



การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม

นายเกียรติสุริยา บุราเลข

นายพัชรพงศ์ ลำตัน

รหัสนักศึกษา 623040214-5

รหัสนักศึกษา 623040293-3

รายงานนี้เป็นรายงานโครงการของนักศึกษาซึ่งเสนอเป็นส่วนหนึ่งในหลักสูตร
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมระบบอิเล็กทรอนิกส์)

สาขาวิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
ปีการศึกษา 2565



Design of an Industrial Robot Arm

Mr. Kiadsuriya Buralek

Student ID 623040214-5

Mr. Patcharapong Lumton

Student ID 623040293-3

This is the report of the students' project assignment submitted in partial fulfillment of the requirement for the degree of bachelor of engineering
(Electronic Systems Engineering)

Department of Electrical Engineering
Faculty of Engineering, Khon Kaen University
Academic Year 2022

ใบประเมินผลงาน

ชื่อเรื่องภาษาไทย การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม

ชื่อเรื่องภาษาอังกฤษ Design of an Industrial Robot Arm

ผู้จัดทำ

นายเกียรติสุริยา บุราเลข

รหัสนักศึกษา 623040214-5

นายพัชรพงศ์ ลำตัน

รหัสนักศึกษา 623040293-3

อาจารย์ที่ปรึกษา

(รศ.ดร.ประมินทร์ อัจฉฤทธิ์)

อาจารย์ผู้ร่วมประเมิน

(ผศ.ดร.จงกฤษณ์ จงอุดมการณ์)

(อ.สกริพร พรนิมิตร)

ประเมินผล ณ วันที่ 4 เมษายน 2566

กิตติกรรมประกาศ

คณะผู้จัดทำโครงการขอขอบพระคุณ รศ.ดร. ประมินทร์ อัจฉฤทธิ์ อาจารย์ที่ปรึกษาโครงการอย่างสูงที่ให้คำปรึกษา คำแนะนำในการทำโครงการ ทำให้โครงการสำเร็จลุล่วงไปได้ด้วยดีและขอขอบคุณคณาจารย์ภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่คอยช่วยเหลือและให้คำแนะนำตลอดระยะเวลาในการทำโครงการในครั้งนี้

ขอขอบคุณคณะวิศวกรรมศาสตร์ ภาควิชาไฟฟ้า มหาวิทยาลัยขอนแก่น ที่อำนวยความสะดวกในการใช้งานเครื่องมือ อุปกรณ์ และสถานที่สำหรับการทำโครงการ

สุดท้ายนี้ คณะผู้จัดทำหวังว่าโครงการฉบับนี้จะเป็นประโยชน์แก่ผู้ที่สนใจศึกษาไม่มากก็น้อย หากมีข้อผิดพลาดประการใด คณะผู้จัดทำขออภัย ณ ที่นี้ด้วยและยินดีน้อมรับฟังคำแนะนำจากผู้ที่เกี่ยวข้องเพื่อนำไปปรับปรุงพัฒนาโครงการนี้ให้ดียิ่งขึ้น

นายเกียรติสุริยา บุราเลข

นายพัชรพงศ์ ลำตัน

บทคัดย่อ

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) โดยมีวัตถุประสงค์เพื่อศึกษาการออกแบบและสร้างชุดจำลองแขนหุ่นยนต์อุตสาหกรรม, เพื่อศึกษาการควบคุมแขนหุ่นยนต์อุตสาหกรรม และเพื่อพัฒนาโปรแกรมวิเคราะห์การเคลื่อนที่ของหุ่นยนต์ เนื่องจากในภาคอุตสาหกรรมที่ต้องการกำลังการผลิตที่มากขึ้น ซึ่งหุ่นยนต์ก็สามารถตอบโจทย์นี้ได้ดี ทางคณะผู้จัดทำเกิดแนวคิดที่จะออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรมประเภท Articulated Robot (Revolute) โดยมีข้อต่อทั้งหมด 6 จุด เพื่อศึกษาส่วนประกอบ กลไก ทฤษฎีที่เกี่ยวข้อง และหลักการทำงานของหุ่นยนต์อุตสาหกรรม ตลอดจนศึกษาการควบคุมแขนหุ่นยนต์อุตสาหกรรม เพื่อที่จะสามารถพัฒนาและต่อยอดองค์ความรู้ในระดับอุตสาหกรรมได้ในอนาคต

Abstract

Project of Design of an Industrial Robot Arm with the objective of studying the design and construction of an industrial robot arm simulator, to study the control of industrial robotic arms and to develop a program for analyzing the movement of robots. Because in the industry that needs more production capacity which the robot can answer this question. Therefore, we have an idea to design an Articulated Robot (Revolute) industrial robot arm with 6 joints to study components, mechanisms, and related theories. and the working principle of industrial robots as well as studying the control of industrial robotic arms in order to be able to develop and extend knowledge at the industrial in the future.

สารบัญ

หน้า

บทคัดย่อ	ก
Abstract	ข
สารบัญ	ค
สารบัญรูป	จ
สารบัญตาราง	ซ
คำอธิบายสัญลักษณ์และคำย่อ	ณ
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของงาน	2
1.4 แนวทางการดำเนินงาน	2
1.5 แผนการดำเนินงาน	3
1.6 ผลที่คาดว่าจะได้รับ	3
1.7 งบประมาณ	4
บทที่ 2 ทฤษฎีและเอกสารที่เกี่ยวข้อง	5
2.1 แขนหุ่นยนต์ (Robot Arm)	5
2.1.1 Cartesian (Gantry) Robot	5
2.1.2 Cylindrical Robot	6
2.1.3 Spherical Robot	6
2.1.4 SCARA Robot	7
2.1.5 Articulated Robot	7
2.2 Articulated Robot	8
2.3 มอเตอร์ไฟฟ้า (Electric Motor)	8
2.3.1 สเต็ปป์มอเตอร์ (Stepping Motor)	9
2.3.2 เซอร์โวมอเตอร์ (Servo Motor)	9
2.4 เกียร์ทดรอบ (Gear speed reducer)	11
2.5 ชุดเฟืองแพลนเนตตารี (Planetary Gear)	11
2.5.1 หน้าที่ของชุดเฟืองแพลนเนตตารี	12
2.5.2 ส่วนประกอบของชุดเฟืองแพลนเนตตารี	12
2.6 ทอร์ก (Torque)	12

สารบัญ (ต่อ)

	หน้า
2.7 บอร์ดอาร์ดูโน เมก้า 2560 (Arduino MEGA 2560)	13
2.8 โมดูล A4988 (A4988 Stepping Driver)	14
2.9 สัญนิยมเดนาวิท-ฮาร์เทรเบอร์ก (Denavit-Hartenberg)	16
2.10 จลนศาสตร์แปรผัน (Forward Kinematic)	17
บทที่ 3 ขั้นตอนการดำเนินงาน	19
3.1 ด้านฮาร์ดแวร์	19
3.1.1 กำหนดขอบเขตของแขนหุ่นยนต์อุตสาหกรรม	19
3.1.2 การคำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรม	21
3.1.3 การออกแบบแขนหุ่นยนต์อุตสาหกรรม	24
3.1.4 การสร้างแขนหุ่นยนต์อุตสาหกรรม	33
3.1.5 ชุดวงจรควบคุมแขนหุ่นยนต์อุตสาหกรรม	34
3.2 ด้านซอฟต์แวร์	35
3.2.1 โปรแกรมสำหรับควบคุมแขนหุ่นยนต์อุตสาหกรรม.....	35
3.2.2 โปรแกรมคำนวณหาตำแหน่งจุดปลายและจำลองการเคลื่อนที่ของแขนหุ่นยนต์ อุตสาหกรรม	36
บทที่ 4 การทดลองและผลการทดลอง	44
4.1 การทดลองควบคุมการเคลื่อนที่เพื่อคำนวณหาตำแหน่งจุดปลายของแขนหุ่นยนต์อุตสาหกรรม	44
4.2 การทดลองหาค่าน้ำหนัก (Load) สูงสุดของแขนหุ่นยนต์อุตสาหกรรม	46
บทที่ 5 สรุปผลการดำเนินงาน	47
5.1 สรุปผลการดำเนินงาน	47
5.2 ปัญหาในการดำเนินงานและแนวทางการแก้ไข	47
5.3 ข้อเสนอแนะ	48
เอกสารอ้างอิง	49
ภาคผนวก	i

สารบัญรูป

หน้า

รูปที่ 2.1 ก) Cartesian (Gantry) Robot ข) พื้นที่การทำงานของ Cartesian (Gantry) Robot	5
รูปที่ 2.2 ก) Cylindrical Robot ข) พื้นที่การทำงานของ Cylindrical Robot	6
รูปที่ 2.3 ก) Spherical Robot ข) พื้นที่การทำงานของ Spherical Robot	6
รูปที่ 2.4 ก) SCARA Robot ข) พื้นที่การทำงานของ Spherical Robot	7
รูปที่ 2.5 ก) Articulated Robot ข) พื้นที่การทำงานของ Spherical Robot	7
รูปที่ 2.6 ภาพเปรียบเทียบร่างกายของมนุษย์กับแขนกลของหุ่นยนต์อุตสาหกรรม	8
รูปที่ 2.7 โครงสร้างสเต็ปปีงมอเตอร์ (Stepping Motor)	9
รูปที่ 2.8 แผนผังการควบคุม Stepping Motor	9
รูปที่ 2.9 ส่วนประกอบของ Servo Motor	10
รูปที่ 2.10 Block diagram การทำงานของเซอร์โวมอเตอร์	10
รูปที่ 2.11 ชุดเฟืองแพลนเนตารี (Planetary Gear)	11
รูปที่ 2.12 แสดงส่วนประกอบของชุดเฟืองแพลนเนตารี	12
รูปที่ 2.13 พอร์ตของ Arduino MEGA 2560	14
รูปที่ 2.14 โมดูล A4988 (A4988 Stepper Driver)	14
รูปที่ 2.15 Pins ของ A4988	16
รูปที่ 2.16 แสดงรายละเอียดของตำแหน่งตัวแปรต่างๆของ DH parameter.....	17
รูปที่ 2.17 การกำหนดเวกเตอร์แกนและจุดกำเนิดให้กับเฟรมทั้งสอง.....	18
รูปที่ 3.1 แผนภาพจำลองความยาวของแขนหุ่นยนต์อุตสาหกรรม	20
รูปที่ 3.2 การหาทอร์ก (Torque) ที่ Joint 1	21
รูปที่ 3.3 การหาทอร์ก (Torque) ที่ Joint 2	21
รูปที่ 3.4 การหาทอร์ก (Torque) ที่ Joint 3	22
รูปที่ 3.5 การหาทอร์ก (Torque) ที่ Joint 4	22
รูปที่ 3.6 การหาทอร์ก (Torque) ที่ Joint 5	23
รูปที่ 3.7 การหาทอร์ก (Torque) ที่ Joint 6	23
รูปที่ 3.8 Industrial Robot Arm - IRB 6700 150 3.20	24
รูปที่ 3.9 Autodesk Inventor Professional	24
รูปที่ 3.10 รูปของ Planetary Gearbox ซึ่งมีอัตราทดแรง 80:1 ออกแบบโดย MichaelRechtin	25
รูปที่ 3.11 รูปของ Planetary Gearbox ซึ่งมีอัตราทดแรง 4:1 ออกแบบโดย IndeterminateDesign ...	25
รูปที่ 3.12 รูปของ Stepper motor Planetary Gearbox ซึ่งมีอัตราทดแรง 100:1	25
รูปที่ 3.13 รูปของ Stepper motor Planetary Gearbox ซึ่งมีอัตราทดแรง 27:1	25

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.14 ส่วนฐานยึด	26
รูปที่ 3.15 ฐานส่วนที่ 1	26
รูปที่ 3.16 ฐานส่วนที่ 2	27
รูปที่ 3.17 ฐานส่วนที่ 3	27
รูปที่ 3.18 ส่วนประกอบ Gear box	28
รูปที่ 3.19 แขนหุ่นยนต์ส่วนที่ 1.....	28
รูปที่ 3.20 แขนหุ่นยนต์ส่วนที่ 2	29
รูปที่ 3.21 แขนหุ่นยนต์ส่วนที่ 3	29
รูปที่ 3.22 แกนยึด	30
รูปที่ 3.23 แขนหุ่นยนต์ส่วนที่ 4	30
รูปที่ 3.24 แขนหุ่นยนต์ส่วนที่ 5	31
รูปที่ 3.25 แขนหุ่นยนต์ส่วนที่ 6	31
รูปที่ 3.26 แขนหุ่นยนต์ส่วนที่ 7	32
รูปที่ 3.27 ภาพแสดงชื่อและตำแหน่งจุดหมุน (Joint)	32
รูปที่ 3.28 ก) ภาพ Assembly เบื้องต้น ข) ภาพแขนหุ่นยนต์อุตสาหกรรมที่ทำการประกอบเสร็จ	33
รูปที่ 3.29 ชุดวงจรควบคุมแขนหุ่นยนต์อุตสาหกรรม	34
รูปที่ 3.30 แผนผัง Flow Chart แสดงการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรม	35
รูปที่ 3.31 แขนหุ่นยนต์อุตสาหกรรม	36
รูปที่ 3.32 การตั้งแกนเพื่อสร้างแบบจำลองทางคณิตศาสตร์ของหุ่นยนต์แขนกลอุตสาหกรรม	37
รูปที่ 3.33 แผนผัง Flow Chart แสดงการทำงานของแบบจำลองทางคณิตศาสตร์	39
รูปที่ 3.34 การสร้างแบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนแกน 3 มิติ	40
รูปที่ 3.35 การสร้างแบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระนาบ X,Z	40
รูปที่ 3.36 การสร้างแบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระนาบ X,Y	41
รูปที่ 3.37 ผลการจำลองการเคลื่อนที่และคำนวณค่าจุดปลายของแขนหุ่นยนต์อุตสาหกรรม	41
รูปที่ 3.38 ผลการจำลองขอบเขตการเคลื่อนที่ทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระบบพิกัด (X,Y,Z) ใน 3 มิติ	42

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.39 ผลการจำลองขอบเขตการเคลื่อนที่ทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระนาบพิกัด (X,Z)	43
รูปที่ 3.40 ผลการจำลองขอบเขตการเคลื่อนที่ทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระนาบพิกัด (X,Y)	43

สารบัญตาราง

	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน	3
ตารางที่ 1.2 รายการงบประมาณเฉลี่ยของโครงการ	4
ตารางที่ 2.1 ข้อมูลจำเพาะของบอร์ดบอร์ดอาร์ดูโน เมก้า 2560 (Arduino MEGA 2560)	13
ตารางที่ 2.2 ข้อมูลทางเทคนิคของโมดูล A4988 (A4988 Stepper Driver)	15
ตารางที่ 2.3 ข้อมูล Pins ต่าง ๆ ของ A4988	15
ตารางที่ 3.1 ความยาวของแขนหุ่นยนต์ (Link)	20
ตารางที่ 3.2 น้ำหนักของแขนหุ่นยนต์	20
ตารางที่ 3.3 ตารางข้อมูลแบบ Denavit Hartenberg ของแขนหุ่นยนต์อุตสาหกรรม 6 แกน.....	37
ตารางที่ 4.1 ตารางแสดงการเปรียบเทียบค่าตำแหน่ง ณ จุดปลายที่วัดได้จริงกับตำแหน่งที่ได้จากแบบจำลองทางคณิตศาสตร์เพื่อหาค่าความคลาดเคลื่อน.....	45
ตารางที่ 4.3 การทดลองหาน้ำหนัก (Load) สูงสุดของแขนหุ่นยนต์อุตสาหกรรม.....	46

คำอธิบายสัญลักษณ์และคำย่อ

DC	ไฟฟ้ากระแสตรง (Direct current)
AC	ไฟฟ้ากระแสสลับ (Alternating current)
DOF	Degrees Of Freedom หรือ องศาอิสระ คือ จำนวนแกน/ทิศทางที่สามารถเคลื่อนที่ได้
PWM	PWM คืออะไร PWM หรือ Pulse Width Modulation คือสัญญาณพัลส์ที่มีค่าความถี่คงที่ แต่ความกว้าง ของพัลส์เปลี่ยนแปลงได้
DH	สัญญาณยืมเดนาวิท – ฮาร์เทรเบอร์ก (Denavit-Hartenberg)

บทที่ 1 บทนำ

1.1 หลักการและเหตุผล

แขนหุ่นยนต์ (Robot Arm) ในปัจจุบันมีบทบาทในด้านอุตสาหกรรมอย่างมาก นอกจากนั้นในด้านสุขภาพหรือทางการแพทย์เริ่มมีการใช้งานแขนหุ่นยนต์สำหรับการผ่าตัด โดยทางด้านอุตสาหกรรมจะช่วยให้การผลิตต้นทุนและเพิ่มกำลังในการผลิต รวมทั้งสร้างความน่าเชื่อถือให้แก่โรงงานอุตสาหกรรม ทำให้มีบริษัทจำนวนมากเกิดการแข่งขันการผลิตแขนหุ่นยนต์เพื่อตอบสนองต่อความต้องการของผู้บริโภค โดยฟังก์ชันการใช้ก็จะหลากหลายตามความต้องการของโรงงาน นอกจากแขนหุ่นยนต์แล้วนั้นสิ่งที่จะอยู่ควบคู่กับแขนหุ่นยนต์คือซอฟต์แวร์ ที่ใช้สำหรับสั่งการให้แขนหุ่นยนต์ทำหน้าที่ตามความต้องการของผู้บริโภค

หุ่นยนต์อุตสาหกรรม คือ หุ่นยนต์ที่มีโครงสร้างคล้ายกับร่างกายของมนุษย์ คือ เอว ข้อศอก แขน และข้อมือ คำว่า แขนกล หมายถึงแขนของหุ่นยนต์อุตสาหกรรม โดยทั่วไปแขนกลที่ใช้ในอุตสาหกรรมมีอยู่หลากหลายชนิดด้วยกัน ขึ้นอยู่กับลักษณะการทำงาน ซึ่งสามารถแบ่งออกเป็น 5 ชนิด คือ หุ่นยนต์คาร์ทีเซียน (Cartesian Coordinated Robot), หุ่นยนต์ทรงกระบอก (Cylindrical Coordinated Robot), หุ่นยนต์ทรงกลม (Spherical Coordinated Robot), หุ่นยนต์ข้อต่อ (Articulated manipulator) และหุ่นยนต์สกาล่า (Scara Robot) การพัฒนาของเทคโนโลยีจนถึงปัจจุบัน ทำให้หุ่นยนต์สามารถทำงานได้หลายรูปแบบ เพื่อตอบสนองกับภาคอุตสาหกรรมที่ต้องการกำลังการผลิตที่มากขึ้น ซึ่งหุ่นยนต์ก็สามารถตอบโจทย์นี้ได้ดี

ด้วยเหตุผลดังกล่าว ทางคณะผู้จัดทำจึงเกิดแนวคิดที่จะออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรมประเภท Articulated Robot (Revolute) โดยมีทั้งหมด 6 ข้อต่อ เพื่อศึกษาส่วนประกอบกลไก ทฤษฎีที่เกี่ยวข้อง และหลักการทำงานของหุ่นยนต์อุตสาหกรรม ตลอดจนศึกษาการควบคุมแขนหุ่นยนต์อุตสาหกรรม เพื่อที่จะสามารถพัฒนาและต่อยอดองค์ความรู้ในระดับอุตสาหกรรมได้ในอนาคต

1.2 วัตถุประสงค์

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) มีวัตถุประสงค์ของโครงการดังต่อไปนี้

1. เพื่อศึกษาการออกแบบและสร้างชุดจำลองแขนหุ่นยนต์อุตสาหกรรม
2. เพื่อศึกษาการควบคุมแขนหุ่นยนต์อุตสาหกรรม
3. เพื่อพัฒนาโปรแกรมวิเคราะห์การเคลื่อนที่ของหุ่นยนต์

1.3 ขอบเขตของงาน

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) มีขอบเขตของงานดังต่อไปนี้

1. เป็นแขนหุ่นยนต์อุตสาหกรรมประเภท Articulated Robot (Revolute) โดยมีข้อต่อ (joint) ทั้งหมด 6 จุด
2. ใช้ motor เป็นตัวขับเคลื่อนแขนหุ่นยนต์
3. แขนหุ่นยนต์อุตสาหกรรมสามารถรับน้ำหนักโหลด (Pay load) ได้ไม่เกิน 500 กรัม
4. แขนหุ่นยนต์อุตสาหกรรมสามารถเคลื่อนที่ได้ตามคำสั่ง

1.4 แนวทางการดำเนินงาน

ทำการเสนอหัวข้อและชื่ออาจารย์ที่ปรึกษาโครงการ และเริ่มศึกษาทฤษฎีที่เกี่ยวข้องกับแขนหุ่นยนต์อุตสาหกรรมประเภท Articulated Robot (Revolute) โดยกำหนดให้แขนหุ่นยนต์อุตสาหกรรมสามารถรับน้ำหนักโหลดได้ไม่เกิน 500 กรัม จากนั้นทำการทำการคำนวณตัวแปรสำคัญต่าง ๆ ที่ใช้ในการออกแบบหุ่นยนต์อุตสาหกรรม จัดหาและเลือกวัสดุ-อุปกรณ์ที่เหมาะสม โดยใช้ motor เป็นตัวขับเคลื่อนแขนหุ่นยนต์ โดยอาจจะใช้ทั้ง DC motor, Stepper motor และ Servo motor ตามความเหมาะสม เพื่อให้แขนหุ่นยนต์มีการตอบสนองทำงานได้อย่างราบรื่นและมีความแม่นยำสูงขึ้น โดยจะมีการใช้ชุดเกียร์เพื่อเพิ่มแรงบิดให้กับมอเตอร์ หลังจากวัดขนาดของอุปกรณ์กำหนดตัวแปรเสร็จสิ้น จะทำการออกแบบแขนหุ่นยนต์อุตสาหกรรม และทำการสร้างชิ้นงานโดยใช้เครื่องพิมพ์ 3 มิติ จากนั้นทำการประกอบแขนหุ่นยนต์อุตสาหกรรม ทำการทดสอบการทำงานของแขนหุ่นยนต์อุตสาหกรรม โดยการทดสอบให้แขนหุ่นยนต์หยิบจับวัตถุที่มีน้ำหนักไม่เกิน 500 กรัม จากนั้นทำการวิเคราะห์ผลผ่านโปรแกรมที่พัฒนาขึ้นและปรับปรุงแก้ไข และสรุปผลการดำเนินงานเพื่อเตรียมนำเสนอโครงการ

1.5 แผนการดำเนินงาน

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) มีแผนการดำเนินงานดังต่อไปนี้

ตารางที่ 1.1 แผนการดำเนินงาน

	รายการ	เดือน								
		พ.ศ. 2565						พ.ศ. 2566		
		ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.	เสนอหัวข้อและชื่ออาจารย์ที่ปรึกษาโครงการ									
2.	เขียนแผนการดำเนินงานและข้อเสนอโครงการ									
3.	ศึกษาและค้นคว้าทฤษฎีที่เกี่ยวข้องกับโครงการ									
4.	จัดหาและเลือกวัสดุ-อุปกรณ์ที่เหมาะสม									
5.	ออกแบบและสร้างชุดจำลองแขนหุ่นยนต์อุตสาหกรรม									
6.	ทดสอบการทำงานของชุดจำลองแขนหุ่นยนต์อุตสาหกรรม									
7.	วิเคราะห์ผลทดสอบและปรับปรุงแก้ไข									
8.	สรุปผลการดำเนินงานและเตรียมนำเสนอโครงการ									

1.6 ผลที่คาดว่าจะได้รับ

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) มีผลที่คาดว่าจะได้รับดังต่อไปนี้

1. มีความรู้ความเข้าใจในส่วนประกอบ กลไก ทฤษฎีที่เกี่ยวข้อง และหลักการทำงานของหุ่นยนต์อุตสาหกรรม
2. สามารถวิเคราะห์และคำนวณตัวแปรสำคัญต่างๆ ที่ใช้ในการออกแบบหุ่นยนต์อุตสาหกรรม
3. สามารถเขียนโปรแกรมควบคุมและทดสอบการทำงานของหุ่นยนต์อุตสาหกรรม
4. ใช้เป็นตัวต้นแบบในการพัฒนาและนำไปใช้จริงในอนาคต

1.7 งบประมาณ

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) มีงบประมาณของโครงการดังต่อไปนี้

ตารางที่ 1.2 รายการงบประมาณเฉลี่ยของโครงการ

ลำดับ ที่	รายการ	จำนวน (หน่วย)	ราคาต่อ หน่วย(บาท)	รวม(บาท)
1.	Nema 17 Stepper motor รุ่น 17HS4401S	4 ตัว	350.00	1,400.00
2.	Nema 17 Stepper motor รุ่น 42SHD4950-20BH	1 ตัว	320.00	320.00
3.	MG995 Servo motor	1 ตัว	220.00	220.00
4.	Arduino MEGA 2560	1 ตัว	275.00	275.00
5.	A4988 Stepper Driver	5 ตัว	85.00	425.00
6.	Deep Groove Ball Bearing 638-ZZ	9 ตัว	30.00	270.00
7.	Deep Groove Ball Bearing 625-ZZ	3 ตัว	70.00	210.00
8.	Deep Groove Ball Bearing 6806-2RS	3 ตัว	100.00	300.00
9.	Closed Loop Timing Belt 2GT-158 mm.	1 เส้น	20.00	20.00
10.	Timing Pulley 20 teeth	2 ชิ้น	35.00	70.00
11.	PLA Filament สำหรับ 3D Printer	2 ม้วน	550.00	1,100.00
12.	อุปกรณ์อื่นๆ	-	550.00	550.00
รวม ห้าพันบาทถ้วน				5,160.00

บทที่ 2 ทฤษฎีและเอกสารที่เกี่ยวข้อง

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) ทางคณะผู้จัดทำได้ทำการศึกษาทฤษฎีและเอกสารที่เกี่ยวข้องดังต่อไปนี้

2.1 แขนหุ่นยนต์ (Robot Arm)

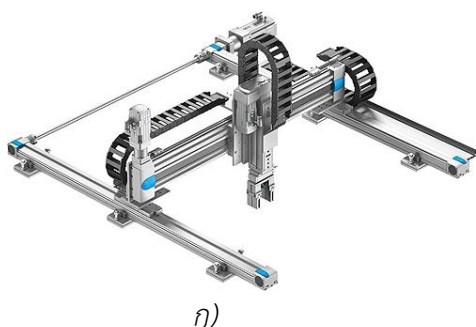
แขนหุ่นยนต์ (Robot Arm) เป็นหุ่นยนต์ชนิดหนึ่งที่น่าสนใจในงานในวงการอุตสาหกรรมการผลิต ได้ถูกนำมาใช้แทนแรงงานมนุษย์ แขนหุ่นยนต์อุตสาหกรรมที่เราพบเห็นได้โดยทั่วไปเช่น ในโรงงานผลิต ประกอบรถยนต์, งานเชื่อมอุตสาหกรรม, งานประกอบเครื่องจักร, งานในโรงงานผลิตเหล็ก, งานเกี่ยวกับ คลังสินค้าขนาดใหญ่ และอื่น ๆ อีกมากมาย

แขนหุ่นยนต์ (Robot Arm) นั้นมีส่วนประกอบอยู่หลายส่วนได้แก่ ฐาน (Base) ของหุ่นยนต์, แขนหุ่นยนต์, ข้อต่อจุดหมุน(Joints), ปลายของแขนหุ่นยนต์ที่ใช้ทำงานยกตัวอย่างเช่น มือคีบจับ หัวเชื่อม อุปกรณ์ประกอบชิ้นส่วน ปืนพ่นสีและ หัวเจาะ เป็นต้น

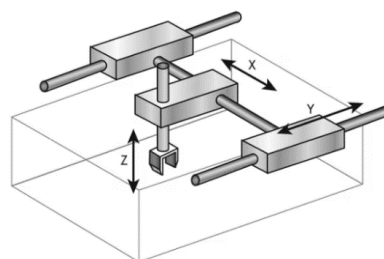
เนื่องจากหุ่นยนต์ (Robot Arm) ได้รับการออกแบบสร้างขึ้นมาเพื่อทำหน้าที่แทนคน โดยปกติแล้ว มักออกแบบเป็นแขนเดียว ในบางแบบได้ออกแบบให้แขนเคลื่อนที่อยู่บนทางเลื่อนได้ อาจจำแนกโครงสร้างของแขนหุ่นยนต์ (Robot Arm) ได้ 5 แบบ ได้แก่

2.1.1 Cartesian (Gantry) Robot

หุ่นยนต์ประเภท Cartesian มีแกนทั้งสามเคลื่อนที่เป็นเชิงเส้น (Prismatic) สามารถเคลื่อนที่เป็นแนวเส้นตรงทั้ง 3 มิติ หุ่นยนต์ประเภทนี้แบ่งออกเป็น 2 ประเภท คือ Gantry และ Cartesian โดย Gantry เป็นหุ่นยนต์รางเลื่อนที่มีลักษณะคล้ายเครนเหนือศีรษะ ส่วน Cartesian เป็นหุ่นยนต์คล้ายรางเลื่อนแต่ไม่มีขาตั้งหรือขาเป็นแบบอื่น โครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ต้องการพื้นที่ติดตั้งมาก บริเวณที่หุ่นยนต์ทำงานจะเล็กกว่าขนาดของตัวหุ่นยนต์ ซึ่งเหมาะสำหรับใช้ในงานเคลื่อนย้ายวัตถุที่มีน้ำหนักมาก ๆ (Pick and place) เช่น งานป้อนชิ้นงานเข้าเครื่องจักร งานจัดเก็บชิ้นงาน และสามารถใช้งานประเภทงานประกอบที่ไม่ต้องการเข้าถึงในลักษณะที่มีมุมหมุน เช่น งานประกอบอุปกรณ์อิเล็กทรอนิกส์ และงานทดสอบต่างๆ



ก)



ข)

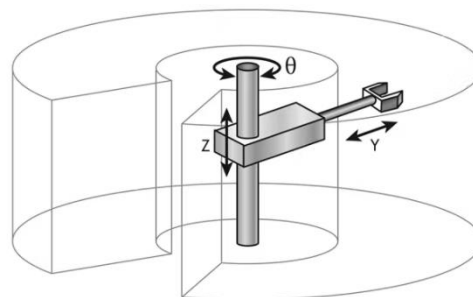
รูปที่ 2.1 ก) Cartesian (Gantry) Robot ข) พื้นที่การทำงานของ Cartesian (Gantry) Robot [1]

2.1.2 Cylindrical Robot

หุ่นยนต์ประเภท Cylindrical จะมีแกนที่ 1 (เอว) เคลื่อนที่แบบหมุนรอบแกน (Revolute) ส่วนแกนที่ 2 (ไหล่) และแกนที่ 3 (ข้อศอก) จะเคลื่อนที่แบบเชิงเส้นทำให้ได้พื้นที่การทำงานเป็นรูปทรงกระบอกสามารถหมุนแกนที่ 1 ได้รอบตัวของหุ่นยนต์ แกนที่ 2 สามารถเคลื่อนที่ขึ้นลงได้ตลอดแนว และแกนที่ 3 สามารถยืดและหดได้ตามแนวแกนรอบตัวหุ่นยนต์ โดยทั่วไปจะใช้หุ่นยนต์ชนิดนี้ในการหยิบยกชิ้นงาน หรือป้อนชิ้นงานเข้าเครื่องจักร เพราะสามารถเข้าถึงบริเวณที่เป็นช่องหรือโพรงขนาดเล็กได้สะดวก



ก)

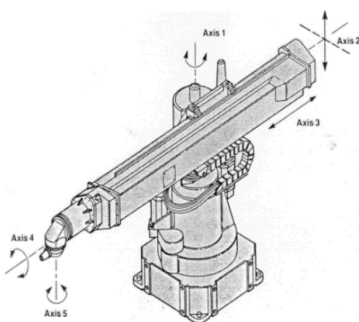


ข)

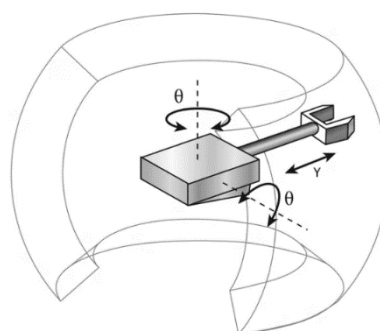
รูปที่ 2.2 ก) Cylindrical Robot ข) พื้นที่การทำงานของ Cylindrical Robot [1]

2.1.3 Spherical Robot

หุ่นยนต์แบบ Spherical หรือเรียกอีกชื่อว่า Polar Robot มีแกนที่ 1 (เอว) และแกนที่ 2 (ไหล่) เคลื่อนที่ในลักษณะหมุนรอบแกน ส่วนแกนที่ 3 (ข้อศอก) จะมีลักษณะการเคลื่อนที่เป็นแนวเส้นตรง หุ่นยนต์ชนิดนี้เหมาะสำหรับใช้ในงานที่มีการเคลื่อนที่ในแนวตั้งเล็กน้อย สามารถก้มลงมาหยิบจับชิ้นงานบนพื้นได้ เช่น งานป้อนชิ้นงานเข้าและนำชิ้นงานออกจากเครื่องปั๊ม (Press) งานเชื่อมจุด (Spot welding)



ก)



ข)

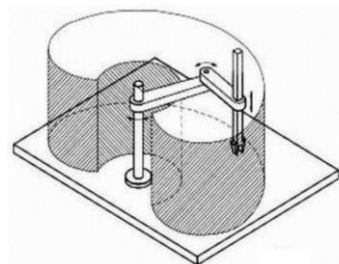
รูปที่ 2.3 ก) Spherical Robot ข) พื้นที่การทำงานของ Spherical Robot [1]

2.1.4 SCARA Robot

หุ่นยนต์ประเภท SCARA หรือ Selective Compliance Assembly Robot Arm มีแกนที่ 1 (เอว) และแกนที่ 3 (ข้อศอก) เคลื่อนที่ในลักษณะหมุนรอบแกนแนวตั้ง ส่วนแกนที่ 2 (ไหล่) จะมีลักษณะการเคลื่อนที่เป็นแบบเชิงเส้น (Prismatic) ในแกนแนวตั้ง หุ่นยนต์ SCARA สามารถเคลื่อนที่ได้รวดเร็วในแนวระนาบ และมีความแม่นยำสูง แต่ไม่สามารถหมุนในลักษณะมุมต่างๆได้ และไม่สามารถรับน้ำหนักได้มาก หุ่นยนต์ประเภท SCARA เหมาะสำหรับงานประกอบชิ้นส่วนทางอิเล็กทรอนิกส์ เนื่องจากสามารถเคลื่อนที่ในแนวระนาบและขึ้นลงได้อย่างรวดเร็ว นอกจากนี้ยังเหมาะกับการตรวจสอบ (Inspection) งานบรรจุภัณฑ์ (Packing)



ก)



ข)

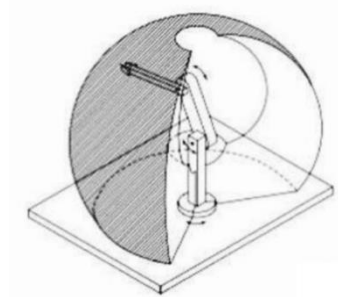
รูปที่ 2.4 ก) SCARA Robot ข) พื้นที่การทำงานของ Spherical Robot [1]

2.1.5 Articulated Robot

หุ่นยนต์ประเภท Articulated จะมีแกนทุกแกนของหุ่นเคลื่อนที่ในลักษณะหมุนรอบแกน (Revolute) โดยจะเลียนแบบส่วนบนของลำตัวมนุษย์ ประกอบด้วยช่วงเอว ท่อนแขนบน ท่อนแขนล่าง รวมไปถึงข้อมือ รูปแบบการเคลื่อนที่ที่มีการเลียนแบบแขนของมนุษย์ ซึ่งการเคลื่อนที่ในลักษณะนี้จะครอบคลุมพื้นที่การทำงานได้มาก หุ่นยนต์ชนิดนี้จึงสามารถใช้งานได้หลากหลายรูปแบบ สามารถเข้าถึงตำแหน่งต่าง ๆ ได้ดี เช่น งานพ่นสี งานประเภทงานเชื่อม (Welding) งานยกสิ่งของ แต่ไม่เหมาะสมกับงานที่ต้องการความแม่นยำมากนักเนื่องจากโครงสร้างของหุ่นยนต์ไม่มั่นคงตลอดช่วงการเคลื่อนที่ โดยเฉพาะบริเวณปลายแขนที่จะมีการสั่นทำให้ความแม่นยำลดลง [1]



ก)

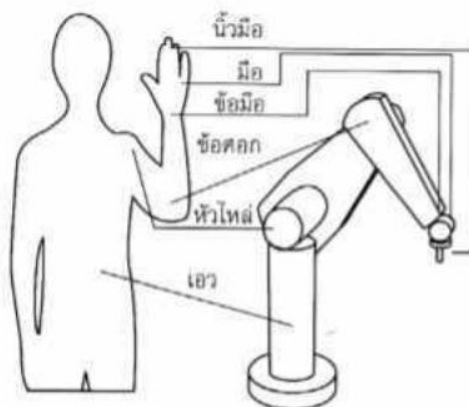


ข)

รูปที่ 2.5 ก) Articulated Robot ข) พื้นที่การทำงานของ Spherical Robot

2.2 Articulated Robot

โครงการ การออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) เลือกทำแขนหุ่นยนต์ (Robot Arm) ประเภท Articulated Robot เนื่องจากมีโครงสร้างที่คล้ายคลึงกับของของมนุษย์ โดยมีข้อต่อ 6 ข้อต่อ โดยคล้ายกับแขนของมนุษย์ที่เริ่มนับจากหัวไหล่ ข้อศอก และมือ ในหุ่นยนต์จะมีฐานหุ่นคล้ายบ่าเพื่อรองรับโครงสร้างที่มีการเคลื่อนที่ เราเรียกข้อต่อจุดหมุนว่าเป็นองศาอิสระ (Degrees Of Freedom: DOF) หมายถึงความสามารถที่จะเคลื่อนไหวได้อย่างอิสระภายใต้ระยะจุดหมุนที่หมุนได้ ถ้าเปรียบเทียบกับแขนมนุษย์ที่สามารถยกแขนให้เคลื่อนที่จากตำแหน่งไปสู่ตำแหน่งหนึ่ง แขนกลก็เหมือนกัน ซึ่งสามารถทำการเคลื่อนที่ได้จากจุดหนึ่งไปสู่จุดหนึ่งในระยะขอบเขตรัศมีการเคลื่อนที่ ในปัจจุบันนี้หุ่นยนต์ชนิดนี้ถูกนำมาประยุกต์ใช้งานให้มีความสามารถมากกว่าทำงานใช้หยิบจับชิ้นงาน ไม่ว่าจะเป็นการประยุกต์ใช้ในกระบวนการงานเชื่อมโลหะต่างๆ งานพ่นสี หรืองาน Spot Gun และบางองค์กรยังมีการพัฒนาให้หุ่นยนต์ชนิดนี้สามารถทำงานในกระบวนการ Machining อีกด้วย [2]



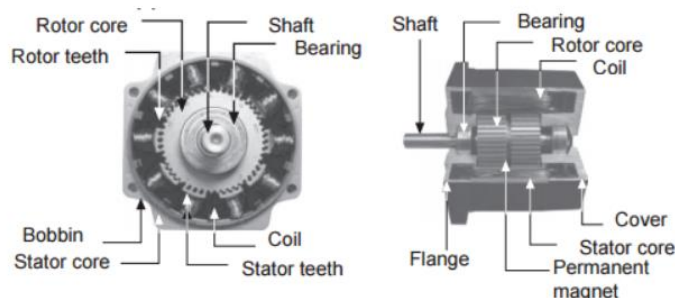
รูปที่ 2.6 ภาพเปรียบเทียบร่างกายของมนุษย์กับแขนกลของหุ่นยนต์อุตสาหกรรม [3]

2.3 มอเตอร์ไฟฟ้า (Electric Motor)

มอเตอร์ไฟฟ้า เป็นอุปกรณ์ไฟฟ้าที่แปลงพลังงานไฟฟ้าเป็นพลังงานกล การทำงานปกติของมอเตอร์ไฟฟ้าส่วนใหญ่เกิดจากการทำงานร่วมกันระหว่างสนามแม่เหล็กของแม่เหล็กในตัวมอเตอร์ และสนามแม่เหล็กที่เกิดจากกระแสในขดลวดทำให้เกิดแรงดูดและแรงผลักของสนามแม่เหล็กทั้งสอง มอเตอร์ไฟฟ้าที่ใช้พลังงานไฟฟ้าเปลี่ยนเป็นพลังงานกลมีทั้งพลังงานไฟฟ้ากระแสสลับและพลังงานไฟฟ้ากระแสตรง มอเตอร์ประกอบด้วย ขดลวดที่พันรอบแกนโลหะที่วางอยู่ระหว่างขั้วแม่เหล็ก โดยเมื่อผ่านกระแสไฟฟ้าเข้าไปยังขดลวดที่อยู่ระหว่าง ขั้วแม่เหล็ก จะทำให้ขดลวดหมุนไปรอบแกน และเมื่อสลับขั้วไฟฟ้า การหมุนของขดลวดจะหมุนกลับทิศทางเดิม ในงานอุตสาหกรรม มอเตอร์มีหลายแบบหลายชนิด ซึ่งควรเลือกใช้งานให้เหมาะสมกับงานที่เกี่ยวข้อง มอเตอร์ (Motor) มี 2 ประเภท คือ 1. มอเตอร์กระแสตรง (DC Motor) 2. มอเตอร์กระแสสลับ (AC Motor) [4]

2.3.1 สเต็ปป์มอเตอร์ (Stepping Motor)

สเต็ปป์มอเตอร์ (Stepping Motor) หรือ (Stepper Motor) เป็นมอเตอร์ไฟฟ้าที่ขับเคลื่อนด้วยพัลส์ โดยโครงสร้างภายในนั้นจะประกอบไปด้วยขั้วแม่เหล็กบนสเตเตอร์ (Stator) ทำมาจากแผ่นเหล็กวงแหวน จะมีซี่ยื่นออกมาประกอบกันเป็นชั้นๆ โดยแต่ละซี่ที่ยื่นออกมานั้นจะมีขดลวด (คอยล์) พันอยู่ เมื่อมีกระแสผ่านคอยล์จะเกิดสนามแม่เหล็กไฟฟ้าขึ้น



รูปที่ 2.7 โครงสร้างสเต็ปป์มอเตอร์ (Stepping Motor) [5]

ในการทำงานของสเต็ปป์มอเตอร์ (Stepping Motor) หรือ (Stepper Motor) นั้นจะไม่สามารถขับเคลื่อนหรือทำงานเองได้ จำเป็นต้องมีวงจรอิเล็กทรอนิกส์ที่ใช้ในการสร้างสัญญาณหรือจ่ายพัลส์ไปให้วงจรขับสเต็ปป์มอเตอร์ (Stepping Motor Drive) การสร้างสัญญาณนั้นจะเป็นต้องสร้างและเรียงลำดับของสัญญาณ [5]



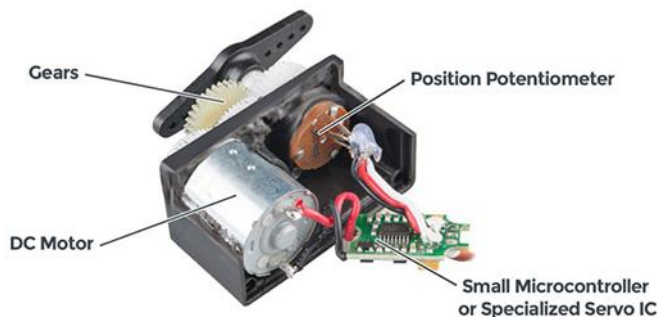
รูปที่ 2.8 แผนผังการควบคุม Stepping Motor [5]

2.3.2 เซอร์โวมอเตอร์ (Servo Motor)

เซอร์โวมอเตอร์ (Servo Motor) เป็นการรวมมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) เข้ากับวงจรควบคุม โดยความแตกต่างที่สำคัญของเซอร์โวมอเตอร์กับมอเตอร์แบบอื่น ๆ คือเซอร์โวมอเตอร์จะรู้ตำแหน่งที่ตัวเองอยู่ และสั่งเปลี่ยนตำแหน่งโดยการเปลี่ยนองศาได้ สามารถควบคุมองศาของเซอร์โวมอเตอร์ (Servo Motor) โดยใช้สัญญาณ PWM

1) ส่วนประกอบของ Servo Motor เซอร์โวมอเตอร์ประกอบด้วย 4 ส่วนหลัก คือ

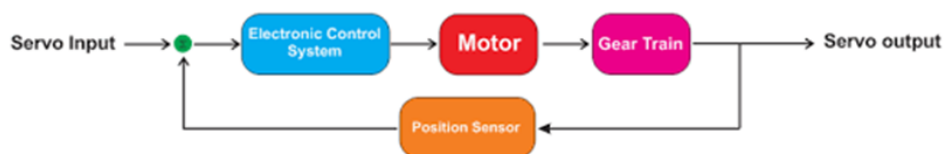
- 1.1) มอเตอร์กระแสตรง (DC Motor)
- 1.2) ชุดเฟืองทดรอบ (Gear)
- 1.3) วอลุ่ม (Potentiometer)
- 1.4) วงจรควบคุม (Control Electronics)



รูปที่ 2.9 ส่วนประกอบของ Servo Motor [6]

2) หลักการทำงานของเซอร์โวมอเตอร์

เมื่อจ่ายสัญญาณพัลส์เข้ามายัง Servo Motor ส่วนวงจรควบคุม(Electronic Control System) ภายใน Servo Motor จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัววัดค่ามุมที่ Motor กำลังหมุนเป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ



รูปที่ 2.10 Block diagram การทำงานของเซอร์โวมอเตอร์ [6]

มุมหรือองศาจะขึ้นอยู่กับความกว้างของสัญญาณพัลส์ที่เป็นบวก ดังนั้นการหาค่าความกว้างของสัญญาณพัลส์ ช่วงบวกของสัญญาณจากค่าองศา สามารถหาได้จากสูตร [6]

$$Pulse_{\text{บวก}} = \frac{Pulse_{\text{max}} - Pulse_{\text{min}}}{180}(\theta) + Pulse_{\text{min}}$$

โดย $Pulse_{\text{บวก}}$	คือ ความกว้างของสัญญาณพัลส์บวกของสัญญาณ
$Pulse_{\text{max}}$	คือ ความกว้างของสัญญาณพัลส์บวกสูงสุด
$Pulse_{\text{min}}$	คือ ความกว้างของสัญญาณพัลส์บวกต่ำสุด
θ	คือ มุมองศาของ servo motor

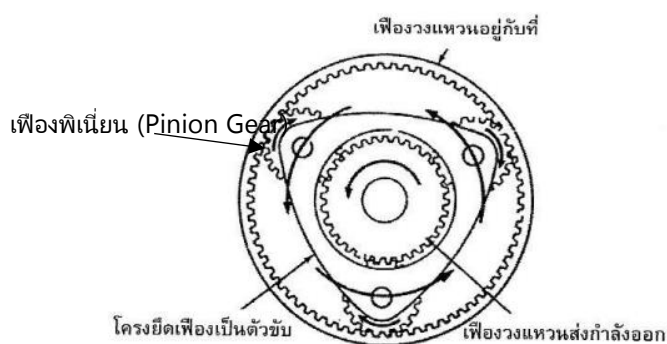
2.4 เกียร์ทดรอบ (Gear speed reducer)

เกียร์ทดรอบ หรือ Gear reducer เป็นอุปกรณ์สำหรับการส่งกำลังแบบปิดอิสระ ใช้ในการลดความเร็ว และเพิ่มแรงบิดเพื่อตอบสนองความต้องการของการทำงานของมอเตอร์ หรือ เครื่องยนต์ ให้เร็วขึ้น หรือ ช้าลง เพื่อตอบสนองความต้องการของการใช้งาน กำลังส่งสูง โครงสร้างเกียร์ตามหลักการออกแบบโมดูลาร์ (Modular) ง่ายต่อการใช้และการบำรุงรักษา ตามระยะเวลาการใช้งาน ส่วนใหญ่นิยมใช้กับเครื่องจักรทั่วไป เช่น สายพานลำเลียง, ถังผสม, เครื่องจักรทางการเกษตร, การยกหรือลากสิ่งของที่มีน้ำหนักมาก

หลักการทำงานของเกียร์ทดรอบคือการส่งถ่ายกำลัง เช่น เครื่องยนต์ หรือ มอเตอร์ไฟฟ้า ทำหน้าที่ด้วยการส่งถ่ายกำลังของเครื่องส่งกำลังผ่านเกียร์ทดรอบ เพื่อลดรอบหมุนที่ส่งออกมาจากเครื่องส่งกำลังมอเตอร์ให้ช้าลง โดยส่งกำลังผ่านฟันเฟืองทำให้มีแรงบิดที่เพิ่มมากขึ้น [7]

2.5 ชุดเฟืองแพลนเนตารี (Planetary Gear)

ชุดเฟืองแพลนเนตารี (Planetary Gear) หรือ ชุดเฟืองพระเคราะห์ ประกอบด้วยชุดเฟืองพินเนียน (Pinion Gear) เฟืองกลาง (Sun Gear) เฟืองวงแหวน (Ring Gear) และโครงยึดเฟือง (Carrier) เป็นชุดทดแรงส่งถ่ายกำลัง

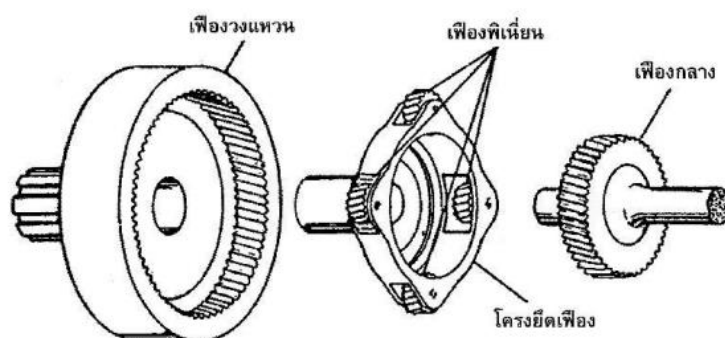


รูปที่ 2.11 ชุดเฟืองแพลนเนตารี (Planetary Gear) [8]

2.5.1 หน้าที่ของชุดเฟืองแพลนเนตารี

- 1) เปลี่ยนทิศทางการหมุน
- 2) ส่งถ่ายกำลังได้หลายอัตราทด
- 3) ใช้เป็นโอเวอร์ไดรฟ์เพื่อเพิ่มความเร็วรอบที่ตัวส่งกำลังออก (Output)

2.5.2 ส่วนประกอบของชุดเฟืองแพลนเนตารี



รูปที่ 2.12 แสดงส่วนประกอบของชุดเฟืองแพลนเนตารี [8]

1) เฟืองกลาง เฟืองกลางจะหมุนอิสระ อยู่บนเพลาส่งกำลัง (Transmission Shaft) สวมอยู่ระหว่างกลางเฟืองพิเนียนจะช่วยลดความเร็วรอบเมื่อเป็นเฟืองขับ และเพิ่มความเร็วรอบเมื่อเป็นเฟืองตาม

2) เฟืองพิเนียนหรือเฟืองแพลนเนตพิเนียน (Planet Pinion Gears) เฟืองพิเนียนจะยึดติดอยู่กับโครงยึดเฟือง และหมุนอิสระรอบตัวเอง เฟืองพิเนียนปกติมีอย่างน้อย 3 ตัว เรียกว่าชุดเฟืองแพลนเนตตารี่ชั้นเดียว ถ้าเป็นแบบ 2 ชั้นจะมีเฟืองพิเนียน 6 ตัว อยู่ระหว่างเฟืองวงแหวนกับเฟืองกลาง ทำหน้าที่เหมือนเฟืองสะพาน หมุนด้วยความเร็วกว่าเพลาส่งกำลังออก

3) เฟืองวงแหวน เฟืองวงแหวนจะขบอยู่กับชุดเฟืองพิเนียน เพลาเป็นร่องจำปา (Spline) สวมต่อกับเพลาส่งกำลังออกความเร็วรอบเท่ากับเพลากลาง และทิศทางการหมุนเหมือนกันเพิ่มความเร็วรอบเมื่อเป็นเฟืองขับ และลดความเร็วรอบเมื่อเป็นเฟืองตาม

4) โครงยึดเฟืองแพลนเนตคาร์ริเออร์ (Planet Carrier) หรือ (Housing) โครงยึดเฟืองแพลนเนตคาร์ริเออร์ ทำหน้าที่ยึดเฟืองพิเนียน ให้เฟืองพิเนียนหมุนรอบตัวได้ เพิ่มความเร็วรอบเมื่อเป็นเฟืองขับ และลดความเร็วรอบเมื่อเป็นเฟืองตาม ถ้าโครงยึดเฟืองอยู่กับที่ ทิศทางการหมุนของเพลาส่งกำลังออกจะกลับทิศทางเป็นเกียร์ถอยหลัง [8]

2.6 ทอร์ก (Torque)

ปริมาณที่ทำให้เกิดการหมุนของวัตถุรอบแกนหมุน เมื่อแรงอยู่ห่างจากจุดหมุนเป็นระยะ r โดยทอร์กเป็นปริมาณเวกเตอร์ มีหน่วยเป็น นิวตัน-เมตร ขนาดของทอร์กจะเท่ากับแรงคูณกับระยะทางที่ตั้งฉากจากจุดหมุนถึงแนวแรง [9]

$$\tau = r \times F$$

τ = คือ ทอร์กของแรง หน่วยเป็น นิวตัน-เมตร

r = คือ รัศมีการหมุนของวัตถุ หน่วยเป็นเมตร

F = คือ แรงที่กระทำต่อวัตถุในทิศตั้งฉากกับรัศมีการหมุน หน่วยเป็นนิวตัน

2.7 บอร์ดอาร์ดูโน เมก้า 2560 (Arduino MEGA 2560)

Arduino MEGA 2560 เป็นบอร์ดไมโครคอนโทรลเลอร์ ATmega328P ชนิดหนึ่ง ซึ่งรวมถึงสิ่งที่จำเป็นทั้งหมดบนบอร์ด โดยไมโครคอนโทรลเลอร์เชื่อมต่อกับพีซีด้วยสาย USB บอร์ด Arduino MEGA 2560 มีข้อมูลจำเพาะดังตารางต่อไปนี้

ตารางที่ 2.1 ข้อมูลจำเพาะของบอร์ดบอร์ดอาร์ดูโน เมก้า 2560 (Arduino MEGA 2560) [10]

ชิปไอซีไมโครคอนโทรลเลอร์	ATmega2560
ใช้แรงดันไฟฟ้า	5V
รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ)	7 – 12V
รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด)	6 – 20V
พอร์ต Digital I/O	54 พอร์ต (มี 15 พอร์ต PWM output)
พอร์ต Analog Input	16 พอร์ต
กระแสไฟที่จ่ายได้ในแต่ละพอร์ต	40mA
กระแสไฟที่จ่ายได้ในพอร์ต 3.3V	50mA
พื้นที่โปรแกรมภายใน	256KB พื้นที่โปรแกรม, 8KB ใช้โดย Bootloader
พื้นที่แรม	2KB
พื้นที่หน่วยความจำถาวร (EEPROM)	4KB
ความถี่คริสตัล	16MHz
น้ำหนัก	25 กรัม

ตารางที่ 2.2 ข้อมูลทางเทคนิคของโมดูล A4988 (A4988 Stepper Driver)

Technical Specs	Value
Operating Voltage	8 - 35 V
Logic Voltage	3 - 5.5 V
Continuous current per phase	1 A
Maximum current per phase	2 A
Micro step resolutions	Full, 1/2, 1/4, 1/8, and 1/16

จากตารางที่ 2.2 คือข้อมูลทางเทคนิคของโมดูล A4988 (A4988 Stepper Driver) โดยความหมายของ Technical Specs มีดังนี้

- Operating Voltage คือแรงดันที่ Stepper Motor นำไปใช้ โดยมาจากแหล่งจ่ายภายนอกเช่น Switching Power Supply หรือ Adapter
- Logic Voltage คือแรงดันสั่งการการหมุนของ Stepper โดยจะใช้แรงดันตั้งแต่ 3 - 5.5 V ในการสั่งการสถานะ HIGH หรือ LOW
- Continuous current per phase คือปริมาณกระแสต่อเฟสสูงสุดที่สามารถทนได้ โดยไม่มีระบบระบายความร้อน
- Maximum current per phase คือปริมาณกระแสต่อเฟสสูงสุดที่สามารถทนได้ เมื่อมีระบบระบายความร้อนแล้ว เช่น Heatsink หรือพัดลมระบายความร้อน
- Micro step resolutions คือความละเอียดของการหมุน ยิ่งน้อย ยิ่งแม่นยำ และสิ้นเปลืองมาก แต่ก็ทำให้เวลาในการทำงานช้าลงด้วย [12]

ตารางที่ 2.3 ข้อมูล Pins ต่างๆ ของ A4988

PINs	Description
VMOT	V Motor ไฟเลี้ยงของมอเตอร์ ตั้งแต่ 8 - 35 V
GND (1)	Ground ของแหล่งจ่ายไฟเลี้ยงมอเตอร์
2B	Output ไปยังสายไฟของมอเตอร์ เส้น 2B
2A	Output ไปยังสายไฟของมอเตอร์ เส้น 2A
1A	Output ไปยังสายไฟของมอเตอร์ เส้น 1A
1B	Output ไปยังสายไฟของมอเตอร์ เส้น 1B
VDD	ไฟเลี้ยง Logic 3 - 5.5 V
GND (2)	Ground ของ ไฟเลี้ยง Logic

DIR	Direction
STEP	Step pulse
Sleep	การเข้าสู่ Sleep Mode
Reset	Reset การทำงาน
MS3	ปรับ Step Resolution
MS2	ปรับ Step Resolution
MS1	ปรับ Step Resolution
Enable	Output Disable / Enable

จากตารางที่ 2.3 โมดูล A4988 มี Pin ใช้งานทั้งหมด 16 ขา ซึ่งแบ่งเป็นขาไฟเลี้ยง 4 ขา , คอนโทรล 8 ขา และ Output 4 ขา



รูปที่ 2.15 Pins ของ A4988 [12]

2.9 สัญนิมเดนาวิท-ฮาร์เทเบอร์ก (Denavit-Hartenberg)

สัญนิมเดนาวิท - ฮาร์เทเบอร์ก หรือ DH ถูกคิดขึ้นโดยนายจาคส์ เดนาวิท (Jacques Denavid) และนายริชาร์ด ฮาร์เทเบอร์ก (Richard Hartenberg) โดยได้นำเสนอวิธีการแก้ปัญหาจลนศาสตร์อย่างเป็นระบบในรูปแบบของเมตริก

Denavit-Hartenberg ในความเป็นแบบแผนแต่ละ Homogeneous Transformation Matrix ของ A_i เป็นการนำเสนอตามทีผลคูณของสี่ลักษณะพื้นฐานในการเปลี่ยนรูป [19]

$$\begin{aligned}
 A_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_ic_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_is_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

กำหนด $a_i, d_i, \alpha_i, \theta_i$ เป็นตัวแปรของข้อต่อ i และจุดต่อ i เป็นตัวแปรต่างๆในสมการโดยปกติจะแทนความหมายตามนี้ a_i ใช้เรียกแทนความยาว α_i ใช้เรียกมุมแทนการบิด d_i ใช้เรียกแทนระยะที่ยื่นออกมาและ θ_i ใช้เรียกแทนมุม เมื่อเมตริก A_i เป็นฟังก์ชันของตัวแปรเดียวซึ่งเกิดขึ้นมาจาก สามสิ่งทีนอกเหนือไปกว่าสี่ขนาดของค่าคงที่ที่ได้มาจากข้อต่อ ในขณะที่สี่ตัวแปรจะมี θ_i ของจุดหมุนและ d_i ของจุดต่อเลื่อนเป็นค่าไม่คงที่ของจุดต่อนั้นๆ ดังแสดงในรูป 2.16

แกน x_i จะอยู่บนจุดต่อ $i + 1$

แกน y_i จะอยู่บนเส้นตั้งฉากระหว่างจุดต่อ i กับ จุดต่อ $i + 1$

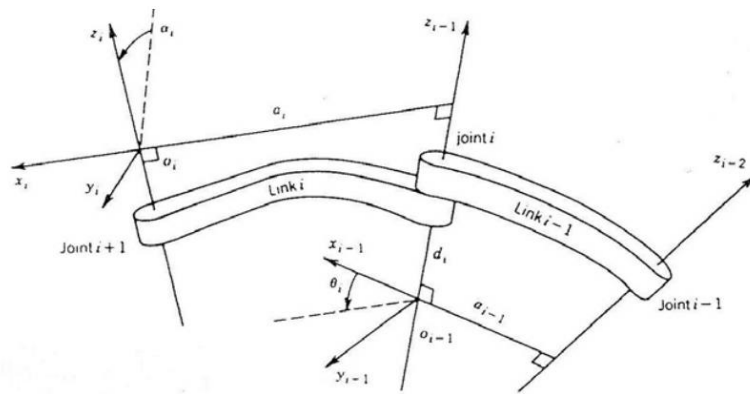
แกน z_i จะเป็นไปตามกฎมือขวา

a_i คือ ความยาวของเส้นตั้งฉากระหว่างแกน z_i และ z_{i-1}

d_i คือ ระยะห่างแกน z_i และ x_{i-1} ตามแนวแกน z_{i-1}

α_i คือ มุมระยะห่างแกน z_i และ z_{i-1} วัดรอบแกน x_i ตามกฎมือขวา

θ_i คือ มุมระยะห่างแกน z_i และ z_{i-1} วัดรอบแกน z_{i-1} ตามกฎมือขวา



รูปที่ 2.16 แสดงรายละเอียดของตำแหน่งตัวแปรต่างๆของ DH parameter [19]

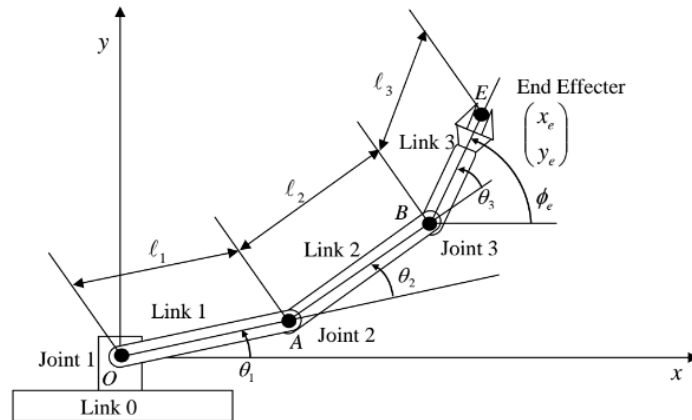
2.10 จลนศาสตร์แปรผัน (Forward Kinematic)

ในการวิเคราะห์จลนศาสตร์แปรผันสำหรับตำแหน่งนั้น ตัวแปรอิสระและพารามิเตอร์ ของก้านต่อและของข้อต่อจะต้องถูกกำหนดไว้ก่อนการวิเคราะห์ แล้วจึงถูกนำมาพิจารณาตามสมการ ของสัญญาณเดนาวิท-ฮาร์เทนเบอร์ก ค่าตอบที่ได้จะมีค่าเพียงค่าเดียวและอยู่ในรูปเมตริกแสดง ความสัมพันธ์ของเฟรมเครื่องมือและเฟรมฐาน โดยอาศัยหลักการ Homogeneous Transformation Matrix จะได้สมการจลนศาสตร์ไปข้างหน้า (Forward Kinematic) ของปลายแขนกล 6 แกน ซึ่งตำแหน่งของปลายแขนกลจะเป็นฟังก์ชันที่แสดงโดยเมตริกซ์ดังนี้ [22]

$$T_1^{i-1} = \begin{bmatrix} \cos\theta_i & \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\alpha_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0_{1 \times 3} & & & 1 \end{bmatrix}$$

$$T_i^0 = T_1^0 T_2^1 \dots \dots \dots T_i^{i-1}$$

$$T_i^0 = \begin{bmatrix} R_{3 \times 3} & p_{1 \times 3} \\ 0_{1 \times 3} & 1 \end{bmatrix}$$



รูปที่ 2.17 การกำหนดเวกเตอร์แกนและจุดกำเนิดให้กับเฟรม [22]

บทที่ 3 ขั้นตอนการดำเนินงาน

ในขั้นตอนการดำเนินการออกแบบชุดจำลองแขนหุ่นยนต์อุตสาหกรรม (Design of an Industrial Robot Arm) ทางคณะผู้จัดทำได้แบ่งขั้นตอนการดำเนินงานเป็น 3 ส่วนดังต่อไปนี้

3.1 ด้านฮาร์ดแวร์

3.1.1 กำหนดขอบเขตของแขนหุ่นยนต์อุตสาหกรรม

3.1.2 การคำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรม

3.1.3 การออกแบบแขนหุ่นยนต์อุตสาหกรรม

3.1.4 การสร้างแขนหุ่นยนต์อุตสาหกรรม

3.1.5 ชุดวงจรควบคุมแขนหุ่นยนต์อุตสาหกรรม

3.2 ด้านซอฟต์แวร์

3.2.1 โปรแกรมสำหรับควบคุมตัวหุ่นยนต์อุตสาหกรรม

3.2.2 โปรแกรมคำนวณหาตำแหน่งส่วนปลายและจำลองการทำงานของหุ่นยนต์อุตสาหกรรม

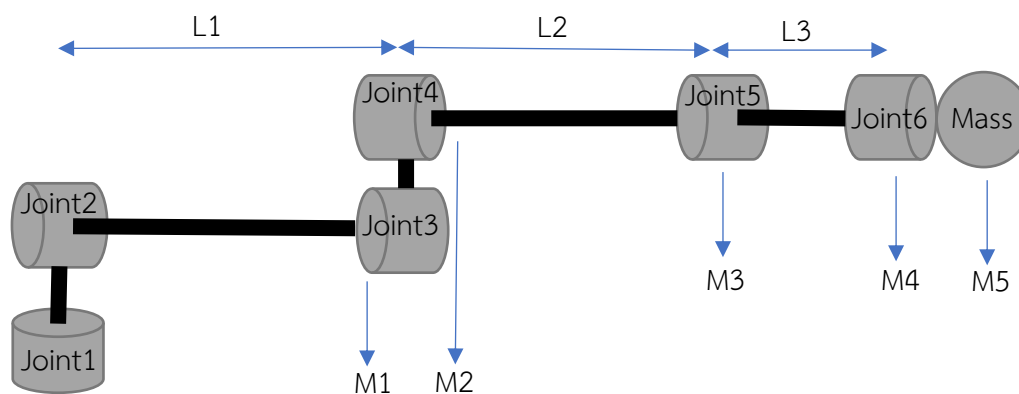
3.1 ด้านฮาร์ดแวร์

ในขั้นตอนการดำเนินงานด้านฮาร์ดแวร์ เป็นขั้นตอนในการออกแบบและสร้างแขนหุ่นยนต์อุตสาหกรรม รวมถึงชุดวงจรควบคุมแขนหุ่นยนต์อุตสาหกรรม โดยมีขั้นตอนการดำเนินงานด้านฮาร์ดแวร์ ดังนี้

3.1.1 กำหนดขอบเขตของแขนหุ่นยนต์อุตสาหกรรม




เป็นการกำหนดความยาวของแขนหุ่นยนต์ (Link) และน้ำหนักโดยประมาณ รวมถึงน้ำหนักมอเตอร์ที่จะนำมาใช้ เพื่อคำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรม และจำเป็นต้องคำนึงถึงขอบเขตของงานที่ตั้งไว้ดังนี้

- 1) เป็นแขนหุ่นยนต์อุตสาหกรรมประเภท Articulated Robot (Revolute) โดยมีข้อต่อ (joint) ทั้งหมด 6 จุด
- 2) ใช้ motor เป็นตัวขับเคลื่อนแขนหุ่นยนต์
- 3) แขนหุ่นยนต์อุตสาหกรรมสามารถรับน้ำหนักโหลด (Pay load) ได้ไม่เกิน 500 กรัม



รูปที่ 3.1 แผนภาพจำลองความยาวของแขนหุ่นยนต์ (Arm)

โดยความหมายของสัญลักษณ์ในแผนภาพจำลองมีดังนี้

-  แทน ข้อต่อจุดหมุน (Joint)
-  แทน น้ำหนักโหลด (Pay load)
-  แทน แขนหุ่นยนต์ (Link)
- L แทน ความยาวแขนหุ่นยนต์
- M แทน มวลของแขนหุ่นยนต์, มอเตอร์ และน้ำหนักโหลด (Pay load)

ตารางที่ 3.1 ความยาวของแขนหุ่นยนต์ (Link)

Link	ความยาว (เมตร)
L1	0.09
L2	0.08
L3	0.05

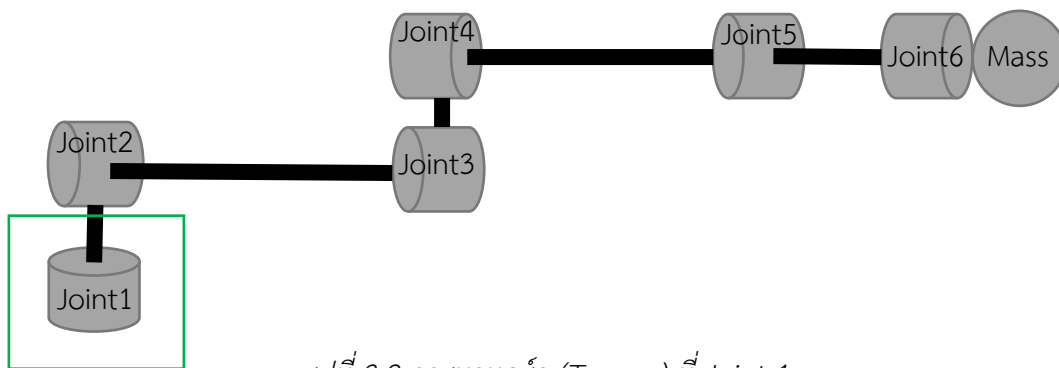
ตารางที่ 3.2 น้ำหนักของแขนหุ่นยนต์

มวล	น้ำหนัก (กิโลกรัม)
M1	0.50
M2	0.50
M3	0.35
M4	0.20
M5	0.50

3.1.2 การคำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรม

คำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรม เพื่อระบุทอร์ก (Torque) ของมอเตอร์ที่ต้องใช้ในการขับเคลื่อนหุ่นยนต์อุตสาหกรรม โดยพิจารณาในแต่ละข้อต่อ (joint) ดังต่อไปนี้

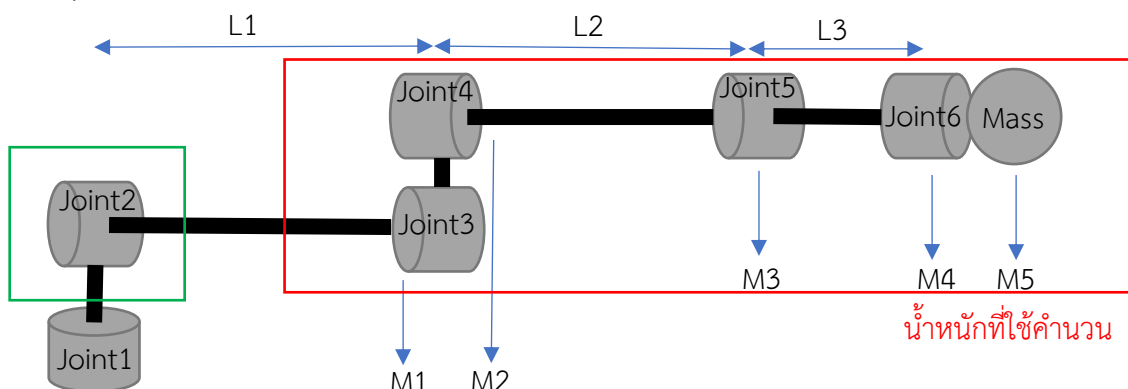
1) การคำนวณแรงขับเคลื่อนที่ Joint 1



รูปที่ 3.2 การหาทอร์ก (Torque) ที่ Joint 1

เนื่องจากแนวของน้ำหนักไม่ตั้งฉากกับแกนหมุน จึงทำให้ไม่สามารถหาคำนวณหาแรงขับเคลื่อนที่ Joint 1 ได้

2) การคำนวณแรงขับเคลื่อนที่ Joint 2



รูปที่ 3.3 การหาทอร์ก (Torque) ที่ Joint 2

จากสมการทอร์ก (Torque) เท่ากับแรงคูณกับระยะทางที่ตั้งฉากจากจุดหมุนถึงแนวแรง

$$\tau = r * F$$

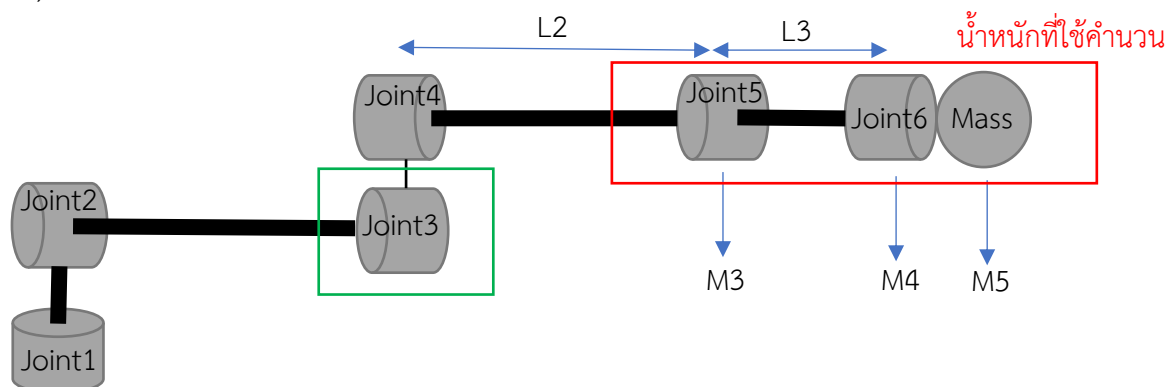
$$\tau = [L1 * ((M1 + M2) * g)] + [(L1 + L2) * (M3 * g)] + [(L1 + L2 + L3) * (M4 + M5) * g]$$

$$\tau = [0.10 * ((0.50 + 0.50) * 9.8)] + [(0.10 + 0.08) * (0.35 * 9.8)] + [(0.10 + 0.08 + 0.05) * ((0.20 + 0.50) * 9.8)]$$

$$\tau = 3.175 \text{ N-m}$$

ดังนั้นทอร์ก (Torque) ของมอเตอร์ที่ต้องใช้ในการขับเคลื่อนที่ Joint 2 ต้องมากกว่า 3.175 N-m

3) การคำนวณแรงขับเคลื่อนที่ Joint 3



รูปที่ 3.4 การหาทอร์ก (Torque) ที่ Joint 3

จากสมการทอร์ก (Torque) เท่ากับแรงคูณกับระยะทางที่ตั้งฉากจากจุดหมุนถึงแนวแรง

$$\tau = r * F$$

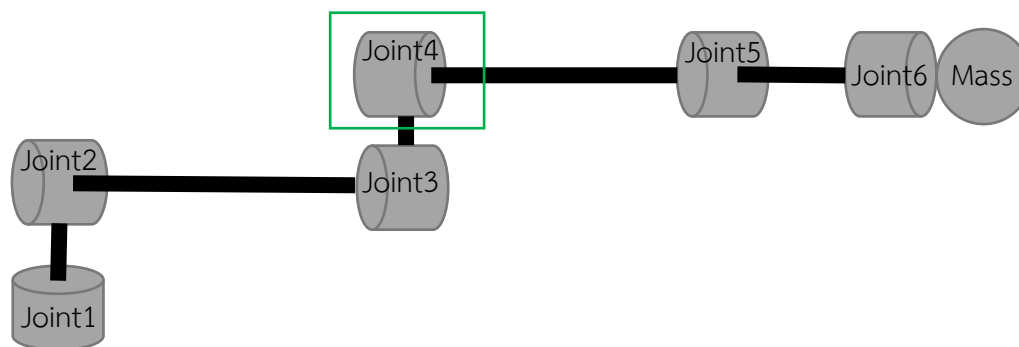
$$\tau = [L2 * (M3 * g)] + [(L2 + L3) * ((M4 + M5) * g)]$$

$$\tau = [0.08 * (0.35 * 9.8)] + [(0.08 + 0.05) * ((0.20 + 0.50) * 9.8)]$$

$$\tau = 1.166 \text{ N-m}$$

ดังนั้นทอร์ก (Torque) ของมอเตอร์ที่จะต้องใช้ในการขับเคลื่อนจุดหมุนที่ 3 ต้องมากกว่า 1.166 N-m

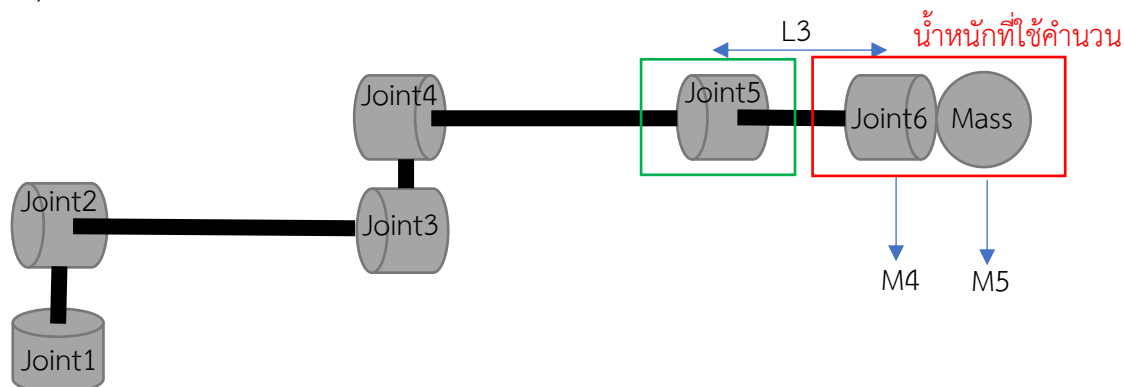
4) การคำนวณแรงขับเคลื่อนที่ Joint 4



รูปที่ 3.5 การหาทอร์ก (Torque) ที่ Joint 4

เนื่องจากแนวของน้ำหนักไม่ตั้งฉากกับแกนหมุน จึงทำให้ไม่สามารถหาคำนวณหาแรงขับเคลื่อนที่ Joint 4 ได้

5) การคำนวณแรงขับเคลื่อนที่ Joint 5



รูปที่ 3.6 การหาทอร์ก (Torque) ที่ Joint 5

จากสมการทอร์ก (Torque) เท่ากับแรงคูณกับระยะทางที่ตั้งฉากจากจุดหมุนถึงแนวแรง

$$\tau = r * F$$

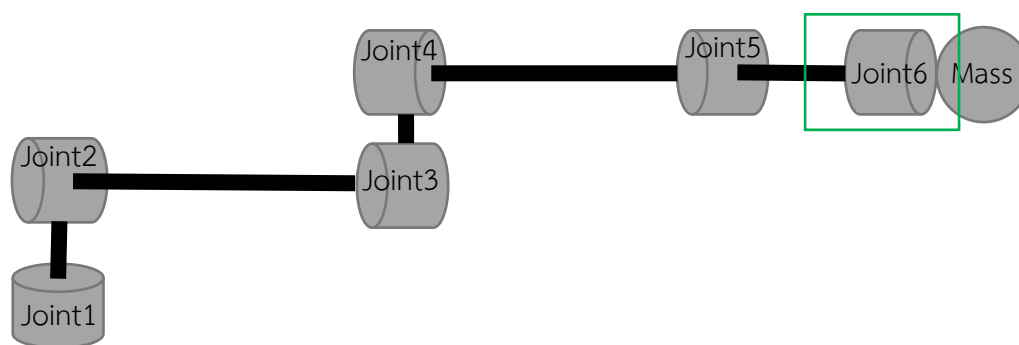
$$\tau = L3 * ((M4 + M5) * g)$$

$$\tau = 0.05 * ((0.20 + 0.50) * 9.8)$$

$$\tau = 0.343 \text{ N-m}$$

ดังนั้นทอร์ก (Torque) ของมอเตอร์ที่จะต้องใช้ในการขับเคลื่อนในจุดหมุนที่ 5 ต้องมากกว่า 0.343 N-m

6) การคำนวณแรงขับเคลื่อนที่ Joint 6



รูปที่ 3.7 การหาทอร์ก (Torque) ที่ Joint 6

เนื่องจากแนวของน้ำหนักไม่ตั้งฉากกับแกนหมุน จึงทำให้ไม่สามารถหาคำนวณหาแรงขับเคลื่อนที่ Joint 6 ได้

3.1.3 การออกแบบแขนหุ่นยนต์อุตสาหกรรม

เมื่อทำการกำหนดขอบเขตของแขนหุ่นยนต์อุตสาหกรรม และคำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรมเสร็จสิ้น จึงทำการออกแบบแขนหุ่นยนต์อุตสาหกรรม โดยแบ่งเป็นแนวทางดังนี้

1) ศึกษาารูปแบบและลักษณะของแขนหุ่นยนต์อุตสาหกรรม

โดยแขนหุ่นยนต์อุตสาหกรรมที่ใช้ศึกษารูปแบบและลักษณะของแขนหุ่นยนต์อุตสาหกรรมคือ Industrial Robot Arm - IRB 6700 150 3.20 ซึ่งเป็นผลงานของ ABB Robotics



รูปที่ 3.8 Industrial Robot Arm - IRB 6700 150 3.20 [13]

2) โปรแกรม Autodesk Inventor Professional

โปรแกรมที่ใช้ในการออกแบบแขนหุ่นยนต์อุตสาหกรรม คือ Autodesk Inventor Professional เป็นซอฟต์แวร์หรือเครื่องมือ สำหรับออกแบบงานเครื่องกลในรูปแบบสามมิติ เขียนแบบเครื่องกล และออกแบบผลิตภัณฑ์ ออกแบบชิ้นส่วนยานยนต์ เครื่องจักรกล เพอร์นิเจอร์ เซรามิค แพคเกจ หีบห่อ โมลด์ แบบหล่อพลาสติก แบบหล่อโลหะ Mold & Die เครื่องกลงานก่อสร้าง งานแผ่นโลหะ โดยใช้ความสามารถในการจำลองต้นแบบดิจิทัลเพื่อพัฒนาและจัดการการออกแบบผลิตภัณฑ์ เพื่อให้ได้งานที่ดีที่สุด [14]



รูปที่ 3.9 Autodesk Inventor Professional [15]

3) แบบของเกียร์ทดรอบ (Gear speed reducer)

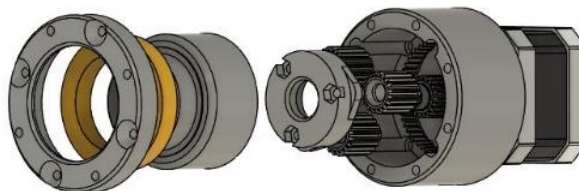
ทำการสืบค้นเพื่อหาแบบของเกียร์ทดรอบ (Gear speed reducer) เพื่อใช้ในการเพิ่มทอร์ก (Torque) ให้กับมอเตอร์ที่ใช้ในการขับเคลื่อนแขนหุ่นยนต์ โดยได้ทำการเลือกใช้เกียร์ทดรอบ (Gear speed reducer) 4 แบบดังนี้

3.1) 80:1 Planetary Gearbox ออกแบบโดย MichaelRechtin

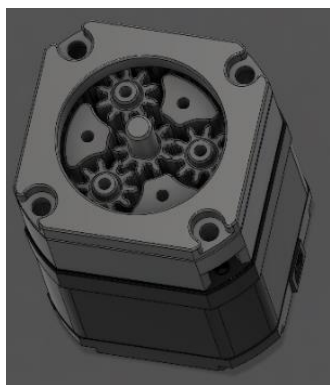
3.2) 4:1 Planetary Gearbox ออกแบบโดย IndeterminateDesign

3.3) 100:1 Stepper motor Planetary Gearbox

3.4) 27:1 Stepper motor Planetary Gearbox



รูปที่ 3.10 รูปของ Planetary Gearbox ซึ่งมีอัตราทดแรง 80:1 ออกแบบโดย MichaelRechtin [16]



รูปที่ 3.11 รูปของ Planetary Gearbox ซึ่งมีอัตราทดแรง 4:1 ออกแบบโดย IndeterminateDesign [17]



รูปที่ 3.12 รูปของ Stepper motor Planetary Gearbox ซึ่งมีอัตราทดแรง 100:1 [20]

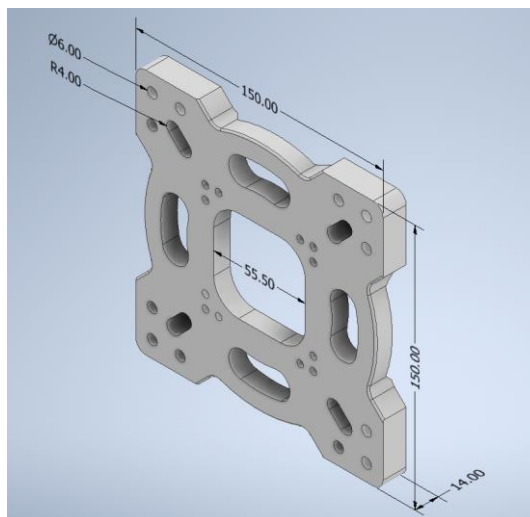


รูปที่ 3.13 รูปของ Stepper motor Planetary Gearbox ซึ่งมีอัตราทดแรง 27:1 [21]

4) แบบของแขนหุ่นยนต์อุตสาหกรรม

ทำการออกแบบแขนหุ่นยนต์อุตสาหกรรมด้วยโปรแกรม Autodesk Inventor Professional โดยมีโมเดลต้นแบบในการออกแบบคือ Industrial Robot Arm - IRB 6700 150 3.20 และความยาวของแขนหุ่นยนต์ (Link) รวมถึงน้ำหนักเป็นขอบเขตในการออกแบบ ซึ่งแยกเป็นส่วนๆ ดังนี้

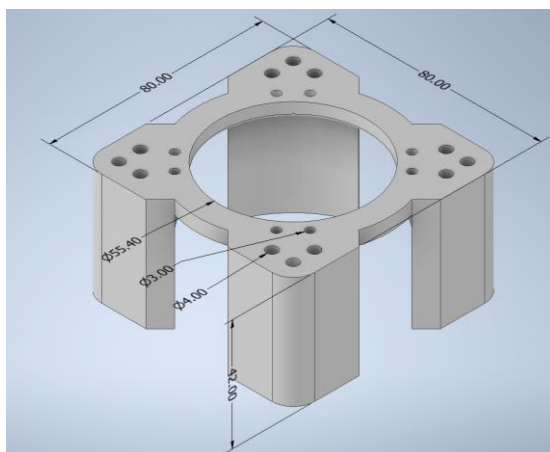
4.1) ฐานยึด



รูปที่ 3.14 ส่วนฐานยึด

จากรูปที่ 3.14 เป็นรูปที่แสดงส่วนที่เป็นส่วนฐานยึดแขนหุ่นยนต์อุตสาหกรรม โดยได้ออกแบบให้สามารถยึดเข้ากับพื้นเพื่อให้แขนหุ่นยนต์อุตสาหกรรมอยู่กับที่

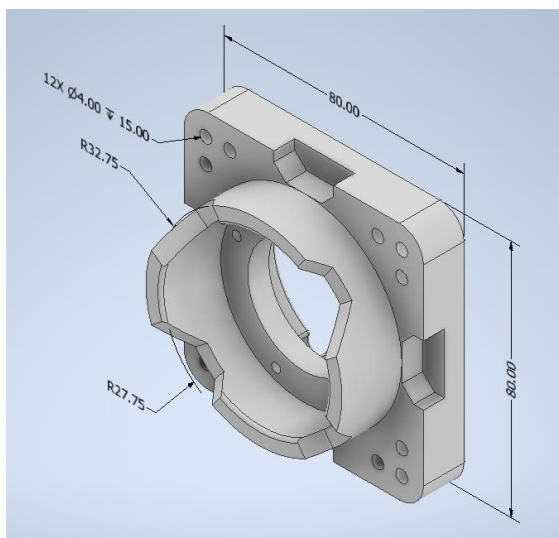
4.2) ฐานส่วนที่ 1



รูปที่ 3.15 ฐานส่วนที่ 1

จากรูปที่ 3.15 เป็นรูปที่แสดงส่วนที่ฐานส่วนที่ 1 ของแขนหุ่นยนต์อุตสาหกรรมซึ่งเป็นส่วนของขาที่ยึดระหว่างฐานยึดและฐานส่วนที่ 2 เข้าด้วยกัน

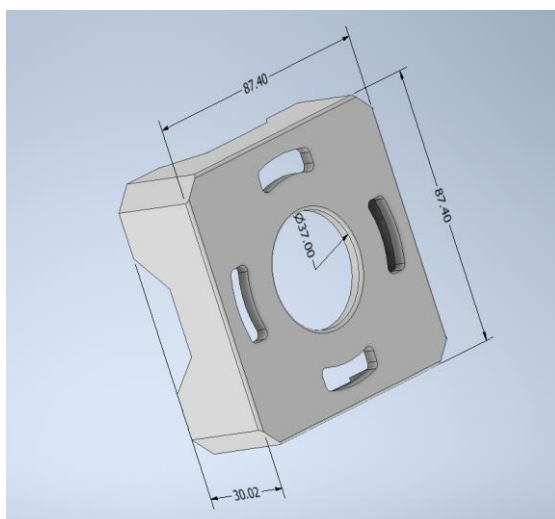
4.11) ฐานส่วนที่ 2



รูปที่ 3.16 ฐานส่วนที่ 2

จากรูปที่ 3.16 เป็นรูปที่แสดงส่วนที่ฐานส่วนที่ 2 ของแขนหุ่นยนต์อุตสาหกรรมซึ่งเป็นส่วนที่ยึดติดกับมอเตอร์ โดยมอเตอร์ดังกล่าวทำหน้าที่ควบคุมการเคลื่อนที่ของจุดหมุนที่ 1 (Joint1) ของแขนหุ่นยนต์อุตสาหกรรม

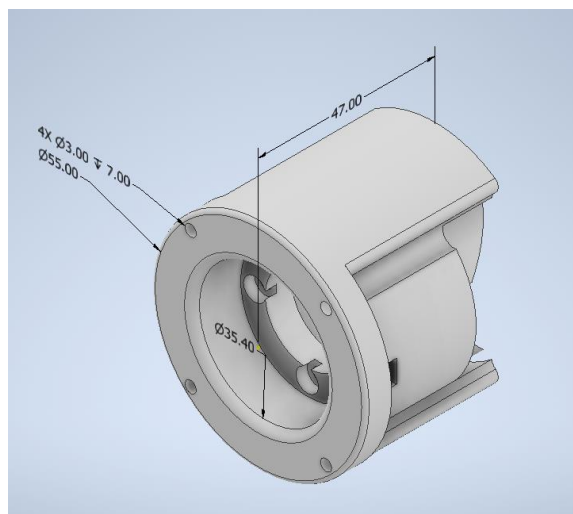
4.11) ฐานส่วนที่ 3



รูปที่ 3.17 ฐานส่วนที่ 3

จากรูปที่ 3.17 เป็นรูปที่แสดงส่วนที่ฐานส่วนที่ 3 ของแขนหุ่นยนต์อุตสาหกรรม โดยเป็นส่วนที่นำมาประกบเข้ากับฐานส่วนที่ 2 เพื่อยึดตลับลูกปืน (Bearing) ไม่ให้หลุดออกจากฐานส่วนที่

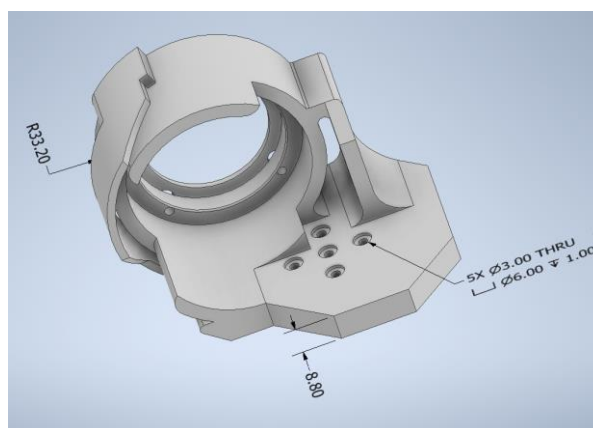
4.12) ส่วนประกอบ Gear box



รูปที่ 3.18 ส่วนประกอบ Gear box

จากรูปที่ 3.18 เป็นรูปที่แสดงส่วนที่เป็นส่วนประกอบ Gear box เนื่องจากต้องการเพิ่มตลับลูกปืน (Bearing) เพื่อลดภาระโหลดที่กระทำต่อแกน Gear box ของ Stepper motor

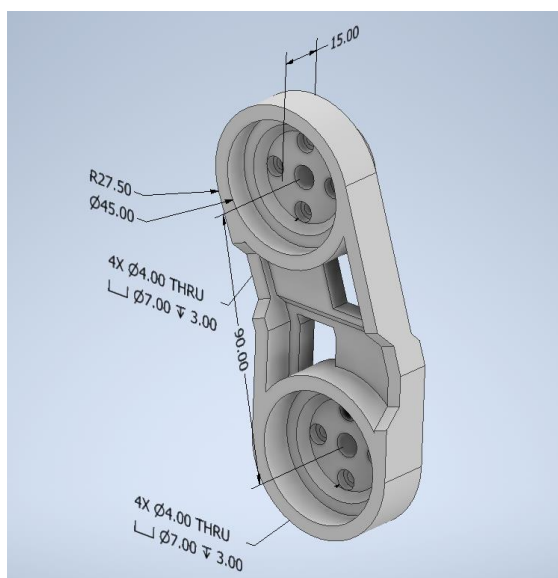
4.3) แขนหุ่นยนต์ส่วนที่ 1



รูปที่ 3.19 แขนหุ่นยนต์ส่วนที่ 1

จากรูปที่ 3.19 เป็นรูปที่แสดงส่วนที่เป็นแขนหุ่นยนต์ส่วนที่ 1 ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นส่วนที่อยู่ของ Stepper motor ที่ทำหน้าที่ขับเคลื่อนจุดหมุนที่ 2 (Joint2)

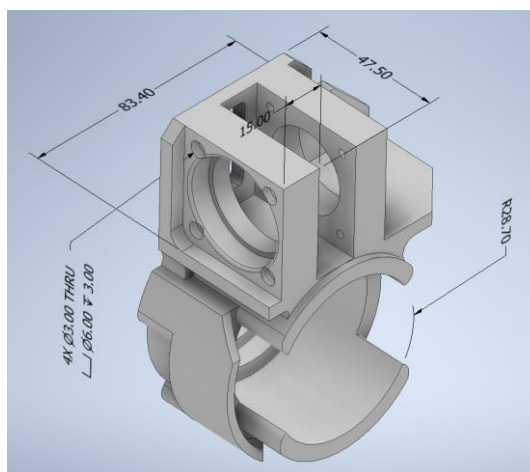
4.4) แขนหุ่นยนต์ส่วนที่ 2



รูปที่ 3.20 แขนหุ่นยนต์ส่วนที่ 2

จากรูปที่ 3.20 เป็นรูปที่แสดงส่วนที่เป็นแขนหุ่นยนต์ส่วนที่ 2 ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นส่วนที่เปรียบเสมือนแขนท่อนบนของมนุษย์

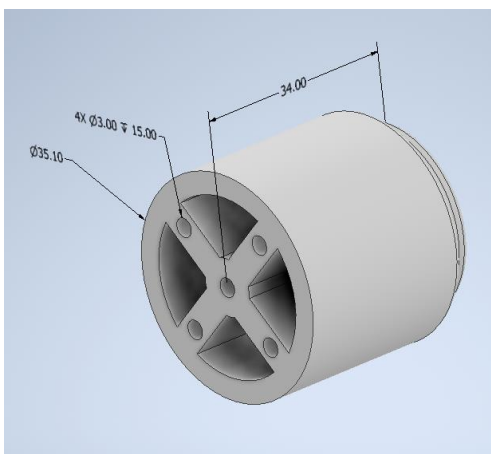
4.5) แขนหุ่นยนต์ส่วนที่ 3



รูปที่ 3.21 แขนหุ่นยนต์ส่วนที่ 3

จากรูปที่ 3.21 เป็นรูปที่แสดงส่วนที่เป็นแขนหุ่นยนต์ส่วนที่ 3 ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นส่วนที่เปรียบเสมือนหัวไหล่ของมนุษย์ เป็นส่วนที่อยู่ของ Stepper motor ที่ทำหน้าที่ขับเคลื่อนจุดหมุนที่ 3 (Joint3) และจุดหมุนที่ 4 (Joint4)

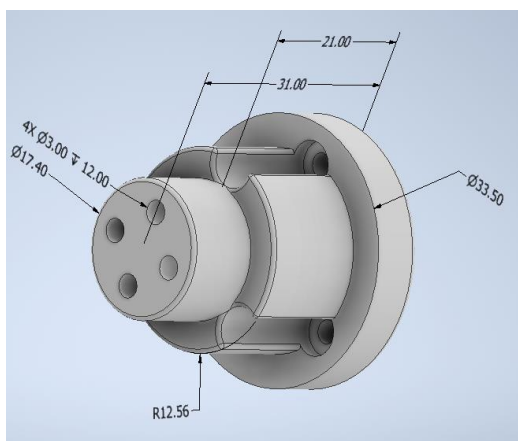
4.6) แกนยึด



รูปที่ 3.22 แกนยึด

จากรูปที่ 3.22 เป็นรูปที่แสดงส่วนที่เป็นแกนยึดระหว่างมอเตอร์ที่ขับเคลื่อนจุดหมุนที่ 1 (Joint1) กับแขนหุ่นยนต์ส่วนที่ 1

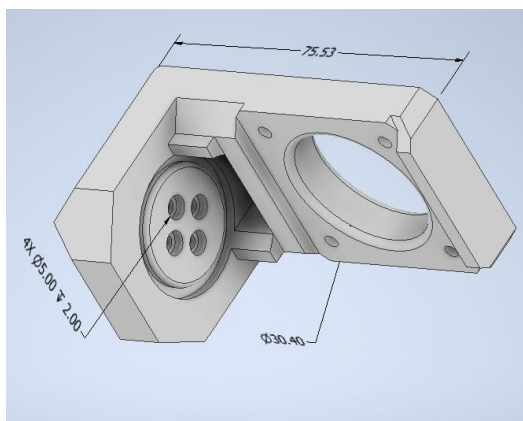
4.7) แขนหุ่นยนต์ส่วนที่ 4



รูปที่ 3.23 แขนหุ่นยนต์ส่วนที่ 4

จากรูปที่ 3.23 เป็นรูปที่แสดงส่วนที่เป็นแขนหุ่นยนต์ส่วนที่ 4 ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นส่วนที่ออกแบบมาเพื่อเชื่อมระหว่างแขนหุ่นยนต์ส่วนที่ 3 และ 6

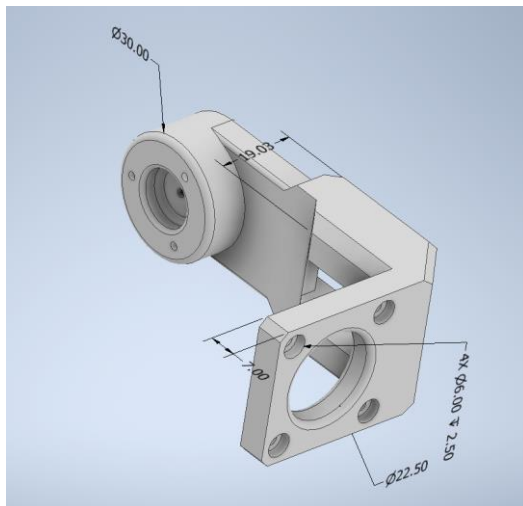
4.8) แขนหุ่นยนต์ส่วนที่ 5



รูปที่ 3.24 แขนหุ่นยนต์ส่วนที่ 5

จากรูปที่ 3.24 เป็นรูปที่แสดงส่วนที่เป็นแขนหุ่นยนต์ส่วนที่ 5 ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นส่วนที่ออกแบบมาเพื่อเชื่อมระหว่างแขนหุ่นยนต์ส่วนที่ 4 และ 7 โดยแขนหุ่นยนต์ส่วนที่ 5 ทำหน้าที่เปรียบเสมือนแขนท่อนล่างของมนุษย์ และเป็นส่วนที่อยู่ของ Stepper motor ที่ทำหน้าที่ขับเคลื่อนจุดหมุนที่ 5 (Joint5)

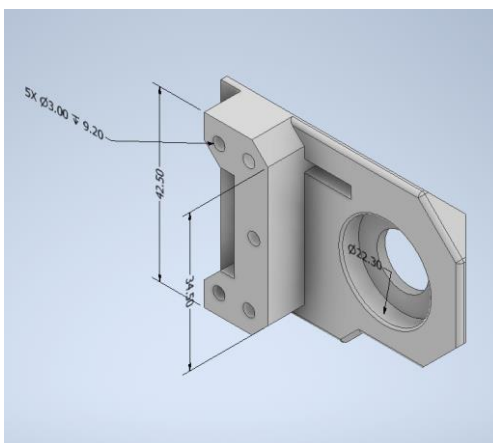
4.9) แขนหุ่นยนต์ส่วนที่ 6



รูปที่ 3.25 แขนหุ่นยนต์ส่วนที่ 6

จากรูปที่ 3.25 เป็นรูปที่แสดงส่วนที่เป็นแขนหุ่นยนต์ส่วนที่ 6 ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นส่วนที่ออกแบบมาเพื่อเชื่อมระหว่าง Stepper motor ที่ทำหน้าที่ขับเคลื่อนข้อต่อที่ 5 (Joint5) และ Stepper motor ที่ทำหน้าที่ขับเคลื่อนจุดหมุนที่ 6 (Joint6)

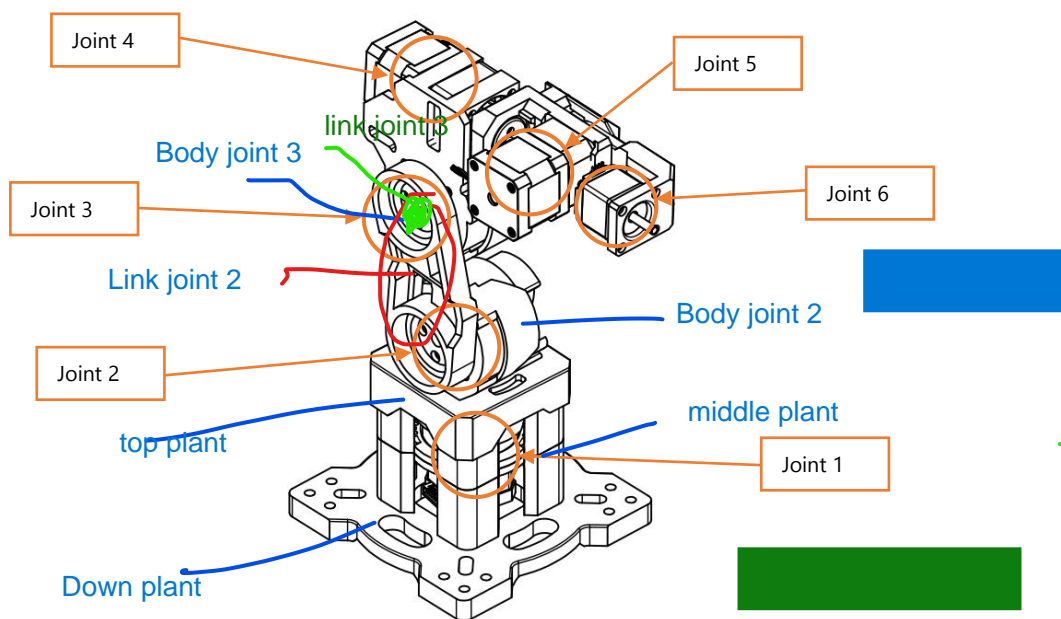
4.10) แขนหุ่นยนต์ส่วนที่ 7



รูปที่ 3.26 แขนหุ่นยนต์ส่วนที่ 7

จากรูปที่ 3.26 เป็นรูปที่แสดงส่วนที่เป็นแขนหุ่นยนต์ส่วนที่ 7 ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นส่วนที่ออกแบบมาเพื่อยึดไม่ให้แขนหุ่นยนต์ส่วนที่ 6 หลุด

5) Gear box ที่ใช้ในแต่ละจุดหมุน (Joint) ของแขนหุ่นยนต์อุตสาหกรรม



รูปที่ 3.27 ภาพแสดงชื่อและตำแหน่งจุดหมุน (Joint)

จากรูปที่ 3.27 ในจุดหมุนที่ 1 (Joint 1) ใช้ Gear box ที่มีอัตราทด 80:1, ในจุดหมุนที่ 2 (Joint 2) ใช้ Gear box ที่มีอัตราทด 100:1, ในจุดหมุนที่ 3 (Joint 3) ใช้ Gear box ที่มีอัตราทด 27:1, ในจุดหมุนที่ 4 (Joint 4) ใช้ Gear box ที่มีอัตราทด 4:1, ในจุดหมุนที่ 5 (Joint 5) ใช้ Gear box ที่มีอัตราทด 4:1 และในจุดหมุนที่ 6 (Joint 6) ไม่ใช้ Gear box

3.1.4 การสร้างแขนหุ่นยนต์อุตสาหกรรม

ในขั้นตอนการสร้างแขนหุ่นยนต์อุตสาหกรรม เป็นขั้นตอนการสร้างชิ้นงานสามมิติของแขนหุ่นยนต์อุตสาหกรรมด้วยเครื่อง 3D Printer โดยเครื่อง 3D Printer ที่ใช้คือ Creality ender 3 โดยมีขั้นตอนดังนี้

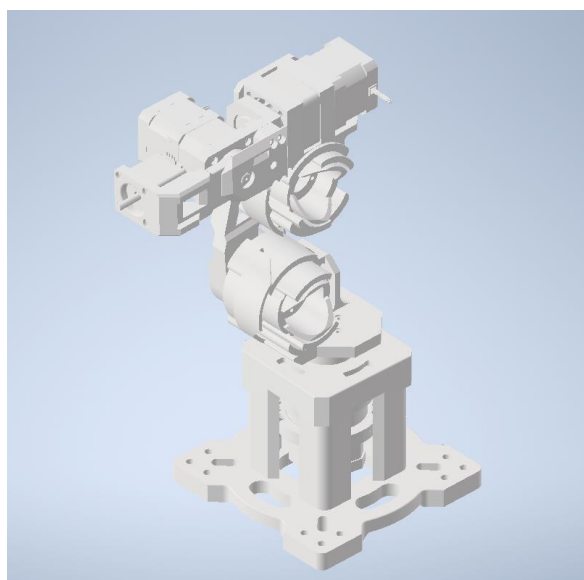
1. ทำการบันทึกไฟล์ชิ้นงานจากโปรแกรม Autodesk Inventor Professional และไฟล์แบบของเกียร์ทดรอบ (Gear speed reducer) จากเว็บไซต์ MakerBot Thingiverse เป็นไฟล์นามสกุล “.STL” (Standard Triangle Language) ซึ่งเป็นไฟล์ที่เก็บข้อมูลสำหรับชิ้นงานสามมิติ

2. ทำการเตรียมเครื่อง 3D Printer สำหรับการสร้างชิ้นงานสามมิติของแขนหุ่นยนต์อุตสาหกรรม และเส้นใยพลาสติกที่ใช้เป็นชนิด PLA ซึ่งจำเป็นต้องปรับอุณหภูมิหัวฉีดของเครื่อง 3D Printer อยู่ระหว่าง 205-220 องศา และอุณหภูมิฐานอยู่ระหว่าง 50-60 องศา

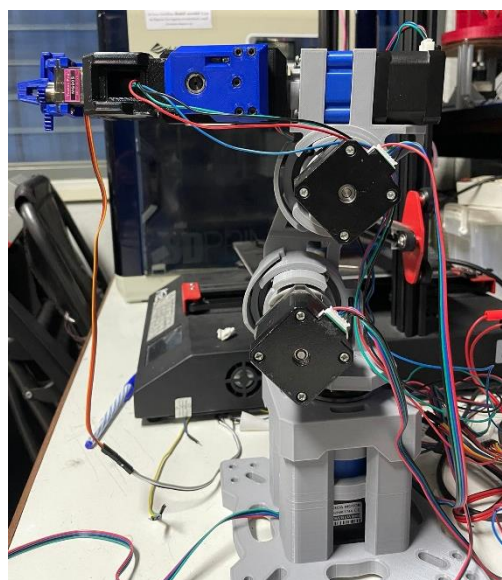
3. ทำการติดตั้งโปรแกรม Ultimaker Cura เพื่อใช้ในการแปลงไฟล์นามสกุล “.STL” เป็นนามสกุล “.gcode” เพื่อใช้สำหรับเครื่อง 3D Printer

4. ทำการสร้างชิ้นงานสามมิติของแขนหุ่นยนต์โดยใช้เครื่อง 3D Printer โดยชิ้นงานสามมิติที่ทำการสร้างโดยใช้เครื่อง 3D Printer

เมื่อทำการสร้างแขนหุ่นยนต์อุตสาหกรรมโดยการสร้างชิ้นงานสามมิติของแขนหุ่นยนต์อุตสาหกรรมด้วยเครื่อง 3D Printer เสร็จสิ้น ก็จะมีการประกอบแขนหุ่นยนต์อุตสาหกรรม โดยจะได้แขนหุ่นยนต์อุตสาหกรรมที่มี มีจำนวน 6 จุดหมุน (Joint) ดังรูปที่ 3.19



ก)

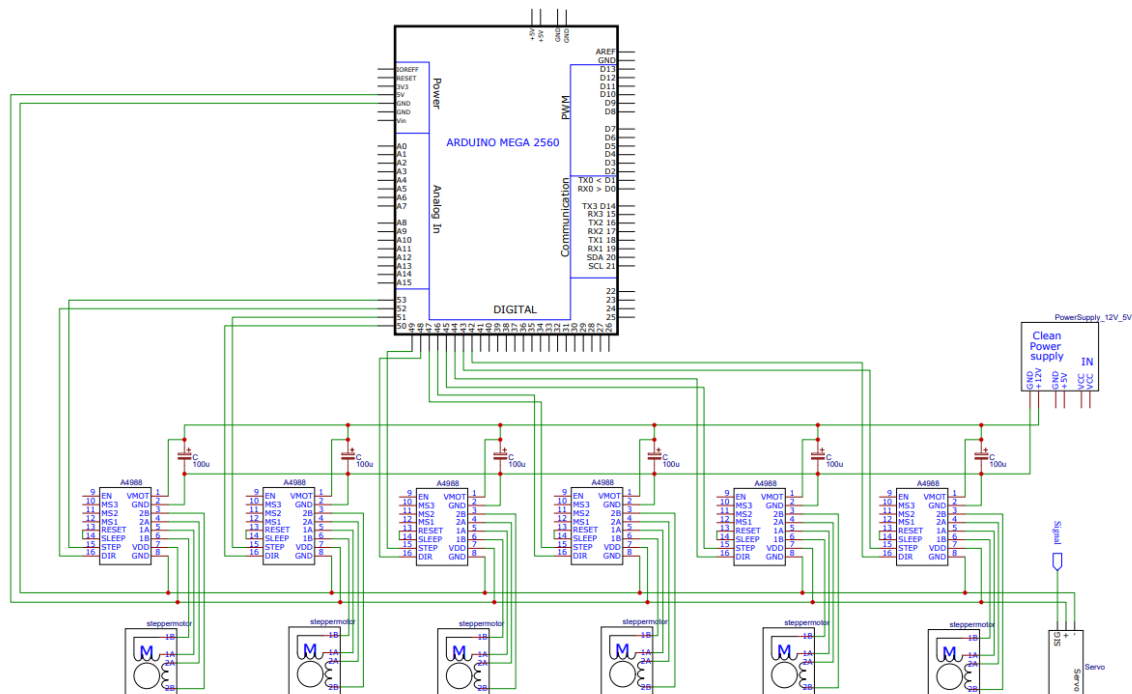


ข)

รูปที่ 3.28 ก) ภาพ Assembly เบื้องต้น ข) ภาพแขนหุ่นยนต์อุตสาหกรรมที่ทำการประกอบเสร็จ

3.1.5 ชุดวงจรควบคุมแขนหุ่นยนต์อุตสาหกรรม

เมื่อทำการสร้างแขนหุ่นยนต์อุตสาหกรรมเสร็จสิ้น ในขั้นตอนต่อมาคือการสร้างคือชุดวงจรเพื่อใช้ในควบคุมแขนหุ่นยนต์อุตสาหกรรม โดยมีชุดวงจรควบคุมแขนหุ่นยนต์อุตสาหกรรมตามรูปที่ 3.29



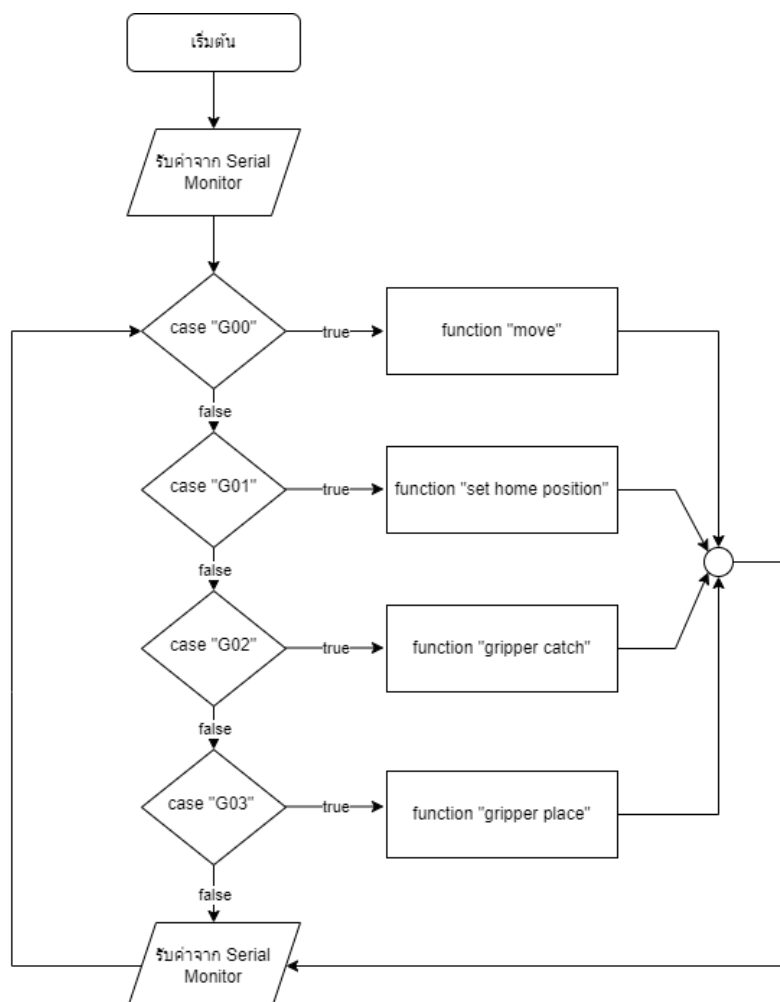
รูปที่ 3.29 ชุดวงจรควบคุมแขนหุ่นยนต์อุตสาหกรรม

จากรูปที่ 3.29 จากการออกแบบกำหนดให้ Stepper Motor ทั้ง 6 ตัว เป็นตัวขับเคลื่อนแขนหุ่นยนต์ในแต่ละข้อต่อ (Joint) มี A4988 เป็นโมดูลบอร์ดสำหรับควบคุม Stepper Motor และใช้ Servo motor เป็น Gripper โดยมี Power supply ขนาด 12 V เป็นแหล่งจ่ายให้กับตัว Stepper Motor ซึ่งบอร์ด Arduino MEGA 2560 เป็นบอร์ดคอนโทรเลอร์ในการสร้างสัญญาณพัลส์ (Pulse) ส่งให้กับโมดูล A4988 เพื่อควบคุม Stepper Motor โดยเมื่อบอร์ด Arduino MEGA 2560 ได้รับค่าองศาจาก Serial Monitor บอร์ด Arduino MEGA 2560 จะทำการแปลงค่าองศาเป็นจำนวน Step และสร้างสัญญาณพัลส์ (Pulse) ตามจำนวน Step ส่งให้กับโมดูล A4988 เพื่อควบคุม Stepper Motor ที่ทำหน้าที่เป็นต้นกำลังขับเคลื่อนแขนหุ่นยนต์อุตสาหกรรมในแต่ละจุดหมุน (Joint) และ Servo motor จะทำหน้าที่ในส่วนที่เป็น Gripper โดยบอร์ด Arduino MEGA 2560 จะเป็นตัวควบคุม Servo motor ในการคืบหรือปล่อยวัตถุ

3.2 ด้านซอฟต์แวร์

3.2.1 โปรแกรมสำหรับควบคุมแขนหุ่นยนต์อุตสาหกรรม

ในการควบคุมแขนหุ่นยนต์อุตสาหกรรมจะใช้ Arduino code ในการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรม โดยหลักการเขียน Arduino code สามารถอธิบายได้ด้วย Flow Chart ดังแสดงในรูป 3.30



รูปที่ 3.30 แผนผัง Flow Chart แสดงการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรม

จากรูปที่ 3.30 เมื่อเริ่มต้นทำงาน จะทำการรับค่าจาก Serial Monitor ของ Arduino IDE เมื่อทำการรับค่า Serial Monitor จะทำการพิจารณาเป็นกรณี หากค่าที่รับมาไม่ตรงกับกรณี (Case) ที่ทำการสร้างไว้ โปรแกรมจะกลับไปรอรับค่าจาก Serial Monitor ใหม่ และหากค่าที่รับมาตรงกับกรณี (Case) ที่ทำการสร้างไว้ โปรแกรมจะทำงานตามฟังก์ชัน (Function) ของแต่ละกรณี และเมื่อโปรแกรมทำงานตามฟังก์ชัน (Function) เสร็จสิ้น โปรแกรมจะกลับไปรอรับค่าจาก Serial Monitor ใหม่ โดยแต่ละกรณี (Case) สามารถอธิบายเพิ่มเติมได้ดังนี้

Case “G00” เมื่อต้องการทำงานในกรณีนี้ให้ป้อนค่าผ่าน Serial Monitor ขึ้นต้นด้วย G00 ตามด้วยค่ามุมในแต่ละข้อต่อ (Joint) แล้วปิดท้ายด้วยเครื่องหมาย ; ตัวอย่างเช่น G00 X45 Y90 Z-45 U90 V0 W25; เป็นต้น โดยที่ X45 คือค่ามุมของจุดหมุนที่ 1 (Joint1) ที่เคลื่อนที่ไป

Y90 คือค่ามุมของจุดหมุนที่ 2 (Joint2) ที่เคลื่อนที่ไป

Z-45 คือค่ามุมของจุดหมุนที่ 3 (Joint3) ที่เคลื่อนที่ไป

U90 คือค่ามุมของจุดหมุนที่ 4 (Joint4) ที่เคลื่อนที่ไป

V0 คือค่ามุมของจุดหมุนที่ 5 (Joint5) ที่เคลื่อนที่ไป

W25 คือค่ามุมของจุดหมุนที่ 6 (Joint6) ที่เคลื่อนที่ไป

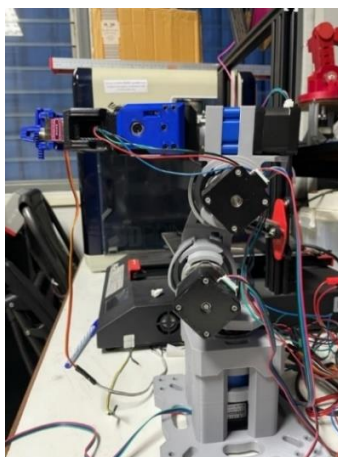
Case “01” เมื่อต้องการทำงานในกรณีนี้ให้ป้อนค่าผ่าน Serial Monitor ขึ้นต้นด้วย G01 แล้วปิดท้ายด้วยเครื่องหมาย ; ตัวอย่างเช่น G01; สำหรับการทำงานในกรณีนี้ เป็นคำสั่งในการกำหนด Home position

Case “02” เมื่อต้องการทำงานในกรณีนี้ให้ป้อนค่าผ่าน Serial Monitor ขึ้นต้นด้วย G02 แล้วปิดท้ายด้วยเครื่องหมาย ; ตัวอย่างเช่น G02; สำหรับการทำงานในกรณีนี้ Gripper จะหนีบเข้า

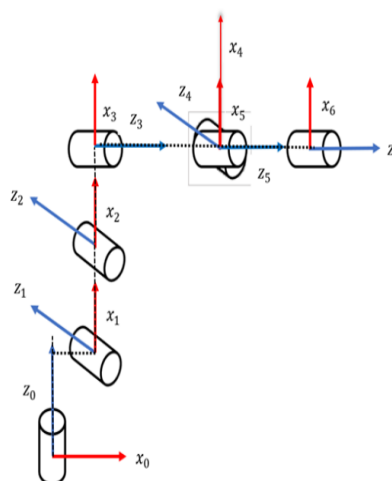
Case “03” เมื่อต้องการทำงานในกรณีนี้ให้ป้อนค่าผ่าน Serial Monitor ขึ้นต้นด้วย G03 แล้วปิดท้ายด้วยเครื่องหมาย ; ตัวอย่างเช่น G03; สำหรับการทำงานในกรณีนี้ Gripper จะคลายออก

3.2.2 โปรแกรมคำนวณหาตำแหน่งจุดปลายและจำลองการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรม

จากการวิเคราะห์โครงสร้างของแขนหุ่นยนต์อุตสาหกรรมแบบ 6 แกนและนำมาเขียนให้อยู่ในรูปแบบทางคณิตศาสตร์ เพื่อให้สามารถคำนวณหาตำแหน่งต่างๆของตำแหน่งส่วนปลายของแขนหุ่นยนต์อุตสาหกรรมแบบ 6 แกนซึ่งจะสามารถเขียนให้อยู่ในรูปแบบของเมทริกซ์การเคลื่อนที่ หรือการแปลงพิกัด และเวกเตอร์ของการบอกตำแหน่งซึ่งจะต้องทำการตั้งแกนของแต่ละจุดหมุน (Joint) ของแขนหุ่นยนต์อุตสาหกรรมเพื่อทำการสร้างแบบจำลองทางคณิตศาสตร์ดังแสดงไว้ในรูปที่ 3.32 โดยกำหนดให้แกน z คือแกนหมุนของแต่ละจุดหมุน (Joint) ของแขนหุ่นยนต์อุตสาหกรรม



รูปที่ 3.31 แขนหุ่นยนต์อุตสาหกรรม



รูปที่ 3.32 การตั้งแกนเพื่อสร้างแบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรม

เมื่อทำการตั้งแกนของแขนหุ่นยนต์อุตสาหกรรม 6 แกน ได้เรียบร้อยแล้วจึงได้ทำการสร้างตารางแสดงความสัมพันธ์ของแกนต่างๆ โดยอาศัยหลักการการเชื่อมต่อทุกตัวเพื่อที่จะใช้หาเมทริกซ์ ที่ใช้แทนความสัมพันธ์ของจุดหมุน (Joint) ต่างๆ ที่เชื่อมต่อกันซึ่งเมทริกซ์นี้จะใช้หาสมการจลศาสตร์แปรผัน (Forward Kinematic) ตามหลักการของ Denavit Hartenberg โดยเมทริกซ์นี้จะอธิบายถึงคุณสมบัติในการเคลื่อนไหวของหุ่นยนต์โดยอาศัยค่าตัวแปรต่างๆ ที่วัดได้จากแขนหุ่นยนต์อุตสาหกรรม เพื่อให้สามารถนำความสัมพันธ์ของระยะและมุมต่างๆ ของแต่ละจุดหมุน (Joint) นำไปคำนวณและสร้างแบบจำลองทางคณิตศาสตร์เพื่อจำลองการเคลื่อนที่และหาตำแหน่งจุดปลายของแขนหุ่นยนต์อุตสาหกรรม โดยสามารถเขียนเมทริกซ์กันไปพิกัดระหว่างการโยงที่ i และ $i-1$ โดยอาศัยหลักการของ Denavit Hartenberg ตามตารางที่ 3.3

ตารางที่ 3.3 ตารางข้อมูลแบบ Denavit Hartenberg ของแขนหุ่นยนต์อุตสาหกรรม 6 แกน

i	θ_{i-1}	d_{i-1}	a_i	α_i
1	q_1	d_1	0	$-\frac{\pi}{2}$
2	$q_2 - \frac{\pi}{2}$	0	a_2	0
3	q_3	0	a_3	$\frac{\pi}{2}$
4	q_4	d_4	0	$-\frac{\pi}{2}$
5	q_5	0	0	$\frac{\pi}{2}$
6	q_6	d_6	0	π

จากนั้นนำค่าที่ได้จากการสร้างตารางของ Denavit Hartenberg แทนค่าลงไปเมทริกซ์การแปลงพิกัดตามสมการที่ (1)

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & \cos\alpha_i & \sin\theta_i\sin\alpha_i & \alpha_i\cos\alpha_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & \alpha_i\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0_{1 \times 3} & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

เมื่อ

a_i คือ ระยะห่างแกน z_i และ z_{i-1} ตามแนวแกน x_i

d_i คือ ระยะห่างแกน z_i และ x_{i-1} ตามแนวแกน z_{i-1}

α_i คือ มุมระยะห่างแกน z_i และ z_{i-1}

θ_i คือ มุมระยะห่างแกน z_i และ x_{i-1}

เมื่อแทนค่า a_i, d_i, α_i และ θ_i ลงไปในเมทริกซ์การแปลงพิกัด ดังนั้นจะสามารถเขียนเมทริกซ์การแปลงพิกัดได้ใหม่ดังนี้

$$T_1^0 = \begin{bmatrix} c_{q_1} & 0 & -s_{q_1} & c_{q_1}a_1 \\ s_{q_1} & 0 & -c_{q_1} & s_{q_1}a_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$T_2^1 = \begin{bmatrix} c_{q_2-\frac{p_i}{2}} & -s_{q_2-\frac{p_i}{2}} & 0 & c_{q_2-\frac{p_i}{2}}a_2 \\ s_{q_2-\frac{p_i}{2}} & c_{q_2-\frac{p_i}{2}} & 0 & s_{q_2-\frac{p_i}{2}}a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$T_3^2 = \begin{bmatrix} c_{q_3} & 0 & s_{q_3} & c_{q_3}a_3 \\ s_{q_3} & 0 & -c_{q_3} & s_{q_3}a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$T_4^3 = \begin{bmatrix} c_{q_4} & 0 & -s_{q_4} & 0 \\ s_{q_4} & 0 & c_{q_4} & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$T_5^4 = \begin{bmatrix} c_{q_5} & 0 & -s_{q_5} & 0 \\ s_{q_5} & 0 & c_{q_5} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

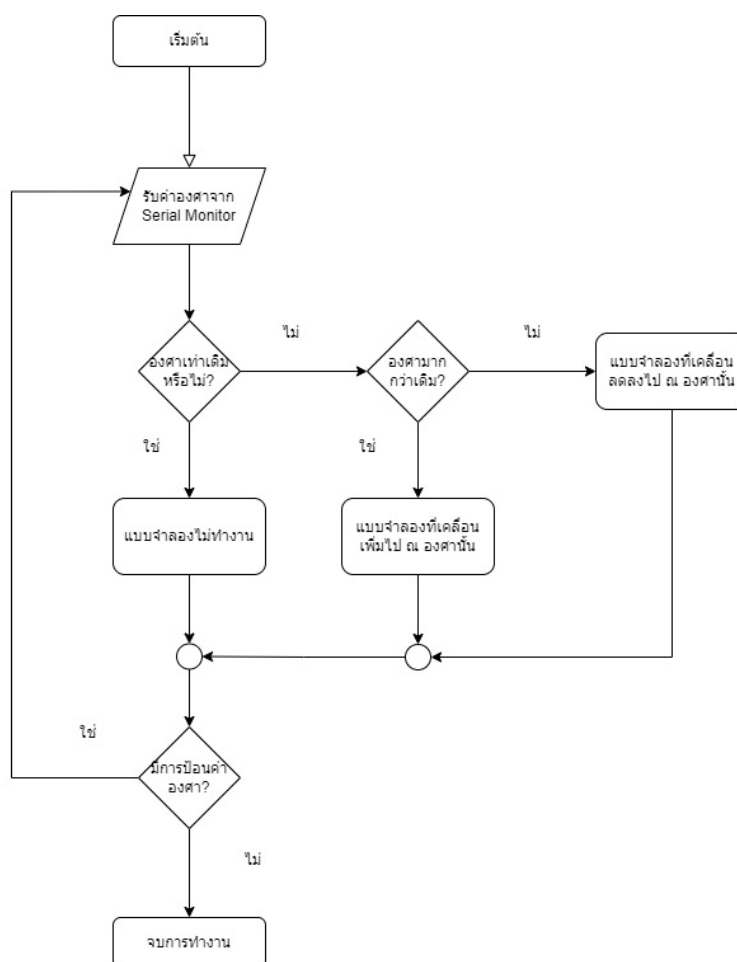
$$T_6^5 = \begin{bmatrix} c_{q_6} & s_{q_6} & 0 & 0 \\ s_{q_6} & -c_{q_6} & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$$T_7^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 \cdot T_6^5 \cdot T_7^6 = \begin{bmatrix} \mu_x & O_x & \alpha_x & p_x \\ \mu_y & O_y & \alpha_y & p_y \\ \mu_z & O_z & \alpha_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

โดยอาศัยหลักการ Homogeneous Transformation Matrix จะได้สมการจลศาสตร์แปรผัน (Forward Kinematic) ของปลายแขนหุ่นยนต์อุตสาหกรรม 6 ซึ่งคำนวณตำแหน่งของแขนหุ่นยนต์อุตสาหกรรมจะเป็นฟังก์ชันที่แสดงโดยเมทริกซ์ดังสมการที่ (3.9)

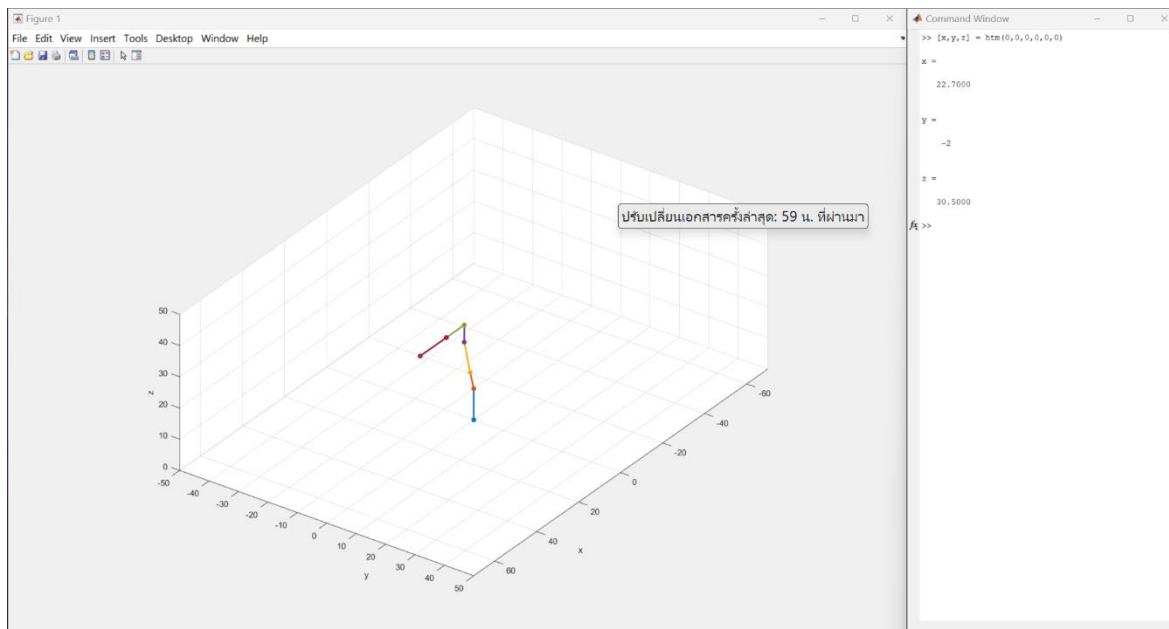
$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \quad (3.9)$$

จากนั้นจึงนำข้อมูลที่ได้นำไปใช้สร้างแบบจำลองทางคณิตศาสตร์ด้วย MATLAB โดยใช้สมการจลศาสตร์แปรผัน (Forward kinematic) ของแขนหุ่นยนต์อุตสาหกรรมแบบ 6 แกนตามสมการและขั้นตอนที่แสดงไว้ในบทที่สองเพื่อใช้้องศาการหมุนของแต่ละจุดหมุน (Joint) เพื่อคำนวณหาตำแหน่งของจุดปลายแขนหุ่นยนต์อุตสาหกรรม โดยใช้ซอฟต์แวร์ MATLAB เข้ามาช่วยในการคำนวณทางคณิตศาสตร์เพื่อช่วยให้ผลการคำนวณทางคณิตศาสตร์มีความรวดเร็วและแม่นยำมากยิ่งขึ้นโดยมีการทำงานของโปรแกรมตาม Flowchart ที่แสดงดังนี้

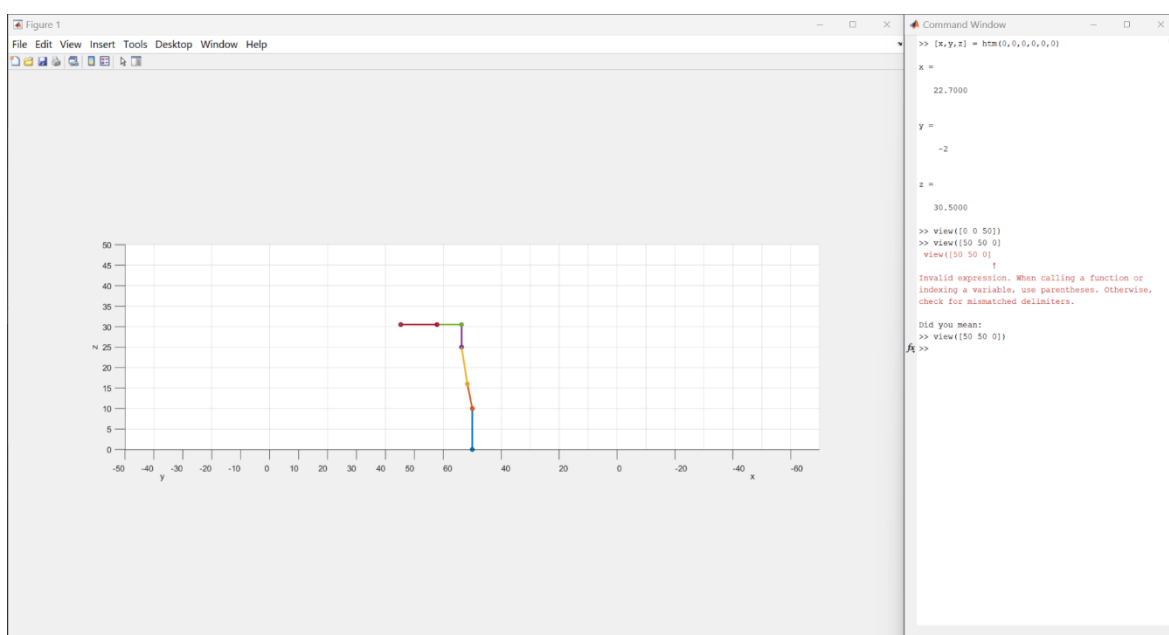


รูปที่ 3.33 แผนผัง Flow Chart แสดงการทำงานของแบบจำลองทางคณิตศาสตร์

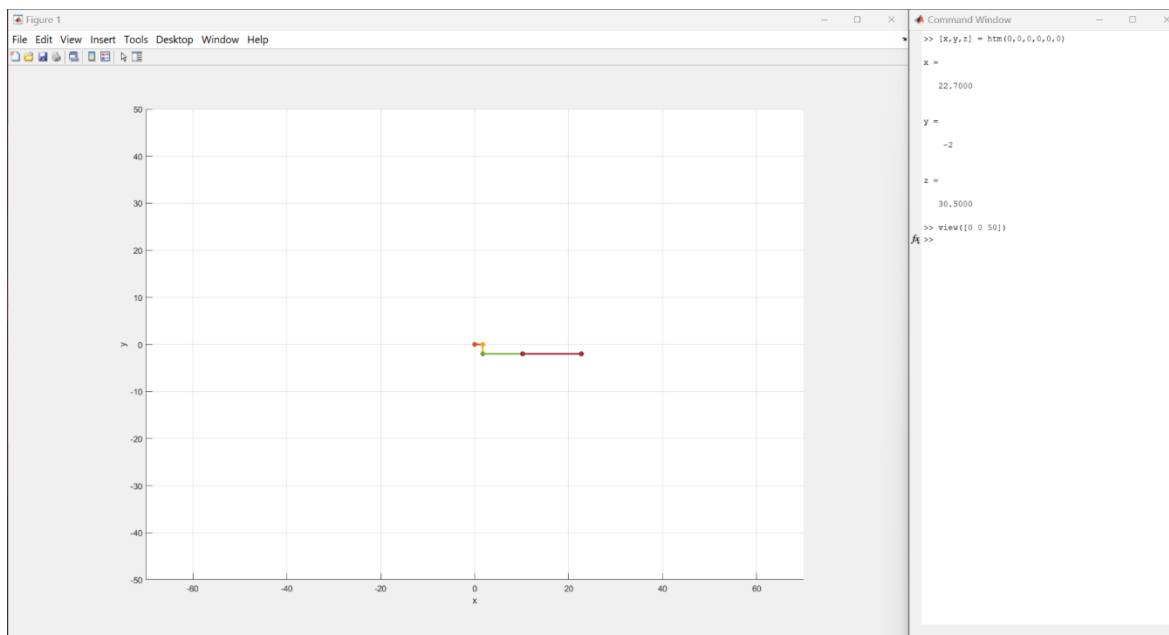
โดยการทำให้แบบจำลองนั้นจะทำได้โดยการป้อนค่าองศาทั้ง 6 จุดหมุนอิสระเข้าไปในโปรแกรมจำลองทางคณิตศาสตร์บน command console ผ่านฟังก์ชันที่กำหนดไว้ หลังจากนั้นโปรแกรมจะทำการจำลองและเคลื่อนที่ไป ณ ตำแหน่งที่เราป้อนค่าเข้าไป ตัวอย่าง เช่น ป้อนค่าองศาทุกจุดหมุน (Joint) เป็น 0 ซึ่งก็คือป้อนค่าเป็น $(x, y, z) = \text{htm}(0, 0, 0, 0, 0, 0)$ จะได้แบบจำลองเป็นไปตามรูปที่ 3.34 ซึ่งเป็นอาศาเริ่มต้นหรือ Home position ของแขนหุ่นยนต์อุตสาหกรรม



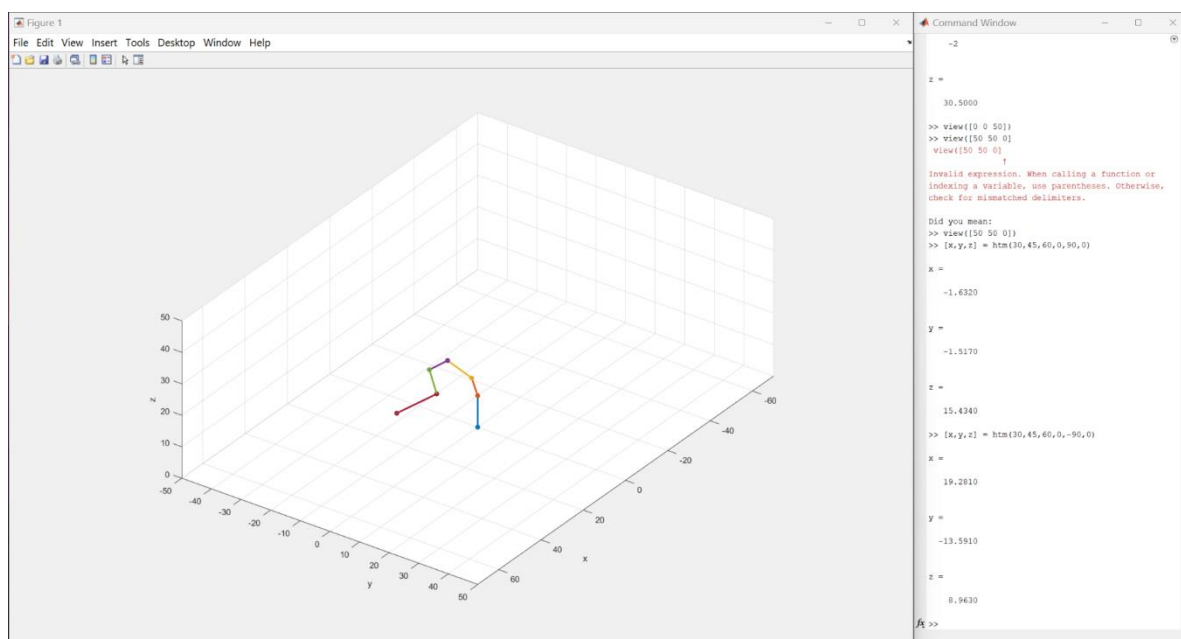
รูปที่ 3.34 การสร้างแบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย
MATLAB บนแกน 3 มิติ



รูปที่ 3.35 การสร้างแบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระนาบ
X,Z

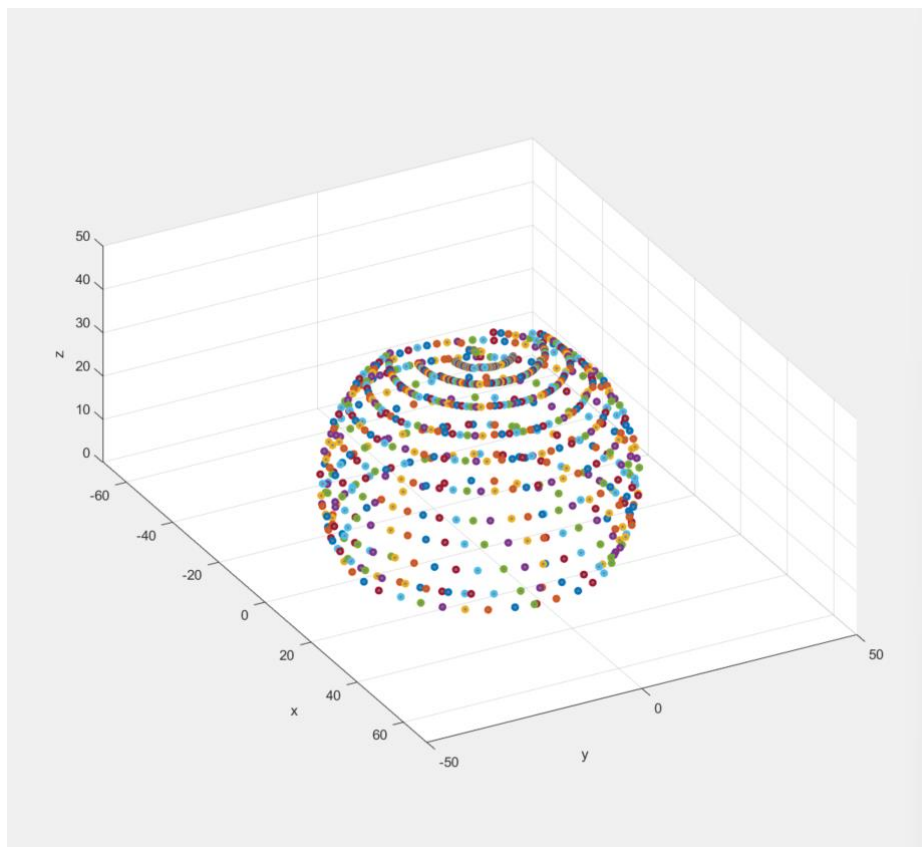


รูปที่ 3.36 การสร้างแบบจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย
MATLAB บนระนาบ X,Y

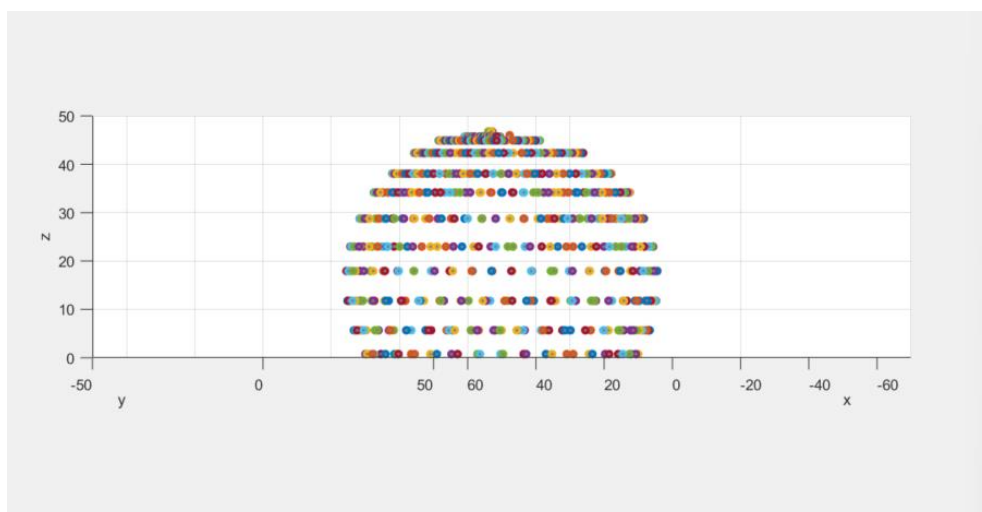


รูปที่ 3.37 ผลการจำลองการเคลื่อนที่และคำนวณค่าจุดปลายของแขนหุ่นยนต์
อุตสาหกรรม

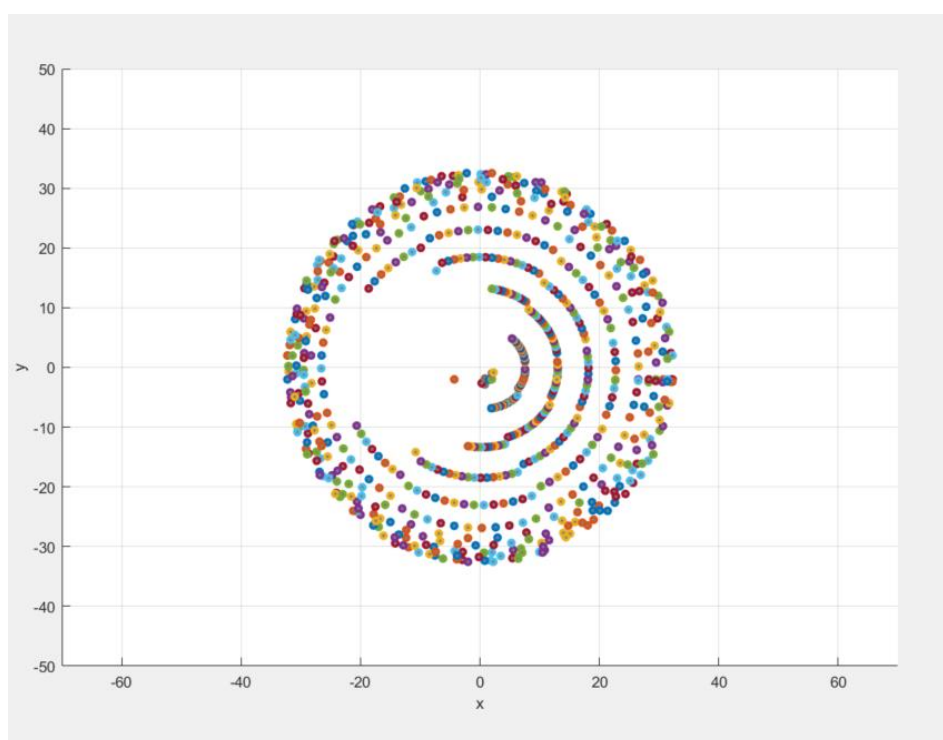
หลังจากทำการทดลองจำลองทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB จึงทำการหาขอบเขตการเคลื่อนในแต่ละแกน เนื่องจากเป็นแขนหุ่นยนต์อุตสาหกรรมชนิดที่ติดตั้งอยู่กับที่จึงมีขีดจำกัดทางด้านการเคลื่อนที่อยู่โดยจากตารางที่ 3.1 ซึ่งจะเป็นการบอกขอบเขตการทำงานของจุดหมุน (Joint) ของแขนหุ่นยนต์อุตสาหกรรมแบบ 6 แกน โดยระยะสูงสุดที่ตำแหน่งจุดปลายของแขนหุ่นยนต์อุตสาหกรรมในแต่ละแกนจะเท่ากับ $X = [31.7, -31.7]$, $Y = [31.7, -31.7]$ $Z = [46, -14]$ จะสามารถสามารถรู้พื้นที่การใช้งานของหุ่นยนต์ได้จากรูปที่ 3.38 ถึง รูปที่ 3.40



รูปที่ 3.38 ผลการจำลองขอบเขตการเคลื่อนที่ทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระบบพิกัด (X,Y,Z) ใน 3 มิติ



รูปที่ 3.39 ผลการจำลองขอบเขตการเคลื่อนที่ทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระนาบพิกัด (X,Z)



รูปที่ 3.40 ผลการจำลองขอบเขตการเคลื่อนที่ทางคณิตศาสตร์ของแขนหุ่นยนต์อุตสาหกรรมด้วย MATLAB บนระนาบพิกัด (X,Y)

บทที่ 4 การทดลองและผลการทดลอง

การทดลองนี้มีวัตถุประสงค์เพื่อทดสอบประสิทธิภาพการควบคุมการทำงานของแขนหุ่นยนต์อุตสาหกรรมและทดลองแบบจำลองทางคณิตศาสตร์ของแขนกลอุตสาหกรรมโดยในบทนี้จะอธิบายถึงผลที่ได้จากการทดลองด้วยการป้อนองศาการเคลื่อนที่ของแต่ละจุดหมุน (Joint) ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งจะแบ่งการทดลองออกเป็น 2 ส่วนคือ

- 4.1 การทดลองควบคุมการเคลื่อนที่เพื่อกำหนดตำแหน่งจุดปลายของแขนหุ่นยนต์แขนกลอุตสาหกรรม
- 4.2 การทดลองหาน้ำหนัก (Load) สูงสุดของแขนหุ่นยนต์อุตสาหกรรม

4.1 การทดลองควบคุมการเคลื่อนที่เพื่อกำหนดตำแหน่งจุดปลายของแขนหุ่นยนต์อุตสาหกรรม

เป็นการทดลองควบคุมการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรม โดยกำหนดองศาในแต่ละจุดหมุนเพื่อกำหนดตำแหน่งจุดปลายของหุ่นยนต์แขนกลอุตสาหกรรมตามทฤษฎีจลนศาสตร์แปรผัน (Forward Kinematic) และนำมาเปรียบเทียบกับค่าคำนวณในแบบจำลองคณิตศาสตร์เพื่อหาค่าความคลาดเคลื่อน ซึ่งมีวิธีการทดลองคือ

1. ทำการติดตั้งกระดากกราฟบนระนาบแนวระดับและกำหนดแกนและขอบเขต
2. ทำการติดตั้งหุ่นยนต์แขนกล ณ ตำแหน่งจุดเริ่มต้นของการทดลอง
3. กำหนดค่าองศาเริ่มต้นในแต่ละจุดหมุน ณ ที่นี้คือองศาที่ 0 หรือ Home position
4. ส่งค่าองศาไปยังแบบจำลองทางคณิตศาสตร์และแต่ละจุดหมุนของแขนหุ่นยนต์อุตสาหกรรมเพื่อให้เคลื่อนที่จำนวนอย่างละ 3 รอบ
5. บันทึกค่าองศาที่ส่งไปและค่าตำแหน่งจุดปลายของแขนหุ่นยนต์อุตสาหกรรม
6. นำค่าตำแหน่งจุดปลายของแขนกลอุตสาหกรรมที่บันทึกได้และนำมาเปรียบเทียบกับค่าในแบบจำลองทางคณิตศาสตร์ เพื่อหาค่าความคลาดเคลื่อน
7. ทำการทดลองซ้ำ โดยส่งค่าองศาในแต่ละจุดหมุนที่แตกต่างกันจำนวน 3 ตำแหน่ง ซึ่งจะได้ผลการทดลองตามตารางที่ 4.1

ตารางที่ 4.1 ตารางแสดงการเปรียบเทียบค่าตำแหน่ง ณ จุดปลายที่วัดได้จริงกับตำแหน่งที่ได้จากแบบจำลองทางคณิตศาสตร์เพื่อหาค่าความคลาดเคลื่อน

มุมที่กำหนด $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$	ตำแหน่งที่วัดได้จริง X, Y, Z (cm)	ตำแหน่งจาก แบบจำลอง X, Y, Z (cm)	ค่าความคลาด เคลื่อน X, Y, Z (%)
0, 0, 0, 0, 0, 0	22.7, -2, 30.5	22.7, -2, 30.5	0, 0, 0
	22.7, -2, 30.5		0, 0, 0
	22.7, -2, 30.5		0, 0, 0
30, 30, 30, 30, 30, 30	14.6, -7.4, 7.5	14.55, -7.25, 7.18	0.34, 2, 4.45
	14.9, -7.5, 7.5		0.34, 2.06, 4.45
	15.1, -7.1, 7		3.78, 2.06, 2.5
45, 45, 45, 45, 45, 45	8, -2, 4.7	8.23, -2.02, 4.58	2.79, 0.99, 2.6
	8.3, -2.1, 4.7		0.85, 3.9, 2.6
	8.6, -2, 4.5		4.49, 0.99, 1.7
30, 45, 60, 75, 90, 105	12, 4.7, 13	12.16, 4.47, 13.04	1.3, 5.1, 0.3
	12.5, 4.5, 13		2.7, 0.6, 0.3
	12, 4.5, 13		1.3, 0.6, 0.3

จากการผลการทดลองสรุปได้ว่าตำแหน่งที่วัดได้จริงเปรียบเทียบกับตำแหน่งจากแบบจำลอง จะเห็นว่ามีค่าความใกล้เคียงกัน โดยมีค่าความคลาดเคลื่อนอยู่ที่ไม่เกิน 5 เปอร์เซ็นต์ หรือไม่เกิน 20 มิลลิเมตร ซึ่งค่าความคลาดเคลื่อนนั้นอาจมาจากทั้งการวัดของผู้ทำการทดลองและเครื่องมือที่ใช้การทดลอง

4.2 การทดลองหาน้ำหนัก (Load) สูงสุดของแขนหุ่นยนต์อุตสาหกรรม

เป็นการทดลองเพื่อหาน้ำหนักโหลด (Load) สูงสุดของแขนหุ่นยนต์อุตสาหกรรมที่สามารถรับได้ โดยมีขั้นตอนการทำการทดลองคือ ให้แขนหุ่นยนต์อุตสาหกรรมหยิบวัตถุที่มีน้ำหนักต่างกัน โดยเริ่มจาก 50 กรัม และเพิ่มขึ้นครั้งละ 50 กรัม จนกว่าแขนหุ่นยนต์อุตสาหกรรมจะสูญเสียการควบคุม โดยมีผลการทดลองดังต่อไปนี้

ตารางที่ 4.3 การทดลองหาน้ำหนัก (Load) สูงสุดของแขนหุ่นยนต์อุตสาหกรรม

น้ำหนัก	ผลลัพธ์
50	✓
100	✓
150	✓
200	✓
250	✗

หมายเหตุ ✓ คือ แขนหุ่นยนต์อุตสาหกรรมสามารถรับโหลดได้โดยไม่สูญเสียการควบคุม
 ✗ คือ แขนหุ่นยนต์อุตสาหกรรมไม่สามารถรับโหลดได้และสูญเสียการควบคุม

จากผลการทดลองว่าแขนหุ่นยนต์อุตสาหกรรมสามารถรับน้ำหนัก (Load) สูงสุดคือ 200 กรัม โดยไม่สูญเสียการควบคุม

บทที่ 5 สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานทางคณะผู้จัดทำได้ดำเนินการตามขั้นตอนการดำเนินงานที่ได้วางแผนไว้ คือการกำหนดขอบเขตของแขนหุ่นยนต์อุตสาหกรรม, การคำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรม, การออกแบบแขนหุ่นยนต์อุตสาหกรรม และการสร้างแขนหุ่นยนต์อุตสาหกรรม ซึ่งเป็นขั้นตอนพื้นฐานในการออกแบบและสร้างแขนหุ่นยนต์อุตสาหกรรม

การออกแบบแขนหุ่นยนต์อุตสาหกรรม โดยแขนหุ่นยนต์อุตสาหกรรมที่ทำการออกแบบจะคำนึงถึงขอบเขตของงานที่เราตั้งไว้, ความยาวของแขนหุ่นยนต์ (Link), น้ำหนักโดยประมาณ และรวมถึงน้ำหนักมอเตอร์ที่จะนำมาใช้ ล้วนมีผลต่อการคำนวณทอร์ก (Torque) ของแขนหุ่นยนต์อุตสาหกรรม ซึ่งจะเป็นการระบุทอร์ก (Torque) ของมอเตอร์ที่จะต้องใช้ในการขับเคลื่อนหุ่นยนต์อุตสาหกรรม

การประยุกต์ใช้แบบจำลองแขนหุ่นยนต์อุตสาหกรรมที่ถูกสร้างขึ้นเพื่อศึกษาส่วนประกอบ กลไก ทฤษฎีที่เกี่ยวข้อง และหลักการทำงานของหุ่นยนต์อุตสาหกรรม ตลอดจนศึกษาการควบคุมแขนหุ่นยนต์อุตสาหกรรม เพื่อที่จะสามารถพัฒนาและต่อยอดองค์ความรู้ในระดับอุตสาหกรรมได้ในอนาคต รวมถึงการสร้างการจำลองการเคลื่อนที่แบบเสมือนด้วยโปรแกรม MATLAB ซึ่งทำให้ผู้ใช้งานสามารถกำหนดค่าพิกัดที่ต้องการในแต่ละจุดหมุน (Joint) และทำการประมวลผลการทำงานของโปรแกรมจำลองทางคณิตศาสตร์เพื่อจำลองการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรม ทำให้เห็นภาพการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรมและตำแหน่งจุดปลายของแขนหุ่นยนต์อุตสาหกรรม

5.2 ปัญหาในการดำเนินงานและแนวทางการแก้ไข

จากการดำเนินงานทำให้พบปัญหาและมีแนวทางการแก้ไขปัญหาดังกล่าวดังนี้

1. การใช้งานการออกแบบแขนหุ่นยนต์อุตสาหกรรมด้วยโปรแกรม Autodesk Inventor Professional ซึ่งเป็นโปรแกรมที่เพิ่งใช้งานเป็นครั้งแรก จึงทำให้ใช้เวลาในการเรียนรู้และใช้งานที่เป็นระยะเวลานาน
2. มีข้อจำกัดต่างๆ มากมายในการสร้างแขนหุ่นยนต์อุตสาหกรรม ดังนั้นจึงต้องคำนึงข้อจำกัดต่างๆ ในการออกแบบแขนหุ่นยนต์อุตสาหกรรม เช่น ข้อจำกัดของเครื่อง 3D Printer, ข้อจำกัดของงบประมาณในการสร้าง เป็นต้น
4. การสร้างชิ้นงานสามมิติด้วยเครื่อง 3D Printer ทำให้ขนาดชิ้นงานมีความคลาดเคลื่อน จึงทำให้ประกอบชิ้นงานไม่ได้ ดังนั้นจึงทำการทดลองสร้างชิ้นงานสามมิติด้วยเครื่อง 3D Printer เพื่อหาระยะห่างระหว่างชิ้นงานที่เหมาะสมของชิ้นงาน
5. การควบคุม Stepper motor โดยควบคุมแบบเปิด (Open Loop Control) ควรพิจารณา Step size ของ Stepper motor หากมีการใช้ Gearbox หรือการทดรอบที่มีอัตราส่วนที่หาร Step size

ลงตัวจะทำให้การควบคุมแบบเปิด (Open Loop Control) มีความแม่นยำมากขึ้น และมีความคลาดเคลื่อนน้อยลง

5.3 ข้อเสนอแนะ

1. เนื่องจากการแขนหุ่นยนต์อุตสาหกรรมแบบเปิด (Open Loop Control) จึงเป็นไปได้ยากที่จะทราบว่าแขนหุ่นยนต์อุตสาหกรรมมีความคลาดเคลื่อนจากตำแหน่งที่ป้อนเข้าไปมากน้อยเพียงใด หากมีการนำตัวเข้ารหัส (Encoder) เข้ามาใช้งานร่วมกับแขนหุ่นยนต์อุตสาหกรรม ก็จะเพิ่มประสิทธิภาพการทำงานได้ดียิ่งขึ้น

2. ในการจำลองการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรม หากสามารถนำโมเดลต้นแบบของแขนหุ่นยนต์อุตสาหกรรมที่ทำการออกแบบมาจำลองการเคลื่อนที่ได้ ก็จะทำให้เห็นภาพการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรมได้ดียิ่งขึ้น

3. แขนหุ่นยนต์อุตสาหกรรมดังกล่าวเป็นการประยุกต์ใช้จลนศาสตร์แปรผัน (Forward Kinematic) ซึ่งสามารถประยุกต์ใช้กับจลนศาสตร์ผกผัน (Inverse Kinematic) ได้ และสามารถพัฒนาระบบในการกำหนดเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์อุตสาหกรรมได้

เอกสารอ้างอิง

- [1] อ.บุญลือ บุญคง, “ประเภทของหุ่นยนต์อุตสาหกรรม (Industrial Robot type)”, 2018. http://www.9engineer.com/index.php?m=article&a=print&article_id=888 (accessed Sep.20, 2022).
- [2] แผนกวิชาเมคคาทรอนิกส์ วิทยาลัยเทคนิคชลบุรี, “Articulated Robot”, 2020. http://ctc.chontech.ac.th/external_newsblog.php?links=606 (accessed Sep.20, 2022).
- [3] ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสยาม, “ภาพเปรียบเทียบร่างกายของมนุษย์กับแขนกลของหุ่นยนต์อุตสาหกรรม”, 2018. <https://e-research.siam.edu/wp-content/uploads/2020/09/engineering-electrical-engineering-2019-coop-Design-and-Construction-of-a-Robot-Arm-for-Making-Coffee-compressed.pdf> (accessed Sep. 23, 2022).
- [4] บจ.เจริญเมือง แมชชีนเนอรี, “มอเตอร์ไฟฟ้า”, 2020. <https://www.xn--m3cuaf12db-7b3hvb5c6a.net> (accessed Sep. 26, 2022).
- [5] บริษัท แฟ็คโตมาร์ท จำกัด, “หลักการทำงานของ Stepping Motor”, 2022. <https://mall.factomart.com/principle-of-stepping-motor> (accessed Sep. 26, 2022).
- [6] สรญา นงนุช, “ทุกเรื่องที่คุณควรรู้เกี่ยวกับเซอร์โวมอเตอร์และการใช้งาน”, 2020. <https://www.artronshop.co.th/article/92> (accessed Sep. 26, 2022).
- [7] บริษัท อินดัสทรีโปร จำกัด, “เกียร์ทดรอบ GEAR SPEED REDUCER”, 2018. <https://industypro.co.th/gear-speed-reducer/> (accessed Sep. 28, 2022).
- [8] วิทยาลัยเทคนิคสทหีบ, “ชุดเฟืองแพลนเนตตารี (Planetary Gear)”, 2011. http://www.tatc.ac.th/files/09021213134814_11060210100937.pdf (accessed Sep. 28, 2022).
- [9] ดิวเตอร์จุฬา, “ทอร์ก คืออะไร”, 2008. <https://shorturl.asia/3m1GN> (accessed Sep. 30, 2022).
- [10] CyberTiceh, “แนะนำ Arduino รุ่นต่างๆกัน”, 2015. <https://www.cybertice.com/article/4> (accessed Sep. 30, 2022).
- [11] Arduino and ATTINY pinouts, “arduino duemilanove pinout”, 2016. https://arduino.pinout.guide/arduino_duemilanove_pinout.png (accessed Sep. 30, 2022).
- [12] commandronstore, “A4988 Stepper Driver”, 2016. <https://commandronstore.com/products/cm4988.php> (accessed Sep. 30, 2022).
- [13] seekpng, “Industrial Robot Arm - lrb 6700 150 3.20”, 2020. <https://www.seekpng.com/ima/u2w7q8o0q8w7y3r5> (accessed Oct. 18, 2022).

- [14] Synergysoft (Thailand) Co.,Ltd., “Autodesk Inventor”, 2019. <https://mail.luvicha.com/2-uncategorised/8-inventor> (accessed Oct. 18, 2022).
- [15] modena.co.za, “autodesk-inventor-professional-icon-400”, 2020. https://www.moderena.co.za/?attachment_id=31099 (accessed Oct. 18, 2022).
- [16] Michael Rehtin, “80:1 Compound Planetary Gearbox for NEMA 17 Stepper Motor”, 2022. <https://www.thingiverse.com/thing:5205799> (accessed Sep. 26, 2022).
- [17] IndeterminateDesign, “Slim Nema 17 Planetary Drive 4.2:1”, 2020. <https://www.thingiverse.com/thing:4533376> (accessed Sep. 26, 2022).
- [18] wellpcb.com, “Hx711- A HX711 and Load Cell Guide”, 2019. <https://www.wellpcb.com/Hx711.html> (accessed Oct. 26, 2022).
- [19] วิรัช สิริทรัพย์ไพบูลย์, “โปรแกรมเชื่อมต่อกับผู้ใช้ทางกราฟิกในการหาสมการการเคลื่อนที่ของหุ่นยนต์ด้วย โปรแกรม MATLAB,” 2016.
- [20] บริษัท อีสาน3ดี จำกัด, “อัตราทด 1:100 Planetary gear stepper motor NEMA17 40mm motor length (40N.cm)”. <https://www.arduino4pro.com/product/2235/planetary-gear-stepper-motor-nema17> (accessed Jan. 3, 2022).
- [21] ห้างหุ้นส่วนจำกัด โซนเมกเกอร์ (สำนักงานใหญ่), “HANPOSE-Planetary Gearbox Ratio 27:1 Nema17 Stepper Motor (17HS4401S-PG27)”. <https://www.zonemaker.com/product/2717/hanpose-planetary-gearbox-ratio-271-nema17-stepper-motor-17hs4401s-pg27> (accessed Jan. 3, 2022).
- [22] REALITY BYTES, “Forward and Inverse Kinematics: Jacobians and Differential Motion”. <https://realitybytes.blog/2017/06/20/forward-and-inverse-kinematics-jacobians-and-differential-motion> (accessed Jan. 10, 2022).

ภาคผนวก

ภาคผนวก ก

Arduino Code โปรแกรมสำหรับควบคุมตัวหุ่นยนต์อุตสาหกรรม

```

#include <AccelStepper.h>
#include <Servo.h>

#define BAUD          (57600) // How fast is the Arduino talking?
#define MAX_BUF       (64)   // What is the longest message Arduino can store?
#define STEPS_PER_TURN (400)  // depends on your stepper motor. most are 200.
#define NUM_AXIES     (6)
#define ARC_CW        (1)
#define ARC_CCW        (-1)

typedef struct {
    long delta; // number of steps to move
} Axis;

Axis a[NUM_AXIES];

Servo myservo;
int pos = 0;

char serialBuffer[MAX_BUF]; // where we store the message until we get a ';'
intsofar;                   // how much is in the buffer
float rpx,rpy,rpz,rpu,rvp,rpw; // Reference position
float px,py,pz,pu,pv,pw;      // position

//set up the pins for each motor for used <AccelStepper.h> name(driver,steppin,dirpin)
AccelStepper motor1(1, 53, 52);
AccelStepper motor2(1, 51, 50);
AccelStepper motor3(1, 49, 48);
AccelStepper motor4(1, 47, 46);

```

```

AccelStepper motor5(1, 45, 44);
AccelStepper motor6(1, 43, 42);

void motor_setup() {
  //set motor1
  motor1.setMaxSpeed(1100);
  motor1.setAcceleration(1100);

  //set motor2
  motor2.setMaxSpeed(1100);
  motor2.setAcceleration(1100);

  //set motor3
  motor3.setMaxSpeed(400);
  motor3.setAcceleration(400);

  //set motor4
  motor4.setMaxSpeed(110);
  motor4.setAcceleration(110);

  //set motor5
  motor5.setMaxSpeed(120);
  motor5.setAcceleration(120);

  //set motor6
  motor6.setMaxSpeed(60);
  motor6.setAcceleration(60);
}

//Set the logical position
void position(float npz,float npu,float npv,float npw) {
  px=npz;
  py=npv;

```

```

pz=npz;
pu=npu;
pv=npv;
pw=npw;
}

```

//Uses bresenham's line algorithm to move both motors

```

void move(float newx,float newy,float newz,float newu,float newv,float neww) {
    a[0].delta = newx-px;
    a[1].delta = newy-py;
    a[2].delta = newz-pz;
    a[3].delta = newu-pu;
    a[4].delta = newv-pv;
    a[5].delta = neww-pw;

```

```

    int i;

```

```

    long degreejoint1 ; // number of steps to move joint 1
    long degreejoint2 ; // number of steps to move joint 2
    long degreejoint3 ; // number of steps to move joint 3
    long degreejoint4 ; // number of steps to move joint 4
    long degreejoint5 ; // number of steps to move joint 5
    long degreejoint6 ; // number of steps to move joint 6

```

```

// JOINT1

```

```

    degreejoint1 = ((a[0].delta)/1.8)*80 ;
    motor1.move(degreejoint1);

```

```

// JOINT2

```

```

    degreejoint2 = ((a[1].delta)/1.8)*100 ;
    motor2.move(degreejoint2);

```

```

// JOINT3

```

```

degreejoint3 = ((a[2].delta)/1.8)*27 ;
motor3.move(degreejoint3);

// JOINT4
degreejoint4 = ((a[3].delta)/1.8)*4 ;
motor4.move(degreejoint4);

// JOINT5
degreejoint5 = ((a[4].delta)/1.8)*4 ;
motor5.move(degreejoint5);

// JOINT6
degreejoint6 = ((a[5].delta)/1.8) ;
motor6.move(degreejoint6);

position(newx,newy,newz,newu,newv,neww);
}

void gripper_catch(){
  for (pos = 0; pos <= 130; pos += 1) {
    myservo.write(pos);
    delay(15);
  }
  return;
}

void gripper_place(){
  for (pos = 130; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(15);
  }
  return;
}

```

//Look for character /code/ in the buffer and read the float that immediately follows it.

```
float parseNumber(char code,float val) {
    char *ptr=serialBuffer;
    while((long)ptr > 1 && (*ptr) && (long)ptr < (long)serialBuffer+sofar) {
        if(*ptr==code) {
            return atof(ptr+1);
        }
        ptr=strchr(ptr,'')+1;
    }
    return val;
}
```

//write a string followed by a float to the serial line.

```
void output(char *code,float val) {
    Serial.print(code);
    Serial.println(val);
}
```

//display helpful information

```
void help() {
    Serial.println(F("Commands:"));
    Serial.println(F("G00 [X/Y/Z/U/V/W(steps)]; - move"));
    Serial.println(F("G01; - set home position"));
    Serial.println(F("G02; - gripper catch"));
    Serial.println(F("G03; - gripper place"));
}
```

//Read the input buffer and find any recognized commands.

```
void processCommand() {
    int cmd = parseNumber('G',-1);
    switch(cmd) {
        case 0: { // move
```

```

    move( parseNumber('X',px),
          parseNumber('Y',py),
          parseNumber('Z',pz),
          parseNumber('U',pu),
          parseNumber('V',pv),
          parseNumber('W',pw) );
    break;
}

```

```

case 1: { // set home position
    position( px=0,
              py=0,
              pz=0,
              pu=0,
              pv=0,
              pw=0 );
    break;break;
}

```

```

case 2: { //gripper catch
    gripper_catch();
    break;
}

```

```

case 3: { //gripper place
    gripper_place();
    break;
}

```

```

default: break;
}
}

```

//prepares the input buffer to receive a new message and tells the serial connected device it is ready for more.

```
void ready() {
   sofar=0;
    Serial.print(F(">"));
}
```

//First thing this machine does on startup. Runs only once.

```
void setup() {
    Serial.begin(BAUD);
    motor_setup();
    help();
    position(0,0,0,0,0,0);
    ready();
    myservo.attach(9);
}
```

//After setup() this machine will repeat loop() forever.

```
void loop() {

    // listen for serial commands
    while(Serial.available() > 0) {
        char c=Serial.read();
        Serial.print(c);
        if(sofar<MAX_BUF-1) serialBuffer[sofar++]=c; // store it
        if(c=='\n') {
            serialBuffer[sofar]=0;
            Serial.print(F("\r\n"));
            processCommand();
            ready();
        }
    }
}
```



```
//Move the motor one step  
motor1.run();  
motor2.run();  
motor3.run();  
motor4.run();  
motor5.run();  
motor6.run();  
}
```

ภาคผนวก ข

Code โปรแกรมคำนวณหาตำแหน่งจุดปลายและจำลองการทำงานของแขนหุ่นยนต์ อุตสาหกรรม

```
function [a,x,y,z] = htm(theta1,theta2,theta3,theta4,theta5,theta6)

q1 = theta1 * pi/180;
q2 = theta2 * pi/180;
q3 = theta3 * pi/180;
q4 = theta4 * pi/180;
q5 = theta5 * pi/180;
q6 = theta6 * pi/180;

function [A] = DH(a,alpha,d,theta)

    A = [cos(theta) -sin(theta)*round(cos(alpha)) sin(theta)*round(sin(alpha)) a*cos(theta)
          sin(theta) cos(theta)*round(cos(alpha)) -cos(theta)*round(sin(alpha)) round(a*sin(theta))
          0 round(sin(alpha)) round(cos(alpha)) d
          0 0 0 1];

end

% l0 l1 l2 l3 l4 l5 l6
% pos 0,0,0,0,0,0 x,y,z = 21.5,-1.5,30
% pos 30,45,0,0,0,0 x,y,z = 15,14.5,11

l0 = 10;
l1 = 1;
l2 = 9;
l3 = 5.5;
l4 = 8.5;
l5 = 7;
l6 = 5.5;

% DH
a1 = 1.7; alpha1 = - pi/2; d1 = 6 + l0;
a2 = l2; alpha2 = 0 ; d2 = - 2 ;
a3 = l3 ; alpha3 = pi/2 ; d3 = 0 ;
a4 = 0 ; alpha4 = -pi/2 ; d4 = -14 ;
a5 = 0 ; alpha5 = pi/2 ; d5 = 0 ;
a6 = 0 ; alpha6 = pi ; d6 = 15 + l6 ;
```

```

for i = 0:2:s
%   for q5 = pi/2:0.1:th5
%       for q4 = pi/2:0.1:th4
%           for q3 = pi/2:0.1:th3
%               for q2 = pi/2:0.1:th2
%                   for q1 = pi/2:0.5:th1
q1 = ra1*i;
q2 = ra2*i;
q3 = ra3*i;
q4 = ra4*i;
q5 = ra5*i;
q6 = ra6*i;

h01 = DH(a1,alpha1,d1,-q1);
h12 = DH(a2,alpha2,d2,q2 - pi/2 );
h23 = DH(a3,alpha3,d3,q3 );
h34 = DH(a4,alpha4,d4,-q4);
h45 = DH(a5,alpha5,d5,q5 - pi );
h56 = DH(a6,alpha6,d6,q6);
h06 = h01*h12*h23*h34*h45*h56 ;

%plot3d
h02 = h01*h12;
h03 = h01*h12*h23;
h04 = h01*h12*h23*h34;
h05 = h01*h12*h23*h34*h45;

decimal = 3;

x01 = round(10^decimal*h01(1,4))/10^decimal;
y01 = round(10^decimal*h01(2,4))/10^decimal;
z01 = round(10^decimal*h01(3,4))/10^decimal;

```

```

x02 = round(10^decimal*h02(1,4))/10^decimal;
y02 = round(10^decimal*h02(2,4))/10^decimal;
z02 = round(10^decimal*h02(3,4))/10^decimal;

x03 = round(10^decimal*h03(1,4))/10^decimal;
y03 = round(10^decimal*h03(2,4))/10^decimal;
z03 = round(10^decimal*h03(3,4))/10^decimal;

x04 = round(10^decimal*h04(1,4))/10^decimal;
y04 = round(10^decimal*h04(2,4))/10^decimal;
z04 = round(10^decimal*h04(3,4))/10^decimal;

x05 = round(10^decimal*h05(1,4))/10^decimal;
y05 = round(10^decimal*h05(2,4))/10^decimal;
z05 = round(10^decimal*h05(3,4))/10^decimal;

x = round(10^decimal*h06(1,4))/10^decimal;
y = round(10^decimal*h06(2,4))/10^decimal;
z = round(10^decimal*h06(3,4))/10^decimal;

plot3([0, 0], [0, 0], [0, 10], ...
      '-o','LineWidth',2,'MarkerSize',4,'MarkerFaceColor',[0.5,0.5,0.5]);
hold on;
plot3([0, x01], [0, y01], [10, z01], ...
      '-o','LineWidth',2,'MarkerSize',4,'MarkerFaceColor',[0.5,0.5,0.5]);
hold on;

plot3([x01, x02], [y01, y02], [z01, z02], ...
      '-o','LineWidth',2,'MarkerSize',4,'MarkerFaceColor',[0.5,0.5,0.5]);
hold on;
plot3([x02, x03], [y02, y03], [z02, z03], ...
      '-o','LineWidth',2,'MarkerSize',4,'MarkerFaceColor',[0.5,0.5,0.5]);
hold on;
plot3([x03, x04], [y03, y04], [z03, z04], ...
      '-o','LineWidth',2,'MarkerSize',4,'MarkerFaceColor',[0.5,0.5,0.5]);
hold on;

```