

หุ่นยนต์สำรวจ 6 ขา
HEXBOT

วสันต์ ฉันทเมธิกุล
วิชวุฒิ แผลงภูทอง
ณัฐวัตร กุมพานิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

หุ่นยนต์สำรวจ 6 ขา

HEXBOT

วสันต์	ฉันทเมธีกุล
วิชวุฒิ	แมลงภู่ทอง
ณัฐวัตร	กุ่มพานิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

HEXBOT

WASAN

CHANTAMATEKUL

WICHAWUT

MALAENGPOTHONG

NATTHAWAT

KUMPANID

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2017

ปริญญาานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์สำรวจ 6 ขา
 HEXBOT

ผู้จัดทำ	นายวสันต์	ฉันทเมธิกุล	57011142
	นายวิษุฒิ	แมลงภูทอง	57011167
	นายณัฐวัตร	กุ่มพานิชย์	57011562

..........อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์)

หุ่นยนต์สำรวจ 6 ขา

โดย

นายวสันต์ ฉันทเมธิกุล 57011142

นายวิษุฒิ แผลงภูทอง 57011167

นายณัฐวัตร กุมาพานิชย์ 57011562

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์

ปีการศึกษา 2560

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ถูกจัดทำขึ้นเพื่อออกแบบและประดิษฐ์หุ่นยนต์ 6 ขา (Hexbot) ส่งการผ่านสัญญาณ Bluetooth ด้วยสมาร์ตโฟนหรือคอมพิวเตอร์ โดยมีจุดประสงค์เพื่อศึกษาเกี่ยวกับสมการแขนกล และประยุกต์ใช้ในการเคลื่อนที่ของหุ่นยนต์ในรูปแบบต่างๆ ได้แก่ เดินหน้า ถอยหลัง หมุนซ้าย หมุนขวาได้ ซึ่งจำแนกรูปแบบการเคลื่อนที่ได้ 2 แบบคือ แบบกำหนดท่าทางการเคลื่อนไหวดำเนิน และแบบคำนวณการเคลื่อนไหวดำเนินจากสมการ ซึ่งตัวต้นแบบถูกสร้างและทดสอบ จากนั้นทำการเปรียบเทียบระหว่างการเคลื่อนที่ทั้ง 2 แบบ

HEXBOT

By

Mr.Wasan Chantamatekul 57011142

Mr.Wichawut Malaengpoothong 57011167

Mr.Natthawat Kumpanid 57011562

Advisor

Asst. Prof. Sumit Panaudomsup

Academic Year 2017

ABSTRACT

This project aims to design and build the prototype of six-legged robot (Hexbot) controlled via Bluetooth signal from smartphone or computer. The purposes are to study about robot arm equation and apply to robotic movements. The movements include forward, backward, rotating left and right. The Hexbot can be controlled by two different control method including fixed motion control and equation-based control. The prototype is built and tested, then the comparison between fixed motion and equation-based control is made.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้และโครงการขึ้นนี้ดำเนินการสำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์ ซึ่งท่านได้ให้ข้อคิดเห็น และคำแนะนำต่างๆ อันเป็นประโยชน์ในการดำเนินการทำโครงการนี้เป็นอย่างมาก อีกทั้งยังช่วยคิดแนวทางแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างทำโครงการขึ้นนี้

ขอขอบคุณคณะอาจารย์ เจ้าหน้าที่ภาควิชาวิศวกรรมการวัดและควบคุม เพื่อนพ้อง ญาติสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง รวมถึงผู้ปกครองและครอบครัวที่ให้การสนับสนุนในด้านต่างๆ อีกทั้งยังคอยเป็นกำลังใจมาโดยตลอดจนสำเร็จการศึกษา คณะผู้จัดทำรู้สึกประทับใจ และซาบซึ้งเป็นอย่างยิ่งสำหรับทุกความช่วยเหลือ

ผู้จัดทำ

นายวสันต์ ฉันทเมธิกุล

นายวิษุฒิ แมลงภู่ทอง

นายณัฐวัตร กุมพานิชย์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของปริญญานิพนธ์	1
1.2 วัตถุประสงค์การทำปริญญานิพนธ์	1
1.3 ขอบเขตของโครงการ	1
1.4 วิธีดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 ระยะเวลาการดำเนินงาน	3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	4
2.1 หุ่นยนต์	4
2.2 แขนกล	5
2.2.1 ข้อต่อ	5
2.2.2 จลนศาสตร์ทางตรงของแขนกล	8
2.2.3 จลนศาสตร์ผกผันของแขนกล	13
2.3 อัลกอริทึมในการควบคุมแขนกล	17
2.4 ไมโครคอนโทรลเลอร์	18
2.4.1 Mini Servo High Torque HS-225BB ข้อต่อ	18
2.4.2 Driver Module 16-Channel PWM/Servo	20
2.4.3 Bluetooth	20
2.4.4 Power Supply	20
2.4.5 Arduino Uno R3	21
2.5 โปรแกรมใช้งาน	22
2.5.1 Arduino IDE	22
2.5.2 MATLAB	22

สารบัญ(ต่อ)

	หน้า
2.5.3 SolidWorks	22
บทที่ 3 การออกแบบและแนวคิดการเขียนโปรแกรม	23
3.1 ขั้นตอนการดำเนินงาน	23
3.2 หลักการออกแบบหุ่นยนต์	23
3.2.1 อุปกรณ์ที่ใช้สำหรับสร้างชิ้นส่วนประกอบหุ่นยนต์	24
3.2.2 ข้อคำนึงในการออกแบบ	24
3.3 แนวคิดการเขียนโปรแกรม	24
3.3.1 การเขียนโปรแกรม	25
บทที่ 4 ผลการดำเนินงาน	26
4.1 โครงสร้างหุ่นยนต์	26
4.2 ส่วนประกอบทางไฟฟ้า	27
4.3 สมการการเคลื่อนที่ของหุ่นยนต์	27
4.4 การทำงานโปรแกรม	31
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	35
5.1 สรุปผลการทดลอง	35
5.2 ปัญหาและแนวทางการแก้ปัญหา	36
5.3 ข้อเสนอแนะ	36
เอกสารอ้างอิง	37
ภาคผนวก	38

สารบัญรูป

รูปที่	หน้า
2.1 การประยุกต์ใช้งานแขนกล	5
2.2 ข้อต่อแบบต่างๆ	5
2.3 แขนกลกำลังไฮดรูลิกในเกลียวที่ต้องการ	6
2.4 ตัวอย่างแขนกลในสองมิติที่มีข้อต่อ 2 ข้อต่อ	7
2.5 เกณฑ์ของสองข้อต่อบนเฟรมที่อยู่ติดกัน	9
2.6 การกำหนดเวกเตอร์และจุดกำเนิดให้กับเฟรมทั้งสอง	10
2.7 แผนผังข้อต่อของแขนกล PUMA 560	11
2.8 จลนศาสตร์ผกผันของแขนกล	14
2.9 แขนกลที่มีสามข้อต่อพร้อมเฟรมที่กำหนดให้แต่ละท่อนแขน	14
2.10 บล็อกไดอะแกรม Inverse Kinematics	17
2.11 Mini Servo High Torque HS-225BB	18
2.12 Driver Module 16-Channel PWM/Servo	20
2.13 Bluetooth HC05	20
2.14 Power Supply	20
2.15 Arduino Uno R3	21
3.1 แผนผังขั้นตอนการดำเนินงาน	23
3.2 ลำดับขั้นตอนการทำงานของโปรแกรม	24
4.1 โครงสร้างหุ่นยนต์ 6 ขา	26
4.2 โครงสร้างส่วนขา	26
4.3 การต่อวงจร	27
4.4 ภาพอย่างง่ายของโครงสร้างส่วนขา	28
4.5 ภาพอย่างง่ายของโครงสร้างหุ่นยนต์ 6 ขา	29
4.6 รูปแสดงการส่งค่าผ่านบลูทูธ	30
4.7 รูปแสดงหน้าต่างส่งข้อมูลสำหรับขา 1 ข้าง	31
4.8 ผลที่ได้จากการคำนวณ x, y, z	31
4.9 หน้าจอแสดงการทำงาน	32
4.10 โปรแกรมขณะทำงาน	33
4.11 แบบโครงสร้างหุ่นยนต์ 6 ขา	33
4.12 แบบโครงสร้างส่วนขาของหุ่นยนต์ 6 ขา	34

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินโครงการระหว่างเดือนสิงหาคม 2560 ถึงเดือนเมษายน 2561	3
2.1 รูปแบบการเคลื่อนที่ของหุ่นยนต์	4
2.2 Kinematic Parameter	11
2.3 พารามิเตอร์ D-H ของแขนกล	15
2.4 Performance Specifications	19
2.5 Physical Specifications	19
4.1 พารามิเตอร์ของแขนกล	28
5.1 ผลการทดลองส่งค่าผ่านบลูทูธ	35

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปริญญาโท

ในปัจจุบันนี้ถือได้ว่าเป็นสังคมแห่งการพัฒนา และการเรียนรู้แบบเปิด ซึ่งมีหลากหลายภาคแขนงวิชาที่กำลังคิดริเริ่มสร้างสิ่งใหม่ ซึ่งเป็นการรวบรวมองค์ความรู้หลายอย่างเข้าด้วยกันทั้งทางด้านเทคโนโลยี การสื่อสาร การคมนาคม และยิ่งรวมไปถึงเทคโนโลยีหุ่นยนต์ที่กำลังเข้ามามีบทบาทในชีวิตมนุษย์ในด้านต่างๆ เช่น ด้านอุตสาหกรรมการผลิต ด้านการสำรวจ ด้านการแพทย์ และในอนาคตหุ่นยนต์จะสามารถทำงานแทนมนุษย์ได้หลากหลายอย่างมากขึ้น เพื่อเพิ่มศักยภาพในแต่ละสายงานทั้งด้านความเร็ว ความปลอดภัย และการลงภาระค่าใช้จ่ายของภาคอุตสาหกรรม

นอกจากนี้ระบบอัตโนมัติในโรงงานอุตสาหกรรม หรือในสถานที่ต่างๆ จะมีเทคโนโลยีที่ได้รับมาจากต่างประเทศมากมาย จึงทำให้อยากริเริ่มที่จะต่อยอดจากองค์ความรู้ที่มีอยู่ให้เกิดศักยภาพเพิ่มมากขึ้น เพื่อให้เกิดองค์ความรู้ใหม่ที่เป็นของประเทศไทยเอง และสามารถที่จะนำไปประยุกต์ใช้งานในด้านต่างๆ ให้เกิดประโยชน์ได้ต่อไป

1.2 วัตถุประสงค์การทำปริญญาโท

1. เพื่อศึกษาการออกแบบโครงสร้างของหุ่นยนต์โดยใช้โปรแกรม SolidWorks
2. เพื่อศึกษาและทำความเข้าใจภาษาซี และการเขียนโปรแกรมเพื่อให้หุ่นยนต์ทำงานตามคำสั่ง
3. เพื่อศึกษาการเคลื่อนที่ของหุ่นยนต์ 6 ขาโดยการนำความรู้จากวิชา INDUSTRIAL ROBOTICS มาประยุกต์ใช้
4. เพื่อศึกษาการทำงานของแขนกล ซึ่งจะเห็นว่าหุ่นยนต์ 6 ขาเกิดจากการทำงานของแขนกลทั้ง 6 แขนจนเกิดการเคลื่อนที่ที่มีความสัมพันธ์กัน

1.3 ขอบเขตของโครงการ

1. ออกแบบหุ่นยนต์ให้มีน้ำหนักเบา โดย Servo แต่ละตัวต้องรับน้ำหนักตัวเองได้
2. สามารถเขียนสมการเคลื่อนที่ของหุ่นยนต์ รวมไปถึงการนำสมการไปเขียนโปรแกรมให้หุ่นยนต์สามารถเคลื่อนที่ตามสมการได้
3. หุ่นยนต์ที่ออกแบบเป็นขนาดจำลองเพื่อการศึกษาการเคลื่อนที่

1.4 วิธีดำเนินการ

1. วางแผนการดำเนินงาน
2. ศึกษาโครงสร้างทางกลของหุ่นยนต์ 6 ขา
3. ออกแบบระบบการทำงาน และการจัดตำแหน่งอุปกรณ์ต่างๆ โดยใช้โปรแกรม SolidWorks
4. สร้างชิ้นส่วนด้วยเครื่องพิมพ์สามมิติและประกอบหุ่นยนต์
5. ศึกษาการใช้งาน Software เช่น Arduino IDE, MATLAB
6. ศึกษาอุปกรณ์อิเล็กทรอนิกส์ที่จำเป็นต้องใช้ในโครงการ
7. เขียนโปรแกรมควบคุมการทำงานของเครื่องจักร
8. ปรับแก้ทั้งส่วนของโปรแกรม และโครงสร้างทางกล (Mechanics) ให้ทำงานควบคู่กันไปได้
อย่างเหมาะสม
9. ทดสอบการทำงาน
10. จัดทำสรุปโครงการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ใช้ความคิดในการทำโครงการ เพื่อเป็นการฝึกให้มีการทำงานเป็นระบบ มีความสามัคคี และมีความรับผิดชอบมากขึ้น
2. เข้าใจการเขียนโปรแกรมด้วยคำสั่งภาษาซี การใช้โปรแกรม MATLAB และการควบคุมการเคลื่อนที่ของหุ่นยนต์ด้วย Arduino มากยิ่งขึ้น
3. เข้าใจการทำงานของแขนกล และสามารถสั่งงานแขนจำนวนมากให้เกิดการทำงานร่วมกัน
4. รู้ถึงปัญหาต่างๆ และสามารถรับมือกับปัญหานั้นได้

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง



2.1 หุ่นยนต์

หุ่นยนต์ หรือโรบอต (Robot) คือ เครื่องจักรกลชนิดหนึ่ง มีลักษณะโครงสร้างและรูปร่างแตกต่างกัน หุ่นยนต์ในแต่ละประเภทจะมีหน้าที่การทำงานในด้านต่างๆ ตามการควบคุมโดยตรงของมนุษย์ การควบคุมระบบต่างๆ ในการสั่งงานระหว่างหุ่นยนต์และมนุษย์ สามารถทำได้โดยทางอ้อมและอัตโนมัติ โดยทั่วไปหุ่นยนต์ถูกสร้างขึ้นเพื่อสำหรับงานที่มีความยากลำบาก เช่น งานสำรวจในพื้นที่บริเวณแคบหรืองานสำรวจดวงจันทร์ ดาวเคราะห์ที่ไม่มีสิ่งมีชีวิต ในปัจจุบันเทคโนโลยีของหุ่นยนต์เจริญก้าวหน้าอย่างรวดเร็ว และเริ่มเข้ามามีบทบาทกับชีวิตของมนุษย์ในด้านต่างๆ เช่น ด้านอุตสาหกรรมการผลิตแตกต่างจากเมื่อก่อนที่หุ่นยนต์มักถูกนำไปใช้ ในงานอุตสาหกรรมเป็นส่วนใหญ่ ปัจจุบันมีการนำหุ่นยนต์มาใช้งานมากขึ้น เช่น หุ่นยนต์ที่ใช้ในทางการแพทย์ หุ่นยนต์สำหรับงานสำรวจ หุ่นยนต์ที่ใช้งานในอวกาศ หรือแม้แต่หุ่นยนต์ที่ถูกสร้างขึ้นเพื่อเป็นเครื่องเล่นของมนุษย์ จนกระทั่งในปัจจุบันนี้ได้มีการพัฒนาให้หุ่นยนต์นั้นมีลักษณะที่คล้ายมนุษย์ เพื่อให้อาศัยอยู่ร่วมกับมนุษย์ ให้ได้ในชีวิตประจำวัน

ประเภทของหุ่นยนต์อุตสาหกรรม

การแบ่งชนิดของหุ่นยนต์จะแบ่งตามลักษณะรูปทรงของพื้นที่ทำงาน (Envelope Geometric) โดยเกิดจากจุดต่อ (Joint) ทั้งสองแบบดังตารางที่ 2.1 เมื่อนำจุดต่อมาต่อเข้าด้วยกันอย่างน้อย 3 แกนหลักจะได้พื้นที่ทำงาน (Work Envelope) ที่มีลักษณะแตกต่างกันไป

ตารางที่ 2.1 รูปแบบการเคลื่อนที่ของหุ่นยนต์

ชนิด	สัญลักษณ์	หมายเหตุ
Revolute (R)		เป็นการหมุน รอบแกน (Rotary)
Prismatic (P)		การเคลื่อน ที่เชิงเส้น (Linear motion)

2.2 แขนกล

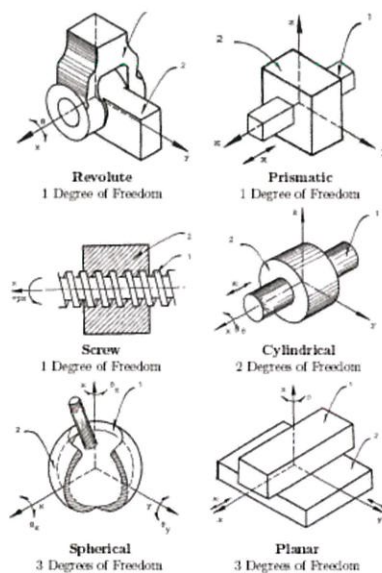
หุ่นยนต์ที่พิจารณาในเรื่องการวางแผนการเคลื่อนที่นั้น ถูกจำกัดว่าเป็นวัตถุชิ้นเดียวเพื่อความสะดวกในการวิเคราะห์ แต่ในความเป็นจริงโดยทั่วไปหุ่นยนต์ประกอบขึ้นจากชิ้นส่วนต่างๆ หลายชิ้นที่เคลื่อนที่สัมพันธ์กันนอกเหนือจากหุ่นยนต์เคลื่อนที่ได้ ยังมีหุ่นยนต์อีกประเภทหนึ่งที่พบเห็นได้ในโรงงานประกอบชิ้นส่วนนั้นคือ แขนกล ปัจจุบันแขนกลได้มีบทบาทข้ามขอบเขตของโรงงานเข้ามาในชีวิตประจำวัน เพิ่มมากขึ้นเรื่อยๆ ดังรูปที่ 2.1



รูปที่ 2.1 การประยุกต์ใช้งานแขนกล

2.2.1 ข้อต่อ

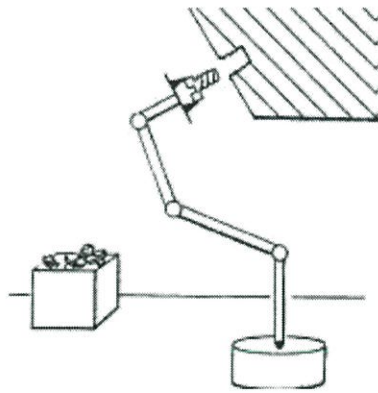
แขนกลคือ หุ่นยนต์ที่ประกอบด้วยท่อนแขน (Link) ที่นำมาประกอบกันด้วยข้อต่อ (Joint) ข้อต่อมีหลายแบบ แต่ละแบบก็จะอนุญาตให้เกิดการเคลื่อนที่แตกต่างกันไป ดังรูปที่ 2.2 แสดงข้อต่อแบบต่างๆ ที่นิยมใช้



รูปที่ 2.2 ข้อต่อแบบต่างๆ

ในการสร้างแขนกลโดยทั่วไป ข้อต่อที่นิยมใช้มากที่สุดคือ ข้อต่อแบบหมุน (Revolute Joint) และข้อต่อแบบเลื่อน (Prismatic Joint) สำหรับข้อต่อแบบหมุน ท่อนแขนสองท่อนถูกยึดติดที่จุดหมุน ซึ่งอยู่บนท่อนแขนโดยแต่ละท่อนสามารถหมุนได้รอบจุดหมุนนี้ สามารถบอกตำแหน่งของสองท่อนที่แขนสัมพันธ์กันด้วยมุมที่ท่อนแขนหมุนไป ส่วนข้อต่อแบบเลื่อนนั้น ท่อนแขนสองท่อนติดอยู่ด้วยกัน ในลักษณะเดียวกันกับเสาอากาศวิทยุรถยนต์ที่ยึดติดได้ โดยท่อนแขนแต่ละท่อนสามารถเลื่อนเข้าออกได้ในทิศทางหนึ่ง สามารถระบุตำแหน่งที่สัมพันธ์กันของสองท่อนแขนได้จากระยะเลื่อนเข้าออกดังกล่าว จะเห็นได้ว่าข้อต่อแบบหมุนและข้อต่อแบบเลื่อนมีระดับเสรีของการเคลื่อนที่เป็นหนึ่ง เรียกว่าตัวแปรที่กำหนดการเคลื่อนที่นี้ซึ่งได้แก่มุมหมุนของข้อต่อแบบหมุน และระยะเลื่อนของข้อต่อแบบเลื่อนว่าเป็นพารามิเตอร์ของข้อต่อ การมีระดับเสรีของการเคลื่อนที่เป็นหนึ่งทำให้ง่ายในการออกแบบและวิเคราะห์ข้อต่อทั้งสองแบบจึงถูกใช้มากที่สุดในการสร้างแขนกล โดยแขนกลที่มีระดับเสรีสูงๆ ก็ สามารถสร้างขึ้นได้ โดยการประกอบท่อนแขนหลายท่อนด้วยข้อต่อสองแบบนี้

แขนกลทำงานด้วยการเคลื่อนที่ของท่อนแขนที่สัมพันธ์กันเพื่อให้ปลายแขน (End Effector) ไปอยู่ในตำแหน่งและทิศทางที่เหมาะสม เพื่อเครื่องมือที่ติดอยู่ที่ปลายแขนจะได้ทำงานที่ต้องการได้ โดยสะดวกและมีประสิทธิภาพดังในรูปที่ 2.3 แสดงให้เห็นถึงความจำเป็นที่จะต้องจัดการให้ปลายแขนอยู่ในตำแหน่งและทิศทางที่เหมาะสม



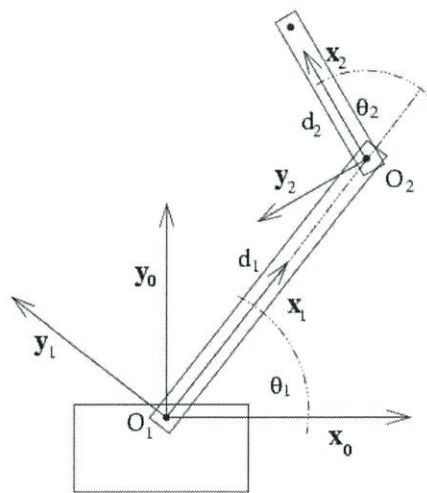
รูปที่ 2.3 แขนกลกำลังไขสกรูลงในเกลียวที่ต้องการ

การคำนวณว่าปลายแขนจะอยู่ที่ตำแหน่งและทิศทางใดจึงเป็นเรื่องสำคัญ การคำนวณดังกล่าวอาศัยการกำหนดให้ท่อนแขนแต่ละท่อนมีพิกัดส่วนตัว ที่เรียกว่าเฟรม เฟรมประกอบไปด้วยจุดกำเนิดและเวกเตอร์แกน โดยเฟรมที่กล่าวถึงจะอยู่ติดแน่นกับท่อนแขนที่เป็นเจ้าของเสมอ หรืออีกนัยหนึ่งก็คือ แต่ละท่อนแขนจะอยู่นิ่งไม่ขยับเขยื้อนเมื่อเทียบกับเฟรมของมัน สำหรับท่อนแขนที่เกิดจากการเรียงต่อกันไป นิยมเรียกท่อนแขนที่อยู่หนึ่งยึดติดกับพื้นว่าฐาน (Base) และเรียกท่อนถัดมาตามชื่อส่วนของแขน ได้แก่ ไหล่ (Shoulder) ข้อศอก (Elbow) แขนท่อนบน (Forearm) และข้อมือ (Wrist) เป็นต้น ตำแหน่งและทิศทางการวางตัวของท่อนแขนหนึ่งๆ เมื่อเทียบกับเฟรมของฐานจึงขึ้นอยู่กับตำแหน่งและทิศทางของท่อนแขนก่อนๆ ด้วย สามารถคำนวณตำแหน่งและทิศทางของปลายแขนได้

ด้วยการใช้การแปลงเอกพันธ์ที่ได้ศึกษาไปแล้ว โดยทำการคูณเมทริกซ์การแปลงแบบซ้ายไปขวา พิจารณาจากฐานไปจนถึงปลายแขน ตัวอย่างต่อไปนี้แสดงการคำนวณตำแหน่งของแขนกลในสองมิติ ที่มีสองข้อต่อ ดังรูปที่ 2.4 ให้เฟรมของฐานคือ F_0 มีจุดกำเนิด O_1 และเวกเตอร์แกน X_0 และ Y_0 ให้เฟรมของ L_1 คือ F_1 มีจุดกำเนิด O_1 และเวกเตอร์แกน X_1 และ Y_1 และให้เฟรมของ L_2 คือ F_2 มีจุดกำเนิด O_2 และเวกเตอร์แกน X_2 และ Y_2 กำหนดให้

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & \theta \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$R(a, b) = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$



รูปที่ 2.4 ตัวอย่างแขนกลในสองมิติที่มีข้อต่อ 2 ข้อต่อ

สามารถเขียน F_1 ได้จากการหมุน F_0 รอบจุดกำเนิดไปเป็นมุม θ_1 ดังนั้นพิกัดใน F_1 เขียนเป็นพิกัดใน F_0 ได้ด้วยการคูณข้างหน้าด้วย $R(\theta_1)$ เช่นเดียวกัน สามารถเขียน F_2 ได้จากการเลื่อน F_1 ไปตามเป็นระยะ d_1 จากนั้นหมุนรอบจุดกำเนิดของเฟรมที่ได้จากการเลื่อนนี้ไปเป็นมุม θ_2 นั่นคือ พิกัดใน F_2 เขียนเป็นพิกัดใน F_1 ด้วยการคูณข้างหน้าด้วย $D(d_1, 0) R(\theta_2)$ เมื่อนำมาประกอบกันจะได้ว่าพิกัดใน F_2 สามารถเขียนเป็นพิกัดใน F_0 ด้วยการคูณข้างหน้าด้วย $R(\theta_1) D(d_1, 0) R(\theta_2)$

ปลายแขนที่มีพิกัดเอกพันธ์ใน F_2 คือ $\begin{bmatrix} d_2 \\ 0 \\ 1 \end{bmatrix}$ จะมีพิกัดใน F_0 คือ

$$\begin{aligned} & R(\theta_1)D(d_1, 0)R(\theta_2) \begin{bmatrix} d_2 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & d_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_2 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} d_2 \cos(\theta_1 + \theta_2) + d_1 \cos\theta_1 \\ d_2 \cos(\theta_1 + \theta_2) + d_1 \sin\theta_1 \\ 1 \end{bmatrix} \end{aligned} \quad (2.3)$$

จะเห็นได้ว่าสามารถเขียนตำแหน่งและทิศทางของปลายแขนในพิกัดของฐานให้อยู่ในรูปของพารามิเตอร์ของข้อต่อ หลักการสำคัญก็คือ การกำหนดพิกัดส่วนตัวที่เหมาะสมให้แก่ละท่อนแขนถึงแม้ว่าในทางทฤษฎีแล้วจุดกำเนิดของพิกัดจะอยู่ที่ใดก็ได้ (ให้ยึดติดกับท่อนแขนที่เป็นเจ้าของ) แต่ในทางปฏิบัตินิยมเลือกให้จุดกำเนิดของพิกัดอยู่ตรงข้อต่อ นอกจากนี้ในตัวอย่างที่ผ่านมา กำหนดให้เวกเตอร์แกน X_1 ผ่านทั้ง O_1 และ O_2 บังคับให้การเคลื่อนที่ต้องการอยู่ในแกนเดียวเท่านั้น เพื่อสะดวกในการคำนวณ

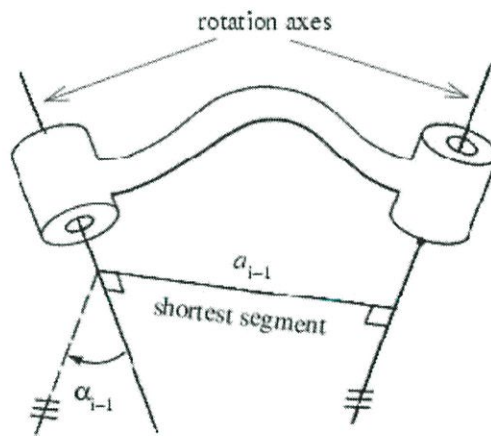
การคำนวณพิกัดของจุดใดๆ ของแขนกลให้อยู่ในรูปของพิกัดฐานเป็นการคำนวณการแปลงในลักษณะต่อเนื่องกันเป็นลูกโซ่ โดยพิจารณาจากพิกัดฐานไปยังพิกัดที่จุดที่ต้องการนั้นเขียนอยู่ นิยมอ้างถึงท่อนแขนที่เรียงต่อกันไปเริ่มจากฐานด้วยจำนวนเต็ม $0, 1, 2, \dots$ จึงเขียนได้ว่า $T_{0,n} = T_{0,1}T_{1,2} \dots T_{n-1,n}$ โดย $T_{i,j}$ เป็นเมทริกซ์การแปลงเอกพันธ์จากพิกัดในเฟรม F_j ไปเป็นพิกัดในเฟรม F_i

2.2.2 จลนศาสตร์ทางตรงของแขนกล (Forward Kinematics of Manipulator)

จลนศาสตร์สนใจเกี่ยวกับการเรียงตัว ตำแหน่ง ทิศทาง และการเปลี่ยนแปลงของสิ่งต่างๆ เหล่านี้ของโครงสร้างโดยไม่คำนึงถึงแรงต้นเหตุ จลนศาสตร์ทางตรงของแขนกลว่าด้วยการคำนวณตำแหน่งในพิกัดที่ถือว่าหยุดนิ่ง ของจุดบนแขนกลที่กำหนดให้ ตัวอย่างในหัวข้อก่อนได้แสดงการคำนวณดังกล่าวในสองมิติ ในหัวข้อนี้จะศึกษาจลนศาสตร์ทางตรงของแขนกลในกรณีสามมิติ หลักการคำนวณยังคงเหมือนในกรณีสองมิติ หากแต่จะเสนอวิธีการจัดระบบตัวแปรต่างๆ อย่างมีระเบียบ ทั้งนี้ก็เพื่อการคำนวณจะทำได้ง่ายมีระบบไม่สับสน

หลักการคือ ท่อนแขนที่เรียงต่อกันจากฐานนับเป็นท่อนที่ 0 ไปจนถึงปลายแขนท่อนที่ n ให้เป็นเฟรมของท่อนแขนที่ i อันประกอบไปด้วยจุดกำเนิด O_i และเวกเตอร์แกน X_i, Y_i, Z_i สามารถคำนวณเช่นเดียวกันกับกรณีสองมิติได้ว่าจุด P ที่มีพิกัดเป็น p_n ในเฟรม F_0 จะมีพิกัดในเฟรม F_0 เป็น

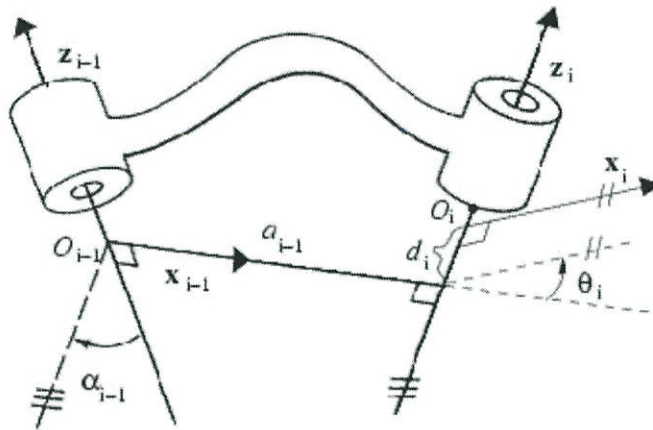
$p_0 = T_{0,n}P_n$ โดยที่ $T_{0,n} = T_{0,1}T_{1,2}\dots T_{n-1,n}$ ดังนั้นสิ่งที่ต้องสนใจก็คือ การหา $T_{i-1,n}$ เริ่มด้วยวิธีการกำหนดจุดกำเนิดและเวกเตอร์แกนให้กับเฟรมต่างๆ ดังรูปที่ 2.5 แสดงข้อต่อที่อยู่ติดกันสองข้อต่อนี้ต้องอยู่บนเฟรมที่ติดกันด้วย สมมติว่าข้อต่อทางซ้ายอยู่ในเฟรม F_{i-1} และข้อต่อทางขวาอยู่ในเฟรม F_i พิจารณาเส้นตรงสองเส้นที่ลากผ่านแกนหมุนของข้อต่อทั้งสอง หากสองเส้นตรงนี้ไม่ขนานกันและไม่ตัดกัน จะสามารถหาส่วนของเส้นตรงที่สั้นที่สุดที่เชื่อมเส้นตรงที่ผ่านแกนนี้ได้ แน่ใจว่าส่วนของเส้นตรงนี้ต้องอยู่ในทิศทางที่ตั้งฉากกับเส้นตรงที่ผ่านแกนทั้งสอง ในรูปที่ 2.5 ระยะห่างที่สั้นที่สุดนี้ก็คือ a_{i-1} และมุมระหว่างภาพฉายในทิศทางของเส้นตรงที่สั้นที่สุดของเส้นตรงที่ผ่านแกนทั้งสองเท่ากับ α_{i-1} มุมนี้ก็คือมุมน้อยที่สุดที่เส้นตรงที่ผ่านแกนต้องบิดเพื่อมาขนานกัน



รูปที่ 2.5 แกนหมุนของสองข้อต่อบนเฟรมที่อยู่ติดกัน

ในรูปที่ 2.6 ให้ L_{i-1} คือ เส้นสั้นที่สุดที่เชื่อมระหว่างแกนหมุนของข้อต่อทางซ้ายกับแกนหมุนของข้อต่อทางขวา กำหนดให้จุดกำเนิดของเฟรม F_{i-1} อยู่ที่ O_{i-1} ซึ่งก็คือ จุดตัดระหว่างแกนหมุนของข้อต่อทางซ้ายกับ L_{i-1} นอกจากนี้ยังกำหนดให้เวกเตอร์แกน Z_{i-1} เป็นเวกเตอร์ที่ขนานกับแกนหมุนของข้อต่อซ้าย และเวกเตอร์แกน X_{i-1} เป็นเวกเตอร์ที่ขนานกับ L_{i-1} โดยชี้จากแกนหมุนของข้อต่อซ้ายไปยังแกนหมุนของข้อต่อขวา ในทำนองเดียวกันให้ L_i เป็นเส้นสั้นที่สุดที่เชื่อมระหว่างแกนหมุนของข้อต่อทางขวากับแกนหมุนของข้อต่อที่อยู่ถัดไป (ไม่ได้แสดงในรูป) กำหนดให้จุด O_i เป็นจุดกำเนิดของเฟรม F_i โดยจุดนี้เป็นจุดตัดระหว่างแกนหมุนของข้อต่อทางขวากับ L_i กำหนดให้เวกเตอร์แกน Z_i เป็นเวกเตอร์ขนานกับแกนหมุนของข้อต่อทางขวาและเวกเตอร์แกน X_i เป็นเวกเตอร์ขนานกับ L_i โดยชี้จากแกนหมุนของข้อต่อทางขวาไปยังแกนหมุนของข้อต่อถัดไป ถึงจุดนี้จึงกำหนดจุดกำเนิดและเวกเตอร์แกนให้กับเฟรม F_{i-1} และ F_i เรียบร้อยแล้ว (Y_{i-1} และ Y_i หาได้จากการใช้กฎมือขวากับเวกเตอร์แกนที่กำหนดไปแล้ว) จึงไม่ยากที่จะหาการแปลง $T_{i-1,i}$ ที่ต้องการ วิธีทำคือ ต้องหาการเคลื่อนที่ซึ่งทำให้เฟรม F_{i-1} มาอยู่ทับกับเฟรมจากรูปที่ 2.6 จะเห็นได้ว่าสิ่งที่ต้องทำก็คือ หมุนเฟรม F_{i-1} รอบแกน X_{i-1} เป็นมุม α_{i-1} จากนั้นเลื่อนเฟรมที่ได้ตามแกนเป็นระยะทาง X_{i-1} ถึงตรงจุดนี้แกน Z_{i-1} จะอยู่ทับกับแกน Z_i แต่

เวกเตอร์แกน X_{i-1} กับ X_i จะยังไม่ขนานกัน รวมทั้ง O_{i-1} และ O_i ห่างกันเป็นระยะ d_i นั่นคือ ต้องหมุน F_{i-1} ต่อบนแกนเป็นมุม θ และเลื่อน F_{i-1} ไปตามแกน Z_{i-1} เป็นระยะ d_i โดยสรุปเขียนเป็นสูตรได้คือ

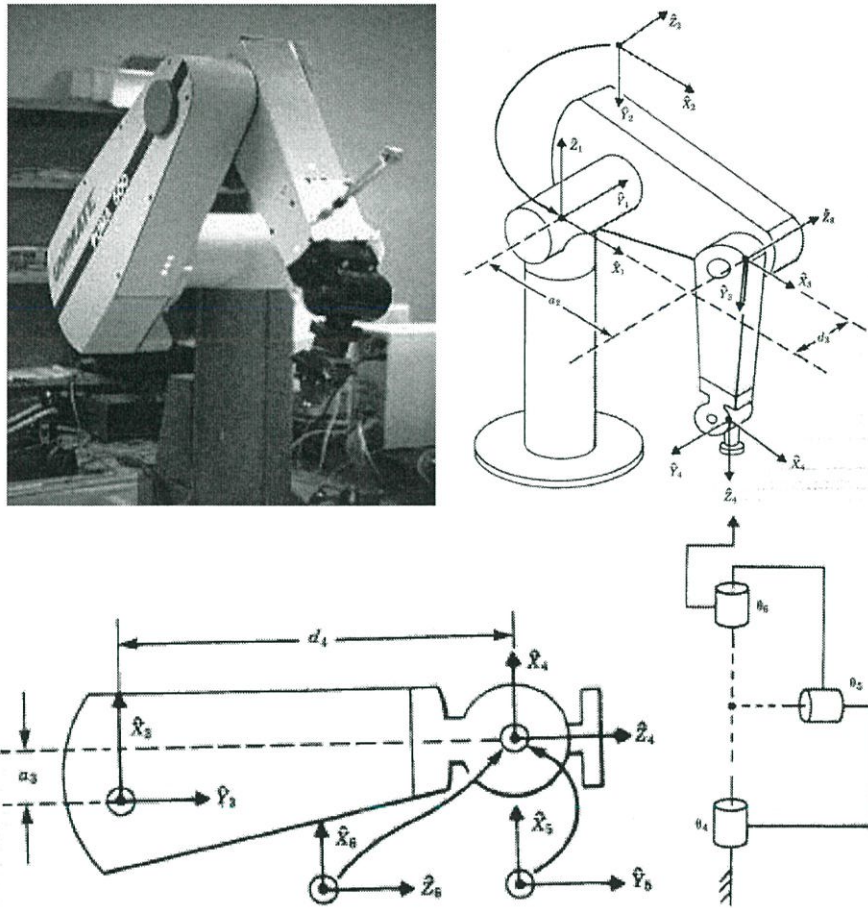


รูปที่ 2.6 การกำหนดเวกเตอร์และจุดกำเนิดให้กับเฟรมทั้งสอง

$$\begin{aligned}
 T_{i,i} &= \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & \cos\alpha_{i-1} & -\sin\alpha_{i-1} & 0 \\ 0 & \sin\alpha_{i-1} & \cos\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 1 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i \cos\alpha_{i-1} & \cos\theta_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i \sin\alpha_{i-1} \\ \sin\theta_i \sin\alpha_{i-1} & \cos\theta_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i \cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)
 \end{aligned}$$

จะเห็นได้ว่าขั้นตอนการหา $T_{0,n} = T_{0,1}T_{0,2}\dots T_{n-1,n}$ เริ่มได้จากการพิจารณาท่อนแขนที่ 0 ไปจนถึงท่อนแขนที่ n แล้วพยายามกำหนดเฟรมของแต่ละท่อนแขน โดยเริ่มจากการกำหนดแกน Z_i ของแต่ละท่อนแขน i (หากเป็นข้อต่อแบบหมุน กำหนดให้แกนนี้ผ่านแกนหมุน) จากนั้นกำหนดจุดกำเนิด O_i และแกน X_i จากส่วนของเส้นตรงสั้นสุด ที่เชื่อมระหว่าง Z_i และ Z_{i+1} (L_i) เมื่อเฟรมถูกกำหนดแล้วค่าของ $a_{i-1}, \alpha_{i-1}, d_i, \theta_i$ สำหรับ $i = 1, 2, \dots, n$ ที่ต้องการก็สามารถคำนวณได้โดยง่าย

ลองพิจารณาการคำนวณที่กล่าวไปแล้วกับแขนกลจริง รูปที่ 2.7 แสดงแผนผังของข้อต่อของแขนกล PUMA 560 แขนกลรุ่นนี้เป็นผลิตภัณฑ์ของบริษัท Unimation ที่ได้รับความนิยมเป็นอย่างมาก มีระดับความเสถียรเป็นหก สามข้อต่อแรกจากฐานควบคุมการเคลื่อนที่แบบหยาบ และสามข้อต่อปลายเป็นตัวหลักในการควบคุมทิศทางของอุปกรณ์ที่ติดอยู่ที่ปลายแขน



รูปที่ 2.7 แผนผังข้อต่อของแขนกล PUMA 560

ตารางที่ 2.2 Kinematic Parameter

i	α_{i-1}	a_{i-1}	d_i	θ
1	0	0	0	θ_1
2	$-\pi/2$	0	0	θ_2
3	0	a_2	d_3	θ_3
4	$-\pi/2$	a_3	d_4	θ_4
5	$\pi/2$	0	0	θ_5
6	$-\pi/2$	0	0	θ_6

เขียนได้ว่า

$$T_{0,1} = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{1,2} = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin q_2 & \cos q_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{2,3} = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & a_2 \\ \sin q_3 & \cos q_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{3,4} = \begin{bmatrix} \cos q_4 & -\sin q_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ \sin q_4 & \cos q_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{4,5} = \begin{bmatrix} \cos q_5 & -\sin q_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin q_5 & \cos q_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{5,6} = \begin{bmatrix} \cos q_6 & -\sin q_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin q_6 & \cos q_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{ຈຸດໃຈ} \quad T_{0,6} = T_{0,1}T_{1,2}T_{2,3}T_{3,4}T_{4,5}T_{5,6} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{ໂຕຢູນິດ}$$

$$r_{11} = \cos\theta_1[\cos(\theta_2 + \theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6) - \sin(\theta_2 + \theta_3)\sin\theta_5\cos\theta_6 + \sin\theta_1[\sin\theta_4\cos\theta_5\cos\theta_6 + \cos\theta_4\sin\theta_6]]$$

$$r_{21} = \sin\theta_1[\cos(\theta_2 + \theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6) - \sin(\theta_2 + \theta_3)\sin\theta_5\cos\theta_1] - \cos\theta_1[\sin\theta_4\cos\theta_5\cos\theta_6 + \cos\theta_4\sin\theta_6]$$

$$r_{31} = -\sin(\theta_2 + \theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6) - \cos(\theta_2 + \theta_3)\sin\theta_5\sin\theta_6$$

$$r_{12} = \cos\theta_1[\cos(\theta_2 + \theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\cos\theta_6) + \sin(\theta_2 + \theta_3)\sin\theta_5\sin\theta_6] + \sin\theta_1[\cos\theta_4\cos\theta_6 - \sin\theta_4\cos\theta_5\sin\theta_6]$$

$$r_{22} = \sin\theta_1[\cos(\theta_2 + \theta_3)(\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\cos\theta_6) + \sin(\theta_2 + \theta_3)\sin\theta_5\sin\theta_6] - \cos\theta_1[\cos\theta_4\cos\theta_6 - \sin\theta_4\cos\theta_5\sin\theta_6]$$

$$r_{32} = -\sin(\theta_2 + \theta_3)(-\cos\theta_4\cos\theta_5\sin\theta_6 - \sin\theta_4\cos\theta_6) + \cos(\theta_2 + \theta_3)\sin\theta_5\sin\theta_6$$

$$r_{13} = -\cos\theta_1(\cos(\theta_2 + \theta_3)\cos\theta_4\sin\theta_5 + \sin(\theta_2 + \theta_3)\cos\theta_5) - \sin\theta_1\sin\theta_4\sin\theta_5$$

$$r_{23} = -\sin\theta_1(\cos(\theta_2 + \theta_3)\cos\theta_4\sin\theta_5 + \sin(\theta_2 + \theta_3)\cos\theta_5) - \cos\theta_1\sin\theta_4\sin\theta_5$$

$$r_{33} = \sin(\theta_2 + \theta_3)\cos\theta_4\sin\theta_5 + \cos(\theta_2 + \theta_3)\cos\theta_5$$

$$p_x = \cos\theta_1[a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3) - d_4\sin(\theta_2 + \theta_3)] - d_3\sin(\theta_1)$$

$$p_y = \sin\theta_1[a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3) - d_4\sin(\theta_2 + \theta_3)] + d_3\sin(\theta_1)$$

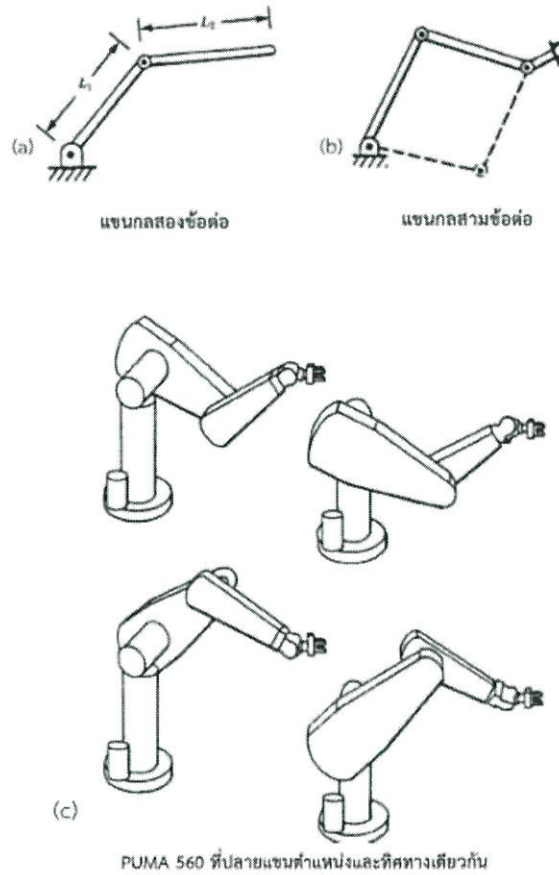
$$p_z = -a_3\sin(\theta_2 + \theta_3) - a_2\sin\theta_2 - d_4\cos(\theta_2 + \theta_3)$$

2.2.3 จลนศาสตร์ผกผันของแขนกล (Inverse Kinematics of Manipulator)

แขนกลส่วนใหญ่จะมาพร้อมกับอุปกรณ์ที่เรียกว่าแป้นสอน (Teach Pendant) สามารถสั่งงานให้แขนกลเคลื่อนไปอยู่ในลักษณะที่ต้องการ โดยการปรับพารามิเตอร์ของข้อต่อจากแป้นสอนจนแขนกลอยู่ในลักษณะที่พอใจ แล้วให้แป้นสอนจดจำพารามิเตอร์ของข้อต่อเหล่านี้ไว้ ในภายหลังหากต้องการให้แขนกลไปอยู่ในลักษณะเดิมอีก เพียงแต่ใช้แป้นสอนสั่งงานให้แขนกลเคลื่อนที่ไปยังจุดที่ได้จำไว้แล้ว แต่จะเห็นได้ว่าการสั่งงานในลักษณะนี้ต้องทำงานในปริภูมิของพารามิเตอร์ของข้อต่อ ซึ่งไม่สะดวกโดยเฉพาะอย่างยิ่งในกรณีที่ต้องการระบุพิกัดคาร์ทีเซียนและทิศทางของปลายแขน ในหัวข้อนี้จะศึกษาการคำนวณที่ตรงข้ามกับจลนศาสตร์ทางตรงของแขนกล

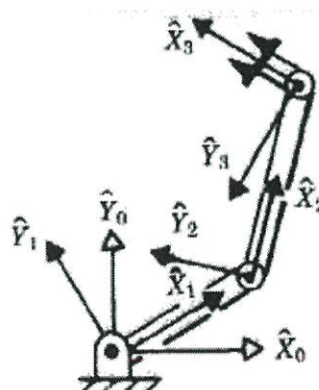
กำหนดพิกัดคาร์ทีเซียนและทิศทางในเฟรมที่หุ้มนิ่งของปลายแขนให้ ต้องการคำนวณพารามิเตอร์ของข้อต่อที่ทำให้ปลายแขนเรียงตัวในลักษณะที่กำหนด เรียกการคำนวณในลักษณะนี้ว่าการคำนวณจลนศาสตร์ผกผันของแขนกล

การคำนวณจลนศาสตร์ผกผันของแขนกลต้องคำนึงถึงสองเรื่องหลัก เรื่องแรกคือ พารามิเตอร์ของข้อต่อที่ต้องการอาจไม่มี เรื่องที่สองคือ พารามิเตอร์คำตอบที่เป็นไปได้อาจมีหลายชุด รูปที่ 2.8(a) แสดงแขนกลที่มีข้อต่อแบบหมุนสองข้อต่อโดยท่อนแขนทั้งสองที่แสดงในรูปมีความยาวเป็น L_1 และ L_2 จะเห็นได้ว่าไม่มีพารามิเตอร์ของข้อต่อที่ทำให้ปลายแขนอยู่ห่างจุดหมุนที่ฐานเกิน L_1+L_2 ไปได้ และหากกำหนดให้จุดปลายแขนอยู่ห่างจากจุดหมุนที่ฐานน้อยกว่า L_1+L_2 หากแต่ละข้อต่อสามารถหมุนได้ครบรอบก็จะได้ว่ามีพารามิเตอร์คำตอบอย่างมาสองชุด ในรูปที่ 2.8(b) แสดงสองชุดของพารามิเตอร์ข้อต่อที่เป็นคำตอบสำหรับตำแหน่งและทิศทางของปลายแขนที่กำหนดให้ การมีพารามิเตอร์คำตอบมากกว่าหนึ่งชุด ถึงแม้จะทำให้ขั้นตอนการหาคำตอบอาจยุ่งยากกว่าการมีคำตอบเดียว แต่คุณสมบัตินี้ทำให้สามารถเลือกชุดพารามิเตอร์ข้อต่อที่เหมาะสมกับงานเป้าหมาย เช่น เลือกใช้ชุดพารามิเตอร์ที่อ้อมหลบสิ่งกีดขวาง แขนกลที่ใช้งานจริงเช่น PUMA 560 ก็ได้ถูกออกแบบให้มีหลายชุดคำตอบ เช่น ตามแสดงในรูปที่ 2.8(c) แขนกลมีสี่ชุดคำตอบที่ให้ตำแหน่งและทิศทางเดียวกันของปลายแขน



รูปที่ 2.8 จลนศาสตร์ผกผันของแขนกล

ตัวอย่างการคำนวณจลนศาสตร์ผกผัน เพื่อให้เกิดความเข้าใจในขั้นตอนการหาพารามิเตอร์คำตอบ พิจารณาแขนกลที่มีสามข้อต่อในรูปที่ 2.9 พร้อมพารามิเตอร์ D-H ที่เกี่ยวข้องใน



รูปที่ 2.9 แขนกลที่มีสามข้อต่อพร้อมเฟรมที่กำหนดให้แต่ละท่อนแขน

ตารางที่ 2.3 พารามิเตอร์ D-H ของแขนกล

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	θ_3

ด้วยเฟรมที่กำหนดกับแต่ละท่อนแขนดังแสดงในรูปที่ 2.9 และพารามิเตอร์ D-H ที่เกี่ยวข้องในตารางที่ 2.2 สามารถเขียนเมทริกซ์การแปลงจากพิกัดในเฟรม 3 ไปยังพิกัดในเฟรม 0 ได้คือ

$$T_{0,3} = \begin{bmatrix} \cos(q_1 + q_2 + q_3) & -\sin(q_1 + q_2 + q_3) & 0 & l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ \sin(q_1 + q_2 + q_3) & \cos(q_1 + q_2 + q_3) & 0 & l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

แต่สำหรับการคำนวณจลนศาสตร์ผกผัน จะรับข้อมูลตำแหน่งและทิศทางของเฟรม 3 ซึ่งก็คือเมทริกซ์การแปลง

$$T_{0,3} = \begin{bmatrix} \cos f & -\sin f & 0 & x \\ \sin f & \cos f & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

เป้าหมายก็คือต้องหา $\theta_1, \theta_2, \theta_3$ ที่ทำให้เมทริกซ์การแปลงทั้งสองในสมการที่ (2.5) และสมการที่ (2.6) ที่กล่าวมาเท่ากัน ซึ่งนั่นก็คือทำให้

$$\cos f = \cos(q_1 + q_2 + q_3) \quad (2.7)$$

$$\sin f = \sin(q_1 + q_2 + q_3) \quad (2.8)$$

$$x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \quad (2.9)$$

$$y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \quad (2.10)$$

ยกกำลังสองสมการที่ (2.9) และสมการที่ (2.10) แล้วนำมาบวกกันและใช้คุณสมบัติ

$$\cos(q_1 + q_2) = \cos q_1 \cos q_2 - \sin q_1 \sin q_2$$

$$\sin(q_1 + q_2) = \sin q_1 \cos q_2 + \cos q_1 \sin q_2$$

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2\cos q_2$$

ซึ่งก็คือ

$$\cos q_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \quad (2.11)$$

จะหาค่า θ_2 ได้ก็ต่อเมื่อค่าของพจน์ทางขวาของสมการที่ (2.11) มีค่าระหว่าง -1 และ 1 ซึ่งค่าดังกล่าวอยู่นอกช่วงก็แสดงว่าแขนกลนี้ สามารถไปยังตำแหน่งและทิศทางที่กำหนดให้ หาค่า θ_2 ได้จากการหาค่า

$$\sin q_2 = \pm \sqrt{1 - \cos^2 q_2} \quad (2.12)$$

และใช้ค่า $\sin \theta_2$ ที่ได้ในสูตร

$$q_2 = \text{atan2}(\sin q_2, \cos q_2) \quad (2.13)$$

เครื่องหมายบวกและลบที่เลือกได้ในสมการที่ (2.12) แสดงถึงการมีพารามิเตอร์คำตอบมากกว่าหนึ่ง ซึ่งเมื่อได้ค่า θ_2 ก็สามารถหาค่า θ_1 ได้จากการแทนค่า θ_2 ที่ได้ลงในสมการที่ (2.9) และสมการที่ (2.10) แล้วจัดรูปด้วยคุณสมบัติทางตรีโกณมิติ ผลที่ได้คือ

$$x = k_1 \cos q_1 - k_2 \sin q_1 \quad (2.14)$$

$$y = k_1 \sin q_1 - k_2 \cos q_1 \quad (2.15)$$

โดยค่าคงตัว k_1, k_2 หาได้จาก $k_1 = l_1 + l_2 \cos \theta_2$ และ $k_2 = l_1 \sin \theta_2$ แต่เพื่อความสะดวกในการแก้สมการที่ (2.14) และสมการที่ (2.15) จึงเปลี่ยนวิธีเขียน k_1, k_2 โดยกำหนดให้ $r = \sqrt{k_1^2 + k_2^2}$ และ $\gamma = \text{atan2}(k_2/r, k_1/r)$ ซึ่งนั่นคือ

$$k_1 = r \cos \gamma \quad (2.16)$$

$$k_2 = r \sin \gamma$$

จึงเขียนสมการที่ (2.14) และสมการที่ (2.15) ใหม่ได้เป็น

$$x/r = \cos \gamma \cos q_1 - \sin \gamma \sin q_1 = \cos(\gamma + q_1)$$

$$y/r = \cos \gamma \sin q_1 - \sin \gamma \cos q_1 = \sin(\gamma + q_1)$$

จึงได้ว่า $\gamma + \theta_1 = \text{atan2}(y/r, x/r)$ ซึ่งนั่นคือ $\theta_1 = \text{atan2}(y/r, x/r) - \text{atan2}(k_2/r, k_1/r)$ และเมื่อรู้ค่า θ_1 และ θ_2 แล้ว จึงหาค่าของ θ_3 ได้จากสมการที่ (2.7) และสมการที่ (2.8)

2.3 อัลกอริทึมในการควบคุมแขนกล

สิ่งหนึ่งที่บ่งบอกถึงประสิทธิภาพในการทำงานของแขนกลคือ ความแม่นยำและเที่ยงตรงซึ่งการเคลื่อนไหวยางงานของแขนกลจะประกอบด้วยเคลื่อนไหวยางฐานไหล่ ข้อศอก ข้อมือ เป็นต้น การเคลื่อนไหวยางส่วนต่างๆ นี้จะส่งผลให้มือเคลื่อนที่ไปตามตำแหน่งต่างๆ ที่ปฏิบัติงาน ฉะนั้นถ้าส่วนต่างๆ นี้มีการควบคุมที่ไม่ดีพอจะส่งผลถึงความแม่นยำและความเที่ยงตรงในการทำงานนั้น ดังนั้นมือของแขนกลจะเคลื่อนที่ไปตามตำแหน่งใดๆ ได้นั้น จะขึ้นอยู่กับข้อต่อต่างๆ เคลื่อนไหวยางตำแหน่งที่เหมาะสมและสอดคล้องกันด้วย ซึ่งการควบคุมที่ต้องควบคุมตำแหน่งของข้อต่อต่างๆ จะแสดงด้วยสมการคณิตศาสตร์ สำหรับสมการที่จะกล่าวถึงในที่นี้มี 2 ลักษณะคือ สมการ Forward Kinematics และสมการ Inverse Kinematics ซึ่งสมการทั้ง 2 ลักษณะใช้แสดงความสัมพันธ์ของข้อต่อต่างๆ และตำแหน่งปลายมือโดยรายละเอียดจะกล่าวในส่วนต่อไป

ก่อนจะกล่าวถึงสมการของแขนกลในแขนกลคือ ดีกรีออฟฟรียดอม (Degree of Freedom) หรือ DOF หมายถึงจำนวนข้อต่อหรือจุดต่อที่เคลื่อนที่เป็นอิสระต่อกันส่วนอีกคำคือ Link หมายถึงสิ่งที่เชื่อมต่อระหว่างข้อต่อกับข้อต่อ ซึ่งข้อต่อนั้นอาจจะเป็นการเคลื่อนที่แบบ R หรือแบบ P ก็ได้สมการ Inverse Kinematics

ความหมายของสมการ Inverse Kinematics จะมีความหมายที่ตรงข้ามกับสมการ Forward Kinematics คือ จากเดิมป้อนค่าของมุมที่ข้อต่อต่างๆ เข้าไปในผลลัพธ์ที่ได้จะเป็นตำแหน่ง X, Y, Z แต่สมการ Inverse Kinematics นี้จะเป็นการป้อนตำแหน่ง X, Y, Z ของปลายมือไปแล้วจะได้ผลลัพธ์เป็นมุมของแต่ละข้อต่อว่าต้องเคลื่อนที่ไปกี่องศาจากตำแหน่ง Home ปลายมือจึงจะไปอยู่ตำแหน่งที่ต้องการ จะแสดงการทำงานดังบล็อกไดอะแกรมการทำงานในรูปที่ 2.10



รูปที่ 2.10 บล็อกไดอะแกรม Inverse Kinematics

จากรูปที่ 2.10 โดยค่าพารามิเตอร์ต่างๆ ยังคงใช้ค่าเหมือนเดิมทุกประการจากความหมายของสมการ Inverse Kinematics เมื่อป้อนตำแหน่ง X, Y, Z ที่ปลายมือ และค่าพารามิเตอร์ที่เกี่ยวข้องผลลัพธ์จะได้เป็นมุมของแกนต่างๆ คือ X, Y, Z โดยสามารถเขียนเป็นอัลกอริทึมของสมการ Inverse Kinematics ได้ดังรูปที่ 2.10

สำหรับการประยุกต์ใช้งานอัลกอริทึมของสมการ Inverse Kinematics อันดับแรกที่เราเห็นได้ชัดคือ ควบคุมการจับวัตถุที่ตำแหน่ง X, Y, Z ต้นทางให้นำไปวางที่ตำแหน่ง X, Y, Z ปลายทางโดยคอมพิวเตอร์จะเป็นตัวคำนวณค่าของมุมที่ข้อต่อต่างๆ เมื่อคำนวณเสร็จคอมพิวเตอร์จะส่งค่ามุมต่างๆ ไปให้กับชุดขับเคลื่อนของชุดขับเคลื่อน ก็จะทำการขับเคลื่อนข้อต่อต่างๆ ไปตามมุมที่ได้มาจากอัลกอริทึมของสมการ Inverse Kinematics ผลลัพธ์ที่ได้ปลายมือจะอยู่ในตำแหน่งที่ต้องการ จากนั้นก็เป็นหน้าที่ของผู้เขียนโปรแกรมควบคุมว่าจะให้ทำอย่างไรต่อไป

อัลกอริทึมของสมการ Inverse Kinematics เป็นเพียงส่วนที่ช่วยคำนวณค่ามุมที่แต่ละข้อต่อเคลื่อนไปเพื่อให้มืออยู่ในตำแหน่งที่ต้องการ ส่วนที่สองของการประยุกต์ใช้งานของอัลกอริทึมของสมการ Inverse Kinematics โดยการใช้ควบคุมกับกล้องในการประมวลผลภาพกล้องจะทำหน้าที่ประมวลสัญญาณภาพที่รับเข้ามาเปลี่ยนให้เป็นตำแหน่ง X, Y, Z ของวัตถุแล้วส่งค่า X, Y, Z ไปให้กับอัลกอริทึมของสมการ Inverse Kinematics ซึ่งจะช่วยให้แขนกลสามารถทำงานได้เหมือนกับมีดวงตาในการมองเห็นวัตถุ

2.4 ไมโครคอนโทรลเลอร์

2.4.1 Mini Servo High Torque HS-225BB



รูปที่ 2.11 Mini Servo High Torque HS-225BB

Servo Motor เป็นมอเตอร์ที่มีหลักการทำงานโดยการส่งของสัญญาณพัลส์มีผลต่อการหมุนของ Servo ทิศทางการหมุนขึ้นอยู่กับความกว้างของพัลส์บวก ในส่วนของ Servo จะใช้ Servo หลายรุ่นด้วยกันทั้ง Servo High Torque 180 องศา ซึ่งจะทำงานในส่วนต่างๆ ของหุ่นยนต์ โดยการเคลื่อนที่พื้นฐานส่วนที่สำคัญคือ ขาใช้ Servo High Torque 180 องศา ทั้งหมด 18 ตัว ตามข้อขยับต่างๆ

ตารางที่ 2.4 Performance Specifications

Performance Specifications	
Operating Voltage Range (Volts DC)	4.8V ~ 6.0V
Speed (Second @ 60°)	0.14 ~ 0.11
Maximum Torque Range oz/in	54 ~ 67
Maximum Torque Range kg/cm	3.9 ~ 4.8
Current Draw at Idle	8 mA
No Load Operating Current Draw	300 mA
Stall Current Draw	1,200 mA
Dead Band Width	8 μ s

ตารางที่ 2.5 Physical Specifications

Physical Specifications	
Dimensions (Inches)	1.27 x 0.66 x 1.22
Dimensions (Metric)	32.4 x 16.8 x 31.0
Weight (Ounces)	0.95
Weight (Gram)	27.0
Circuit Type	HT7003 Analog SMT
Motor Type	3 Pole Metal Brush Ferrite
Gear Material	Nylon
Bearing Type	Top Ball Bearing
Output Shaft (type/ \varnothing mm)	Standard 24
Case Material	Physical Specifications
Dust/Water Resistance	N/A

2.4.2 Driver Module 16-Channel PWM/Servo



รูปที่ 2.12 Driver Module 16-Channel PWM/Servo

โมดูลขับ Servo 16 ช่อง 12 Bit แบบ PWM/Servo ติดต่อบนอินเตอร์เฟส 12C ใช้ในการควบคุม Servo หลายๆ ตัวพร้อมกัน

2.4.3 Bluetooth



รูปที่ 2.13 Bluetooth HC05

Bluetooth Module รุ่น HC05 เป็นโมดูล Bluetooth ซึ่งเป็นรุ่นที่รองรับกับอุปกรณ์ส่วนใหญ่ในปัจจุบัน นำมาใช้ในการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ที่หุ่นยนต์กับคอมพิวเตอร์

2.4.4 Power Supply



รูปที่ 2.14 Power Supply

Power Supply คือ แหล่งจ่ายไฟให้กับอุปกรณ์ไฟฟ้าต่างๆ โดยทำหน้าที่แปลงแรงดันไฟฟ้ากระแสสลับ (AC) เป็นแรงดันไฟฟ้ากระแสตรง (DC) สามารถปรับขนาดกระแสไฟฟ้า (I) และแรงดันไฟฟ้า (V) เพื่อให้เหมาะสมกับอุปกรณ์ที่นำมาใช้งาน

2.4.5 Arduino Uno R3



รูปที่ 2.15 Arduino Uno R3

Arduino อ่านว่า (อา-ดู-อิ-โน้ หรือ อาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

บอร์ด Arduino ถือว่าเป็นบอร์ดไมโครคอนโทรลเลอร์ยอดนิยม และใช้กันอย่างแพร่หลาย ที่สามารถนำไปพัฒนาโครงการได้หลากหลาย เรียนรู้ได้ง่ายและเหมาะสำหรับผู้เริ่มต้น อีกทั้งยังมีราคาที่ไม่แพง

รายละเอียดบอร์ด Arduino UNO R3

1. เชื่อมต่อกับคอมพิวเตอร์ได้โดยตรงผ่าน USB
2. สามารถเขียนและโปรแกรมตัวบอร์ดด้วย Arduino IDE
3. ตัวบอร์ดใช้ชิป AT Mega 328 P ประมวลผลไวขึ้น เป็นชิปตัวใหม่อัปเดตจากรุ่นก่อน
4. ตัวบอร์ดใช้ไฟเลี้ยง 5V

2.5 โปรแกรมใช้งาน

2.5.1 Arduino IDE

เป็นโปรแกรมที่ใช้ทำงานควบคู่กับบอร์ดไมโครคอนโทรลเลอร์ที่ชื่อว่า Arduino ใช้ภาษา C ในการสั่งงานทำให้ง่ายต่อการพัฒนา และยังมี Library พื้นฐานให้เรียนรู้ ซึ่งสามารถนำไปต่อยอดได้

2.5.2 MATLAB

เป็นภาษาเชิงตอบโต้ระดับสูงที่มาพร้อมด้วยสภาพแวดล้อมเชิงโต้ตอบ ซึ่งสามารถคำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็วมากกว่าภาษาคอมพิวเตอร์อื่นๆ เช่น ภาษา C, ภาษา C++ และยังสามารถแสดงผลกราฟฟิก เขียนแอปพลิเคชัน ทำให้สามารถคำนวณผลลัพธ์ พัฒนาอัลกอริทึมสร้างแบบจำลองได้ง่ายและรวดเร็วมาก ซึ่งนำโปรแกรม MATLAB มาใช้ทำงานควบคู่กับโปรแกรม Arduino

2.5.3 SolidWorks

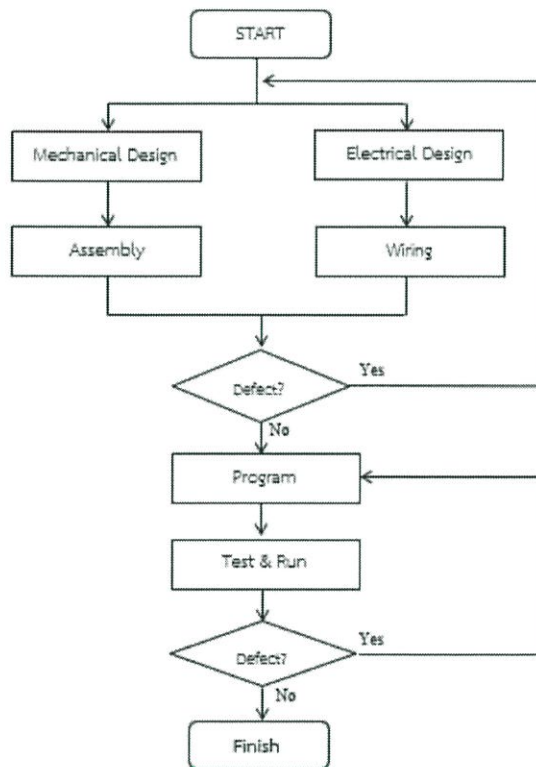
เป็นโปรแกรมซอฟต์แวร์สำหรับการออกแบบที่สามารถจำลองทางด้าน Mechanical Engineering และยังสามารถนำไปใช้คำนวณในทางวิศวกรรม ซึ่งนำมาใช้งานในการออกแบบชิ้นส่วนต่างๆ ของหุ่นยนต์แมงมุมเกือบทุกชิ้นส่วน และนำชิ้นส่วนไปพิมพ์ 3 มิติต่อไป

บทที่ 3

การออกแบบและแนวคิดการเขียนโปรแกรม

3.1 ขั้นตอนการดำเนินงาน

ในการจัดทำโครงงานนี้จะมีการแบ่งขั้นตอนในการดำเนินงานซึ่งสามารถเขียนเป็นแผนภาพได้ดังรูปที่ 3.1



รูปที่ 3.1 แผนผังขั้นตอนการดำเนินงาน

3.2 หลักการออกแบบหุ่นยนต์

เริ่มแรกค้นคว้าข้อมูลตามอินเทอร์เน็ต ตามวิชาเรียนที่ได้เรียนมา และสอบถามอาจารย์ที่ปรึกษา แล้วจึงนำมาออกแบบในโปรแกรม SolidWorks ที่ละชิ้นส่วนของหุ่นยนต์ โดยเริ่มจากส่วนขาให้มีการเคลื่อนที่ได้อิสระ ซึ่งทั้ง 6 ขาจะออกแบบให้เหมือนกันทั้งหมด และแบบการวางอุปกรณ์อิเล็กทรอนิกส์ภายใน

3.2.1 อุปกรณ์ที่ใช้สำหรับสร้างชิ้นส่วนประกอบหุ่นยนต์

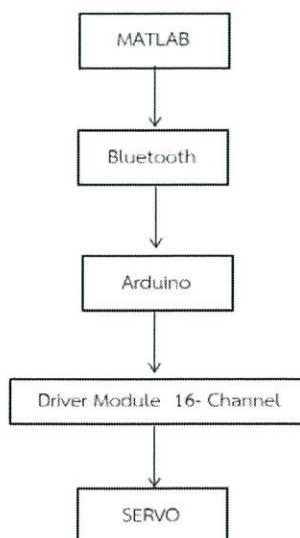
1. พลาสติก PLA Plastic เพื่อผลิตชิ้นส่วนหุ่นยนต์
2. นี้อตตัวผู้และตัวเมียขนาดเส้นผ่าศูนย์กลาง 2 มิลลิเมตร
3. ไมโครคอนโทรลเลอร์ Arduino Uno
4. High Torque Servo Motor HS-225BB
5. Module HC-05
6. Driver Module 16-Channel PWM/Servo
7. Power Supply

3.2.2 ข้อคำนึงในการออกแบบ

1. หุ่นยนต์ที่ออกแบบต้องมีน้ำหนักเบา สามารถรับน้ำหนักของอุปกรณ์ทั้งหมดและน้ำหนักของตัวเองได้
2. การออกแบบการวางอุปกรณ์ภายในหุ่นยนต์ควรมีความสมดุลในเรื่องน้ำหนักเพื่อให้ตัวหุ่นมีน้ำหนักสมดุลทุกจุด
3. ควรเลือกใช้วัสดุในการพิมพ์ 3D ให้เหมาะสมกับชิ้นส่วนที่ใช้งาน

3.3 แนวคิดการเขียนโปรแกรม

หลักการเขียนโปรแกรมการควบคุมหุ่นยนต์มีหลักการทำงานใหญ่ๆ หลายส่วนคือ โปรแกรม MATLAB ทำการวิเคราะห์สมการ Kinematics ที่คำนวณไว้ แล้วส่งค่ามุมไปที่โปรแกรม Arduino IDE ผ่านโมดูล Bluetooth ซึ่งโปรแกรมใน Arduino IDE จะส่งค่ามุมต่างๆ ไปที่ Servo แต่ละตัวต่อไป



รูปที่ 3.2 ลำดับขั้นตอนการทำงานของโปรแกรม

3.3.1 การเขียนโปรแกรม

1. โปรแกรม MATLAB

ใช้โปรแกรม MATLAB ในการคำนวณสมการเชิงคณิตศาสตร์ ซึ่งก็คือสมการ Forward Kinematics และ Inverse Kinematics โดยโปรแกรมแมทแลปจะรับค่าจากการสั่งงานให้หุ่นยนต์เคลื่อนที่ตามที่ออกแบบไว้ แล้วจึงนำมาแปลงเป็นค่ามุมผ่านสมการทางคณิตศาสตร์ และส่งไปให้โปรแกรม Arduino ทำงานต่อไปโดยผ่านโมดูล Bluetooth เป็นตัวส่งข้อมูล

2. โปรแกรม Arduino IDE

ทำหน้าที่รับค่ามุมที่คำนวณได้จากโปรแกรม MATLAB มาสั่งงานไปที่ Servo แต่ละตัว โดยผ่าน Driver Module 16-Channel PWM/Servo เป็นตัวกระจายสัญญาณไปที่ Servo ต่อไป

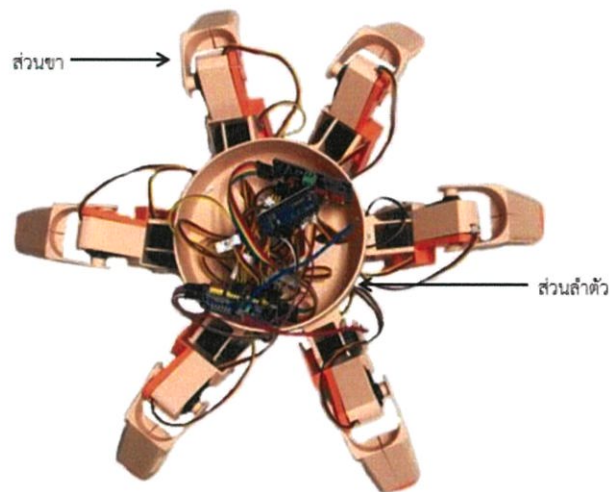
บทที่ 4

ผลการดำเนินงาน

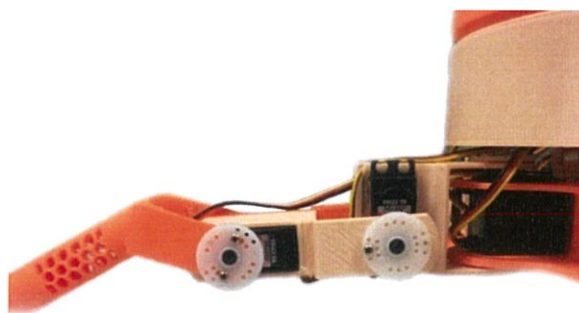
ในส่วนของการดำเนินโครงการนี้ จะกล่าวถึงโครงสร้างหุ่นยนต์ ส่วนประกอบทางไฟฟ้าและสมการการเคลื่อนที่ของหุ่นยนต์

4.1 โครงสร้างหุ่นยนต์

ส่วนประกอบของหุ่นยนต์แมงมุม 6 ขา ถูกดำเนินการประกอบจนเสร็จสิ้น โดยหุ่นยนต์แบ่งออกเป็นหลายส่วน โดยส่วนแรกจะเป็นส่วนของขาหุ่นยนต์ทั้ง 6 ขา ซึ่งเป็นโครงสร้างในการเคลื่อนที่ และในส่วนต่อมาคือ ส่วนลำตัว และการออกแบบจะใช้โปรแกรม SolidWorks ในการออกแบบแล้วนำมาพิมพ์ 3D จนนำมาประกอบและแก้ไขต่อไป



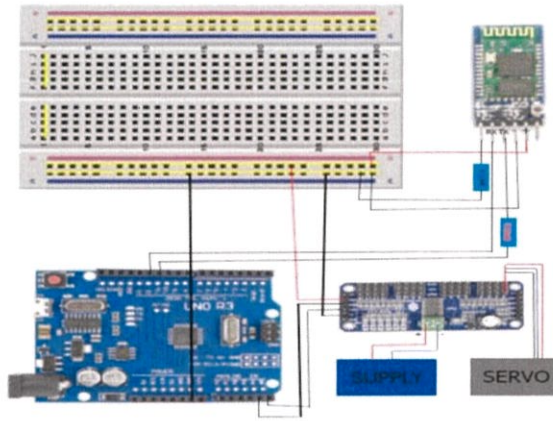
รูปที่ 4.1 โครงสร้างหุ่นยนต์ 6 ขา



รูปที่ 4.2 โครงสร้างส่วนขา

4.2 ส่วนประกอบทางไฟฟ้า

ส่วนของการออกแบบทางไฟฟ้า และการนำอุปกรณ์อิเล็กทรอนิกส์ต่างๆมาต่อวงจรเพื่อนำไปใช้งานได้ดังรูปภาพที่ 4.3



รูปที่ 4.3 การต่อวงจรทางไฟฟ้า

4.3 สมการการเคลื่อนที่ของหุ่นยนต์

จากรูปที่ 4.2 ซึ่งเป็นส่วนที่นำมาคำนวณการเคลื่อนที่ทางคณิตศาสตร์ตามทฤษฎี ซึ่งได้ตั้งสมการด้านล่าง

$$\begin{bmatrix} (c_{q_1})(c_{q_2}) & (c_{q_1})(-s_{q_2}) & s_{q_1} & (a_1 c_{q_1}) + (c_{q_1})(a_2 c_{q_2}) \\ (s_{q_1})(c_{q_2}) & (s_{q_1})(-s_{q_2}) & -c_{q_1} & (a_1 s_{q_1}) + (s_{q_1})(a_2 c_{q_2}) \\ s_{q_2} & c_{q_2} & 0 & a_2 s_{q_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (q_3) & (-s_{q_3}) & 0 & (a_3 c_{q_3}) \\ (q_3) & (c_{q_3}) & 0 & (a_3 s_{q_3}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos q_1 (\cos(q_2 + q_3)) & (c_{q_1})(-s_{q_2}) & s_{q_1} & \cos q_1 (a_1 + a_2 \cos q_2 + a_3 \cos(q_2 + q_3)) \\ -\sin q_1 (\cos(q_2 + q_3)) & (s_{q_1})(-s_{q_2}) & -c_{q_1} & \sin q_1 (a_1 + a_2 \cos q_2 + a_3 \cos(q_2 + q_3)) \\ \sin(q_2 + q_3) & \cos(q_2 + q_3) & 0 & a_2 \sin q_2 + a_3 \sin(q_2 + q_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

โดยที่ $a_1 = 42$, $a_2 = 57.3$, $a_3 = 94$ $r = a_2 \cos q_2 + a_3 \cos(q_2 + q_3)$ จะได้

$$\cos q_1 (42 + 57.3 \cos q_2 + 94 \cos(q_2 + q_3)) = x$$

$$\sin q_1 (42 + 57.3 \cos q_2 + 94 \cos(q_2 + q_3)) = y$$

$$(57.3 \sin q_2 + 94 \sin(q_2 + q_3)) = z$$

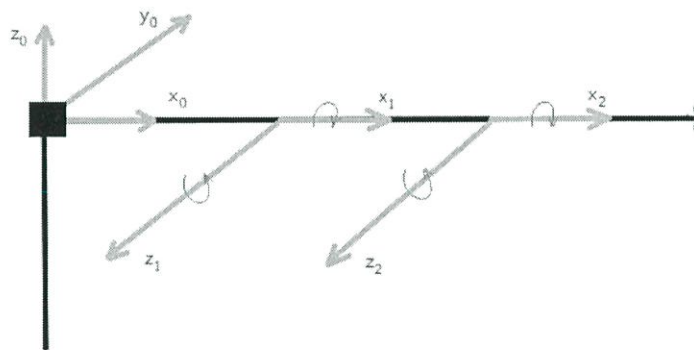
จากสมการที่ (4.1) คำนวณหาค่า $\theta_1, \theta_2, \theta_3$ ได้ดังนี้

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

$$\theta_2 = \sin^{-1} \left[\frac{a_3 \sin \theta_3}{\sqrt{z^2 + r^2}} \right] - \tan^{-1} \frac{z}{r}$$

$$\theta_3 = \cos^{-1} \frac{z^2 + r^2 - a_2^2 - a_3^2}{2a_2a_3}$$

ซึ่งนำค่า $\theta_1, \theta_2, \theta_3, x, y, z$ ไปใช้เป็นตัวแปรในการเขียนโปรแกรมต่อไป



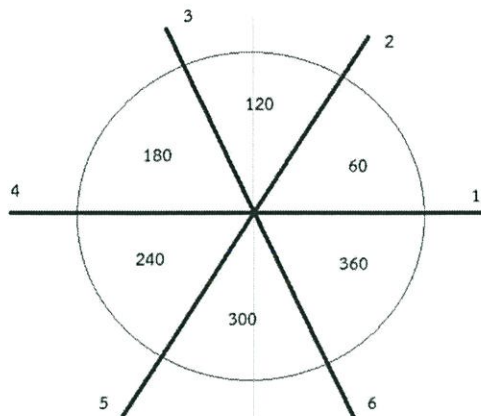
รูปที่ 4.4 ภาพอย่างง่ายของโครงสร้างส่วนขา

จากรูปที่ 4.4 นำมาเขียนตารางได้ดังตารางที่ 4.1

ตารางที่ 4.1 พารามิเตอร์ของแขนกล

L	q	d	a	α	Home
Link1	q_1	0	42.0	$\pi/2$	0
Link2	q_2	0	57.3	0	0
Link3	q_3	0	74.0	0	0

และ



รูปที่ 4.5 ภาพอย่างง่ายของโครงสร้างหุ่นยนต์ 6 ขา

ซึ่งนำมาเขียนสมการเชิงคณิตศาสตร์ของขาหุ่นยนต์ที่จุด Home ของแต่ละขาตามหมายเลขในรูปที่ 4.3 เพื่อนำไปใช้ในการเขียนโปรแกรมต่อไป จะได้ดังสมการด้านล่าง

ขาที่ 1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

ขาที่ 2

$$\begin{bmatrix} 0.5 & -0.866 & 0 & 0 \\ 0.866 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & -0.866 & 0 & 25 \\ 0.866 & 0.5 & 0 & 43.3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

ขาที่ 3

$$\begin{bmatrix} -0.5 & -0.866 & 0 & 0 \\ 0.866 & -0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.5 & -0.866 & 0 & -25 \\ 0.866 & -0.5 & 0 & 43.3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

ขาที่ 4

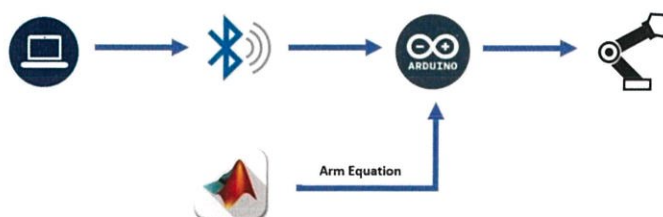
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

ขาที่ 5

$$\begin{bmatrix} -0.5 & 0.866 & 0 & 0 \\ 0.866 & -0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.5 & -0.866 & 0 & -25 \\ 0.866 & -0.5 & 0 & -43.3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

ขาที่ 6

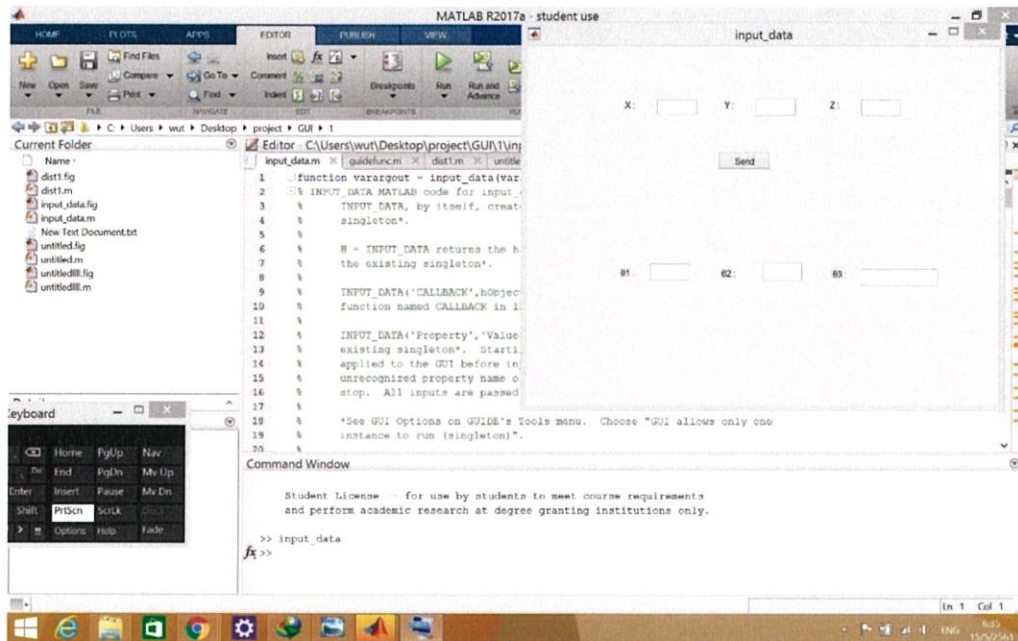
$$\begin{bmatrix} 0.5 & -0.866 & 0 & 0 \\ 0.866 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.5 & -0.866 & 0 & 25 \\ 0.866 & -0.5 & 0 & 43.3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$



รูปที่ 4.6 รูปแสดงการส่งค่าผ่านบลูทูธ

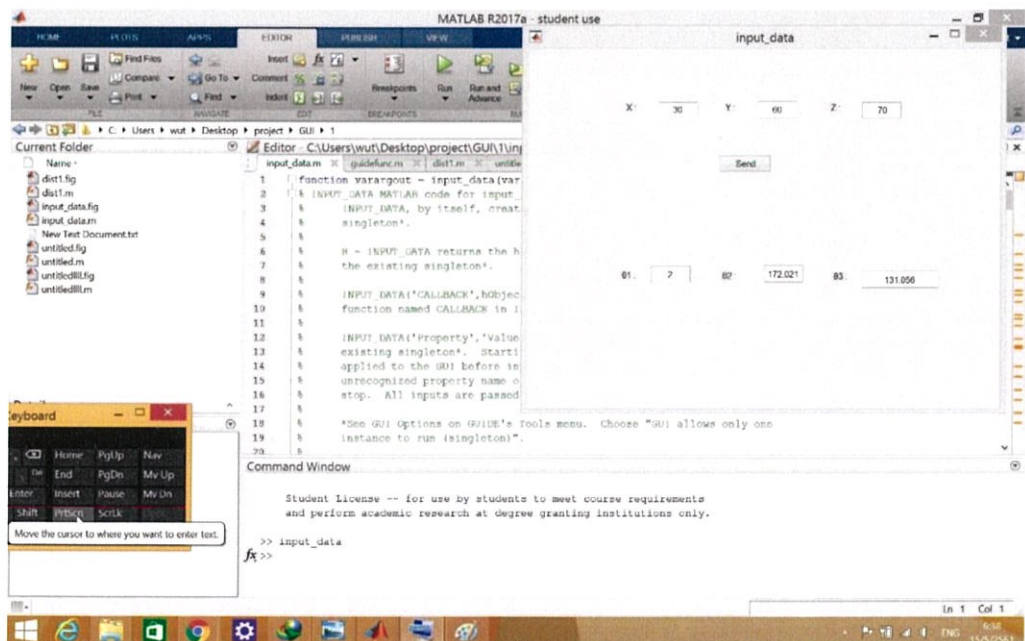
จากรูปที่ 4.6 เป็นการแสดงลำดับการส่งค่ามุมที่คำนวณได้จากสมการแขนกล โดยค่าที่คำนวณได้จากแมทแลบส่งผ่านบลูทูธไปยัง Arduino เป็นตัวรับค่าเพื่อสั่งการ Servo ทั้ง 16 ตัว

4.4 การทำงานโปรแกรม

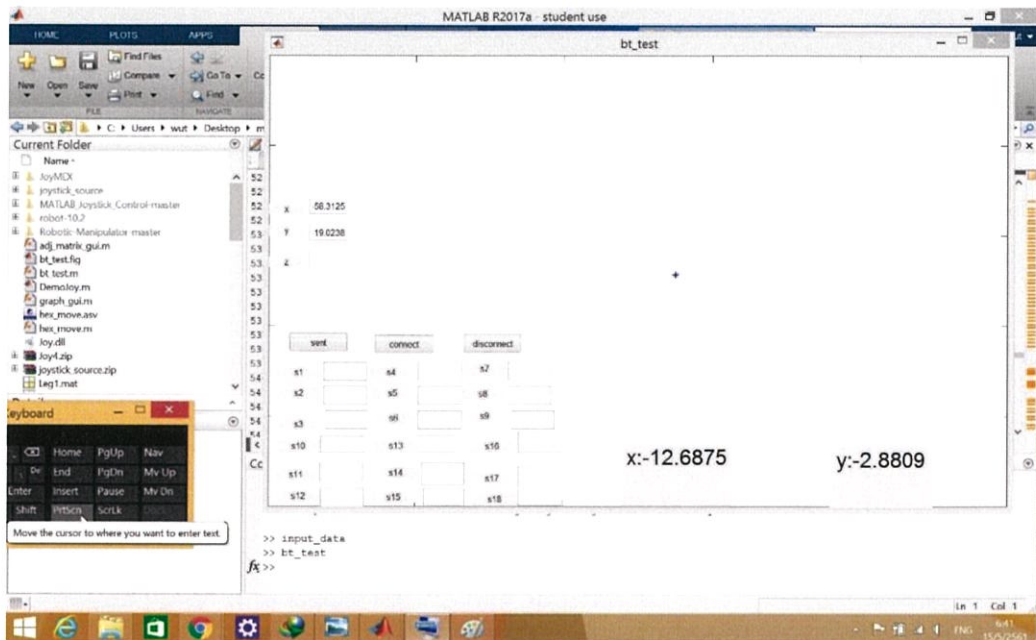


รูปที่ 4.7 รูปแสดงหน้าต่างส่งข้อมูล สำหรับขา 1 ข้าง

จากรูปที่ 4.7 เมื่อป้อนค่า x , y , z โปรแกรมจะทำการคำนวณตามสมการแขนกลได้เป็นค่ามุมทั้ง 3 ออกมา

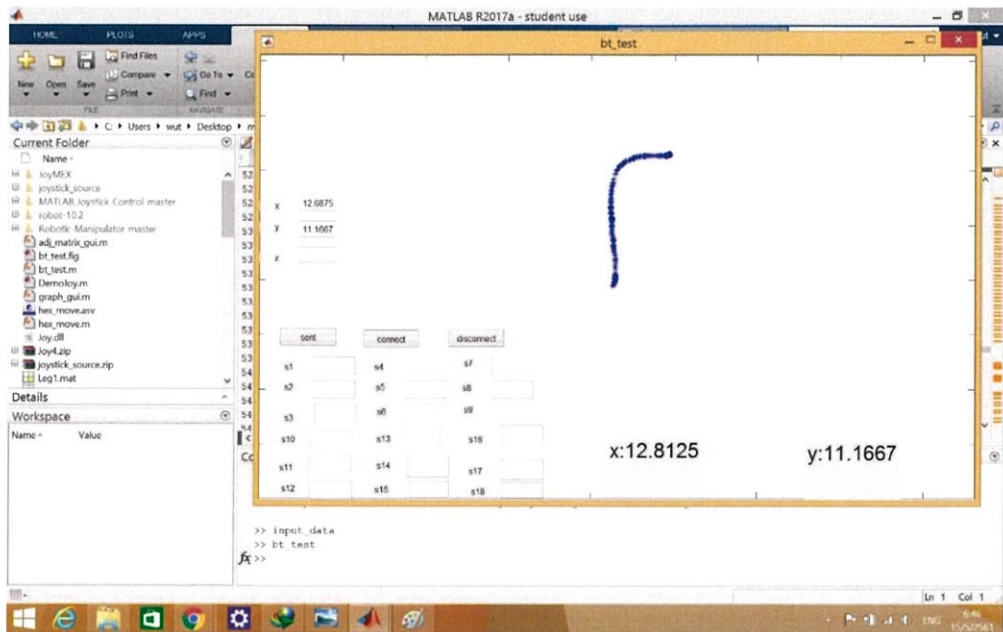


รูปที่ 4.8 ผลที่ได้จากการคำนวณ x , y , z



รูปที่ 4.9 หน้าจอแสดงการทำงาน

รูปที่ 4.9 เป็นรูปที่แสดงหน้าต่างโปรแกรมที่จุด Home โดยหน้าจอจะบอกค่าตำแหน่ง x, y และมีปุ่มรีเซ็ตค่า x, y ให้เป็น 0 เพื่อให้หุ่นเข้าสู่จุด Home การใช้งานโปรแกรมนี้มีหน้าที่กำหนดเส้นทางของหุ่นโดยการลากเส้นค่า x, y จะเปลี่ยนไปตามทิศทางที่ลากเส้น

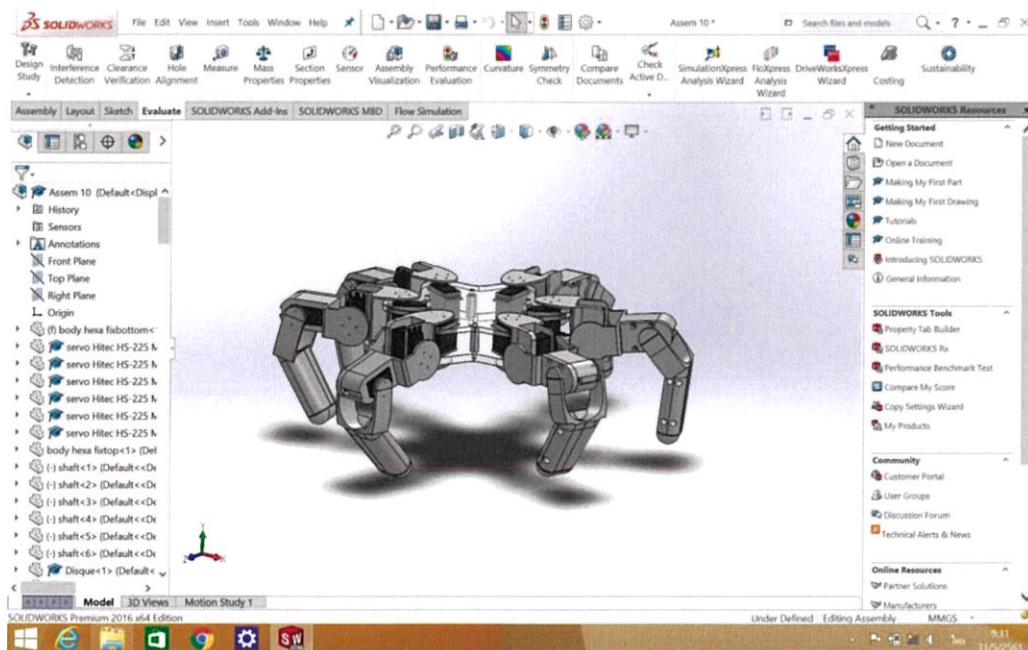


รูปที่ 4.10 โปรแกรมขณะทำงาน

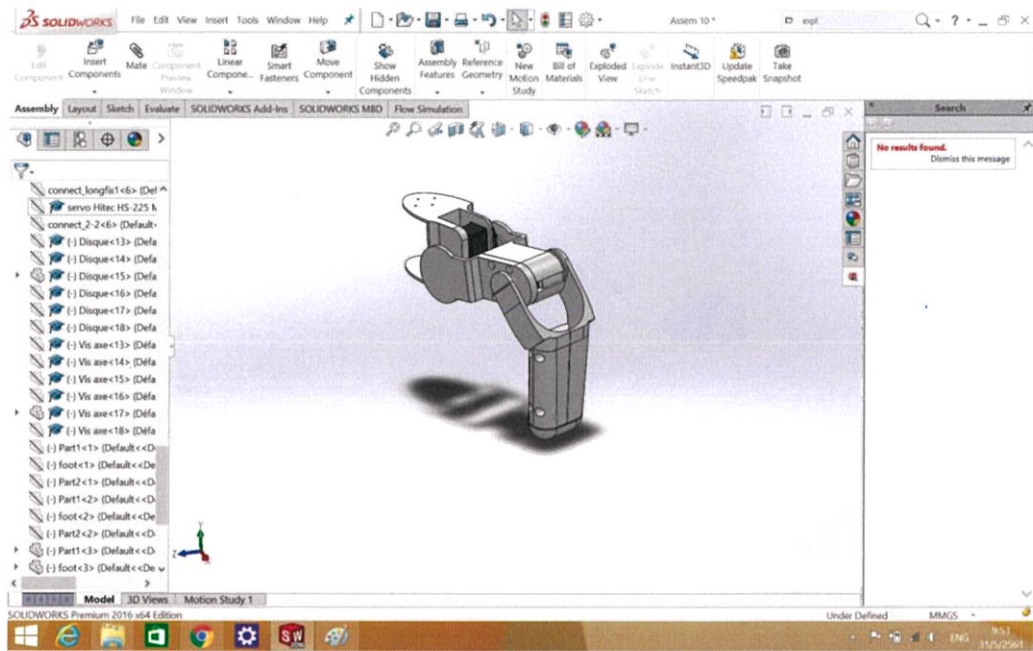
จากรูปที่ 4.10 จะเห็นว่าเมื่อทำการลากเส้นค่า x y จะเปลี่ยนแปลงตลอดทำให้ค่ามุมที่คำนวณเปลี่ยนแปลงและส่งเป็น ค่ามุมของ Servo ทั้ง 18 ตัว

4.5 การออกแบบ

ออกแบบโครงสร้างส่วนต่างๆ ด้วยโปรแกรม SolidWorks



รูปที่ 4.11 แบบโครงสร้างหุ่นยนต์ 6 ขา



รูปที่ 4.12 แบบโครงสร้างส่วนขาของหุ่นยนต์ 6 ขา

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการออกแบบและการเขียนโปรแกรมหุ่นยนต์ 6 ขา ซึ่งใช้ในการศึกษาการเคลื่อนที่ทางคณิตศาสตร์ของแขนกลทั้ง 6 แขนซึ่งก็คือ ขาของหุ่นยนต์ให้สามารถทำงานพร้อมกันอย่างเป็นระบบ

จากการทำโครงงานหุ่นยนต์ 6 ขา ตามที่ได้ออกแบบไว้ซึ่งออกแบบให้มีน้ำหนักเบา และออกแบบการวางตำแหน่งของ Servo ในขาแต่ละขาให้มีแรงกระทำจากน้ำหนักหุ่นน้อยที่สุด ต่อมาจึงออกแบบสมการการเดินของหุ่นยนต์ตามทฤษฎี Kinematics แล้วนำสมการไปใส่ในโปรแกรมเพื่อให้หุ่นยนต์ทำงานตามสมการ ผลปรากฏว่าหุ่นยนต์สามารถทำงานตามที่ออกแบบไว้ได้

ตารางที่ 5.1 แสดงผลการทดลองส่งค่าผ่านบลูทูธ

ครั้งที่	ผลการส่งข้อมูลผ่านบลูทูธ
1	ส่งสำเร็จ
2	ส่งสำเร็จ
3	ส่งไม่สำเร็จ
4	ส่งสำเร็จ
5	ส่งไม่สำเร็จ
6	ส่งไม่สำเร็จ
7	ส่งสำเร็จ
8	ส่งสำเร็จ
9	ส่งสำเร็จ
10	ส่งสำเร็จ

จากตารางที่ 5.1 จะพบว่าจากการทดลองส่งค่าผ่านบลูทูธทั้งหมด 10 ครั้ง พบว่าส่งค่าผ่านสำเร็จ 7 ครั้ง ส่งไม่สำเร็จ 3 ครั้ง เกิดจากสัญญาณที่เชื่อมต่อบลูทูธไม่เสถียรและสายสัญญาณหลุดจากพอร์ท ทำให้ค่าที่คำนวณได้ส่งไม่สำเร็จ

5.2 ปัญหาและแนวทางแก้ไขปัญหา

1. การออกแบบของหุ่นยนต์ในครั้งแรกพบว่ามิน้ำหนักมาก ทำให้ Servo ต้องรับภาระแรงที่มากระทำสูง ทำให้ต้องออกแบบชิ้นส่วนที่ขาให้มีน้ำหนักเบาลง
2. การส่งข้อมูลผ่านบลูทูธเกิดปัญหาเนื่องจากการเชื่อมต่อบลูทูธ ไม่เสถียรทำให้การส่งค่ามุมจาก MATLAB มายัง Controller ส่งค่าไม่สำเร็จ

5.3 ข้อเสนอแนะ

1. การวางแผนการดำเนินงานมีความสำคัญต่อการทำงานเป็นอย่างมาก ดังนั้นจะต้องมีการวางแผนดำเนินงานโดยเผื่อระยะเวลาไว้เนื่องจากอาจเกิดปัญหาอื่นๆ ที่อาจทำให้ใช้ระยะเวลาดำเนินงานที่นานขึ้น
2. ควรศึกษา ค้นคว้าข้อมูลต่างๆ ที่เกี่ยวกับงานที่ต้องทำให้มีความเข้าใจก่อนลงมือทำจริง เพื่อให้ไม่เกิดความผิดพลาดค่าใช้จ่ายที่เกินจากที่ประเมินไว้
3. ควรสื่อสารและแบ่งงานให้ชัดเจนระหว่างเพื่อนร่วมงาน
4. ก่อนลงมือทำสิ่งสำคัญควรปรึกษาอาจารย์ที่ปรึกษาก่อน

เอกสารอ้างอิง

- [1] “Arduino Libraries” (Online). Available
<https://www.arduino.cc/en/Reference/Libraries>
- [2] “MathWorks” (Online). Available
<https://www.mathworks.com/products/matlab.html>
- [3] “ตัวอย่างการออกแบบ” (Online). Available
<https://grabcad.com/>
- [4] “kinematic robotic arm mat work” (Online). Available
<https://www.mathworks.com/help/fuzzy/examples/modeling-inverse-kinematics-in-a-robotic-arm.html>
- [5] “inverse-kinematics” (Online). Available
<https://robotacademy.net.au/lesson/inverse-kinematics-for-a-2-joint-robot-arm-using-algebra/>
- [6] “ทฤษฎีแขนกล” (online). Available
<file:///C:/Users/MSI-GE62.7RE/Desktop/mani.p>

ภาคผนวก

ภาคผนวก

Code การทำงานของหุ่นยนต์ 6 ขา

1. การเชื่อมต่อ บลูทูธ ใน MATLAB

```
function varargout = bt_test(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @bt_test_OpeningFcn, ...
                  'gui_OutputFcn', @bt_test_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function bt_test_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
clear bt;
clc;
global bt;
bt = Bluetooth('HC-05', 1);
fopen(bt)
function varargout = bt_test_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function servo_1_Callback(hObject, eventdata, handles)
function servo_1_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function servo_4_Callback(hObject, eventdata, handles)
function servo_4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function servo_7_Callback(hObject, eventdata, handles)
function servo_7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function servo_10_Callback(hObject, eventdata, handles)
function servo_10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function servo_2_Callback(hObject, eventdata, handles)
function servo_2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function servo_5_Callback(hObject, eventdata, handles)
function servo_5_CreateFcn(hObject, eventdata, handles)
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function servo_8_Callback(hObject, eventdata, handles)
function servo_8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function servo_11_Callback(hObject, eventdata, handles)
function servo_11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function servo_3_Callback(hObject, eventdata, handles)
function servo_3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function servo_6_Callback(hObject, eventdata, handles)
function servo_6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function servo_9_Callback(hObject, eventdata, handles)
function servo_9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```
        set(hObject,'BackgroundColor','white');
    end
function servo_12_Callback(hObject, eventdata, handles)
function servo_12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function servo_13_Callback(hObject, eventdata, handles)
function servo_13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function servo_14_Callback(hObject, eventdata, handles)
```

```
function servo_14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function servo_15_Callback(hObject, eventdata, handles)
function servo_15_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function servo_16_Callback(hObject, eventdata, handles)
function servo_16_CreateFcn(hObject, eventdata, handles)
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function servo_17_Callback(hObject, eventdata, handles)
function servo_17_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function servo_18_Callback(hObject, eventdata, handles)

function servo_18_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function pushbutton1_Callback(hObject, eventdata, handles)
a = getappdata(0, 'servo_1');
b = getappdata(0, 'servo_2');
c = getappdata(0, 'servo_3');
d = getappdata(0, 'servo_4');
e = getappdata(0, 'servo_5');
f = getappdata(0, 'servo_6');
g = getappdata(0, 'servo_7');
h = getappdata(0, 'servo_8');
i = getappdata(0, 'servo_9');
j = getappdata(0, 'servo_10');
k = getappdata(0, 'servo_11');
l = getappdata(0, 'servo_12');
m = getappdata(0, 'servo_13');
n = getappdata(0, 'servo_14');
o = getappdata(0, 'servo_15');

```

```
p = getappdata(0, 'servo_16');
q = getappdata(0, 'servo_17');
r = getappdata(0, 'servo_18');
global a
global b
global c
global d
global e
global f
global g
global h
global i
global j
global k
global l
global m
global n
global o
global p
global q
global r
```

```
a = str2num(get(handles.servo_1, 'String'));
b = str2num(get(handles.servo_2, 'String'));
c = str2num(get(handles.servo_3, 'String'));
d = str2num(get(handles.servo_4, 'String'));
e = str2num(get(handles.servo_5, 'String'));
f = str2num(get(handles.servo_6, 'String'));
g = str2num(get(handles.servo_7, 'String'));
h = str2num(get(handles.servo_8, 'String'));
i = str2num(get(handles.servo_9, 'String'));
j = str2num(get(handles.servo_10, 'String'));
k = str2num(get(handles.servo_11, 'String'));
```

```

l = str2num(get(handles.servo_12, 'String'));
m = str2num(get(handles.servo_13, 'String'));
n = str2num(get(handles.servo_14, 'String'));
o = str2num(get(handles.servo_15, 'String'));
p = str2num(get(handles.servo_16, 'String'));
q = str2num(get(handles.servo_17, 'String'));
r = str2num(get(handles.servo_18, 'String'));
global bt
fwrite(bt, int16([a b c d e f g h i j k l m n o p q r]))
function pushbutton2_Callback(hObject, eventdata, handles)

```

```

function pushbutton3_Callback(hObject, eventdata, handles)
global bt
fclose(bt)
function figure1_WindowButtonMotionFcn(hObject, eventdata, handles)
pos = get(hObject, 'currentpoint'); % get mouse location on figure
x = pos(1);
y = pos(2); % assign locations to x and y
global xi;
global yi;
set(handles.lbl_x,'string', ['x:' num2str(xi)]);
set(handles.lbl_y,'string', ['y:' num2str(yi)]);
set(gca,'XLim',[0 172.375],'YLim',[0 40.619],'ZLim',[-200 200])
plot(handles.axes2,x,y,'b*');
xi = pos(1) - 86.1875;
set(handles.p1,'String',xi);
yi = pos(2) - 20.3095;
set(handles.p2,'String',yi);
global z1
global z2
global z3
global a
global b

```

```

global c
global d
global e
global f
global g
global h
global i
global j
global k
global l
global m
global n
global o
global p
global q
global r
global z
z1 = getappdata(0, 'p1');
z2 = getappdata(0, 'p2');
z3 = getappdata(0, 'data_z');
z = sqrt((57.3)^2+(94^2)-(2*57.3*94*cosd(180-acosd((z3^2+(z1-42)^2-(57.3)^2-(94)^2)/(2*57.3*94))))));
%a=datax b=datay c=dataz
zeta1 = z2 / z1;
zeta3 = acosd((z3^2+(z1-42)^2-(57.3)^2-(94)^2)/(2*57.3*94));
zeta2 = acosd(z3/94)+asind(94*sind(180-acosd((z3^2+(z1-42)^2-(57.3)^2-(94)^2)/(2*57.3*94)))/z)+asind(57.3*sind(180-acosd((z3^2+(z1-42)^2-(57.3)^2-(94)^2)/(2*57.3*94)))/z)-90;

set(handles.servo_3,'String',zeta1);
set(handles.servo_2,'String',zeta2);
set(handles.servo_1,'String',zeta3);
function figure1_WindowButtonDownFcn(hObject, eventdata, handles)

```

```
hold(handles.axes2,'on');
function p1_Callback(hObject, eventdata, handles)
function p1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function p2_Callback(hObject, eventdata, handles)
function p2_CreateFcn(hObject, eventdata, handles)
.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function figure1_WindowButtonUpFcn(hObject, eventdata, handles)
hold(handles.axes2,'off');
function axes2_CreateFcn(hObject, eventdata, handles)
set(gca,'XLim',[0 172.375],'YLim',[0 40.619],'ZLim',[-200 200])
function data_z_Callback(hObject, eventdata, handles)

function data_z_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

2. การแปลงค่ามุม

```

function varargout = input_data(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @input_data_OpeningFcn, ...
                  'gui_OutputFcn', @input_data_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    ] varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function input_data_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = input_data_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function data_x_Callback(hObject, eventdata, handles)
x = str2num(get(handles.data_x, 'String'));
setappdata(0, 'data_x', x);
function data_x_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function data_y_Callback(hObject, eventdata, handles)
y = str2num(get(handles.data_y, 'String'));
setappdata(0, 'data_y', y);

```

```
function data_y_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function data_z_Callback(hObject, eventdata, handles)
z = str2num(get(handles.data_z, 'String'));
setappdata(0, 'data_z', z);

function data_z_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function send_Callback(hObject, eventdata, handles)
% hObject    handle to send (see GCBO)
```

3. การกำหนดตัวแปร และสร้างสมการ

```

a = getappdata(0, 'data_x');
b = getappdata(0, 'data_y');
c = getappdata(0, 'data_z');
l = getappdata(0, 'zeta1');
m = getappdata(0, 'zeta2');
n = getappdata(0, 'zeta3');
z = sqrt((57.3)^2+(94^2)-(2*57.3*94*cosd(180-(acosd((c^2+(a-42)^2-(57.3)^2-(94)^2)/(2*57.3*94))))));
zeta 1= b / a;
zeta 3= acosd((c^2+(a-42)^2-(57.3)^2-(94)^2)/(2*57.3*94));
zeta 2= acosd(c/94)+asind(94*sind(180-acosd((c^2+(a-42)^2-(57.3)^2-(94)^2)/(2*57.3*94)))/z)+asind(57.3*sind(180-acosd((c^2+(a-42)^2-(57.3)^2-(94)^2)/(2*57.3*94)))/z)-90;
zeta 1= num2str(zeta1);
set(handles.zeta1,'String',zeta1);
set(handles.zeta3,'String',zeta3);
set(handles.zeta2,'String',zeta2);

```