบควบคุมเซอร์โว.....	27
2.4.3 ส่วนประกอบของ DC ServoMotor.....	27
2.4.4 หลักการทำงานของ DC ServoMotor.....	28
2.5 อุปกรณ์FPGA.....	30
2.5.1 ASIC	30

สารบัญ(ต่อ)

	หน้า
2.5.2 การแบ่งหมวดของ FPGA	31
2.5.2.1 FPGA แบ่งตามโครงสร้างสถาปัตยกรรมภายในของอุปกรณ์.....	31
2.5.2.2 การแบ่งโครงสร้างสถาปัตยกรรมของ FPGA	31
2.5.2.2.1 Coarse-grained.....	31
2.5.2.1.2 Fine-grained.....	31
2.5.2.3 เทคโนโลยีการโปรแกรม (Programming Technology).....	32
2.5.3 ข้อแตกต่างอื่นๆ ของโครงสร้างสถาปัตยกรรมของ FPGA.....	32
2.6.3.1 FPGA ชนิด SRAM.....	32
2.6.3.2 FPGA ชนิด Fuse หรือ Anti-fuse	32
2.6.3.3 FPIC (Field Programmable Interconnect device).....	32
2.5.4 การออกแบบด้วย FPGA	32
2.6.4.1 ซอฟต์แวร์ Design Entry.....	32
2.6.4.2 Design Implement.....	33
2.6.4.3 Device Programming.....	33
บทที่ 3 หลักการในการออกแบบแขนกล.....	34
3.1 บทนำ.....	34
3.1.1 ส่วนของการเคลื่อนที่และการทำงานของหุ่นโดยรวม.....	34
3.1.2 ส่วนของแขนกล (Hard ware).....	34
3.1.3 ส่วนโปรแกรม (Soft ware).....	34
3.2 การออกแบบส่วนควบคุมการเคลื่อนที่ และ ตำแหน่งของแขนกล.....	35
3.2.1 ควบคุมการเคลื่อนที่และการทำงานของแขนกลโดยรวม.....	35
3.2.2 การออกแบบสมการของการเคลื่อนที่ของแขนกล.....	36
3.2.2.1 สมการของการเคลื่อนที่ Forward Kinematic.....	36
3.2.2.2 สมการของการเคลื่อนที่ Inverse Kinematic.....	36
3.3 ส่วนของซอฟต์แวร์.....	36
3.3.1 ส่วนของโปรแกรมหลัก.....	36
3.3.1.1 การทำงานของโปรแกรมหลัก.....	37
3.4 ส่วนของแขนกลและระบบการควบคุมแขนกล(Hard Ware).....	38

สารบัญ(ต่อ)

	หน้า
3.4.1 ส่วนตัวแขนกล (Robot Arm).....	38
3.4.2 ส่วนของชุดควบคุมแขนกล (Motion Controller).....	39
3.4.2.1 แหล่งจ่ายไฟให้กับ Servo Motor.....	39
3.4.2.2 Microcontroller.....	39
3.4.2.3 การ์ด FPGA.....	39
บทที่ 4 การควบคุมแบบ Real-time โดยใช้ Matlab.....	41
4.1 การควบคุมโดย Joystick.....	41
4.2 การควบคุมโดยแบบ Forward kinematics.....	42
4.3 การควบคุมโดยแบบ Inverse kinematics.....	42
บทที่ 5 การทดลองและผลการทดลอง.....	45
5.1 บทนำ.....	45
5.2 อุปกรณ์และเครื่องมือที่ใช้ในการทดลอง.....	45
5.3 การทดลอง.....	46
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	49
6.1 สรุปผลการวิจัย.....	49
6.2 ข้อเสนอแนะและแนวทางการแก้ปัญหา.....	49
บรรณานุกรม.....	50
ภาคผนวก.....	51
ประวัติผู้เขียน.....	60

สารบัญรูป

รูปที่	หน้า
2.1 แขนกลที่มีลักษณะคล้ายมนุษย์	3
2.2 Gantry Robot	5
2.3 Work envelope of Gantry Robot	5
2.4 Cylindrical Robot	6
2.5 Work envelope of Cylindrical Robot	6
2.6 Spherical Robot	7
2.7 Work envelope of Spherical Robot	7
2.8 Scara Robot	8
2.9 Work envelope of Scara Robot	8
2.10 Articulated Robot	9
2.11 Work envelope of Articulated Robot	9
2.12 ระยะที่เอื้อมถึงและสโตรคของหุ่นยนต์ทรงกระบอกลูก11	11
2.13 yaw –pitch-roll ของ Tool	11
2.14 ลำดับการหมุนแกน.....	12
2.15 ระบบ YPR	13
2.16 ระบบ RPY	13
2.17 ลักษณะการทำงานแบบสวิตช์โดยอาศัยโรบอทซีมีมอเตอร์	17
2.18 แสดงการหมุนของโครงพิกัดเทียบกับโครงพิกัดอยู่นิ่ง.....	19
2.19 แสดงโครงพิกัดเคลื่อนที่เทียบกับโครงพิกัดอยู่นิ่ง.....	19
2.20 แขนกลที่มีลักษณะเป็น links	21
2.21 แสดงโมเดลโครงพิกัดของแขนกล.....	22
2.22 แสดงพารามิเตอร์ของโครงพิกัดแขนกล.....	23
2.23 แสดงค่าตัวแปรแต่ละข้อของแขนกล.....	24
2.24 ขนาดต่างๆของ DC Servo motor.....	26
2.25 แสดงชุดเฟืองเกียร์ของ Servo Motor.....	26
2.26 แสดงส่วนประกอบภายในของ Servo Motor.....	27
2.27 โครงสร้างระบบควบคุมดิจิทัลเซอร์โว.....	27
2.28 แสดงการส่งสัญญาณ pulse เพื่อควบคุมตำแหน่ง Servo Motor.....	29

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.29 แสดงตำแหน่งการหมุน90 องศา.....	29
2.30 แสดงตำแหน่งการหมุน45 องศา.....	29
2.31 ชนิดของวงจรรวม(ASIC: Application Specific Integrated Circuit).....	30
3.1 บล็อกไดอะแกรมของระบบโดยรวม.....	34
3.2 แสดงการทำงานโดยรวมของ Robot Arm.....	35
3.3 แสดงหน้าจอโปรแกรมควบคุมหลัก.....	37
3.4 แสดงตัวแขนกล 5 แขน.....	38
3.5 แสดงการ์ด Microcontroller ที่ใช้ในงานวิจัย.....	39
3.6 แสดงการ์ด FPGA ที่ในงานวิจัยนี้.....	40
4.1 แสดงลักษณะจุดเชื่อมต่อต่างๆของแขนกล.....	41
4.2 แสดงการใช้ joystick ควบคุมแขนกล.....	41
4.3 แสดงการควบคุมของมุม θ_1	42
4.4 แสดงการควบคุมของมุม θ_2, θ_3	42
4.5 แสดงการควบคุมพิกัด X, Y, Z coordinate.....	43
4.6 แสดงการควบคุมพิกัด X, Z coordinate.....	43
5.1 แสดงการทำงานอุปกรณ์ที่ใช้ทั้งหมด.....	44
5.2 แสดง DC Servo Motor ของแต่ละแกน.....	45
5.3 Trajectory of a Robot arm	47
5.4 Working area of a Robot arm in XY plane.....	47
5.5 Working area of a Robot arm in XZ plane.....	47
5.9 แสดงผลของกราฟิกของแขนกลที่แสดงออกทางจอภาพ.....	47
5.10 แสดงการทำงานของแขนกลเมื่อทำการหยิบลูกเหล็ก.....	59

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงการทำงานของจุดต่อหุ่นยนต์	4
2.2 แสดงแกนการทำงานของหุ่นยนต์แต่ละชนิด.....	5
2.3 คุณลักษณะของหุ่นยนต์	10
2.4 การเคลื่อนที่แบบ Yaw ,Pitch และ row	13
5.1 แสดงการควบคุมเปรียบเทียบผลลัพธ์การใช้ forward and inverse kinematics.....	46

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้กระบวนการผลิตต่างๆในโรงงานอุตสาหกรรมส่วนใหญ่ นั้นจะเป็นการผลิตเชิงปริมาณ (Mass Production) คือ ผลิตชิ้นงานจำนวนมากๆ และซ้ำๆ กัน การใช้มนุษย์ในการผลิตอาจจะได้ชิ้นงานไม่ตรงตามมาตรฐานและผลิตได้ช้าไม่ทันต่อความต้องการ ด้วยเหตุผลนี้เองหุ่นยนต์จึงได้ถูกนำมาใช้งานเป็นจำนวนมากเพื่อทำงานทดแทนมนุษย์ ทั้งยังมีประสิทธิภาพในการทำงานที่ดีกว่ามนุษย์และสามารถทำงานในสภาวะแวดล้อมที่มนุษย์ไม่สามารถทำได้ และยังเป็น การลดต้นทุนในการผลิตอีกด้วย แต่อย่างไรก็ตาม การทำงานของหุ่นจำเป็นต้องมีการควบคุมจากมนุษย์ เพื่อที่จะทำให้มันทำงานได้ตรงตามวัตถุประสงค์ที่ต้องการ ด้วยเหตุผลนี้แบบจำลองการเคลื่อนที่ของแขนหุ่นยนต์สำคัญมากที่จะวิเคราะห์และเจาะจงพื้นที่การทำงานเพื่อรับเส้นทางความปลอดภัยก่อนการพัฒนาจริงของแขนหุ่นยนต์ ถึงแม้ว่าโปรแกรมแขนหุ่นยนต์จำนวนมากๆที่มีอยู่ส่วนมากสำหรับโปรแกรมในอุตสาหกรรมเป็นสิ่งที่ยากที่จะแก้ไข ดังเช่น การพัฒนาของวิศวกรรมหุ่นยนต์โปรแกรมการใช้ MATLAB คือการแก้ปัญหาที่ดีเนื่องจากเป็นโปรแกรมที่ง่ายและเร็วที่จะแก้ไขพารามิเตอร์ต่างๆและทดลองกับความหลากหลายหนึ่งความหลากหลายของผู้ควบคุม คุณประโยชน์ของที่แบบตัวอย่างแขนหุ่นยนต์จะศึกษาการออกแบบและควบคุมการเคลื่อนไหวของแขนหุ่นยนต์ตั้งแต่นั้น ยังสามารถช่วยเพื่อพัฒนาซอฟต์แวร์สำหรับการควบคุมงานของอุตสาหกรรมเช่นเดียวกันกับขบวนการอัตโนมัติอื่นๆ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อศึกษาทฤษฎีการควบคุมการเคลื่อนที่ของแขนกลมาประยุกต์ใช้ในการออกแบบพฤติกรรมและลักษณะของการเคลื่อนที่ของแขนกลโดยใช้โปรแกรมการจำลองแบบเวลาจริงมาช่วยจำลองวิถีทางเดินที่ปลอดภัยทั้งพื้นที่การทำงานและการเคลื่อนไหวในตำแหน่งต่างๆของแขนกล

1.3 สมมุติฐานของการศึกษา

แบบจำลองการเคลื่อนที่ของแขนหุ่นยนต์สำคัญมากที่จะวิเคราะห์และเจาะจงพื้นที่การทำงานเพื่อรับเส้นทางเคลื่อนที่ของแขนหุ่นยนต์ซึ่งบางจุดอาจจะมีคำตอบของสมการอยู่หลายคำตอบที่จะทำให้เกิดผิดพลาดในการเคลื่อนที่ของแขนหุ่นยนต์

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

โปรแกรมออกแบบสมการทางกลศาสตร์ (kinematics) จากแขนกลต้นแบบ(Hardware) และนำสมการที่ได้ไปใช้ในการเขียนโปรแกรมการควบคุมการเคลื่อนที่ของแขนกลในตำแหน่งต่างๆ จากจุดเริ่มต้นไปยังจุดใดๆ และทำการตรวจสอบค่าตำแหน่งบนพิกัดผ่านทางตัวควบคุม (Controller) เพื่อใช้ในการตรวจสอบความผิดพลาดที่เกิดจากการเคลื่อนที่จริงของ Robot Arm โดยเปรียบเทียบกับค่านับกับควบคุมสำหรับนำไปสั่งงานได้จริงต่อไปเพื่อทราบถึงพฤติกรรมการทำงานและการเคลื่อนที่ของแขนกล

1.5 ขอบเขตการวิจัย

เป็นการออกแบบสมการ (Kinematics) และออกแบบโปรแกรมควบคุมการทำงานโดยผ่านทางคอมพิวเตอร์แบบจำลองการทำงาน (Simulation) ตลอดจนควบคุมจำลองการเคลื่อนที่ของแขนกลแบบ 4 แกนเพื่อกำหนดขอบเขตของการเคลื่อนที่ได้ทำการออกแบบไว้แล้วว่าสามารถเคลื่อนที่และสามารถทำงานให้ไปยังตำแหน่งปลายทางที่ต้องการให้เคลื่อนที่ไปได้หรือไม่ และทดสอบถึงความเป็นไปได้ของการทำมุมของแต่ละส่วนเชื่อมต่อกลไก(Link) เพื่อให้ทำงานในขอบเขตที่กำหนดได้

1.6 ขั้นตอนของการศึกษา

การทำงานวิจัยฉบับนี้มีขั้นตอนการศึกษาเริ่มจากการศึกษาค้นคว้าเทคนิคสมการการเคลื่อนที่ของสมการการเคลื่อนที่ของแขนกล ซึ่งจะนำทฤษฎีมาเป็นแนวทางในการออกแบบและสร้างแบบจำลองคณิตศาสตร์ด้วยโปรแกรม MATLAB ทำการจำลองการเคลื่อนที่ของแขนกล เพื่อทดลองขอบเขตของการเคลื่อนที่ได้ทำการออกแบบว่าสามารถเคลื่อนที่และสามารถทำงานไปยังตำแหน่งปลายทางที่ต้องการให้เคลื่อนที่ไปได้หรือไม่แล้ว

บทที่ 2

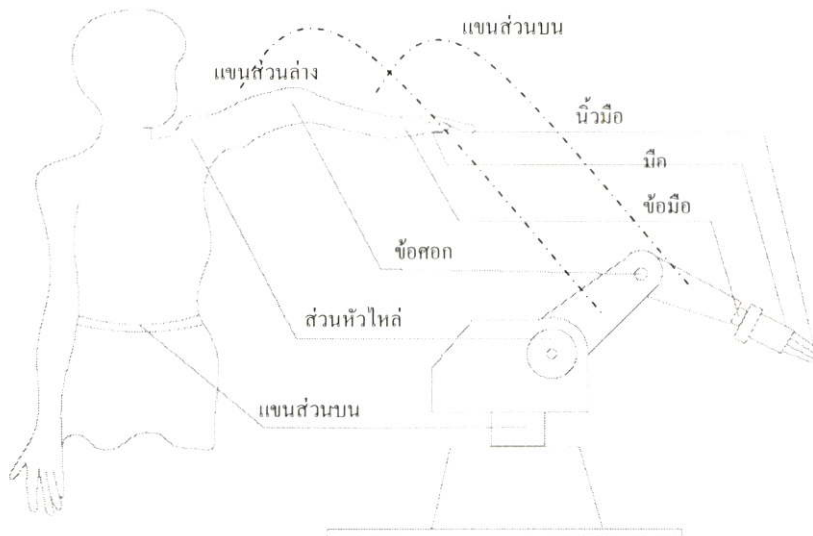
ทฤษฎีพื้นฐานของการวิเคราะห์และควบคุมหุ่นยนต์ (Fundamental of Robotic Analysis and Control)

2.1 บทนำ

หุ่นยนต์ คือ เครื่องจักรที่ถูกควบคุมอัตโนมัติ สามารถเขียน โปรแกรมใหม่ได้ ใช้งาน เอนกประสงค์ โปรแกรมการเคลื่อนที่ที่จะต้องสามารถโปรแกรมให้เคลื่อนที่ได้อย่างน้อย 3 แกนหรือ มากกว่าหุ่นยนต์อาจจะยึดอยู่กับที่หรือย้ายตำแหน่ง (Mobile) เพื่อใช้ในการอุตสาหกรรม

หุ่นยนต์ (Robotics) มีมากมายหลายชนิดขึ้นอยู่กับจุดประสงค์ของการใช้งาน เช่น หุ่นยนต์ที่เป็นของเล่น หุ่นยนต์ใช้ในการแพทย์ หุ่นยนต์เพื่องานวิจัย หุ่นยนต์อุตสาหกรรม และหุ่นยนต์สำรวจ ฯลฯ

การทำงานของหุ่นยนต์อุตสาหกรรมจะเลียนแบบร่างกายของมนุษย์โดยจะเลียนแบบเฉพาะ ส่วนของร่างกายที่จะนำไปใช้ประโยชน์ ในอุตสาหกรรมเท่านั้น นั่นคือช่วงแขนของมนุษย์ ดังนั้น บางคนอาจจะได้ยินคำว่าแขนกล ซึ่งก็หมายถึงหุ่นยนต์อุตสาหกรรมนั่นเอง โดยการทำงานของ หุ่นยนต์อุตสาหกรรมเปรียบเทียบกับแขนมนุษย์ แสดงดังรูปที่ 2.1



รูปที่ 2.1 แขนกลที่มีลักษณะคล้ายมนุษย์

ปัจจุบันและในอนาคตหุ่นยนต์อุตสาหกรรมจะเข้ามามีบทบาทในอุตสาหกรรมมากขึ้น โดยจะทำงานแทนมนุษย์ในงานต่าง ๆ เหล่านี้งานที่อันตราย เช่น งานยกเหล็กเข้าเตาหลอม งานที่เกี่ยวข้องกับสารเคมีงานฆ่าซากน่าเบื่อ เช่น งานยกสินค้าจากสายงานการผลิต งานประกอบ งานบรรจุผลิตภัณฑ์งานที่ต้องการคุณภาพมาตรฐานเดียวกัน เช่น งานเชื่อม งานตัด งานที่ต้องใช้ทักษะความชำนาญสูง เช่น งานเชื่อมแนว เชื่อมเลเซอร์งานที่ต้องใช้ความละเอียดประณีต เช่น งานประกอบชิ้นส่วนอิเล็กทรอนิกส์ งานตรวจสอบ (Inspection) ฯลฯ

2.1.1 การแบ่งประเภทของแขนกล



เราสามารถแบ่งประเภทแขนกล โดยพิจารณาจากพื้นฐานใหญ่ ๆ ดังนี้

1. โครงสร้างภายนอกและการเคลื่อนที่ของแขนกล
2. ทางของแขนกล ซึ่งขึ้นอยู่กับควบคุมการเคลื่อนที่
3. ประสิทธิภาพของแขนกล

การพิจารณาแบ่งประเภทของแขนกลในขั้นนี้ เราจะอาศัยเฉพาะพื้นฐานข้อ 1. และ ข้อ2. เท่านั้น แม้ว่าพื้นฐานของข้อ3. จะมีความสำคัญมากที่สุด แต่จะต้องอาศัยความรู้ทางด้านจลนศาสตร์(Kinematics) และพลศาสตร์ (Dynamics) ขั้นสูง ซึ่งยากต่อการทำความเข้าใจ การแบ่งประเภทแขนกลในระดับนี้จึงจะไม่นำพื้นฐานข้อ 3. มาคิด

โดยทั่วไปการแบ่งชนิดของหุ่นยนต์จะแบ่งตามลักษณะรูปทรงของพื้นที่ทำงาน (Envelope Geometric) แต่ก่อนจะอธิบายชนิดของหุ่นยนต์ขออธิบายการทำงานของจุดต่อ (Joint) ของหุ่นยนต์อุตสาหกรรมซึ่งในขั้นพื้นฐานมี 2 ชนิดด้วยกัน

ตารางที่ 2.1 แสดงการทำงานของจุดต่อหุ่นยนต์

ชนิด	สัญลักษณ์	หมายเหตุ
Revolute (R)		เป็นการหมุนรอบแกน (Rotary)
Prismatic (P)		การเคลื่อนที่เชิงเส้น (Linear motion)

จุดต่อ (Joint) ทั้งสองแบบเมื่อนำมาต่อเข้าด้วยกันอย่างน้อย 3 แกนหลักจะได้พื้นที่ทำงาน (Work envelope) ที่มีลักษณะแตกต่างกันไป ซึ่งสามารถนำมาแบ่งชนิดของหุ่นยนต์ได้ดังต่อไปนี้

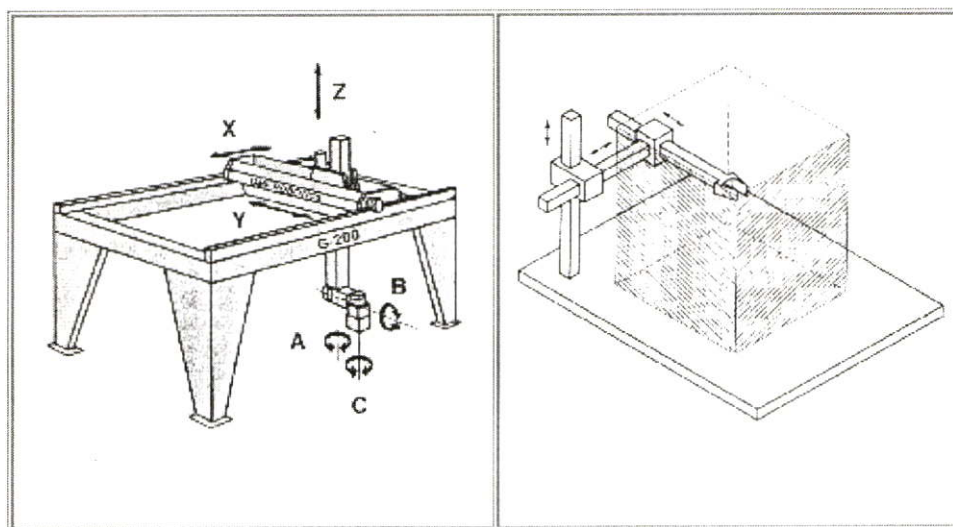
ตารางที่ 2. 2 แสดงแผนการทำงานของหุ่นยนต์แต่ละชนิด

ชนิดของหุ่นยนต์	แกนที่ 1 (เอว)	แกนที่ 1 (เอว)	แกนที่ 3 (ข้อศอก)
Cartesian (gantry)	P	P	P
Cylindrical	R	P	P
Spherical (Polar)	R	R	P
SCARA	R	P	R
Articulated	R	R	R

R = Revolute, P = Prismatic

2.1.1.1 Cartesian (gantry) Robot

หุ่นยนต์แบบ Cartesian แกนทั้ง 3 ของหุ่นยนต์จะเคลื่อนที่เป็นแบบเชิงเส้น (Prismatic) ถ้าโครงสร้างมีลักษณะคล้าย Overhead Crane จะเรียกว่าเป็นหุ่นยนต์ชนิด gantry แต่ถ้าหุ่นยนต์ไม่มีขาตั้งหรือขาเป็นแบบอื่น เรียกว่า ชนิด Cartesian



รูปที่ 2.2 Gantry Robot

รูปที่ 2.3 Work envelope of Gantry Robot

ข้อดี

- เคลื่อนที่เป็นเส้นตรงทั้ง 3 มิติ
- การเคลื่อนที่สามารถทำความเข้าใจง่าย
- มีส่วนประกอบง่าย ๆ
- โครงสร้างแข็งแรงตลอดการเคลื่อนที่

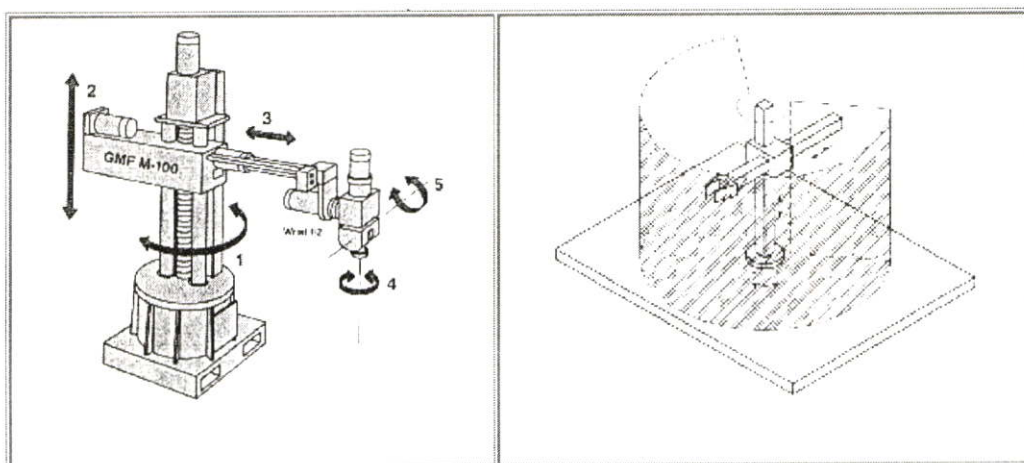
ข้อเสีย

- ต้องการพื้นที่ติดตั้งมาก
- บริเวณที่หุ่นยนต์เข้าไปทำงานได้จะเล็กกว่าขนาดของหุ่นยนต์
- ไม่สามารถเข้าถึงวัตถุจากทิศทางข้างใต้ได้
- แกนแบบเชิงเส้นจะ Seal เพื่อป้องกันฝุ่นและของเหลวได้ยาก

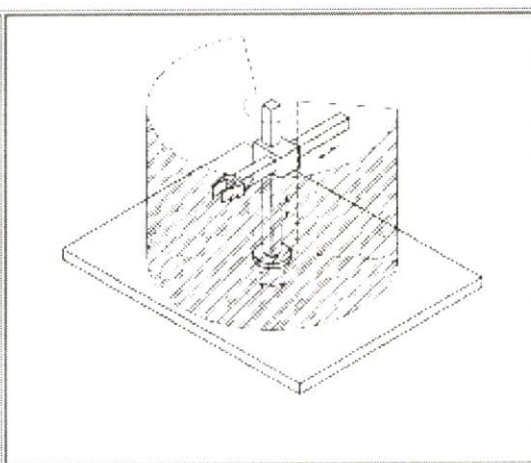
การประยุกต์ใช้งานเนื่องจากโครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ดังนั้นจึงเหมาะสมกับการเคลื่อนย้ายของหนัก ๆ หรือเรียกว่างาน Pick-and-Place เช่น ใช้โหลดชิ้นงานเข้าเครื่องจักร (Machine loading) ใช้จัดเก็บชิ้นงาน (Stacking) นอกจากนี้ยังสามารถใช้ในงานประกอบ (Assembly) ที่ไม่ต้องการเข้าถึงในลักษณะที่มีมุมมน เช่น ประกอบอุปกรณ์อิเล็กทรอนิกส์ และงาน Test ต่าง ๆ

2.1.1.2 Cylindrical Robot

หุ่นยนต์แบบ Cylindrical นี้จะมีแกนที่ 2(ไหล่) และแกนที่ 3(ข้อศอก) เป็นแบบ prismatic ส่วนแกนที่ 1(เอว) จะเป็นแบบหมุน (revolute) ซึ่งทำให้การเคลื่อนที่ได้พื้นที่การทำงานเป็นรูปทรงกระบอก ดังรูปที่ 2.4 และ 2.5



รูปที่ 2.4 Cylindrical Robot



รูปที่ 2.5 Work envelope of Cylindrical Robot

ข้อดี

- มีส่วนประกอบไม่ซับซ้อน
- การเคลื่อนที่สามารถเข้าใจได้ง่าย สามารถเข้าถึงเครื่องจักรที่มีการเปิด-ปิด หรือเข้าไปในบริเวณที่เป็นช่องหรือโพรงได้ง่าย (Loading) เช่น การ โหลดชิ้นงานเข้าเครื่อง CNC

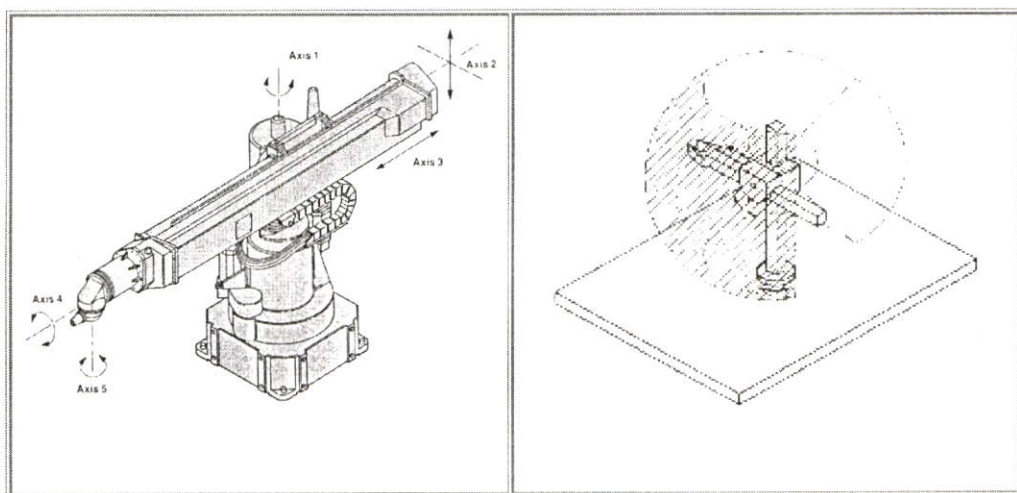
ข้อเสีย

- มีพื้นที่ทำงานจำกัด
- แกนที่เป็นเชิงเส้นมีความยุ่งยากในการ Seal เพื่อป้องกันฝุ่นและของเหลว

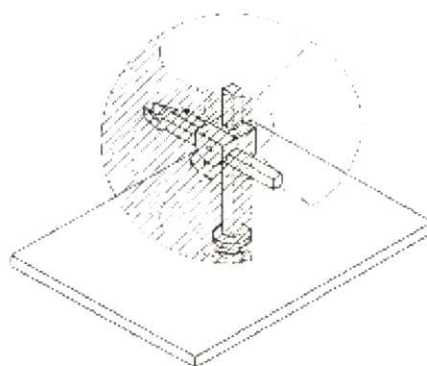
การประยุกต์ใช้งาน โดยทั่วไปจะใช้ในการหยิบยกชิ้นงาน(Pick-and-Place) หรือป้อนชิ้นงานเข้าเครื่องจักรเพราะสามารถเคลื่อนที่เข้าออกบริเวณที่เป็นช่องโพรงเล็ก ๆ ได้สะดวก

2.1.1.3 Spherical Robot (Polar)

หุ่นยนต์แบบ Spherical มีสองแกนที่เคลื่อนที่ในลักษณะการหมุน (Revolute Joint) คือ แกนที่ 1 (เอว) และแกนที่ 2 (ไหล่) ส่วนแกนที่ 3 (ข้อศอก) จะเป็นลักษณะของการเคลื่อนที่แนวเส้นตรง ดังรูปที่ 2.6 ซึ่งทำให้ได้พื้นที่การทำงาน ดังรูปที่ 2.7



รูปที่ 2.6 Spherical Robot



รูปที่ 2.7 Work envelope of Spherical Robot

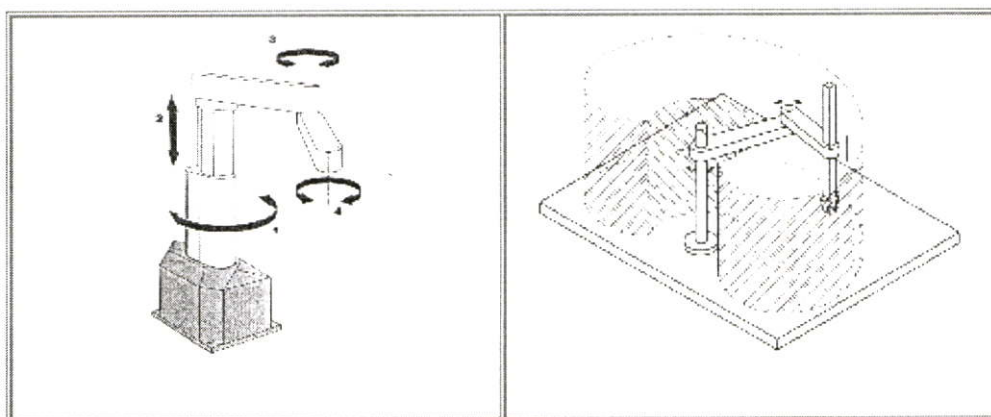
ข้อดี

- มีระบบพิกัด (Coordinate) และส่วนประกอบที่ซับซ้อน
- การเคลื่อนที่และระบบควบคุมมีความซับซ้อนขึ้น

การประยุกต์ใช้งานในงานที่มีการเคลื่อนที่ในแนวตั้ง (Vertical) เพียงเล็กน้อย เช่น การโหลดชิ้นงานเข้าออกจากเครื่องปั๊ม (Press) หรืออาจจะใช้งานเชื่อมจุด (Spot Welding)

2.1.1.4 SCARA Robot

หุ่นยนต์แบบ SCARA (Selective Compliance Assembly Robot Arm) จะมีลักษณะแกนที่ 1 (เอว) และแกนที่ 3 (ข้อศอก) หมุนรอบแกนแนวตั้ง และแกนที่ 2 จะเป็นลักษณะการเคลื่อนที่ขึ้นลง (Prismatic) ดังรูปที่ 2.8 ทำให้ได้พื้นที่การทำงานดัง รูปที่ 2.9 หุ่นยนต์ SCARA จะเคลื่อนที่ได้รวดเร็วในแนวระนาบ และมีความแม่นยำสูงเป็นหุ่นยนต์ที่มีแขน ที่ใช้ในการเคลื่อนที่ประกอบด้วยกัน 2 แขน และมือที่ใช้ทำงานอีกหนึ่งชิ้น ซึ่งแขนทั้งสองจะถูกยึดติดอยู่บนฐาน โดยแขนหลักของหุ่นยนต์จะติดอยู่ที่จุดหมุนของฐานหุ่นยนต์ และแขนที่สองจะถูกติดตั้งที่ปลายของแขนหลัก ส่วนมือในการทำงานจะติดอยู่ที่ปลายของแขนที่สอง ในส่วนของการทำงานของแขนหุ่นยนต์ทั้งสอง จะถูกจำกัดการเคลื่อนที่เฉพาะในแนวราบหรือแกน x,y ส่วนมือก็จะทำงานเฉพาะในแนวตั้งหรือแกน z เท่านั้น



รูปที่ 2.8 Scara Robot

รูปที่ 2.9 Work envelope of Scara Robot

ข้อดี

- สามารถเคลื่อนที่ในแนวระนาบ และขึ้นลงได้รวดเร็ว
- มีความแม่นยำสูง

ข้อเสีย

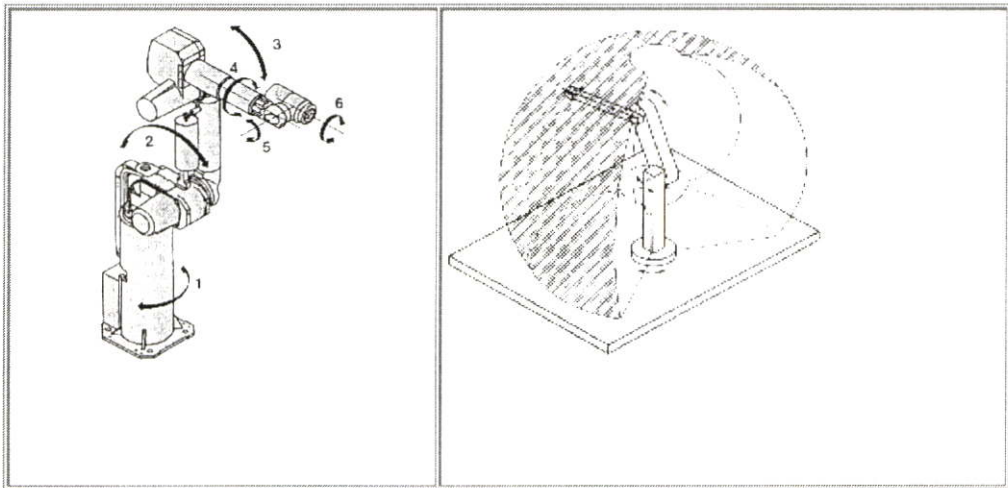
- มีพื้นที่การทำงานจำกัด
- ไม่สามารถหมุน (Rotation) ในลักษณะมุมต่าง ๆ ได้
- สามารถยกน้ำหนัก (Pay load) ได้ไม่มากนัก

การประยุกต์ใช้งาน เนื่องจากการเคลื่อนที่ในแนวระนาบและและขึ้นลงได้รวดเร็วจึงเหมาะกับงานประกอบชิ้นส่วนทางอิเล็กทรอนิกส์ที่ต้องการความรวดเร็วและการเคลื่อนที่ที่ไม่ต้องการ การ

หมุนมากนัก แต่จะไม่เหมาะสมกับการประกอบชิ้นงานทางกล (Mechanical Part) ซึ่งส่วนใหญ่การประกอบจะอาศัยการหมุน (Rotation) ในลักษณะมุมต่าง ๆ นอกจากนี้ SCARA ROBOT ยังไม่เหมาะกับงานตรวจสอบ (Inspection) งานบรรจุภัณฑ์ (Packaging)

2.1.1.5 Articulated Robot (Revolute)

หุ่นยนต์แบบ Articulated เป็นหุ่นยนต์ที่มีทุกแกนการเคลื่อนที่จะเป็นแบบหมุน (Revolute) รูปแบบการเคลื่อนที่จะคล้ายกับแขนคน ซึ่งจะประกอบด้วยช่วงเอว ท่อนแขนบนท่อนแขนล่าง ข้อมือ การเคลื่อนที่ทำให้ได้พื้นที่การทำงาน ดังรูปที่ 2.11



รูปที่ 2.10 Articulated Robot

รูปที่ 2.11 Work envelope of Articulated Robot

ข้อดี

- เนื่องจากทุกแกนจะเคลื่อนที่ในลักษณะ ของการหมุนทำให้ความยืดหยุ่นสูงในการเข้าไปในจุดต่าง ๆ
- บริเวณข้อต่อ (Joint) สามารถ Seal เพื่อป้องกันฝุ่น ความชื้นหรือน้ำได้
- มีพื้นที่การทำงานมาก
- สามารถเข้าถึงชิ้นงานทั้งด้านบนด้านล่าง
- เหมาะกับการใช้งานมอเตอร์ไฟฟ้าเป็นชุดขับเคลื่อน

ข้อเสีย

- มีระบบพิกัด(Coordinate) ที่ซับซ้อน
- การเคลื่อนที่และระบบควบคุมทำความเข้าใจ ได้ยากขึ้น
- ควบคุมให้เคลื่อนที่ในแนวเส้นตรง (Linear) ได้ยาก

- โครงสร้างไม่มั่นคงตลอดช่วงการเคลื่อนที่ เพราะบริเวณ Work Envelope ปลายแขนจะมีการสั่นทำให้ความแม่นยำลดลง

2.1.2 ลักษณะเฉพาะของหุ่นยนต์ (Robot Specification)

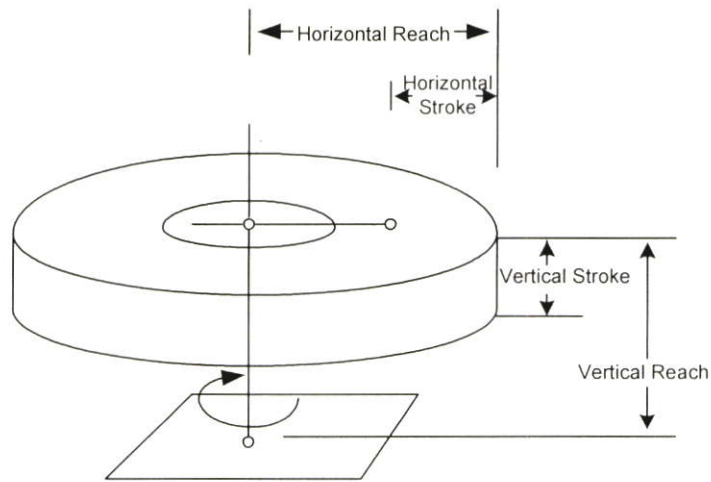
นอกจากเกณฑ์ต่าง ๆ ในการแบ่งประเภทของหุ่นยนต์ที่กล่าวมาแล้วแล้ว ยังมีลักษณะบางอย่างที่ช่วยให้ผู้ใช้ในการเลือกหุ่นยนต์ที่ต้องการ ลักษณะดังต่อไปนี้ดังตารางที่ 2.3

ตารางที่ 2.3 คุณลักษณะของหุ่นยนต์

คุณลักษณะ (Characteristics)	หน่วย (Unit)
จำนวนแกน	-
ความสามารถในการยกน้ำหนัก	Kg
ความเร็วสูงสุด, เวลาครบรอบ	sec/mm
ระยะเอื้อมถึงและสโตรค	mm
การหมุนของเครื่องมือ	Deg
การซ้ำตำแหน่งเดิม	Mm
ความละเอียดและความแม่นยำ	Mm
สภาพแวดล้อมการทำงาน	-

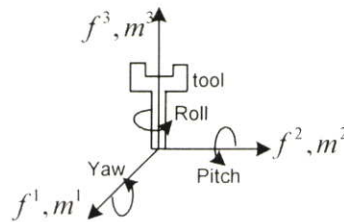
ระยะที่เอื้อมถึงและสโตรค (Reach and Stroke) เป็นการวัดขนาดของพื้นที่การทำงาน (Work Envelope) อย่างคร่าว ๆ ระยะที่เอื้อมถึงในแนวนอน (Horizontal Reach) คือระยะทางตามแนวรัศมีที่มากที่สุดที่มือสามารถเอื้อมไปถึงโดยวัดจากแกนแนวดิ่งที่ทำให้หุ่นยนต์หมุน สโตรคแนวนอน (Horizontal Stroke) คือระยะทางตามแนวรัศมีทั้งหมดที่ข้อมือสามารถไปได้ ดังนั้นระยะทางที่เอื้อมถึงในแนวนอนลบด้วย สโตรคในแนวนอน คอระยะตามแนวรัศมีที่น้อยที่สุดที่ข้อมือสามารถไปถึงได้โดยวัดจากแกนฐานซึ่งมีค่าเป็นบวกเสมอ จึงได้ว่า $Stroke \geq Reach$

ตัวอย่างเช่น ระยะที่เอื้อมถึงในแกนนอน (Horizontal Reach) ของหุ่นยนต์พิกัดทรงกระบอกคือ มีรัศมีเป็นทรงกระบอกอันนอกจากพื้นที่การทำงาน ในขณะที่สโตรคแนวนอน (Horizontal Stroke) คือผลระหว่างรัศมีแนวในและแนวนอก ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 ระยะที่เอื้อมถึงและสโตรคของหุ่นยนต์ทรงกระบอก

การหมุนของ Tool (Tool Orientation) ในขณะที 3 แกนหลักใช้ในการพิจารณารูปทรงของพื้นที่การทำงานแกนที่เหลื่อจึงใช้เป็นการพิจารณาพื้นที่การทำงาน แกนที่เหลื่อจึงใช้ในการพิจารณาการหมุนของ Tool จะใช้ในระบบ yaw-pitch-roll (YPR) ดังรูปที่ 2.13



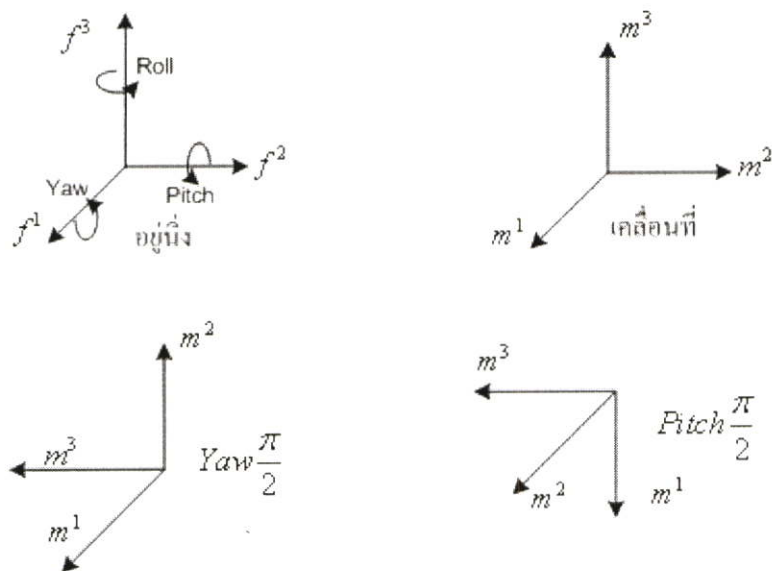
รูปที่ 2.13 yaw-pitch-roll ของ Tool

ในการระบุการหมุน จะติดโครงพิกัด tool $M = \{m^1, m^2, m^3\}$ ไว้ที่ tool และโครงนี้จะเคลื่อนที่ไปพร้อมกับ tool ดังแสดงในรูปที่ 2.13 แกน m^3 ขนานอยู่แนวเดียวกับแกนของ tool และชี้ออกจากข้อมือ แกน m^3 ขนานกับทิศทางที่ปลายนิ้วของ tool เลื่อนเปิด-ปิด ส่วนแกน m^1 เป็นไปตามกฎมือขวาของโครงพิกัด M

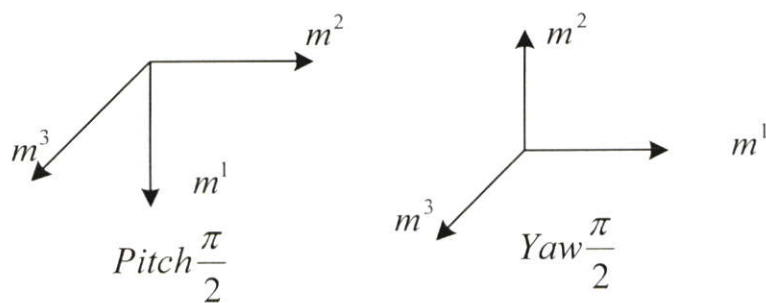
การเคลื่อนที่แบบ yaw-pitch-roll จะกระทำตามลำดับกับแกนที่อยู่หนึ่ง เมื่อเริ่มต้นโครงพิกัดเคลื่อนที่ ของ tool M จะทับอยู่กับโครงพิกัดอยู่นิ่งของข้อมือ $F = \{f^1, f^2, f^3\}$ ซึ่งติดกันอยู่ที่ส่วนปลายของปลายแขน (Forearm) yaw คือการหมุน tool รอบแกนข้อมือ f^1 , pitch คือการหมุน tool

รอบแกนข้อมือ f^2 , roll คือการหมุน tool รอบแกนข้อมือ f^3 ในแต่ละกรณีมุมบวกเกิดจากการหมุนทวนเข็มนาฬิกาโดยมองจากปลายแกนเข้าสู่จุดกำเนิด

ลำดับการเคลื่อนที่ yaw, pitch, roll มีความสำคัญมากเพราะมีผลต่อลักษณะการวางตัวของสุดท้ายของ tool ตัวอย่างเช่น หมุน yaw ไป 90° ตามด้วย pitch 90° จะมีลักษณะการวางตัวของสุดท้ายต่างกับการหมุน pitch 90° ไป แล้วตามด้วย yaw 90° เป็นต้น



หมุนYaw ตามด้วย Pitch



หมุน Pitch ตามด้วยYaw

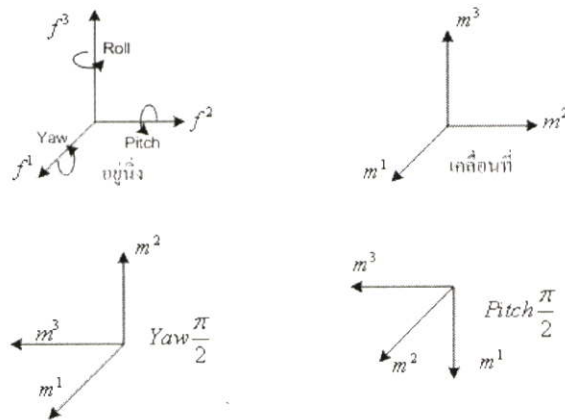
รูปที่ 2.14 ลำดับการหมุนแกน

การกำหนด YPR ตามลำดับการหมุนแสดงดังตารางที่ 2.4

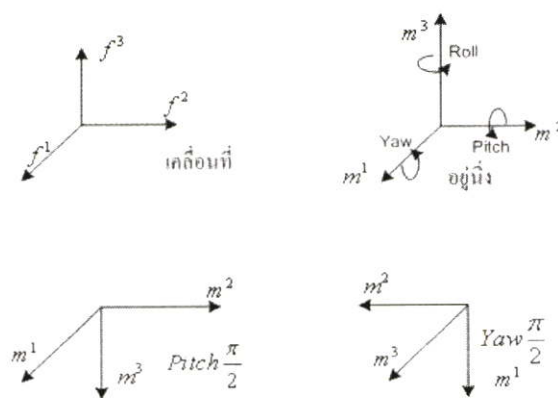
ตารางที่ 2.4 การเคลื่อนที่แบบ yaw,pitch และ row

ขั้นตอน	การอธิบาย	แกน
1	Yaw	f^1
2	Pitch	f^2
3	Roll	f^3

อีกหนึ่งวิธีที่ใช้ระบุการหมุนคือ ระบุลำดับการหมุนลำดับกับ YPR และคิดเทียบกับโครงพิคัดเคลื่อนที่ของ tool M แทนที่โครงพิคัดที่อยู่หนึ่งของข้อมือ F ดังรูปที่ 2.10 โดยเริ่มแรก roll คือการหมุนรอบแกน m^3 , pitch คือการหมุนรอบแกน m^2 , yaw คือการหมุนรอบแกน m^1 การคิดลักษณะนี้เรียกว่า เป็นระบบ RPY จะให้ผลลัพธ์เหมือนแบบ YPR



รูปที่ 2.15 ระบบ YPR หมุน pitch ตาม Yaw



รูปที่ 2.16 ระบบ RPY

2.1.3 ความแม่นยำของการเคลื่อนที่

ความสำคัญอีกประการหนึ่งของการสร้างหุ่นยนต์คือ ความแม่นยำในการทำงาน ซึ่งความแม่นยำขึ้นอยู่กับตัวอย่าง 3 อย่าง

2.1.3.1 Spatial Resolution

คือช่วงการเคลื่อนที่มีระยะทางสั้นที่สุดที่หุ่นยนต์แต่ละตัวสามารถที่จะทำได้ซึ่ง Spatial Resolution นี้ขึ้นอยู่กับองค์ประกอบสำคัญ 2 ประการ คือ

1) ระบบการควบคุม (Control System) ระบบการควบคุมนี้จะรวมถึงการวัดสัญญาณป้อนกลับของหุ่นยนต์ด้วย

2) ความคลาดเคลื่อนเชิงกล (Mechanical inaccuracy) ของหุ่นยนต์แต่ละตัวขึ้นอยู่กับลักษณะของข้อหมุน (joint) และข้อต่อ (link) และระบบต้นกำลังของหุ่นยนต์ตัวนั้นด้วย

2.1.3.2 Accuracy

คือตัวที่แสดงถึงความสามารถของหุ่นยนต์ในการเคลื่อนที่เข้าใกล้จุดเป้าหมายตามที่เราสั่ง accuracy สามารถที่จะกำหนดให้อยู่ในเทอมของ Spatial Resolution ได้ทั้งนี้เพราะว่าการเคลื่อนที่ให้เข้าใกล้จุดเป้าหมาย ก็ต้องขึ้นอยู่กับช่วงของการเคลื่อนที่ที่มีความละเอียดมากน้อยเพียงใด ในการทำงานเราต้องวางจุดที่เราต้องการให้หุ่นยนต์ทำงานอยู่ระหว่างกลางของตำแหน่งของการเคลื่อนที่ของหุ่นยนต์ ทั้งนี้เพราะว่าความคลาดเคลื่อนเชิงกลมีผลต่อความแม่นยำของหุ่นยนต์ ความแม่นยำของหุ่นยนต์กำหนดให้เท่ากับครึ่งหนึ่งของระยะทางการเคลื่อนที่ที่สั้นที่สุดของหุ่นยนต์ที่สามารถทำได้ โดยขึ้นอยู่กับองค์ประกอบต่อไปนี้

พื้นที่การทำงานของหุ่นยนต์ ถ้าแขนทำงานในพื้นที่การทำงานจะมีความแม่นยำมากกว่าแขนออกนอกพื้นที่การทำงาน

- 1) วงรอบการทำงาน ถ้าวงรอบการทำงานเป็นวงรอบที่แน่นอนทำให้ความแม่นยำมีมากขึ้น
- 2) น้ำหนักที่ได้รับ ถ้าหุ่นยนต์ทำงานได้รับน้ำหนักมากๆ นั้นจะมีผลให้ความแม่นยำลดลง

2.1.3.3 Repeatability

คือความสามารถของหุ่นยนต์ในการกลับมาทำงานซ้ำที่เดิมหลายๆ ครั้งได้เนื่องจากสาเหตุต่างๆ เช่น Backlash ในเฟืองและความยืดหยุ่น (Flexibility) ของส่วนต่างๆ ซึ่งมักทำให้เกิดความผิดพลาดใน Repeatability

2.1.4 การเคลื่อนที่ของหุ่นยนต์

ลักษณะการเคลื่อนที่ของหุ่นยนต์อุตสาหกรรมนี้ จะคล้ายกับการเคลื่อนที่ของอุปกรณ์ในเครื่องจักรเอ็นซี กล่าวคือมีได้ 2 ลักษณะคือ การเคลื่อนที่จากจุดต่อจุด (point to point) แบบต่อเนื่อง (Continuous path) โดยแบบแรกจะเป็นการกำหนดตำแหน่ง (เริ่มต้นถึงสิ้นสุด) ให้กับหุ่นยนต์เพื่อให้มันเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยไม่สนใจว่าจะใช้เส้นทางไหน ซึ่งหุ่นยนต์ที่มี

การเคลื่อนที่ในลักษณะนี้ สามารถนำไปใช้ในงานประเภทหยิบ จับ สิ่งของ หรืองานเชื่อมตามจุด (Spot Welding) ฯลฯ โดยแบบหลังจำเป็นต้องใช้หน่วยความจำในการสั่งงานมากกว่าแบบแรก

2.1.5 คุณสมบัติอื่น ๆ ของหุ่นยนต์อุตสาหกรรมทั่วไป

ลักษณะของวงจรถับ (Drive system)

2.1.5.1 ระบบไฮดรอลิก

ซึ่งเป็นระบบที่ง่ายต่อการดูแล ทนทาน และเคลื่อนที่ทำงานได้เร็ว

2.1.5.2 ระบบมอเตอร์ไฟฟ้า

เป็นระบบที่ใช้สเตปมอเตอร์ (Stepping motor) หรือ เซอร์โวมอเตอร์ (Servomotor) เป็นตัวไปควบคุมการเคลื่อนที่ส่วนต่าง ๆ หุ่นยนต์ที่ใช้ในระบบนี้จะมีกำลังไม่สูงเหมือนแบบแรก แต่จะมีความละเอียด ไม่ว่าในแง่ของตำแหน่งในการเคลื่อนที่ หรือการทำซ้ำจะดีกว่า

2.1.5.3 ระบบนิวเมติกส์

ระบบนี้ใช้กับหุ่นยนต์ขนาดเล็ก อีกทั้งระบบไม่ยุ่งยาก

2.1.6 End Effector

End Effector หรือ “มือ” ของหุ่นยนต์นี้เป็นอุปกรณ์ที่หุ่นยนต์ใช้ทำงานจริง (ส่วนแขนนั้นใช้เพื่อเลื่อนตำแหน่ง “มือ” นี้ให้ไปอยู่ในจุดที่ต้องการเท่านั้น) งานที่ส่วนนี้ทำก็ อาทิเช่น การจับชิ้นส่วน การบัดกรีตามจุด การพ่นสี ฯลฯ ดังนั้นลักษณะของส่วนนี้จึงต่างกันออกไปสุดแต่แต่จินตนาการของผู้ออกแบบซึ่งต้องออกแบบให้เหมาะสมกับงาน แต่ถ้าจะมีการแบ่งกันเป็นชนิดแล้ว “End Effector” นี้จะออกได้เป็น 2 กลุ่มใหญ่ๆ คือ แบบมือจับ (Gripper) แบบมือที่เป็นอุปกรณ์ (Tool as End Effector)

2.1.6.1 Gripper

Gripper นี้เป็นอุปกรณ์ที่ใช้จับชิ้นส่วนหรือเครื่องมือโดยออกแบบมีได้หลายลักษณะ ทั้งนี้ขึ้นอยู่กับการใช้งาน เช่น

1) แบบมือคีบ (อาศัยแรงเสียดทานในการจับ) ใช้งานได้กับวัสดุทุกประเภท แต่หากเป็นวัสดุบอบบางอาจต้องมีอุปกรณ์อื่นช่วยเสริม

2) แบบ Suction Cup อาศัยคุณสมบัติการทำให้เกิดสภาพสุญญากาศ เป็นตัวดึงชิ้นงานใช้ได้กับวัสดุผิวเรียบเท่านั้น

3) แบบตะขอ (Hook) สำหรับเกี่ยวของ เช่น ของที่อยู่บนสายพาน ฯลฯ

4) แบบตัก (Scoop) สำหรับการลำเลียงของที่เป็นของเหลวหรือฝุ่นผง

2.1.6.2 มือที่เป็นอุปกรณ์ (Tool as End Effector)

เนื่องจาก Gripper มีข้อจำกัด ใช้ได้เฉพาะกับงานหยิบจับวัสดุเท่านั้น แต่การใช้งานของหุ่นยนต์มีขอบเขตกว้างขวางจึงได้มีการออกแบบอุปกรณ์ใช้งานติดกับส่วนปลายหรือส่วนมือนี้เพื่อใช้ในการทำงาน โดยอาจจะเป็นอุปกรณ์ที่ติดตายหรืออาจถอดเปลี่ยนได้ก็แล้วแต่การออกแบบตัวอย่างของมือที่เป็นอุปกรณ์นี้มีมากมายอาทิ เช่น

- ตัดอุปกรณ์เชื่อมเฉพาะจุด (Spot Welding Gun)
- ตัดอุปกรณ์พ่นสี (Spray Paint Gun)
- ตัดอุปกรณ์สว่าน (Drilling Spindle) ฯลฯ

2.1.7 การควบคุมการทำงานด้วยหุ่นยนต์

โดยปกติหุ่นยนต์อุตสาหกรรมมักจะทำงานเชื่อมกับระบบอื่น ๆ เช่น ระบบสายพาน(ในกรณีของขึ้น/ลง) เครื่องมือต่าง ๆ (ในการประกอบชิ้นงาน) ฯลฯ ซึ่งหมายความว่าการทำงานได้ต้องมีการควบคุมให้การทำงานของหุ่นยนต์และระบบข้างเคียงทำงานอย่างสอดคล้องกันได้ เพื่อให้งานมีลักษณะต่อเนื่อง ดังนั้นจึงต้องมีระบบควบคุมให้การทำงานแต่ละจุดได้จังหวะสอดคล้องกันทั้งระบบ ส่วนที่ไปควบคุมการทำงานนี้ เรียกว่า หน่วยควบคุม (Workstation Controller) และส่วนที่ถูกควบคุมทั้งหมด (อาทิ เช่น หุ่นยนต์ ระบบสายพาน ระบบเครื่องมือ ฯลฯ) นี้เรียกว่า เวิร์กสเตชัน (Workstation) หรือ (Work Cell)

2.1.8 การสั่งงาน

การสั่งงานให้หุ่นยนต์อุตสาหกรรมทำงานตามที่กำหนดนี้ อาจทำได้ในหลายลักษณะคือ

- แบบแมนนวล
- แบบลีดทรู (lead true)
- แบบการสั่งงานด้วยโปรแกรม

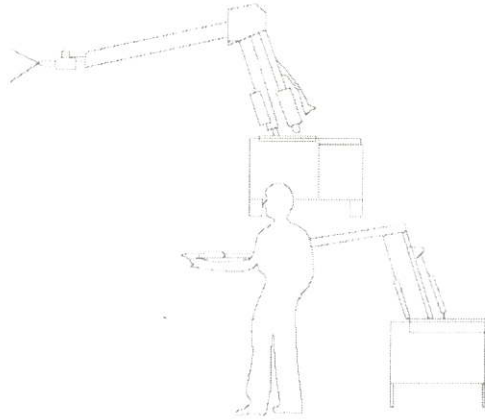
2.1.8.1 การสั่งงานแบบแมนนวล

การสั่งงานแบบนี้เป็นการสั่งงานที่ง่ายที่สุด และใช้กันมากที่สุดในการสั่งงานให้หุ่นยนต์ทำงานแบบจุดต่อจุด (point to point) โดยวิธีการจะเป็นการเคลื่อนที่ของแกนต่าง ๆ ไปยังตำแหน่งที่ต้องการ ไปยังตำแหน่งที่ต้องการ โดยการกดปุ่มบังคับ ดังนั้นลักษณะนี้จึงคล้ายกับการเซ็ทอัพ (Set up) มากกว่าการโปรแกรม กล่าวคือ มีการเคลื่อนที่ไปยังจุดเริ่มต้นแล้วก็มีกำหนดค่าไว้ จากนั้นก็เคลื่อนที่ไปยังตำแหน่งต่อไป แล้วกำหนดค่าไว้ จากนั้นก็เคลื่อนที่ไปยังตำแหน่งต่อไป แล้วกำหนดค่าไว้ โดยค่าที่กำหนดไว้จะเก็บไว้ในหน่วยความจำ เรียกออกมาใช้ภายหลังได้

2.1.8.2 การสั่งงานแบบลีดทรู

การสั่งงานแบบนี้คล้าย ๆ กับการจับมือหุ่นยนต์ให้เคลื่อนที่ไปตามแนวที่ต้องการในลักษณะที่เป็นการเคลื่อนที่แบบต่อเนื่อง แล้วจึงเก็บค่าตำแหน่งต่าง ๆ ที่เคลื่อนที่ แต่ในทางปฏิบัติ

ไม่สามารถทำได้เนื่องจากข้อจำกัดในแง่เทคนิค (อาทิเช่น ระบบเกียร์ ฯลฯ) ดังนั้นจึงมีการคิดค้น อุปกรณ์ซึ่งเลียนแบบแขนหุ่นยนต์โดยสามารถเคลื่อนที่ไปยังตำแหน่งต่าง ๆ ได้ดังรูปที่ 2.17 อุปกรณ์นี้เรียกว่า ไรบอทซิมูเลเตอร์ (Robot Simulator) ซึ่งการทำงานก็ใช้หลักการข้างต้น การสั่งงานชนิดนี้ใช้มากในงานด้านการพ่นสีชิ้นส่วนรถยนต์ อุปกรณ์ ฯลฯ



รูปที่ 2.17 ลักษณะการสั่งงานแบบลีดทรูโดยอาศัยไรบอทซิมูเลเตอร์

2.1.8.3 การสั่งงานด้วยโปรแกรม

การสั่งงานให้หุ่นยนต์อุตสาหกรรมทำงาน โดยการใช้โปรแกรมสั่งงานนั้นจะเป็นการให้ข้อมูลกับหุ่นยนต์เพื่อให้แขนของหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งที่ต้องการและทำงานตามที่กำหนด โดยในปัจจุบันได้มีผู้เขียนภาษาคอมพิวเตอร์เพื่อสั่งให้หุ่นยนต์ทำงานมากมายหลายภาษา โดยโปรแกรมจะถูกเขียนบนคอมพิวเตอร์แล้วส่งไปควบคุมหุ่นยนต์อีกต่อหนึ่ง

2.2 คิเนมาติกส์ของแขนกล(Robot Arm Kinematics)

คิเนมาติกส์ของแขนกล (Kinematic Arm) เป็นวิธีวิเคราะห์การเคลื่อนที่และตำแหน่งซึ่งสามารถนำมาใช้กับแขนกลโดยให้จุดที่กำหนดเป็นจุดอ้างอิงการเคลื่อนที่ ทั้งนี้การวิเคราะห์หาการเคลื่อนที่แบบคิเนมาติกส์นี้ จะอธิบายการเคลื่อนที่และตำแหน่ง (The Spatical Displacement) ซึ่งเป็นฟังก์ชันของเวลาแต่ละข้อ(joint) โดยการบอกถึงตำแหน่ง และทิศทางของเอ็นเอฟเฟคเตอร์ของแขนกลโดยใช้จุดกำเนิดเป็นจุดอ้างอิง

พิกัดโฮโมจีเนียส(Homogeneous Coordinates)

การหมุน(Rotation)ใช้อธิบายลักษณะการของวางตัว (Orientation) ของเครื่องมือแต่ในการอธิบายตำแหน่งของเครื่องให้สัมพันธ์กับโครงพิกัดที่อยู่กับฐานของแขนกลนั้นจำเป็นต้องใช้การถ่ายโอนอีกแบบอีกอย่างหนึ่งที่เรียกว่า การเลื่อนตำแหน่ง(Translation) การเลื่อนตำแหน่งนั้นมีข้อ

แตกต่างจากการหมุนที่สำคัญประการหนึ่งคือในการหมุนจุดกำเนิดของโครงพิกัดที่ถูกหมุนนั้นเป็นตำแหน่งตำแหน่งเดียวกับจุดกำเนิดของโครงพิกัดดั้งเดิม เราจึงสามารถแสดงการหมุนในปริภูมิสามมิติได้ด้วยเมตริกซ์ 3×3 แต่จุดกำเนิดของโครงที่เลื่อนตำแหน่งไปนั้นไม่เป็นจุดเดียวกันกับจุดกำเนิดของโครงพิกัดเดิม ดังนั้นจึงไม่สามารถแสดงการเลื่อนตำแหน่ง R^3 ด้วยเมตริกซ์ 3×3 ได้ในการนี้เราจะใช้ปริภูมิ 4 มิติ เพื่อช่วยในการนิยามพิกัดโฮโมจีเนียส

พิกัดโฮโมจีเนียส(Homogeneous Coordinates) ให้ q เป็นจุดใน R^3 และ F เป็นโครงพิกัดฉากสำหรับ R^3 ถ้า σ เป็นสเกลาร์เวกเตอร์ที่ไม่เป็นศูนย์ ดังนั้นพิกัดโฮโมจีเนียสของ q เทียบกับ F เขียนเป็น $[q]^F$ นิยามได้ว่า

$$[q]^F \equiv \sigma[q_1, q_2, q_3, 1]^T \quad (2.1)$$

ดังนั้น พิกัดโฮโมจีเนียสของ q ใน R^3 แสดงได้ด้วยเวกเตอร์ $[q]^F$ ในปริภูมิ 4 มิติ R^4 ในการแปลงพิกัดโฮโมจีเนียสขนาด 3×4 (Homogeneous coordinate conversion matrix)

$$H_\sigma = \frac{1}{\sigma} [I, 0] \quad (2.2)$$

จะสังเกตได้ว่า พิกัดโฮโมจีเนียส $[q]^F$ มีได้หลายค่า ขึ้นอยู่กับสเกลาร์แฟกเตอร์ $\sigma \neq 0$ ที่ใช้แต่จะให้ผลเป็นเวกเตอร์ 3 มิติ q เช่นกัน ในหุ่นยนต์มักจะใช้ $\sigma = 1$ ในกรณีนี้เราหาพิกัดโฮโมจีเนียส จากพิกัด 3 มิติได้โดยเติม 1 ให้เป็นสมาชิกตัวที่ 4 ในทำนองเดียวกันเราจะแปลงพิกัดโฮโมจีเนียส จาก 4 มิติเป็น 3 มิติได้ด้วยการตัดสมาชิกตัวที่ 4 ทิ้ง

ถ้าจุดในปริภูมิ 3 มิติ ถูกแสดงในเทอมของพิกัดโฮโมจีเนียส และเราต้องการเปลี่ยนโครงพิกัดให้ไปเป็นโครงอื่น เราจะใช้เมตริกซ์ถ่ายโอนโฮโมจีเนียส (Homogeneous transform matrix) ขนาด 4×4 โดยปกติเมตริกซ์ T นี้แปรได้เป็นเมตริกซ์ย่อย 4 ส่วนคือ

$$T \equiv \begin{bmatrix} R & p \\ \eta^T & \sigma \end{bmatrix} \quad (2.3)$$

โดย R คือ Rotation matrix แสดงการหมุนของโครงพิกัดเคลื่อนที่เทียบกับโครงพิกัดที่อยู่หนึ่ง

p คือ Translation vector แสดงตำแหน่งจุดกำเนิดของโครงพิกัดเคลื่อนที่เทียบกับโครงพิกัดอยู่หนึ่ง

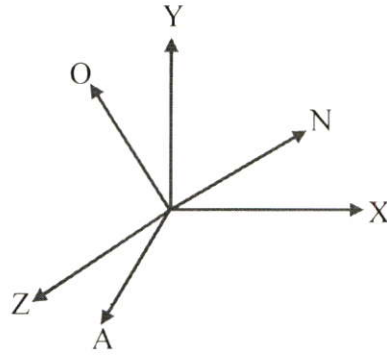
σ คือ Scalar factor ที่ไม่เป็นศูนย์ ปกติให้เท่ากับหนึ่ง

η^T คือ Perspective vector ปกติเท่ากับศูนย์

ในเทอมของแขนกล p แสดงตำแหน่งของปลายเครื่องมือ, R แสดงการหมุนของเครื่องมือ η แสดงจุดในการมองของอุปกรณ์ เช่น กล้องจับภาพ ในกรณีนี้จะให้ค่าที่ไม่เป็นศูนย์, ค่าของ σ ปกติใช้หนึ่ง เป็นค่ามาตรฐาน

การเลื่อนตำแหน่งและการหมุน(Translation and Rotations)

การกระทำพื้นฐานของการเลื่อนตำแหน่งและการหมุนจัดว่าเป็นกรณีพิเศษในเมตริกซ์ถ่ายโอนโฮโมจีเนียส 4×4 เช่น กรณีการหมุนสมมติให้ F และ M เป็นโครงพิกัดฉากที่เริ่มต้นทับกันสนิท



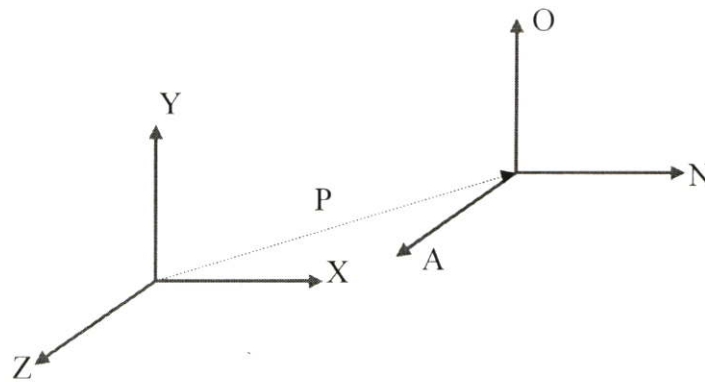
รูปที่ 2.18 แสดงการหมุนของโครงพิกัดเทียบกับโครงพิกัดอยู่หนึ่ง

ถ้าหมุน M ไป ϕ รอบเวกเตอร์ลำดับที่ k ของ F ดังนั้น จะแสดงในเทอมของกิกัดโฮโมจีเนียส เป็นเมตริกซ์ 4×4 , ดังนั้นเมตริกซ์การหมุนสามารถอธิบายได้ในสมการที่(2.5)

$$Rot(\phi, k) \cong \begin{bmatrix} & & 0 \\ R_k & & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad 1 \leq k \leq 3 \quad (2.4)$$

โดย $R_k(\phi)$ เป็นเมตริกซ์การหมุนพื้นฐานลำดับที่ k เราเรียก $Rot(\phi, k)$ ว่า เป็นเมตริกซ์การหมุนโฮโมจีเนียส พื้นฐานลำดับที่ k ของเมตริกซ์ผสมจะสามารถสร้างขึ้นได้ในทำนองเดียวกันกับอัลกอริทึมการหมุนผสม จะสังเกตเห็นว่าเวกเตอร์ตำแหน่ง p ในแต่ละกรณีเป็นศูนย์

อีกทั้งพิกัดโฮโมจีเนียสยังเป็นประโยชน์ในการในการแสดงการเลื่อนตำแหน่งได้ เช่น ให้ F และ M เป็นโครงพิกัดฉากที่ทับกันสนิท สมมุติให้เลื่อนจุดกำเนิดของโครงเคลื่อนที่ M ไป P_k ตามแนวยูนิต เวกเตอร์ลำดับที่ k ของ F เมื่อ $1 \leq k \leq 3$



รูปที่ 2.19 โครงพิกัดเคลื่อนที่เทียบกับโครงพิกัดอยู่หนึ่ง

ดังนั้นในเทอมของพิกัดโฮโมจีเนียสสามารถแสดงได้ด้วยเมตริกซ์ 4×4 , โดยที่เมตริกซ์เลื่อนตำแหน่งโฮโมจีเนียสนี้สามารถอธิบายได้ดังสมการที่(2.5)

$$Tran(p) \cong \left[\begin{array}{ccc|c} 1 & 0 & 0 & p_1 \\ 0 & 1 & 0 & p_2 \\ 0 & 0 & 1 & p_3 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (2.5)$$

เมื่อ $Tran(p)$ ว่าเป็นเมตริกซ์แสดงการเลื่อนตำแหน่งโฮโมจีเนียสพื้นฐาน(Fundamental homogeneous translation matrix) จะสังเกตได้ว่าเวกเตอร์ตำแหน่งใน (p) ในขณะที่เมตริกซ์การหมุน 3×3 จะเป็นเมตริกซ์เอกลักษณ์

สมการ Forward Kinematic ของ Denavit-Hartenberg

ในปี 1955 Denavit และ Hartenberg ได้คิดวิธีการหาสมการการเคลื่อนที่ของ Robot ต่อมาเทคนิคนี้ได้กลายมาเป็นมาตรฐานที่สำคัญในการเรียน โดยที่สามารถจะใช้สมการนี้ได้กับทุกเงื่อนไข โดยมีกฎดังนี้คือ

1. ทำการหมุนรอบแกน Z_n ให้เป็นตัวแปร Z_{n+1} โดยจะทำให้ X_n ขนานกับ x_{n+1}
2. ทำการเลื่อนตำแหน่งตามแนวแกน Z_n เป็นระยะ d_{n+1} ทำให้ X_n และ x_{n+1} มีจุดกำเนิดที่จุดเดียวกัน
3. ทำการเลื่อนตำแหน่งตามแนวแกน X_n เป็นระยะ a_{n+1} ทำให้ Z_n และ Z_{n+1} มีจุดกำเนิดที่จุดเดียวกัน
4. ทำการหมุนรอบแกน X_n ด้วยมุม α_n จะทำให้ frame n และ frame $n+1$ อยู่ที่จุดเดียวกัน

การวิเคราะห์แบบกีนมาติกส์ของแขนกล มี 2 ประเภท

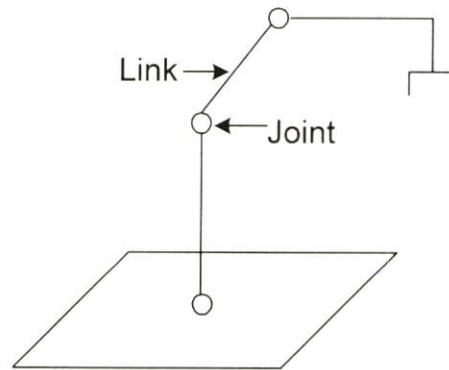
1. กีนมาติกส์แบบไดเรกหรือฟอร์เวิร์ด (Direct or Forward Kinematics)
2. กีนมาติกส์แบบอินเวิร์ส (Inverse Kinematics) or (Arm Solution)

จากการเคลื่อนที่ที่เป็นอิสระต่อกันในแต่ละข้อของแขนกล จึงต้องกำหนดเป้าหมายที่จะเคลื่อนไปโดยการกำหนด (The Reference Coordinate Frame) การหาตัวแปรในแต่ละข้อโดยใช้วิธีเมตริกซ์แบบ 4×4 โฮโมจีเนียส ทรานส์ฟอร์มเมชัน (Homogeneous Transformation Matrix)

2.2.1. กีนมาติกส์แบบ ไดเรก

แขนกลมีลักษณะเป็นวัตถุเกร็ง (Rigid body) เป็น link หลายๆ อันนำมาเชื่อมต่อกันเข้าด้วยกัน โดยใช้ข้อต่อโดยปลายข้างหนึ่งของ link ริมสุดจะติดแน่นกับฐานส่วนปลายของ link อีกอันที่อยู่คนละด้านจะเคลื่อนที่ได้อย่างอิสระปลายข้างที่เคลื่อนที่นี้จะต่ออยู่กับเครื่องมือ (Tool or End-

effector) ตามปกติข้อต่อ 2 ชนิดที่ใช้เชื่อมต่อ links คือ ข้อต่อหมุน (Revolute joint) และ ข้อต่อเลื่อน (Prismatic joint) ดังอธิบายในรูปที่ 2.20



รูปที่ 2.20 แขนกลที่มีลักษณะเป็น links

จุดประสงค์คือ การควบคุมทั้งตำแหน่งและการหมุนของเครื่องมือในระนาบ 3 มิติ เพื่อที่โปรแกรมให้เครื่องมือเคลื่อนที่ไปตามเส้นทางที่กำหนดไว้ เพื่อใช้ในการเคลื่อนย้ายวัตถุดิบในพื้นที่ทำงาน โดยการโปรแกรมการเคลื่อนที่ของเครื่องมือนี้จำเป็นจะต้องคำนวณความสัมพันธ์ระหว่างตัวแปรข้อต่อกับตำแหน่งการหมุนของอุปกรณ์ซึ่งเรียกว่า ปัญหา Direct Kinematics โดยมีนิยามดังนี้

ปัญหา Direct Kinematics คือ การกำหนดเวกเตอร์ของตัวแปรข้อต่อ (joint Variables) ของแขนกลโดยแยกตามลักษณะข้อต่อเป็น

- ข้อต่อหมุน ตัวแปรได้แก่ ค่าองศาการหมุนรอบแกนข้อต่อ
- ข้อต่อเลื่อน ตัวแปรได้แก่ ระยะการเลื่อนที่ไปตามแนวแกนข้อต่อแล้วนำมาคำนวณหา

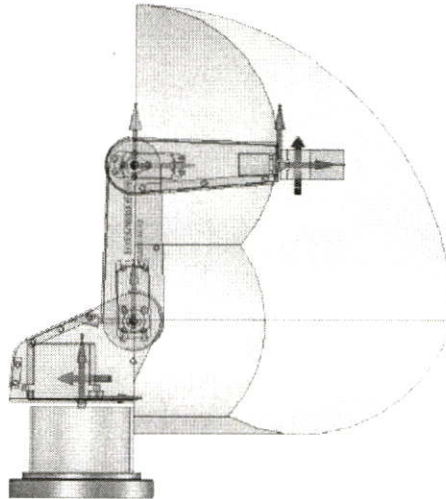
ตำแหน่ง (Position) และการหมุน (Orientation) ของเครื่องมือเทียบกับโครงพิกัดที่ติดอยู่กับฐานของแขนกล

จากสมการของ Denavit-Hartenberg

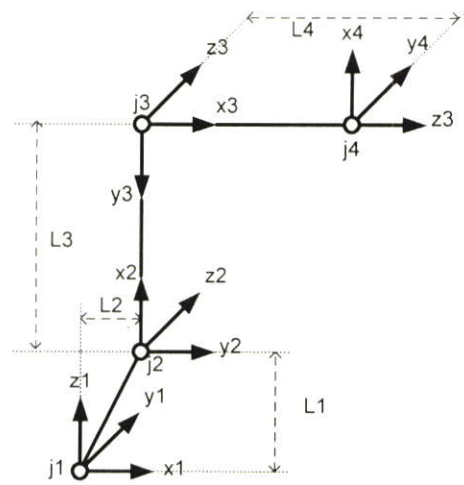
โดยการแสดงความสัมพันธ์ของการเคลื่อนที่จาก frame หนึ่งไปอีก frame หนึ่งแสดงได้โดยอธิบายในสมการที่(2.6) และลำดับขั้นตอนดังนี้

$$\begin{aligned}
{}^n T_{n+1} &= A_{n+1} = \text{Rot}(z, \theta_{n+1}) \times \text{Tran}(0, 0, d_{n+1}) \times \text{Tran}(a_{n+1}, 0, 0) \times \text{Rot}(x, \alpha_{n+1}) \\
&= \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1} & 0 & 0 \\ S\theta_{n+1} & C\theta_{n+1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & a_{n+1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_{n+1} & -S\alpha_{n+1} & 0 \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
A &= \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1}C\alpha_{n+1} & S\theta_{n+1}S\alpha_{n+1} & a_{n+1}C\theta_{n+1} \\ S\theta_{n+1} & C\theta_{n+1}C\alpha_{n+1} & -C\theta_{n+1}S\alpha_{n+1} & a_{n+1}S\theta_{n+1} \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}
\end{aligned}$$

โมเดลและพารามิเตอร์ของโครงพิกัดแขนกลแสดงดังรูปที่ 2.21 และ 2.22



รูปที่ 2.21 โมเดลโครงพิกัดของแขนกล



รูปที่ 2.22 พารามิเตอร์ของโครงฟัดแขนกล

ในทอมของฟัดโฮโมจีเนียส เราจะใช้เมตริกซ์ถ่ายโอนโฮโมจีเนียส(Homogeneous transform matrix) ขนาด 4x4 โดยเมตริกซ์ T นี้ในการอธิบายตำแหน่งความสัมพันธ์ตำแหน่งหรือลิงค์ข้างเคียงได้ ดังสมการที่ (2.7)

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$T_1^2 = \begin{bmatrix} \sin(\theta_2) & \cos(\theta_2) & 0 & L_2 \cos(\theta_2) \\ 0 & 0 & 1 & L_2 \sin(\theta_2) \\ \cos(\theta_2) & -\sin(\theta_2) & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$$T_2^3 = \begin{bmatrix} \sin(\theta_3) & -\cos(\theta_3) & 0 & L_3 \cos(\theta_3) \\ \cos(\theta_3) & \sin(\theta_3) & 0 & L_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$T_3^4 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & L_4 \cos(\theta_3) \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & L_4 \sin(\theta_4) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

จากนั้นนำ แต่ละ Link คูณกัน โดยเรียงตามลำดับของแต่ละ Link

$$T_{base}^{tool} = T_0^1 T_1^2 T_2^3 T_3^4 \quad (2.11)$$

ซึ่งจะได้ดังนี้จากสมการที่ (2.12) เมื่อนำมาคูณกันแล้วจะได้เมตริกซ์ขนาด 4x4 โดยนำ 1x4 เป็นค่าของ พิกัดของ X,Y,Z ดังสมการที่ (2.12)- (2.14)

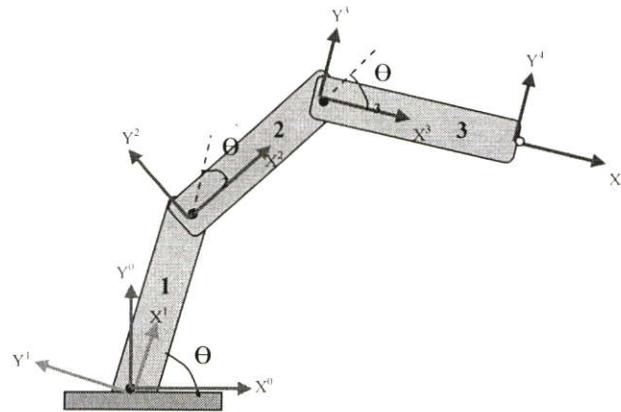
$$Px = (c_1 s_2 s_3 + c_1 c_2 c_3) L_4 + L_3 c_1 s_2 + L_2 c_1 \quad (2.12)$$

$$Py = (s_1 s_2 s_3 + s_1 s_2 c_3) L_4 + L_3 s_1 s_2 + L_2 s_2 \quad (2.13)$$

$$Pz = (c_2 s_3 - c_3 s_2) L_4 + L_3 c_2 + L_1 \quad (2.14)$$

2.2.2 อินเวอร์สคิเนมาติก (The Inverse Kinematics)

ทฤษฎีนี้ไม่มีคำอธิบายในรูปทั่วไป ต้องแก้สมการเพื่อหาค่าตัวแปรแต่ละข้อ ($q_1, q_2, q_3, \dots, q_n$) ของแขนกลนั้นๆ เองหรือใช้หลักการของตรีโกณมิติช่วยแก้ปัญหาได้โดยสิ่งที่ทราบคือ $L1, L2, L3, X, Y, Z$ และให้หาค่าของ $\theta_1, \theta_2, \theta_3, \theta_4$ ดังสมการที่ (2.15)-(2.18) และสมการที่ (2.12)- (2.14) ตัวแปรต่างๆ ในแต่ละข้อของแขนกลแสดงในรูปที่ 2.23



รูปที่ 2.23 ค่าตัวแปรแต่ละข้อของแขนกล

$$\theta_1 = \text{Atan } 2(p_y, p_x) \quad (2.15)$$

$$\theta_2 = \text{Atan } 2(-s_2, c_2) \quad (2.16)$$

เมื่อ

$$s_2 = (Pz - L_1)c_3L_4 + \left(\frac{Px}{c_1} - L_2\right)(s_3L_4 + L_3) \quad (2.17)$$

$$c_2 = \left(\frac{Px}{c_1} - L_2\right)c_3L_4 + (Pz - L_1)(s_3L_4 + L_3) \quad (2.18)$$

$$\theta_3 = \sin^{-1} \left[\frac{\left(\frac{Px}{c_1} - L_1\right)^2 + (Pz - L_2)^2 - L_3^2 - L_4^2}{2L_3L_4} \right] \quad (2.19)$$

ในส่วนของ θ_4 นั้นแล้วแต่เราจะกำหนดองศาของการหมุนของข้อต่อนี้

$$\theta_4 = A \tan 2(p_{w_y}, p_{w_x}) \quad (2.20)$$

2.3 โครงสร้างของระบบควบคุมแขนกล

สามารถแบ่งโครงสร้างของระบบออกเป็น 3 ส่วนดังนี้

2.3.1 ส่วนตรวจจับ (Sensor & Encoder)

เป็นส่วนประกอบที่สำคัญในระบบที่มีการป้อนกลับ ตัวตรวจจับที่ใช้กันมากในระบบควบคุมเช่น โปเทนทีโอมิเตอร์ (Potentiometer), ออปติคัลเอ็นโคเดอร์ (Optical encoder) เป็นต้น โดยสัญญาณอนาล็อกจากส่วนตรวจจับนี้จะถูกแปลงเป็นสัญญาณดิจิทัล หลังจากนั้นจะถูกส่งไปยังหน่วยประมวลผลกลางต่อไป

2.3.2 หน่วยประมวลผลกลางและควบคุม

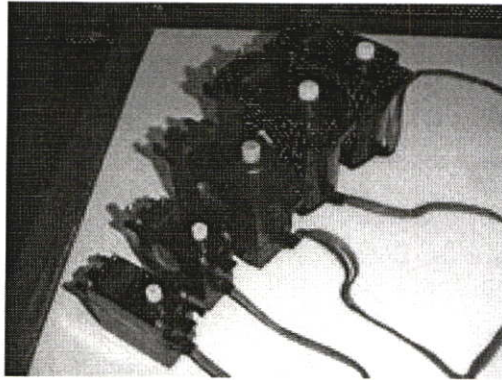
ได้แก่ คอมพิวเตอร์ (Computer) และไมโครคอนโทรลเลอร์ (Microcontroller) ที่ใช้ในการควบคุมการทำงานของแขนกล โดยการตัดสินใจของหน่วยประมวลผลกลางและนำมาใช้ควบคุม โดยระบบจะรับสัญญาณป้อนกลับจากส่วนตรวจจับมาใช้ในการประมวลผลแล้วส่งสัญญาณไปควบคุมการทำงานของแขนกลต่อไป

2.3.3 วงจรขับมอเตอร์

เป็นวงจรที่ควบคุมการหมุนของมอเตอร์ซึ่งเป็นอุปกรณ์ที่ทำให้แขนกลเกิดการเคลื่อนไหวมอเตอร์จะถูกติดตั้งสำหรับควบคุมการเคลื่อนไหวนของข้อต่อต่างๆ ของแขนกล โดยวงจรขับมอเตอร์จะได้รับสัญญาณควบคุมการทำงานจากหน่วยประมวลผลกลางและควบคุม

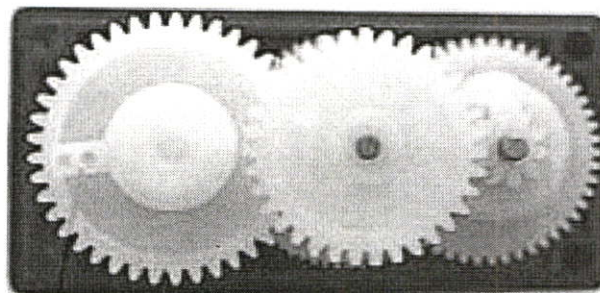
2.4 มอเตอร์ไฟฟ้ากระแสตรงแบบดีซีเซอร์โวมอเตอร์ (DC Servo Motor)

2.4.1 ดีซี เซอร์โวมอเตอร์ (DC Servo Motor)

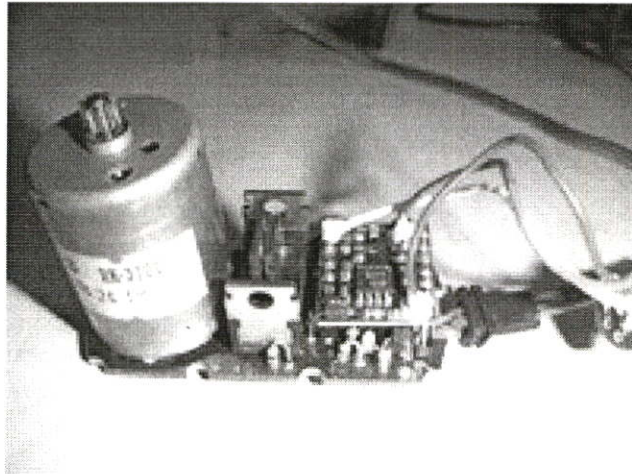


รูปที่ 2.24 ขนาดต่างๆของ DC Servo motor

DC Servo Motor เป็นชนิดหนึ่งของมอเตอร์ซึ่งแบ่งเป็นมอเตอร์ไฟฟ้ากระแสตรง(DC MOTOR) หรือกระแสสลับ (AC Motor) ภายใน Servo motor ก็จะมีส่วนของเฟืองเกียร์และบอร์ดควบคุม มีสายสัญญาณในการควบคุมการหมุนของ Servo motor 1 เส้น และอีก 2 เส้นเป็น VCC กับ Ground โดยตัวของ Servo motor จะสามารถหมุนได้ 180 องศา แต่จะมีบางรุ่นที่สามารถหมุนได้ถึง 210 องศา โดยสามารถที่จะหมุนไปทางซ้าย 90 องศาและทางขวาอีก 90 องศา ดังแสดงขนาดส่วนประกอบและชุดเฟืองเกียร์ของ DC Servo motor ในรูปที่ 2.24-2.26



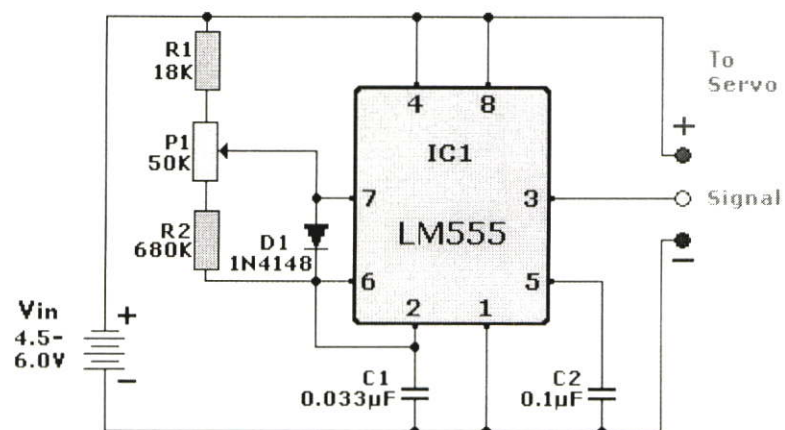
รูปที่ 2.25 ชุดเฟืองเกียร์ของ DC Servo motor



รูปที่ 2.26 ส่วนประกอบภายในของ DC Servo Motor

2.4.2 โครงสร้างระบบควบคุมของ Servo Motor

ระบบควบคุมเซอร์โวมอเตอร์จะเป็นระบบควบคุมแบบลูปปิด (Closed loop control) ประกอบด้วย 3 โหมดการควบคุมคือโหมดการควบคุมแรงบิด (Torque Control Mode) ซึ่งอยู่วงรอบหรือลูปล่าสุดโหมดการควบคุมอัตราเร่ง (Velocity Control Mode) และโหมดการควบคุมตำแหน่ง (Position Control Mode) ซึ่งอยู่ลูปลัดนอกสุด โดยมีอุปกรณ์ที่สำคัญๆ ดังรูปที่ 2.27



รูปที่ 2.27 โครงสร้างระบบควบคุมดีซีเซอร์โว

2.4.3 ส่วนประกอบของดีซีเซอร์โว (DC Servo Motor)

- 1) ดีซีเซอร์โวมอเตอร์ (DC Servo Motor)

2) ชุดควบคุมการขับเคลื่อนเซอร์โว (Servo Drive or Servo Amplifier)

3) อุปกรณ์ป้อนกลับ (Feedback Device เช่น Speed encoder และ Position Sensor)

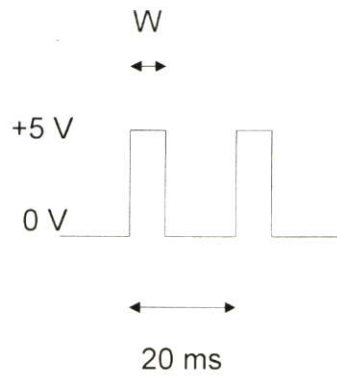
โดยทั่วไปจะมี Servo motor จะมีอยู่หลายแบบแตกต่างกันออกอย่างไรก็ตาม Servo Motor ก็จะมีคุณสมบัติคล้าย ๆ กันดังนี้

- Servo motor จะหมุนเมื่อมีสัญญาณ pulse เข้าไปโดยสัญญาณความกว้าง pulse นี้จะทำหน้าที่ควบคุมการหมุนและทิศทางของ Servo motor เพื่อให้ motor หมุนไปตามองศาที่เราต้องการ
- Servo motor กินไฟประมาณ 4-6 volt กินกระแส 9.7 mA (Idle) และ 130 mA (Moving)
- Servo motor จะสามารถหมุนได้ 180 องศา (บางรุ่นหมุนได้ 210 องศา) ที่เป็นเช่นนี้ก็เพราะป้องกันความเสียหายที่อาจเกิดขึ้นกับวงจรภายใน
- Servo motor ยังมีขนาดเล็กและน้ำหนักเบา

2.4.4 หลักการทำงานของ Servo Motor

Servo จะถูกควบคุมด้วยวงจรถอนิกส์ และตัววัดระดับไฟฟ้า (Potentiometer) ซึ่งปลายของ Motor จะเชื่อมต่อกับแกนที่ยื่นออกมาผ่านชุดเฟือง และส่วนหนึ่งของชุดเฟืองก็จะทำการหมุนแกนของ Potentiometer ซึ่ง Potentiometer จะคอยตรวจสอบตำแหน่งเชิงมุมของ Servo ในขณะนั้น หรือตำแหน่งการทำงานของ motor นั้นเอง เมื่อแกนของ Servo หมุนมาอยู่ในตำแหน่งที่ถูกต้อง Motor ของ Servo ก็จะหยุดการทำงาน และถ้าวงจรตรวจพบว่าถ้ามุมของของแกน Servo ไม่ถูกต้อง วงจรจะสั่งงานให้ motor หมุนไปเรื่อยๆจนมุมของ Servo อยู่ในตำแหน่งที่ถูกต้อง แกนของ Servo นั้นสามารถหมุนได้เป็นมุมมากถึง 180 องศา และ Servo บางรุ่นสามารถหมุนได้เป็นมุมกว้างถึง 210 องศา ซึ่งจะแตกต่างกันไปตามที่ผู้ประดิษฐ์ได้ออกแบบมา แต่โดยทั่วไปแล้ว Servo ที่ถูกเลือกใช้สำหรับควบคุมการเคลื่อนไหวนั้นจะอยู่ระหว่าง 0 - 180 องศา ซึ่ง Servo ทั่วไป จะไม่สามารถหมุนทำมุมได้มากกว่านี้ เนื่องจากถูกควบคุมที่ตำแหน่งชุดเฟืองเกียร์

เราสามารถควบคุมการทำงานของ Servo motor ได้โดยป้อนสัญญาณความกว้าง pulse ให้กับ Servo motor ซึ่งความกว้างของสัญญาณ pulse นี้จะไปกำหนดตำแหน่งของ Servo motor ว่าจะหมุนไปอยู่ตำแหน่งใด ซึ่งเราสามารถป้อนความกว้าง pulse ได้ตั้งแต่ 5.0 ms - 2.5 ms โดยที่คาบเวลา (Period) ในการส่งสัญญาณความกว้าง pulse เท่ากับ 20 ms ดังรูปที่ 2.28

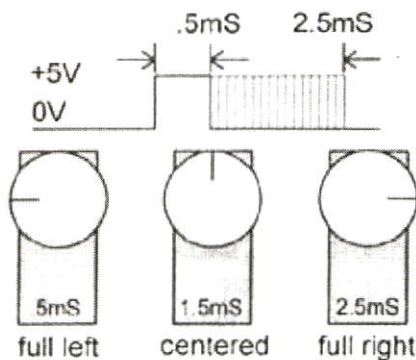


รูปที่ 2.28 การส่งสัญญาณ pulse เพื่อควบคุมตำแหน่ง DC Servo Motor

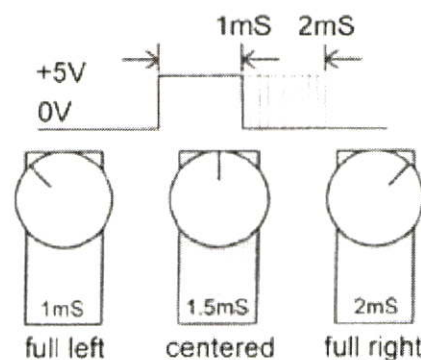
การควบคุมการทำงานจะมีจุดที่อ้างอิง 3 จุดด้วยกันดังแสดงในรูปที่ 2.29 และสามารถอธิบายขั้นตอนการทำงานดังต่อไปนี้

- 1) ตำแหน่งซ้ายสุด เราสามารถกำหนดให้ Servo motor เคลื่อนที่ไปยังตำแหน่งซ้ายสุด (-90 องศา) ได้โดยการป้อนความกว้าง pulse เท่ากับ 0.5 ms
- 2) ตำแหน่งตรงกลาง เราสามารถกำหนดให้ Servo motor เคลื่อนที่ไปยังตำแหน่งตรงกลาง (0 องศา) ได้โดยการป้อนความกว้าง pulse เท่ากับ 1.5 ms
- 3) ตำแหน่งขวาสุด เราสามารถกำหนดให้ Servo motor เคลื่อนที่ไปยังตำแหน่งขวาสุด (90 องศา) ได้โดยการป้อนความกว้าง pulse เท่ากับ 2.5 ms

จากจุดอ้างอิงทั้ง 3 เราสามารถที่จะคำนวณตำแหน่งที่ต้องการให้ Servo motor หมุนไปได้ เช่น ถ้าเราต้องการให้ Servo motor หมุนไปยังตำแหน่ง -45 องศา เราก็ทำการป้อน $(0.5+1.5)/2 = 1.0$ ms ให้กับ Servo motor



รูปที่ 2.29 ตำแหน่งการหมุน 90 องศา



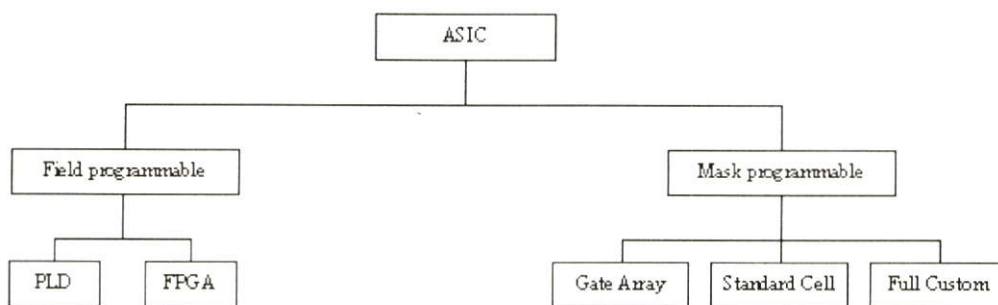
รูปที่ 2.30 ตำแหน่งการหมุน 45 องศา

2.5 อุปกรณ์ FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามที่ผู้ออกแบบต้องการ ในการทำ FPGA เมื่อเทียบกับการทำ ASIC (Application Specific Integrated Circuits) แล้วนั้นก็ทั้งข้อดีและข้อเสีย คือ การทำ FPGA จะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในอุปกรณ์ FPGA จะมีจำนวน gate ให้ใช้จำนวนจำกัด และการทำ FPGA ก็เหมาะกับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำ FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียน code อธิบายฮาร์ดแวร์จนกระทั่ง download นั้นน้อยกว่าการทำ ASIC มาก และการตรวจสอบหรือแก้ไข design ก็ทำได้สะดวก การทำ FPGA ในปัจจุบันมีประสิทธิภาพมากขึ้นและสะดวกขึ้น ทั้งนี้เนื่องจากทางบริษัทผู้ผลิตอุปกรณ์ FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบภายใน หรือ ปรับปรุงโครงสร้างสถาปัตยกรรมภายใน และยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ทำ PPR (Partitioning, Placement and Routing) สำหรับอุปกรณ์นั้นๆด้วย

2.5.1 ASIC

ความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ปัจจุบันทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ต่างๆ มากมายซึ่งทำให้เกิดการลดค่าใช้จ่าย การสิ้นเปลืองพลังงานและขนาด ในขณะเดียวกันก็มีการเพิ่มประสิทธิภาพและระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจากเทคโนโลยีไมโครโพรเซสเซอร์และหน่วยความจำปัจจุบัน ทุกๆ ครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างวงจรรวมและไอซีมาตรฐานมากขึ้น ในการพัฒนาเพิ่มความหนาแน่นและจำนวน ฟังก์ชันลอจิก ที่เหมาะสม นักออกแบบอุปกรณ์ทางด้านดิจิทัลได้พิจารณาถึงการผลิตให้ขนาดมากๆ และการผลิตวงจรรวม (ASIC: Application Specific Integrated Circuit) ซึ่งวงจรรวม จะแบ่งตามการสร้างออกเป็น 2 กลุ่ม คือ Field programmable และ Mask programmable ดังแสดงในรูปที่ 2.33



รูปที่ 2.31 ชนิดของวงจรรวม(ASIC: Application Specific Integrated Circuit)

Programmable Logic สามารถแบ่งได้เป็นหลายชนิด โดยแบ่งตามเทคโนโลยีที่ใช้ผลิต ความซับซ้อนของอุปกรณ์ หรือโครงสร้างภายในอุปกรณ์เอง ซึ่งสามารถแบ่งได้ดังนี้

- Simple Programmable Logic Device (SPLD)
- Complex Programmable Logic Device (CPLD)
- Field Programmable Gate Array (FPGA)
- Field Programmable Interconnects (FPIC)

FPGA มีโครงสร้างที่แตกต่างไปจาก SPLD และ CPLD และมักจะมีควมหนาแน่นของเกตมากกว่า โครงสร้างของ FPGA จะประกอบไปด้วย แผงของบล็อกลอจิก ล้อมรอบด้วยส่วน I/O ที่สามารถโปรแกรมได้ และมีการเชื่อมต่อแบบโปรแกรมได้เช่นกัน FPGA มีขนาดตั้งแต่ระดับ 1000 เกตจนถึงหลายล้านเกต

2.5.2 การแบ่งหมวดของ FPGA

การแบ่งหมวดของ FPGA นั้นสามารถแบ่งแยกตามคุณลักษณะได้ดังนี้

- โครงสร้างสถาปัตยกรรมภายใน
- เทคโนโลยีการ โปรแกรม
- I/O
- Interconnect

2.5.2.1 FPGA สามารถแบ่งตามโครงสร้างสถาปัตยกรรมภายในของอุปกรณ์ดังนี้

- LCA (Logic Cell Array)
- PASIC (Programmable ASIC)
- FLEX, APEX
- ACT (Actel)

2.5.2.2 การแบ่งโครงสร้างสถาปัตยกรรมของ FPGA

การแบ่งโครงสร้างสถาปัตยกรรมของ FPGA แบ่งได้เป็น 2 ลักษณะคือ

2.5.2.2.1 Coarse-grained

โครงสร้างสถาปัตยกรรมแบบ Coarse-grained จะประกอบด้วยลอจิกบล็อกขนาดใหญ่ เช่นมักประกอบด้วย LUT และ Flip-Flops 2 อันหรือมากกว่า ตัวอย่าง FPGA ได้แก่ Xilinx ตระกูล XC4K, Spartan, Virtex หรือ Altera ตระกูล FLEX, APEX

2.5.2.1.2 Fine-grained

ประกอบด้วย ลอจิกบล็อกแบบง่าย จำนวนมาก ลอจิกบล็อกประกอบด้วย ลอจิก ฟังก์ชัน หรือ 2 อินพุต หรือมัลติเพล็กซ์เซอร์แบบ 1-4 และฟลิปฟล็อป ตัวอย่าง FPGA แบบ fine-grained เช่น ตระกูล ACT ของบริษัท ACTEL

2.5.2.3 เทคโนโลยีการโปรแกรม (Programming Technology)

เทคนิคการโปรแกรมไอซี FPGA มีดังนี้

- SRAM
- Fuse/Anti-fuse
- EPROM/EEPROM
- FLASH

2.5.3 ข้อแตกต่างอื่นๆ ของโครงสร้างสถาปัตยกรรมของ FPGA

ข้อแตกต่างอื่นๆ ของโครงสร้างสถาปัตยกรรมของ FPGA คือ เทคโนโลยีที่ใช้ในการผลิตตัวชิป FPGA ซึ่งได้แก่ SRAM และ anti-fuse FPGA ที่ใช้เทคโนโลยี

2.5.3.1 FPGA ชนิด SRAM

นั้นสามารถทำการโปรแกรมซ้ำๆ ได้หลายครั้ง แต่ในการใช้งานในภาคสนามจำเป็นต้องใช้ไอซี สำหรับเก็บข้อมูลคอนฟิกูเรชันของวงจร การโหลดข้อมูลจากตัวเก็บบิตสามารถเกิดขึ้นได้โดยอัตโนมัติ เมื่อเปิดสวิตช์ หรืออาจโหลด โปรแกรมผ่านไมโคร โปรแกรมเมอร์ก็ได้ เช่น โหลดโปรแกรมด้วย PC ผ่านสายคานน์ โหลด ไอซี FPGA นอกจากจะมีขนาดของเกตสูงแล้ว ยังสามารถโปรแกรมซ้ำได้บ่อยตามที่ต้องการ

2.5.3.2 FPGA ชนิด Fuse หรือ Anti-fuse

จะสามารถทำการโปรแกรมได้เพียงครั้งเดียว (OTP) และไม่สามารถแก้ไขหรือโปรแกรมซ้ำได้อีก แต่ข้อมูลการโปรแกรมไม่สูญหายเมื่ วงจรถูกตัดแหล่งจ่ายไฟ การโปรแกรมไอซี FPGA ชนิดนี้จะต้องใช้เครื่องโปรแกรมไอซี

2.5.3.3 FPIC (Field Programmable Interconnect device)

FPIC จริงๆ แล้วไม่ได้ไอซีทางด้านลอจิก แต่จะเป็นอุปกรณ์ที่สามารถโปรแกรมส่วนที่เชื่อมต่อ(wiring) เช่นการเชื่อมต่อให้ไอซีหนึ่งตัวสามารถเลือกได้ว่าให้เชื่อมต่อเข้ากับไอซีตัวไหนได้ บริษัทที่ผลิตอุปกรณ์ประเภทนี้ได้แก่ ไอซี Digital Crosspoint Switch ของบริษัท I-Cube Digital crosspoint device ของบริษัท Lattice หรือ อุปกรณ์ Hardware emulator ของบริษัท Aptix

2.5.4 การออกแบบด้วย FPGA

การออกแบบวงจรดิจิทัลด้วย FPGA โดยทั่วไปมีองค์ประกอบ 3 ส่วน

2.5.4.1 ซอร์ฟแวร์ Design Entry

- โดยใช้ Schematic Design Entry ใช้ไลบรารีของ FPGA
- ใช้ภาษา HDL เช่น VHDL, Verilog

การออกแบบโดยใช้ภาษาชั้นสูงนั้นการทดสอบของวงจรนั้นยังไม่ขึ้นกับเทคโนโลยีเป้าหมาย (Technology independent) ผู้ออกแบบไม่จำเป็นต้องกังวลถึงค่าความหน่วงทางเวลาของอุปกรณ์ที่มาถึงเทคโนโลยีนั้นการทดสอบความถูกต้องเป็นในลักษณะการตรวจสอบระดับฟังก์ชันการทำงานโดยใช้ซอฟต์แวร์สำหรับจำลองการทำงาน (Simulation)

2.5.4.2 Design Implement

ขั้นตอนนี้ ต่อเนื่องจากขั้นตอนที่ 1 ซึ่งเกี่ยวข้องกับการแปลงแบบที่ได้ออกแบบจาก Schematic หรือ HDL ให้เป็นลอจิก ซึ่งอาจใช้ซอฟต์แวร์สำหรับสังเคราะห์วงจร (Logic synthesis) แล้วทำการแบ่งลอจิกเป็นส่วนๆ (Partitioning) และวางตำแหน่ง (Placement) ของลอจิกทำการเชื่อมต่อสายสัญญาณ (routing) สุดท้ายเป็นการสร้างไฟล์สำหรับโปรแกรมลงชิป (bit file)

2.5.4.3 Device Programming

การโปรแกรมอุปกรณ์หรือชิป FPGA นั้น มีเทคนิคหรือวิธีใหญ่ๆ 3 ลักษณะ ทั้งนี้ตัวชิปจะต้องสนับสนุนการทำงานในโหมดของการโปรแกรมเหล่านี้ด้วย

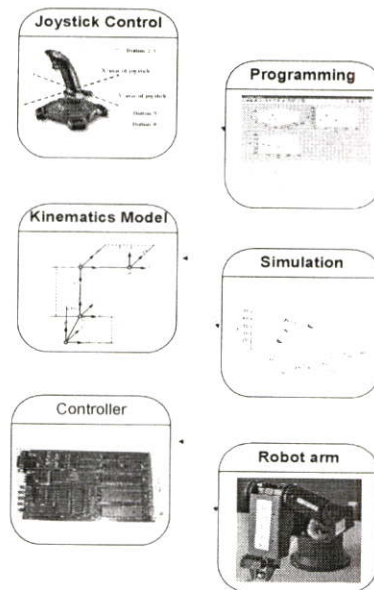
- การโปรแกรมโดยผ่านสายดาวน์โหลด หรือผ่าน JTAG หรือผ่าน ISP
- การโปรแกรมโดยใช้ ตัวเก็บข้อมูลฟลैชบิต
- การ โปรแกรมโดยใช้เครื่องโปรแกรมไอซี

บทที่ 3

หลักการในการออกแบบแขนกล

3.1 บทนำ

หลักของการออกแบบ Robot Arm มีส่วนประกอบหลักๆ อยู่ด้วยกันทั้งหมด 3 ส่วนด้วยกันคือ



รูปที่ 3.1 บล็อกไดอะแกรมของระบบโดยรวม

3.1.1 ส่วนของการเคลื่อนที่และการทำงานของหุ่นโดยรวม

ในส่วนนี้เราจะทำการกำหนดว่าแขนกลของเราจะมีการเคลื่อนที่อย่างไร การเคลื่อนที่เป็นแบบจุดต่อจุดหรือแบบต่อเนื่อง และจะนำแขนกลไปใช้ทำอะไร โดยการศึกษาส่วนนี้เราต้องใช้ข้อมูลของแขนกลทางด้าน Hardware มาประกอบการออกแบบ เมื่อเราได้แนวคิดทั้งหมดแล้วเราก็จะนำไปหาสมการของการเคลื่อนที่เพื่อใช้ในการเขียนโปรแกรมควบคุมหุ่นยนต์ต่อไป

3.1.2 ส่วนของแขนกล (Hardware)

ในการออกแบบส่วน Hardware ของแขนกลนี้ก็มีความสำคัญเช่นกัน ต้องมีการเลือกมอเตอร์ที่ใช้ในการเคลื่อนแขนกล การเลือกชุดขับเคลื่อนมอเตอร์ การเชื่อมต่อคอนโทรลเลอร์กับ RS 232 จากโปรแกรม และส่วนต่างๆ อีกดังนั้นในส่วนนี้เราต้องมีการออกแบบเพื่อให้มีความเหมาะสมเช่นกัน

3.1.3 ส่วนบอร์ดควบคุมการเคลื่อนที่

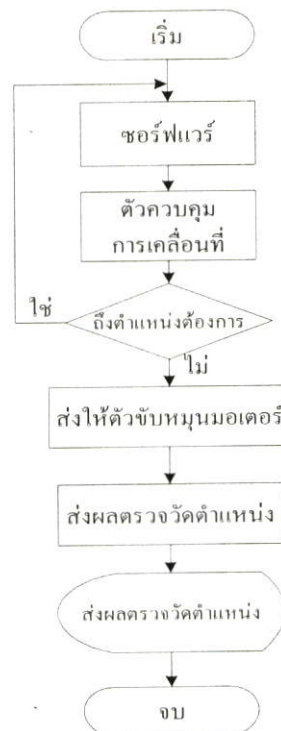
ในส่วนนี้เราจะนำเสนอสมการที่เราได้ออกแบบจากส่วนของเคลื่อนที่มาเขียนเป็น โปรแกรมเพื่อใช้ควบคุมแขนกล โดยโปรแกรมที่เราใช้คือ Matlab ใช้การสื่อสารข้อมูลแบบ RS 232

การที่จะให้แขนกลเคลื่อนที่ได้มันจะต้องมีการกำหนดจุดหรือตำแหน่งของการเคลื่อนที่ที่แน่นอน (x, y, z) จากโปรแกรมก่อน แล้วจึงส่งค่าผ่านพอร์ตอนุกรมของคอมพิวเตอร์ก่อนส่งไปยัง Motion Controller เพื่อให้ไปควบคุมการทำงานของมอเตอร์ต่อไป ดังนั้นในโครงงานนี้ ก็จะเน้นไปที่การออกแบบสมการการเคลื่อนที่และ โปรแกรมสำหรับควบคุมตำแหน่งของแขนกล หรือการกำหนดตำแหน่ง (x, y, z) ให้แขนกลเป็นหลักส่วนทางด้าน HardWare จะเป็นการปรับปรุงมาจากแขนกลต้นแบบ ที่มีอยู่แล้วนำมาปรับปรุงและพัฒนาต่อ

3.2 การออกแบบส่วนควบคุมการเคลื่อนที่และตำแหน่งของแขนกล

3.2.1 ควบคุมการเคลื่อนที่และการทำงานของแขนกลโดยรวม

ส่วนนี้จะเป็นส่วนที่ทำให้ทราบถึงบทบาทและหน้าที่ในการทำงานของแขนกล ซึ่งการทำงานในรูปแบบต่างๆ นั้น จะสามารถแสดงให้เห็นในส่วนนี้การทำงานสามารถอธิบายได้ดังนี้



รูปที่ 3.2 แสดงการทำงาน โดยรวมของ Robot Arm

การทำงานของโครงงานโดยรวมเริ่มจากผู้ควบคุมการทำงานสั่งการทำงานบนคอมพิวเตอร์ผ่านโปรแกรม Matlab ที่เขียนขึ้นมา ซึ่งสามารถควบคุมได้ทั้งแบบ Auto หรือ Manual จากนั้นก็ส่งงานผ่านพอร์ตอนุกรม โดยผ่านสาย RS-232 ส่งข้อมูลไปให้ชุด Motion Controller เพื่อกำหนดการเคลื่อนที่ของมอเตอร์ว่าให้เคลื่อนที่ไปตรงตามที่เราร้องการ โดยตัวที่ทำให้เกิดการหมุนของเซอร์โวมอเตอร์คือ การ์ด FPGA ตัวการ์ดนี้จะทำหน้าที่ส่งสัญญาณ pulse ไปยัง Servo Motor เพื่อควบคุมให้การหมุนได้ตามองศาตรงตามกำหนดซึ่งถ้ายังหมุนไม่ได้ตามองศาที่กำหนดของแกนกลที่กระทำมุมระหว่างกันแต่ละข้อ โปรแกรมจะทำการประมวลผลและส่งสัญญาณ pulse ไปอีกที

3.2.2 การออกแบบสมการของการเคลื่อนที่ของแขนกล

ในส่วนนี้เราจะนำทฤษฎีบทของ Kinematic มาหาสมการทางคณิตศาสตร์ของการเคลื่อนที่ของแขนกล ให้สามารถเคลื่อนที่ไปได้ในขอบเขตที่กำหนดและเป้าหมายที่ต้องการเคลื่อนที่ไปยังตำแหน่งสุดท้ายที่เราต้องการ โดยการหามุมของแต่ละข้อของแขนกลในส่วนนี้เป็นส่วนที่สำคัญ เพราะเป็นหัวใจหลักของงานวิจัยนี้ ซึ่งสามารถออกแบบสมการของการเคลื่อนที่ Kinematic ได้ดังนี้

3.2.2.1 สมการของการเคลื่อนที่ Forward Kinematic

จากทฤษฎีบทของ Kinematic ในส่วนของ Forward Kinematic นั้นจะเป็นหาดำแหน่งจุดปลายที่ Gripper ของแขนกลว่าเคลื่อนไปได้ที่ตำแหน่งใด จากการคำนวณย้อนกลับมาจาก Inverse Kinematic ซึ่งจะได้มุมที่กระทำกันแต่ละลิง เพื่อนำมาหาดำแหน่งที่เคลื่อนไปถึง ซึ่งจะได้ดังนี้จากสมการที่ (2.12) เมื่อนำมาคูณกันแล้วจะได้เมตริกซ์ขนาด 4×4 โดยนำ 1×4 เป็นค่าของ พิกัดของ X, Y, Z ดังสมการ (2.13), (2.14), (2.15) และนำไปเขียนในโปรแกรม Matlab

3.2.2.2 สมการของการเคลื่อนที่ Inverse Kinematic

ในส่วนของสมการของการเคลื่อนที่ในส่วนของ Inverse Kinematic นั้น ผลที่ได้จากสมการนี้จะเป็นการหามุมที่แต่ละข้อของแขนกลจะทำมุมกันได้ที่องศา เพื่อให้ตำแหน่งปลายสุดของแขนกลเคลื่อนไปยังตำแหน่งที่ต้องการได้อย่างถูกต้อง โดยตัวแปรของ $\theta_1, \theta_2, \theta_3, \theta_4$ ซึ่งจะได้ดังสมการ(2.16), (2.17), (2.20), (2.21) และนำไปเขียนในโปรแกรม Matlab

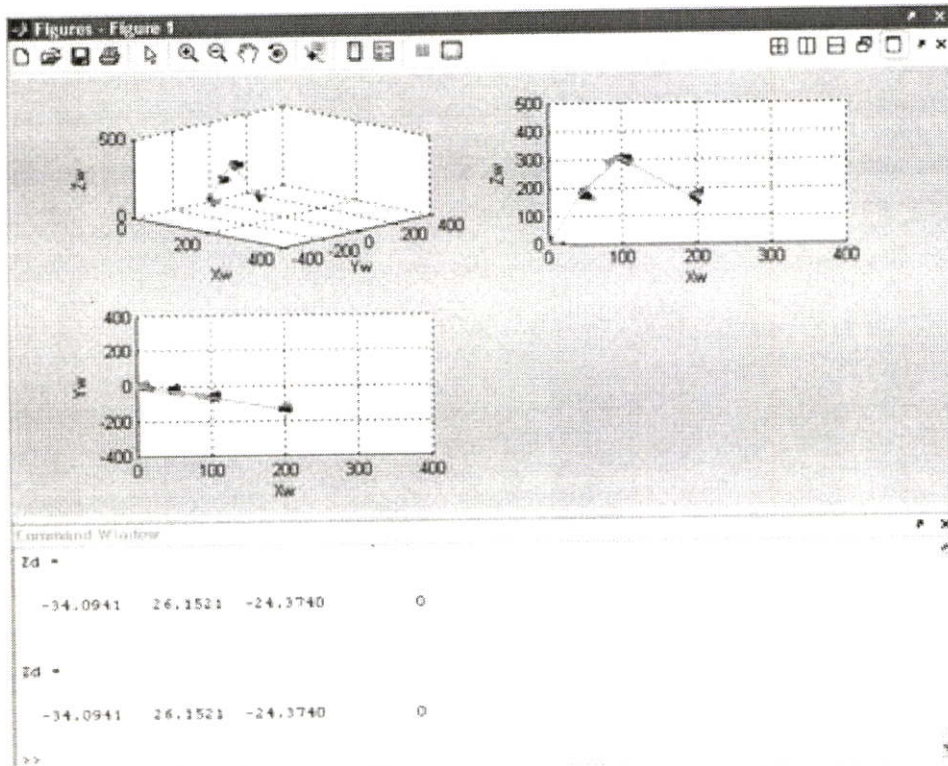
3.3 ส่วนของซอฟต์แวร์

ส่วนนี้จะเป็นส่วนการควบคุมการทำงานของแขนกลและทำการประมวลผลการเคลื่อนที่ของแขนที่ได้แสดงเป็นกราฟิกผ่านออกมาทางจอภาพ การออกแบบจะต้องเริ่มศึกษาจากการสมการของเคลื่อนที่ของแขนกลเสียก่อน จากนั้นนำสมการของการเคลื่อนที่ที่ได้นั้นนำมาเขียนในโปรแกรมเพื่อโปรแกรมนั้นทำการประมวลผลเพื่อส่งคำสั่งไปยังชุด Motion Controller ในส่วนของการทำงานของแขนกลนี้คือ การหยิบจับวัตถุจากที่หนึ่งไปยังอีกที่หนึ่ง โดยใช้ Joystick ในการบังคับตัวแขน

โปรแกรมนี้เขียนขึ้นมาเพื่อจะเน้นประโยชน์หลักจากการควบคุมการทำงานของแขนกลผ่านทางจอภาพซึ่งสามารถควบคุมได้ในระยะไกลโดยไม่ต้องไปอยู่ในพื้นที่หน้างานจริง ซึ่งสามารถดูกราฟที่แสดงขึ้นมาผ่านทางหน้าจอได้ดังรูปที่ 3.2

3.3.1 ส่วนของโปรแกรมหลัก

ในส่วนของโปรแกรมหลักนี้ในรูปที่ 3.2 จะเป็นส่วนของการแสดงผลกราฟที่เกิดขึ้นของการกระทำของแขนกลตามองศาที่เป็นไปและส่วนล่างเป็นแสดงค่าขององศาแต่ละแกน



รูปที่ 3.3 แสดงหน้าจอโปรแกรมควบคุมหลัก

3.3.1.1 การทำงานของโปรแกรมหลัก

การทำงานของโปรแกรมหลักนั้นเริ่มจากผู้ใช้งานทำการรันโปรแกรมขึ้น โปรแกรมจะทำการติดต่อกับ Joystick และ ตัวชุด Motion Controller เมื่อทำการโยก Joystick โปรแกรม Matlab จะทำการรับค่าจาก Joystick โดยในส่วนของคันโยกจะเป็นค่าระหว่าง -1 ถึง 1 และในส่วนของปุ่มกดจะเป็นค่าคงที่ของแต่ละปุ่มกดซึ่งเมื่อโปรแกรม Matlab รับค่ามาจะรู้ได้ว่าปุ่มใดสั่งการ เมื่อโปรแกรมหลักรับค่ามาจะทำการคำนวณคำสั่งที่สั่งการเข้ามาเพื่อหาตำแหน่งที่ต้องการเคลื่อนที่ไป

จากโปรแกรมที่เราได้เขียนลงไป แล้วทำการส่งตำแหน่งที่ได้ในรูปของ ค่าตัวแปรของแต่ละแกน ไปยังชุด Motion Controller เพื่อให้ชุด Motion Controller ทำการสร้างเป็นสัญญาณ Pulse การ์ด FPGA เพื่อทำการสร้างสัญญาณ เพื่อไปควบคุมตำแหน่งการทำงานของ DC Servo Motor ต่อไป

3.4 ส่วนของแขนกลและระบบการควบคุมแขนกล (HardWare)

ในส่วนนี้เป็นส่วนทางด้าน Hardware ซึ่งเป็นส่วนที่สำคัญกันไม่แพ้กับส่วนอื่นเช่นกัน ซึ่งในส่วนของ Hardware นี้จะประกอบไปด้วยชุด Motion Controller และ ตัว Robot Arm

3.4.1 ส่วนตัวแขนกล (Robot Arm)

ในส่วนของแขนกลนี้ซึ่งประกอบไปด้วย 4 แกน คือ X , Y , Z , Position , Grip โดยเป็นแขนกลชนิด Articulated Robot คือหมุนรอบแกนในตัวเอง



รูปที่ 3.4 แสดงตัวแขนกล 5 แกน

ในส่วนของมอเตอร์ที่ใช้ในการหมุนของแต่ละแกนนั้นใช้เป็น Servo Motor ในการขับเคลื่อนไปตามองศาที่เราสั่งการซึ่งในโครงการนี้กำหนดใช้มอเตอร์สำหรับแกนต่างๆ ดังนี้

- แกน 1 ใช้ Servo : GWS รุ่น S666-N
- แกน 2 ใช้ Servo : GWS รุ่น S666-N
- แกน 3 ใช้ Servo : GWS รุ่น S04-BBM
- แกน 4 ใช้ Servo : GWS รุ่น S03T-2BB

- แกน Gripper ใช้ Servo : GWS รุ่น S02N-STD

3.4.2 ส่วนของชุดควบคุมแขนกล (Motion Controller)

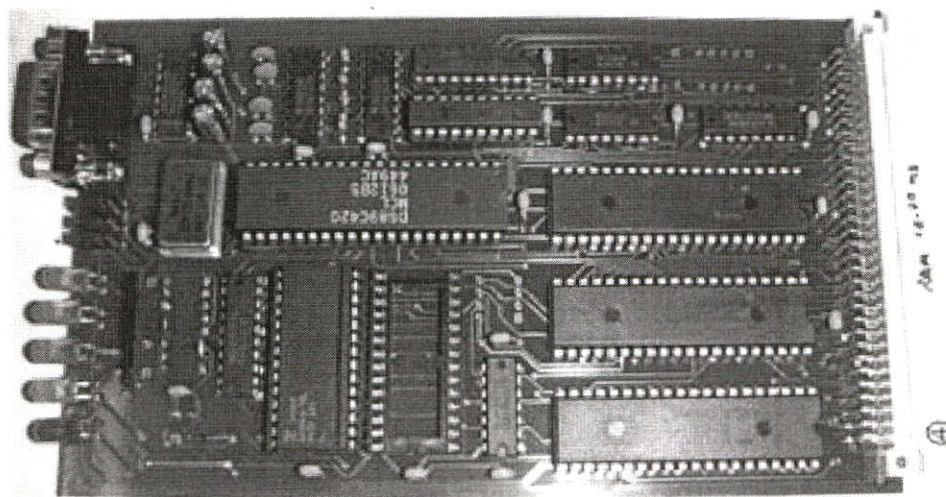
ในส่วนของชุด Motion Control นั้นเป็นส่วนที่ทำหน้าที่ควบคุมการทำงานของมอเตอร์ของแต่ละแกนให้สามารถทำงานได้สัมพันธ์และเป็นไปอย่างราบรื่นในช่วงเวลาเดียวกัน ซึ่งประกอบไปด้วยดังนี้

3.4.2.1 แหล่งจ่ายไฟให้กับ Servo Motor

ในงานวิจัยนี้ได้ใช้ Supply ชนิด Single Output Switching Power Supply แบบ 50W 1A ยี่ห้อ Meanwell รุ่น S-50-5 ซึ่งใช้เป็นแหล่งจ่ายไฟให้กับ Servo Motor ทั้ง 5 แกนของแขนกล

3.4.2.2 Microcontroller

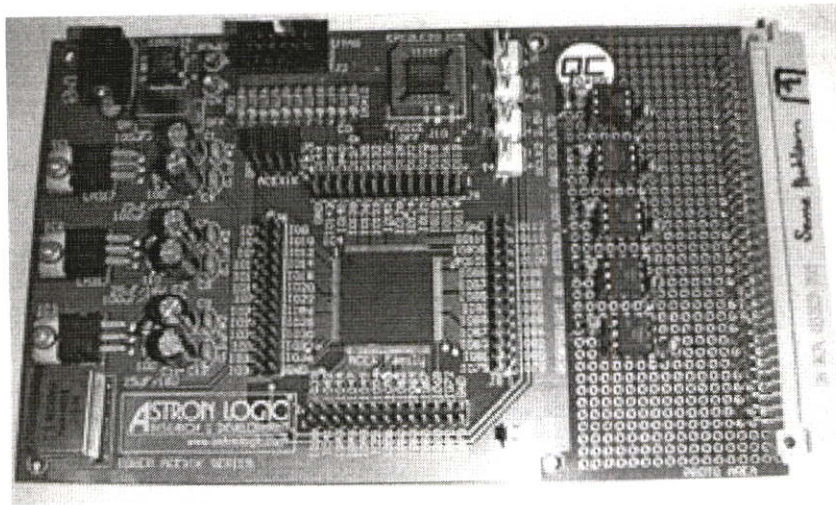
ในส่วนของ Microcontroller นี้จะทำหน้าที่ติดต่อกันระหว่าง Computer กับ Motion Controller ซึ่ง Microcontroller จะหน้าที่ในการควบคุมการ์ด FPGA



รูปที่ 3.5 แสดงการ์ด Microcontroller ที่ใช้ในงานวิจัย

3.4.2.3 การ์ด FPGA

ในโครงการนี้ได้เลือกใช้การ์ด FPGA ของ Astron Logic เพื่อทำการสร้างสัญญาณ Pulse เพื่อไปควบคุมตำแหน่งการทำงานของ DC Servo Motor



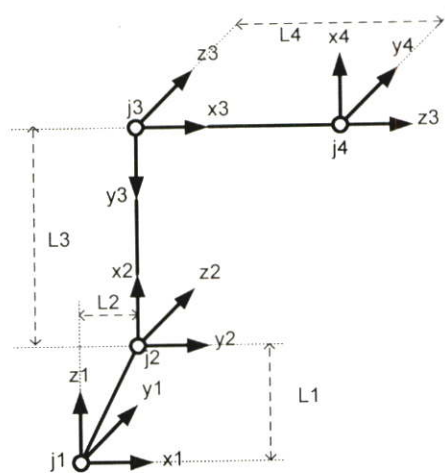
รูปที่ 3.6 แสดงการ์ด FPGA ของ Astron Logic

บทที่ 4

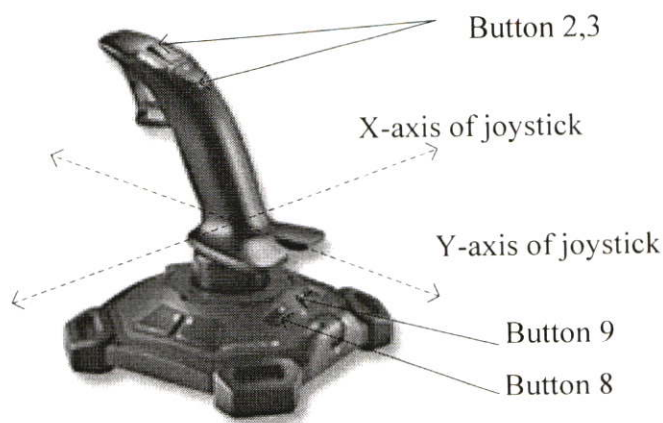
การควบคุมแบบ Real-time โดยใช้ Matlab

4.1 การควบคุมโดย Joystick

โดยการจำลองจาก M- file coded นั้นเป็นฟังก์ชันที่มีอยู่ใน Matlab สามารถถูกแสดงจาก พิกัด เครื่องมือที่เกี่ยวข้องของข้อต่อเฟรม 1 ถึงข้อต่อเฟรม 4 โดยแสดงถึงค่าพารามิเตอร์ต่างของแขนกล แผนภาพในรูปที่ 4.1 ในพิกัดอาศัยที่เกี่ยวข้องให้คำนวณผ่านหน่วยควบคุมของแขนหุ่นยนต์โดยการ ใช้ joystick เป็นตัวสั่งงาน โดยผ่านทางพอร์ตUSBที่ให้แผนภาพในรูปที่ 4.2



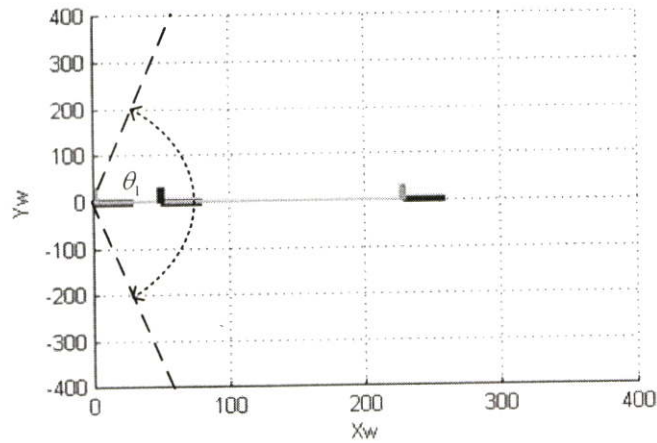
รูปที่ 4.1 แสดงลักษณะจุดเชื่อมต่อต่างๆของแขนกล



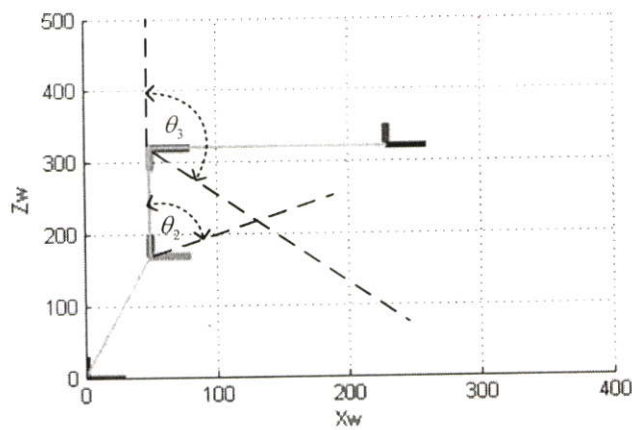
รูปที่ 4.2 แสดงการใช้ joystick ควบคุมแขนกล

4.2 การควบคุมโดยแบบ Forward Kinematics

เป็นฟังก์ชันดังต่อไปนี้ เส้นแนวนอนแกน X และแกน Y ในใช้การคำนวณทางคณิตศาสตร์ของ analog - 1 ถึง 1 ที่กึ่งกลางคือ ควบคุมการเคลื่อนที่ θ_1 and θ_2 ตามลำดับปุ่มและปุ่ม 2, 3 ควบคุม θ_3 ปุ่ม 8 ถูกใช้เพื่อให้แขนหุ่นยนต์กลับตำแหน่งที่เริ่มต้นตามที่แสดงในรูปที่ 4.3 โปรแกรมจะรันอย่างต่อเนื่องจนกระทั่งปุ่มกด 9 จึงจะหยุดโดยคำนวณจากสมการที่ (3.6), (3.7), (3.8) โดยการควบคุมแบบ Forward kinematics จะรับคำสั่งเป็น $\theta_1, \theta_2, \theta_3$



รูปที่ 4.3 แสดงการควบคุมของมุม θ_1

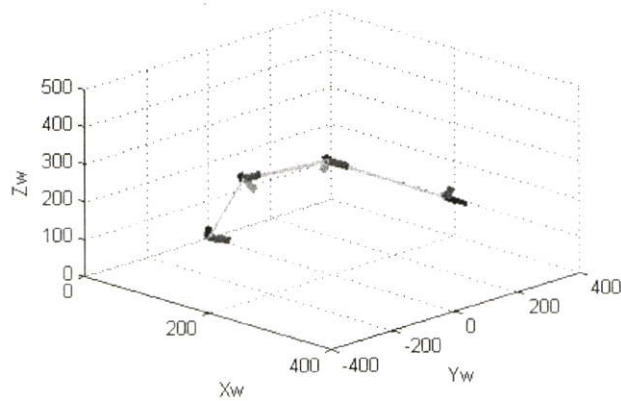


รูปที่ 4.4 แสดงการควบคุมของมุม θ_2, θ_3

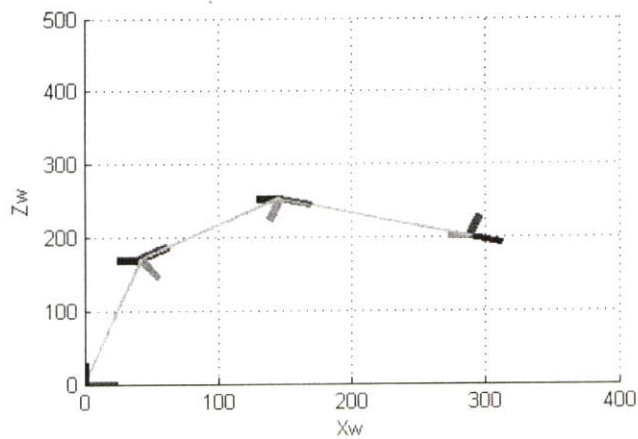
4.3 การควบคุมโดยแบบ Inverse Kinematics

โปรแกรมหน่วยควบคุมกลศาสตร์การเคลื่อนที่แบบ X, Y, Z coordinate เส้นแนวนอนแกน X และแกน Y ในใช้การคำนวณทางคณิตศาสตร์ ของ analog - 1 ถึง 1 ที่กึ่งกลางคือ ควบคุมการเคลื่อนที่

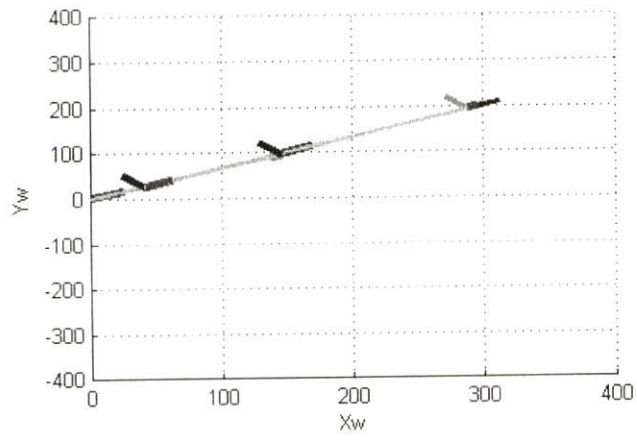
เส้นแนวแกน X และแกน Y ของ Robot ตามลำดับปุ่มและปุ่ม 2, 3 ควบคุมเส้นแนวแกน Z ปุ่ม 8 ถูกใช้เพื่อให้แขนหุ่นยนต์กลับตำแหน่งที่เริ่มต้น โปรแกรมจะรันอย่างต่อเนื่องจนกระทั่งปุ่มกด 9 จึงจะหยุดโดยคำนวณจากสมการที่ (3.9), (3.10), (3.11), (3.12) ถูกแสดงจากที่แผนภาพในรูปที่ 4.5 โดยการควบคุมแบบ Inverse kinematics จะรับคำสั่งเป็นตำแหน่ง X, Y, Z



รูปที่ 4.5 แสดงการควบคุมพิกัด X, Y, Z coordinate



รูปที่ 4.6 แสดงการควบคุมพิกัด X, Z coordinate



รูปที่ 4.7 แสดงการควบคุมพิกัด X, Y coordinate

กลศาสตร์การเคลื่อนที่จะถูกแสดงบนจอภาพคอมพิวเตอร์เพื่อที่อธิบายการเคลื่อนไหวในตำแหน่งของมัน โดยการใช้โปรแกรม Matlab มาช่วยในการคำนวณ

บทที่ 5

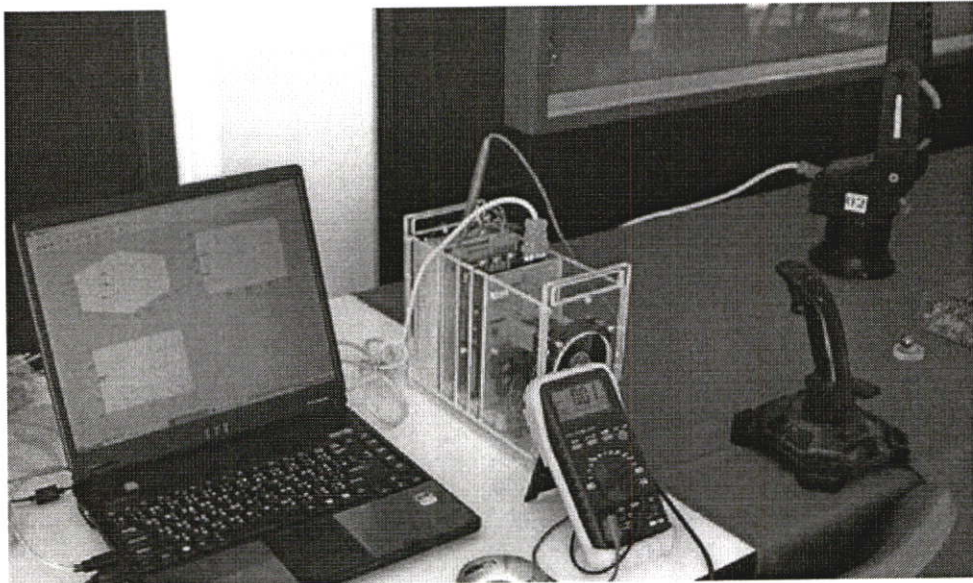
การทดลองและผลการทดลอง

5.1 บทนำ

การจำลองของเราประกอบด้วย 2 ส่วนคือ แบบตัวอย่าง โปรแกรมและหน่วยควบคุมการจำลอง การโดยใช้ Joystick และการคำนวณสมการกลศาสตร์การเคลื่อนที่โดยให้คำนวณพื้นที่การทำงาน เป็นไปได้ของแขนหุ่นยนต์โดยเรากำหนดมุมและทางเชื่อมดังต่อไปนี้ โดยกำหนดให้ $\theta_1 = -80^\circ$ to 80° , $\theta_2 = 0^\circ$ to 75° , $\theta_3 = -60^\circ$ to 90° , $L_1 = 80$ mm, $L_2 = 50$ mm, $L_3 = 150$ mm, and $L_4 = 180$ mm

ตามลำดับ เมื่อได้ทำการออกแบบสมการของการเคลื่อนที่ของแขนกลและการเขียนโปรแกรม เพื่อทำการประมวลผลจากคำสั่งที่รับเข้ามาเพื่อส่งผลลัพธ์ที่ได้ไปยังชุดควบคุมของแขนกล (Motion Controller) เพื่อทำการควบคุมการทำงานของ Servo Motor แต่ละตัวให้ทำงานสัมพันธ์กัน เพื่อให้เคลื่อนที่และทำงานของแต่ละข้อให้ได้ตามที่ต้องการ

5.2 อุปกรณ์และเครื่องมือที่ใช้ในการทดลอง



รูปที่ 5.1 แสดงการทำงานอุปกรณ์ที่ใช้ทั้งหมด

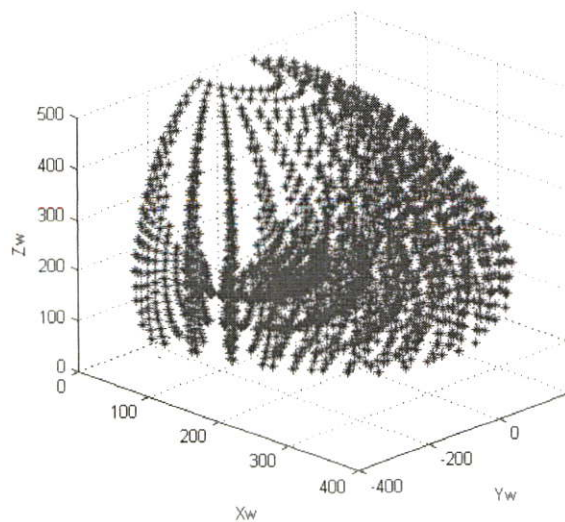
5.3 การทดลอง

การควบคุมการจำลองให้การเปรียบเทียบผลลัพธ์การใช้ forward and inverse kinematics การเคลื่อนที่กลับกันผ่านเครื่อง joystick สำหรับการเคลื่อนที่ forward kinematics เราควบคุมมุมของข้อต่อแต่ละอันที่จะได้ตำแหน่ง ขณะที่เราควบคุม inverse kinematics ที่จะได้มุมของข้อต่อแต่ละอันโดยคำนวณย้อนกลับ

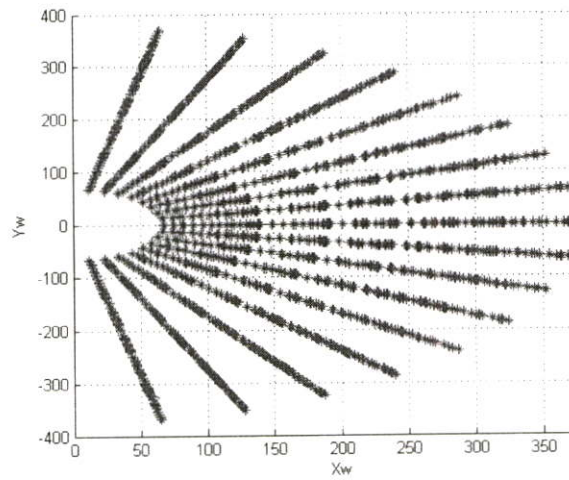
ตารางที่ 5.1 แสดงการควบคุมเปรียบเทียบผลลัพธ์การใช้ forward and inverse kinematics

Position	Forward kinematics	Inverse kinematics
θ_1	32.9294°	33.4732°
θ_2	57.0846°	55.7698°
θ_3	40.3109°	38.1860°
P_x	292.3109 mm	288.2893 mm
P_y	189.3137 mm	190.6140 mm
P_z	199.5633 mm	200.0000 mm

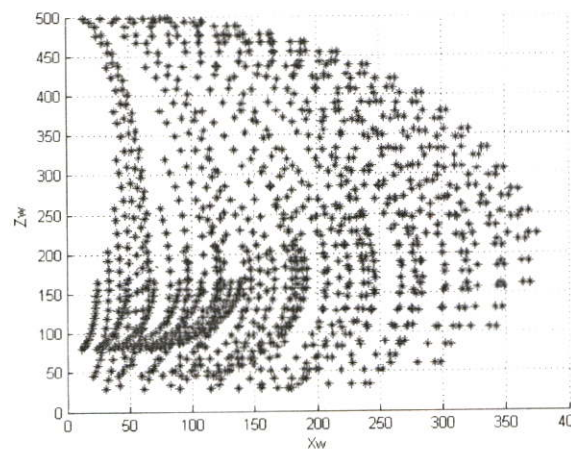
การคำนวณเพื่อตัดสินใจว่าพื้นที่การทำงานของแขนกลสามารถถูกคำนวณการโดยใช้วิธี Forward kinematics การเคลื่อนที่จากตำแหน่งและการเจาะจงทางเดินดังในรูปที่ 5.6 เครื่องหมาย (*) แสดงเพราะเส้นโคจรให้ความสามารถไปยังตำแหน่งที่ต้องการบนพื้นที่การทำงานของแขนกลที่สามารถทำได้จากผลลัพธ์การจำลองถูกกำหนดเนื่องจากพื้นที่การทำงานจริงโดยกำหนดให้ $\theta_1 = -80^\circ$ to 80° , $\theta_2 = 0^\circ$ to 75° , $\theta_3 = -60^\circ$ to 90° , $L_1 = 80$ mm, $L_2 = 50$ mm, $L_3 = 150$ mm, and $L_4 = 180$ mm



รูปที่ 5.2 Trajectory of a Robot arm

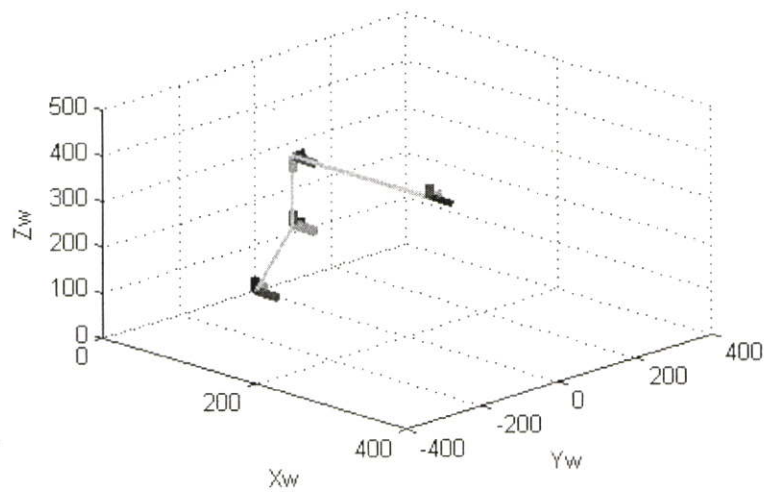


รูปที่ 5.3 Working area of a Robot arm in XY plane



รูปที่ 5.4 Working area of a Robot arm in XZ plane

จากการทดลองได้ทำการทดลองโดยควบคุมการทำงานของแขนกลโดยผ่านทางโปรแกรม Matlab ที่ได้เขียนขึ้นมา ผลปรากฏว่าการเคลื่อนที่ของแขนกลได้มีความสัมพันธ์กับกราฟที่แสดงขึ้นมาบนจอภาพได้อย่างถูกต้อง



รูปที่ 5.5 แสดงผลของกราฟิกของแขนกลที่แสดงออกทางจอภาพ

จากนั้นได้ทดลองการนำแขนกลไปหยิบจับวัตถุ เพื่อทดลองการ Take Load แขนกลว่าสามารถที่จะหยิบจับวัตถุต่างๆ ได้หรือไม่ จากการทดสอบโดยการทดสอบการหยิบลูกเหล็กที่มีน้ำหนักอยู่ที่ 300 กรัม ซึ่งผลปรากฏว่าสามารถหยิบจับลูกเหล็กได้อย่างสมบูรณ์



รูปที่ 5.6 แสดงการทำงานของแขนกลเมื่อทำการหยิบลูกเหล็ก

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

เทคนิควิธีที่นำมาใช้นี้คือนำแบบตัวอย่างและการจำลองการเคลื่อนที่ของแขนหุ่นยนต์โดยอาศัยหลักการคำนวณทางคณิตศาสตร์ควบคุมกลศาสตร์การเคลื่อนที่แบบไปข้างหน้าและแบบย้อนกลับ แสดงถึงการจัดเตรียมการวิเคราะห์และเจาะจงพื้นที่การทำงานก่อนการรับเส้นทางเดินที่มีความปลอดภัยของพื้นที่การทำงานจริงก่อนการส่งคำสั่งจริงไปยังของแขนหุ่นยนต์โดยมีการเปรียบเทียบผลลัพธ์การใช้ forward and inverse kinematics การเคลื่อนที่กลับกันผ่านเครื่อง joystick สำหรับการเคลื่อนที่ forward kinematics เราควบคุมมุมของข้อต่อแต่ละอันที่จะได้ตำแหน่งขณะที่เราทำควบคุม inverse kinematics ที่จะได้มุมของข้อต่อแต่ละอันโดยคำนวณย้อนกลับการคำนวณเพื่อตัดสินใจว่าพื้นที่การทำงานของแขนกลสามารถถูกคำนวณการจะแสดงผลผ่านกราฟิกเป็นเส้นโคจรให้สามารถไปยังตำแหน่งที่ต้องการบนพื้นที่การทำงานของแขนกลที่สามารถทำได้จากผลลัพธ์การจำลองถูกกำหนดเนื่องจากพื้นที่การทำงานจริงหลักการที่นำมาใช้ในแบบจำลองทางคณิตศาสตร์ได้อย่างง่ายดาย โดยนำแบบจำลองคณิตศาสตร์ที่ได้นั้นมาใช้ในการกำหนดเส้นทางของการเคลื่อนที่ของแขนกลให้อยู่ในเส้นทางตามที่ต้องการได้ โดยได้นำโปรแกรมแมทแลป(MATLAB) มาช่วยในการจำลองและประมวลผล

6.2 ข้อเสนอแนะและแนวทางการพัฒนา

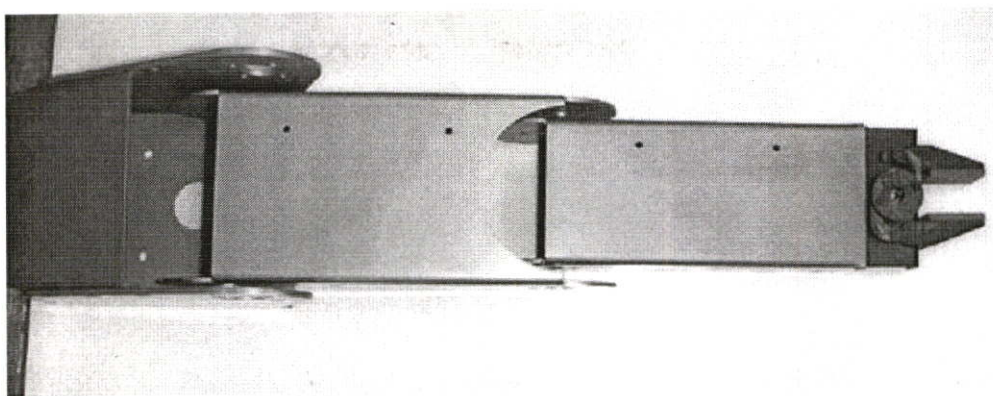
จากการทดลองจะสังเกตได้ว่า การสั่งการให้ตัวหุ่นยนต์เคลื่อนที่ไปตามที่กำหนด โดยป้อนตำแหน่งผ่าน joystick นั้นก็เกิดค่า Error อยู่เนื่องจาก joystick ที่นำมาทดลองไม่ใช่แบบอุตสาหกรรมซึ่งถ้า joystick ที่ดีก็จะทำให้เกิดข้อผิดพลาดซึ่งจะง่ายต่อการควบคุมและ สังเกตได้ว่า ในแต่ละข้อต่อของหุ่นยนต์ต้นแบบที่อิงสาขาของข้อต่อเป็นตัวอาร์โพเทนทิโอเมเตอร์ที่เป็นตัวตรวจวัดองศาการเคลื่อนที่ก็จะเปลี่ยนแปลงค่าไป ทำให้ค่าที่นำไปเปรียบเทียบกับค่า Set point มีค่า Error เกิดขึ้นแล้วส่งค่า Error ไปให้ตัวควบคุมทำการประมวลผล ดังนั้นเทคนิควิธีที่นำมาใช้ควบคุมทางเดินจึงทำให้หุ่นยนต์แขนกลมีการเคลื่อนที่ในช่วงของค่าผิดพลาดที่ถูกกำหนดไว้ได้

บรรณานุกรม

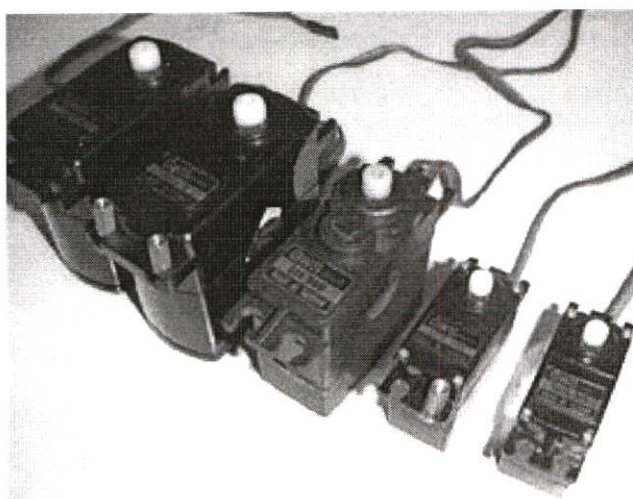
- [1] Marc Bodson. 2003. "Fun Control Experiments with Matlab and a joystick". In **the IEEE Conference on Decision and Control**: Maui, HI,
- [2] Navinda Kottege & D.U.J. Sonnadara. 2004. **Virtual robot arm control model**. Sri Lanka : Institute of Physics
- [3] Saeed B. Niku. 2001. **Introduction to Robotics Analysis Systems Applications**. Upper Saddle River : Prentice Hall
- [4] P.G. Ranky C.Y.Ho. 1985. **Robot Modelling control and applications with software** Great Britain : Academic Press.
- [5] Jettiya Sripituk. 2007 "A Simple Technique to Modeling and Simulation Four-Axe Robot-Arm Control" 2180-2184. in **International Conference on Control, Automation and Systems 2007** : Seoul, Korea

ภาคผนวก

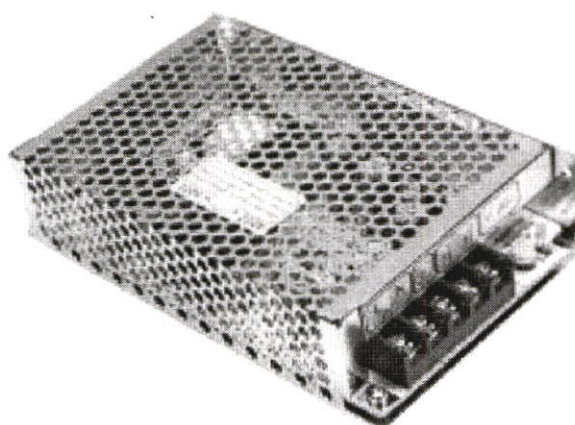
ภาคผนวก ก.
รายละเอียดของอุปกรณ์ที่ใช้ในการทดลอง



รูปที่ ก.1 แสดงส่วน โครงสร้างของแขนกล



รูปที่ ก.2 แสดง DC Servo Motor ของแต่ละแกน



รูปที่ ก.3 แสดง Power Supply ของ Servo Motor

ภาคผนวก ข.

A Simple Technique to Modeling and Simulation Four-Axe Robot-Arm Control

Jettiya Sripituk¹, Teerawat Thepmanee², and Prapart Ukakimapurn³

^{1,2,3}Department of Instrumentation Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand

¹ (Tel :+ 66-2-739-2406; Email : jettiya_y@yahoo.co.uk)

² (Tel :+ 66-2-739-2406; Email :ktteeraw@kmitl.ac.th)

³ (Tel :+ 66-2-739-2406; Email :kuprapart@kmitl.ac.th)

Abstract: This paper presents modeling and real-time simulation programs that were developed for robot arm control. The technique analyzes the safety path, working area and step motion of robot. The implementation of experimental forward and inverse kinematics control method as Matlab m-files is described. The visualization and animation capabilities of Matlab provide a realistic perception of behavior of the test beds without robot or before send the command to robot. Control mode in manual and automatic by a joystick and library file via PC respectively.

Keywords: Modeling and Simulation, Robot-Arm Control

1. Introduction

Robot arm modeling is very important to analyze and specify the working area in order to obtain the safety path before developing a real of robot arm. Although many robot arm programs are available mostly for industrial application which is difficult to modify. Thus, the development of program robot engineering using MATLAB is a good solution, since it is easy and fast to modify codes and experiment with a variety of controllers [1]. The benefit of robot arm modeling is to study the design and motion control of robot arm by forward and inverse kinematics. Moreover, it could help to develop software for industrial work controlling such as automation assembly, welding etc.

2. Robot arm design and kinematics analysis

Since the robot arm is a chain of rigid links, the coordinates of the mobile end can be mapped to the coordinates of the fixed base using the techniques discussed earlier.

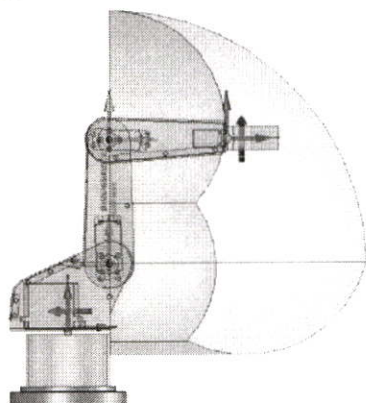


Figure1 KMITL Robot

However, in order to do this process, coordinate frames need to be assigned to each of the links, defining kinematic parameters associated with each of the links and joints make the process much easier.

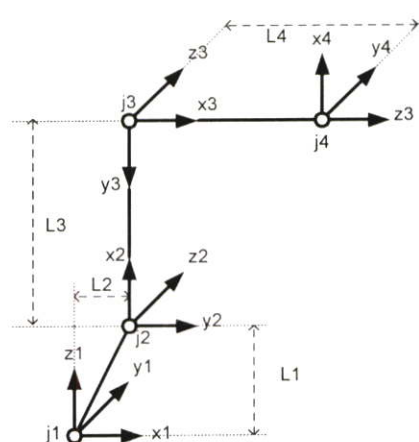


Figure2 Model link coordinates diagram design

Once, it has been decided that the robot arm contains 4 joints containing 4 axes, in order to determine its kinematic parameters, the D-H representation and link coordinate diagram are used. It should be noted that the link coordinate diagram is not unique for a given robotic manipulator. Depending on the initial orientation of the link, the D-H representation will assign different joint angles. Following link coordinate diagrams illustrate in Figure 1 with the associated kinematics parameters of robot arm.

2.1 Kinematics of the Robot Manipulator

The matrix for the robot arm would be the composite homogeneous transformation matrix, which maps the tool coordinates relative to the frame of joint no. 4 into the base coordinates relative to the frame of joint no. 1, the individual homogeneous transformation matrices for robot arm were constructed as follows[2].

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_1^2 = \begin{bmatrix} \sin(\theta_2) & \cos(\theta_2) & 0 & L_2 \\ 0 & 0 & 1 & 0 \\ \cos(\theta_2) & -\sin(\theta_2) & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_2^3 = \begin{bmatrix} \sin(\theta_3) & -\cos(\theta_3) & 0 & L_3 \\ \cos(\theta_3) & \sin(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_3^4 = \begin{bmatrix} 0 & 0 & 0 & L_4 \\ -\cos(\theta_4) & \sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The direct kinematics problem can be defined as determining positions and orientations of the tool with respect to the coordinate frame, which is attached to the robot base and given the vector of joint variables of the robotic manipulator.

Almost all techniques need to solve the above problem. What needs to be done is to construct the arm matrix, which leads to the arm equation. The solution to arm equation yields the solution to direct kinematics problem. To construct the arm matrix, it is necessary to determine all kinematics parameters associated with each of the links. This determination can do by drawing a link coordinate diagram and using a technique of systematic notation proposed by Denavit and Hartenberg [3-4] for assigning right handed orthonormal coordinate frame, which is one to each link in an open kinematic chain of link.

For solving the forward kinematics problem of robot arm, the arm equation can be stated as in Eq. (5) to obtain the tool tip position $P = [Px, Py, Pz]$, where

Px , Py , and Pz are elements of [1,4], [2,4], and [3,4] of the arm matrix, respectively.

$$T_{base}^{tool} = T_0^1 T_1^2 T_2^3 T_3^4 \quad (5)$$

Once, these elements have been simplified using basic trigonometric identities, they can be written as follows.

$$Px = (c_1 s_2 s_3 + c_1 c_2 c_3) L_4 + L_3 c_1 s_2 + L_2 c_1 \quad (6)$$

$$Py = (s_1 s_2 s_3 + s_1 s_2 c_3) L_4 + L_3 s_1 s_2 + L_2 s_2 \quad (7)$$

$$Pz = (c_2 s_3 - c_3 s_2) L_4 + L_3 c_2 + L_1 \quad (8)$$

2.2 Solution of the Inverse Kinematics Problem

The direct kinematics equation, establishes the functional relationship between the joint variables and the end-effector position and orientation. The inverse kinematics problem consists of the determination of the joint variables corresponding to a given end-effector position and orientation. The solution to this problem is of fundamental importance in order to transform the motion specifications, assigned to the end-effector in the operational space, into the corresponding joint space motions that allow execution of the desired motion.

Where Px, Py , and Pz are the tool positions, and c_1, c_2, c_3, s_1, s_2 , and s_3 are $\cos(\theta_1), \cos(\theta_2), \cos(\theta_3), \sin(\theta_1), \sin(\theta_2)$, and $\sin(\theta_3)$, respectively. Solving inverse kinematic problem of robot arm to obtain the angle of each joint as the following expressions.

$$\theta_1 = A \tan 2(p_y, p_x) \quad (9)$$

Once θ_1 is known, the resulting structure is planar with regard to the variables θ_2, θ_3 and θ_4 .

Hence,

$$\theta_2 = A \tan 2(-s_2, c_2) \quad (10)$$

When

$$s_2 = (Pz - L_1) c_3 L_4 + \left(\frac{Px}{c_1} - L_2 \right) (s_3 L_4 + L_3)$$

$$c_2 = \left(\frac{Px}{c_1} - L_2 \right) c_3 L_4 + (Pz - L_1) (s_3 L_4 + L_3)$$

And

$$\theta_3 = \sin^{-1} \left[\frac{\left(\frac{p_x - L_1}{c_1} \right)^2 + (p_z - L_2)^2 - L_3^2 - L_4^2}{2L_3L_4} \right] \quad (11)$$

where to a given end-effector position p_w

$$\theta_4 = A \tan 2(p_{w_x}, p_{w_y}) \quad (12)$$

Three above expressions could provide the robot arm for all values of the joint vector. In this paper, the angles were set as $\theta_1 = -80^\circ$ to 80° , $\theta_2 = 0^\circ$ to 75° , and $\theta_3 = -60^\circ$ to 90° . The defined distances of link $L_1 = 80$ mm, $L_2 = 50$ mm, $L_3 = 150$ mm and $L_4 = 180$ mm were chosen.

3. Visualization and real-time control using Matlab

The simulations are coded as m-file in Matlab by taking advantage of simple, yet powerful Matlab functions of drawing and animation. As it can be seen from the diagram given in Figure 1, the tool coordinates relative to the frame joint 4 into base coordinates relative to frame joint 1 have the position and orientation as shown in Figure 3.

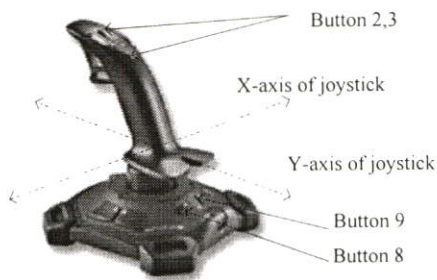


Figure 3 Controlling a simulated robot arm through joystick

The control of robot arm using joystick as shown in Figure 2 via USB port contains of the following functions. X-axis and Y-axis provide analog -1 to 1. The middle of stick is 0. Buttons 2,3 are used to control up and down. Button 8 is used to reset in order to let the robot arm return to started position as shown in Figure 3. The program will run continuously until pressing button 9 to stop.

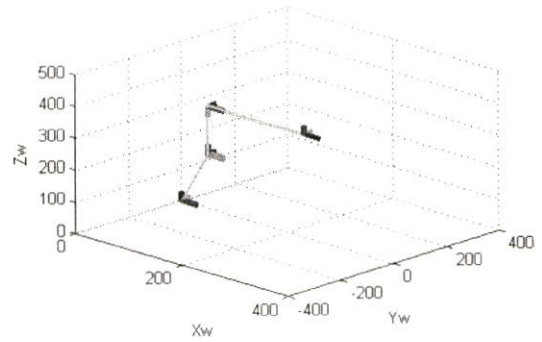


Figure 4 Setup robot arm design on Matlab

The forward kinematics control program receives the input from joystick. θ_1 and θ_2 are moved on X-axis and Y-axis of joystick, respectively. To control θ_3 uses buttons 2,3 to move up or down. The position of the three angles ($\theta_1, \theta_2, \theta_3$) as shown in Figures 4 and 5 are calculated by forward kinematics functions as shown in Eqs. (6)-(8).

Thus, the forward kinematics can be applied to control the moving angles of each joint in order to move the end of robot arm according to our control.

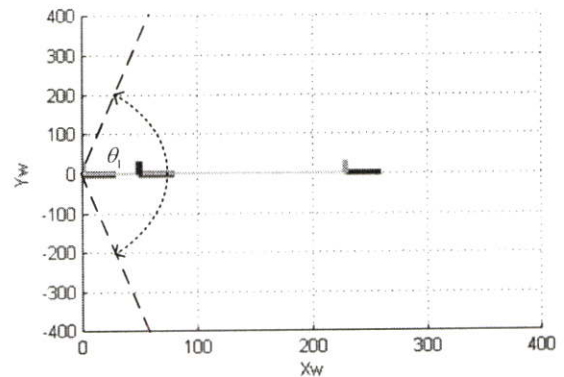


Figure 5 Range of Angle θ_1

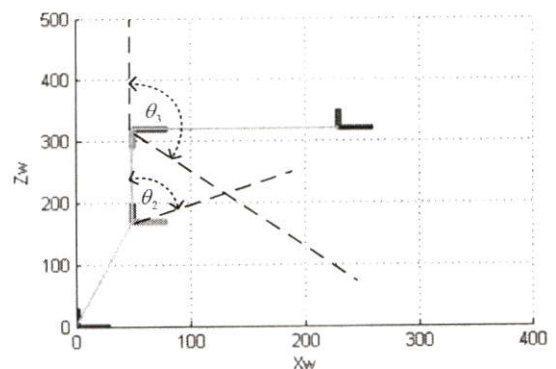


Figure 6 Ranges of Angles θ_2 and θ_3

The inverse kinematics program is used to control the tool coordinate. It is very easy by pushing the joystick by X-axis and Y-axis. While the buttons 2,3 controls Z-axis.

There are many reasons that make this method difficulty to control. For example, the movements are over from the calculated ranges of $(\theta_1, \theta_2, \theta_3)$ or the link of each arm is not long enough to move as shown in simulation results. Therefore, pulling the joystick uses to reset the robot or when we press the button 8, the robot arm will return to the start position as shown in Figure 3. The program is running even if no pressing button 9.

Unlike mechanical implementation for a robotic manipulator, when it is simulated, most of the design issues such as joint friction, motor drive capacity, torque, or link weight can be ignored. However, a different setting of problem arises when it is attempted to depict the robot manipulator and its motion on computer screen using a Matlab program.

4. Simulation results

Our simulation consists of two sections; the program modeling and simulation control using joystick with forward and inverse kinematics methods to determine the possible working area of robot arm. To design the modeling, we assigned the angles and links as follows; $\theta_1 = -80^\circ$ to 80° , $\theta_2 = 0^\circ$ to 75° , $\theta_3 = -60^\circ$ to 90° , $L_1 = 80$ mm, $L_2 = 50$ mm, $L_3 = 150$ mm, and $L_4 = 180$ mm, respectively.

Table 1 shows the comparison between the controlled simulation results using the forward and inverse kinematics methods through a joystick. For forward kinematics method, we control the angles of each joint to get the position, while we control the end of tool to get the angles of each joint using inverse kinematics. The positions and orientations of simulated robot arm in XYZ, XZ, and XY planes are shown in Figures 6-8, respectively.

Table 1. Forward and inverse kinematic controlling

Position	Forward kinematics	Inverse kinematics
θ_1	32.9294°	33.4732°
θ_2	57.0846°	55.7698°
θ_3	40.3109°	38.1860°
P_x	292.3109 mm	288.2893 mm
P_y	189.3137 mm	190.6140 mm
P_z	199.5633 mm	200.0000 mm

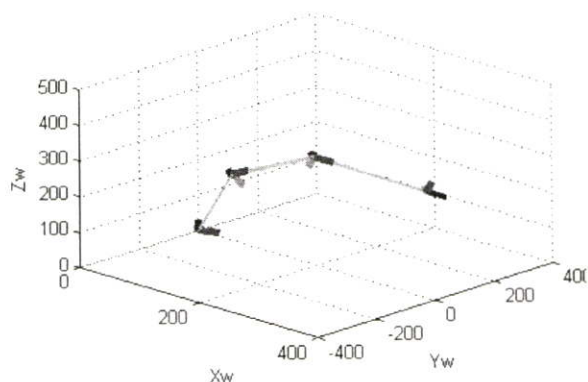


Figure 7 Simulated robot arm in XYZ Plane

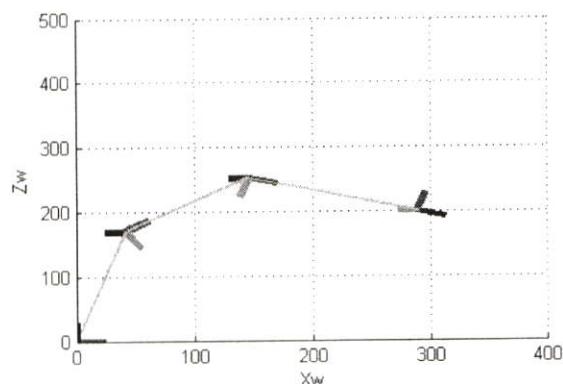


Figure 8 Simulated robot arm in XZ Plane

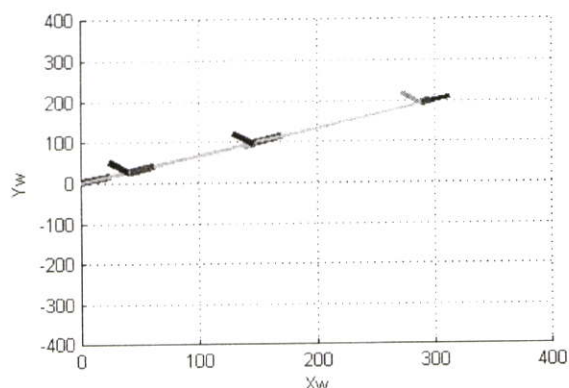


Figure 9 Simulated robot arm in XY Plane

To determine the working area, it can be calculated using forward kinematics method from the angle and link specifications as illustrated in Figures 9-11. The sign (*) shows the capability of trajectory of the end of tool from the desired angle and link specifications. The working area from simulation results is assigned as the safety path of a real working area and the orientation for inverse kinematics control. Before the real robot-arm implementation, if the change of working area is needed, the adjustments of angle and link will be

applied. (The design and real robot should be considered)

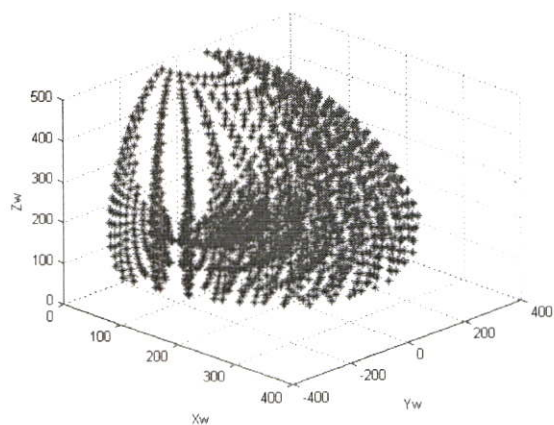


Figure 10 Trajectory of a Robot arm

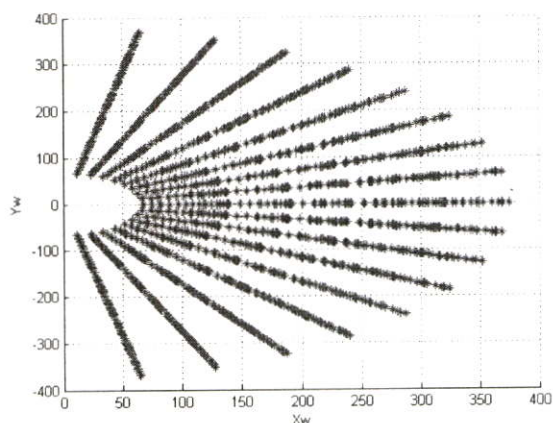


Figure 11 Working area of a Robot arm in XY plane

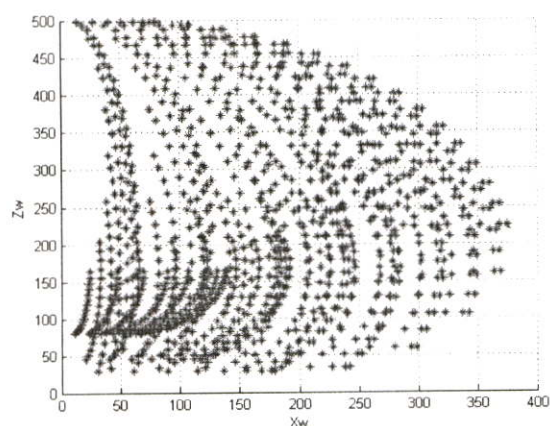


Figure 12 Working area of a Robot arm in XZ plane

5. Conclusion

The modeling and simulation of robot arm using the forward and inverse kinematics control has been presented in this paper. The proposed technique provides the analyzed and specified working area in order to obtain the safety path of a real working area before developing a prototype of robot arm.



Figure 13 Robot arm pick up a ball

(Now, the prototype has been developed and submitted by King Mongkut's Institute of Technology Ladkrabang and has been shown in the KMITL Exhibition 2006.)

References

- [1] Marc Bodson "Fun Control Experiments with Matlab and a joystick", Proc. of the IEEE Conference on Decision and Control, Maui, HI, Dec. 2003.
- [2] Navinda Kottege & D.U.J. Sonnadara, *Virtual robot arm control model*, Institute of Physics, Proc. 20th Technical Session, Sri Lanka, March 2004.
- [3] Saeed B. Niku, *Introduction to Robotics Analysis, Systems, Applications*, Prentice Hall, Upper Saddle River, NJ, 2001.
- [4] P.G. Ranky, C.Y.Ho, *Robot Modelling control and applications with software*, Springer-Verlag, 1985.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นาย เจตติยะ ศรีพิทักษ์
วัน เดือน ปีเกิด	4 สิงหาคม 2521 ที่จังหวัดสุพรรณบุรี
ที่อยู่	8/54 ลาซาลปาร์ค สุขุมวิท 105 แขวงบางนา เขตบางนา กทม. 10260 E-mail: Jettiya_y@hotmail.com
ประวัติการศึกษา	2545 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประวัติการทำงาน	
พ.ศ.2546	เป็นตัวแทนของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง นำเสนอผลงานด้าน ROBOT ผ่านรายการที่ประเทศไทย ทางช่อง 5
พ.ศ.2546-2548	ผู้ช่วยอาจารย์ภาควิชาวิศวกรรมการวัดคุม และผู้ช่วยอาจารย์วิชากราฟฟิคดีไซน์ ภาควิชาสถาปัตยกรรมศาสตร์ คณะครุศาสตร์อุตสาหกรรม
พ.ศ.2548-2549	วิศวกรออกแบบด้าน INDUSTRIAL ROBOT บ.บางกอกอโต้เมชันเอ็นจิเนียริง จำกัด
พ.ศ.2549-ปัจจุบัน	ผู้จัดการโครงการฝ่ายแมคคาทรอนิกส์ บ.ไทยเซ็นทรัลแมคคานิกส์ จำกัด
ผลงานวิจัย	
พ.ศ.2550	นำเสนอผลงานวิจัยเรื่อง A SIMPLE TECHNIQUE TO MODELING AND SIMULATION FOUR-AXIS ROBOT-ARM CONTROL , ณ การ ประชุมวิชาการนานาชาติ ICCAS 2007 ประเทศเกาหลี

- Welcome Message
- Conference Organization
- Conference Information
- Plenary Talks, Invited Talks & Special Session
- Table of Contents
- Author Index
- Financial Contribution
- E-proceeding Search
- EXIT



October 17-20, 2007
COEX, Seoul, Korea

IEEE Catalog Number: 07EX1874 ISBN: 978-89-950038-6-2-98560
in conjunction with KRC 2007



A Simple Technique to Modeling and Simulation Four-Axe Robot-Arm Control

Teerawat Thepmanee, Jettiya Sripituk, and Prapart Ukakimapurn

Department of Instrumentation Engineering , King Mongkut's Institute of Technology Ladkrabang, Thailand

¹ (Tel :+ 66-2-739-2406; Email : jettiya_y@yahoo.co.uk)

² (Tel :+ 66-2-739-2406; Email :ktteeraw@kmitl.ac.th)

³ (Tel :+ 66-2-739-2406; Email :kuprapart@kmitl.ac.th)

Abstract: This paper presents modeling and real-time simulation programs that were developed for robot arm control. The technique analyzes the safety path, working area and step motion of robot. The implementation of experimental forward and inverse kinematics control method as Matlab m-files is described. The visualization and animation capabilities of Matlab provide a realistic perception of behavior of the test beds without robot or before send the command to robot. Control mode in manual and automatic by a joystick and library file via PC respectively.

Keywords: Modeling and Simulation , Robot-Arm Control

1. Introduction

Robot arm modeling is very important to analyze and specify the working area in order to obtain the safety path before developing a real of robot arm. Although many robot arm programs are available mostly for industrial application which is difficult to modify. Thus, the development of program robot engineering using MATLAB is a good solution, since it is easy and fast to modify codes and experiment with a variety of controllers [1]. The benefit of robot arm modeling is to study the design and motion control of robot arm by forward and inverse kinematics. Moreover, it could help to develop software for industrial work controlling such as automation assembly, welding etc.

2. Robot arm design and kinematics analysis

Since the robot arm is a chain of rigid links, the coordinates of the mobile end can be mapped to the coordinates of the fixed base using the techniques discussed earlier.

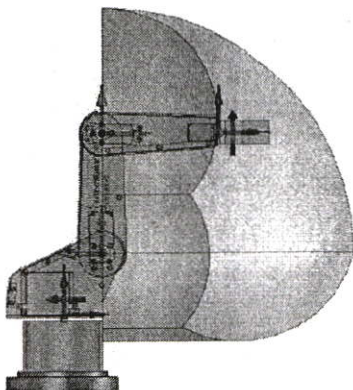


Figure1 KMITL Robot

However, in order to do this process, coordinate frames need to be assigned to each of the links, defining kinematic parameters associated with each of the links and joints make the process much easier.

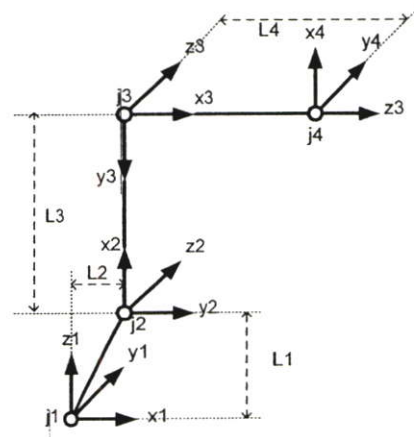


Figure2 Model link coordinates diagram design

Once, it has been decided that the robot arm contains 4 joints containing 4 axes, in order to determine its kinematic parameters, the D-H representation and link coordinate diagram are used. It should be noted that the link coordinate diagram is not unique for a given robotic manipulator. Depending on the initial orientation of the link, the D-H representation will assign different joint angles. Following link coordinate diagrams illustrate in Figure 1 with the associated kinematics parameters of robot arm.

2.1 Kinematics of the Robot Manipulator

The matrix for the robot arm would be the composite homogeneous transformation matrix, which maps the tool coordinates relative to the frame of joint no. 4 into the base coordinates relative to the frame of joint no. 1, the individual homogeneous transformation matrices for robot arm were constructed as follows[2].

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_1^2 = \begin{bmatrix} \sin(\theta_2) & \cos(\theta_2) & 0 & L_2 \\ 0 & 0 & 1 & 0 \\ \cos(\theta_2) & -\sin(\theta_2) & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_2^3 = \begin{bmatrix} \sin(\theta_3) & -\cos(\theta_3) & 0 & L_3 \\ \cos(\theta_3) & \sin(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_3^4 = \begin{bmatrix} 0 & 0 & 0 & L_4 \\ -\cos(\theta_4) & \sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The direct kinematics problem can be defined as determining positions and orientations of the tool with respect to the coordinate frame, which is attached to the robot base and given the vector of joint variables of the robotic manipulator.

Almost all techniques need to solve the above problem. What needs to be done is to construct the arm matrix, which leads to the arm equation. The solution to arm equation yields the solution to direct kinematics problem. To construct the arm matrix, it is necessary to determine all kinematics parameters associated with each of the links. This determination can do by drawing a link coordinate diagram and using a technique of systematic notation proposed by Denavit and Hartenberg [3-4] for assigning right handed orthonormal coordinate frame, which is one to each link in an open kinematic chain of link.

For solving the forward kinematics problem of robot arm, the arm equation can be stated as in Eq. (5) to obtain the tool tip position $P = [Px, Py, Pz]$, where

$Px, Py,$ and Pz are elements of [1,4], [2,4], and [3,4] of the arm matrix, respectively.

$$T_{base}^{tool} = T_0^1 T_1^2 T_2^3 T_3^4 \quad (5)$$

Once, these elements have been simplified using basic trigonometric identities, they can be written as follows.

$$Px = (c_1 s_2 s_3 + c_1 c_2 c_3) L_4 + L_3 c_1 s_2 + L_2 c_1 \quad (6)$$

$$Py = (s_1 s_2 s_3 + s_1 s_2 c_3) L_4 + L_3 s_1 s_2 + L_2 s_2 \quad (7)$$

$$Pz = (c_2 s_3 - c_3 s_2) L_4 + L_3 c_2 + L_1 \quad (8)$$

2.2 Solution of the Inverse Kinematics Problem

The direct kinematics equation, establishes the functional relationship between the joint variables and the end-effector position and orientation. The inverse kinematics problem consists of the determination of the joint variables corresponding to a given end-effector position and orientation. The solution to this problem is of fundamental importance in order to transform the motion specifications, assigned to the end-effector in the operational space, into the corresponding joint space motions that allow execution of the desired motion.

Where $Px, Py,$ and Pz are the tool positions, and $c_1, c_2, c_3, s_1, s_2,$ and s_3 are $\cos(\theta_1), \cos(\theta_2), \cos(\theta_3), \sin(\theta_1), \sin(\theta_2),$ and $\sin(\theta_3),$ respectively. Solving inverse kinematic problem of robot arm to obtain the angle of each joint as the following expressions.

$$\theta_1 = A \tan 2(p_y, p_x) \quad (9)$$

Once θ_1 is known, the resulting structure is planar with regard to the variables θ_2, θ_3 and $\theta_4.$

Hence,

$$\theta_2 = A \tan 2(-s_2, c_2) \quad (10)$$

When

$$s_2 = (Pz - L_1) c_3 L_4 + \left(\frac{Px}{c_1} - L_2 \right) (s_3 L_4 + L_3)$$

$$c_2 = \left(\frac{Px}{c_1} - L_2 \right) c_3 L_4 + (Pz - L_1) (s_3 L_4 + L_3)$$

And

$$\theta_3 = \sin^{-1} \left[\frac{\left(\frac{p_x - L_1}{c_1} \right)^2 + (p_z - L_2)^2 - L_3^2 - L_4^2}{2L_3L_4} \right] \quad (11)$$

where to a given end-effector position p_w

$$\theta_4 = A \tan 2(p_{w_y}, p_{w_x}) \quad (12)$$

Three above expressions could provide the robot arm for all values of the joint vector. In this paper, the angles were set as $\theta_1 = -80^\circ$ to 80° , $\theta_2 = 0^\circ$ to 75° , and $\theta_3 = -60^\circ$ to 90° . The defined distances of link $L_1 = 80$ mm, $L_2 = 50$ mm, $L_3 = 150$ mm and $L_4 = 180$ mm were chosen.

3. Visualization and real-time control using Matlab

The simulations are coded as m-file in Matlab by taking advantage of simple, yet powerful Matlab functions of drawing and animation. As it can be seen from the diagram given in Figure 1, the tool coordinates relative to the frame joint 4 into base coordinates relative to frame joint 1 have the position and orientation as shown in Figure 3.

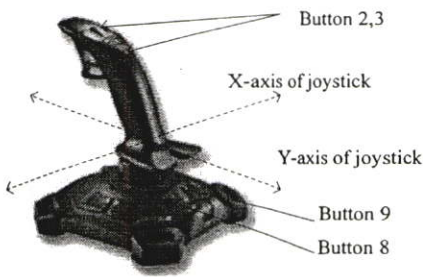


Figure 3 Controlling a simulated robot arm through joystick

The control of robot arm using joystick as shown in Figure 2 via USB port contains of the following functions. X-axis and Y-axis provide analog -1 to 1. The middle of stick is 0. Buttons 2,3 are used to control up and down. Button 8 is used to reset in order to let the robot arm return to started position as shown in Figure 3. The program will run continuously until pressing button 9 to stop.

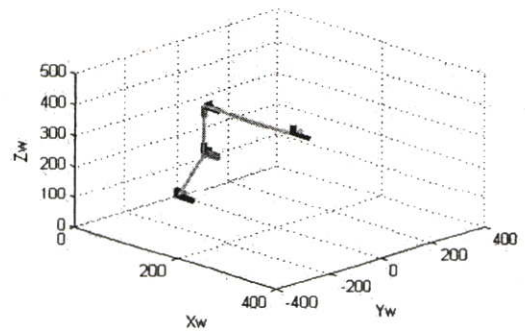


Figure 4 Setup robot arm design on Matlab

The forward kinematics control program receives the input from joystick. θ_1 and θ_2 are moved on X-axis and Y-axis of joystick, respectively. To control θ_3 uses buttons 2,3 to move up or down. The position of the three angles ($\theta_1, \theta_2, \theta_3$) as shown in Figures 4 and 5 are calculated by forward kinematics functions as shown in Eqs. (6)-(8).

Thus, the forward kinematics can be applied to control the moving angles of each joint in order to move the end of robot arm according to our control.

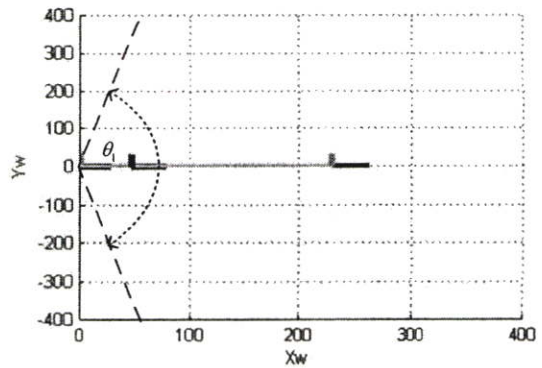


Figure 5 Range of Angle θ_1

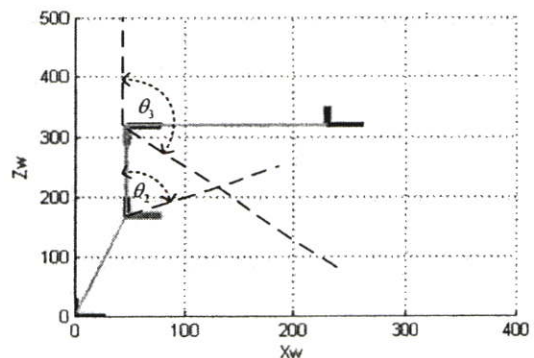


Figure 6 Ranges of Angles θ_2 and θ_3

The inverse kinematics program is used to control the tool coordinate. It is very easy by pushing the joystick by X-axis and Y-axis. While the buttons 2,3 controls Z-axis.

There are many reasons that make this method difficulty to control. For example, the movements are over from the calculated ranges of $(\theta_1, \theta_2, \theta_3)$ or the link of each arm is not long enough to move as shown in simulation results. Therefore, pulling the joystick uses to reset the robot or when we press the button 8, the robot arm will return to the start position as shown in Figure 3. The program is running even if no pressing button 9.

Unlike mechanical implementation for a robotic manipulator, when it is simulated, most of the design issues such as joint friction, motor drive capacity, torque, or link weight can be ignored. However, a different setting of problem arises when it is attempted to depict the robot manipulator and its motion on computer screen using a Matlab program.

4. Simulation results

Our simulation consists of two sections; the program modeling and simulation control using joystick with forward and inverse kinematics methods to determine the possible working area of robot arm. To design the modeling, we assigned the angles and links as follows; $\theta_1 = -80^\circ$ to 80° , $\theta_2 = 0^\circ$ to 75° , $\theta_3 = -60^\circ$ to 90° , $L_1 = 80$ mm, $L_2 = 50$ mm, $L_3 = 150$ mm, and $L_4 = 180$ mm, respectively.

Table 1 shows the comparison between the controlled simulation results using the forward and inverse kinematics methods through a joystick. For forward kinematics method, we control the angles of each joint to get the position, while we control the end of tool to get the angles of each joint using inverse kinematics. The positions and orientations of simulated robot arm in XYZ, XZ, and XY planes are shown in Figures 6-8, respectively.

Table 1. Forward and inverse kinematic controlling

Position	Forward kinematics	Inverse kinematics
θ_1	32.9294°	33.4732°
θ_2	57.0846°	55.7698°
θ_3	40.3109°	38.1860°
P_x	292.3109 mm	288.2893 mm
P_y	189.3137 mm	190.6140 mm
P_z	199.5633 mm	200.0000 mm

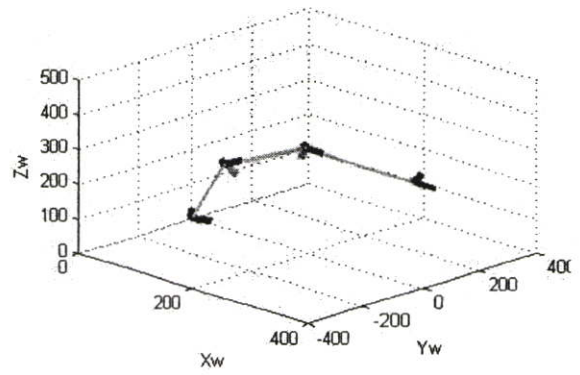


Figure 7 Simulated robot arm in XYZ Plane

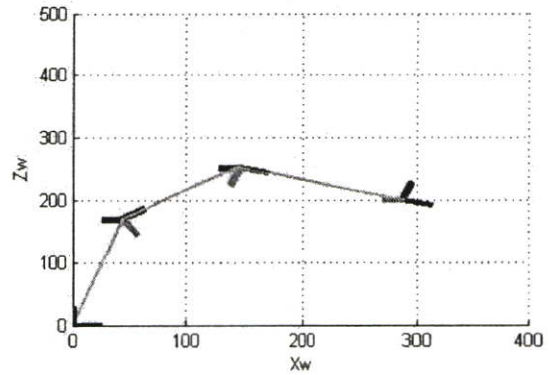


Figure 8 Simulated robot arm in XZ Plane

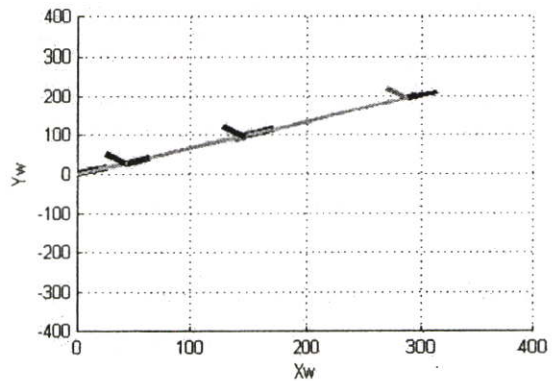


Figure 9 Simulated robot arm in XY Plane

To determine the working area, it can be calculated using forward kinematics method from the angle and link specifications as illustrated in Figures 9-11. The sign (*) shows the capability of trajectory of the end of tool from the desired angle and link specifications. The working area from simulation results is assigned as the safety path of a real working area and the orientation for inverse kinematics control. Before the real robot-arm implementation, if the change of working area is needed, the adjustments of angle and link will be

applied. (The design and real robot should be considered)

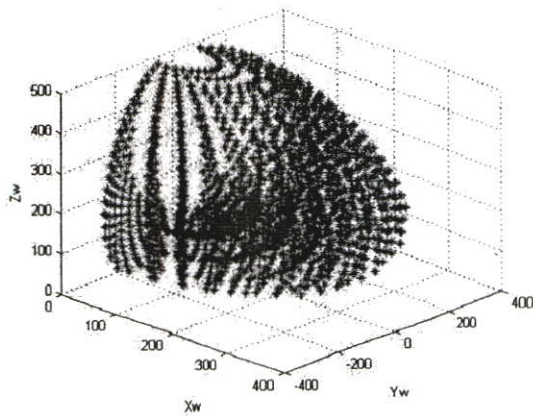


Figure 10 Trajectory of a Robot arm

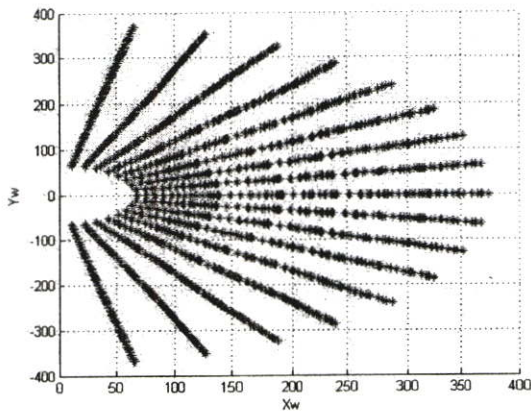


Figure 11 Working area of a Robot arm in XY plane

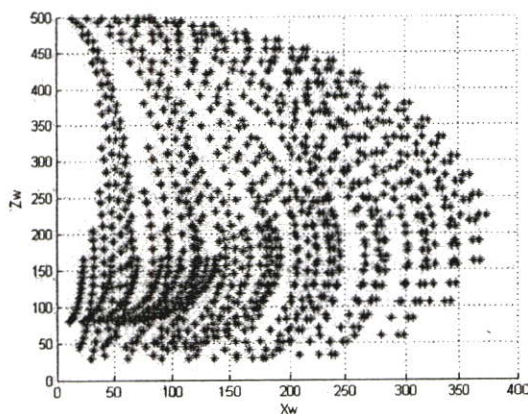


Figure 12 Working area of a Robot arm in XZ plane

5. Conclusion

The modeling and simulation of robot arm using the forward and inverse kinematics control has been presented in this paper. The proposed technique provides the analyzed and specified working area in order to obtain the safety path of a real working area before developing a prototype of robot arm.

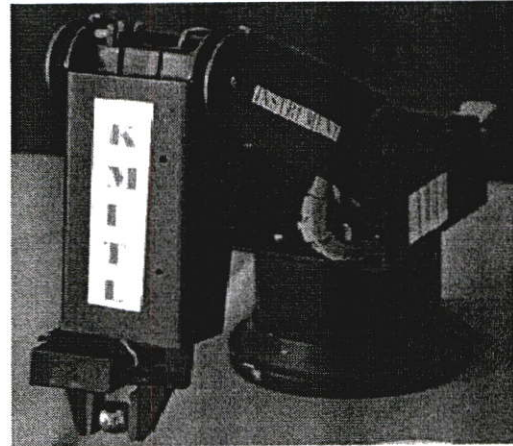


Figure 13 Robot arm pick up a ball

(Now, the prototype has been developed and submitted by King Mongkut's Institute of Technology Ladkrabang and has been shown in the KMITL Exhibition 2006.)

References

- [1] Marc Bodson "Fun Control Experiments with Matlab and a joystick", Proc. of the IEEE Conference on Decision and Control, Maui, HI, Dec. 2003.
- [2] Navinda Kottege & D.U.J. Sonnadara, *Virtual robot arm control model*, Institute of Physics, Proc. 20th Technical Session, Sri Lanka, March 2004.
- [3] Saeed B. Niku, *Introduction to Robotics Analysis, Systems, Applications*, Prentice Hall, Upper Saddle River, NJ, 2001.
- [4] P.G. Ranky ,C.Y.Ho, *Robot Modelling control and applications with software*, Springer-Verlag, 1985.

Optimal Vertical Maneuvers of the Aircraft with Thrust Vectoring In the Rigid Body Model Hiroyuki Takano	2100
Multi-vehicle Formation Using Range-Only Measurements Sunghwan Kim, Chang-Kyung Ryoo, Keeyoung Choi and Chunbae Park	2104
Controller Design of Missile Actuator Using DSP2812 Daeseong Jeong, Kwang-Jin Choi, Hyun-Wook Woo and Jong-Gi Kim	2110

FE10 : OS009 Navigation

The Development of an Artificial Vision based Navigation System for Helicopter Using modified NDGPS/INS Integration Jae Hyung Kim and Joon Lyou	2114
Robust and Precise Position Determination for Vehicles Using GPS Carrier Phase Measurements Eunsung Lee, Young Min Yoon Yoon, Moon Beom Heo and Ki Wook Nam	2119
Obstacle Avoidance using Active Window and Flexible Vector Field with a Laser Range Finder You Jun Choi, Jin Pyo Hong and Kyi Whan Park	2123
GNSS Receiver Tracking Loop Design for Spinning Vehicles Jeong Won Kim, Liu Melin, Dong-Hwan Hwang and Sang Jeong Lee	2129
A Study of INS/CDGPS Integration with a Scalar Adaptive Filter Junkyu Lim	2133

FE11 : OS032 Machine Vision for Display Inspection

Detecting Low-Contrast Defect Regions on LCD Glasses Using Highly Robust Model-Fitting (MF) Estimator Chang-Hwan Oh, Hyonam Joo and Keun-Ho Rew	2138
--	------

FE12 : OS016 Soft Computing Techniques and Their Applications

Detecting Method of Music to Match the User's Mood in Prefrontal Cortex EEG Activity Using the GA Shin-ichi Ito, Yasue Mitsu, Minoru Fukumi and Jianting Cao	2142
An Apparent Age Estimation System Using the Evolutionary Algorithm Hironobu Fukai, Takimoto Hironori, Mitsukura Yasue and Fukumi Minoru	2146
Utilization of Evolutionary Algorithms for Making the Environmental Game Much More Exciting Norio Baba and Hisashi Handa	2150
ANN based NGN IP Traffic Prediction in Thailand Suriya Satsri, Sriwichai Ardhan, Ornlarp Sangaroon and Vanvisa Chutchavong	2154
The Room Design System of Individual Preference with IGA Yosuke Fukada, Keiko Sato, Yasue Mitsukura and Minoru Fukumi	2158
PTGVLR: Fast MLP Learning using Parallel Tangent Gradient with Variable Learning Rates Payman Moallem and Arash Kiyoumars	2162

FE13 : Automation & Modeling (2)

Temperature Control of Nonlinear System in a Waste Disposal Process by Selective Control Yi-chun Yeh, Masatoshi Ogawa, Harutoshi Ogai and Kazuo Sakiyama	2166
Modeling and Analysis of a Hybrid Mechanism with Kinematic and Force Redundancies Jae Heon Chung, Byung-Ju Yi and Whee Kuk Kim	2170
Implementation of a Virtual Environment for Development of a Spent Fuel Disposal Process Jong Youl Lee, Heuijoo choi and Jongwon Choi	2176

A Simple Technique to Modeling and Simulation Four-Axe Robot-Arm Control Jettiya Sripituk, Teerawat Thepmanee and Prapart Ukakimapurn	2180
Application of the Agent Concept to E-Beam Manufacturing System Sun Jong Lim, Jun Yeob Song and Joon Lyoo	2186
Optimal Assembly Path Planning Algorithm for Aircraft Part Maintenance Christiand . and Jungwon Yoon	2190

FE14 : Nonlinear and Optimal Control

Optimising LQR and Pole Placement to Control Buck Converter by Genetic Algorithm Mohammad Bayati Poodeh, Saeid Eshtehardiha, Arash Kiyoumarsari and Mohammad Ataei	2195
Digital Controller of Novel Voice Coil Motor Actuator for Optical Image Stabilizer Dong Hae Yeom	2201
Modular Method for the Computation of the Defining Polynomial of the Algebraic Riccati Equation Takuya Kitamoto and Tetsu Yamaguchi	2207
Locally Optimal and Robust Backstepping Design for C1 Vector Fields Sejin Kang, Juhoon Back, Hyungbo Shim and Jin Heon Seo	2212
Simulation Study of Modified Extremum Seeking Control Hiroaki Shimono, Hitoshi Takata and Tomohiro Hachino	2218
Decentralized Motion/Force Control of Cooperative Multi-manipulator Systems Alireza Ghaffarkhah and Nasser Sadati	2223

FEP : Poster Session (4)

Combination of Multiple Classifiers Using Probabilistic Method Heesung Lee, Sungjun Hong and Euntai Kim	2230
Human Identification Based on Gait Analysis Sungjun Hong, Heesung Lee, Imran Fareed Nizami, Sung-Je An and Euntai Kim	2234
Morphological Approach of Stereo Camera Based Human Motion Capture System SeWoong Jun, Joongeon Park, Changwoo Park, Il-Kyun Jung, Bongseok Kim and Young-Ouk Kim	2238
3D Head Pose-Normalization Using 2D and 3D Interaction JoongRock Kim, Sunjin Yu and Sangyoun Lee	2242
Large Free Form Measurement Using Diffractive Optical Elements Byoungchang Kim and Se Heon Kim	2246
The Curvature Profile Measurement Using White-light Scanning Interferometry Byoungchang Kim	2250
Bidirectional Homopolar Magnetic Bearing with Fault Tolerant Capability Uhnjoo Na	2254
Stabilization Control of an Aerial Robot by Experimental Setup Se Han Lee	2258
Simulation of Attitude Control for a Wheeled Inverted Pendulum Se Han Lee	2264
Design of a Novel 3-DOF Parallel-type Haptic Device with Redundant Actuation Han Sung Kim, Joon-Woo Kim, Sung-Hyun Han and Duk-Hee Park	2270
Design and Control of Two Types of Planar Translational Parallel Manipulators with Parallelogram Mechanisms Han Sung Kim, Joon-Woo Kim, Se-Han Lee and Joon-Ho Kim	2274
Topological Design of the 3-DOF Manipulator with a Constraining Mechanism Kun-Woo Park, Chi-Hyo Kim, Sung-Joo Kim and Jin-Ho Kyung	2278
Specification of Driving Constraints for Dynamics Simulation of a Parallel-Serial Manipulator Jin-Ho Kyung	2283
Stiffness Analysis and Design of a 3-DOF Parallel Robot with One Constraining Leg	