



การพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง



สุภาพ มายาง

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
ปีการศึกษา 2564
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

การพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง



วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร
เพื่อเป็นส่วนหนึ่งของการศึกษา หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
ปีการศึกษา 2564
ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

วิทยานิพนธ์ เรื่อง "การพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง"

ของ สุภาพ มายาง

ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล

คณะกรรมการสอบวิทยานิพนธ์

ประธานกรรมการสอบวิทยานิพนธ์

.....
(ผู้ช่วยศาสตราจารย์ ดร.กมล วสะภิญโญกุล)

ประธานที่ปรึกษาวิทยานิพนธ์

.....
(ผู้ช่วยศาสตราจารย์ ดร.สุเมธ เหมะวัฒน์ชัย)

กรรมการผู้ทรงคุณวุฒิภายใน

.....
(ผู้ช่วยศาสตราจารย์ ดร.อนันต์ชัย อยู่แก้ว)

อนุมัติ

.....
(รองศาสตราจารย์ ดร.กรองกาญจน์ ชูทิพย์)

คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง	การพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง
ผู้วิจัย	สุภาพ มายาง
ประธานที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สุเมธ เหมะวัฒนะชัย
ประเภทสารนิพนธ์	วิทยานิพนธ์ วศ.ม. สาขาวิชาวิศวกรรมเครื่องกล, มหาวิทยาลัยนเรศวร , 2564
คำสำคัญ	หุ่นยนต์เคลื่อนที่, แขนหุ่นยนต์, จลนศาสตร์แบบไปข้างหน้าและแบบ ผกผัน

บทคัดย่อ

งานวิจัยนี้เป็นการออกแบบและพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง เพื่อช่วยในการหยิบจับสิ่งของขนาดเล็ก เช่น กระจุกยา และรีโมททีวี ซึ่งสามารถควบคุมหุ่นยนต์ด้วยแอปพลิเคชันบนสมาร์ตโฟนผ่านสัญญาณผ่านสัญญาณบลูทูธ และ Wi-Fi พร้อมกับแสดงภาพจากกล้องที่ติดกับหุ่นยนต์บนหน้าจอสมาร์ตโฟนได้ตามเวลาจริง ในงานวิจัยนี้แบ่งการพัฒนาเป็นสองส่วนคือ หุ่นยนต์เคลื่อนที่กับแขนหุ่นยนต์ โดยส่วนของหุ่นยนต์เคลื่อนที่ที่มีการติดตั้งล้อเป็นแบบประเภท Differential Drive Robot คือ มีการขับเคลื่อนด้วยสองล้อหน้า และมีล้ออิสระด้านหลัง โดยมีระบบควบคุมด้วยไมโครคอนโทรลเลอร์ ใช้ สัญญาณ Pulse Width Modulation ควบคุมความเร็วและทิศทางของมอเตอร์กระแสตรง สั่งการผ่านแอปพลิเคชันบนสมาร์ตโฟนที่ได้พัฒนาขึ้น การทดลองสำหรับหุ่นยนต์เคลื่อนที่แบ่งเป็นสามการทดลองคือ การทดลองแรกหุ่นยนต์เคลื่อนที่เป็นเส้นตรง แบ่งการทดสอบโดยปราศจากน้ำหนักและทดสอบเพิ่มภาระน้ำหนักให้ตัวหุ่นยนต์เป็น 10 kg และ 20 kg จากผลการทดสอบพบว่าหุ่นยนต์สามารถเคลื่อนที่ได้โดยสามารถรับภาระน้ำหนักเพิ่มขึ้นจากภายนอก โดยที่ความเร็วของหุ่นยนต์ยังคงที่ การทดลองที่สองให้หุ่นยนต์การเลี้ยวซ้ายและเลี้ยวขวา ผลการทดลองค่าเส้นรัศมีการเลี้ยวโค้งของหุ่นยนต์ที่ได้จากการวัดเปรียบเทียบกับค่าจากการคำนวณโมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์ พบว่ามีค่าความคลาดเคลื่อนที่ยอมรับได้ ข้อผิดพลาดอาจเกิดจากการวัดโดยสายตามนุษย์และเครื่องมือที่ใช้ในการวัด โดยนำค่า pwm ที่เหมาะสมไปกำหนดการเลี้ยวโค้งซ้ายและเลี้ยวโค้งขวาในโปรแกรมของหุ่นยนต์ การทดลองที่สามให้ผู้ใช้ควบคุมหุ่นยนต์ให้เคลื่อนที่ไป-กลับ อ้อมระหว่างกรวยซึ่งห่างกัน 5 m ผลการทดลองพบว่าผู้ใช้งานคนแรกใช้เวลาประมาณ 86.2 วินาที ความเร็วเฉลี่ยประมาณ 11.61 cm/s ต่อมาคนที่สองใช้เวลาประมาณ 100 วินาที ความเร็วเฉลี่ยประมาณ 10.01 cm/s และคนสุดท้ายใช้เวลาประมาณ 90 วินาที ความเร็วเฉลี่ยประมาณ 11.13 cm/s ซึ่งส่วนเบี่ยงเบนมาตรฐานของการทดสอบผู้ใช้งานทั้ง

สามคนมีค่าออกมาน้อย แสดงว่ากลุ่มข้อมูลมีการกระจายที่น้อยข้อมูลส่วนใหญ่อยู่ใกล้ค่าเฉลี่ยมาก ต่อมาเป็นการพัฒนาส่วนของแขนหุ่นยนต์ โดยมีการประยุกต์นำเอาทฤษฎีจลนศาสตร์แบบไปข้างหน้า และแบบผกผัน มาวิเคราะห์หาพื้นที่ทำงานที่เหมาะสมของแขนหุ่นยนต์ และมีการพัฒนาซอฟต์แวร์ เพื่อควบคุมเซอร์โวมอเตอร์หลายตัวพร้อมกัน เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่แบบเส้นตรงตามแนว หน้า-หลัง (+x, -x) และแนวขึ้น-ลง (+y, -y) เหมาะสำหรับการควบคุมผ่านหน้าจอสมาาร์ทโฟน ในการทดลองเพื่อประเมินสมรรถนะของระบบแขนหุ่นยนต์ที่ได้พัฒนาขึ้น แบ่งเป็นสองการทดลองหลัก คือ การทดลองแรกเป็นการทดสอบความสามารถในการควบคุมมือจับของแขนหุ่นยนต์ ซึ่งผลการทดลอง พบว่ามือจับสามารถเคลื่อนที่ได้ตามขอบเขตพื้นที่ทำงานตามกำหนด และการทดลองที่สองเป็นการ ทดสอบด้านระยะเวลาที่ใช้ในการควบคุมแขนหุ่นยนต์เพื่อหยิบจับสิ่งของ จากผลการทดลองพบว่าการควบคุมแบบเคลื่อนที่เชิงเส้นแบบหลายแกนที่พัฒนาขึ้นมาในงานวิจัยนี้ใช้เวลาน้อยกว่าประมาณ 86% เมื่อเทียบกับการควบคุมแบบทีละข้อต่ออิสระ



Title	DEVELOPMENT OF SERVICE ROBOT FOR BEDRIDDEN PATIENT
Author	SUPAP MAYANG
Advisor	Assistant Professor Sumet Heamawatanachai, Ph.D.
Academic Paper	M.Eng. Thesis in Mechanical Engineering, Naresuan University, 2021
Keywords	Mobile Robot, Robot Arm, Forward and Inverse Kinematics

ABSTRACT

This paper is a design and development of Service Robot for Bedridden Patient that can pick up small objects, such as pill boxes and TV remotes, etc. For patients or people who have difficulty moving. The robot is controlled by an application on a smartphone via Bluetooth and Wi-Fi and can show images from the camera attached to the robot on the phone's screen in real-time. In this research the development is divided into two parts the mobile robot and the robot arm. The part of the mobile robot is equipped with differential drive robot wheels, which is driven by two front wheels. and has independent wheels on the back. It has a microcontroller control system using Pulse Width Modulation signals to control the speed and direction of the DC motor. Command through an application on a smartphone that has been developed. There are three experiments for mobile robots. In the first experiment the robot moves in a linear. Divide the weightless test and increase the load on the robot to 10 kg and 20 kg. The test results showed that the robot was able to move with increased external load. while the speed of the robot remains constant. The second experiment gave the robot a left and right turn. The results of the experimental values of the turning radius of the robot obtained from the measurement and comparing the values from the calculation of the robot's curving motion control system model. Tolerances were found to be acceptable. Errors may be caused by human eye measurements and measuring instruments. The appropriate pwm is used to determine the left and right turns in the robot program.

The third experiment had users control the robot to move back and forth. between the cones, which are 5 m apart. The results showed that the first user took about 86.2 seconds, the average speed was about 11.61 cm/s, the second took about 100 seconds, the average speed was about 10.01 cm/s, and the last person took about 90 seconds, the average speed was about 90 seconds, 11.13 cm/s The standard deviation of the three user tests was less. Shows that the data group has a less distribution, most of the data is very close to the mean. Part of the development of the robot arm. The theory of forward and inverse kinematics was applied to analyze the appropriate working area of the robot arm. In order to make the system suitable for control via a smartphone's screen, the software in this research was developed to be able to control multiple servo motors simultaneously to move the gripper of the robot arm linearly, forwards-backwards (+x, -x) and in up-down directions (+y, -y). There were two main experiments to evaluate the performance of the system. The first was to test the movement ability of the arm, and the results showed that it was able to move within the specified working area. The second examined the amount of time spent manipulating the robot arm to pick up objects. From the result of this experiment, it was found that the multi-axis linear motion control technique developed in this research, was approximately 86% less time consuming than the independent single-joint control technique.

ประกาศคุณูปการ

วิทยานิพนธ์เรื่องการพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียงนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี โดยได้รับทุนสนับสนุนจากสำนักงานการวิจัยแห่งชาติ ภายใต้โครงการแผนพัฒนาศักยภาพบัณฑิตวิจัยรุ่นใหม่ ซึ่งได้สนับสนุนค่าใช้จ่ายในการทำวิจัยรวมถึงค่าวัสดุอุปกรณ์ต่าง ๆ เพื่อให้ได้ผลงานสิ่งประดิษฐ์ขึ้นมาในครั้งนี้ ทางผู้วิจัยต้อง ขอขอบพระคุณเป็นอย่างสูง

ผู้วิจัยขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร. สุเมธ เหมะวิฒนะชัย ประธานที่ปรึกษาวิทยานิพนธ์ ที่ได้สละเวลาอันมีค่ามาให้ปรึกษาพร้อมทั้งคำแนะนำต่าง ๆ มาโดยตลอดระยะเวลาของการทำวิทยานิพนธ์ฉบับนี้ จึงขอขอบพระคุณเป็นอย่างสูง

ในโอกาสนี้ต้องขอขอบคุณบิดา มารดาพร้อมทั้งครอบครัว รวมถึงเพื่อน ๆ ที่สนับสนุนในด้านต่าง ๆ รวมถึงการให้กำลังใจในช่วงเวลาที่ผ่านมา จนการวิจัยได้เสร็จสิ้น ขอขอบคุณภาควิชาวิศวกรรมเครื่องกล ที่เอื้อเฟื้อสถานที่ในการทำ วิจัย และสุดท้ายต้องขอขอบพระคุณทุกกำลังใจที่มอบให้ผ่านพ้นอุปสรรคมาได้ งานวิจัยฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ขอขอบพระคุณจากใจ

สุภาพ มายาง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	จ
ประกาศคุณูปการ.....	ช
สารบัญ.....	ฅ
สารบัญตาราง.....	ฉ
สารบัญภาพ.....	ค
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 จุดมุ่งหมายของการศึกษา.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตของงานวิจัย.....	3
1.5 สมมติฐานของการวิจัย.....	3
1.6 วิธีดำเนินการวิจัย.....	3
1.7 แผนการดำเนินงาน.....	5
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 หลักการและทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 ความหมายและความสำคัญของของหุ่นยนต์.....	6
2.1.2 ชนิดของหุ่นยนต์.....	7
2.1.3 ประโยชน์และความสามารถของหุ่นยนต์.....	8

2.1.4 ประเภทของแขนหุ่นยนต์	10
2.1.5 ส่วนประกอบของหุ่นยนต์.....	15
2.1.6 วัตถุแข็งเกร็ง (Rigid body)	17
2.1.7 องศาอิสระ (degrees of freedom)	17
2.1.8 ชนิดของการเคลื่อนที่ (Type of Motion).....	18
2.1.9 ระบบจั่นต่อโยง (Linkage).....	20
2.1.10 โซ่กลนศาสตร์ (Kinematic chain)	21
2.1.11 Pulse width modulation (PWM)	22
2.1.12 วัสดุอุปกรณ์ และซอฟต์แวร์ที่ใช้ในงานวิจัย.....	23
2.2งานวิจัยที่เกี่ยวข้อง	33
2.2.1 งานวิจัยเกี่ยวกับหุ่นยนต์เคลื่อนที่.....	33
2.2.2 งานวิจัยเกี่ยวกับแขนหุ่นยนต์.....	37
2.3 หุ่นยนต์ที่ใช้งานในโรงพยาบาล	40
บทที่ 3 การออกแบบและพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง.....	45
3.1 การออกแบบโครงสร้างหุ่นยนต์และสร้างหุ่นยนต์เคลื่อนที่.....	47
3.2 การออกแบบและสร้างวงจรควบคุมหุ่นยนต์เคลื่อนที่	49
3.3 โมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์	51
3.4 การออกแบบโครงสร้างแขนหุ่นยนต์และสร้างแขนหุ่นยนต์.....	53
3.5 การออกแบบและสร้างวงจรควบคุมแขนหุ่นยนต์	57
3.6 โมเดลระบบควบคุมแขนหุ่นยนต์สำหรับการเคลื่อนที่เชิงเส้นแบบหลายแกน.....	58
3.6.1 โมเดลระบบควบคุมแขนหุ่นยนต์สำหรับการเคลื่อนที่เชิงเส้นแบบหลายแกน	59

3.6.2 การหาพื้นที่ทำงานของแขนหุ่นยนต์.....	63
3.6.3 การกำหนดพื้นที่ทำงานของแขนหุ่นยนต์.....	65
3.7 การออกแบบและสร้างแอปพลิเคชันบนสมาร์ตโฟนเพื่อใช้ในการติดต่อและควบคุมหุ่นยนต์	65
3.8 กระบวนการทำงานของหุ่นยนต์ โดยการควบคุมไร้สายผ่านแอปพลิเคชันบนสมาร์ตโฟน.....	67
3.8.1. วิธีการใช้งานแอปพลิเคชันสำหรับการควบคุมหุ่นยนต์เคลื่อนที่	68
3.8.2 วิธีการใช้งานแอปพลิเคชันสำหรับการควบคุมแขนหุ่นยนต์	71
บทที่ 4 ผลการวิจัย และการอภิปรายผล.....	77
4.1 การทดลองการเคลื่อนที่ของหุ่นยนต์.....	77
4.2 การทดลองส่วนของแขนหุ่นยนต์.....	85
4.2.1 การทดลองการเคลื่อนที่ของแขนหุ่นยนต์ในระนาบสองมิติ.....	85
4.2.2 การทดลองการควบคุมแขนหุ่นยนต์ผ่านโทรศัพท์มือถือ	89
บทที่ 5 สรุปผลการทดลอง และข้อเสนอแนะ	92
5.1 สรุปผลการทดลอง	92
5.2 ปัญหาและอุปสรรค.....	93
5.3 ข้อเสนอแนะและแนวทางในการแก้ไข	94
บรรณานุกรม	95
บรรณานุกรม	96
ภาคผนวก ก ผลการทดลองการเคลื่อนที่เป็นเส้นตรงของหุ่นยนต์.....	100
ภาคผนวก ข โปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์.....	103
ภาคผนวก ค โปรแกรมการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์	120



สารบัญตาราง

หน้า

ตาราง 1 แสดงแผนการดำเนินงาน	5
ตาราง 2 แสดงการเปรียบเทียบการใช้งานของหุ่นยนต์ที่ใช้ในโรงพยาบาล	44
ตาราง 3 การวิเคราะห์แขนหุ่นยนต์ด้วย D-H parameter.....	60
ตาราง 4 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง ภายในห้องทดสอบ ไม่มีภาระน้ำหนักเพิ่มเติม.....	78
ตาราง 5 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง ภายในห้องทดสอบ เพิ่มน้ำหนักที่ 10 kg.....	79
ตาราง 6 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง ภายในห้องทดสอบ เพิ่มน้ำหนักที่ 20 kg.....	79
ตาราง 7 การทดสอบการเลี้ยวโค้งซ้ายของหุ่นยนต์.....	82
ตาราง 8 การทดสอบการเลี้ยวโค้งขวาของหุ่นยนต์.....	82
ตาราง 9 การทดสอบควบคุมหุ่นยนต์ไป-กลับอ้อมกรวยภายในห้องทดสอบ	84
ตาราง 10 แรงบิดแต่ละช่วงต่อของแขนหุ่นยนต์ และแรงบิดสูงสุดของเซอร์โวมอเตอร์.....	89
ตาราง 11 ผลการทดลองการควบคุมแขนหุ่นยนต์.....	90
ตาราง 12 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินไปข้างหน้า ระยะทาง 5 m ไม่มีภาระน้ำหนักเพิ่มเติม.....	100
ตาราง 13 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินถอยหลัง ระยะทาง 5 m ไม่มีภาระน้ำหนักเพิ่มเติม.....	100
ตาราง 14 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินหน้า ระยะทาง 5 m เพิ่มน้ำหนักที่ 10 kg.....	101

ตาราง 15 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินถอยหลัง ระยะทาง 5 m เพิ่มน้ำหนักที่ 10 kg.....	101
ตาราง 16 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินหน้า ระยะทาง 5 m เพิ่มน้ำหนักที่ 20 kg.....	101
ตาราง 17 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินถอยหลัง ระยะทาง 5 m เพิ่มน้ำหนักที่ 20 kg.....	102



สารบัญภาพ

	หน้า
ภาพ 1 สັดส์คอายุประชากรไทย ในปี 2563	1
ภาพ 2 หุ่นยนต์ที่ติดตั้งอยู่กับที่ ไม่สามารถเคลื่อนที่ได้	7
ภาพ 3 หุ่นยนต์ที่สามารถเคลื่อนไหวและเคลื่อนที่ได้.....	8
ภาพ 4 Cartesian robot arm.....	10
ภาพ 5 Cylindrical robot arm.....	10
ภาพ 6 Spherical robot arm.....	11
ภาพ 7 Scara robot arm.....	11
ภาพ 8 Articulate robot arm.....	12
ภาพ 9 Anthropomorphic arm.....	12
ภาพ 10 Delta Robot	13
ภาพ 11 Rostock delta robot	14
ภาพ 12 Stewart platform	14
ภาพ 13 แผนภาพส่วนประกอบหลักของหุ่นยนต์.....	15
ภาพ 14 ส่วนประกอบทางกลของแขนหุ่นยนต์.....	16
ภาพ 15 ชนิดของการเคลื่อนที่.....	17
ภาพ 16 การเคลื่อนที่แบบหมุนอย่างเดียว	19
ภาพ 17 การเคลื่อนที่แบบเลื่อนตำแหน่งเดียว	19
ภาพ 18 การเคลื่อนที่แบบซับซ้อน (Complex Motion).....	20
ภาพ 19 ชนิดของจิ้นต่อ โยงแบ่งตามจำนวนของ Node	20

ภาพ 20 ชนิดของชิ้นต่อโยงในชุดกลไก.....	21
ภาพ 21 การกำหนดความกว้างของสัญญาณพัลส์.....	22
ภาพ 22 แสดงส่วนประกอบและการทำงานของมอเตอร์ไฟตรง.....	24
ภาพ 23 ส่วนประกอบของเซอร์โวมอเตอร์.....	24
ภาพ 24 บล็อกไดอะแกรมของเซอร์โวมอเตอร์.....	25
ภาพ 25 ความกว้างสัญญาณพัลส์ เพื่อกำหนดมุมองศาเซอร์โวมอเตอร์	26
ภาพ 26 กำหนดมุมองศาเซอร์โวมอเตอร์ที่ 45 องศา.....	26
ภาพ 27 บอร์ด Arduino Uno R3	28
ภาพ 28 บอร์ด Arduino Mega 2560 R3	29
ภาพ 29 บอร์ด Raspberry pi 3	29
ภาพ 30 ส่วนประกอบของ Kodular.....	30
ภาพ 31 หน้าต่างส่วน Blocks สำหรับเขียนโปรแกรม	31
ภาพ 32 หุ่นยนต์เคลื่อนที่โดยอาศัยข้อมูลจากการตรวจวัดเพื่อหลบหลีกสิ่งกีดขวางแบบอัตโนมัติ.....	33
ภาพ 33 Behavior-based Mobile Robot and Advanced Software Collaboration.....	34
ภาพ 34 การพัฒนาระบบการเคลื่อนที่ของหุ่นยนต์ด้วยพีซีลจิก	34
ภาพ 35 หุ่นยนต์เคลื่อนที่สำหรับการติดตาม การเฝ้าระวังและตรวจจับการล้มของผู้สูงอายุ.....	35
ภาพ 36 หุ่นยนต์เคลื่อนที่รอบทิศทางบนพื้นฐานของล้อแมคคานัม.....	36
ภาพ 37 การประยุกต์การควบคุมแขนหุ่นยนต์ 2 แขนแบบระนาบกรณีขับเรืออย่างสมบูรณ์ด้วยวิธีการ ป้อนกลับเพียงตรงให้เป็นเชิงเส้น	37
ภาพ 38 Design and Construction of Small Five-axis Robot Arm	38

ภาพ 39 การพัฒนาชุดแกนกล 5 แกน เพื่อการศึกษาเชิงพาณิชย์	38
ภาพ 40 แกนหุ่นยนต์จัดเรียงกระถางกระดาษของต้นกล้า.....	39
ภาพ 41 แกนหุ่นยนต์เพื่อป้อนอาหารคนพิการ	40
ภาพ 42 หุ่นยนต์ Healthy Bot	41
ภาพ 43 หุ่นยนต์ HAPYBot.....	42
ภาพ 44 หุ่นยนต์ Pinto	43
ภาพ 45 หุ่นยนต์ Moxi.....	43
ภาพ 46 แผนผังการทำงานของหุ่นยนต์บริการสำหรับผู้พิการติดเตียง	46
ภาพ 47 การออกแบบโครงสร้างหุ่นยนต์เคลื่อนที่.....	47
ภาพ 48 การออกแบบตำแหน่งของล้อหุ่นยนต์เคลื่อนที่	48
ภาพ 49 การติดตั้งมอเตอร์เข้ากับล้อ ผ่านแกนเพลาประกอบติดกับ bearing housing และ Coupling.....	49
ภาพ 50 โครงสร้างหุ่นยนต์บริการสำหรับผู้พิการติดเตียงต้นแบบ	49
ภาพ 51 วงจรของหุ่นยนต์เคลื่อนที่.....	50
ภาพ 52 การออกแบบและติดตั้งเซนเซอร์เอน โค้ดเดอร์.....	51
ภาพ 53 โมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้ง	51
ภาพ 54 เซอร์โวมอเตอร์รุ่น TD8120MG.....	53
ภาพ 55 ส่วนฐานของแกนหุ่นยนต์.....	54
ภาพ 56 ส่วนหัวไหล่ของแกนหุ่นยนต์.....	54
ภาพ 57 ส่วนข้อศอกของแกนหุ่นยนต์.....	55
ภาพ 58 ส่วนมือจับของแกนหุ่นยนต์	55
ภาพ 59 ส่วนประกอบมอเตอร์ของแกน	56

ภาพ 60 แขนหุ่นยนต์.....	56
ภาพ 61 วงจรของแขนหุ่นยนต์.....	57
ภาพ 62 ส่วนประกอบระบบควบคุมการเคลื่อนที่ของแขนหุ่นยนต์.....	58
ภาพ 63 แขนหุ่นยนต์ 3 ก้านต่อ ในระนาบ 2 มิติ.....	59
ภาพ 64 พื้นที่ทำงานของแขนหุ่นยนต์	63
ภาพ 65 ตั้งค่าทิศทางของเซอร์โวมอเตอร์	64
ภาพ 66 แสดงพื้นที่ทำงานแขนหุ่นยนต์จากการกำหนดมุม	64
ภาพ 67 กำหนดขอบเขตพื้นที่การทำงานของแขนหุ่นยนต์.....	65
ภาพ 68 หน้าแรกของแอปพลิเคชัน เพื่อเลือกส่วนควบคุม	66
ภาพ 69 หน้าจอส่วนควบคุมแขนหุ่นยนต์ของแอปพลิเคชัน	66
ภาพ 70 หน้าจอส่วนควบคุมหุ่นยนต์เคลื่อนที่ของแอปพลิเคชัน	67
ภาพ 71 การควบคุมหุ่นยนต์ผ่านแอปพลิเคชันบนสมาร์ตโฟน	68
ภาพ 72 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มเดินหน้า.....	68
ภาพ 73 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มถอยหลัง.....	69
ภาพ 74 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนขวา.....	69
ภาพ 75 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนซ้าย.....	69
ภาพ 76 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนขวาพร้อมกัน	70
ภาพ 77 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนซ้ายพร้อมกัน	70
ภาพ 78 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มถอยหลังกับปุ่มหมุนขวาพร้อมกัน	70
.....	70
ภาพ 79 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มถอยหลังกับปุ่มหมุนซ้ายพร้อมกัน	71
.....	71

ภาพ 80 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมแบบ MALM Control ใน แนวขึ้น-ลง	71
ภาพ 81 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมแบบ MALM Control ใน แนวหน้า-หลัง.....	72
ภาพ 82 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมส่วนฐานของแกน หุ่นยนต์.....	72
ภาพ 83 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมส่วนการหมุนข้อมือของ แกนหุ่นยนต์.....	73
ภาพ 84 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมส่วนหัวไหล่ของแกน หุ่นยนต์.....	73
ภาพ 85 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมส่วนข้อศอกของแกน หุ่นยนต์.....	74
ภาพ 86 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมส่วนข้อมือของแกน หุ่นยนต์.....	74
ภาพ 87 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยการควบคุมส่วนมือจับของแกน หุ่นยนต์.....	74
ภาพ 88 ลักษณะการเคลื่อนที่ของแกนหุ่นยนต์ โดยปุมฟังก์ชันวางของอัตโนมัติ.....	75
ภาพ 89 ปุมฟังก์ชันสำหรับช่วยในการตั้งค่าตำแหน่งเริ่มต้นให้แกนหุ่นยนต์พร้อมใช้งาน	75
ภาพ 90 การทดลองหุ่นยนต์เคลื่อนที่เป็นเส้นตรง	78
ภาพ 91 ความสัมพันธ์ค่า PWM กับความเร็วของหุ่นยนต์เคลื่อนที่ข้างหน้า	80
ภาพ 92 ความสัมพันธ์ค่า PWM กับความเร็วของหุ่นยนต์เคลื่อนที่ข้างหลัง	81
ภาพ 93 การทดลองการเลี้ยวโค้งของหุ่นยนต์	81

ภาพ 94 การทดสอบการเคลื่อนที่ของหุ่นยนต์ไป-กลับ 84

ภาพ 95 โปรแกรมประมวลผลภาพ ตรวจจับจุดสีแดง เคลื่อนที่ตามปลายแขนหุ่นยนต์. 86

ภาพ 96 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวราบ 87

ภาพ 97 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวตั้ง 88

ภาพ 98 การเคลื่อนที่ตามขอบเขตการทำงานของแขนหุ่นยนต์..... 88

ภาพ 99 เวกเตอร์แรงและโมเมนต์ที่กระทำต่อแต่ละช่วงข้อต่อ 89

ภาพ 100 การทดลองให้ผู้ผู้ใช้ ควบคุมบังคับแขนหุ่นยนต์ผ่าน โทรศัพท์มือถือ 90



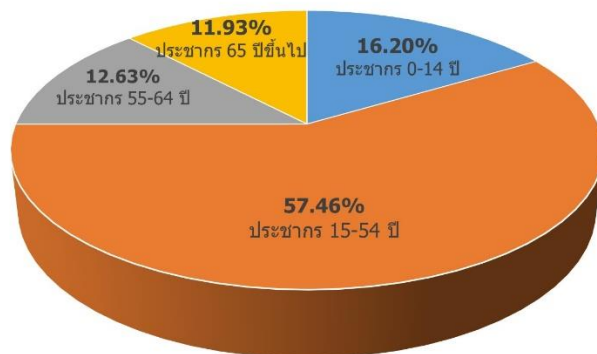
บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ปัจจุบันโลกกำลังก้าวเข้าสู่สังคมผู้สูงอายุ จากรายงานของสำนักงานคณะกรรมการส่งเสริมการลงทุน (บีโอไอ) ระบุว่าในปี พ.ศ. 2563 ประเทศไทยมีประชากรผู้สูงอายุที่มีอายุ 65 ปีขึ้นไป ประมาณ 8 ล้านคน คิดเป็นร้อยละ 11.93 ของประชากรทั้งหมดในประเทศ (Thai population statistics, 2020) ซึ่งกลุ่มคนประเภทดังกล่าวมักมีปัญหาทางด้านความช่วยเหลือตัวเอง เช่น การเดิน การขับตัว รวมถึงการหยิบจับสิ่งของ การเพิ่มสูงขึ้นของผู้สูงอายุ และผู้ป่วยติดเตียงนั้น ส่งผลทำให้ขาดบุคลากรทางการแพทย์ และมีจำนวนบุคลากรไม่เพียงพอต่อความต้องการ ทำให้เกิดการดูแลที่ไม่ทั่วถึง ส่งผลกระทบต่อสถานะความเป็นอยู่ที่ดีของกลุ่มบุคคลทั้งสองประเภท อีกทั้งยังเป็นสาเหตุทำให้ ค่าดูแลรักษาทางการแพทย์ในระดับประเทศเพิ่มขึ้น เพื่อเป็นการทดแทนบุคลากร และสนับสนุนการดำเนินงานทางการแพทย์ ในการส่งเสริมคุณภาพชีวิตที่ดีขึ้นของผู้ป่วย ผู้วิจัยจึงมีแนวคิดในการนำเทคโนโลยีด้านหุ่นยนต์มาใช้ในการช่วยเหลือผู้ป่วย เพื่อที่จะรองรับการเข้าสู่สังคมผู้สูงอายุ และผู้ป่วยติดเตียงได้อย่างสมบูรณ์ แต่ในปัจจุบันหุ่นยนต์ในท้องตลาดยังมีราคาที่สูง จึงทำให้คนส่วนใหญ่ไม่สามารถจะจัดหามาใช้งานได้ สัตตอายุประชากรไทย ในปี 2563 แสดงดังภาพที่ 1

สัตตอายุประชากรไทย ในปี 2563



- ประชากร 0-14 ปี 10.7 ล้านคน
- ประชากร 15-54 ปี 38 ล้านคน
- ประชากร 55-64 ปี 8.4 ล้านคน
- ประชากร 65 ปีขึ้นไป 7.9 ล้านคน

ภาพ 1 สัตตอายุประชากรไทย ในปี 2563

ที่มา: ดัดแปลงมาจาก: (Thai population statistics, 2020)

ในปัจจุบันเทคโนโลยีของหุ่นยนต์เจริญก้าวหน้าอย่างก้าวกระโดด และเริ่มเข้ามามีบทบาทกับชีวิตของมนุษย์ในด้านต่าง ๆ ไม่ว่าจะเป็นอุตสาหกรรมการผลิต งานสำรวจ การเกษตร และการแพทย์ (Robotics for Aging Society, 2019) เพื่อนำมาช่วยงานมนุษย์ งานที่มนุษย์ไม่สามารถทำได้หรือทำได้น้อยกว่า ซึ่งหุ่นยนต์สามารถแก้ปัญหาได้หลากหลายแล้วแต่นำไปใช้ในวัตถุประสงค์ไหน จากปัญหาที่กล่าวมาข้างต้น ผู้วิจัยจึงมีแนวคิดสร้างหุ่นยนต์ในราคาประหยัด เพื่อให้คนทั่วไปได้เข้าถึงเทคโนโลยีหุ่นยนต์ได้ง่ายขึ้น ซึ่งงานวิจัยนี้เป็นการออกแบบและพัฒนาหุ่นยนต์เคลื่อนที่เพื่อช่วยในการหยิบจับสิ่งของขนาดเล็กสำหรับผู้ป่วย หรือคนที่ขยับตัวลำบาก ซึ่งผู้ป่วยจะสามารถควบคุมหุ่นยนต์ด้วยแอปพลิเคชันบนสมาร์ตโฟนผ่านสัญญาณไร้สาย พร้อมกับแสดงภาพจากกล้องที่ติดกับหุ่นยนต์บนหน้าจอสมาร์ตโฟนได้ตามเวลาจริง สำหรับส่วนของหุ่นยนต์เคลื่อนที่ที่ใช้ สัญญาณ Pulse Width Modulation ควบคุมความเร็วและทิศทางของมอเตอร์กระแสตรง และส่วนของแขนหุ่นยนต์เป็นการประยุกต์นำเอาทฤษฎีจลนศาสตร์แบบไปข้างหน้าและจลนศาสตร์ผกผัน เพื่อมาวิเคราะห์หาพื้นที่ทำงานของแขนหุ่นยนต์ อีกทั้งยังพัฒนาซอฟต์แวร์ให้การควบคุมแขนหุ่นยนต์ได้ง่ายขึ้น คือระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-Axis Linear Motion Control -MALM Control) เป็นการควบคุมมอเตอร์หลายตัวพร้อมกัน เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่ตามแนวหน้า-หลัง และแนวขึ้น-ลง เหมาะสำหรับการควบคุมผ่านหน้าจอสมาร์ตโฟน และมีการพัฒนาแอปพลิเคชันบนโทรศัพท์มือถือ ใช้ในการควบคุมแขนหุ่นยนต์ โดยมีหน้าจอที่แสดงภาพจากกล้องที่ติดอยู่กับแขนหุ่นยนต์ เพื่อให้ผู้ป่วยสามารถควบคุมแขนหุ่นยนต์ได้สะดวกขึ้น โดยสามารถควบคุมผ่านสัญญาณไร้สาย (Wireless) และแสดงภาพตามเวลาจริง (Real Time)

1.2 จุดมุ่งหมายของการศึกษา

1. พัฒนาหุ่นยนต์ต้นแบบสำหรับบริการสำหรับผู้ป่วยติดเตียง
2. พัฒนาหุ่นยนต์สามารถเคลื่อนไปในตำแหน่งต่าง ๆ ได้
3. มีแขนหุ่นยนต์ขนาดเล็กเพื่อช่วยในการหยิบจับสิ่งของน้ำหนักเบาตามทิศทางที่กำหนด
4. สามารถควบคุมโดยผู้ใช้แบบไร้สายได้
5. พัฒนาแอปพลิเคชันบนโทรศัพท์มือถือเพื่อใช้ในการควบคุมการเคลื่อนที่ และแขนหุ่นยนต์
6. สามารถแสดงภาพจากกล้องที่ติดกับหุ่นยนต์บนแอปพลิเคชันแบบเรียลไทม์ได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. องค์ความรู้ในการพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียงต้นแบบ
2. ได้นวัตกรรมหุ่นยนต์และซอฟต์แวร์สำหรับควบคุมหุ่นยนต์
3. ได้แอปพลิเคชันสำหรับควบคุมหุ่นยนต์
4. ได้ระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-Axis Linear Motion Control -MALM Control) สำหรับควบคุมแขนหุ่นยนต์

1.4 ขอบเขตของงานวิจัย

1. สร้างหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียงต้นแบบ
2. หุ่นยนต์เคลื่อนที่โดยผู้ใช้งานแอปพลิเคชันบนสมาร์ตโฟน
3. แขนหุ่นยนต์ที่พัฒนาขึ้นสามารถหยิบจับสิ่งของที่มึ่รูปร่างไม่ซับซ้อน น้ำหนักไม่เกิน 100 กรัม และมีขนาดเล็ก
4. ระยะการควบคุมแบบไร้สายได้ไกลสุดไม่น้อยกว่า 20 เมตร

1.5 สมมติฐานของการวิจัย

หุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง เป็นหุ่นยนต์ที่พัฒนาขึ้นมาสำหรับช่วยเหลือผู้ที่ขยับตัวลำบาก หรือต้องเคลื่อนที่เพื่อลูกไปหยิบจับสิ่งของ เช่น ผู้ป่วย ผู้สูงอายุ เป็นต้น โดยหุ่นยนต์มีความสามารถในการเคลื่อนที่ไปยังตำแหน่งต่าง ๆ ตามที่ผู้ใช้กำหนด อีกทั้งมีแขนหุ่นยนต์สำหรับหยิบจับสิ่งของ ควบคุมโดยผู้ป่วยหรือผู้ดูแลผ่านแอปพลิเคชันบนโทรศัพท์มือถือ นอกจากนี้ความสามารถหลักในการเคลื่อนที่ไปหยิบจับสิ่งของแล้ว หุ่นยนต์นี้ยังมีความสามารถอื่น ๆ อีก เช่น ในตัวหุ่นยนต์เคลื่อนที่มีเซนเซอร์ตรวจจับวัตถุ เมื่อหุ่นยนต์เข้าใกล้สิ่งกีดขวางในระยะที่กำหนดจะมีเสียงสัญญาณแจ้งเตือนให้ผู้ใช้งานได้ทราบ เพื่อป้องกันการชนเข้ากับสิ่งกีดขวาง เป็นการลดข้อผิดพลาดจากจุดอับสายตาของมุกกล้อง และหุ่นยนต์เคลื่อนที่ยังมีเซนเซอร์เอ็นโค้ดเดอร์ที่สามารถวัดความเร็ว ระยะทาง และทิศทางการหมุนของล้อได้อีกด้วย ส่วนแขนหุ่นยนต์จะมีการควบคุมสองแบบคือควบคุมแบบที่ละข้อต่ออิสระ และแบบการควบคุมมอเตอร์หลายตัวพร้อมกันซึ่งเป็นระบบระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่ตามแนวหน้า-หลัง และแนวขึ้น-ลง เหมาะสำหรับการควบคุมผ่านหน้าจอสมาธิ์โฟน ในงานวิจัยนี้มุ่งเน้นทำหุ่นยนต์ที่มีราคาประหยัดเพื่อให้คนไทยสามารถเข้าถึงและใช้งานเทคโนโลยีหุ่นยนต์ได้มากยิ่งขึ้น การเลือกวัสดุอุปกรณ์บางอย่างจึงมีข้อจำกัดทางด้านราคา เพื่อให้สามารถทำหุ่นยนต์ให้มีราคาประหยัดมากที่สุดและมีประสิทธิภาพที่ดีสามารถใช้งานได้อย่างสมบูรณ์

1.6 วิธีดำเนินการวิจัย

1. ศึกษาหลักการที่เกี่ยวข้อง

เนื่องจากการพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียงมีการใช้ในวิชาหลายแขนง ดังนั้นจึงต้องมีการศึกษาทฤษฎีและข้อมูลก่อน ไม่ว่าจะเป็นการศึกษาทฤษฎีพื้นฐานการทำงานของแขนหุ่นยนต์ เช่น degrees of freedom, ศึกษาและพัฒนา workspace หรือ พื้นที่การทำงานที่ปลายแขนหุ่นยนต์สามารถเคลื่อนที่ไปได้, ศึกษาและพัฒนา โครงสร้างแขนแต่ละท่อนและข้อต่อ ส่วนต่อมา

ก็ศึกษาทฤษฎีพื้นฐานของการเคลื่อนที่ของหุ่นยนต์ และส่วนสุดท้ายศึกษาการทำแอปพลิเคชันเพื่อ ออกแบบให้ควบคุมหุ่นยนต์ได้

2. การออกแบบฮาร์ดแวร์และซอฟต์แวร์

หุ่นยนต์บริการสำหรับผู้พิการติดเตียงที่จะถูกพัฒนาขึ้นมา นั้น จำเป็นต้องมีการออกแบบทั้ง ฮาร์ดแวร์และซอฟต์แวร์ ในส่วนของการออกแบบฮาร์ดแวร์ต้องคำนึงถึงลักษณะการนำไปใช้งานของ หุ่นยนต์ ต้องมีขนาดที่ความสูงที่สัมพันธ์กับโต๊ะหรือที่วางของ เพื่อที่แขนหุ่นยนต์จะสามารถหยิบ สิ่งของได้ การเลือกวัสดุอุปกรณ์ เช่น มอเตอร์ต้องมีแรงบิดที่เพียงพอต่อลักษณะการใช้งานของ หุ่นยนต์ เซนเซอร์ตรวจจับวัตถุต้องมีความแม่นยำในการตรวจจับวัตถุ เป็นต้น ต่อมาส่วนของ ซอฟต์แวร์ ซอฟต์แวร์ของแขนหุ่นยนต์มีการพัฒนาระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-Axis Linear Motion Control -MALM Control) เป็นการควบคุมมอเตอร์หลายตัวพร้อมกัน เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่ตามแนวหน้า-หลัง และแนวขึ้น-ลง เหมาะสำหรับการควบคุมผ่าน หน้าจอสมาร์โฟน การพัฒนาระเบียบวิธีในการวางแผนการเคลื่อนที่แขนหุ่นยนต์ ในการปรับเปลี่ยน มุมมองและการเข้าจับวัตถุ

3. การทดลอง

ทดลองการทำงานต่าง ๆ ของหุ่นยนต์ว่ามีประสิทธิภาพตรงตามขอบเขตที่กำหนด ทดลอง การทดลองให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรงทั้งเดินหน้า และถอยหลัง โดยการปรับค่า PWM ของล้อ ทั้งสองข้าง ที่ระยะ 5 m แบ่งการทดลองโดยปราศจากน้ำหนักและทดลองด้วยมีภาระน้ำหนัก และ การทดลองการเลี้ยวโค้งของหุ่นยนต์ ต่อมาในด้านการหยิบจับวัตถุนั้นแขนหุ่นยนต์จะทดสอบ ความสามารถในการควบคุมมือจับของแขนหุ่นยนต์เคลื่อนที่เป็นเส้นตรงในแนวหน้า-หลัง และแนว ขึ้น-ลง ซึ่งเป็นการเคลื่อนที่ของแขนหุ่นยนต์ในระนาบสองมิติ การทดลองที่ต่อมาเป็นการเป็นการ ทดสอบด้านระยะเวลาที่ใช้ควบคุมแขนหุ่นยนต์เพื่อหยิบสิ่งของ ทั้งนี้จะมีการทดสอบระบบการ ควบคุมแบบไร้สายจากผู้ใช้งานว่าสามารถควบคุมการเคลื่อนที่และแขนกลของหุ่นยนต์ได้อย่างมี ประสิทธิภาพหรือไม่

1.7 แผนการดำเนินงาน

ตาราง 1 แสดงแผนการดำเนินงาน

กิจกรรม	เดือน											
	พ.ย. 63	ธ.ค. 63	ม.ค. 64	ก.พ. 64	มี.ค. 64	เม.ย. 64	พ.ค. 64	มิ.ย. 64	ก.ค. 64	ส.ค. 64	ก.ย. 64	
1. ศึกษาทฤษฎีพื้นฐานการเคลื่อนที่ของหุ่นยนต์ และการหลบหลีกสิ่งกีดขวาง	←→											
2. ศึกษาทฤษฎีพื้นฐานของการจับและการทำงานของแขนหุ่นยนต์		←→										
3. พัฒนาหุ่นยนต์ต้นแบบ ออกแบบฮาร์ดแวร์และซอฟต์แวร์				←→								
4. พัฒนาแอปพลิเคชันสำหรับการควบคุมหุ่นยนต์												
5. พัฒนาระเบียบวิธีในการวางแผนการเคลื่อนที่แขนหุ่นยนต์ ใน การปรับเปลี่ยนมุมมองและการเข้าจับวัตถุ							←→					
6. ทำการทดลองและบันทึกผลการทดลอง							←→					
7. วิเคราะห์ผลการทดลอง								←→				
8. ปรับปรุงแก้ไขงาน								←→				
9. จัดทำรูปเล่มวิทยานิพนธ์									←→			

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

สำหรับทฤษฎีและงานวิจัยที่เกี่ยวข้องกับงานวิจัย สามารถแบ่งออกเป็นสามส่วนคือ ส่วนแรกเป็นความหมายและชนิดของหุ่นยนต์แต่ละประเภท หลักการพื้นฐานของหุ่นยนต์ ส่วนที่สองจะเป็นส่วนของหุ่นยนต์เคลื่อนที่ (Mobile Robot) ประกอบไปด้วยระบบทางกลของหุ่นยนต์ โครงสร้างของหุ่นยนต์เคลื่อนที่ได้จะประกอบไปด้วยกลไกที่ใช้สำหรับขับเคลื่อนหุ่นยนต์ไปยังบริเวณที่ต้องการ โดยการขับเคลื่อนด้วยล้อ และระบบควบคุมหุ่นยนต์ ประกอบด้วยอุปกรณ์ควบคุมซึ่งควบคุมการทำงานของหุ่นยนต์โดยใช้ไมโครคอนโทรลเลอร์ ระบบควบคุมนี้ทำหน้าที่เป็นสมองติดต่อกับอุปกรณ์ต่างๆ เช่น เซนเซอร์ มอเตอร์ เป็นต้น โดยสั่งหุ่นยนต์ให้ทำงานตรวจสอบและควบคุมรายละเอียดของการทำงานให้ถูกต้อง และส่วนที่สามจะเป็นส่วนของแขนหุ่นยนต์ ประกอบไปด้วยข้อต่อ (link) ที่นำมาประกอบกันด้วยข้อต่อ (joint) อีกทั้งยังได้นำเอาทฤษฎีจลนศาสตร์แบบไปข้างหน้าและแบบผกผันมาประยุกต์ใช้เพื่อหาพื้นที่ทำงานของแขนหุ่นยนต์ และออกแบบระบบควบคุมเซอร์โวมอเตอร์หลายตัวพร้อมกัน

2.1 หลักการและทฤษฎีที่เกี่ยวข้อง

2.1.1 ความหมายและความสำคัญของของหุ่นยนต์

หุ่นยนต์ (robot) คือเครื่องจักรกลชนิดหนึ่ง มีลักษณะโครงสร้างและรูปร่างแตกต่างกัน หุ่นยนต์ในแต่ละประเภทจะมีหน้าที่การทำงานในด้านต่าง ๆ สามารถทำงานที่ซ้ำ ๆ และซับซ้อนได้รวมทั้ง งานที่มีความยากลำบากที่มนุษย์ไม่สามารถทำได้ ซึ่งการควบคุมหุ่นยนต์สามารถควบคุมได้โดยตรงกับมนุษย์หรือการควบคุมแบบอัตโนมัติ การควบคุมระบบต่าง ๆ สามารถทำได้โดยทางอ้อมและอัตโนมัติ ในการสั่งงานระหว่างหุ่นยนต์และมนุษย์ ซึ่งโดยทั่วไปแล้วหุ่นยนต์ถูกสร้างขึ้นเพื่อสำหรับงานที่มีความยากลำบากหรืองานที่เป็นอันตรายต่อมนุษย์ เช่น งานสำรวจในพื้นที่บริเวณแคบ งานสำรวจดาวเคราะห์ที่ไม่มีสิ่งมีชีวิต หรืองานสำรวจใต้ท้องทะเลลึก ซึ่งในปัจจุบันเทคโนโลยีของหุ่นยนต์มีความเจริญก้าวหน้าอย่างรวดเร็ว และได้เริ่มเข้ามามีบทบาทกับชีวิตของมนุษย์ในด้านต่าง ๆ เช่น หุ่นยนต์ที่ใช้ในทางการแพทย์ หุ่นยนต์ที่ใช้ทางด้านการเกษตร หุ่นยนต์สำหรับงานสำรวจ หุ่นยนต์ที่ใช้ในงานด้านอุตสาหกรรมการผลิต และหุ่นยนต์ที่เข้ามาช่วยเพิ่มความอำนวยความสะดวกภายในชีวิตประจำวันของมนุษย์

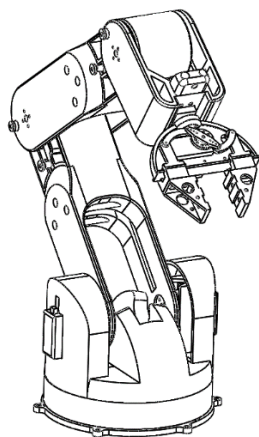
2.1.2 ชนิดของหุ่นยนต์

หุ่นยนต์สามารถแบ่งแยกได้หลากหลายรูปแบบตามลักษณะเฉพาะของการทำงาน ซึ่งถูกแบ่งแบ่งออกเป็น 2 ประเภท ได้แก่ การแบ่งแยกตามการเคลื่อนที่ และการแบ่งแยกตามลักษณะรูปร่างภายนอกของหุ่นยนต์

การแบ่งประเภทตามการเคลื่อนที่ได้

1. หุ่นยนต์ที่ติดตั้งอยู่กับที่ ไม่สามารถเคลื่อนที่ได้ ที่ (fixed robot)

หุ่นยนต์ที่ติดตั้งอยู่กับที่คือ หุ่นยนต์ที่ไม่สามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง แต่สามารถเคลื่อนไหวไปมาได้ เช่น แขนของหุ่นยนต์ ที่สามารถเคลื่อนไหวและขยับเฉพาะแต่ละข้อต่อภายในตัวเองเท่านั้น เท่านั้น มักนำไปใช้ในงานด้านต่าง ๆ เช่น แขนของหุ่นยนต์ที่ใช้ในงานด้านการแพทย์ นำแขนหุ่นยนต์ไปใช้ในการผ่าตัด อุตสาหกรรมต่าง ๆ อย่างงานด้านอุตสาหกรรมผลิตรถยนต์ โดยหุ่นยนต์ที่ติดตั้งอยู่กับที่ประเภทนี้ มักมีการใช้พลังงานจากแหล่งจ่ายพลังงานภายนอกให้สามารถเคลื่อนไหวได้ และจะมีการกำหนดขอบเขตการเคลื่อนไหวของหุ่นยนต์เอาไว้ ทำให้หุ่นยนต์สามารถเคลื่อนไหวไปมาได้เฉพาะที่ที่กำหนดเอาไว้เท่านั้น หุ่นยนต์ที่ติดตั้งอยู่กับที่ ไม่สามารถเคลื่อนที่ได้ แสดงดังภาพที่ 2

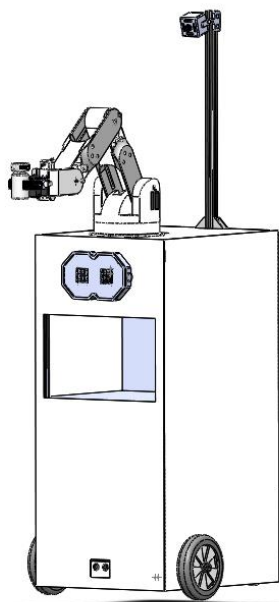


ภาพ 2 หุ่นยนต์ที่ติดตั้งอยู่กับที่ ไม่สามารถเคลื่อนที่ได้

2. หุ่นยนต์ที่สามารถเคลื่อนไหวและเคลื่อนที่ได้

หุ่นยนต์ที่สามารถเคลื่อนที่ได้ หมายความว่าหุ่นยนต์ที่สามารถเคลื่อนย้ายตัวเองจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งได้อย่างอิสระ สามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง โดยการใช้ล้อหรือการใช้ขา หุ่นยนต์ประเภทนี้จะแตกต่างจากหุ่นยนต์ที่ติดตั้งอยู่กับที่ เพราะ โดยประเภทของหุ่นยนต์ สามารถแบ่งแยกได้หลากหลายรูปแบบตามลักษณะเฉพาะของการทำงาน ได้แก่ การแบ่งประเภทตามการเคลื่อนที่ นอกจากนี้อาจจำแนกตามรูปลักษณ์ภายนอกด้วยก็ได้ ซึ่งหุ่นยนต์ที่สามารถเคลื่อนไหวได้นี้ ถูกออกแบบลักษณะของโครงสร้างให้มีขนาดเล็กและมีระบบเคลื่อนที่ไปมา อีกทั้งมี

แหล่งจ่ายพลังงานสำรองหรือแบตเตอรี่จากภายในหุ่นยนต์เอง หุ่นยนต์ที่สามารถเคลื่อนไหวและเคลื่อนที่ได้ แสดงได้ดังภาพที่ 3



ภาพ 3 หุ่นยนต์ที่สามารถเคลื่อนไหวและเคลื่อนที่ได้

2.1.3 ประโยชน์และความสามารถของหุ่นยนต์

ในปัจจุบันเทคโนโลยีหุ่นยนต์ได้รับการพัฒนาอย่างต่อเนื่องในปัจจุบัน และได้เริ่มเข้ามามีบทบาทกับชีวิตประจำวันของมนุษย์เรื่อยมา การพัฒนาเทคโนโลยีหุ่นยนต์ตลอดมาทำให้ ความสามารถของหุ่นยนต์พัฒนาขึ้นอย่างรวดเร็ว สามารถทำงานต่าง ๆ เพื่อทดแทนงานบางอย่างที่มนุษย์ไม่สามารถทำได้หรือทำได้น้อยกว่า ซึ่งประโยชน์และความสามารถของหุ่นยนต์สามารถแบ่งประเภทได้ ดังนี้

1. ความสามารถในการแพทย์

ปัจจุบันจึงได้มีการพัฒนาหุ่นยนต์ขึ้นมาเพื่อใช้ในทางการแพทย์ เช่น หุ่นยนต์ส่งยาหรืออาหาร การใช้แขนหุ่นยนต์มาช่วยทำการผ่าตัดคนไข้ เนื่องจากหุ่นยนต์นั้นสามารถทำงานในด้านที่มีความละเอียดสูงที่เกินกว่ามนุษย์จะทำได้ เช่น การผ่าตัดตมอม เป็นต้น, หุ่นยนต์ที่ช่วยให้ผู้ป่วยสามารถฟื้นฟูสภาพร่างกายได้ เช่น ผู้ป่วยที่เส้นเลือดในสมองอุดตัน เป็นต้น หุ่นยนต์ที่ช่วยดูแลผู้ป่วย ผู้สูงอายุ เช่น หุ่นยนต์จ่ายยา เป็นต้น และหุ่นยนต์ที่ทำงานซ้ำ ๆ เช่น หุ่นยนต์ฉายแสงยูวีเพื่อทำความสะอาดห้อง (กำจัดเชื้อโรค), หุ่นยนต์เคลื่อนที่สำหรับขนส่งอุปกรณ์และเครื่องมือทางการแพทย์ และยา เป็นต้น

2. ความสามารถในงานวิจัย

หุ่นยนต์สามารถพัฒนาด้านวิทยาศาสตร์และทำวิจัยให้กับมนุษย์ เช่น สำรวจท้องมหาสมุทรที่มีความลึกเป็นอย่างมากที่มนุษย์ไม่สามารถเข้าไปถึงได้ การสำรวจอวกาศ หรือกระทั่งงานสำรวจบริเวณปากปล่องภูเขาไฟเพื่อเก็บบันทึกข้อมูลการเปลี่ยนแปลงต่าง ๆ ที่มีความเสี่ยงกับมนุษย์ เป็นต้น งานเหล่านี้ล้วนแล้วเป็นงานที่เกินขอบเขตความสามารถของมนุษย์ และยังมีภัยอันตรายที่ไม่สามารถปฏิบัติงานสำรวจเช่นนี้ได้ จนทำให้ปัจจุบันนี้ ได้มีการพัฒนาหุ่นยนต์เพื่อใช้ในสำรวจและวิจัย เพื่อให้หุ่นยนต์สามารถทนต่อสภาพแวดล้อม ทั้งนี้ยังมีระบบคอนโทรลที่มีเซนเซอร์ติดตั้งที่ตัวหุ่นยนต์ เพื่อให้มนุษย์สามารถทำการควบคุมหุ่นยนต์ได้ในระยะไกล สำหรับใช้การสำรวจวัดระยะทาง และงานวิจัยเก็บข้อมูลส่วนต่าง ๆ ทางด้านวิทยาศาสตร์

3. ความสามารถในงานอุตสาหกรรม

งานอุตสาหกรรมด้านการเทคโนโลยีหุ่นยนต์ได้มีบทบาทเป็นอันดับแรก ๆ ซึ่งแต่ก่อนงานในด้านอุตสาหกรรม แรงงานจำเป็นต้องใช้เป็นจำนวนมาก จึงมีการจ้างแรงงานที่เป็นมนุษย์จำนวนมากเพื่อใช้ในงานอุตสาหกรรม ทำให้ต้นทุนในการจ้างงานสูง และยังมีงานบางอย่างที่มนุษย์นั้นไม่สามารถเข้าไปทำงานได้ในงานอุตสาหกรรมบางประเภท เช่น งานที่มีความเสี่ยงและเป็นอันตราย หรืองานอุตสาหกรรมที่ต้องใช้ความแม่นยำและรวดเร็วในการผลิต ทำให้เทคโนโลยีหุ่นยนต์กลายเป็นทางออกของงานด้านอุตสาหกรรม ทั้งด้านต้นทุน และระยะเวลาในการผลิตลดลงแต่ได้ชิ้นงานเพิ่มขึ้น

4. ความสามารถในด้านความมั่นคง

หุ่นยนต์ก็มีบทบาทหลากหลายด้าน ซึ่งด้านความมั่นคงเทคโนโลยีหุ่นยนต์ก็ได้เข้ามามีบทบาท เช่น โดรนใต้น้ำสำหรับต่อต้านเรือดำน้ำ เครื่องบินสอดแนมผู้ก่อการร้ายซึ่งติดตั้งเรดาร์คอยตรวจจับสิ่งต้องสงสัย หรือหุ่นยนต์สำหรับตรวจจับวัตถุระเบิดและเก็บกู้ เป็นต้น

5. ความสามารถในด้านบันเทิง

หุ่นยนต์ได้เข้ามามีบทบาทใกล้ชิดกับชีวิตประจำวันของมนุษย์มากยิ่งขึ้น เพื่ออำนวยความสะดวก กระทั่งสร้างความบันเทิง ซึ่งหุ่นยนต์ประเภทนี้ได้รับการพัฒนาให้สามารถตอบโต้ เล่นหรือสัตว์เลื้อย ซึ่งมีในรูปแบบต่าง ๆ เช่น แมว สุนัข หรือมีรูปลักษณ์ที่ออกแบบที่ดูเป็นมิตรกับมนุษย์ และกระทั่งสร้างความบันเทิงทางเพศให้กับมนุษย์ได้อีกด้วย เป็นต้น

6. ความสามารถในงานครัวเรือน

หุ่นยนต์สามารถอำนวยความสะดวกให้ชีวิตประจำวันของมนุษย์ง่ายขึ้น สามารถทำงานที่ซ้ำ ๆ ได้ เพื่อทดแทนแรงงานที่มีความปลอดภัยไม่ต้องจ้างแรงงาน

7. ความสามารถในด้านการเกษตร

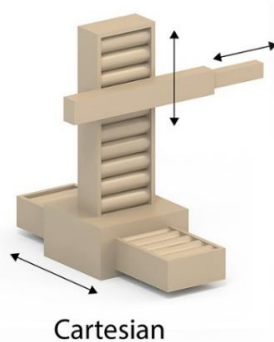
หุ่นยนต์ในปัจจุบันสามารถนำมาช่วยงานด้านการเกษตร เพื่อในการลดค่าใช้จ่ายในการจ้างแรงงาน สามารถเพิ่มผลผลิตได้มากขึ้น และลดการสูญเสียปัจจัยการผลิตที่มีผลให้ค่าแรงแพง

2.1.4 ประเภทของแขนหุ่นยนต์

แขนหุ่นยนต์แบ่งเป็น 2 ประเภทหลัก ๆ ได้แก่ แขนหุ่นยนต์โครงสร้างอนุกรม (serial link manipulator) และแขนหุ่นยนต์โครงสร้างขนาน (parallel link manipulator)

แขนหุ่นยนต์โครงสร้างอนุกรม (serial link manipulator)

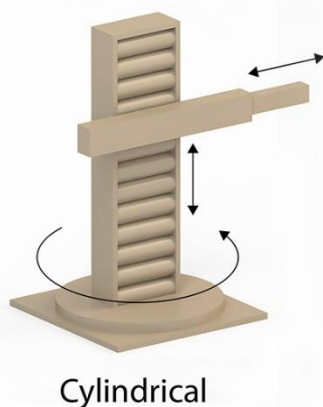
เป็นแขนหุ่นยนต์ที่มีข้อต่อที่มีต้นกำลังขับเคลื่อนต่อกันไปเรื่อย ๆ แบบอนุกรม ตัวอย่างแขนหุ่นยนต์โครงสร้างอนุกรม เช่น Cartesian robot arm เป็นแขนกลที่เคลื่อนที่บนแกนตั้งฉากกัน 3 แกน คือ ขึ้น-ลง, ซ้าย-ขวา, หน้า-หลัง แสดงดังภาพที่ 4



ภาพ 4 Cartesian robot arm

ที่มา: (Sumipol, 2020)

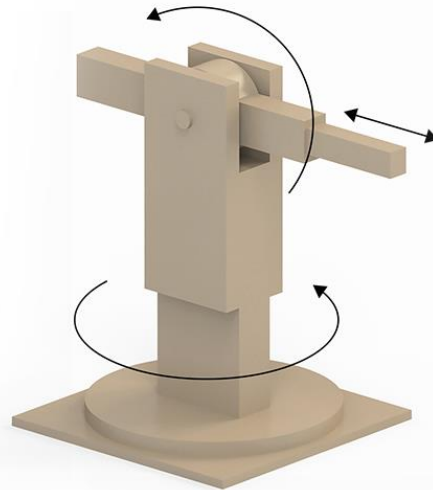
Cylindrical robot arm เป็นแขนกลที่เคลื่อนที่ในพื้นที่ยกทรงกระบอก คือ ขึ้น - ลง, เข้า - ออก ตามแนวรัศมี, หมุนรอบแกน แสดงดังภาพที่ 5



ภาพ 5 Cylindrical robot arm

ที่มา: (Sumipol, 2020)

Spherical robot arm เป็นแขนกลที่เคลื่อนที่ในพื้นที่ทรงกลม คือ หมุนซ้าย - ขวา, หมุนขึ้น - ลง, เลื่อนเข้า - ออกตามแนวรัศมี แสดงดังภาพที่ 6

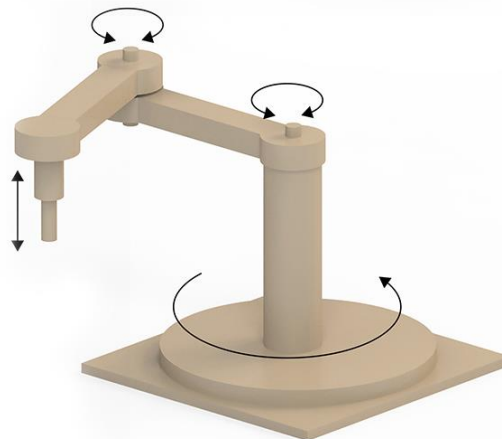


Polar

ภาพ 6 Spherical robot arm

ที่มา: (Sumipol, 2020)

Scara robot arm เป็นแขนกลที่เคลื่อนที่มีแกนหมุน 2 แกนขนานกัน เพื่อเคลื่อนที่ได้บนระนาบ และอีกแกนสำหรับขึ้น-ลง แสดงดังภาพที่ 7

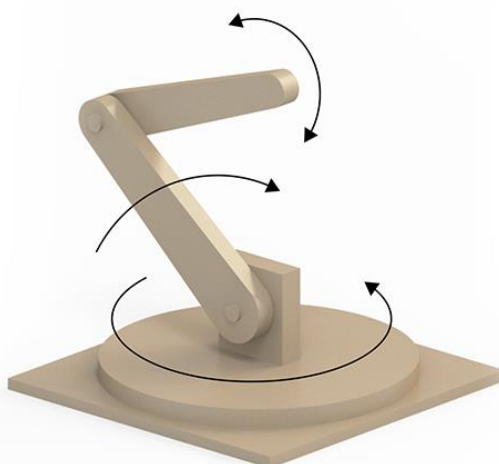


Scara

ภาพ 7 Scara robot arm

ที่มา: (Sumipol, 2020)

Articulate robot arm หรือ Jointed Arm เป็นแขนกลที่เคลื่อนที่ในพื้นที่ค่อนข้างเป็นทรงกลม คล้ายในพวงรถชุด/แบคโฮ คือ หมุนซ้าย-ขวา, และมีแกนหมุนอีก 2 แกนที่ขนานกันเพื่อเคลื่อนที่ในระนาบแนวตั้ง แสดงดังภาพที่ 8

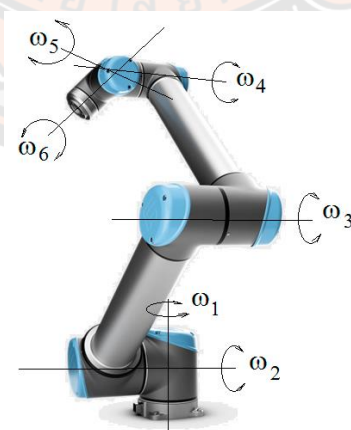


Jointed-arm

ภาพ 8 Articulate robot arm

ที่มา: (Sumipol, 2020)

Anthropomorphic arm เป็นแขนกลที่มีลักษณะแบบแขนมนุษย์ คือ มีไหลที่หมุนได้ 2 แกน ซ้าย-ขวา, บน-ล่าง มีแกนหมุนรอบตัวเองที่โครงสร้างแกนแรก และมีแกนหมุนที่ข้อศอก แสดงดังภาพที่ 9



ภาพ 9 Anthropomorphic arm

ที่มา: (Staretu, 2021)

ข้อดีแขนหุ่นยนต์โครงสร้างอนุกรม

มีพื้นที่การทำงานมาก เพราะข้อต่อต่อกันออกไปเรื่อย ๆ ทำให้สามารถขยับการทำงานได้หลากหลาย และการคำนวณการควบคุมค่อนข้างง่าย ตรงไปตรงมา เพียงแค่นำ transformation matrix ของแต่ละแกนคูณกันไปเรื่อย ๆ

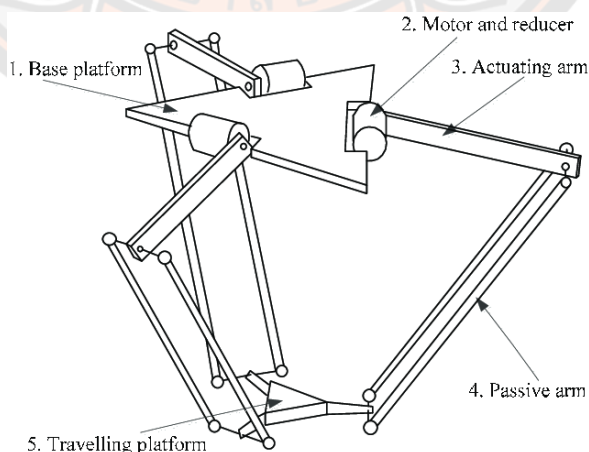
ข้อเสียแขนหุ่นยนต์โครงสร้างอนุกรม

มีความแข็งแรงต่ำและความเที่ยงตรง (precision) ต่ำ เพราะมีลักษณะเป็นโครงสร้างยื่นยาวออกไปจากฐานทำให้ความคลาดเคลื่อนจะสะสมเพิ่มขึ้นตามลำดับของข้อต่อ การเคลื่อนที่ช้า เพราะการติดตั้งต้นกำลังไปที่แต่ละข้อต่อทำให้แขนมีมวลมาก และมีกำลังในการออกแรงที่น้อย เพราะโครงสร้างไม่แข็งแรงมาก และการติดตั้งต้นกำลังที่มีกำลังมากก็จะมีน้ำหนักมากตาม

แขนหุ่นยนต์โครงสร้างขนาน (parallel link manipulator)

เป็นแขนหุ่นยนต์ที่มีข้อต่อที่มีต้นกำลังขับเคลื่อนหลายตัวอยู่บริเวณฐานและมีโครงสร้างข้อต่อจากต้นกำลังแต่ละตัวไปยังปลายแขนเป็นโครงสร้างแบบปิดเป็นวงรอบ

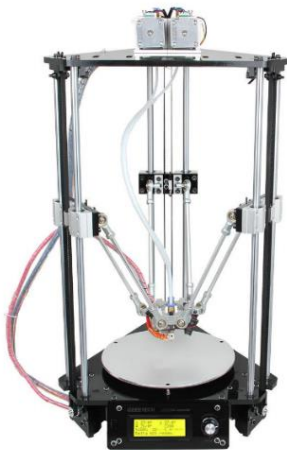
Delta Robot จะมีจุดแข็งในเรื่อง ความเร็ว ความแม่นยำ และความแข็งแรงทนทาน ด้วยส่วนประกอบระบบเซ็นเซอร์และระบบการมองเห็นที่ติดตั้งไว้ที่แขนหุ่นยนต์ ทำให้การจับและการวางสิ่งของที่กำลังเคลื่อนที่มีความแม่นยำสูงมากดังนั้นหุ่นยนต์ประเภทนี้จึงเหมาะกับการประกอบชิ้นส่วน แพคเกจจิ้ง จัดวัตถุดิบตามหมวดหมู่ และการขนส่ง แสดงดังภาพที่ 10



ภาพ 10 Delta Robot

ที่มา: (Lu & Liu, 2016)

Rostock delta robot เป็นหุ่นยนต์ที่นิยมใช้ในเครื่องปริ้นสามมิติ ซึ่งสามารถพิมพ์โมเดลได้รวดเร็วและแม่นยำ แสดงดังภาพที่ 11



ภาพ 11 Rostock delta robot

ที่มา: (Geeetech Wiki, 2015)

Stewart platform เป็นอุปกรณ์หุ่นยนต์ขนานที่ใช้สำหรับการวางตำแหน่งและการควบคุมการเคลื่อนไหว อุปกรณ์ประกอบด้วยแผ่นสองแผ่นที่มีขาปรับได้หกตัวที่อนุญาตให้ปรับและควบคุมได้อย่างแม่นยำ แสดงดังภาพที่ 12



ภาพ 12 Stewart platform

ที่มา: (Ovaktechnologies, 2016)

ข้อดีของแขนหุ่นยนต์โครงสร้างขนาน

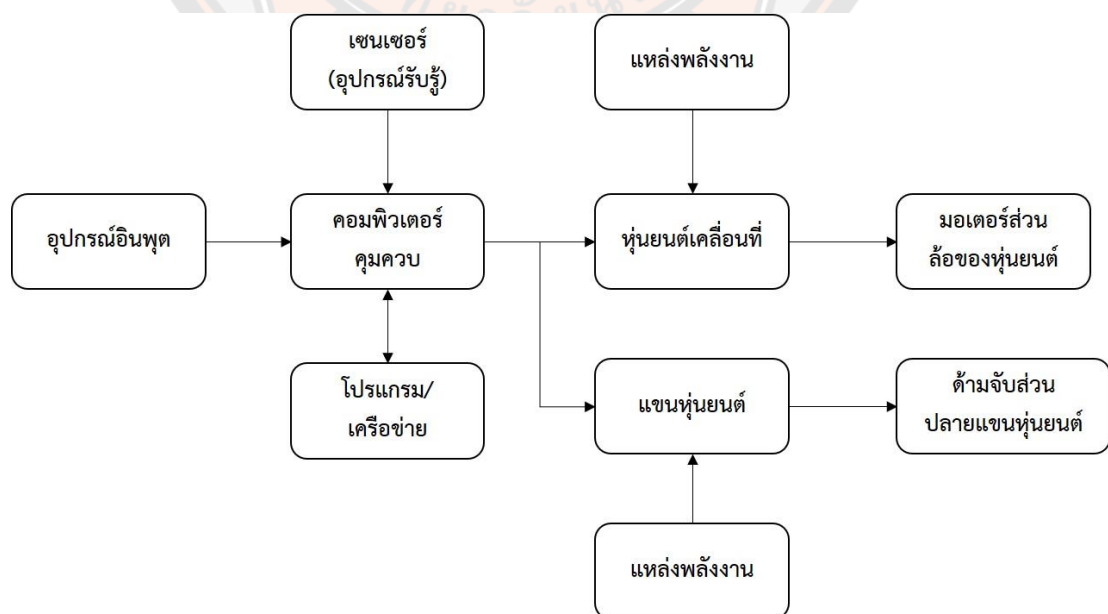
ความแข็งแรงสูงและความเที่ยงตรง (precision) สูง เพราะเป็นโครงสร้างแบบปิดทำให้ปลายแขนต่อมายังต้นกำลังทุกตัวที่ฐาน และยังคงเคลื่อนที่ได้เร็ว เพราะมวลส่วนใหญ่อยู่ที่ฐาน ปลายแขนจึงมีมวลน้อย การออกแรงได้มาก เพราะต้นกำลังทุกตัวช่วยกันออกแรง

ข้อเสียของแขนหุ่นยนต์โครงสร้างขนาน

พื้นที่การทำงานแคบและมีความซับซ้อนในการคำนวณการควบคุม

2.1.5 ส่วนประกอบของหุ่นยนต์

หุ่นยนต์จำเป็นต้องอาศัยความรู้ด้านวิศวกรรมศาสตร์ และวิทยาศาสตร์หลายสาขา เช่น วิศวกรรมเครื่องกล วิศวกรรมไฟฟ้า วิศวกรรมคอมพิวเตอร์ คณิตศาสตร์ อีกทั้งยังต้องอาศัยด้านวิศวกรรมวัสดุและวิศวกรรมอุตสาหการ ศาสตร์ทางด้านหุ่นยนต์จึงเป็นวิชาที่ต้องมีการวิจัยและพัฒนาอย่างต่อเนื่อง โดยเฉพาะทางด้านคอมพิวเตอร์ที่เป็นส่วนควบคุมเพื่อเพิ่มความฉลาดให้หุ่นยนต์ แผนภาพส่วนประกอบหลักของหุ่นยนต์แสดงได้ดังภาพที่ 13 หุ่นยนต์จะถูกควบคุมโดยคอมพิวเตอร์ที่ทำงานตามโปรแกรมที่เก็บไว้ในตัวควบคุม อุปกรณ์ส่วนขับเคลื่อนของหุ่นยนต์ เช่น มอเตอร์ส่วนขอล้อในหุ่นยนต์เคลื่อนที่ ด้านจับส่วนปลายแขนหุ่นยนต์ เพื่อให้การทำงานมีประสิทธิภาพจะต้องมีตัวรับรู้ข้อมูลที่สำคัญ เช่น เซ็นเซอร์วัดตำแหน่งและความเร็วของล้อหุ่นยนต์ กล้องเพื่อรับภาพ เป็นต้น และหุ่นยนต์จะต้องมีแหล่งจ่ายกำลังเพื่อให้สามารถทำงานได้



ภาพ 13 แผนภาพส่วนประกอบหลักของหุ่นยนต์

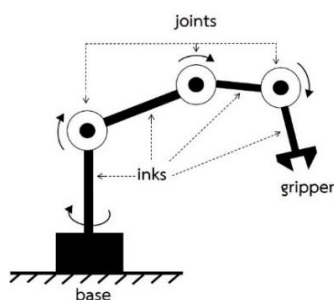
โดยโครงสร้างหุ่นยนต์สามารถแยกส่วนประกอบย่อยสองส่วน คือ ส่วนประกอบทางไฟฟ้า และคอมพิวเตอร์ และส่วนประกอบทางกล

ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์

ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์พื้นฐานของหุ่นยนต์ การทำงานของหุ่นยนต์เปรียบเสมือนกับการทำงานของมนุษย์เมื่อต้องการทำงานอย่างหนึ่ง เช่นหยิบของบนโต๊ะ มอเตอร์เสมือนเป็นกล้ามเนื้อที่ทำให้แขน ข้อมือ ข้อศอกและนิ้วมือ เคลื่อนที่สู่เป้าหมายได้ หากต้องการให้การทำงานได้อย่างแม่นยำต้องมีการป้อนกลับผ่านตัวรับรู้ เช่น กล้องในการรับภาพเปรียบตั้งส่วนของดวงตา หรือเซ็นเซอร์ต่าง ๆ เปรียบเสมือนประสาทสัมผัสต่าง ๆ ส่วนตัวประมวลผลทำหน้าที่เป็นสมองควบคุมการทำงานทั้งหมดของหุ่นยนต์ หากหุ่นยนต์ต่อกับเครือข่ายก็เปรียบได้กับคนงานรับคำสั่งคนรับสั่งคำสั่งกันผ่านอุปกรณ์สื่อสาร

ส่วนประกอบทางกลของหุ่นยนต์

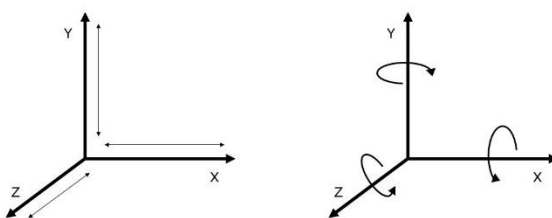
ส่วนประกอบหลักทางกลของหุ่นยนต์มีความสำคัญในการวิเคราะห์หุ่นยนต์ โดยเฉพาะส่วนของแขนหุ่นยนต์ เริ่มจากส่วนของหุ่นยนต์เคลื่อนที่ส่วนประกอบสำคัญได้แก่ส่วนการขับเคลื่อน คือมีมอเตอร์ที่ช่วยในการขับเคลื่อนล้อของหุ่นยนต์ ต่อมากมีอุปกรณ์รับรู้ต่าง ๆ เช่น กล้อง เซนเซอร์วัดตำแหน่งและความเร็วของหุ่นยนต์ และเซนเซอร์ตรวจจับสิ่งกีดขวาง ต่อมาส่วนประกอบหลักทางกลของแขนหุ่นยนต์ เริ่มจาก ฐาน (base) ที่ยึดติดกับพื้นหรือติดตั้งบนตัวหุ่นยนต์เคลื่อนที่ ข้อต่อ (joints) คือตัวช่วยให้แขนเคลื่อนที่ได้ เปรียบได้กับอวัยวะของมนุษย์คือไหล่ ข้อศอก ข้อมือ เป็นต้น ส่วนที่เชื่อมต่อระหว่างข้อต่อเรียกว่า ก้านต่อ (inks) เปรียบได้กับท่อนแขนมนุษย์ และ ส่วนสัมผัสกับชิ้นงานเรียกว่า มือจับ (gripper) สำหรับหยิบจับสิ่งของและเคลื่อนย้ายวัตถุ ส่วนประกอบทางกลของแขนหุ่นยนต์ แสดงดังภาพที่ 14



ภาพ 14 ส่วนประกอบทางกลของแขนหุ่นยนต์

2.1.6 วัตถุแข็งเกร็ง (Rigid body)

วัตถุแข็งเกร็ง (Rigid body) หมายถึง ระบบอนุภาคที่ประกอบกันเป็นวัตถุ โดยที่อนุภาคทั้งหลายจะยังคงมีตำแหน่งที่สัมพันธ์ระหว่างกันคงที่ไม่เปลี่ยนแปลง ถึงแม้จะมีแรงภายนอกมากระทำต่อวัตถุนั้น ซึ่งหมายความว่าวัตถุแข็งเกร็งจะเป็นวัตถุที่มีรูปร่างไม่เปลี่ยนแปลงในขณะที่วัตถุมีการเคลื่อนที่นั่นเอง ซึ่งการเคลื่อนที่ของวัตถุเกร็งอาจกล่าวได้เป็นสองแบบคือ การเคลื่อนที่แบบเลื่อนตำแหน่ง (Prismatic or Translational motion) และการเคลื่อนที่แบบหมุน (Rotational motion) รวมกัน การเคลื่อนที่แบบเลื่อนสามารถเคลื่อนที่เลื่อนไปได้ใน 1 แกน 2 แกน หรือ 3 แกน และการเคลื่อนที่แบบหมุนก็สามารถเคลื่อนที่หมุนได้ใน 1 แกน 2 แกน หรือ 3 แกน เช่นเดียวกัน แสดงดังภาพที่ 15



การเคลื่อนที่แบบเลื่อน (ก) การเคลื่อนที่แบบหมุน

ภาพ 15 ชนิดของการเคลื่อนที่

2.1.7 องศาอิสระ (degrees of freedom)

องศาอิสระ คือ จำนวนแกน ทิศทางที่สามารถเคลื่อนที่ได้ ถ้าเปรียบเทียบกับความสามารถในการเคลื่อนที่บนโลกสามมิติของเราก็จะมีการเคลื่อนที่ 3 แกน ขึ้น-ลง, ซ้าย-ขวา, หน้า-หลัง และการหมุนอีก 3 แกน เช่น กรณีของแขนหุ่นยนต์ที่สามารถเคลื่อนที่ได้ในระนาบ และไม่สามารถปรับมุมการหมุนได้ จะมี 2 degrees of freedom ต่อมาแขนหุ่นยนต์ที่เคลื่อนที่ได้ในสามมิติ แต่ไม่สามารถปรับมุมการเอียงของปลายแขนได้จะมี 3 degrees of freedom ต่อมาแขนหุ่นยนต์ที่เคลื่อนที่ได้ในสามมิติ และควบคุมการหมุนของปลายแขนได้รอบทิศทาง ก็จะมี 6 degrees of freedom และหุ่นยนต์ที่มีมากกว่า 6 degrees of freedom คือ หุ่นยนต์ที่สามารถเคลื่อนปลายแขนไปยังตำแหน่งใด ๆ ในสามมิติ และควบคุมการหมุนของปลายแขนได้รอบทิศ ได้มากกว่า 1 ทำทางเพื่อให้ได้ตำแหน่งปลายแขนตำแหน่งเดียวกัน (เช่น แขนคนมี 7 degrees of freedom เราสามารถขยับแขนได้หลายท่า โดยที่ปลายมือเราอยู่ที่เดิม)

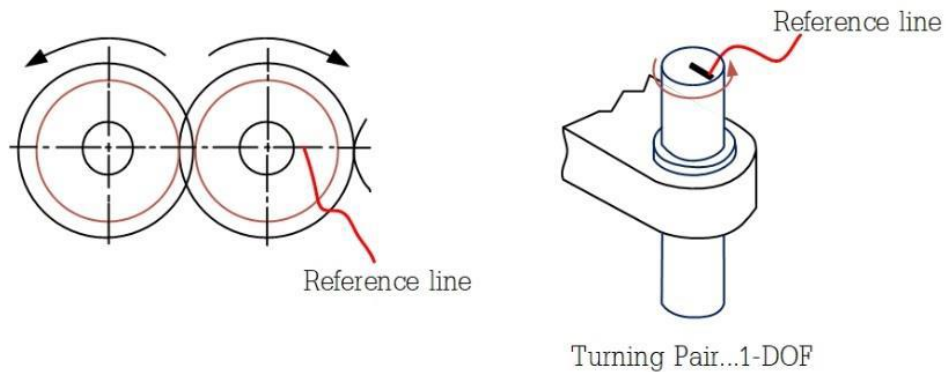
คำจำกัดความแขนหุ่นยนต์

1. โครงสร้างแขนแต่ละท่อน (link)
2. พื้นที่การทำงาน (workspace) คือ พื้นที่การทำงานที่ปลายแขนหุ่นยนต์สามารถเคลื่อนที่ไปได้
3. ข้อต่อ (joint) คือส่วนที่เป็นข้อต่อเชื่อมกับก้านต่อของแขนหุ่นยนต์ สามารถแบ่งตามความสามารถในการเคลื่อนที่เป็น active joint คือ ข้อต่อที่มีต้นกำลังติดตั้งอยู่ สามารถควบคุมการเคลื่อนที่ได้, passive joint คือ ข้อต่อที่ไม่มีต้นกำลังติดตั้งอยู่ ไม่สามารถควบคุมการเคลื่อนที่ได้ การเคลื่อนที่จะเป็นไปตามการเคลื่อนที่ของข้อต่ออื่น ๆ ที่เชื่อมโยงถึงกัน หรือแบ่งตามรูปแบบการเคลื่อนที่เป็น, revolute joint คือ ข้อต่อที่หมุนได้ (1 degree of freedom), prismatic joint คือ ข้อต่อที่เคลื่อนที่เข้า-ออกเป็นแนวเส้น (1 degree of freedom), screw joint คือ ข้อต่อที่หมุนและเคลื่อนที่เข้า-ออกสัมพันธ์กัน (1 degree of freedom), spherical joint คือ ข้อต่อที่หมุนได้รอบ (2 degrees of freedom)
4. base คือ link แรกสุดที่ถูกยึดอยู่กับที่
5. wrist คือ ข้อต่อที่ไม่ทำให้เกิดการเคลื่อนที่ในสามมิติ แต่ทำให้เกิดการหมุน มักไว้เป็นข้อต่อสุดท้ายที่ปลายแขน
6. end-effector คือ อุปกรณ์ที่ติดตั้งไว้ที่ปลายแขนเพื่อใช้ทำงาน อาจจะเป็นมือจับ ตัวดูดสุญญากาศ สว่าน เป็นต้น

2.1.8 ชนิดของการเคลื่อนที่ (Type of Motion)

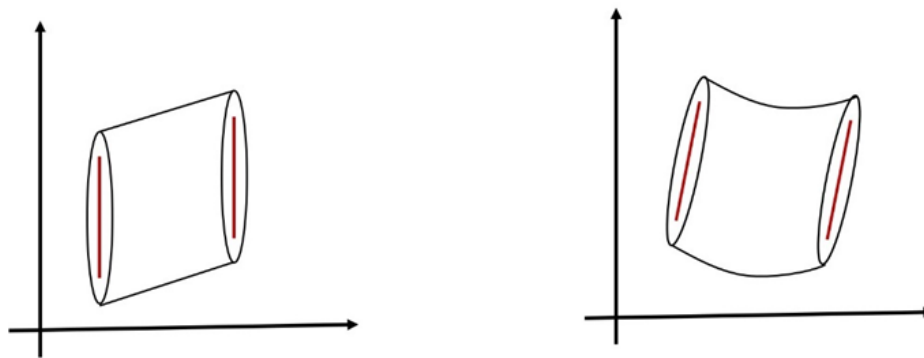
การเคลื่อนที่ของวัตถุอย่างอิสระในปริภูมิสามมิติ โดยทั่วไปจะเป็นการเคลื่อนที่แบบ Complex Motion นั่นคือ วัตถุมีการเคลื่อนที่ทิศทางไปตามแนวแกนและหมุนรอบแกนไปพร้อมกัน สำหรับการเคลื่อนที่ในระนาบจะสามารถเคลื่อนที่ไปตามแนวแกนและหมุนรอบแกนที่ตั้งฉากกับระนาบ การเคลื่อนที่ในระนาบจะแบ่งเป็น ดังต่อไปนี้

1. การเคลื่อนที่แบบหมุน (Rotation Motion) คือการหมุนรอบแกนอ้างอิงใด ๆ ซึ่งหากวัตถุเคลื่อนที่แบบ Pure Rotation หมายถึง ไม่มีการเคลื่อนที่ตามแนวแกน เมื่อเทียบกับ แกนอ้างอิง (Reference Frame) หรือ เส้นอ้างอิง (Reference Line) เปลี่ยนเฉพาะพิกัดเชิงมุม (Angular Orientation) ดังภาพที่ 16



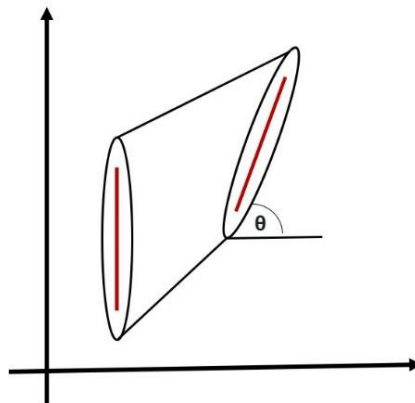
ภาพ 16 การเคลื่อนที่แบบหมุนอย่างเดียว

2. การเคลื่อนที่แบบเลื่อนตำแหน่ง (Translation Motion) หรือ การเคลื่อนที่ไปในทิศทางอ้างอิงใดๆ ซึ่งหากวัตถุเคลื่อนที่แบบ Pure Translation หมายถึง ทุกจุดบนวัตถุ จะมีเส้นทางการเคลื่อนที่ขนานกันไป เมื่อเทียบกับเส้นอ้างอิง(Reference line) เปลี่ยนเฉพาะตำแหน่งเชิงเส้นตามแกนอ้างอิง (Linear Position) ดังแสดงภาพที่ 17



ภาพ 17 การเคลื่อนที่แบบเลื่อนตำแหน่งเดียว

3. การเคลื่อนที่แบบซับซ้อน (Complex Motion) หมายถึง วัตถุเคลื่อนที่ตามแนวแกนและหมุนรอบแกนที่ตั้งฉากกับระนาบ (Rotation Translation) ถ้าใน 2-D Coordination System หรือ Planar Motion กำหนดให้ (x, y) เป็นพิกัดเชิงเส้น (Translation component) และ (θ) เป็นพิกัดเชิงมุม (Rotation Component) เมื่อวัตถุเคลื่อนที่แบบ Complex พิกัดเชิงเส้นและพิกัดเชิงมุมจะมีการเปลี่ยนแปลงเมื่อเทียบกับ เส้นอ้างอิง (Reference Line) บนวัตถุดังภาพที่ 18



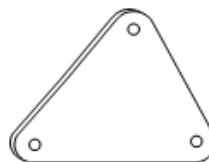
ภาพ 18 การเคลื่อนที่แบบซับซ้อน (Complex Motion)

2.1.9 ระบบข้อต่อโยง (Linkage)

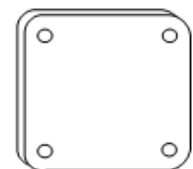
ระบบข้อต่อโยง คือ เป็นกลไกที่ประกอบด้วยชิ้นส่วน (Link) ที่เชื่อมต่อกัน (Joint) เพื่อทำให้เกิดการเคลื่อนที่แบบลูกโซ่หรือแบบต่อเนื่อง แสดงดังภาพที่ 19 ประกอบด้วย ข้อต่อโยง (Link) คือ วัตถุเกร็ง (Rigid Body) ที่เป็นชิ้นส่วนแต่ละชิ้นของกลไก จะมีจุดต่อ (Node) อย่างน้อย 2 จุด ซึ่งเป็นจุดเชื่อมต่อกับ Link อื่น ๆ เพื่อถ่ายเทการเคลื่อนที่และแรงระหว่างกัน ชนิดของ Link ที่แบ่งตามจำนวน Node ของข้อต่อโยง คือ



(ก) Binary Link



(ข) Ternary Link

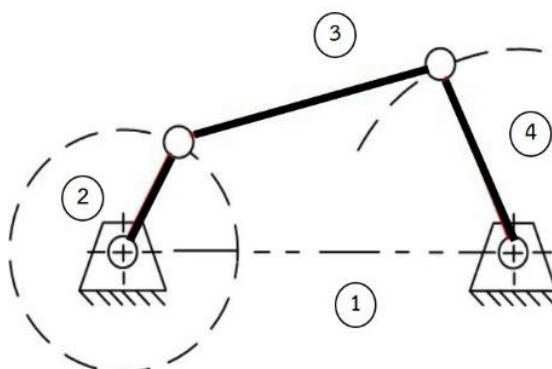


(ค) Quaternary Link

ภาพ 19 ชนิดของข้อต่อโยงแบ่งตามจำนวนของ Node

กลไกเคลื่อนที่ในระนาบ (Planar Mechanism)

สายโซ่จลนศาสตร์ (Kinematics Chain) คือ Links ที่นำมาต่อกันโดย Input Link จะทำการควบคุมการเคลื่อนที่ของ Output และ กลไก (Mechanisms) คือ Kinematics Chain ที่มีอย่างน้อย 1 Link ติดอยู่กับ Ground หรือ Reference Frame ซึ่งนิยามของข้อต่อโยงในชุดกลไก จากภาพที่



ภาพ 20 ชนิดของขั้วต่อโยงในชุดกลไก

จากภาพที่ 20 หมายเลข 1 Ground หมายถึง Link ที่ถูกยึดติดอยู่กับ Reference Frame หมายเลข 2 Crank หรือ ขั้ว เหวียง หมายถึง Link ที่สามารถเคลื่อนที่ ครอบรอบ รอบจุดยึดที่ติดอยู่กับ Ground หมายเลข 3 Coupler หรือ Complex Link หมายถึง Link ที่มี Complex Motion และไม่มีจุดที่ยึดติดกับ Ground และหมายเลข 4 Rocker หมายถึง Link ที่แกว่งไปมา หรือ โยกไปมารอบจุดยึดที่ติดอยู่กับ Ground

2.1.10 โซ่จลนศาสตร์ (Kinematic chain)

จลนศาสตร์ หรือ Kinematics คือ การศึกษาการเคลื่อนที่โดยไม่สนใจแรงกระทำ ความสัมพันธ์ระหว่างระยะกระจัด (Displacement, S), ความเร็ว (Velocity, v) และความเร่ง (Acceleration, a) โดยที่ จนศาสตร์ (Kinetics) คือ การศึกษาแรงของระบบที่มีการเคลื่อนที่ ความสัมพันธ์ระหว่างแรงกระทำกับวัตถุ, มวลของวัตถุ และการเคลื่อนที่ของวัตถุ ในการวิเคราะห์ กลไก จะเป็นการวิเคราะห์ Kinetics และวิเคราะห์พลศาสตร์การเคลื่อนที่

1. จลนศาสตร์แบบไปข้างหน้า (Forward Kinematics)

การคำนวณหาตำแหน่งที่ปลายของแขนกล (End Effectors) โดยคำนวณจากสมการ การเคลื่อนที่ของหุ่นยนต์คำตอบที่ออกมาอยู่ในรูปของจุด (X, Y, Z) หรือระบบคาร์ทีเซียน โดย ต้องทราบ ตัวแปรที่ข้อต่อของแต่ละข้อต่อ (Joint Variables) กรณีข้อต่อแบบหมุนตัวแปรที่ต้อง ทราบค่าคือ มุม ซึ่งค่าที่ทราบในแต่ละข้อต่อนำไปคำนวณหาค่าพิกัดที่ปลายของแขนหุ่นยนต์

2. จลนศาสตร์แบบผกผัน (Inverse Kinematics)

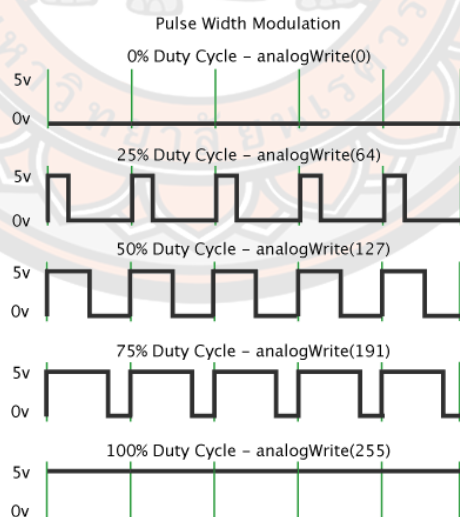
การคำนวณค่าตัวแปรของแต่ละข้อต่อโดยกำหนดตำแหน่งที่ปลายของแขนกลวิธีการ คำนวณจลนศาสตร์แบบผกผันนี้ ในบางครั้ง มีได้หลายคำตอบหรือไม่สามารถหาคำตอบได้ การ คำนวณนี้ค่อนข้างยุ่งยากกว่าการคำนวณจลนศาสตร์แบบไปข้างหน้า ซึ่งคำตอบของวิธีการ คำนวณ จลนศาสตร์ผกผันนี้อาจเป็นได้ 2 รูปแบบคือ แบบปิด (Closed Form) แบบคำนวณซ้ำ เชิงตัวเลข (Numerical Iterative Form) โดยวิธีการแบบปิด (Closed Form) นั้นสามารถหา คำตอบให้อยู่ใน รูป ของฟังก์ชันซึ่งง่ายต่อการคำนวณหาค่าเพราะสามารถแทนค่าในตัวแปรของฟังก์ชัน ส่วนวิธีการ

แบบคำนวณซ้ำเชิงตัวเลข (Numerical Iterative From) นั้นใช้วิธีสมมุติค่า ตัวเลขเริ่มต้นแล้วทำซ้ำไป จนกว่าค่านั้นไม่มีการเปลี่ยนแปลง

2.1.11 Pulse width modulation (PWM)

Pulse width modulation (PWM) คือ สัญญาณพัลส์ที่มีค่าความถี่คงที่แต่ความกว้างของพัลส์เปลี่ยนแปลงได้ ซึ่งการควบคุมพลังงานที่ส่งออกไปยังอุปกรณ์ที่ต้องการควบคุม เช่น ความเร็วมอเตอร์ ความสว่างของหลอดไฟ LED โดยปกติหมายถึงการลดแรงดันที่ส่งออกไปยังมอเตอร์ แต่การลดแรงดันนั้นเป็นแนวทางที่ต้องใช้วงจรที่ซับซ้อนมีความยุ่งยากค่อนข้างมาก ดังนั้นโดยทั่วไปจึงนิยมใช้เทคนิคที่เรียกว่า Pulse Width Modulation (PWM) ซึ่งไม่ได้ลดแรงดัน หากแต่ใช้หลักการเปิด/ปิดมอเตอร์ด้วยความเร็วสูง ๆ จนผลค่าเฉลี่ยของแรงดันที่ได้ออกมาเทียบเท่ากับการเปลี่ยนแรงดันโดยตรง เทคนิคนี้ทำให้ไม่ต้องใช้วงจรซับซ้อน แต่เป็นเรื่องเทคนิคการเขียนโปรแกรม

PWM นั้นโดยทั่วไปจะมีการสร้างลูกคลื่นสี่เหลี่ยม (Square Wave) โดยสัญญาณพัลส์ เมื่อเทียบ % ของช่วงเวลาที่ เป็น HIGH (หรือก็คือ % ของ Pulse Width) กับ % ของคาบเวลา (Period) ของพัลส์ลูกนั้นๆ โดยทั่วไปจะเรียกว่า Duty Cycle สำหรับโมดูล PWM ของ Arduino มีความละเอียด 8-bit หรือ ปรับได้ 255 ระดับ ดังนั้นค่าสัญญาณ 0 โวลต์ถึง 5 โวลต์ จะถูกแสดงเป็นสัญญาณแบบดิจิตอล จะได้ 0 ถึง 255 ซึ่งสามารถเทียบสัดส่วนคำนวณจากเลขจริง เป็น เลขทางดิจิตอลได้ แสดงดังภาพที่ 21



ภาพ 21 การกำหนดความกว้างของสัญญาณพัลส์

ที่มา: (thetempedia, 2017)

2.1.12 วัสดุอุปกรณ์ และซอฟต์แวร์ที่ใช้ในงานวิจัย

มอเตอร์กระแสตรง และเซอร์โวมอเตอร์ (Direct current motor and servo motor)

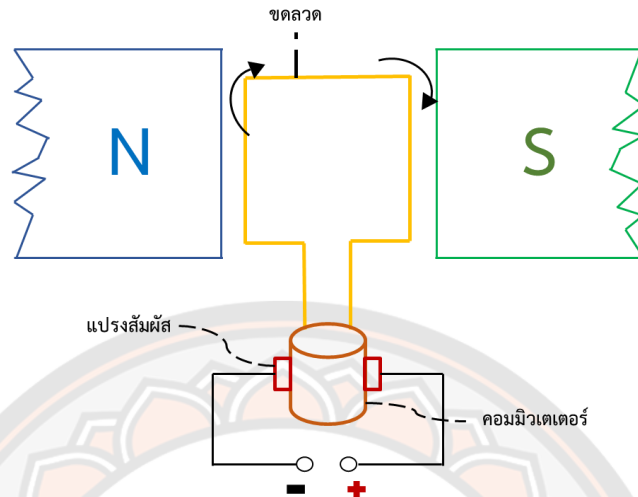
งานวิจัยนี้ได้ใช้มอเตอร์สองชนิดคือ มอเตอร์กระแสตรงซึ่งทำหน้าที่เป็นส่วนขับเคลื่อนการเคลื่อนที่ของหุ่นยนต์ และเซอร์โวมอเตอร์ทำหน้าที่ในการขับเคลื่อนแขนหุ่นยนต์ ซึ่งมอเตอร์ไฟฟ้าหมายถึง อุปกรณ์เครื่องกลทางไฟฟ้าชนิดหนึ่ง ทำหน้าที่เปลี่ยนแปลงพลังงานไฟฟ้ามาเป็นพลังงานกล มอเตอร์ไฟฟ้าที่ใช้พลังงานไฟฟ้าเปลี่ยนเป็นพลังงานกลมีทั้งไฟฟ้า กระแสสลับและพลังงานไฟฟ้ากระแสตรง

มอเตอร์กระแสตรง (Direct current motor)

มอเตอร์ไฟฟ้ากระแสตรงเป็นมอเตอร์ที่ป้อนด้วยไฟฟ้ากระแสตรง เป็นเครื่องกลไฟฟ้ากระแสตรงที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล มีคอมมิวเตเตอร์กับแปรงถ่านทำหน้าที่ในการจัดเรียงกระแสไฟฟ้าในขดลวดอาร์มาเจอร์ กระแสที่ไหลในขดลวดอาร์มาเจอร์ภายใต้หน้าขั้วแม่เหล็กเดียวกัน จะมีทิศทางเดียวกัน ส่วนที่อยู่ภายใต้ขั้วตรงข้ามจะมีทิศทางการไหลของกระแสไฟฟ้าตรงข้ามกัน ส่งผลให้แรงบิดเกิดขึ้นเสริมกัน มอเตอร์ไฟฟ้ากระแสตรง มีทั้งชนิดกระตุ้นฟิลด์จากภายนอก (Separated excited motor) และชนิดกระตุ้นฟิลด์ด้วยตัวเอง (Self excited motor) หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง เมื่อแรงดันไฟฟ้ากระแสตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะ แปรงถ่านผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้น จนเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกันตามคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดกัน ทิศทางตรงข้ามจะหักล้างกันและทิศทางเดียวจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ อาร์มาเจอร์ที่หมุนนี้เรียกว่า โรเตอร์ (Rotor)

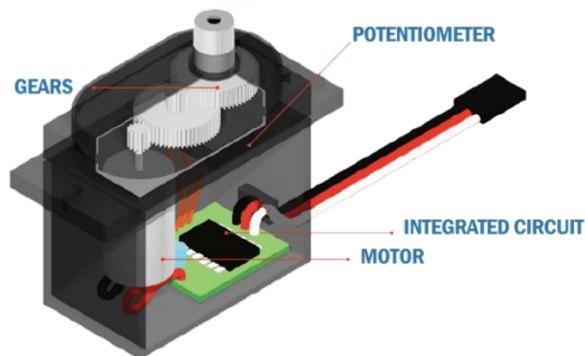
การทำงานของมอเตอร์ไฟตรง การขับหรือทำให้มอเตอร์ไฟตรงทำงานเพื่อหมุนแกนนั้นง่ายมาก เพียงจ่ายไฟเข้าที่ขั้วของมอเตอร์เท่านั้น และเมื่อกลับขั้วของการจ่ายไฟมอเตอร์ก็จะหมุนกลับทิศทาง สำหรับการอธิบายการทำงานของมอเตอร์โดยทั่วไปจะอ้างถึงมอเตอร์แบบ 2 ขั้ว ดังในภาพที่ 22 เมื่อจ่ายไฟให้แก่มอเตอร์ผ่านทางแปรงสัมผัสซึ่งต่ออยู่กับคอมมิวเตเตอร์และขดลวด เมื่อกระแสไฟฟ้าไหลผ่านขดลวดทำให้เกิดสนามแม่เหล็กขึ้น และเกิดแรงดูดจากแม่เหล็กถาวร ทำให้ขดลวดสามารถหมุนได้ แต่ด้วยการใช้ขดลวดเพียง 2 ขั้ว การหมุนของมอเตอร์จะขาดเสถียรภาพ เพราะในความเป็นจริงเมื่อคอมมิวเตเตอร์หมุนไป 90 องศาจะทำให้เกิดการลัดวงจรคอมมิวเตเตอร์ทั้ง 2 ขั้ว ทำให้กระแสไฟฟ้าหยุดไหล แต่แกนของมอเตอร์ยังหมุนไปได้ด้วยแรงเฉื่อย ทำให้จังหวะการทำงานนั้นไม่ต่อเนื่อง และทำให้อัตราเร็วในการหมุนไม่คงที่ ซึ่งทางแก้ไขนั้นจะใช้มอเตอร์แบบมี

ขดลวด 3 ขั้ว ที่มีการพันในทิศทางที่สลับกัน ส่วนประกอบและการทำงานของมอเตอร์ไฟตรงแสดงดังภาพที่ 22



ภาพ 22 แสดงส่วนประกอบและการทำงานของมอเตอร์ไฟตรง เซอร์โวมอเตอร์ (Servo motor)

Servo Motor เป็นการรวมมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) เข้ากับวงจรควบคุม ใช้สำหรับงานที่ต้องการควบคุมตำแหน่งความเร็ว แรงบิด ความแม่นยำ และความเร็ว เพื่อให้ระบบทางกลและไฟฟ้าคอนโทรลทำงานสอดคล้องกันอย่างมีประสิทธิภาพ โดยความแตกต่างที่สำคัญของเซอร์โวมอเตอร์กับมอเตอร์แบบอื่น ๆ คือเซอร์โวมอเตอร์จะรู้ตำแหน่งที่ตัวเองอยู่ และสิ่งเปลี่ยนตำแหน่งโดยการเปลี่ยนองศาได้ โดยใช้การควบคุมแบบป้อนกลับ (Feedback Controller) ซึ่งเป็นระบบควบคุมที่มีการวัดค่าเอาต์พุตของระบบ แล้วนำมาเปรียบเทียบกับค่าอินพุตเพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่า เท่ากับหรือใกล้เคียงกับค่าอินพุต นิยมใช้งานในเครื่องบินบังคับ เรือบังคับ โดยใช้กำหนดทิศทางของหางเสือเป็นองศา



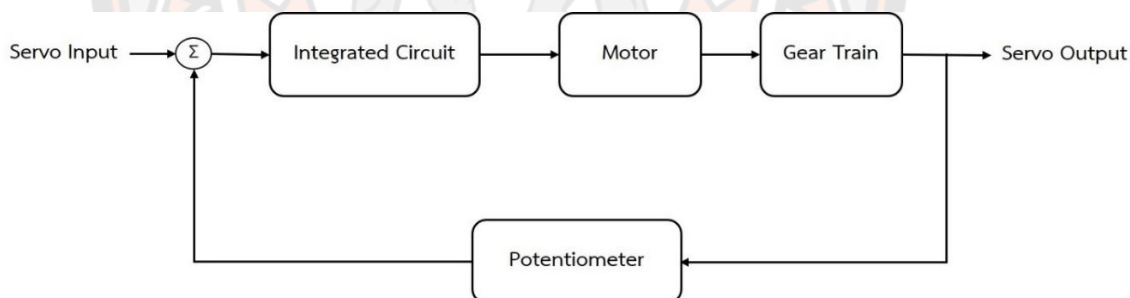
ภาพ 23 ส่วนประกอบของเซอร์โวมอเตอร์

ที่มา: (Arduino makerzone, 2020)

ส่วนประกอบของเซอร์โวมอเตอร์จากภาพที่ 23 ประกอบไปด้วย มอเตอร์กระแสตรง (DC Motor) ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานทางกล, เกียร์ (Gears) ทำหน้าที่ลดรอบความเร็วหรือทดรอบแรงบิด, โปเทนซีโอมิเตอร์ (potentiometer) เป็นทรานสดิวเซอร์วัดตำแหน่งและระยะทาง ชนิดเปลี่ยนแปลงค่าความต้านทาน และวงจรรวม (Integrated Circuit : IC) เป็นอุปกรณ์ที่นำเอา อุปกรณ์อิเล็กทรอนิกส์ชนิดต่าง ๆ ได้แก่ ตัวต้านทาน ตัวเก็บประจุ ไดโอด ทรานซิสเตอร์ ส่วนประกอบต่าง ๆ ของวงจรมาต่อรวมกัน เพื่อใช้ในการควบคุม และประมวลผล

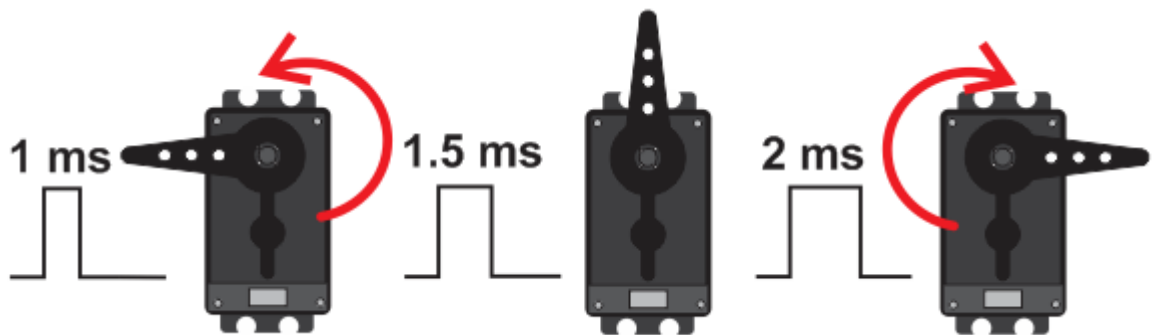
หลักการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ทำงานโดยการรับสัญญาณพัลส์เข้าวงจรควบคุม (Electronic Control System) แล้วจะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่ง องศาที่ต้องการให้ มอเตอร์หมุนเคลื่อนที่ไปยังตำแหน่งนั้น ๆ โดยมี potentiometer เป็นตัวเซ็นเซอร์ที่วัดค่ามุมของมอเตอร์ แล้วส่งค่า Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้อง และแม่นยำ แผนภาพการทำงานของเซอร์โวมอเตอร์แสดงดังภาพที่ 24



ภาพ 24 บล็อกไดอะแกรมของเซอร์โวมอเตอร์

ตัวอย่างการสั่งงานเซอร์โวมอเตอร์ด้วยการสัญญาณพัลส์ เช่น กำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms เซอร์โวมอเตอร์จะหมุนไปทางด้านซ้ายจนสุดทาง ในทางกลับกันถ้ากำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms เซอร์โวมอเตอร์จะหมุนไปทางด้านขวาจนสุดทาง แต่ถ้ากำหนดที่ 1.5 ms เซอร์โวมอเตอร์จะหมุนมาอยู่ที่ตำแหน่งตรงกลาง แสดงดังภาพที่ 25

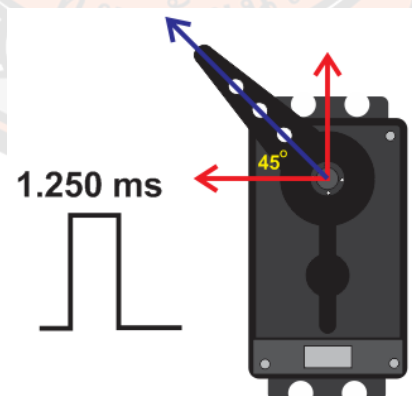


ภาพ 25 ความกว้างสัญญาณพัลส์ เพื่อกำหนดมุมองศาเซอร์โวมอเตอร์

ที่มา: (thaieasyelec, 2017)

ดังนั้นสามารถกำหนดองศาการหมุนของเซอร์โวมอเตอร์ได้โดยการเทียบค่า เช่น เซอร์โวมอเตอร์สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ความกว้างพัลส์เท่ากับ 1000 us ที่ 180 องศาความกว้างพัลส์เท่ากับ 2000 us เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ความกว้างพัลส์ต่างกัน $(2000-1000)/180$ เท่ากับ 5.55 us

จากการหาค่าความกว้างพัลส์ที่มุม 1 องศาข้างต้น หากต้องการกำหนดให้เซอร์โวมอเตอร์หมุนไปที่มุม 45 องศาจะหาค่าพัลส์ที่ต้องการได้จาก 5.55×45 เท่ากับ 249.75 us แต่ที่มุม 0 องศาเริ่มที่ความกว้างพัลส์ 1ms หรือ 1000 us เพราะฉะนั้นความกว้างพัลส์ที่ใช้กำหนดให้เซอร์โวมอเตอร์หมุนไปที่ 45 องศา คือ $1000 + 249.75$ เท่ากับประมาณ 1250 us แสดงดังภาพที่ 26



ภาพ 26 กำหนดมุมองศาเซอร์โวมอเตอร์ที่ 45 องศา

ที่มา: (thaieasyelec, 2017)

บอร์ดไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ควบคุมที่มีขนาดเล็ก โดยภายในมีการรวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งมีการทำงานความคล้ายคลึงกับคอมพิวเตอร์ และส่วนประกอบหลักๆ หลักการทำงานของไมโครคอนโทรลเลอร์คือ การควบคุมระบบหรืออุปกรณ์ต่างๆ เช่น sensor motor เป็นต้น สามารถนำมาประยุกต์ใช้งานได้อย่างหลากหลาย โดยผ่านการออกแบบวงจรให้เหมาะกับลักษณะงานนั้นๆ และสามารถรับโปรแกรมคำสั่งภาษาซี เพื่อควบคุม INPUT/OUTPUT ในการสั่งงานไปควบคุมระบบหรืออุปกรณ์นั้นๆ สามารถใช้ทำงานได้ทั้งระบบระบบอนาล็อก และดิจิทัล

คอนโทรลเลอร์ Arduino

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีขนาดเล็กเป็นตัวประมวลผลและสั่งงาน ซึ่งบอร์ด Arduino มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์ และซอฟต์แวร์ เหมาะสำหรับการเรียนรู้เกี่ยวกับระบบควบคุม และสามารถนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุม input/output กับการรับข้อมูลหรือส่งออกข้อมูลให้กับอุปกรณ์อื่น ๆ เช่น เซนเซอร์, มอเตอร์, รีเลย์, หลอดไฟ, โมดูลสื่อสาร เป็นต้น ทั้งนี้การควบคุมแบบ input/output สามารถใช้ได้ในรูปแบบของอนาล็อกและดิจิทัล โดย Arduino มีจุดเด่นในเรื่องของความง่ายในการพัฒนาโปรแกรม เหมาะสำหรับการเรียนรู้ เนื่องจากปัจจุบันความนิยมในการใช้งานเพิ่มมากขึ้น ทำให้มีเอกสารที่เกี่ยวข้อง ตัวอย่างการใช้งาน คลิปวิดีโอ Library และ Source code ต่าง ๆ มากมายให้ศึกษาเรียนรู้ นำไปพัฒนาต่อยอดต่อยอดได้ ซึ่งบอร์ด Arduino ประกอบไปด้วยส่วนของ Hardware และ Software ในส่วนระบบฮาร์ดแวร์ ผู้พัฒนาสามารถสร้าง และประกอบชิ้นมาขึ้นอยู่กับระบบควบคุม ว่ามีอุปกรณ์อะไรบ้าง ซึ่งส่วนนี้จะมีการใช้ความรู้ด้านอิเล็กทรอนิกส์ และส่วนซอฟต์แวร์ การเขียนโปรแกรมลงบนบอร์ด Arduino เองนั้น ผู้พัฒนาสามารถดาวน์โหลดโปรแกรม Arduino ide มาใช้งานได้ฟรี ซึ่งภาษาสำหรับเขียนโปรแกรมควบคุมมีไวยากรณ์แบบเดียวกับภาษา C/C++ ตัวโปรแกรม Arduino ide เป็นเครื่องมือสำหรับเขียนโค้ดโปรแกรม การคอมไพล์โปรแกรม (การแปลงไฟล์ภาษาซีให้เป็นภาษาเครื่อง) และอัปโหลดโปรแกรมลงบอร์ด

ประเภทของบอร์ด Arduino มีมากมายหลายรุ่น การเลือกแต่ละรุ่นขึ้นอยู่กับความเหมาะสมของงานที่จะพัฒนา ซึ่งในงานวิจัยนี้ได้เลือกบอร์ด Arduino มาสองรุ่นคือบอร์ด Arduino Mega 2560 R3 เพื่อนำมาพัฒนาในระบบควบคุมการเคลื่อนที่ของหุ่นยนต์เนื่องจากส่วนนี้ประกอบไปด้วยการควบคุมควบหลายส่วน ทำให้ใช้พอร์ตมาก ทั้งรับข้อมูลจากเซนเซอร์ต่าง ๆ การส่งสัญญาณ PWM ในการควบคุมมอเตอร์ การใช้โมดูลสื่อสารสำหรับติดต่อกับโทรศัพท์มือถือ และบอร์ด Arduino Uno

R3 และใช้สำหรับการเคลื่อนที่ของแขนหุ่นยนต์ ซึ่งใช้บอร์ดนี้ร่วมกับบอร์ด Shields สำหรับควบคุม เซอร์โวมอเตอร์ ที่ออกแบบให้ใช้ร่วมงานกับบอร์ด Arduino Uno R3

บอร์ด Arduino Uno R3

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ได้รับความนิยมอีกบอร์ดหนึ่ง เนื่องจากมี ราคา ไม่แพง ซึ่งส่วนใหญ่โปรเจกต์และไลบรารีต่าง ๆ ที่พัฒนาขึ้นมาถูกอ้างอิงกับบอร์ดนี้เป็นหลัก เพราะเป็น ขนาดที่เหมาะสมกับการเริ่มต้นการเรียนรู้ เนื่องจากเป็นขนาดที่เหมาะสมสำหรับการเริ่มต้น เรียนรู้ Arduino และมี Shields ให้เลือกใช้งานได้มากกว่าบอร์ด Arduino รุ่นอื่นๆที่ออกแบบมา เฉพาะมากกว่า โดยบอร์ด Arduino Uno ได้มีการพัฒนาเรื่อยมา ตั้งแต่ R2 R3 และรุ่นย่อยที่เปลี่ยน ชิปไอซีเป็นแบบ SMD และข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ ง่าย บอร์ด Arduino Uno R3 แสดงดังภาพที่ 27

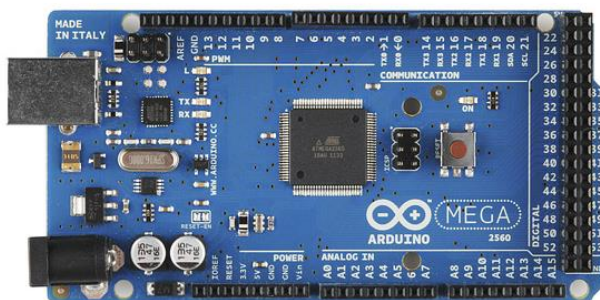


ภาพ 27 บอร์ด Arduino Uno R3

ที่มา: (Robotsiam, 2016)

บอร์ด Arduino Mega 2560 R3

บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3 เป็นบอร์ดที่ออกมาแบบมาสำหรับ งานที่ต้องใช้พอร์ตรับส่งสัญญาณที่มากกว่าบอร์ด Arduino Uno R3 เช่น งานที่ต้อง ควบคุมมอเตอร์ Servo หลายๆ ตัว หรืองานที่ต้องการรับสัญญาณจาก Sensor ทำให้พอร์ต input/output ของ บอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำ แบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ใน ความเร็วของ MCU ที่เท่ากัน บอร์ด Arduino Mega 2560 R3 แสดงดังภาพที่ 28



ภาพ 28 บอร์ด Arduino Mega 2560 R3

ที่มา: (Arduino-indy, 2018)

บอร์ด Raspberry pi

บอร์ด Raspberry pi เป็นบอร์ดคอมพิวเตอร์ขนาดเล็ก มีความสามารถคล้ายกับคอมพิวเตอร์ เช่น ใช้เป็นคอมพิวเตอร์ส่วนบุคคล เพื่อดูหนัง ฟังเพลง เล่นอินเทอร์เน็ต เป็นต้น สามารถเชื่อมต่อกับ จอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ อีกทั้งนอกเหนือจากที่คอมพิวเตอร์ทั่ว ๆ ไปทำได้ก็คงเป็นเรื่องของการสื่อสาร และควบคุมอุปกรณ์ไฟฟ้า อิเล็กทรอนิกส์ต่างๆ เช่น ใช้ทำระบบตรวจจับใบหน้า, ใช้ควบคุมการทำงานของหุ่นยนต์ หรือเครื่องจักร, ระบบเปิด/ปิดหลอดไฟ หรือเครื่องใช้ไฟฟ้าแบบอัตโนมัติ, ใช้ทำระบบกล้องวงจรปิด, ใช้ควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน ด้วยคำสั่งเสียง เป็นต้น ซึ่งในงานวิจัยนี้ได้ใช้บอร์ด Raspberry pi 3 ในส่วนของการแสดงภาพจากกล้องที่ติดกับหุ่นยนต์ ขึ้นสัญญาณภาพบนหน้าในแอปพลิเคชันแบบเรียลไทม์ ตัวอย่างบอร์ด Raspberry pi 3 แสดงดังภาพที่

29



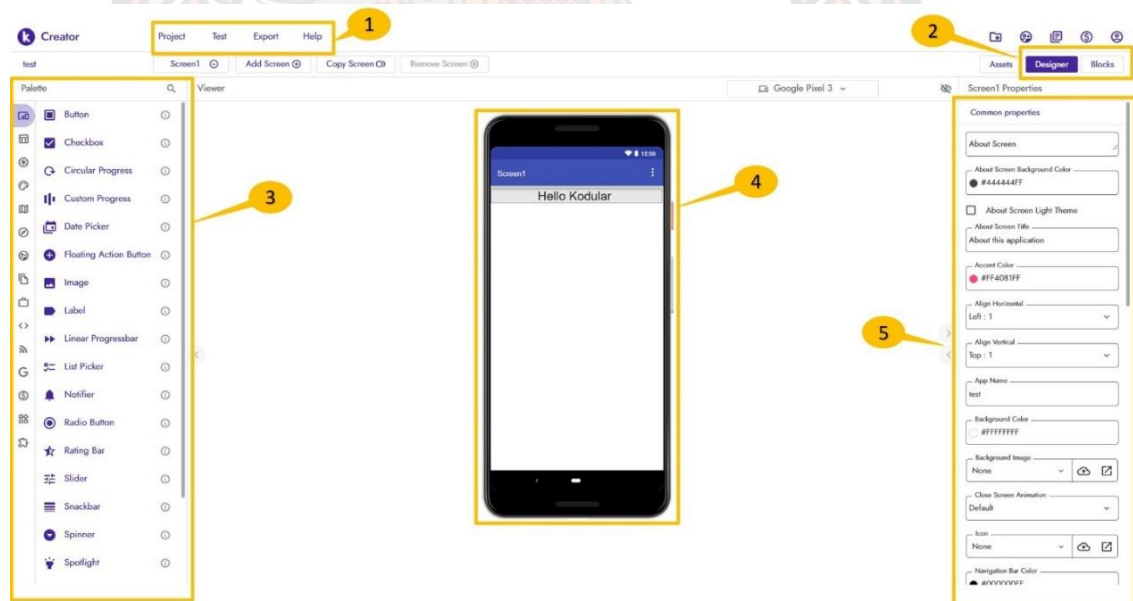
ภาพ 29 บอร์ด Raspberry pi 3

ที่มา : (RS Components, 2017)

เครื่องมือสำหรับเขียนแอปพลิเคชัน Kodular

Kodular เป็นเครื่องมือในการเขียนแอปพลิเคชัน ที่ไม่จำเป็นต้องติดตั้งโปรแกรมลงบนคอมพิวเตอร์ สามารถเขียนบนออนไลน์ หรือเว็บไซต์ ซึ่ง Kodular เปิดให้คนทั่วไปสามารถสร้างแอปพลิเคชันเป็นของตัวเองได้ง่ายๆ โดยไม่ต้องเรียนรู้การเขียนโปรแกรม เพียงแค่ลาก-วางบล็อก มาต่อกันแบบจิ๊กซอว์ สามารถออกแบบและสร้างแอปพลิเคชัน รองรับบนระบบปฏิบัติการ Android ข้อดีของ Kodular คือ ผู้พัฒนาสามารถสร้างแอปพลิเคชันได้ฟรี, ปัจจุบัน Kodular เริ่มเป็นที่นิยมในกลุ่มผู้สร้างแอปทำให้ถ้าเกิดติดปัญหาบางอย่างที่ไม่เข้าใจสามารถเข้าไปสอบถามใน Kodular Community ได้ตลอดเวลา, การใช้งานต่าง ๆ ที่ง่ายของ kodular ทั้งสามารถพัฒนาแอปพลิเคชันบนออนไลน์ การเขียนโปรแกรมแค่ลาก-วางบล็อก และยังเหมาะสำหรับเรียนรู้สำหรับกลุ่มคนที่อยากเริ่มศึกษาการสร้างแอปพลิเคชันเบื้องต้น

ส่วนประกอบของ Kodular แบ่งเป็นสองส่วนหลักคือ 1. ส่วนการออกแบบ UI ของแอปพลิเคชัน 2. การเขียนโปรแกรมสำหรับการใช้งานแอปพลิเคชัน โดยส่วนประกอบของ Kodular แสดงดังภาพที่ 30



ภาพ 30 ส่วนประกอบของ Kodular

จากภาพที่ 30 แสดงส่วนประกอบของ Kodular จะแบ่งเป็น 5 ส่วนหลัก ๆ ดังนี้

ส่วนที่ 1 คือ Menu เป็นส่วนคำสั่งสำหรับเครื่องมือสร้างแอปพลิเคชัน มีรายละเอียดดังนี้
 Project เป็นเมนูสำหรับจัดการ เช่น สร้าง Project ใหม่ หรือ ลบ Project เป็นต้น, Test เป็นส่วนจำลองแอปพลิเคชันที่ผู้พัฒนาสร้างขึ้นมา โดยจำลองได้ผ่านมือถือ, Export สามารถนำ Project เรา

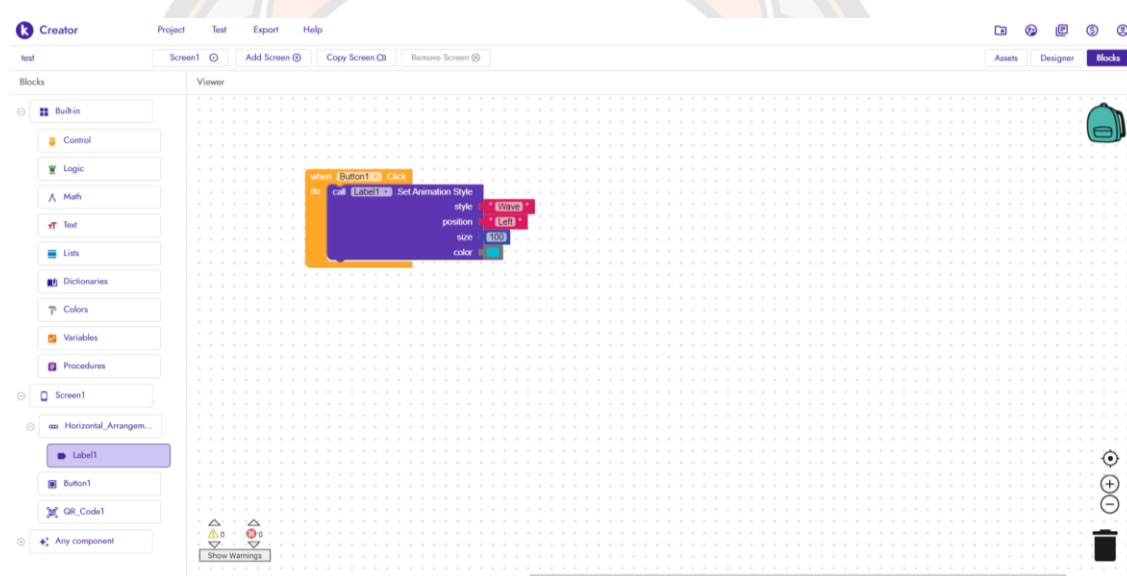
แปลงเป็นไฟล์ APK ไฟล์ที่สามารถติดตั้งบนมือถือ และส่วน Help เป็นส่วนช่วยเหลือเมื่อมีปัญหาสามารถแจ้งได้

ส่วนที่ 2 เป็นส่วนเลือก Designer สำหรับสร้าง UI และส่วน Blocks ใช้ออกแบบการทำงานของแอปพลิเคชันลาก - วาง บล็อกการทำงาน ที่เปรียบเสมือนการเขียน Code

ส่วนที่ 3 Component เป็นส่วนที่ใช้ออกแบบหน้าจอแอปพลิเคชันที่จะสร้าง มีส่วนประกอบต่าง ๆ เช่น ข้อความ รูป ปุ่ม เครื่องเล่นเพลง เว็บ เป็นต้น

ส่วนที่ 4 Viewer เป็น Screen แสดงผลการออกแบบหน้าจอแอปพลิเคชัน ใช้สำหรับลากวางส่วนประกอบต่าง ๆ เพื่อแสดงผลการออกแบบหน้าจอแอปพลิเคชัน

ส่วนที่ 5 Property ใช้กำหนดคุณสมบัติของแอปพลิเคชัน และกำหนดคุณสมบัติของส่วนประกอบ ต่าง ๆ



ภาพ 31 หน้าต่างส่วน Blocks สำหรับเขียนโปรแกรม

จากภาพที่ 31 เป็นหน้าต่างส่วน Blocks ใช้ในการเขียนโปรแกรม ลาก - วาง บล็อกมาต่อกันแบบจิ๊กซอว์ การทำงานคล้ายกับการเขียน Code เพื่อให้แอปพลิเคชันทำงานตามต้องการ โดยการทำงานแต่ละบล็อกสามารถศึกษาได้ทั่วไป อีกทั้งผู้พัฒนา Kodular มีเอกสารและวิดีโอ ที่เป็นเนื้อหาภาษาอังกฤษให้ผู้สนใจสามารถศึกษาได้

การติดต่อสื่อสารแบบไร้สาย

งานวิจัยนี้ได้ใช้เทคโนโลยีไร้สายมาควบคุมหุ่นยนต์ โดยใช้สัญญาณบลูทูธ (Bluetooth) สำหรับควบคุมการเคลื่อนที่ของหุ่นยนต์ในส่วนของล้อ และการเคลื่อนที่ของแขนหุ่นยนต์ และสัญญาณ Wi-Fi สำหรับรับส่งสัญญาณภาพแบบเรียลไทม์

การสื่อสารไร้สาย (Wireless communication) เป็นการติดต่อระหว่างจุดสองจุดหรือมากกว่าสองจุด ซึ่งเป็นการโอนถ่ายข้อมูลสารสนเทศที่ไม่ได้เชื่อมต่อกันด้วยตัวนำไฟฟ้า ในการถ่ายโอนข้อมูลโดยไม่ต้องใช้สาย ข้อมูลจะถูกถ่ายโอนในลักษณะนี้ทั้งในระยะทางสั้นและระยะทางไกล ในปัจจุบันเทคโนโลยีเครือข่ายไร้สายเป็นที่นิยมเป็นอย่างมาก สืบเนื่องจากที่ตามบ้านเรือนแหล่งที่พักอาศัยของบุคคลทั่วไป มักมีการติดตั้งสัญญาณเครือข่ายไร้สายมากมาย ทั้งนี้คงเป็นเพราะเหตุผลของเทคโนโลยีทางด้านเครือข่ายไร้สายที่เจริญเติบโตขึ้นอย่างรวดเร็วและงบประมาณที่ใช้ในการจัดตั้งจุดเชื่อมต่อเครือข่ายไร้สายมีราคาถูกลงมาก รวมถึงอุปกรณ์ที่หาซื้อได้ง่ายและสามารถเรียนรู้วิธีการเพื่อจัดตั้งจุดเชื่อมต่อเครือข่ายไร้สายได้ไม่ยากเย็น

Bluetooth คือ เทคโนโลยีที่ใช้ในการเชื่อมต่อของอุปกรณ์ อิเล็กทรอนิกส์ สำหรับใช้รับส่งข้อมูลระหว่างอุปกรณ์ไร้สาย เป็นการเชื่อมต่ออุปกรณ์สองตัวเข้าหากันด้วยในรูปแบบคลื่นวิทยุความถี่ช่วง 2.4 GHz ซึ่งมีระยะการทำงานที่ไกลประมาณ 10 เมตรเป็นอย่างมาก ซึ่งจุดประสงค์ในการสร้าง Bluetooth ขึ้นมาก็เพื่อมาแทนที่สายไฟ ในการเชื่อมโยงอุปกรณ์สองตัวเข้าหากัน เช่น โทรศัพท์มือถือกับอุปกรณ์ต่าง ๆ ไม่ว่าจะเป็นลำโพง หูฟัง เป็นต้น โดยข้อจำกัดของ Bluetooth จะมีระยะสัญญาณที่สั้น อีกทั้งความเร็วในการรับส่งข้อมูลยังมีประสิทธิภาพต่ำกว่าสัญญาณ Wi-Fi แต่หากนำมาใช้ในการส่งข้อมูลที่ปริมาณไม่มาก เหมาะสำหรับการเชื่อมต่ออุปกรณ์ Device เล็กเข้ากัน ด้วยระยะทางเพียงสั้นๆ

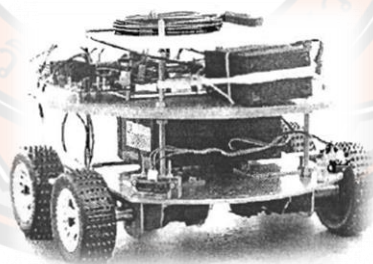
wireless fidelity (Wi-Fi) อุปกรณ์ระบบเครือข่ายไร้สายใดๆ ที่สามารถติดต่อสื่อสารกับอุปกรณ์อื่นๆ เป็นการสื่อสารด้วยเทคโนโลยี 802.11 ซึ่งเป็นเทคโนโลยีที่ได้รับความนิยมที่ช่วยให้อุปกรณ์อิเล็กทรอนิกส์ในการแลกเปลี่ยนข้อมูลหรือการเชื่อมต่ออินเทอร์เน็ตแบบไร้สายโดยใช้คลื่นวิทยุ ซึ่งข้อดีของ wi-fi สามารถส่งข้อมูลไร้สายได้รวดเร็ว ดังนั้นมันจึงเหมาะที่จะนำมาสร้างเครือข่ายไร้สายสำหรับการเชื่อมคอมพิวเตอร์เข้าหากัน หรืออาจจะใช้สมาร์ตโฟนมาเชื่อมกับคอมพิวเตอร์ก็ยิ่งได้ เพราะในปัจจุบันเทคโนโลยีของสมาร์ตโฟนก็สามารถรองรับ Wi-Fi ได้อยู่แล้ว ข้อดีอีกอย่างของ Wi-Fi คือสามารถติดต่อสื่อสารทำงานได้ในระยะที่ไกล ซึ่งมีความคล้ายคลึงกับระบบ Network แบบมีสายมากที่สุด

2.2งานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้เป็นการออกแบบและพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง จุดประสงค์การใช้งานเพื่อให้หุ่นยนต์สามารถเคลื่อนที่ได้ และมีแขนหุ่นยนต์ที่ช่วยในการหยิบจับสิ่งของสำหรับผู้ป่วยหรือคนที่เคลื่อนย้ายลำบาก ดังนั้นการศึกษางานวิจัยที่เกี่ยวข้องจะสามารถแบ่งออกเป็น 2 ส่วนคือ 1. หุ่นยนต์เคลื่อนที่ 2. ชุดแขนหุ่นยนต์ เพื่อศึกษาจุดประสงค์การนำไปใช้งาน การนำเอาหลักการทฤษฎีต่าง ๆ นำมาประยุกต์ใช้งาน หรือการนำเอาเทคโนโลยีใหม่ๆเข้ามาประยุกต์ใช้ให้เหมาะสมกับงานวิจัย

2.2.1 งานวิจัยเกี่ยวกับหุ่นยนต์เคลื่อนที่

โดยหุ่นยนต์เคลื่อนที่สามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง สามารถเคลื่อนย้ายตัวเองจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งได้อย่างอิสระ โดยการใช้ล้อหรือการใช้ขา โดยหุ่นยนต์ประเภทนี้มีอยู่อย่างหลากหลาย เริ่มต้นด้วยงานวิจัย หุ่นยนต์เคลื่อนที่โดยอาศัยข้อมูลจากการตรวจวัดเพื่อหลบหลีกสิ่งกีดขวางแบบอัตโนมัติ ของ อนันต์ สืบสำราญ (Suebsamran, 2007) ทำวิจัยด้านหุ่นยนต์เคลื่อนที่ไปตามตำแหน่งโดยใช้ GPS และเมื่อมีวัตถุหรือสิ่งกีดขวาง หุ่นยนต์ก็จะสามารถหลบได้ โดยหุ่นยนต์อาศัยการทำงานจาก GPS เพื่อกำหนดตำแหน่งให้ไปยังเป้าหมาย และหุ่นยนต์สามารถหลบสิ่งกีดขวางได้อย่างอัตโนมัติโดยใช้อินฟราเรด การทำงานคือ ใช้ GPS สำหรับระบุตำแหน่งเป้าหมายและตำแหน่งเริ่มต้น ซึ่งในระหว่างการเคลื่อนที่ เมื่อเซนเซอร์ตัวใดตัวหนึ่งตรวจจับสิ่งกีดขวางได้ ก็จะส่งสัญญาณไปให้ไมโครคอนโทรลเลอร์เพื่อทำการประมวลผลผลและสั่งให้หลบสิ่งกีดขวางโดยอัตโนมัติ



ภาพ 32 หุ่นยนต์เคลื่อนที่โดยอาศัยข้อมูลจากการตรวจวัดเพื่อหลบหลีกสิ่งกีดขวางแบบอัตโนมัติ ที่มา: (Suebsamran, 2007)

งานวิจัยต่อมาศึกษาเรื่อง Behavior-based Mobile Robot and Advanced Software Collaboration โดย พูลศักดิ์ โกษิยาภรณ์ และนริศร แสงคนอง (Kosiyaporn & Saengkanong, 2004) งานวิจัยนี้นำเสนอวิธีการตรวจสอบโปรแกรมการทำงานของหุ่นยนต์เคลื่อนที่ในแนวราบ โปรแกรมการทำงานนี้ทำงานเป็นพฤติกรรมอัจฉริยะของหุ่นยนต์ที่ใช้กับงานเคลื่อนที่หลบสิ่งกีดขวาง วิธีการสร้างเส้นแบบสุ่มถูกนำมาใช้ในการจำลองพฤติกรรมของเซนเซอร์โดยใช้เซนเซอร์ตรวจจับระยะแบบคลื่นสะท้อนที่สามารถวัดได้ในลักษณะตำแหน่งที่อยู่ภายในทรงกรวยโดยข้อมูลระยะนี้นำมาใช้

เป็นตัวป้องกันกลับของระบบควบคุม พฤติกรรมของหุ่นยนต์ที่ใช้ในการเคลื่อนที่ไปตามผนังโดยวิธีการของตัวควบคุมฟัซซีได้ถูกออกแบบและทดสอบ ผลการทดสอบพบว่าโปรแกรมที่ใช้ในการทดสอบในระบบจำลองสามารถนำมาใช้จริงโดยการปรับเล็กน้อย ระบบนี้สามารถนำมาใช้ในการออกแบบหุ่นยนต์ที่ใช้เซนเซอร์วัดระยะ



ภาพ 33 Behavior-based Mobile Robot and Advanced Software Collaboration

ที่มา: (Kosiyaporn & Saengkanong, 2004)

งานวิจัย การพัฒนาระบบการเคลื่อนที่ของหุ่นยนต์ด้วยฟัซซีลอจิก โดย Kataphon Prewthaisong งานวิจัยนี้เป็นการศึกษาและพัฒนาระบบการเคลื่อนที่ของ หุ่นยนต์ด้วยฟัซซีลอจิก (Prewthaisong, 2013) กฎฟัซซีที่ได้จากการออกแบบถูกนำมา ใช้ในการวิเคราะห์การเรียนรู้เส้นทาง การเคลื่อนที่ของ หุ่นยนต์ที่ควบคุมด้วยฟัซซีลอจิก การเรียนรู้ของหุ่นยนต์จะ ใช้ตัวตรวจจับแบบอินฟราเรดเซนเซอร์มาตรวจจับวัตถุและ อ่านค่าระยะของวัตถุซึ่งใช้ในการอ้างอิงตำแหน่งในการเคลื่อนที่และในส่วนฮาร์ดแวร์จะใช้ไมโครคอนโทรลเลอร์ ตระกูล PIC 16F877 เป็นตัวควบคุมหลัก และกฎฟัซซีจะใช้ ในการควบคุมการตัดสินใจเดินตามเส้นทางด้วยตัวเองได้ โดยปราศจากการควบคุมจากมนุษย์ จากผลการวิจัยและ พัฒนาระบบการเคลื่อน ที่ของหุ่นยนต์ด้วยฟัซซีลอจิกพบว่า หุ่นยนต์เคลื่อนที่สามารถใช้ข้อมูลที่ได้จากการตรวจจับวัตถุ มาประยุกต์ใช้กับกฎฟัซซีแล้วนำไปควบคุมให้หุ่นยนต์ เคลื่อนที่ได้อย่างถูกต้อง



ภาพ 34 การพัฒนาระบบการเคลื่อนที่ของหุ่นยนต์ด้วยฟัซซีลอจิก

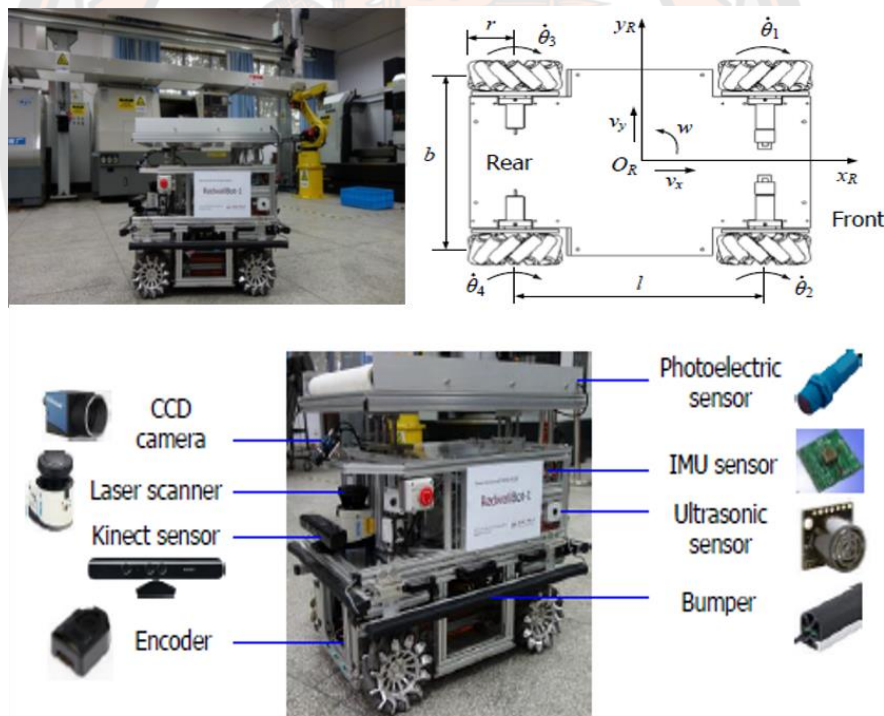
ที่มา: (Prewthaisong, 2013)

ต่อมาเป็นงานวิจัยเกี่ยวกับ หุ่นยนต์เคลื่อนที่ตามวัตถุเป้าหมายที่กำหนดโดยใช้การประมวลผลภาพ โดย วณพันธ์ วิทยุฒิ (Waywut, 2008) งานวิจัยนี้นำเสนอหุ่นยนต์ที่เคลื่อนที่โดยใช้ล้อ โดยเคลื่อนที่ไปตามภาพที่กำหนด โดยใช้กล้องจับภาพเข้ามา แล้วนำมาประมวลผล แล้วเปรียบเทียบว่าเป็นภาพที่ถูกระบุไว้หรือไม่ โดยใช้เครือข่ายแบบเครือข่าย ถ้าใช้ก็จะเข้าหา การเคลื่อนที่ที่จะถูกกำหนดให้อยู่ในแนวราบ ถ้าภาพที่กำหนดไม่อยู่ตรงกลาง หุ่นยนต์จะเคลื่อนที่ไปทางซ้ายหรือขวา เพื่อให้ภาพอยู่ในแนวกลาง บทความนี้จะนำเสนอเนื้อหาของผลการประมวลผลภาพ งานวิจัยต่อมา หุ่นยนต์เคลื่อนที่สำหรับการติดตาม การเฝ้าระวังและตรวจจับการล้นของผู้สูงอายุ โดย Tomoya และคณะ (Tomoya, et al. 2017) งานวิจัยนี้พูดถึงปัญหาของผู้สูงอายุที่ต้องพักอาศัยอยู่บ้านเพียงลำพัง มักจะเกิดอุบัติเหตุได้ง่าย เช่น การล้นของผู้สูงอายุซึ่งเป็นอันตรายมาก หุ่นยนต์เคลื่อนที่สำหรับการติดตาม การเฝ้าระวังและตรวจจับการล้นของผู้สูงอายุ โดยหลัก ๆ ตัวหุ่นยนต์ประกอบด้วยสามส่วนคือ ส่วนของฐานสำหรับให้หุ่นยนต์เคลื่อนที่ ต่อมาเป็นส่วนเซนเซอร์สำหรับตรวจจับการล้นของผู้สูงอายุ และส่วนสุดท้ายเป็นเซนเซอร์หลบสิ่งกีดขวาง โดยหุ่นยนต์เคลื่อนที่เป็นแบบเดียวกับหุ่นยนต์ดูดฝุ่นแบบวงกลมที่ใช้กันอย่างแพร่หลายในบ้าน Kobuki มีน้ำหนักเบาและมีล้อและกันชน พอร์ต USB ใช้สำหรับการต่อแบบอนุกรม เนื่องจากการเชื่อมต่อกับคอมพิวเตอร์พีซีเป็นเรื่องง่าย ทางด้านผู้วิจัยคิดว่าประเภทนี้เหมาะสำหรับการพัฒนา นอกจากนี้เรายังใช้เซ็นเซอร์ Kinect เพื่อตรวจสอบพฤติกรรมของผู้สูงอายุซึ่งช่วยให้เราสามารถทราบระยะห่างระหว่างหุ่นยนต์กับวัตถุได้ Kinect สามารถคำนวณความลึกของแต่ละจุดบนหน้าจอได้ด้วย Triangulation โดยใช้การสะท้อนของเลเซอร์อินฟราเรดที่คาดการณ์ไว้ ดังนั้นจึงสามารถตรวจจับระยะทางในรูปแบบสามมิติได้



ภาพ 35 หุ่นยนต์เคลื่อนที่สำหรับการติดตาม การเฝ้าระวังและตรวจจับการล้นของผู้สูงอายุ ที่มา: (Tomoya, et al. 2017)

งานวิจัยต่อมาเป็นเรื่อง The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System โดย Jun Qian และคณะ (Qian, et al. 2017) หุ่นยนต์เคลื่อนที่รอบทิศทางบนพื้นฐานของล้อแมคคานัมจำนวน 4 ล้อ ได้รับการออกแบบ ระบบกลไกของหุ่นยนต์เคลื่อนที่ประกอบด้วยสามชั้นที่สามารถแยกออกได้เพื่อให้การผสมผสานและการปรับโครงสร้างทำได้ง่ายขึ้น ซึ่งช่วยให้มั่นใจได้ถึงเสถียรภาพในการเคลื่อนที่และช่วยให้ระยะห่างของล้อทั้งสี่มีค่าไม่เปลี่ยนแปลง ระบบควบคุมประกอบด้วยตัวควบคุมสองระดับที่ใช้การควบคุมการเคลื่อนที่และการประมวลผลข้อมูลหลายเซ็นเซอร์ตามลำดับ เพื่อให้หุ่นยนต์เคลื่อนที่เคลื่อนที่ไปในสภาพแวดล้อมภายในอาคารที่มีโครงสร้างกึ่งโครงสร้างที่ไม่รู้จักข้อมูลจากตัวจับสัญญาณภาพ Kinect และควบคุมทั้งสี่ล้อ เพื่อกำหนดตำแหน่งหุ่นยนต์เคลื่อนที่ โดยใช้ Kalman filter มีการประมวลผลเฉพาะ สุดท้ายหุ่นยนต์เคลื่อนที่นำไปใช้ในระบบการผลิตสำหรับการลำเลียงวัสดุ

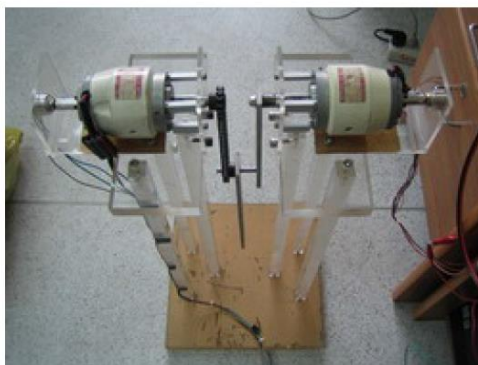


ภาพ 36 หุ่นยนต์เคลื่อนที่รอบทิศทางบนพื้นฐานของล้อแมคคานัม

ที่มา: (Qian, et al. 2017)

2.2.2 งานวิจัยเกี่ยวกับแขนหุ่นยนต์

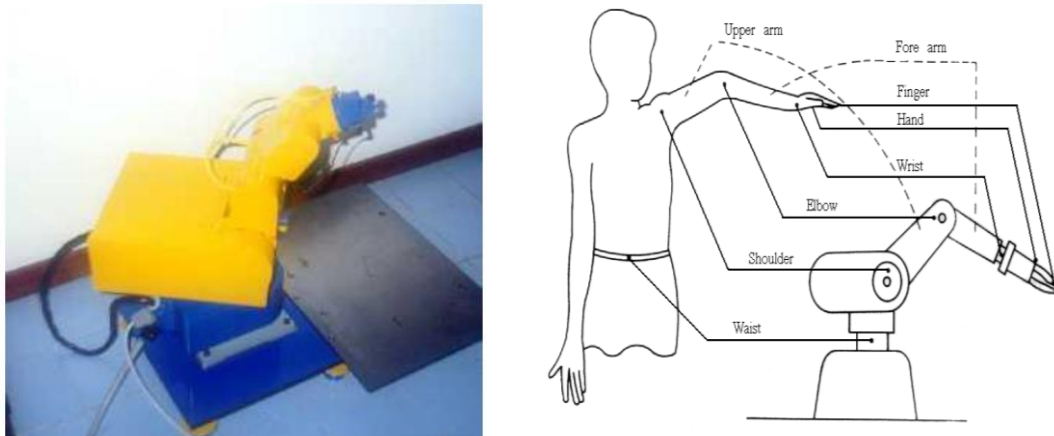
แขนหุ่นยนต์เป็นหุ่นยนต์ที่ไม่สามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง แต่สามารถเคลื่อนไหวไปมาได้ การที่หุ่นยนต์จะสามารถหยิบจับสิ่งของได้ จำเป็นต้องมีชุดแขนกล ซึ่งงานวิจัยเกี่ยวกับแขนหุ่นยนต์ก็มีหลากหลาย อย่างงานวิจัยของ อำนาจ ภูสิทธิศักดิ์ และ พิระยศ แสนโกชน์ เรื่อง การประยุกต์การควบคุมแขนหุ่นยนต์ 2 แขนแบบระนาบกรณิซับริ้วอย่างสมบูรณ์ด้วยวิธีการป้อนกลับเที่ยงตรงให้เป็นเชิงเส้น (Phoosittisak & Sanposh, 2006) บทความนี้จะนำเสนอการประยุกต์การควบคุมแขนหุ่นยนต์ 2 แขนแบบระนาบกรณิซับริ้วอย่างสมบูรณ์ (two-link planar robot arm: fully actuated system) ซึ่งเป็นระบบที่ไม่เชิงเส้น (nonlinear system) ซึ่งลักษณะทางกายภาพของระบบจะเป็นการรวมกันของระบบกรณิซับริ้วไม่สมบูรณ์ 2 ลักษณะรวมกัน การควบคุมจะใช้ หลักการการควบคุมแบบไม่เชิงเส้นด้วยวิธีการป้อนกลับเที่ยงตรงให้เป็นเชิงเส้น โดยได้จัดสร้างแขนหุ่นยนต์ขึ้นมา ทดสอบทฤษฎีข้างต้น ซึ่งจะควบคุมแขนหุ่นยนต์จากตำแหน่งล่างสุด ไปยังตำแหน่งบนสุดและยังคงอยู่ในภาวะ สมดุลแม้จะมีสัญญาณรบกวนจากภายนอก



ภาพ 37 การประยุกต์การควบคุมแขนหุ่นยนต์ 2 แขนแบบระนาบกรณิซับริ้วอย่างสมบูรณ์ด้วยวิธีการ ป้อนกลับเที่ยงตรงให้เป็นเชิงเส้น

ที่มา: (Phoosittisak & Sanposh, 2006)

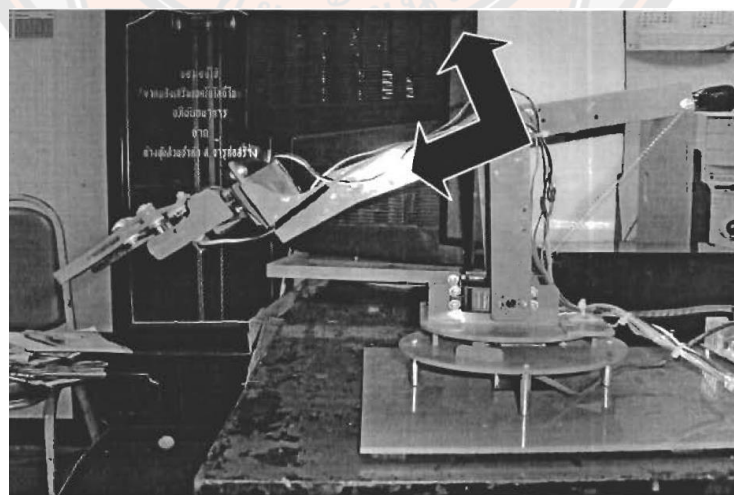
งานวิจัยต่อมา Design and Construction of Small Five-axis Robot Arm โดย รองศาสตราจารย์ชาติ ตระการกุล (Trakankul, 1999) ในการวิจัยครั้งนี้เป็นการออกแบบและสร้างแขนกลขนาดเล็กชนิด 5 แกน โดยอาศัยพื้นฐานจากหลักการเคลื่อนไหวของร่างกายมนุษย์ในส่วนของเอวและแขน ประเด็นสำคัญ คือ การออกแบบโครงสร้างและชิ้นส่วนต่างๆ ตลอดจนการออกแบบระบบควบคุมให้แขนกลสามารถเลียนแบบการเคลื่อนที่ของร่างกายมนุษย์ได้อย่างมีประสิทธิภาพและแม่นยำตามเป้าหมายที่ตั้งไว้



ภาพ 38 Design and Construction of Small Five-axis Robot Arm

ที่มา: (Trakankul, 1999)

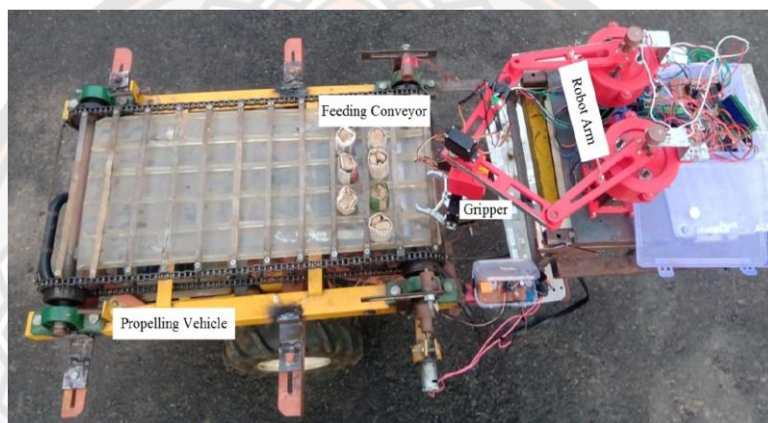
ต่อมาเป็นการวิจัยเกี่ยวกับ การพัฒนาชุดแขนกล 5 แกน เพื่อการศึกษาเชิงพาณิชย์ โดย เฉลิม ศรีสุวรรณ (Srisawan, 2007) การวิจัยครั้งนี้เป็นการวิจัยที่มีวัตถุประสงค์เพื่อ เพื่อการศึกษา ออกแบบ ชุดแขนกล แขน ประกอบด้วย ส่วนสังงานเซอร์โวมอเตอร์ ควบคุมด้วยไมโครคอนโทรล 5 เลอร์ตระกูลMCS-51 เบอร์ AT89S52 ใช้ภาษาซีในการเขียนโปรแกรม หลักการทำงานเมื่อจ่ายไฟ ให้กับชุดแขนกล ชุดแขนกลจะทำการกำหนดค่าเริ่มต้นของเซอร์โวมอเตอร์ที่ละตัว เริ่มจากฐานจนถึง ตัวคลิปปเปอร์ตามลำดับ สามารถบังคับแขนกลได้โดยผ่านปุ่มบังคับ และสามารถสั่งการทาง คอมพิวเตอร์ได้โดยผ่านพอร์ตอนุกรม



ภาพ 39 การพัฒนาชุดแขนกล 5 แกน เพื่อการศึกษาเชิงพาณิชย์

ที่มา: (Srisawan, 2007)

นอกจากนี้การควบคุมแขนหุ่นยนต์ได้นำเอาหลักการจลนศาสตร์แบบไปข้างหน้า (Forward Kinematics) และจลนศาสตร์แบบผกผัน (Inverse Kinematics) เพื่อให้ทราบตำแหน่งของแขนหุ่นยนต์ จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่า แขนหุ่นยนต์สามารถนำมาใช้งานได้อย่างหลากหลาย ในงานวิจัยของ (Jung & Bae, 2015) ได้นำเอาหลักการจลนศาสตร์แบบไปข้างหน้า และจลนศาสตร์แบบผกผัน มาวิเคราะห์หาพื้นที่การทำงานของแขนหุ่นยนต์ เพื่อใช้ในการบำบัดผู้พิการ ช่วยในการพยุงและฝึกการเคลื่อนที่ของแขน งานวิจัยของ (Rahul, et al. 2019) ได้ทำการออกแบบและพัฒนา แขนหุ่นยนต์เคลื่อนที่ในแนวระนาบ เพื่อใช้ในการจัดเรียงกระถางกระดาศของต้นกล้า ซึ่งทำงานร่วมกับสายพานลำเลียง โดยใช้หลักการจลนศาสตร์แบบไปข้างหน้าและแบบผกผัน แขนหุ่นยนต์จัดเรียงกระถางกระดาศของต้นกล้าแสดงดังภาพที่ 40



ภาพ 40 แขนหุ่นยนต์จัดเรียงกระถางกระดาศของต้นกล้า

ที่มา: (Rahul, et al. 2019)

ต่อมาเป็นงานวิจัยของ (Manikphan, et al. 2010) ได้จำลองการทำงานแขนหุ่นยนต์แบบ 6 แกน เพื่อหาพื้นที่ทำงานของแขนหุ่นยนต์ ได้ใช้ระบบโครงข่ายประสาทเทียม โดยนำโมเดลคำนวณในรูปแบบ D-H หรือ Denavit – Hartenberg ในการวิเคราะห์การทำงานของแขนหุ่นยนต์จำลองก่อน จะใช้สมการจลนศาสตร์แบบไปข้างหน้า ในการหาความสัมพันธ์ของข้อต่อแต่ละข้อ เพื่อนำมาใช้ในการแก้ปัญหาสมการจลนศาสตร์แบบผกผันของ แขนกลแบบหมุน 6 ข้อต่อ สุดท้ายเป็นงานวิจัยของ (Charoenseang & Chuakong, 2013) เป็นการสร้างแขนหุ่นยนต์เพื่อป้อนอาหารคนพิการ ทำงานร่วมกับระบบคอมพิวเตอร์วิชั่น ติดตั้งอยู่บนรถเข็นคนพิการ ซึ่งใช้หลักการจลนศาสตร์แบบผกผันเพื่อหาตำแหน่งของข้อต่อ เป็นการสร้างแขนหุ่นยนต์แขนกลแบบ SCARA ที่มี 5 องศาอิสระ ซึ่งติดตั้งบนรถเข็นผู้พิการ เพื่อป้อนอาหารคนพิการ ทำงานร่วมกับระบบคอมพิวเตอร์วิชั่น ติดตั้งอยู่บนรถเข็นคนพิการ ซึ่งใช้หลักการจลนศาสตร์แบบผกผันเพื่อหาตำแหน่งของข้อต่อ แขนหุ่นยนต์เพื่อป้อนอาหารคนพิการแสดงดังภาพที่ 41



ภาพ 41 แขนหุ่นยนต์เพื่อป้อนอาหารคนพิการ

ที่มา: (Charoenseang & Chuakong, 2013)

2.3 หุ่นยนต์ที่ใช้งานในโรงพยาบาล

จากการศึกษาเบื้องต้นมีหุ่นยนต์ที่นำมาใช้ในการช่วยงานภายในโรงพยาบาล เพื่อใช้งานในด้านบริการต่าง ๆ แทนบุคลากรโรงพยาบาล เช่น งานด้านการส่งยาเวชภัณฑ์ อาหาร น้ำดื่ม รวมถึงสิ่งของอื่นๆ ให้กับผู้ป่วย ติดต่อสื่อสารกับแพทย์ในระยะไกลโดยมีหุ่นยนต์เป็นสื่อกลาง เป็นต้น ทั้งนี้จะขอยกตัวอย่างหุ่นยนต์ที่ใช้ในโรงพยาบาล ที่มีอยู่ประมาณ 4 ตัว หุ่นยนต์ที่นำมายกตัวอย่างนี้เป็นหุ่นยนต์ที่ใช้ในโรงพยาบาลในประเทศไทย เริ่มที่หุ่นยนต์ตัวแรก เฮลท์ดีบอท (Healthy Bot) เป็นหุ่นยนต์จากบริษัท N Health ที่นำมาใช้ในโรงพยาบาลเพื่อเพิ่มอำนวยความสะดวก สะดวก ในกระบวนการจัดส่งยาเวชภัณฑ์ อาหารและน้ำดื่ม สำหรับจุดบริการผู้ป่วยนอกของโรงพยาบาล พร้อมทั้งให้ข้อมูลประชาสัมพันธ์ต้อนรับด้านหน้า และมีความสามารถ สื่อสารกับผู้ป่วยด้วยโปรแกรมสื่อสารพูดคุยผ่านจอมอนิเตอร์ที่ติดตั้งไว้บนตัวหุ่นยนต์ ซึ่งถูกควบคุมด้วยเครือข่ายสัญญาณอินเทอร์เน็ต หุ่นยนต์สามารถส่งงานผ่านหน้าจอด้านบนเครื่อง หรือสามารถส่งผ่านมือถือหรือแท็บเล็ต และมีระบบการสร้างแผนที่เพื่อกำหนดเส้นทางการเดินอัตโนมัติ และเซ็นเซอร์ที่ช่วยวิเคราะห์เพื่อหลบหลีกอุปสรรคและสิ่งกีดขวาง มีการติดตั้งติดกล้องไว้ในหุ่นยนต์ เพิ่มความสามารถ ในการสนทนา เห็นหน้าของอีกฝ่าย (N Health, 2021) หุ่นยนต์เฮลท์ดีบอทแสดงดังภาพที่ 42



ภาพ 42 หุ่นยนต์ Healthy Bot

ที่มา : (N Health, 2021)

ต่อมาเป็นหุ่นยนต์ HAPYBot ที่มีความสามารถในการเคลื่อนที่อัตโนมัติ เพื่อช่วยในการรับ-ส่ง อุปกรณ์และเวชภัณฑ์ทางการแพทย์ ตัวหุ่นยนต์มีช่องเก็บแบบปิด, เปิดและล็อคด้วยไฟฟ้า ขนาดจุ 17 ลิตร รองรับน้ำหนักบรรทุกทุกได้ 10-15 กิโลกรัม ในช่องเก็บของในตัวหุ่นยนต์สามารถใส่ประเภทน้ำได้ เพราะเป็นช่องหล่อในตัวขึ้นเดียว ไม่มีรอยต่อ กันน้ำไม่ให้อุปกรณ์ระบบหุ่นยนต์ พร้อมทั้งความสามารถในการนำทางและให้คำแนะนำผู้ป่วย ผ่านการสื่อสารของแพทย์ได้ สามารถสั่งการผ่านหน้าจอสัมผัสบนตัวหุ่นยนต์หรือสมาร์ทโฟนได้ โดยไม่ต้องเชื่อมต่อกับระบบ network ของโรงพยาบาล ทำให้การจัดส่งยาไปยังที่หมายได้หลายที่ ขึ้นกับคำสั่งการใช้งาน ขณะนี้ในด้านความเร็วเคลื่อนที่ของหุ่นยนต์จะเร็วประมาณ 2.5 ถึง 3 กิโลเมตร ต่อชั่วโมง สามารถหลบหลีกสิ่งกีดขวางได้แบบ real time และไม่เกิดอุบัติเหตุต่อผู้มารับบริการ บุคลากร คนป่วยนั่งรถเข็นอุปกรณ์ วัสดุเครื่องใช้ และเฟอร์นิเจอร์ในโรงพยาบาล อีกทั้งยังมีระบบตรวจการรบกวนการทำงานของหุ่นยนต์โดยการแจ้งเป็นเสียงเตือน เมื่อมีการกระทำกรกับตัวหุ่นยนต์ เช่น ยก โยก คว่า หุ่นยนต์ ทางหุ่นยนต์ก็จะส่งเสียงสัญญาณ หุ่นยนต์สามารถสร้างแผนการเดินทางได้เองอัตโนมัติ และสามารถกำหนดพื้นที่ที่ต้องการหลีกเลี่ยงการเดินทางได้เองโดยใช้ระบบ AI เข้ามาช่วยตัดสินใจเส้นทางใหม่ ซึ่งเมื่อมีอุปสรรคก็จะหลบหลีกแบบอัตโนมัติได้อย่างทันที อีกทั้งยังมีวิธีปลดล็อคช่องเก็บของฉุกเฉิน มีระบบล็อคการเปิดเครื่องด้วยกุญแจ สามารถตั้งความดังของเสียง และจำแนกความดังของเสียงทั้งเวลากลางวัน และกลางคืนได้ รวมทั้งกำหนดเสียงได้ทั้งภาษาไทย และภาษาอังกฤษ อีกทั้งยังสามารถติดอุปกรณ์เสริม เพื่อสแกนบัตรผู้รับยา แบบ RFID ได้ โดยทางโรงพยาบาลสามารถเพิ่มสิทธิ์เองได้

สามารถตรวจสอบ และออกรายงาน สามารถออกรายงานวันที่ เวลาที่ส่ง/รับยา จุดเริ่มต้น/ปลายทาง ระยะทาง ระยะเวลา รวมทั้งผู้ส่ง รับของได้ หุ่นยนต์ HAPYBot แสดงดังภาพที่ 43



ภาพ 43 หุ่นยนต์ HAPYBot

ที่มา : (med mahidol, 2020)

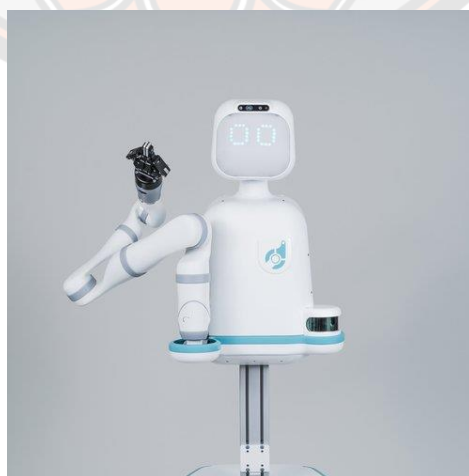
ต่อมา หุ่นยนต์ปั่นโต หุ่นยนต์พัฒนาเพื่อการขนส่งและการสื่อสารทางไกลเพื่อช่วยบุคลากรทางการแพทย์ในช่วงสถานการณ์การระบาดของโควิด โดยหุ่นยนต์ปั่นโต ทำหน้าที่ในการขนส่งอาหาร ยา เวชภัณฑ์แก่คนไข้ โดยการใช้รถเข็นควบคุมทางไกล ทำให้บุคลากรทางการแพทย์ลดความเสี่ยงลดจำนวนครั้งที่ต้องเข้าไปอยู่ใกล้ชิดผู้ติดเชื้อ และยังสามารถคอยสอดส่องคนไข้อย่างใกล้ชิดได้ตลอด เพราะมีการสื่อสารได้ระยะไกลด้วยหน้าจอที่ติดอยู่กับรถเข็นควบคุมทางไกล จะทำให้บุคลากรทางการแพทย์ สามารถคอยดูแลคนไข้ได้ตลอด โดยที่ระบบนี้จะช่วยลดทั้งอุปกรณ์การแพทย์ที่ต้องทิ้งทุกครั้งเมื่อเปลี่ยนโซนการตรวจ ลดระยะเวลาการเข้าใกล้ชิดผู้ติดเชื้อ ลดความเสี่ยงทำให้เพิ่มประสิทธิภาพ และยังคงดูแลผู้ป่วยได้อย่างเต็มที่ หุ่นยนต์ปั่นโต แสดงดังภาพที่ 44



ภาพ 44 หุ่นยนต์ Pinto

ที่มา : (Pinto, 2020)

หุ่นยนต์ Moxi ได้รับการออกแบบโดย Austin-based Diligent Robotics กำลังถูกนำมาทดลองใช้ที่โรงพยาบาลเท็กซัส เพื่อหารูปแบบการทำงานร่วมกับมนุษย์ โดยหุ่นยนต์ Moxi ทำหน้าที่ในการช่วยเตรียมสิ่งของต่าง ๆ ที่พยาบาลต้องใช้ เช่น ผ้าพันแผล น้ำเกลือ อุปกรณ์เข็มฉีดยา หรืออื่น ๆ ที่อยู่ในขอบเขตที่ทำได้ ขณะหุ่นยนต์เคลื่อนที่แล้วทำกล่องอุปกรณ์ล่องลงพื้น ตัวหุ่นยนต์ยังสามารถแก้ปัญหาด้วยเก็บอุปกรณ์ทั้งหมดขึ้นมาใหม่และนำมันไปยังพื้นที่สำหรับความสะดวกด้วยตนเอง เปรียบให้เห็นถึงการแก้ปัญหาเฉพาะหน้า หุ่นยนต์ Moxi แสดงดังภาพที่ 45



ภาพ 45 หุ่นยนต์ Moxi

ที่มา : (Moxi, 2020)

จากหุ่นยนต์ที่นำมาใช้ในโรงพยาบาลที่ยกตัวอย่างมา นำมาเปรียบเทียบกับลักษณะการทำงาน และความสามารถของหุ่นยนต์ แสดงดังตารางที่ 2

ตาราง 2 แสดงการเปรียบเทียบการใช้งานของหุ่นยนต์ที่ใช้ในโรงพยาบาล

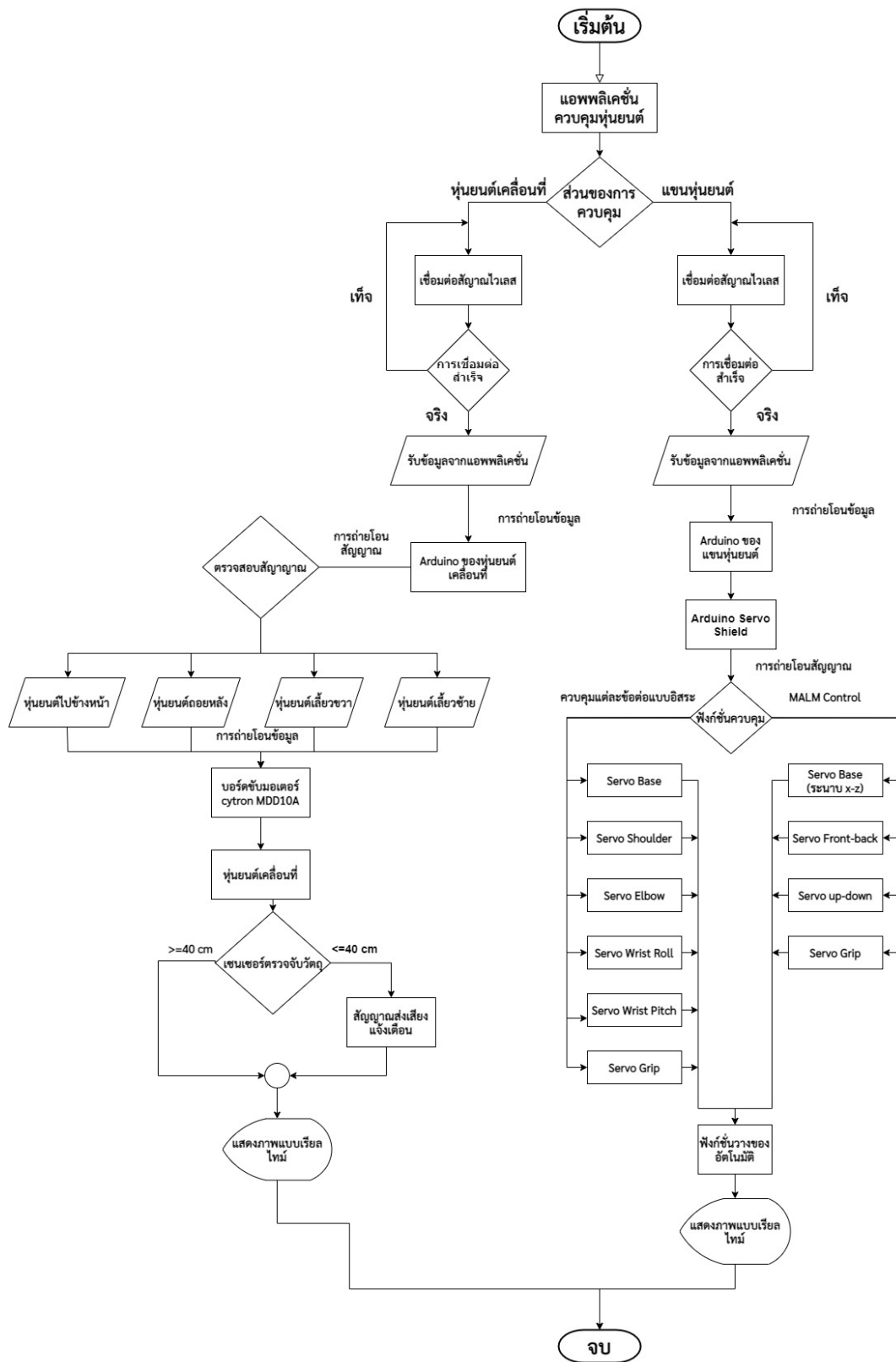
หุ่นยนต์ การใช้งาน	Healthy Bot	HAPYBot	Pinto	Moxi	หุ่นยนต์ใน งานวิจัยนี้
ควบคุมไร้สาย	✓	✓	✓	✓	✓
มีการแสดงภาพ แบบเรียลไทม์	✓	✓	✓	✓	✓
แขนหุ่นยนต์ หยิบสิ่งของ				✓	✓
สร้างแผนที่เพื่อ กำหนดเส้นทาง การเดินอัตโนมัติ	✓	✓		✓	
ช่องว่างสิ่งของ เช่นยา อาหาร	✓	✓	✓	✓	✓
เซนเซอร์ ตรวจจับวัตถุ	✓	✓		✓	✓

จากตารางที่ 2 หุ่นยนต์ในงานวิจัยนี้มีคุณสมบัติที่ครอบคลุมการใช้งานเมื่อเปรียบเทียบกับหุ่นยนต์ที่ใช้งานในโรงพยาบาลที่ยกตัวอย่างมาข้างต้น ด้วยฟังก์ชันต่าง ๆ ทั้งการควบคุมไร้สายที่สามารถควบคุมผ่านแอปพลิเคชันบนสมาร์ตโฟน อีกทั้งยังสามารถแสดงภาพแบบเรียลไทม์ได้ มีแขนหุ่นยนต์สำหรับหยิบจับสิ่งของ มีช่องใส่ของต่าง ๆ เช่น ยาเวชภัณฑ์ อาหาร น้ำดื่ม รวมถึงสิ่งของอื่นๆ เพื่ออำนวยความสะดวกให้กับผู้ใช้งาน และยังมีเซนเซอร์ต่าง ๆ ที่ช่วยให้หุ่นยนต์มีความสามารถที่หลากหลายครบครัน เช่น เซนเซอร์ตรวจจับวัตถุ เซนเซอร์วัดความเร็วและระยะทาง เป็นต้น

บทที่ 3

การออกแบบและพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง

ในบทนี้จะกล่าวถึงวิธีการออกแบบและพัฒนาตลอดจนถึงขั้นตอนการทำงานต่างๆ ของหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง วิธีการควบคุมหุ่นยนต์ ซึ่งเป็นหุ่นยนต์ที่พัฒนาขึ้นมาสำหรับช่วยเหลือผู้พิการท่อนล่าง ผู้สูงอายุ ผู้ป่วยติดเตียง หรือผู้ที่ขยับตัวลำบากในการเคลื่อนที่เพื่อจะลุกไปหยิบจับสิ่งของ หุ่นยนต์จะประกอบไปด้วยส่วนของหุ่นยนต์เคลื่อนที่ซึ่งทำหน้าที่เคลื่อนไปยังตำแหน่งหรือจุดหมายที่กำหนด โดยมีความสามารถในการหลบหลีกสิ่งกีดขวาง โดยภายในหุ่นยนต์เคลื่อนที่จะมีเซนเซอร์ตรวจจับวัตถุ เมื่อหุ่นยนต์เข้าใกล้สิ่งกีดขวางในระยะที่กำหนดจะมีเสียงสัญญาณแจ้งเตือนให้ผู้ใช้งานได้ทราบ เพื่อป้องกันการชนเข้ากับสิ่งกีดขวาง เป็นการลดข้อผิดพลาดจากจุดอับสายตาของมวมกล้อง ต่อมาส่วนของแขนหุ่นยนต์ที่สามารถหยิบจับสิ่งของได้ด้วยระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน และส่วนสุดท้ายเป็นแอปพลิเคชันบนโทรศัพท์มือถือ ใช้ในการควบคุมหุ่นยนต์ ซึ่งการทำงานของหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง จะมีการทำงานร่วมกันหลายส่วน เริ่มต้นจากผู้ใช้ควบคุมหุ่นยนต์ผ่านแอปพลิเคชัน โดยแอปพลิเคชันจะแบ่งการควบคุมเป็นสองส่วนคือ ส่วนของการควบคุมหุ่นยนต์เคลื่อนที่ และส่วนของการควบคุมแขนหุ่นยนต์ ภายในหน้าจอของแอปพลิเคชันจะประกอบไปด้วยส่วนต่างๆ เช่น ส่วนของการเลือกสัญญาณเชื่อมต่อกับอุปกรณ์ หน้าจอแสดงผลจากกล้องตามเวลาจริง ส่วนของปุ่มบังคับเพื่อใช้ในการควบคุมหุ่นยนต์ เป็นต้น ต่อมาในส่วนของหุ่นยนต์เคลื่อนที่จะมีความสามารถในการหลบหลีกสิ่งกีดขวางได้ เมื่อเข้าใกล้วัตถุตามระยะที่กำหนด เซนเซอร์จะสามารถตรวจจับและส่งสัญญาณเสียงให้ผู้ใช้งานรู้ว่าหุ่นยนต์เข้าใกล้กับสิ่งกีดขวางมากเกินไป และส่วนของแขนหุ่นยนต์จะมีระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-Axis Linear Motion Control -MALM Control) เป็นการควบคุมมอเตอร์หลายตัวพร้อมกัน เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่ตามแนวหน้า-หลัง และแนวขึ้น-ลง เหมาะสำหรับการควบคุมผ่านหน้าจอสมาร์ตโฟน การทำงานของหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง แสดงดังภาพที่ 46 ในงานวิจัยนี้มุ่งเน้นทำหุ่นยนต์ที่มีราคาประหยัดเพื่อให้คนไทยสามารถเข้าถึงและใช้งานเทคโนโลยีหุ่นยนต์ได้มากยิ่งขึ้น การเลือกวัสดุอุปกรณ์บางอย่างจึงมีข้อจำกัดทางด้านราคา เพื่อให้สามารถทำหุ่นยนต์ให้มีราคาประหยัดมากที่สุดและมีประสิทธิภาพที่ดี สามารถใช้งานได้อย่างสมบูรณ์



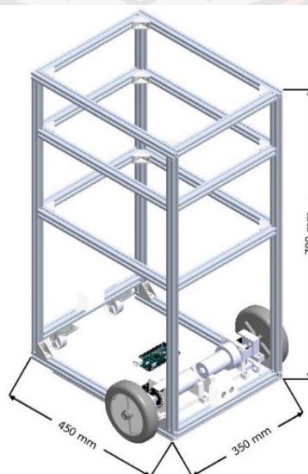
ภาพ 46 แผนผังการทำงานของหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง

โดยงานวิจัยนี้เป็นงานวิจัยที่มุ่งเน้นการออกแบบและสร้างหุ่นยนต์ขึ้นมาใหม่ อีกทั้งเขียนแอปพลิเคชันบนสมาร์ตโฟนเพื่อใช้ในการควบคุมชิ้นงาน ดังนั้นขั้นตอนและวิธีการดำเนินการวิจัยจึงแบ่งออกเป็นหัวข้อต่างๆ เพื่อให้การดำเนินการเป็นไปด้วยความเรียบร้อยและตรงตามแผนที่วางไว้ โดยมีรายละเอียดที่แสดงไว้ในหัวข้อต่างๆ ดังต่อไปนี้

1. การออกแบบโครงสร้างหุ่นยนต์และสร้างหุ่นยนต์เคลื่อนที่
2. การออกแบบและสร้างวงจรควบคุมหุ่นยนต์เคลื่อนที่
3. โมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์
4. การออกแบบโครงสร้างแขนหุ่นยนต์และสร้างแขนหุ่นยนต์
5. การออกแบบและสร้างวงจรควบคุมแขนหุ่นยนต์
6. โมเดลระบบควบคุมแขนหุ่นยนต์สำหรับการเคลื่อนที่เชิงเส้นแบบหลายแกน
7. การออกแบบและสร้างแอปพลิเคชันบนสมาร์ตโฟนเพื่อใช้ในการติดต่อและควบคุมหุ่นยนต์
8. กระบวนการทำงานของหุ่นยนต์ โดยการควบคุมผ่านแอปพลิเคชันบนสมาร์ตโฟน

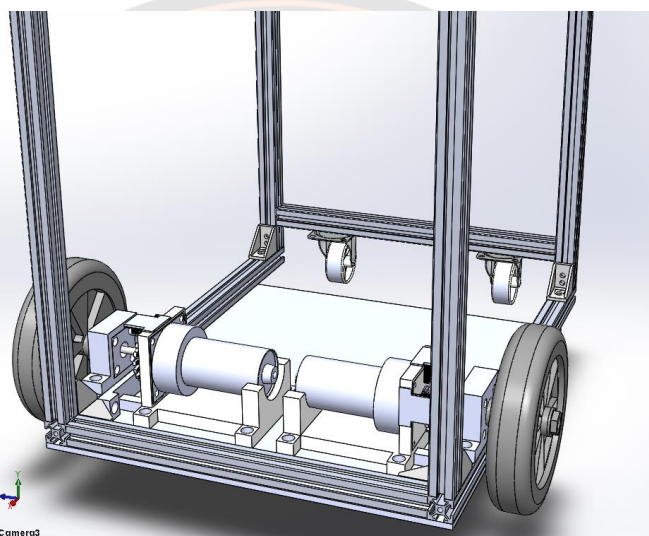
3.1 การออกแบบโครงสร้างหุ่นยนต์และสร้างหุ่นยนต์เคลื่อนที่

โครงสร้างหุ่นยนต์เคลื่อนที่นี้ถือว่าเป็นโครงสร้างหลักของหุ่นยนต์บริการสำหรับผู้พิการติดเตียง ซึ่งในงานวิจัยนี้จะมีแขนหุ่นยนต์ติดตั้งอยู่ด้านบนของตัวหุ่นยนต์เคลื่อนที่ การออกแบบหุ่นยนต์เคลื่อนที่สำหรับบริการผู้พิการติดเตียง ต้องมีการคำนึงถึงการใช้งานที่สะดวกเหมาะสมกับหน้าที่การใช้งานของหุ่นยนต์ เช่น ความสูงของหุ่นยนต์ต้องที่ขนาดที่เหมาะสมกับโต๊ะวางของในขนาดมาตรฐานทั่วไป เพื่อที่แขนหุ่นยนต์สามารถเคลื่อนที่เข้าไปหยิบสิ่งของได้ และขนาดตัวของหุ่นยนต์เองต้องมีลักษณะที่มีความคล่องตัว โดยทำการออกแบบโครงสร้างในโปรแกรมจำลองที่มีชื่อว่า Solid work เพื่อกำหนดรูปร่างของหุ่นยนต์ให้สอดคล้องการการประยุกต์ใช้งานกับผู้พิการติดเตียง หุ่นยนต์มีขนาดที่กว้าง 350 mm ยาว 450 mm และสูงที่ 790 mm แสดงดังภาพที่ 47



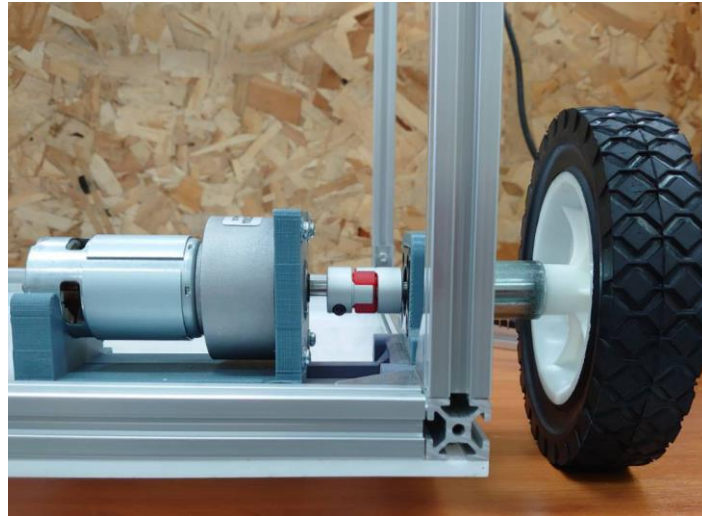
ภาพ 47 การออกแบบโครงสร้างหุ่นยนต์เคลื่อนที่

หุ่นยนต์เคลื่อนที่ในงานวิจัยนี้เป็นหุ่นยนต์ประเภท Differential Drive Robot แสดงดังภาพที่ 48 ซึ่งเป็นรูปแบบการเคลื่อนที่ด้วยล้อที่ง่ายที่สุด คือ ด้านหน้าจะมี ล้อซ้าย และล้อขวา ส่วนล้อหลังจะเป็นล้อที่สามารถเคลื่อนที่ได้อย่างอิสระ โดยถ้าต้องการให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า ให้ทำการหมุนซ้ายและขวาไปทางเดียวกันด้วยความเร็วเท่ากัน การถอยหลังก็ให้ทำนองเดียวกัน ต่อมา การเลี้ยวซ้าย ให้หมุนล้อขวาไปข้างหน้าให้เร็วกว่าล้อซ้าย หรือการเลี้ยวขวา ให้หมุนล้อซ้ายไปข้างหน้าให้เร็วกว่าล้อขวา และการให้หุ่นยนต์ทำการหมุนตัวได้ ให้ทำการหมุนล้อข้างหนึ่งไปข้างหน้า และล้ออีกข้างหมุนกลับหลังด้วยความเร็วเท่ากัน



ภาพ 48 การออกแบบตำแหน่งของล้อหุ่นยนต์เคลื่อนที่

วัสดุของหุ่นยนต์เคลื่อนที่จะถูกออกแบบให้มีลักษณะที่ไม่ซับซ้อน มีขนาดที่สมมาตรโดยโครงสร้างจะใช้อลูมิเนียมโปรไฟล์เป็นวัสดุในการทำโครง เพื่อง่ายต่อการประกอบ และมีความแข็งแรงแก้ปัญหาในส่วนของการที่ต้องรับน้ำหนักที่มากเกินไป ในส่วนของมอเตอร์ที่ใช้ในการขับเคลื่อน เป็นส่วนที่ต้องแบกรับน้ำหนักทั้งหมดของหุ่นยนต์ ถ้าทำการติดตั้งมอเตอร์เข้ากับล้อโดยตรง อาจทำให้มอเตอร์แบกรับน้ำหนักที่มากเกินไป อาจจะเกิดความเสียหายได้กับมอเตอร์ได้ ดังนั้นทางผู้วิจัยได้ทำการออกแบบการติดตั้งมอเตอร์ให้สามารถรองรับน้ำหนักของตัวหุ่นยนต์ ป้องกันความเสียหายกับตัวมอเตอร์ได้ โดยเริ่มจากสวมแกนเพลลาเข้ากับล้อ จากนั้นประกอบติดกับ bearing housing และ Coupling จากนั้นถึงสวมแกนมอเตอร์เข้ากับ Coupling แสดงดังภาพที่ 49



ภาพ 49 การติดตั้งมอเตอร์เข้ากับล้อ ผ่านแกนเพลลาประกอบติดกับ bearing housing และ Coupling

เมื่อทำออกแบบโครงสร้างของหุ่นยนต์เคลื่อนที่ในโปรแกรมจำลองแล้ว ทำการประกอบโครงสร้างหุ่นยนต์เคลื่อนที่ต้นแบบขึ้นมา เมื่อประกอบโครงสร้างเสร็จแล้ว จากนั้นติดตั้งอุปกรณ์ เช่น ชุดล้อ กล้อง อุปกรณ์ควบคุม และเซนเซอร์ต่าง ๆ แสดงดังภาพที่ 50

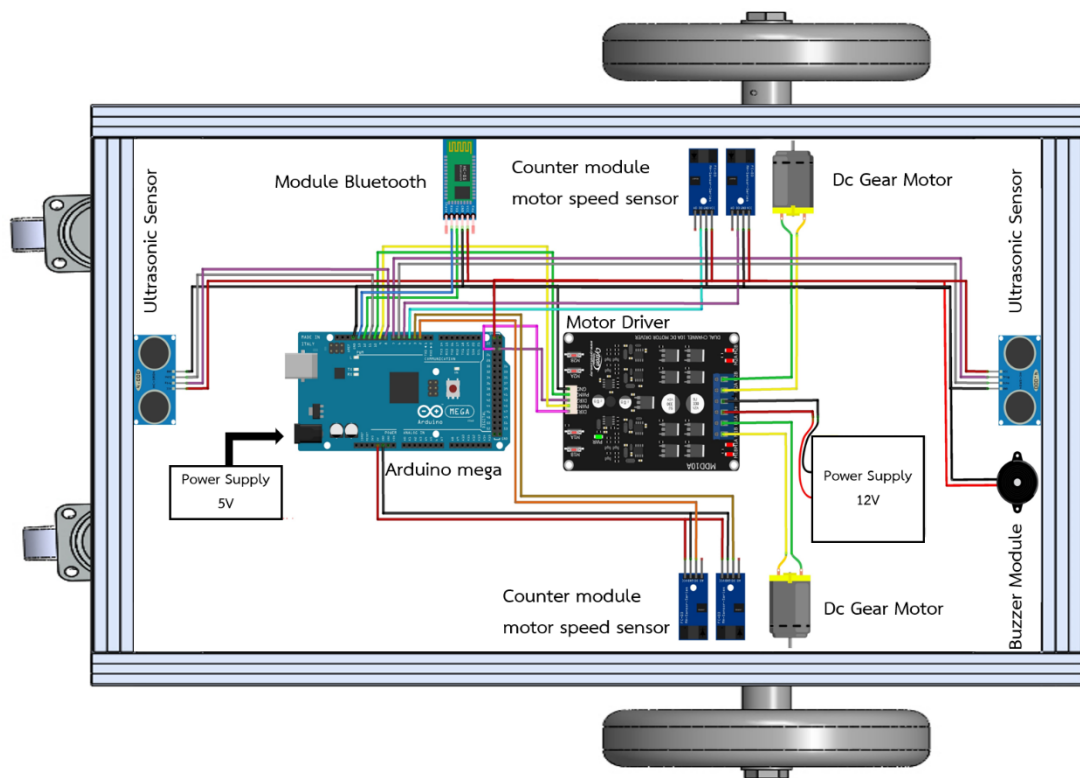


ภาพ 50 โครงสร้างหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียงต้นแบบ

3.2 การออกแบบและสร้างวงจรควบคุมหุ่นยนต์เคลื่อนที่

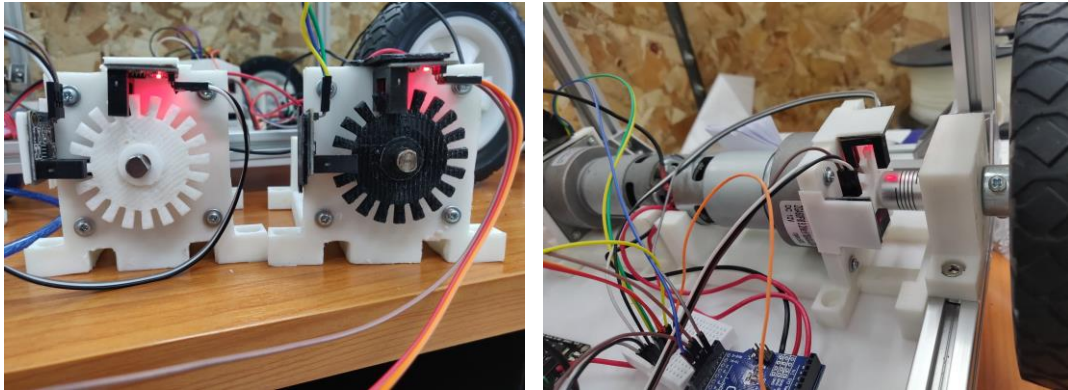
ในงานวิจัยนี้เลือกใช้บอร์ด Arduino เป็นชุดคอนโทรลเลอร์ทำงานร่วมกับแอปพลิเคชันบนสมาร์ทโฟน ซึ่งติดต่อสื่อสารผ่านสัญญาณ Bluetooth เพื่อควบคุมการทำงานของหุ่นยนต์เคลื่อนที่

ไม่ว่าจะเป็นควบคุมความเร็วและทิศทางของมอเตอร์กระแสตรง และมีบอร์ดขับมอเตอร์ ทำหน้าที่ให้พลังงานแก่มอเตอร์รวมถึงส่งสัญญาณไฟฟ้า เพื่อให้มอเตอร์เคลื่อนที่ไปยังตำแหน่งต่าง ๆ ซึ่งทำให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมายที่กำหนดตามที่ผู้ใช้กำหนดผ่านแอปพลิเคชัน นอกจากนี้ยังมีเซ็นเซอร์ต่าง ๆ ในการตรวจวัดค่าสถานะของหุ่นยนต์ ได้แก่ เซ็นเซอร์เอนโค้ดเดอร์ที่ใช้ในการวัดรอบ ความเร็ว และทิศทางของมอเตอร์ และเซ็นเซอร์ ultrasonic มีหน้าที่ในการตรวจจับวัตถุหรือสิ่งกีดขวาง เพื่อสามารถแจ้งเตือนผ่านสัญญาณเสียงด้วย buzzer module ให้ผู้ใช้ทราบ แสดงดังภาพที่ 51



ภาพ 51 วงจรของหุ่นยนต์เคลื่อนที่

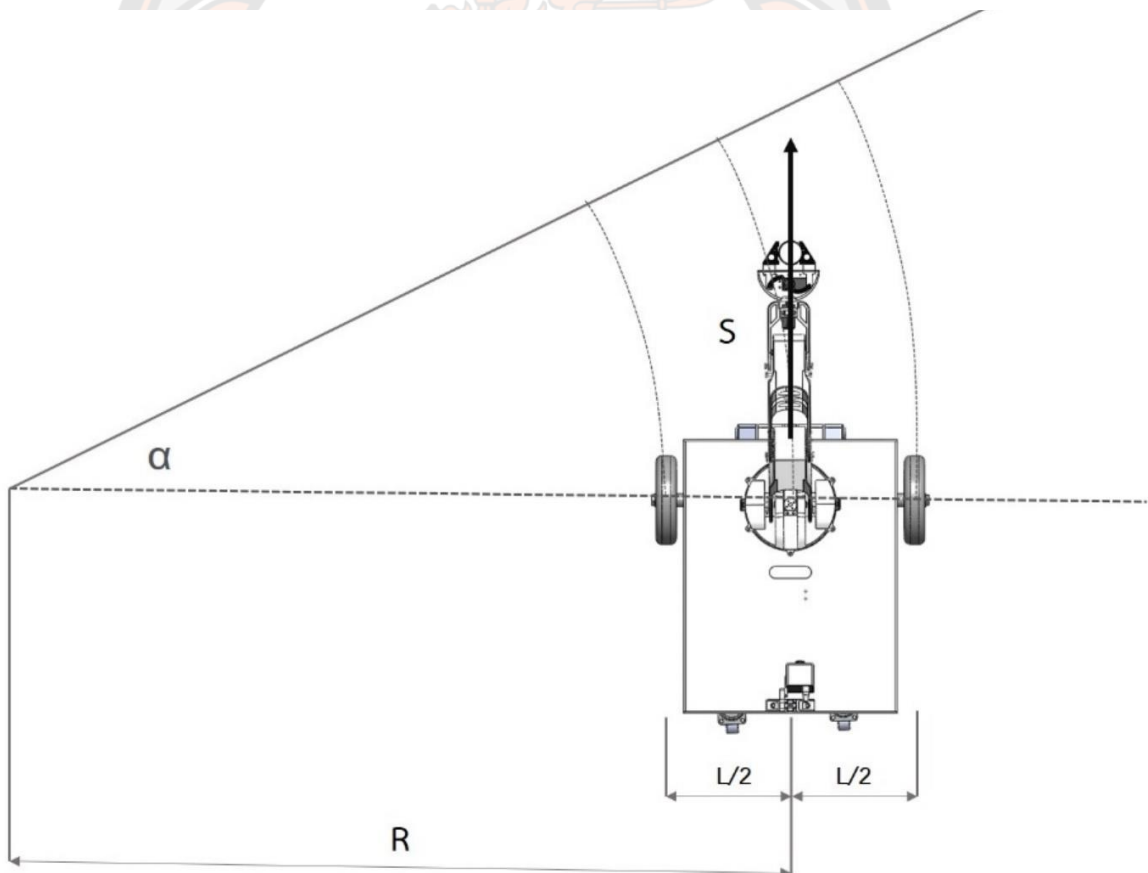
ในส่วนของการวัดความเร็วและทิศทางของมอเตอร์กระแสตรงได้ จำเป็นต้องมีอุปกรณ์ตรวจวัด เช่น เซ็นเซอร์เอนโค้ดเดอร์ เป็นต้น ซึ่งในงานวิจัยนี้ได้ประยุกต์นำเอา counter module motor speed sensor มาทำการออกแบบและประกอบติดตั้งกับมอเตอร์กระแสตรง โดยออกแบบเซ็นเซอร์ที่ 20 พัลส์ต่อรอบ ซึ่งเป็นจำนวนที่เพียงพอในงานวิจัยนี้ แสดงดังภาพที่ 52



ภาพ 52 การออกแบบและติดตั้งเซนเซอร์เอนโค้ดเดอร์

3.3 โมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์โดยทั่วไปจะมีเดินหน้า-ถอยหลัง และมีการบังคับเลี้ยวซ้าย-ขวา เพื่อเป็นการเคลื่อนที่การเลี้ยวโค้ง งานวิจัยนี้ได้คิดโมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์ ซึ่งเป็นการแสดงถึงความสัมพันธ์ระหว่างค่า PWM กับค่าความเร็ว แสดงได้ดังภาพที่ 53



ภาพ 53 โมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้ง

จากภาพที่ 53 จะได้ความสัมพันธ์ตามสมการที่ 1

$$R\alpha = 2\pi r \quad (1)$$

$$\alpha = \frac{S}{R} \quad (2)$$

จากสมการ (1) และสมการ (2) จะได้

$$\left(R - \frac{L}{2}\right)\alpha = S_1 \quad (3)$$

$$\left(R + \frac{L}{2}\right)\alpha = S_2 \quad (4)$$

โดยที่ S คือระยะทางของล้อเคลื่อนที่ในแนวระนาบกับพื้น และระยะทางที่เกิดขึ้นในแนวระนาบเกิดจากระยะที่เกิดจากการหมุนของล้อ สมการ (5) แสดงความสัมพันธ์ระหว่างระยะในแนวระนาบกับการหมุนของล้อ

$$S = r\theta \quad (5)$$

โดยที่ S คือระยะที่เกิดจากการหมุนของล้อ

θ คือมุมที่ล้อได้กวาดไป

r คือรัศมีของล้อ

เมื่อแทน S ในสมการ (3) และ (4) ด้วยความสัมพันธ์จากสมการ (5) จะได้สมการ (6) และ (7)

$$\left(R - \frac{L}{2}\right)\alpha = r\theta_1 \quad (6)$$

$$\left(R + \frac{L}{2}\right)\alpha = r\theta_2 \quad (7)$$

เมื่อนำสมการ (6) หาค่าด้วยสมการ (7) จะได้อัตราส่วนมุมของล้อด้านซ้ายและขวาตามสมการ (8)

$$\frac{\left(R - \frac{L}{2}\right)}{\left(R + \frac{L}{2}\right)} = \frac{\theta_1}{\theta_2} \quad (8)$$

ด้วยความสัมพันธ์ของมุมและอัตราเร็วเชิงมุมจากสมการ (9) ในสภาวะเวลาที่เดียวกันสมการ (8) จึงเขียนอยู่ในรูปอัตราส่วนของอัตราเร็วเชิงมุม แล้วแทนด้วยอัตราส่วนของมุมได้ และสามารถจัดรูปเพื่อหาค่าของรัศมีความโค้งได้ตามสมการ (10) แสดงให้เห็นว่ารัศมีความโค้งของหุ่นยนต์ขึ้นอยู่กับอัตราเร็วเชิงมุมของล้อ ดังนั้นหากควบคุมอัตราเร็วของล้อได้ก็จะสามารถควบคุมทิศทางของหุ่นยนต์ให้เคลื่อนตามรัศมีความโค้งที่ต้องการ

$$\theta = \int_{t_1}^{t_2} \omega dt \quad (9)$$

$$R = \frac{L(\omega_2 + \omega_1)}{2(\omega_2 - \omega_1)} \quad (10)$$

ทั้งนี้ ในการควบคุมมีสมมุติฐานว่า โมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์ ซึ่งเป็นการแสดงถึงความสัมพันธ์ระหว่างค่า PWM กับค่าความเร็ว จาก จากสมการที่ 10 ถ้าทำการ ป้อน PWM จะสัมพันธ์กับ ω จะได้

ω_1 เป็นฟังก์ชันของ PWM₁

ω_2 เป็นฟังก์ชันของ PWM₂

ซึ่ง PWM₁ และ PWM₂ เป็นสัญญาณที่ระบบป้อนให้กับล้อซ้าย และล้อขวา ตามลำดับ

เนื่องจากการหมุนของล้อถูกควบคุมผ่านระบบไฟฟ้า จึงต้องใช้ความสัมพันธ์ของสัญญาณไฟฟ้าที่ ป้อนให้กับบอร์ดควบคุมกับอัตราเร็วเชิงมุมของล้อเพื่อนำไปใช้ควบคุมการหมุนของล้อตามที่ต้องการ

3.4 การออกแบบโครงสร้างแขนหุ่นยนต์และสร้างแขนหุ่นยนต์

ในการออกแบบกลไกของแขนหุ่นยนต์ ผู้วิจัยได้ใช้โปรแกรมช่วยออกแบบที่มีชื่อว่า Solid works จากนั้นนำมาขึ้นรูปโดยใช้เครื่องพิมพ์สามมิติ (3D Printer) ในการควบคุมการเคลื่อนที่ของ แขนหุ่นยนต์ ใช้เซอร์โวมอเตอร์ควบคุมตำแหน่งและความเร็วรอบ ผ่านไมโครคอนโทรลเลอร์อาศัยโน้ตกับบอร์ดขับเคลื่อนมอเตอร์ โดยเซอร์โวมอเตอร์ที่เลือกมาให้สอดคล้องกับงานวิจัยนี้ที่สร้างขึ้น เพื่อทำ แขนหุ่นยนต์ที่มีราคาประหยัด มอเตอร์ที่เลือกมาจึงมีราคาที่ไม่สูงมากนัก และมีประสิทธิภาพในการตอบสนองเป็นที่ยอมรับได้ โดยเซอร์โวมอเตอร์ที่เลือกใช้งานแสดงในภาพที่ 54

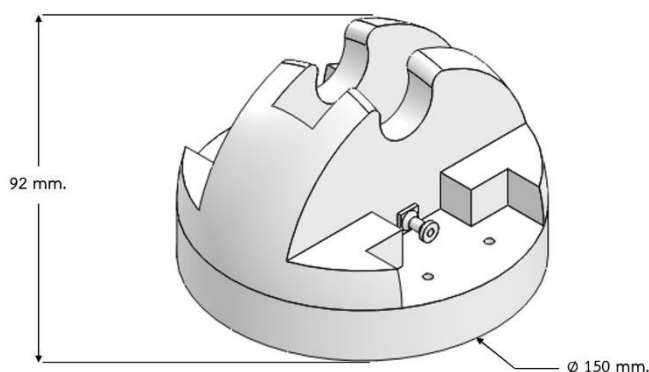


ภาพ 54 เซอร์โวมอเตอร์รุ่น TD8120MG

การออกแบบส่วนประกอบของแขนหุ่นยนต์

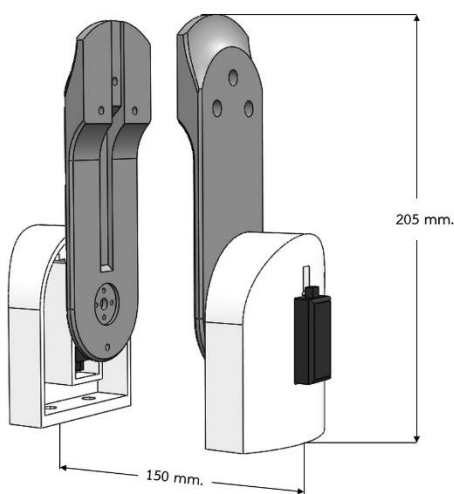
โครงสร้างแขนหุ่นยนต์ในงานวิจัยนี้ออกแบบให้สามารถมีพื้นที่ทำงานให้ครอบคลุม ซึ่ง รูปแบบของแขนหุ่นยนต์จะเป็นประเภท Articulated Arm Robot โดยการเคลื่อนที่ของแขนหุ่นยนต์

ในแต่ละข้อต่อเป็นแบบหมุน (Revolute joint) โดยแขนหุ่นยนต์ประกอบไปด้วย จุดหมุน 6 จุด ขนาดและโครงสร้างของแต่ละจุดหมุนและก้านต่อถูกออกแบบดังแสดงในภาพที่ 55 ถึงภาพที่ 58



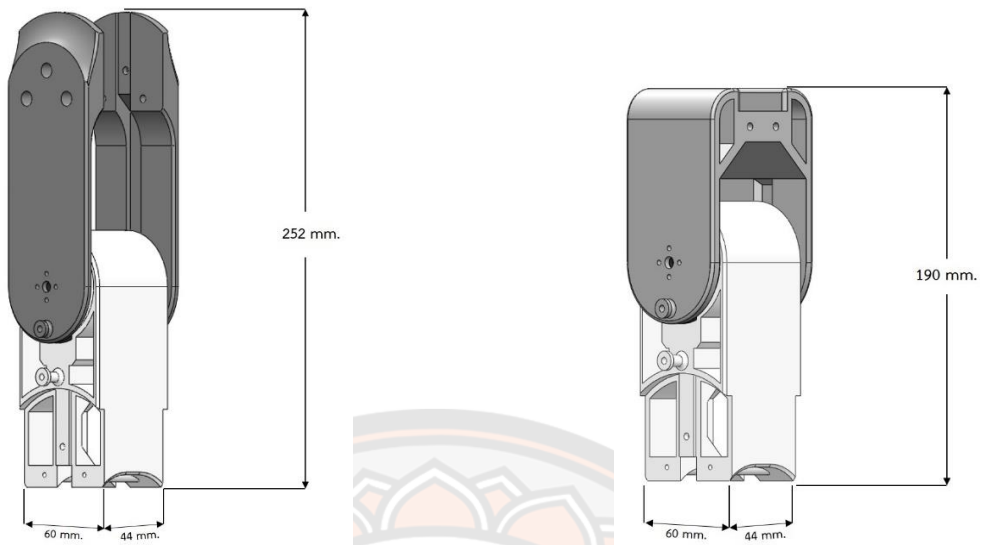
ภาพ 55 ส่วนฐานของแขนหุ่นยนต์

จากภาพที่ 55 เป็นภาพที่แสดงส่วนฐานของแขนหุ่นยนต์มีขนาดเส้นผ่าศูนย์กลางที่ 150 mm ส่วนใต้ฐานจะมีเซอร์โวมอเตอร์ 1 ตัว เพื่อทำหน้าที่ในการขับเคลื่อน เปรียบเสมือนส่วนเอวของแขนหุ่นยนต์



ภาพ 56 ส่วนหัวไหล่ของแขนหุ่นยนต์

จากภาพที่ 56 แสดงส่วนหัวไหล่ของแขนหุ่นยนต์ มีขนาดความยาวที่ 205 mm ซึ่งในแต่ละก้านต่อของแขนหุ่นยนต์ถูกออกแบบให้มีลักษณะรูปร่างที่ดูเป็นมิตรกับผู้ใช้งาน ต้องมีน้ำหนักที่เบาแต่ยังมีความแข็งแรงเพื่อป้องกันปัญหาในการรับน้ำหนักที่มากเกินไป โดยใช้เครื่องพิมพ์สามมิติในการขึ้นรูปพลาสติก ส่วนหัวไหล่ของแขนหุ่นยนต์ได้ติดตั้งมอเตอร์ในการขับเคลื่อนจำนวนสองตัว เนื่องจากจุดหมุนของส่วนหัวไหล่รับน้ำหนักของส่วนอื่นๆของแขนหุ่นยนต์ ทำให้รับน้ำหนักน้ำที่มาก

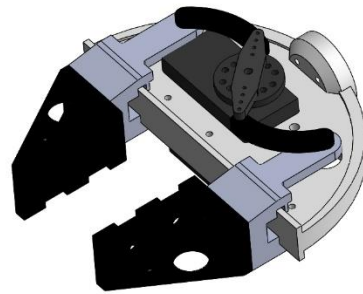


(ก) ส่วนข้อศอกของแขนหมุนยนต์

(ข) ส่วนข้อมือของแขนหมุนยนต์

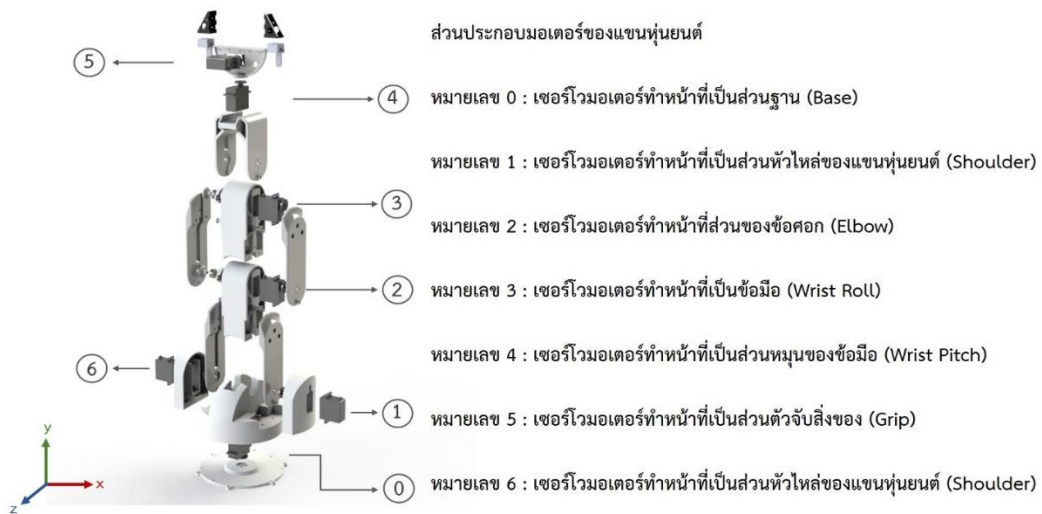
ภาพ 57 ส่วนข้อศอกของแขนหมุนยนต์

จากภาพที่ 57 แสดงส่วนข้อศอกและข้อมือของแขนหมุนยนต์ มีขนาดความยาวที่ 252 mm และ 190 mm ตามลำดับ ซึ่งทั้งสองส่วนนี้ได้ติดตั้งมอเตอร์ในการขับเคลื่อนจำนวนส่วนละหนึ่งตัว



ภาพ 58 ส่วนมือจับของแขนหมุนยนต์

จากภาพที่ 58 แสดงส่วนมือจับของแขนหมุนยนต์ โดยสามารถปรับความกว้างได้มากที่สุดเพื่อจะหยิบจับสิ่งของได้ที่ระยะ 50 mm



ภาพ 59 ส่วนประกอบมอเตอร์ของแขน

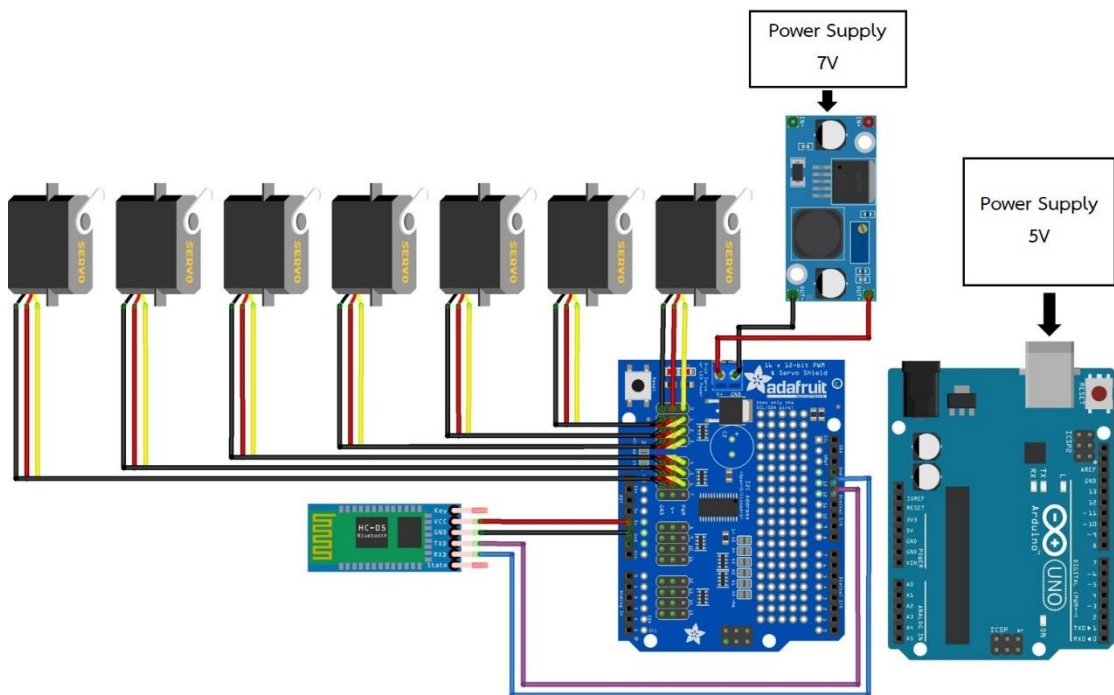
การออกแบบแขนหุ่นยนต์จะประกอบไปด้วยข้อต่อแบบ 6 ข้อต่อ ประกอบไปด้วยเซอร์โวมอเตอร์ทั้งหมด 7 ตัว แสดงดังภาพที่ 59 เซอร์โวมอเตอร์หมายเลข 0 ทำหน้าที่ส่งกำลังผ่านชุดเฟืองเพื่อให้แขนหุ่นยนต์สามารถหมุนรอบตัวได้ โดยแขนหุ่นยนต์นี้สามารถหมุนได้ 180 องศา ในแนวราบ (ระนาบ x-z), โดยเซอร์โวมอเตอร์หมายเลข 1 และหมายเลข 6 เป็นเซอร์โวมอเตอร์ที่รับน้ำหนักมากที่สุดซึ่งทำหน้าที่ขับเคลื่อนในส่วนของส่วนหัวไหล่ (Shoulder) ของแขนหุ่นยนต์ ในส่วนของเซอร์โวมอเตอร์หมายเลข 2,3,4 และ 5 ที่ติดตั้งในส่วนข้อศอก (Elbow), ข้อมือ (Wrist Roll), ส่วนหมุนของข้อมือ (Wrist Pitch) และส่วนจับสิ่งของ (Grip) ตามลำดับ โดยส่วนประกอบต่างๆของแขนหุ่นยนต์ถูกขึ้นรูปจากเครื่องพิมพ์โมเดลสามมิติ และเมื่อนำชิ้นส่วนจากการขึ้นรูป มาประกอบเข้าด้วยกันจะได้แขนหุ่นยนต์ดังแสดงในภาพที่ 60



ภาพ 60 แขนหุ่นยนต์

3.5 การออกแบบและสร้างวงจรควบคุมแขนหุ่นยนต์

ในส่วนของแขนหุ่นยนต์ให้ทำงานได้นั้นต้องมีอุปกรณ์ต่าง ๆ เช่น บอร์ดไมโครคอนโทรลเลอร์ที่ใช้ในการควบคุมติดต่อกับเซอร์โวมอเตอร์หรือเซนเซอร์ แบตเตอรี่สำหรับเป็นแหล่งจ่ายพลังงานให้กับอุปกรณ์เซอร์โวมอเตอร์ในการขับเคลื่อนจุดต่างๆ และอุปกรณ์ติดต่อสื่อสารกับแอปพลิเคชัน อุปกรณ์วงจรในแขนหุ่นยนต์ มีดังนี้



ภาพ 61 วงจรของแขนหุ่นยนต์

ระบบขับเคลื่อน และระบบควบคุมการเคลื่อนที่

ระบบควบคุมจะประกอบไปด้วย แอปพลิเคชันบนโทรศัพท์มือถือ และไมโครคอนโทรลเลอร์ การทำงานในส่วนของระบบควบคุมตามภาพที่ 61 เริ่มจากผู้ใช้งานสามารถสั่งงานแอปพลิเคชันที่ผู้วิจัยได้พัฒนาขึ้นบนโทรศัพท์มือถือ ส่งคำสั่งในการเคลื่อนที่ของแขนหุ่นยนต์ผ่านตัวรับสัญญาณบลูทูธ (หมายเลข 3) ไปยังไมโครคอนโทรลเลอร์จากนั้นไมโครคอนโทรลเลอร์จะส่งสัญญาณ PWM ไปที่ระบบขับเคลื่อน ซึ่งระบบขับเคลื่อนประกอบด้วย แบตเตอรี่ลิเธียม-โพลีเมอร์ (หมายเลข 8) และบอร์ดขับเคลื่อน โดยบอร์ดขับเคลื่อนทำหน้าที่ให้พลังงานกลแก่มอเตอร์รวมถึงส่งสัญญาณไฟฟ้าเพื่อให้เซอร์โวมอเตอร์เคลื่อนที่ไปยังตำแหน่งต่าง ๆ ตามคำสั่งที่ได้รับจากแอปพลิเคชันบนมือถือ ใน

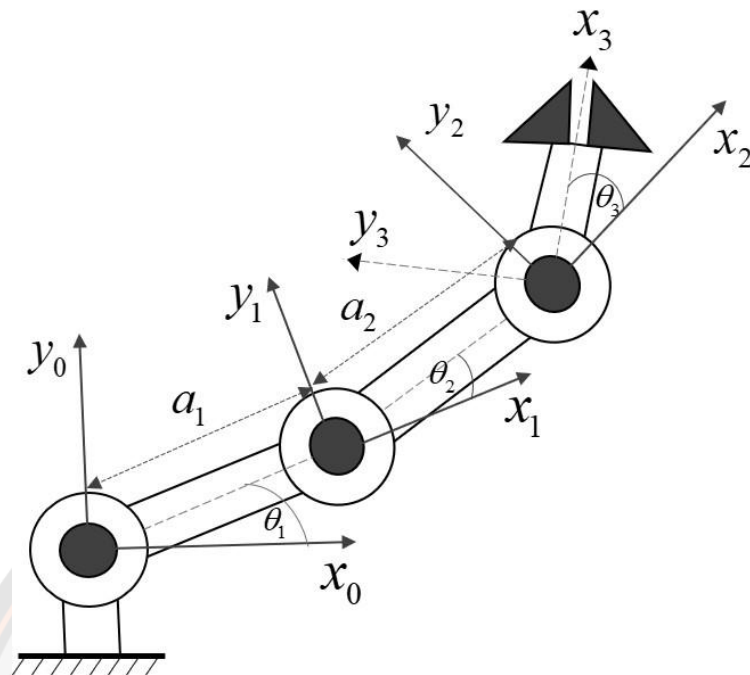
การตรวจสอบการเคลื่อนที่ของแขนหุ่นยนต์ทำได้โดยอาศัยสัญญาณภาพจากกล้อง ซึ่งถูกประมวลด้วยไมโครคอมพิวเตอร์ก่อนที่จะถูกส่งเข้าไปแสดงผลในแอปพลิเคชันบนมือถือด้วยสัญญาณ Wi-Fi โดยระบบขับเคลื่อน และระบบควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ แสดงในภาพที่ 62



ภาพ 62 ส่วนประกอบระบบควบคุมการเคลื่อนที่ของแขนหุ่นยนต์

3.6 โมเดลระบบควบคุมแขนหุ่นยนต์สำหรับการเคลื่อนที่เชิงเส้นแบบหลายแกน

การสั่งงานให้แขนหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งที่กำหนดไว้ สามารถควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ได้สองวิธี คือ การควบคุมโดยป้อนค่ามุมของข้อต่อ (Joint Angle) ให้กับแขนหุ่นยนต์โดยตรง และการควบคุมโดยการป้อนตำแหน่งของปลายแขน (End Effector) (Phanomrattanak & Pomsukviton, 2017) ซึ่งงานวิจัยนี้ทางผู้วิจัยได้พัฒนาระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-Axis Linear Motion Control -MALM Control) ซึ่งเป็นการควบคุมเซอร์โวมอเตอร์หลายตัวพร้อมกัน เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่แนวเส้นตรงตามแนวหน้า-หลัง (แกน $+x$, $-x$) และแนวขึ้น-ลง (แกน $+y$, $-y$) ตามหน้าจอในแอปพลิเคชัน โดยในงานวิจัยนี้ได้พัฒนาแขนหุ่นยนต์ที่มี joint angle หลักอยู่ 4 มุม คือ ส่วนของส่วนหัวไหล่ (Shoulder), ส่วนข้อศอก (Elbow), ข้อมือ (Wrist Roll) และส่วนของฐาน (Base) แต่การวิเคราะห์แขนหุ่นยนต์ด้วยจลนศาสตร์แบบไปข้างหน้าและแบบผกผันสำหรับงานวิจัยนี้ เป็นการวิเคราะห์ โดยสนใจ joint angle เพียง 3 มุม คือ ส่วนหัวไหล่, ส่วนข้อศอก และข้อมือ เพื่อควบคุมมือจับด้วยโมเดลในระนาบ 2 มิติ ให้อยู่ในตำแหน่งที่ต้องการอ้างอิงจากพิกัดแกนหัวไหล่ (x_0 , y_0) ดังแสดงในภาพที่ 63 โดยมุมที่ฐาน (Base) เป็นการหมุนแขนหุ่นยนต์รอบแกน y_0 เพื่อหันแขนหุ่นยนต์ไปตามทิศทางที่ต้องการ



ภาพ 63 แขนหุ่นยนต์ 3 ก้านต่อ ในระนาบ 2 มิติ

3.6.1 โมเดลระบบควบคุมแขนหุ่นยนต์สำหรับการเคลื่อนที่เชิงเส้นแบบหลายแกน

การควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-Axis Linear Motion Control - MALM Control) ในงานวิจัยนี้เป็นการควบคุมเซอร์โวมอเตอร์หลายตัว ให้สามารถทำมุมสัมพันธ์กัน เพื่อให้มือจับเคลื่อนที่เชิงเส้นตรงตามแนวแกน x และแกน y ได้ โดยจากภาพที่ 61 กำหนดการเคลื่อนที่เป็นเส้นตรงในแนวหน้า-หลัง $(+x_0, -x_0)$ และแนวขึ้น-ลง $(+y_0, -y_0)$ ในการหาค่ามุมแต่ละข้อต่อทำได้โดยการกำหนดค่าตัวแปรต่าง ๆ ของแขนหุ่นยนต์ด้วย ดีเอช พารามิเตอร์ (Denavit-Hartenberg parameters: DH) ซึ่งเป็นวิธีการหนึ่งในการแก้ปัญหาสายโซ่จลนศาสตร์อย่างเป็นระบบ เพื่อใช้ในการวิเคราะห์พื้นที่ทำงานของแขนหุ่นยนต์ (Zar & Ei, 2017) โดยคำนวณด้วยหลักการจลนศาสตร์แบบไปข้างหน้าและแบบผกผัน ซึ่งการควบคุมแขนหุ่นยนต์ทำได้โดยกำหนดมุมมองศาให้กับเซอร์โวมอเตอร์แต่ละตัวด้วยสัญญาณทางไฟฟ้าในรูปแบบ Pulse Width Modulation (PWM) ซึ่งนิยมใช้ในการควบคุมตำแหน่งของเซอร์โวมอเตอร์

3.6.1.1 กำหนด ค่า D-H parameters

ในงานวิจัยนี้สนใจแขนหุ่นยนต์ 3 ก้านต่อ ในระนาบ 2 มิติ $(x-y)$ ในการหาพื้นที่ทำงานของแขนหุ่นยนต์ ทำได้โดยหาจากรางความสัมพันธ์ที่มีชื่อว่า Denavit-Hartenberg (D-H) ซึ่งเป็นรูปแบบมาตรฐานการกำหนดค่าตัวแปรต่าง ๆ ของแขนหุ่นยนต์ เพื่อใช้วิเคราะห์การเคลื่อนที่ของแขน

หุ่นยนต์ โดยมีตัวแปร คือ มุมของแต่ละข้อต่อ (θ_i), ระยะห่างระหว่างข้อต่อ (d_i), ความยาวของแต่ละข้อต่อ (a_i) และมุมบิดของแต่ละข้อต่อ (α_i) (Tuchinda, 2016) เพื่อหาความสัมพันธ์แต่ละข้อต่อของแขนหุ่นยนต์ นำค่ามาแทนลงในรูปแบบเมทริกซ์ แสดงในตารางที่ 3

ตาราง 3 การวิเคราะห์แขนหุ่นยนต์ด้วย D-H parameter

Link	θ_i	d_i	a_i	α_i
1	θ_1	0	a_1	0
2	θ_2	0	a_2	0
3	θ_3	0	0	0

จลนศาสตร์แบบไปข้างหน้า

จลนศาสตร์แบบไปข้างหน้าเป็นการหาค่าพิกัดและทิศทาง (ในรูปการแปลงเอกพันธ์) ของมือจับเมื่อทราบตำแหน่งและมุมของข้อต่อทั้งหมด จากการคำนวณโดยใช้ค่า D-H parameter ดังตารางที่ 3 ด้วยวิธีการหาจลนศาสตร์แบบไปข้างหน้า จะได้

$${}^0T_3 = \begin{pmatrix} c_{123} & -s_{123} & 0 & a_1c_1 + a_2c_{12} \\ s_{123} & c_{123} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (11)$$

แต่สำหรับหุ่นยนต์ในระนาบสองมิตินี้ จัดให้อยู่ในรูป

$${}^0T_3 = \begin{pmatrix} c_\phi & -s_\phi & 0 & p_x \\ s_\phi & c_\phi & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

โดย 0T_3 คือ Transformation Matrix ของพิกัดและจุดอ้างอิงจากเฟรมอ้างอิงของแกนที่ 3 เทียบกับไปยังเฟรมอ้างอิงของแกนที่ 0,

p_x และ p_y คือ พิกัดที่ตำแหน่งข้อต่อที่ 3 ของแขนหุ่นยนต์ เทียบกับแกนอ้างอิง 0

จลนศาสตร์แบบผกผัน

จลนศาสตร์แบบผกผันเป็นการหาค่าตำแหน่งและมุมของข้อต่อเมื่อทราบค่าเมทริกซ์เอกพันธ์ของตัวทำงานส่วนปลายเทียบกับเฟรมฐาน

โดย ϕ คือทิศทางของก้านต่อ 3 เทียบกับพิกัดอ้างอิง 0 จากการ เปรียบเทียบสมการ (11) และ (12) จะได้ว่า

$$c_\phi = c_{123} = \cos(\theta_1 + \theta_2 + \theta_3) \quad (13)$$

$$s_\phi = s_{123} = \sin(\theta_1 + \theta_2 + \theta_3) \quad (14)$$

$$p_x = a_1 c_1 + a_2 c_{12} \quad (15)$$

$$p_y = a_1 s_1 + a_2 s_{12} \quad (16)$$

โดย $c_\phi = \cos \phi$, $s_\phi = \sin \phi$

$$c_1 = \cos \theta_1, \quad c_{12} = \cos(\theta_1 + \theta_2)$$

$$s_1 = \sin \theta_1, \quad s_{12} = \sin(\theta_1 + \theta_2)$$

จากสมการ (13) ถึง (16) ใช้วิธีการทางพีชคณิตเพื่อหาค่าของมุมข้อต่อ จากสมการ (15) และ (16) นำไปยกกำลังสองและรวมเข้าด้วยกันเป็น

$$p_x^2 + p_y^2 = a_1^2 + a_2^2 + 2a_1 a_2 c_2 \quad (17)$$

จัดรูปเพื่อหา c_2

$$c_2 = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad (18)$$

หาค่า s_2 จากสมการความสัมพันธ์ $s_2^2 + c_2^2 = 1$ จะได้คำตอบที่เป็นไปได้ 2 คำตอบคือ

$$s_2 = \pm \sqrt{1 - c_2^2} \quad (19)$$

ในกรณีนี้ กำหนดให้ s_2 มีค่าเป็นลบ เพื่อให้แขนหุ่นยนต์มีทิศทางที่เหมาะสมตามต้องการ

หาค่า θ_2 จากความสัมพันธ์

$$\theta_2 = \arctan 2(s_2, c_2) \quad (20)$$

ต่อไปหาค่า θ_1 จากการแก้สมการ (15),(16) โดยการใช้เอกลักษณ์ตรีโกณ เขียนสมการใหม่ในรูป

$$p_x = k_1 c_1 - k_2 s_1 \quad (21)$$

$$p_y = k_1 s_1 + k_2 c_1 \quad (22)$$

โดย

$$k_1 = a_1 + a_2 c_2 \quad (23)$$

$$k_2 = a_2 s_2 \quad (24)$$

วิธีการแก้สมการในรูปนี้คือเปลี่ยนตัวแปร สมมติให้

$$r = +\sqrt{k_1^2 + k_2^2} \quad (25)$$

$$\gamma = \arctan 2(k_2, k_1) \quad (26)$$

ดังนั้น

$$k_1 = r \cos \gamma \quad (27)$$

$$k_2 = r \sin \gamma \quad (28)$$

$$\frac{p_x}{r} = \cos \gamma \cos \theta_1 - \sin \gamma \sin \theta_1 \quad (29)$$

$$\frac{p_y}{r} = \cos \gamma \sin \theta_1 + \sin \gamma \cos \theta_1 \quad (30)$$

จัดรูปใหม่โดยใช้เอกลักษณ์ตรีโกณ

$$\cos(\gamma + \theta_1) = \frac{p_x}{r} \quad (31)$$

$$\sin(\gamma + \theta_1) = \frac{p_y}{r} \quad (32)$$

เมื่อใช้ฟังก์ชัน $\arctan 2$ จะได้ว่า

$$\gamma + \theta_1 = \arctan 2\left(\frac{p_y}{r}, \frac{p_x}{r}\right) = \arctan 2(p_y, p_x) \quad (33)$$

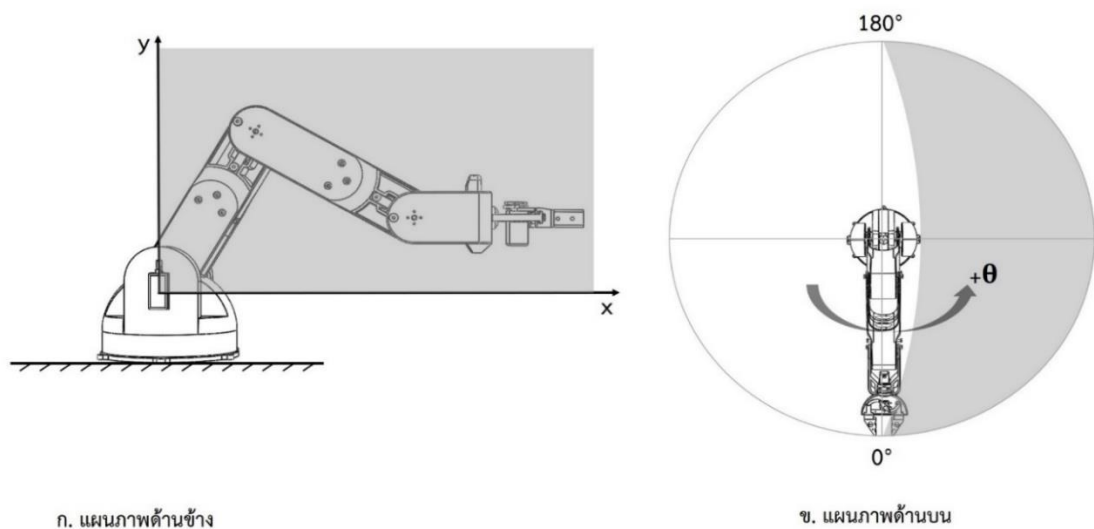
ดังนั้นค่ามุมของ θ_1 มีค่าเท่ากับ

$$\theta_1 = \arctan 2(p_y, p_x) - \arctan 2(k_2, k_1) \quad (34)$$

สุดท้ายหาคำตอบ θ_3 จากความสัมพันธ์ $\theta_1 + \theta_2 + \theta_3 = \phi$ จะได้

$$\theta_3 = \phi - \theta_1 - \theta_2 \quad (35)$$

ในการออกแบบซอฟต์แวร์การทำงานของแขนหุ่นยนต์ ผู้วิจัยได้ใช้โปรแกรม MATLAB ในการคำนวณ เพื่อหาพื้นที่ทำงานหรือขอบเขตของแขนหุ่นยนต์ และใช้โปรแกรม Arduino IDE สำหรับพัฒนาโปรแกรมลงในบอร์ดไมโครคอนโทรลเลอร์ เพื่อคำนวณจลนศาสตร์แบบไปข้างหน้าและจลนศาสตร์ผกผัน และรับส่งสัญญาณเซนเซอร์เพื่อใช้ในการควบคุมแขนหุ่นยนต์



ก. แผนภาพด้านข้าง

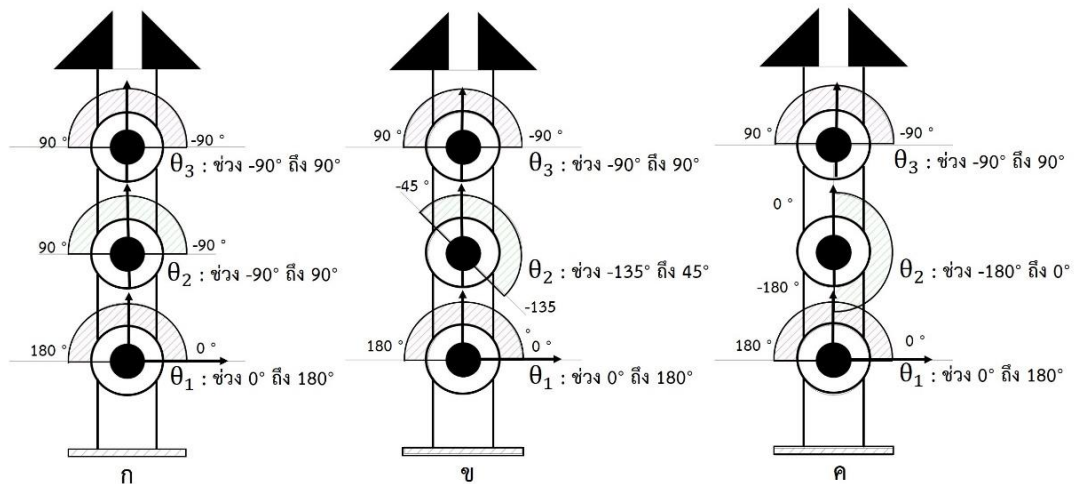
ข. แผนภาพด้านบน

ภาพ 64 พื้นที่ทำงานของแขนหุ่นยนต์

3.6.2 การหาพื้นที่ทำงานของแขนหุ่นยนต์

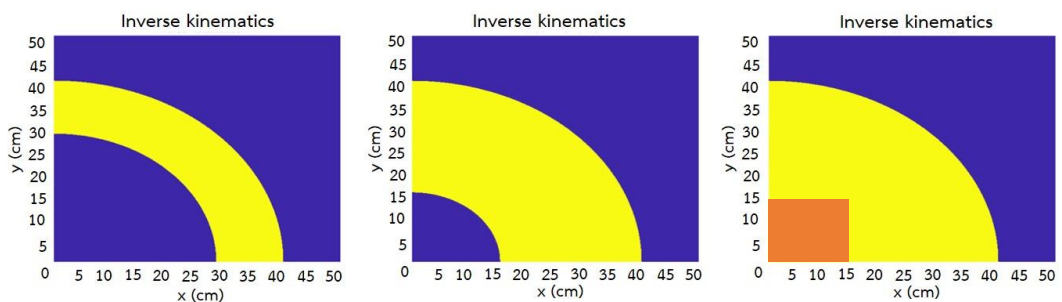
พื้นที่ทำงาน (workspace) จากภาพที่ 64 คือพื้นที่ ที่ส่วนปลายของแขนหุ่นยนต์เข้าถึงได้ การหาพื้นที่ทำงานของแขนหุ่นยนต์ในระนาบ x-z หาได้จาก ครึ่งหนึ่งของพื้นที่วงกลม ส่วนระนาบ x-y หาได้โดยการนำค่า D-H Parameters จากตารางที่ 3 ไปใช้ร่วมกับสมการจลนศาสตร์แบบไปข้างหน้า และจลนศาสตร์แบบผกผัน ซึ่งแขนหุ่นยนต์ในงานวิจัยนี้ประกอบไปด้วย ข้อต่อ 3 ข้อต่อ ประกอบไปด้วยมุมในการหมุนของข้อต่อ (θ_i) มีดังนี้ θ_1 คือมุมในส่วนของหัวไหล่ของแขนหุ่นยนต์ (Shoulder), θ_2 คือมุมในส่วนของข้อศอก (Elbow) และ θ_3 คือมุมในส่วนข้อมือ (Wrist Roll) เนื่องจากเซอร์โวมอเตอร์สามารถหมุนได้แค่ 180 องศา θ_2 จึงส่งผลกระทบต่อขอบเขตการทำงานของแขนหุ่นยนต์มากที่สุด ดังนั้นผู้วิจัยจึงทำการเปรียบเทียบการตั้งค่าของ θ_2 ในช่วง -90 ถึง 90

องศา, -135 ถึง 45 องศา และ -180 ถึง 0 องศา โดยการตั้งค่าช่วงมุมของเซอร์โวมอเตอร์ แสดงดังภาพที่ 65



ภาพ 65 ตั้งค่าทิศทางของเซอร์โวมอเตอร์

จากการคำนวณพื้นที่ทำงานของแขนหุ่นยนต์ ตามการกำหนดช่วงมุมของเซอร์โวมอเตอร์ รูปแบบต่าง ๆ โดยใช้โปรแกรม MATLAB แสดงดังภาพที่ 66 โดยจากผลการคำนวณพบว่าหากตั้งค่าเซอร์โวมุม θ_2 อยู่ในช่วง -90 ถึง 90 องศา พบว่าพื้นที่ทำงานของแขนหุ่นยนต์มีค่าไม่เหมาะสมในการใช้งาน ในการตั้งค่ามุม θ_2 ให้เป็น -135 ถึง 45 องศา มีพื้นที่ทำงานได้มากขึ้น เมื่อเทียบกับการตั้งค่ามุมในช่วง -90 ถึง 90 องศา และแขนหุ่นยนต์สามารถเคลื่อนที่ได้เหมาะสมกับการใช้งานจริง ในส่วนของการตั้งค่ามุม θ_2 ให้เป็น -180 ถึง 0 องศา มีพื้นที่ทำงานได้มากที่สุด แต่ไม่เหมาะสมในการใช้งานจริง เพราะแขนหุ่นยนต์ไม่สามารถไปยังจุดอ้างอิงได้ซึ่งเป็นบริเวณโครงสร้างของฐาน ดังนั้น การตั้งค่ามุม θ_2 ที่เหมาะสมอยู่ในช่วง -135 ถึง 45 องศา

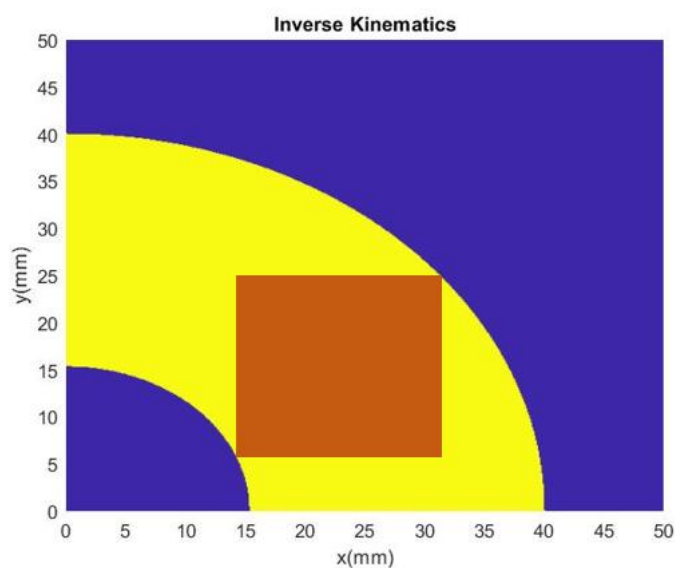


ก. กำหนดมุม θ_2 ช่วง -90 ถึง 90 องศา ข. กำหนดมุม θ_2 ช่วง -135 ถึง 45 องศา ค. กำหนดมุม θ_2 ช่วง -180 ถึง 0 องศา

ภาพ 66 แสดงพื้นที่ทำงานแขนหุ่นยนต์จากการกำหนดมุม

3.6.3 การกำหนดพื้นที่ทำงานของแขนหุ่นยนต์

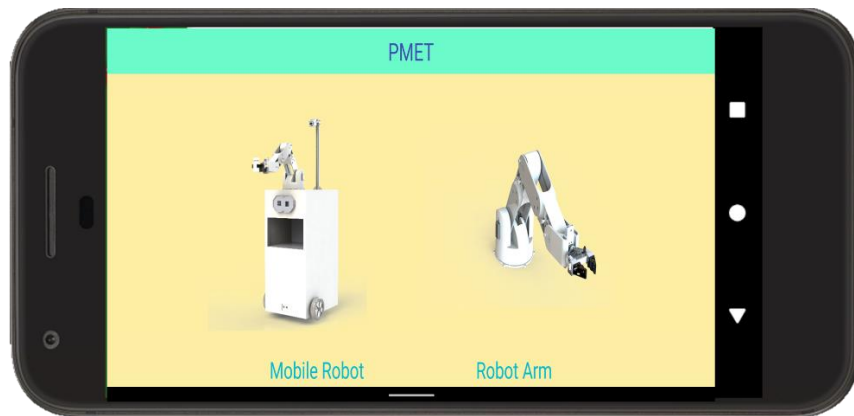
จากการใช้โปรแกรม MATLAB จำลองพื้นที่การทำงานของแขนหุ่นยนต์ แสดงดังภาพที่ 67 จะได้การกำหนดมุมของเซอร์โวมอเตอร์ที่เหมาะสม จากนั้นกำหนดขอบเขตการทำงานของแขนหุ่นยนต์ โดยกำหนดจุดน้อยที่สุดไปมากที่สุดของแนวนราบ (Horizontal) ที่ 15 เซนติเมตร ถึง 32 เซนติเมตร และแนวตั้ง (Vertical) ที่ 7 เซนติเมตร ถึง 24 เซนติเมตร จากนั้นนำข้อมูลที่ได้ไปกำหนดค่าในโปรแกรม Arduino IDE เพื่อกำหนดระยะทำงานของปลายแขนหุ่นยนต์ คือ 17x17 เซนติเมตร



ภาพ 67 กำหนดขอบเขตพื้นที่การทำงานของแขนหุ่นยนต์

3.7 การออกแบบและสร้างแอปพลิเคชันบนสมาร์ตโฟนเพื่อใช้ในการติดต่อและควบคุมหุ่นยนต์

งานวิจัยนี้เป็นการควบคุมหุ่นยนต์ โดยแอปพลิเคชันบนสมาร์ตโฟนผ่านสัญญาณไร้สาย พร้อมกับแสดงภาพบนหน้าจอตามเวลาจริง ซึ่งหน้าจอของแอปพลิเคชันจะประกอบไปด้วยหลายส่วน โดยแบ่งการควบคุมเป็นส่วนของหุ่นยนต์เคลื่อนที่และแขนหุ่นยนต์ เมื่อออกแบบโครงสร้างและวงจรส่วนของฮาร์ดแวร์เสร็จแล้ว ต่อมาเป็นส่วนการออกแบบและสร้างแอปพลิเคชันในการควบคุมหุ่นยนต์ และแขนหุ่นยนต์ โดยใช้โปรแกรม Kodular ซึ่งเป็นโปรแกรมสำหรับเขียนแอปพลิเคชันสำหรับระบบปฏิบัติการแอนดรอยด์ การออกแบบ UI ของแอปพลิเคชันจะมีส่วนควบคุมการเคลื่อนที่ของหุ่นยนต์ และส่วนการควบคุมแขนหุ่นยนต์ แสดงดังภาพที่ 68



ภาพ 68 หน้าแรกของแอปพลิเคชัน เพื่อเลือกส่วนควบคุม

ในส่วนหน้าจอของการควบคุมแขนหุ่นยนต์ จะประกอบไปด้วยส่วนที่ 1 เป็นส่วนเลือกสัญญาณเชื่อมต่อกับอุปกรณ์, ส่วนที่ 2 เป็นหน้าจอแสดงผลจากกล้องตามเวลาจริง ซึ่งแสดงผลจากกล้องที่ติดกับส่วนปลายแขนหุ่นยนต์, ส่วนที่ 3 เป็นหน้าจอแสดงผลจากกล้องตัวที่ 2 จะแสดงภาพมุมกว้างที่เห็นภาพรวมของแขนหุ่นยนต์, ส่วนที่ 4 เป็นปุ่มการควบคุมแขนหุ่นยนต์แบบ MALM-Control เข้ามาช่วยให้ผู้ป่วยสามารถควบคุมแขนหุ่นยนต์ได้ให้ง่ายขึ้น, ส่วนที่ 5 เป็นการควบคุมทีละข้อต่อแบบอิสระ และส่วนที่ 6 เป็นปุ่มฟังก์ชันสำหรับการช่วยในการตั้งค่าตำแหน่งเริ่มต้นให้พร้อมใช้งาน และปุ่มฟังก์ชันเมื่อผู้ป่วยทำการจับสิ่งของได้แล้ว แขนหุ่นยนต์จะนำสิ่งของไปวางไว้ที่อีกตำแหน่ง โดยผู้ป่วยไม่ต้องบังคับเอง ส่วนประกอบของหน้าจอส่วนควบคุมแขนหุ่นยนต์ของแอปพลิเคชัน แสดงดังภาพที่

69

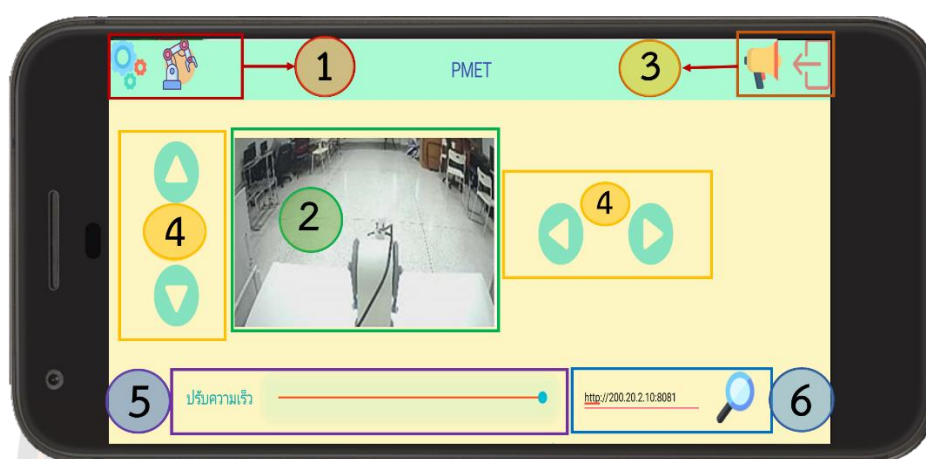


ภาพ 69 หน้าจอส่วนควบคุมแขนหุ่นยนต์ของแอปพลิเคชัน

ต่อมาเป็นส่วนหน้าจอกการควบคุมการเคลื่อนที่ของหุ่นยนต์ ส่วนประกอบต่าง ๆ จะคล้ายกับหน้าจอควบคุมแขนหุ่นยนต์ ประกอบไปด้วย ส่วนที่ 1 เป็นส่วนเลือกสัญญาณเชื่อมต่อกับอุปกรณ์ และปุ่มสำหรับกดไปที่หน้าจอสำหรับควบคุมแขนหุ่นยนต์, ส่วนที่ 2 เป็นหน้าจอแสดงผลจากกล้อง

ตามเวลาจริง มีมุมมองอยู่ด้านบน ทำให้เห็นภาพรวมรอบข้างของหุ่นยนต์, ส่วนที่ 3 เป็นปุ่มกดสำหรับส่งสัญญาณเสียง และปุ่มออกหน้าจอ, ส่วนที่ 4 เป็นปุ่มการควบคุมการเคลื่อนที่ของหุ่นยนต์, ส่วนที่ 5 เป็นการปรับความเร็วของหุ่นยนต์ และส่วนที่ 6 เป็นส่วนสำหรับการค้นหาเลข IP ของสัญญาณภาพ ส่วนประกอบของหน้าจอส่วนควบคุมหุ่นยนต์เคลื่อนที่ของแอปพลิเคชัน แสดงดังภาพที่

70



ภาพ 70 หน้าจอส่วนควบคุมหุ่นยนต์เคลื่อนที่ของแอปพลิเคชัน

3.8 กระบวนการทำงานของหุ่นยนต์ โดยการควบคุมไร้สายผ่านแอปพลิเคชันบนสมาร์ตโฟน

การทำงานของหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง สามารถควบคุมไร้สายโดยการสั่งงานผ่านแอปพลิเคชันบนสมาร์ตโฟน ซึ่งแบ่งการควบคุมออกเป็น 2 ส่วน คือ การควบคุมส่วนของหุ่นยนต์เคลื่อนที่ และการควบคุมส่วนของแขนหุ่นยนต์ แสดงดังภาพที่ 71 โดยผู้ใช้งานต้องทำการเลือกการควบคุมอย่างใดอย่างหนึ่ง เช่น ผู้ใช้ต้องควบคุมส่วนของหุ่นยนต์ให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการ จากนั้นเปลี่ยนมาควบคุมส่วนของแขนหุ่นยนต์ให้หยิบจับสิ่งของที่ต้องการ เป็นต้น โดยแอปพลิเคชันจะแบ่งการควบคุมเป็นสองส่วนเช่นกัน คือ หน้าจอส่วนของการควบคุมหุ่นยนต์เคลื่อนที่ และหน้าจอส่วนของการควบคุมแขนหุ่นยนต์ ซึ่งภายในหน้าจอของแอปพลิเคชันของทั้งสองส่วนจะมีจุดที่คล้ายกัน แต่จะมีบางส่วนที่แตกต่างกันออกไป เพื่อให้เหมาะกับการควบคุมกับลักษณะการทำงานของหุ่นยนต์ จะประกอบไปด้วยส่วนต่าง ๆ เช่น ส่วนของการเลือกสัญญาณเชื่อมต่อกับอุปกรณ์ หน้าจอแสดงผลจากกล้อง ส่วนของปุ่มบังคับเพื่อใช้ในการควบคุมหุ่นยนต์ เป็นต้น



ภาพ 71 การควบคุมหุ่นยนต์ผ่านแอปพลิเคชันบนสมาร์ทโฟน

3.81. วิธีการใช้งานแอปพลิเคชันสำหรับการควบคุมหุ่นยนต์เคลื่อนที่

แอปพลิเคชันสำหรับการควบคุมหุ่นยนต์ จะมี 2 หน้าจอ คือ หน้าจอสำหรับควบคุมหุ่นยนต์เคลื่อนที่ และหน้าจอสำหรับควบคุมแขนหุ่นยนต์ โดยการควบคุมทั้งสองส่วนจะมีปุ่มควบคุมการเคลื่อนที่ในลักษณะต่าง ๆ ในงานวิจัยนี้ผู้วิจัยได้ออกแบบปุ่มสำหรับควบคุมหุ่นยนต์ มีลักษณะคล้ายกับ joystick เพื่อให้บังคับได้ง่าย เข้าใจการใช้งานได้รวดเร็ว โดยปุ่มควบคุมการเคลื่อนที่ของหุ่นยนต์ แสดงดังรูปภาพที่ 72 ถึงภาพที่ 79 จะเป็นการแสดงรายละเอียดการทำงานของแต่ละปุ่ม พร้อมกับแสดงลักษณะการเคลื่อนที่ของหุ่นยนต์



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 72 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มเดินหน้า

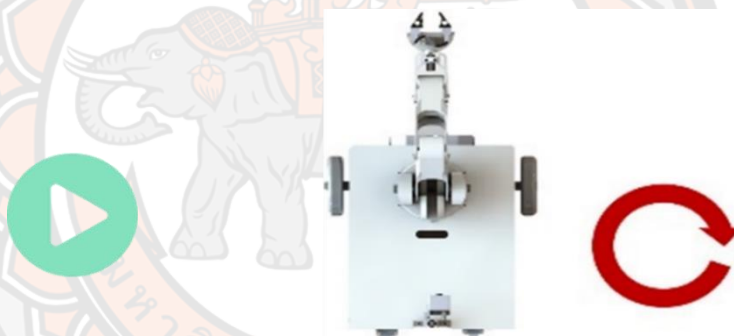
จากที่ 72 เมื่อกดปุ่มเดินหน้า หุ่นยนต์จะเคลื่อนที่ไปข้างเดิน โดยที่ล้อซ้ายกับล้อขวา จะมีความเร็วในการเคลื่อนที่ที่เท่ากัน



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 73 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มถอยหลัง

จากที่ 73 เมื่อกดปุ่มถอยหลัง หุ่นยนต์จะเคลื่อนที่ไปข้างหลัง โดยที่ล้อซ้ายกับล้อขวา เคลื่อนที่ไปข้างหลังในความเร็วที่เท่ากัน



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 74 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนขวา

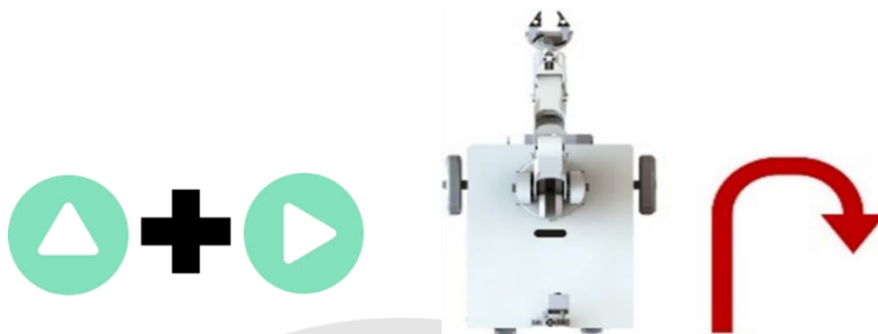
จากที่ 74 เมื่อกดปุ่มหมุนขวา หุ่นยนต์จะทำการหมุนตัวไปข้างขวา (หมุนล้อซ้ายไปข้างหน้า และล้อขวาหมุนกลับหลังด้วยความเร็วเท่ากัน)



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 75 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนซ้าย

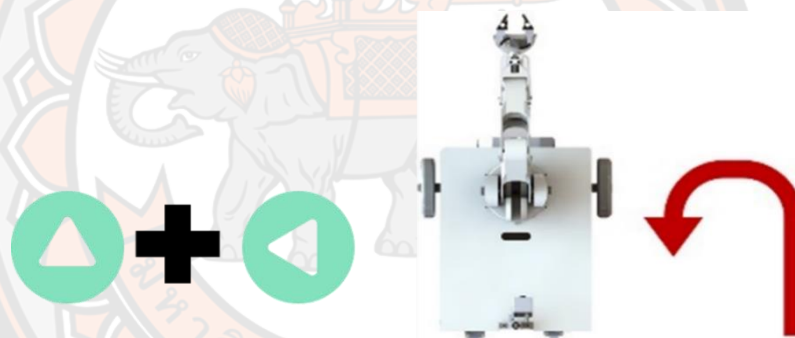
จากที่ 75 เมื่อกดปุ่มหมุนซ้าย หุ่นยนต์จะทำการหมุนตัวไปข้างซ้าย (หมุนล้อขวาไปข้างหน้า และล้อซ้ายหมุนกลับหลังด้วยความเร็วเท่ากัน)



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 76 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนขวาพร้อมกัน

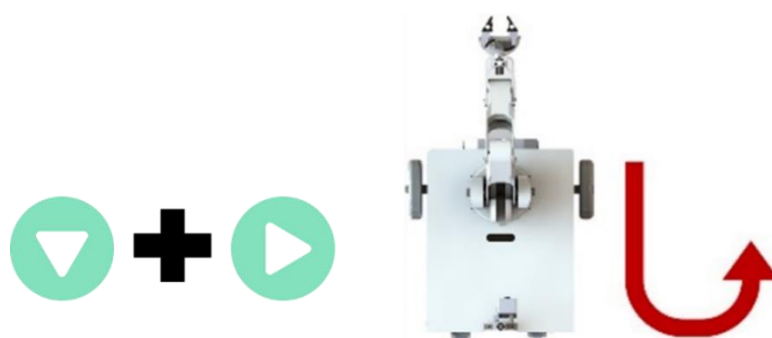
จากที่ 76 เมื่อกดปุ่มเดินหน้ากับปุ่มหมุนขวาพร้อมกัน หุ่นยนต์จะทำการเลี้ยวขวาแบบตีวงโค้ง (หมุนล้อซ้ายกับล้อขวาไปข้างหน้า ในขณะที่ความเร็วล้อซ้ายมากกว่าล้อขวา)



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 77 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มหมุนซ้ายพร้อมกัน

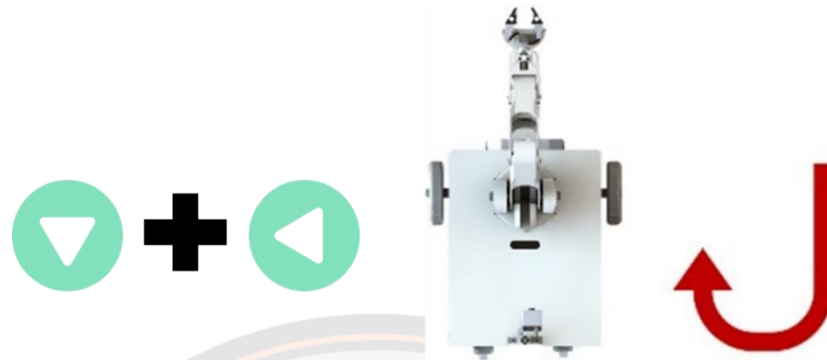
จากที่ 77 เมื่อกดปุ่มเดินหน้ากับปุ่มหมุนซ้ายพร้อมกัน หุ่นยนต์จะทำการเลี้ยวซ้ายแบบตีวงโค้ง (หมุนล้อซ้ายกับล้อขวาไปข้างหน้า ในขณะที่ความเร็วล้อขวามากกว่าล้อซ้าย)



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 78 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มถอยหลังกับปุ่มหมุนขวาพร้อมกัน

จากที่ 78 เมื่อกดปุ่มถอยหลังกับปุ่มหมุนขวาพร้อมกัน หุ่นยนต์จะถอยหลังแล้วเลี้ยวขวาแบบตีวงโค้ง (หมุนล้อซ้ายกับล้อขวาไปข้างหลัง ในขณะที่ความเร็วล้อซ้ายมากกว่าล้อขวา)



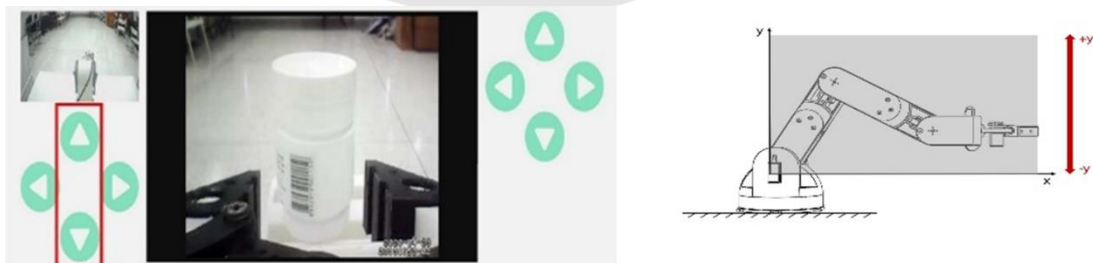
(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 79 ลักษณะการเคลื่อนที่ของหุ่นยนต์ เมื่อกดปุ่มถอยหลังกับปุ่มหมุนซ้ายพร้อมกัน

จากที่ 79 เมื่อกดปุ่มถอยหลังกับปุ่มหมุนซ้ายพร้อมกัน หุ่นยนต์จะถอยหลังแล้วเลี้ยวซ้ายแบบตีวงโค้ง (หมุนล้อซ้ายกับล้อขวาไปข้างหลัง ในขณะที่ความเร็วล้อขวามากกว่าล้อซ้าย)

3.8.2 วิธีการใช้งานแอปพลิเคชันสำหรับการควบคุมแขนหุ่นยนต์

ต่อมาเป็นปุ่มควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ซึ่งส่วนหน้าจอของการควบคุมแขนหุ่นยนต์ จะมีปุ่มที่เยอะกว่าหน้าจอของการควบคุมหุ่นยนต์เคลื่อนที่ เพราะจะมีปุ่มสำหรับควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-Axis Linear Motion Control -MALM Control) เป็นการควบคุมมอเตอร์หลายตัวพร้อมกัน เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่แบบเส้นตรง เช่น แนวหน้า-หลัง (ตามแนวแกน $+x$, $-x$) และแนวขึ้น-ลง (ตามแนวแกน $+y$, $-y$) และปุ่มสำหรับควบคุมเซอร์โวมอเตอร์แบบทีละข้อต่ออิสระ รายละเอียดการทำงานของแต่ละปุ่มแสดงดังภาพที่ 80 ถึงภาพที่ 89

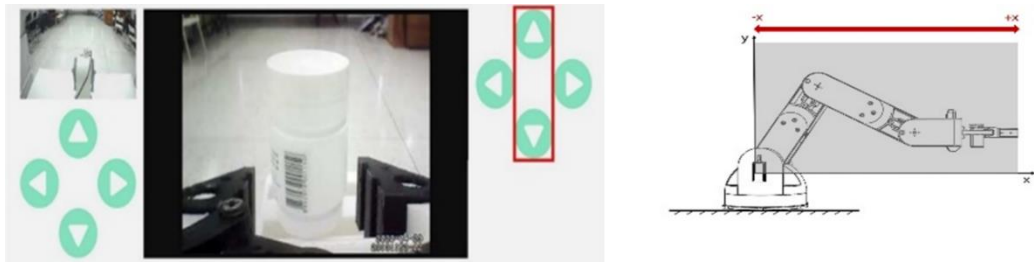


(ก) การกดปุ่มสำหรับการควบคุม

(ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 80 ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ โดยการควบคุมแบบ MALM Control ในแนว
ขึ้น-ลง

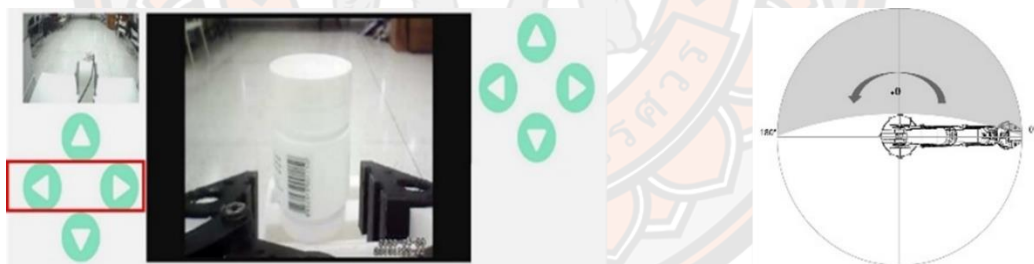
จากภาพที่ 80 เป็นการควบคุมแบบ MARM Control เมื่อกดปุ่มในกรอบสีแดงที่แสดงดังภาพ โดยลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ จะเคลื่อนที่ในแนวแกน y หรือแนวขึ้น-ลง ตามหน้าจอแอปพลิเคชัน



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 81 ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ โดยการควบคุมแบบ MARM Control ในแนวหน้า-หลัง

จากภาพที่ 81 เป็นการควบคุมแบบ MARM Control เมื่อกดปุ่มในกรอบสีแดงที่แสดงดังภาพ โดยลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ จะเคลื่อนที่ในแนวแกน x หรือแนวหน้า-หลัง ตามหน้าจอแอปพลิเคชัน



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 82 ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ โดยการควบคุมส่วนฐานของแขนหุ่นยนต์

จากภาพที่ 82 เป็น การควบคุมทีละข้อต่อแบบอิสระ โดยเป็นส่วนการควบคุมฐานของแขนหุ่นยนต์เมื่อกดปุ่มในกรอบสีแดงที่แสดงดังภาพ ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ จะเคลื่อนที่ในแนวแกน z หรือแนวซ้าย-ขวา ตามหน้าจอแอปพลิเคชัน จากขอบเขตของเซอร์โวมอเตอร์ที่ทำมุมได้ 180 องศา จึงเป็นข้อจำกัดอย่างหนึ่งในการเคลื่อนที่ในระนาบนี้

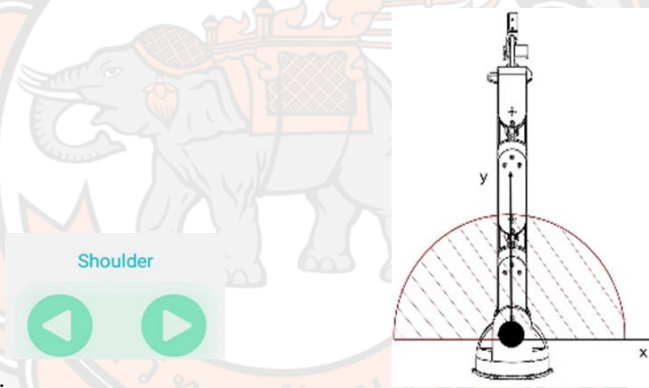


(ก) การกดปุ่มสำหรับการควบคุม

(ข) การเคลื่อนที่ของข้อมือ

ภาพ 83 ลักษณะการเคลื่อนที่ของแขนข้อมือ โดยการควบคุมส่วนการหมุนข้อมือของแขนข้อมือ

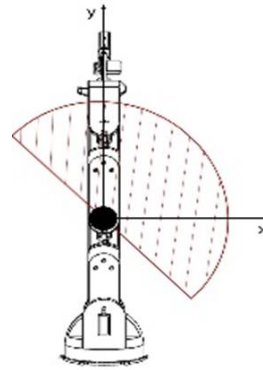
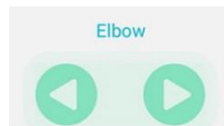
จากภาพที่ 83 เป็น การควบคุมที่ละเอียดแบบอิสระ โดยเป็นส่วนการควบคุมเซอร์โวมอเตอร์ ส่วนการหมุนข้อมือของแขนข้อมือ เมื่อกดปุ่มในกรอบสีแดงที่แสดงดังภาพ ลักษณะการเคลื่อนที่ของแขนข้อมือ จะเคลื่อนที่ระนาบ yz ตามหน้าจอแอปพลิเคชัน



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของข้อมือ

ภาพ 84 ลักษณะการเคลื่อนที่ของแขนข้อมือ โดยการควบคุมส่วนหัวไหล่ของแขนข้อมือ

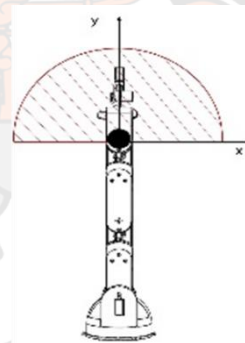
จากภาพที่ 84 เป็น การควบคุมที่ละเอียดแบบอิสระ โดยเป็นส่วนการควบคุมเซอร์โวมอเตอร์หัวไหล่ของแขนข้อมือ เมื่อกดปุ่มส่วน Shoulder แสดงดังภาพ ลักษณะการเคลื่อนที่ของแขนข้อมือ จะเคลื่อนที่ระนาบ xy ตามหน้าจอแอปพลิเคชัน



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 85 ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ โดยการควบคุมส่วนข้อศอกของแขนหุ่นยนต์

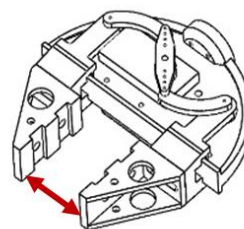
จากภาพที่ 85 เป็น การควบคุมที่ละเอียดแบบอิสระ โดยเป็นส่วนการควบคุมเซอร์โวมอเตอร์ข้อศอกของแขนหุ่นยนต์ เมื่อกดปุ่มส่วน Elbow แสดงดังภาพ ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ จะเคลื่อนที่ระนาบ xy ตามหน้าจอบอพลีเคชั่น



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 86 ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ โดยการควบคุมส่วนข้อมือของแขนหุ่นยนต์

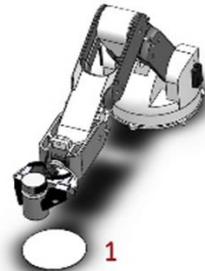
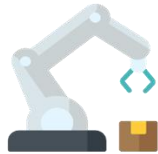
จากภาพที่ 86 เป็น การควบคุมที่ละเอียดแบบอิสระ โดยเป็นส่วนการควบคุมเซอร์โวมอเตอร์ส่วนข้อมือของแขนหุ่นยนต์ เมื่อกดปุ่มส่วน Wrist roll แสดงดังภาพ ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ จะเคลื่อนที่ระนาบ xy ตามหน้าจอบอพลีเคชั่น



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 87 ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ โดยการควบคุมส่วนมือจับของแขนหุ่นยนต์

จากภาพที่ 87 เป็น การควบคุมที่ละเอียดต่อแบบอิสระ โดยเป็นส่วนการควบคุมเซอร์โวมอเตอร์ส่วนมือจับของแขนหุ่นยนต์ เมื่อกดปุ่มส่วน Grip แสดงดังภาพ วัตถุประสงค์การใช้งานเพื่อเป็นส่วนในการจับสิ่งของหรือปล่อยสิ่งของ

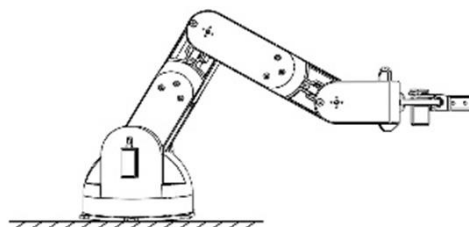


(ก) การกดปุ่มสำหรับการควบคุม

(ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 88 ลักษณะการเคลื่อนที่ของแขนหุ่นยนต์ โดยปุ่มฟังก์ชันวางของอัตโนมัติ

จากภาพที่ 88 เป็นปุ่มฟังก์ชันเมื่อผู้ใช้ทำการจับสิ่งของได้แล้ว แขนหุ่นยนต์จะนำสิ่งของไปวางไว้ที่อีกตำแหน่ง โดยผู้ใช้ไม่ต้องบังคับเอง โดยเป็นส่วนการควบคุมเซอร์โวมอเตอร์หลายตัวพร้อมกัน เพื่อให้ทำตามวัตถุประสงค์ คือเมื่อผู้ใช้ทำการบังคับแขนหุ่นยนต์ให้จับสิ่งของที่ต้องการได้แล้ว กดปุ่มนี้ แสดงดังภาพ 89(ก) แขนหุ่นยนต์จะขยับจากตำแหน่งเดิมไปยังอีกตำแหน่ง แล้วนำสิ่งของมาวางไว้ในกล่องสำหรับใส่ของ วัตถุประสงค์เพื่อ กรณีตำแหน่งที่ผู้ใช้ให้หุ่นยนต์เคลื่อนที่ไป มีสิ่งของที่ต้องการหลายชิ้น เช่น กระปุกยา ขวดน้ำ รีโมททีวี เป็นต้น ผู้ใช้ต้องบังคับหุ่นยนต์ให้หยิบจับสิ่งของ แล้วให้หุ่นยนต์เคลื่อนที่กลับมายังตำแหน่งที่ผู้ใช้อยู่ที่ละชิ้น จึงทำให้ไป-กลับหลายรอบ ซึ่งกรณีนี้ผู้ใช้ต้องบังคับหุ่นยนต์ให้หยิบจับสิ่งของหลายรอบ ซึ่งเป็นขั้นตอนที่ยุ่งยาก และทำให้ต้องใช้เวลามาก ดังนั้นปุ่มฟังก์ชันนี้จึงแก้ไขปัญหาที่กล่าวมาข้างต้น ลดขั้นตอนที่ยุ่งยาก ลดเวลาที่ต้องให้หุ่นยนต์เคลื่อนที่ไป-กลับหลายรอบ และทำให้ผู้ใช้งานได้สะดวกมากยิ่งขึ้น



(ก) การกดปุ่มสำหรับการควบคุม (ข) การเคลื่อนที่ของหุ่นยนต์

ภาพ 89 ปุ่มฟังก์ชันสำหรับช่วยในการตั้งค่าตำแหน่งเริ่มต้นให้แขนหุ่นยนต์พร้อมใช้งาน

จากภาพที่ 89 เป็นปุ่มฟังก์ชันสำหรับช่วยในการตั้งค่าตำแหน่งเริ่มต้นให้แขนหุ่นยนต์พร้อมใช้งาน วัตถุประสงค์การใช้งานเพื่อเมื่อผู้ใช้กดปุ่มฟังก์ชันวางของอัตโนมัติเรียบร้อยแล้ว จากภาพที่ 88(ข) ตำแหน่งของแขนหุ่นยนต์จะไปอยู่ตำแหน่งหมายเลข 2 ทำให้ผู้ใช้ต้องบังคับแขนหุ่นยนต์ให้กลับสู่ตำแหน่งที่พร้อมใช้งานสำหรับหยิบจับสิ่งของชิ้นต่อไป จึงทำให้อาจเกิดความยุ่งยากขึ้นในขั้นตอนนี้ ดังนั้นปุ่มฟังก์ชันสำหรับช่วยในการตั้งค่าตำแหน่งเริ่มต้นให้แขนหุ่นยนต์พร้อมใช้งาน สร้างมาเพื่อลดขั้นตอนที่ยุ่งยากในส่วนนี้ลง และเพิ่มความสะดวกให้ผู้ใช้งานได้ง่ายยิ่งขึ้น



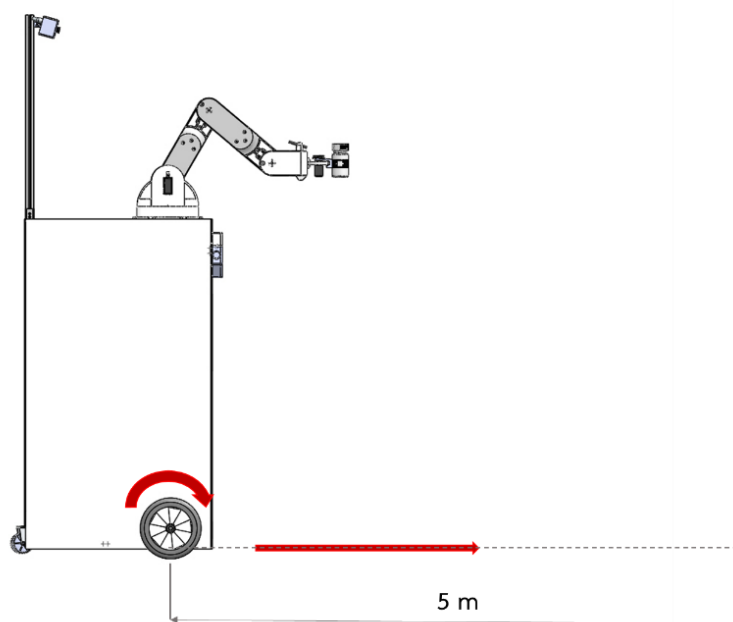
บทที่ 4

ผลการวิจัย และการอภิปรายผล

ในงานวิจัยนี้ทางผู้วิจัยได้ออกแบบการทดลองของหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง โดยแบ่งการทดลองเป็น 2 การทดลองหลัก คือ 1. การทดลองการเคลื่อนที่ของหุ่นยนต์ แบ่งเป็น 3 การทดลองย่อย การทดลองให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรงที่ PWM ต่างกัน ต่อมาให้หุ่นยนต์เคลื่อนที่เป็นเส้นโค้ง เพื่อหาความโค้งที่เหมาะสมให้กับหุ่นยนต์ และการทดสอบให้ผู้ควบคุมหุ่นยนต์ให้เคลื่อนที่ตามกำหนด บังคับผ่านแอปพลิเคชันบนสมาร์ตโฟน 2. การทดลองแขนหุ่นยนต์ ซึ่งจะแบ่งออกไปอีก 2 การทดลองย่อย คือ การทดลองแรกเป็นการทดสอบความสามารถในการควบคุมมือจับของแขนหุ่นยนต์เคลื่อนที่เป็นเส้นตรงในแนวหน้า-หลัง (+x, -x) และแนวขึ้น-ลง (+y, -y) ซึ่งเป็นการเคลื่อนที่ของแขนหุ่นยนต์ในระนาบสองมิติ โดยใช้หลักการจลนศาสตร์แบบไปข้างหน้าและจลนศาสตร์แบบผกผัน และการทดลองที่ต่อมาเป็นการเป็นการทดสอบด้านระยะเวลาที่ใช้ควบคุมแขนหุ่นยนต์เพื่อหยิบสิ่งของ โดยแบ่งการควบคุมเป็นควบคุมแบบ MALM-Control และการควบคุมด้วยข้อต่ออิสระทั้ง 6 ข้อต่อ

4.1 การทดลองการเคลื่อนที่ของหุ่นยนต์

ในงานวิจัยนี้ได้ออกแบบหุ่นยนต์ที่มีการติดตั้งล้อเป็นแบบประเภท Differential Drive Robot คือ มีการขับเคลื่อนด้วยสองล้อหน้า และมีล้ออิสระด้านหลัง โดยมีระบบควบคุมด้วยไมโครคอนโทรลเลอร์ สั่งการผ่านแอปพลิเคชันบนสมาร์ตโฟนที่ได้พัฒนาขึ้น ซึ่งตัวหุ่นยนต์มีการติดตั้งกล้องเพื่อตรวจจับภาพบริเวณด้านหน้า แล้วส่งข้อมูลภาพผ่านสัญญาณไร้สายไปยังโทรศัพท์ ทำให้ผู้ใช้สามารถควบคุมให้หุ่นยนต์ให้เคลื่อนที่ไปตามจุดต่าง ๆ ตามที่กำหนดได้ โดยการทดลองแบ่งเป็นการทดสอบปรับค่า PWM ของล้อทั้งสองข้างให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรง จากนั้นจับเวลาที่หุ่นยนต์เคลื่อนที่ไปในระยะ 5 m เพื่อหาความเร็วการของหุ่นยนต์ เพื่อดูกราฟความสัมพันธ์ระหว่างค่า PWM กับค่าความเร็วของหุ่นยนต์ ต่อมาเป็นการทดสอบการโค้งของหุ่นยนต์ โดยปรับค่า PWM ของล้อซ้ายและล้อขวา เพื่อกำหนดการเลี้ยวโค้งที่เหมาะสมของหุ่นยนต์ และการทดลองสุดท้ายเป็นการทดสอบให้ผู้ควบคุมหุ่นยนต์ให้เคลื่อนที่ไป-กลับ อ้อมระหว่างกรวยซึ่งห่างกัน 5 m แสดงดังภาพที่ 90



ภาพ 90 การทดลองหุ่นยนต์เคลื่อนที่เป็นเส้นตรง

การทดลองให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรงทั้งเดินหน้า และถอยหลัง โดยการปรับค่า PWM ของล้อทั้งสองข้าง ที่ระยะ 5 m แบ่งการทดลองโดยปราศจากน้ำหนักและทดลองด้วยมีภาระน้ำหนัก โดยการเพิ่มภาระน้ำหนักให้ตัวหุ่นยนต์เป็น 10 kg และ 20 kg แล้วแสดงผลการทดลองดังตารางที่ 4 ถึงตารางที่ 6 ซึ่งการทดสอบนี้เป็นการทดสอบความเร็วในการเคลื่อนที่ ในทิศทางตรงไปข้างหน้า และเพื่อแยกปัจจัยด้านน้ำหนัก

ตาราง 4 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง ภายในห้องทดสอบ ไม่มีภาระน้ำหนักเพิ่มเติม

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 0 kg				ความเร็วเฉลี่ย (m/s)	
	เดินหน้า	ค่าเบี่ยงเบน	ถอยหลัง	ค่าเบี่ยงเบน	เดินหน้า	ถอยหลัง
		มาตรฐาน		มาตรฐาน		
50	71.0	0.89	72.6	0.77	0.07	0.07
100	32.0	0.38	32.9	1.14	0.16	0.15
150	20.9	0.61	21.1	0.91	0.24	0.24
200	15.4	0.55	15.6	0.37	0.32	0.32
250	12.1	0.23	12.4	0.46	0.41	0.40

ตาราง 5 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง ภายในห้องทดสอบ เพิ่มน้ำหนักที่ 10 kg

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 10 kg				ความเร็วเฉลี่ย (m/s)	
	เดินหน้า	ค่าเบี่ยงเบน	ถอยหลัง	ค่าเบี่ยงเบน	เดินหน้า	ถอยหลัง
		มาตรฐาน		มาตรฐาน		
50	81.9	1.10	84.5	0.53	0.06	0.06
100	34.0	0.83	34.7	0.87	0.15	0.14
150	21.7	0.82	22.1	0.79	0.23	0.23
200	15.8	0.65	16.1	0.42	0.32	0.31
250	12.4	0.45	12.8	0.29	0.40	0.39

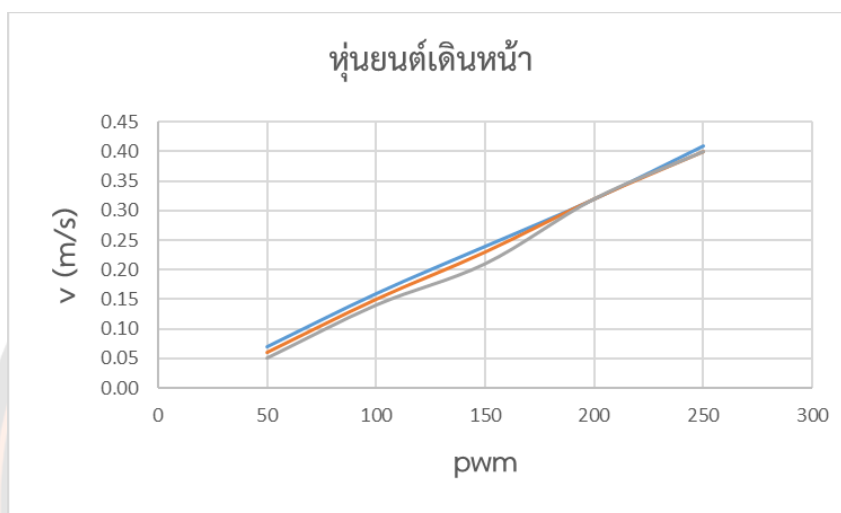
ตาราง 6 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง ภายในห้องทดสอบ เพิ่มน้ำหนักที่ 20 kg

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 20 kg				ความเร็วเฉลี่ย (m/s)	
	เดินหน้า	ค่าเบี่ยงเบน	ถอยหลัง	ค่าเบี่ยงเบน	เดินหน้า	ถอยหลัง
		มาตรฐาน		มาตรฐาน		
50	92.6	0.71	95.5	0.71	0.05	0.05
100	35.2	0.32	37.3	0.65	0.14	0.13
150	23.7	0.56	21.9	0.48	0.21	0.23
200	15.5	0.37	15.9	0.71	0.32	0.31
250	12.4	0.58	13.1	0.40	0.40	0.38

จากตาราง 4 ถึง ตารางที่ 6 แสดงผลการทดลองการเคลื่อนที่เป็นเส้นของหุ่นยนต์ โดยปรับค่า PWM ของล้อทั้งสองข้าง เริ่มต้นที่ 50 pulse จนถึง 250 pulse เพิ่มทีละ 50 pulse จากการทดลองการเคลื่อนที่ไปข้างหน้ากับถอยหลัง พบว่าค่าความเร็วเฉลี่ยมีค่าใกล้เคียงกัน เมื่อเพิ่มภาระน้ำหนักให้กับตัวหุ่นยนต์ และค่าเบี่ยงเบนมาตรฐานของปัจจัยการตั้งค่าเดียวกันมีค่าน้อย แสดงว่ากลุ่มข้อมูลมีการกระจายที่น้อยข้อมูลส่วนใหญ่อยู่ใกล้ค่าเฉลี่ยมาก จากผลการทดสอบพบว่า การควบคุมให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรงที่มีภาระน้ำหนักเพิ่มขึ้นไม่ส่งผลต่อความเร็วของหุ่นยนต์ จึงสรุปได้ว่าหุ่นยนต์สามารถเคลื่อนที่ได้โดยสามารถรับภาระน้ำหนักเพิ่มขึ้นจากภายนอก โดยที่ความเร็วของหุ่นยนต์ยังคงที่

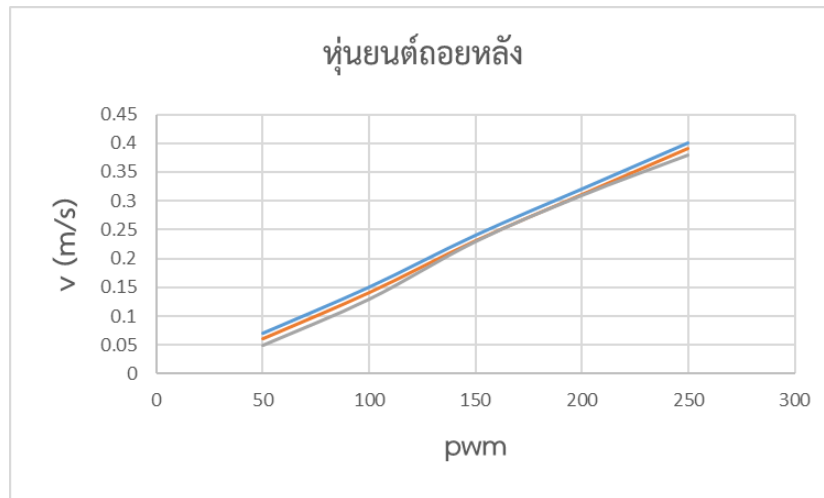
ผลการทดลองจากตารางที่ 4 ถึงตารางที่ 6 ทำให้ทราบความเร็วในการเคลื่อนที่ของหุ่นยนต์ จากนั้นพล็อตกราฟความสัมพันธ์ระหว่างค่า PWM กับความเร็วของหุ่นยนต์ โดยการทดลองจะแบ่งให้

หุ่นยนต์เดินหน้า และถอยหลัง กราฟหุ่นยนต์เคลื่อนที่ไปข้างหน้าหน้าแสดงดังภาพที่ 91 มีรายละเอียดคือ เส้นสีน้ำเงินเป็นเส้นที่แสดงความสัมพันธ์ค่า PWM กับความเร็ว ที่ไม่มีภาระน้ำหนักเพิ่มเติม ต่อมาเส้นสีส้มที่ภาระน้ำหนักเท่ากับ 10 kg และสุดท้ายเส้นสีเทาที่ภาระน้ำหนักเท่ากับ 20 kg จากเส้นกราฟทั้งสามเส้น สามารถเกาะกลุ่มกันได้ดี แสดงให้เห็นว่าภาระน้ำหนักที่เพิ่มขึ้นไม่ส่งผลกับความเร็วของหุ่นยนต์มากนัก



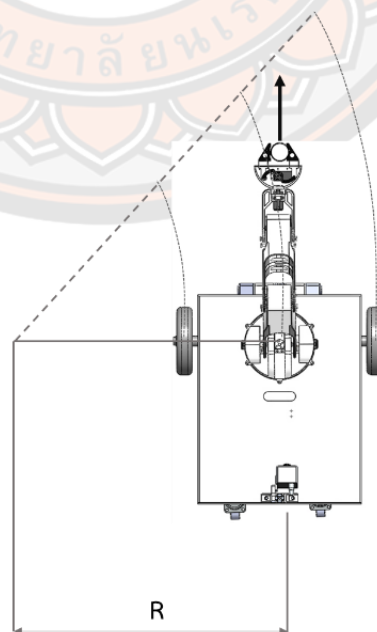
ภาพ 91 ความสัมพันธ์ค่า PWM กับความเร็วของหุ่นยนต์เคลื่อนที่ไปข้างหน้า

ต่อมากราฟความสัมพันธ์ระหว่างค่า PWM กับความเร็ว ของหุ่นยนต์เคลื่อนที่ไปข้างหลัง แสดงดังภาพที่ 92 โดยรายละเอียดคือ เส้นสีน้ำเงินเป็นเส้นภาระน้ำหนักเท่ากับ 0 kg ต่อมาเส้นสีส้มที่ภาระน้ำหนักเท่ากับ 10 kg และสุดท้ายเส้นสีเทาที่ภาระน้ำหนักเท่ากับ 20 kg จากกราฟทั้งสามเส้นที่ภาระน้ำหนักต่าง ๆ ที่เพิ่มขึ้น แสดงถึงความเร็วที่คงที่ของหุ่นยนต์ ดังนั้นหุ่นยนต์สามารถรับภาระน้ำหนักที่เพิ่มขึ้นได้โดยความเร็วของหุ่นยนต์ยังคงที่อยู่



ภาพ 92 ความสัมพันธ์ค่า PWM กับความเร็วของหุ่นยนต์เคลื่อนที่ข้างหลัง

การทดสอบให้หุ่นยนต์เคลื่อนที่ด้วยการเลี้ยวโค้ง เพื่อหาความสัมพันธ์ระหว่างอัตราเร็วเชิงมุมกับ PWM โดยความเร็วของล้อถูกควบคุมด้วยตัวขับซึ่งรับคำสั่งด้วยสัญญาณ PWM การทดลองนี้ได้ทดสอบความสัมพันธ์ระหว่างสัญญาณที่ป้อนเข้า (PWM) และความเร็วเชิงมุมที่เกิดขึ้น ให้สามารถควบคุมความเร็วของหุ่นยนต์ได้อย่างถูกต้อง โดยปรับค่า PWM ของล้อซ้ายและล้อขวา เพื่อกำหนดการเลี้ยวโค้งที่เหมาะสมของหุ่นยนต์ จากภาพที่ 91 กับภาพที่ 92 กราฟความสัมพันธ์ระหว่างค่า PWM กับความเร็วของหุ่นยนต์ ทำให้ทราบอัตราเร็วเชิงมุม ทำให้สามารถหาค่ารัศมีการเคลื่อนที่ของหุ่นยนต์ได้จากโมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์ การทดลองการเลี้ยวโค้งของหุ่นยนต์ แสดงดังภาพที่ 93



ภาพ 93 การทดลองการเลี้ยวโค้งของหุ่นยนต์

การทดลองแบ่งเป็นแบบไม่มีภาระน้ำหนักเพิ่มเติม จากนั้นเพิ่มภาระน้ำหนักให้ตัวหุ่นยนต์ เป็น 10 kg และ 20 kg แสดงผลการทดลองดังตารางที่ 7 กับตารางที่ 8

ตาราง 7 การทดสอบการเลี้ยวโค้งซ้ายของหุ่นยนต์

PWM ของล้อ ข้างซ้าย	PWM ของล้อ ข้างขวา	รัศมีจากการคำนวณ	รัศมีการเลี้ยวโค้ง (m)		
			0 kg	10 kg	20 kg
80	160	0.66	0.69	0.68	0.68
80	180	0.58	0.60	0.61	0.60
80	200	0.52	0.54	0.54	0.55
100	180	0.78	0.85	0.86	0.85
100	200	0.67	0.70	0.70	0.69
100	220	0.60	0.62	0.62	0.62
ค่าความคลาดเคลื่อนรัศมีจากการคำนวณกับค่าที่ได้			4.38%	4.61%	3.88%
จากการวัด					

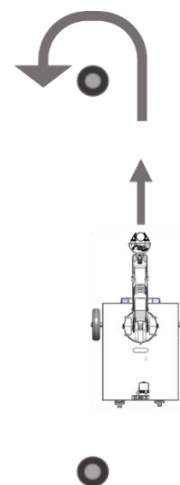
ตาราง 8 การทดสอบการเลี้ยวโค้งขวาของหุ่นยนต์

PWM ของล้อ ข้างซ้าย	PWM ของล้อ ข้างขวา	รัศมีจากการ คำนวณ	รัศมีการเลี้ยวโค้ง (m)		
			0 kg	10 kg	20 kg
160	80	0.66	0.61	0.60	0.60
180	80	0.58	0.54	0.57	0.57
200	80	0.52	0.50	0.51	0.51
180	100	0.78	0.72	0.71	0.70
200	100	0.67	0.63	0.65	0.66
220	100	0.60	0.56	0.55	0.55
ค่าความคลาดเคลื่อนรัศมีจากการคำนวณกับค่าที่ได้			7.13%	6.05%	5.82%
จากการวัด					

การทดสอบการเลี้ยวโค้งของหุ่นยนต์จากตารางที่ 7 กับตารางที่ 8 จะได้ผลการทดลองค่า เส้นรัศมีการเลี้ยวโค้งของหุ่นยนต์ที่ได้จากการวัด ทำการเปรียบเทียบค่าที่ได้จากการทดลองวัดจริงกับค่าที่ได้จากการคำนวณจากโมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์ จากผลการทดลองการเลี้ยวโค้งซ้ายของหุ่นยนต์ รัศมีจากการคำนวณเปรียบเทียบกับค่ารัศมีที่ได้

จากการวัดจะได้ค่าความคลาดเคลื่อนมาตรฐานที่ภาระน้ำหนักต่าง ๆ ดังต่อไปนี้ ไม่มีการเพิ่มภาระน้ำหนักค่าความคลาดเคลื่อนได้เท่ากับ 4.38% ต่อมาที่ภาระน้ำหนักที่ 10 kg มีค่าความคลาดเคลื่อนเท่ากับ 4.61% และที่ภาระน้ำหนัก 20 kg มีค่าความคลาดเคลื่อนเท่ากับ 3.88% ต่อมาผลการเปรียบเทียบค่าร้อยละจากการคำนวณกับค่าร้อยละที่ได้จากการคำนวณของการเลี้ยวโค้งขวาของหุ่นยนต์จะได้ ค่าความคลาดเคลื่อนที่ภาระน้ำหนักต่าง ๆ ดังต่อไปนี้ ไม่มีการเพิ่มภาระน้ำหนักได้ค่าความคลาดเคลื่อนเท่ากับ 7.13% ต่อมาที่ภาระน้ำหนักที่ 10 kg มีค่าความคลาดเคลื่อนเท่ากับ 6.05% และที่ภาระน้ำหนัก 20 kg มีค่าความคลาดเคลื่อนเท่ากับ 5.82% จากการเปรียบเทียบพบว่าหุ่นยนต์สามารถเคลื่อนที่โดยการเลี้ยวโค้งค่าร้อยละจากการคำนวณกับร้อยละจากการวัดมีค่าความคลาดเคลื่อนที่ยอมรับได้ ทั้งนี้ค่าความคลาดเคลื่อนที่อาจเกิดจากหลายปัจจัย เช่น ความคลาดเคลื่อนจากการติดตั้งอุปกรณ์ ของล้อซ้ายขวาไม่เท่ากัน ทำให้แรงที่ลงของล้อซ้ายและขวาอาจจะไม่เท่ากัน จากสมการโมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์ ผลการทดลองเป็นไปตามทฤษฎี คือการเลี้ยวโค้งของล้อแต่ละข้าง เช่น การโค้งซ้าย ความเร็วล้อซ้ายต้องน้อยกว่าล้อขวา วงเลี้ยวจะแคบลง เมื่อความเร็วล้อขวาเพิ่มขึ้น กลับกันการโค้งขวาความเร็วล้อขวาต้องน้อยกว่าล้อซ้าย วงการเลี้ยวจะแคบลง เมื่อความเร็วล้อซ้ายเพิ่มขึ้น ในการวิจัยนี้ได้ออกแบบให้หุ่นยนต์เคลื่อนที่ได้ในทิศทางต่าง ๆ โดยการเลี้ยวโค้งของหุ่นยนต์ต้องกำหนดค่า PWM ของล้อแต่ละที่ที่เหมาะสมกับการควบคุม

การทดลองสุดท้ายของการเคลื่อนหุ่นยนต์ คือทดสอบให้ผู้ใช้ควบคุมหุ่นยนต์ให้เคลื่อนที่ไป-กลับ อ้อมระหว่างกรวยซึ่งห่างกัน 5 m ทดลองกับผู้ใช้งานเป็นจำนวน 3 คน โดยการทดลองนี้วัตถุประสงค์เพื่อให้ทราบถึงการใช้งานจริงที่ผู้ใช้ต้องบังคับไปยังจุดต่าง ๆ ในการทดสอบออกแบบมาให้มีทั้งระยะการเคลื่อนที่ทางตรง และมีการเลี้ยวโค้ง เพื่อให้ผู้ใช้งานได้ควบคุมหุ่นยนต์ที่มีเส้นทางหลากหลาย การทดสอบจะทดลองซ้ำเป็นจำนวน 5 ครั้ง การทดสอบการเคลื่อนที่ของหุ่นยนต์ไป-กลับ แสดงดังภาพที่ 94



ภาพ 94 การทดสอบการเคลื่อนที่ของหุ่นยนต์ไป-กลับ

ผลการทดลองควบคุมให้ผู้ใช้ควบคุมหุ่นยนต์ผ่านแอปพลิเคชันบนโทรศัพท์มือถือ ให้เคลื่อนที่ไป-กลับ อ้อมระหว่างกรวย แสดงดังตารางที่ 9

ตาราง 9 การทดสอบควบคุมหุ่นยนต์ไป-กลับอ้อมกรวยภายในห้องทดสอบ

	ครั้งที่	ระยะเวลา	ความเร็วเฉลี่ย
คนที่1	1	89 second	11.24 cm/s
	2	87 second	11.49 cm/s
	3	88 second	11.36 cm/s
	4	85 second	11.76 cm/s
	5	82 second	12.2 cm/s
เฉลี่ย		86.2 second	11.61 cm/s
ค่าเบี่ยงเบนมาตรฐาน		2.77	0.38
คนที่2	1	102 second	9.80 cm/s
	2	105 second	9.52 cm/s
	3	97 second	10.31 cm/s
	4	101 second	9.90 cm/s
	5	95 second	10.53 cm/s
เฉลี่ย		100 second	10.01 cm/s
ค่าเบี่ยงเบนมาตรฐาน		4.00	0.40
	1	95 second	10.53 cm/s
	2	92 second	10.87 cm/s

คนที่3	3	88 second	11.36 cm/s
	4	90 second	11.11 cm/s
	5	85 second	11.76 cm/s
เฉลี่ย		90 second	11.13 cm/s
ค่าเบี่ยงเบนมาตรฐาน		3.81	0.47

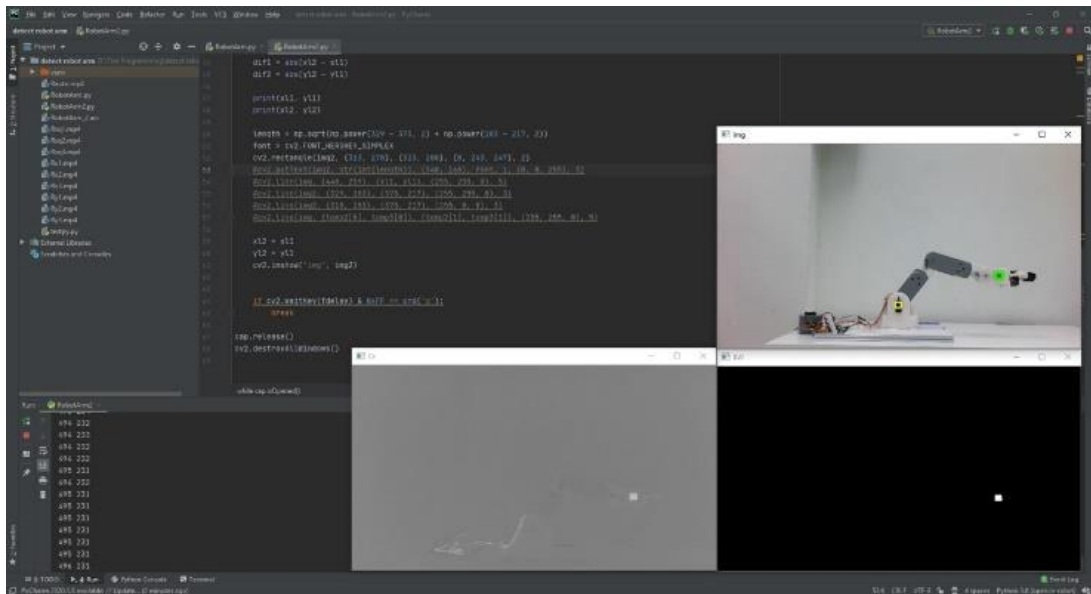
จากตารางที่ 9 พบว่าผู้ใช้งานคนแรกใช้เวลาประมาณ 86.2 วินาที หรือความเร็วเฉลี่ยประมาณ 11.61 cm/s ต่อมาคนที่สองใช้เวลาประมาณ 100 วินาที หรือความเร็วเฉลี่ยประมาณ 10.01 cm/s และคนสุดท้ายใช้เวลาประมาณ 90 วินาที หรือความเร็วเฉลี่ยประมาณ 11.13 cm/s ซึ่งส่วนเบี่ยงเบนมาตรฐานของการทดสอบผู้ใช้งานทั้งสามคนมีค่าออกมาน้อย แสดงว่าข้อมูลการทดลองที่วัดได้แต่ละครั้งมีค่าใกล้เคียงกับค่าเฉลี่ย

4.2 การทดลองส่วนของแขนหุ่นยนต์

การทดลองของแขนหุ่นยนต์ โดยแบ่งการทดลองเป็น 2 การทดลอง คือ ทดลองการเคลื่อนที่ของแขนหุ่นยนต์ในแนวระนาบ โดยเคลื่อนที่เป็นเส้นตรงในแนวหน้า-หลัง และแนวขึ้น-ลง และการทดลองให้ผู้ควบคุมแขนหุ่นยนต์เพื่อหยิบสิ่งของ โดยควบคุมเป็นควบคุมแบบ MARM-Control และการควบคุมด้วยข้อต่ออิสระทั้ง 6 ข้อต่อ

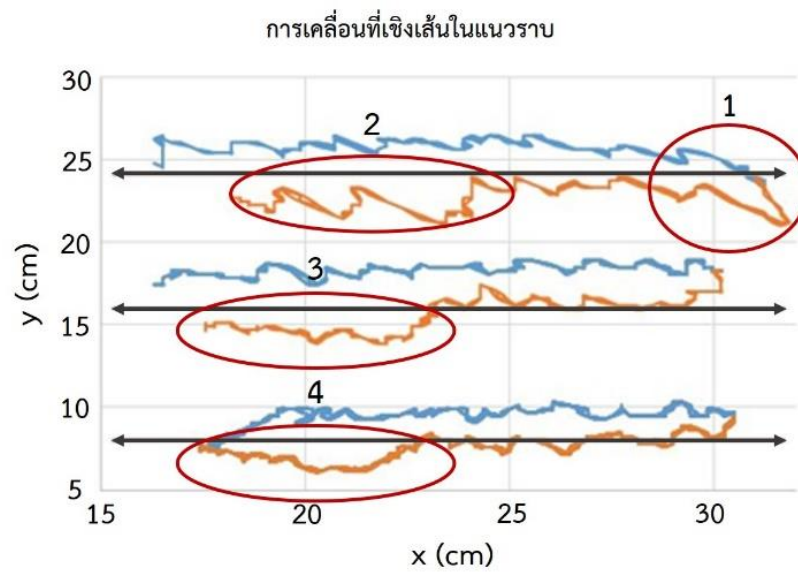
4.2.1 การทดลองการเคลื่อนที่ของแขนหุ่นยนต์ในระนาบสองมิติ

การทดลองการเคลื่อนที่ของแขนหุ่นยนต์ในระนาบสองมิติ ซึ่งเป็นการให้แขนหุ่นยนต์เคลื่อนที่เป็นเส้นตรงตามแนวราบและแนวตั้ง จากนั้นทำการตรวจสอบว่าแขนหุ่นยนต์เคลื่อนที่เป็นเส้นตรงได้ถูกต้องหรือไม่ โดยใช้เทคโนโลยีประมวลผลภาพ (Image Processing) แสดงดังภาพที่ 95 ทำการเขียนโปรแกรมด้วยไลบรารีที่ชื่อว่า Open CV ด้วยภาษาไพทอน (Python) โดยทำเครื่องหมายสีแดงไว้ที่ตำแหน่งปลายแขนหุ่นยนต์ จากนั้นโปรแกรมจะให้กล้องตรวจจับเครื่องหมายสีแดง แล้วติดตามการเคลื่อนที่ตามปลายแขนหุ่นยนต์



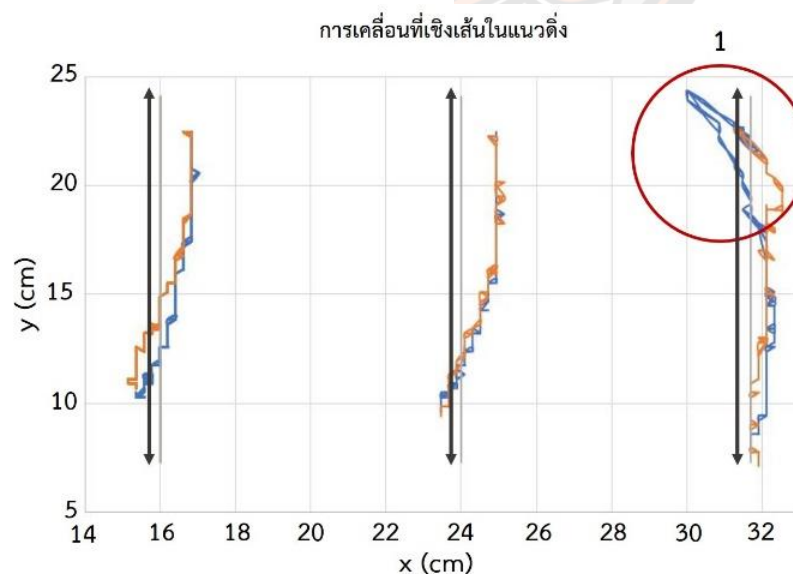
ภาพ 95 โปรแกรมประมวลผลภาพ ตรวจจับจุดสีแดง เคลื่อนที่ตามปลายแขนหุ่นยนต์

ผลการทดลองให้แขนหุ่นยนต์เคลื่อนที่เชิงเส้นตามแนวราบ โดยให้เคลื่อนที่ไปกลับ ทำการทดลองทั้งหมด 3 ระดับ คือที่ค่า $y = 8 \text{ cm}$, 16 cm และ 24 cm จากภาพที่ 96 แสดงถึงเส้นอ้างอิงตำแหน่งที่ขอบแขนหุ่นยนต์เคลื่อนที่ได้ในแนวราบ โดยแบ่งออกเป็นสามระดับคือ min medium max ต่อมาเส้นการเคลื่อนที่ขาไป (เส้นสีน้ำเงิน) และเส้นการเคลื่อนที่ขากลับ(เส้นสีส้ม) จากการวิเคราะห์การเคลื่อนที่ของแขนหุ่นยนต์มีจุดสังเกตคือ จุดที่ 1 มีตำแหน่งคลาดเคลื่อนจากแนวที่กำหนดไว้ประมาณ 2-3 cm ซึ่งเกิดจากการยึดแขนไปสุดพื้นที่ทำงาน ทำให้แรงบิดของมอเตอร์ส่วนหัวไหล่และข้อศอกไม่เพียงพอ แขนหุ่นยนต์จึงไม่สามารถกลับไปสู่ตำแหน่งที่เดิมได้ จุดที่ 2, 3 และ 4 เกิดจากมอเตอร์ส่วนหัวไหล่ และข้อศอกพยายามดึงกลับมายังตำแหน่งของโปรแกรม ทำให้เกิดการกระตุกในช่วงนี้ และการเคลื่อนที่ของแขนหุ่นยนต์ทั้งขาไปและขากลับ ไม่เรียบสม่ำเสมอ ซึ่งเกิดจากการโครงสร้างของแขนหุ่นยนต์ส่วนของข้อมือมีความเคลื่อนตัวมาก



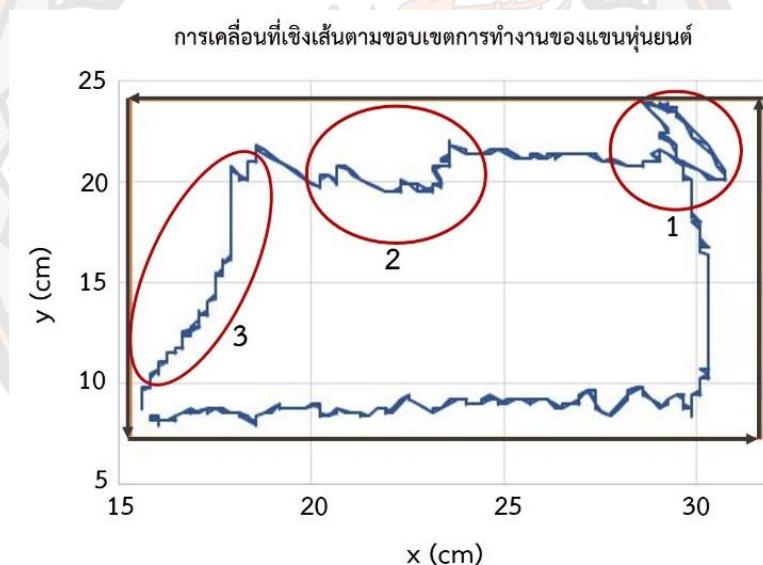
ภาพ 96 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวราบ

ต่อมาเป็นการทดลองเคลื่อนที่เชิงเส้นตามแนวตั้ง โดยทดลอง 3 ระดับ คือที่ค่า $x = 16$ cm, 24 cm และ 31.7 cm จากภาพที่ 97 แสดงถึงเส้นอ้างอิงตำแหน่งที่ขอบแขนหุ่นยนต์เคลื่อนที่ได้ในแนวตั้ง โดยแบ่งออกเป็นสามระดับคือ min medium max การเคลื่อนที่มีรูปแบบที่ใกล้เคียงกัน ทั้งขาไปและขากลับ มีตำแหน่งคลาดเคลื่อนจากแนวที่กำหนดไว้ประมาณ 1 cm ซึ่งมีจุดสังเกตคือการเคลื่อนที่ช่วง $x = 31.7$ cm ซึ่งเกิดจากการยึดแขนหุ่นยนต์เข้าใกล้กับแนวขอบเขตพื้นที่ทำงาน ทำให้แรงบิดของมอเตอร์ส่วนหัวไหล่ และข้อศอกไม่เพียงพอ



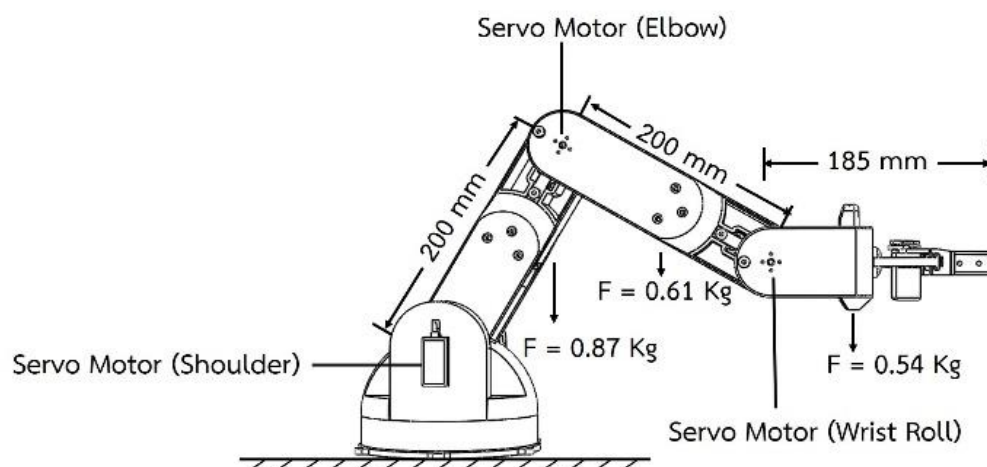
ภาพ 97 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวตั้ง

ต่อมาเป็นการทดลองเคลื่อนที่เชิงเส้นตามขอบเขตการทำงานของแขนหุ่นยนต์ จากภาพที่ 98 แสดงเส้นอ้างอิงการเคลื่อนที่ของแขนหุ่นยนต์ จากการใช้โปรแกรมประมวลผลภาพจับตำแหน่งที่กำหนด (ภาพที่ 93) ทำให้เกิดค่าระยะห่างระหว่างเส้นอ้างอิงกับเส้นการเคลื่อนที่ของแขนหุ่นยนต์ จากการวิเคราะห์หามีจุดสังเกตคือ จุดที่ 1 เป็นตำแหน่งที่ใกล้กับแนวขอบเขตพื้นที่ทำงานของแขนหุ่นยนต์ ดังนั้นการยืดแขนหุ่นยนต์ไปยังตำแหน่งดังกล่าว ทำให้แรงบิดของมอเตอร์ส่วนหัวไหล่ และข้อศอกไม่เพียงพอ ต่อมาจุดที่ 2 คือการเคลื่อนที่ขากลับ เกิดจากมอเตอร์ส่วนหัวไหล่ และข้อศอกพยายามดึงกลับมายังตำแหน่งของโปรแกรม ทำให้เกิดการกระตุกในช่วงนี้ และจุดที่ 3 เกิดจากข้อจำกัดทางโครงสร้างของแขนหุ่นยนต์



ภาพ 98 การเคลื่อนที่ตามขอบเขตการทำงานของแขนหุ่นยนต์

จากการทดลองการเคลื่อนที่เชิงเส้นของแขนหุ่นยนต์ในระนาบสองมิติพบว่า มีความคลาดเคลื่อนเล็กน้อยอันเนื่องมาจากแรงบิดของเซอร์โวมอเตอร์ไม่เพียงพอ ทางผู้วิจัยจึงได้ทำการทดสอบขีดจำกัดในด้านน้ำหนักสูงสุดในการหยิบจับสิ่งของที่แขนหุ่นยนต์สามารถเคลื่อนย้ายได้ ในรูปแบบของเวกเตอร์แรงและโมเมนต์ที่กระทำต่อแต่ละข้อต่อของแขนหุ่นยนต์ แสดงดังภาพที่ 99 และเปรียบเทียบระหว่างแรงบิดแต่ละข้อต่อที่แขนหุ่นยนต์ต้องการกับแรงบิดสูงสุดของเซอร์โวมอเตอร์ แสดงดังตารางที่ 10 โดยน้ำหนักสูงสุดของวัตถุกำหนดให้เป็น 200 g



ภาพ 99 เวกเตอร์แรงและโมเมนต์ที่กระทำต่อแต่ละช่วงข้อต่อ

ตาราง 10 แรงบิดแต่ละช่วงต่อของแขนหุ่นยนต์ และแรงบิดสูงสุดของเซอร์โวมอเตอร์

ตำแหน่งเซอร์โวมอเตอร์	น้ำหนัก (kg)	แรงบิดจากการคำนวณ (kg-cm)	แรงบิดสูงสุดของมอเตอร์ (kg-cm)
Shoulder	1.74	34.76	41.40
Elbow	1.21	24.24	20.70
Wrist Roll	0.68	12.49	20.70

จากตารางที่ 10 จะเห็นได้ว่าแรงบิดของส่วน Shoulder มีค่าที่ 34.76 kg-cm ใกล้กับแรงบิดสูงสุดตามสเปกของเซอร์โวมอเตอร์ที่ 41.40 kg-cm ทั้งนี้ในการเคลื่อนที่จริงจะมีแรงบิดเพิ่มเติมที่เซอร์โวมอเตอร์ ทำให้ในการใช้งานจริงอาจเกิดความคลาดเคลื่อนได้ ต่อมาส่วน Elbow แรงบิดมีค่าที่ 24.24 kg-cm ซึ่งมากกว่าค่าแรงบิดสูงสุดของเซอร์โวมอเตอร์มีที่ 20.70 kg-cm เพียงเล็กน้อย ทำให้อาจเกิดการคลาดเคลื่อนในการใช้งาน และในส่วนแรงบิดที่แขนหุ่นยนต์ต้องการของ Wrist Roll มีค่าน้อยกว่าแรงบิดสูงสุดของเซอร์โวมอเตอร์

4.2.2 การทดลองการควบคุมแขนหุ่นยนต์ผ่านโทรศัพท์มือถือ

การทดลองต่อมาให้ผู้ใช้ควบคุมแขนหุ่นยนต์ผ่านแอปพลิเคชันบนโทรศัพท์มือถือ โดยให้แขนหุ่นยนต์หยิบจับสิ่งของจากจุดหนึ่งไปไว้ที่อีกจุดหนึ่ง ซึ่งผู้ใช้ต้องมองผ่านจอที่แสดงภาพจากกล้องตามเวลาจริง จากนั้นแบ่งการทดลองเป็นการควบคุมด้วยข้อต่ออิสระทั้ง 6 ข้อต่อ และควบคุมแบบ

MALM-Control มาช่วยในการเคลื่อนที่ ซึ่งทำการทดลองให้ผู้ใช้งาน 3 คน ทดลองทั้งหมด 3 ครั้ง โดยโปรแกรมจำลองแสดงการหยิบจับสิ่งของจากจุดหนึ่งไปไว้ที่จุดหนึ่ง การทดลองให้ผู้ใช้งาน ควบคุม บังคับแขนหุ่นยนต์ผ่านโทรศัพท์มือถือ แสดงดังรูปที่ 100



ภาพ 100 การทดลองให้ผู้ใช้งาน ควบคุมบังคับแขนหุ่นยนต์ผ่านโทรศัพท์มือถือ

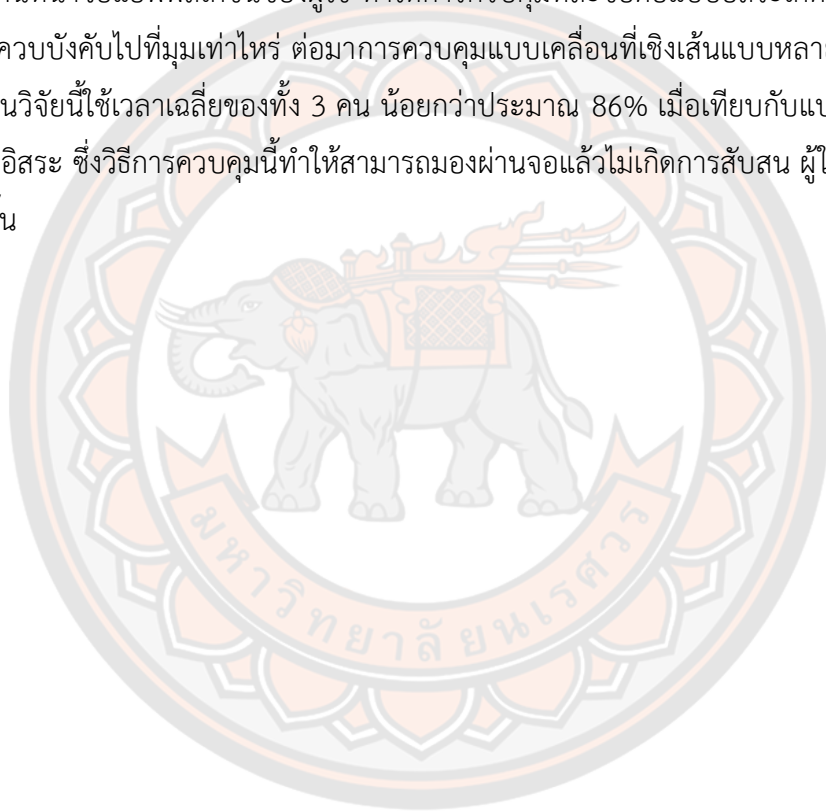
ผลการทดลองควบคุมให้ผู้ใช้งานควบคุมแขนหุ่นยนต์ผ่านแอปพลิเคชันบนโทรศัพท์มือถือ หยิบจับ สิ่งของจากจุดหนึ่งไปไว้ที่จุดหนึ่ง แสดงดังตารางที่ 11

ตาราง 11 ผลการทดลองการควบคุมแขนหุ่นยนต์

	ครั้งที่	ควบคุมแต่ละข้อต่อแบบ อิสระ	ควบคุมแบบ MALM-Control
คนที่1	1	6.12 minute	1.18 minute
	2	4.10 minute	0.48 minute
	3	2.34 minute	0.38 minute
เวลาเฉลี่ย		4.19 minute	1.08 minute
คนที่2	1	5.22 minute	0.54 minute
	2	3.24 minute	0.44 minute
	3	2.18 minute	0.27 minute

เวลาเฉลี่ย		3.55 minute	0.42 minute
คนที่3	1	5.48 minute	0.49 minute
	2	2.36 minute	0.36 minute
	3	1.43 minute	0.28 minute
เวลาเฉลี่ย		3.09 minute	0.38 minute

จากตารางที่ 11 แสดงผลการทดลองการควบคุมแขนหุ่นยนต์ให้หยิบจับสิ่งของ โดยการควบคุมทีละข้อต่อแบบอิสระใช้เวลาในการควบคุมที่มากในครั้งแรก เนื่องจากการไม่คุ้นชินกับการบังคับผ่านหน้าจอแอปพลิเคชันของผู้ใช้ ทำให้การควบคุมทีละข้อต่อแบบอิสระเกิดความสับสนว่าข้อต่อไหนควบบังคับไปที่มุมเท่าไร ต่อมาการควบคุมแบบเคลื่อนที่เชิงเส้นแบบหลายแกนที่พัฒนาขึ้นมาในงานวิจัยนี้ใช้เวลาเฉลี่ยของทั้ง 3 คน น้อยกว่าประมาณ 86% เมื่อเทียบกับแบบควบคุมทีละข้อต่อแบบอิสระ ซึ่งวิธีการควบคุมนี้ทำให้สามารถมองผ่านจอแล้วไม่เกิดการสับสน ผู้ใช้สามารถควบคุมได้ง่ายขึ้น



บทที่ 5

สรุปผลการทดลอง และข้อเสนอแนะ

ในบทนี้จะเป็นบทสรุปของงานวิจัยเรื่องการพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง โดยงานวิจัยนี้เป็นการออกแบบส่วนฮาร์ดแวร์ต่าง ๆ ของหุ่นยนต์ ไม่ว่าจะเป็นส่วนของโครงสร้าง รูปร่าง ขนาดที่เหมาะสมกับวัตถุประสงค์ที่นำไปใช้งาน รวมถึงการออกแบบวงจร และการเลือกอุปกรณ์ต่าง ๆ ที่เหมาะสมกับงานวิจัย ส่วนต่อมาพัฒนาซอฟต์แวร์ของหุ่นยนต์ ไม่ว่าจะเป็นพัฒนาระบบควบคุม การเคลื่อนที่ของหุ่นยนต์ ระบบควบคุมของแขนหุ่นยนต์อย่างระบบควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน และยังพัฒนาส่วนของแอปพลิเคชันสำหรับให้ผู้ใช้งานได้ควบคุมหุ่นยนต์ผ่านโทรศัพท์มือถือ บทนี้ยังมีข้อเสนอแนะเพื่อการปรับปรุงและพัฒนาสิ่งประดิษฐ์ให้ดียิ่งขึ้นในอนาคต

5.1 สรุปผลการทดลอง

งานวิจัยนี้เป็นการพัฒนาหุ่นยนต์บริการสำหรับผู้ป่วยติดเตียง การทดลองจึงแบ่งเป็นส่วนของหุ่นยนต์เคลื่อนที่กับส่วนแขนของหุ่นยนต์ โดยการทดลองของหุ่นยนต์เคลื่อนที่แบ่งเป็น 3 การทดลอง คือ 1. การทดลองให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรงทั้งเดินหน้ากับถอยหลัง ทำการตั้งค่า pwm ของล้อที่ระยะ 5 m แบ่งการทดสอบโดยปราศจากน้ำหนักและทดสอบเพิ่มภาระน้ำหนักให้ตัวหุ่นยนต์เป็น 10 kg และ 20 kg จากผลการทดสอบพบว่าการควบคุมให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรงที่มีภาระน้ำหนักเพิ่มขึ้นไม่ส่งผลต่อความเร็วของหุ่นยนต์ จึงสรุปได้ว่าหุ่นยนต์สามารถเคลื่อนที่ได้โดยสามารถรับภาระน้ำหนักเพิ่มขึ้นจากภายนอก โดยที่ความเร็วของหุ่นยนต์ยังคงที่ 2. การทดลองหุ่นยนต์เคลื่อนที่เป็นเส้นโค้ง ทั้งการเลี้ยวซ้ายและเลี้ยวขวา ผลการทดลองค่าเส้นรัศมีการเลี้ยวโค้งของหุ่นยนต์ที่ได้จากการวัดเปรียบเทียบค่าจากการคำนวณโมเดลระบบการควบคุมการเคลื่อนที่การเลี้ยวโค้งของหุ่นยนต์ พบว่ามีค่าความคลาดเคลื่อนที่ยอมรับได้ ทั้งนี้ค่าความคลาดเคลื่อนที่อาจเกิดจากหลายปัจจัย เช่น ความคลาดเคลื่อนจากการติดตั้งอุปกรณ์ ของล้อซ้ายขวาไม่เท่ากัน ทำให้แรงที่ลงของล้อซ้ายและขวาอาจจะไม่เท่ากัน โดยนำค่า pwm ที่เหมาะสมไปกำหนดการเลี้ยวโค้งซ้ายและเลี้ยวโค้งขวาในโปรแกรมของหุ่นยนต์ 3. การทดลองให้ผู้ใช้ควบคุมหุ่นยนต์ให้เคลื่อนที่ไป-กลับ อ้อมระหว่างกรวยซึ่งห่างกัน 5 โดยผู้ใช้อาจบังคับไปยังจุดต่าง ๆ ที่มีทั้งการเคลื่อนที่ทางตรง และมีการเลี้ยวโค้ง ผลการทดลองพบว่าผู้ใช้งานคนแรกใช้เวลาประมาณ 86.2 วินาที หรือความเร็วเฉลี่ยประมาณ 11.61 cm/s ต่อมาคนที่สองใช้เวลาประมาณ 100 วินาที หรือความเร็วเฉลี่ยประมาณ 10.01 cm/s และคนสุดท้ายใช้เวลาประมาณ 90 วินาที หรือความเร็วเฉลี่ยประมาณ 11.13 cm/s ซึ่งส่วนเบี่ยงเบนมาตรฐานของการทดสอบผู้ใช้งานทั้งสามคนมีค่าออกมาน้อย แสดงว่ากลุ่มข้อมูลมีการกระจายที่น้อยข้อมูลส่วนใหญ่อยู่ใกล้ค่าเฉลี่ยมาก

การทดลองส่วนของแขนหุ่นยนต์ เป็นการประยุกต์นำเอาทฤษฎีจลนศาสตร์แบบไปข้างหน้า และจลนศาสตร์ผกผัน เพื่อมาวิเคราะห์หาพื้นที่ทำงานของแขนหุ่นยนต์ อีกทั้งยังพัฒนาซอฟต์แวร์ให้การควบคุมแขนหุ่นยนต์ได้ง่ายขึ้น ซึ่งเป็นระบบที่มีชื่อว่า การควบคุมการเคลื่อนที่เชิงเส้นแบบหลายแกน (Multi-axis linear motion control -MALM Control) จากการทดลองแรก การเคลื่อนที่ของแขนหุ่นยนต์ทั้ง 3 การทดลองคือ คือ เคลื่อนที่เชิงเส้นตามแนวหน้า-หลัง (ตามแนวแกน $+x$, $-x$), แนวขึ้น-ลง (ตามแนวแกน $+y$, $-y$) และตามขอบเขตการทำงานของแขนหุ่นยนต์ ผลการทดลองมีจุดสังเกตที่เหมือนกันคือ การเคลื่อนที่ของแขนหุ่นยนต์ทั้งขาไปและขากลับ ไม่เรียบสม่ำเสมอ ซึ่งเกิดจากการโครงสร้างของแขนหุ่นยนต์ส่วนข้อมือตามมีจุดขยับที่มาก ต่อมาการเคลื่อนที่ใกล้กับขอบเขตพื้นที่ทำงาน เกิดจากแรงบิดของมอเตอร์ส่วนหัวไหล่ และข้อศอกไม่เพียงพอต่อการกลับไปสู่ตำแหน่งเดิม และข้อจำกัดของโครงสร้างแขนหุ่นยนต์ และการเคลื่อนที่ขากลับจะเกิดการกระชาก เนื่องจากมอเตอร์ส่วนหัวไหล่ และข้อศอกพยายามดึงกลับมายังตำแหน่งของโปรแกรม ซึ่งผลการทดลองพบว่า ส่วนปลายแขนหุ่นยนต์ สามารถเคลื่อนที่ในแนวหน้า-หลัง ($+x$, $-x$), แนวขึ้น-ลง ($+y$, $-y$) ได้ตามขอบเขตที่กำหนด ต่อมาเป็นการทดลองควบคุมให้ผู้ใช้ควบคุมแขนหุ่นยนต์ผ่านแอปพลิเคชันและมองผ่านจอที่แสดงภาพจากกล้องตามเวลาจริง โดยให้หยิบจับสิ่งของจากจุดหนึ่งไปไว้อีกจุดหนึ่ง จากผลการทดลองคือ การควบคุมแบบที่ละข้อต่อแบบอิสระจะใช้เวลามากโดยเฉพาะรอบแรก ซึ่งผู้ใช้ควบคุมได้อย่างลำบาก เนื่องจากผู้ใช้อาจต้องควบคุมผ่านการมองทางหน้าจอของโทรศัพท์เท่านั้น ทำให้ไม่รู้ว่าต้องปรับมุมแต่ละข้อต่อไปที่ตำแหน่งไหนบ้าง ส่วนการควบคุมการควบคุมแบบเคลื่อนที่เชิงเส้นแบบหลายแกนที่พัฒนาขึ้นมาในงานวิจัยนี้ ใช้เวลาเฉลี่ยของทั้ง 3 คน ได้น้อยลงถึง 85% ซึ่งการควบคุมแบบนี้ผู้ใช้สามารถควบคุมแขนหุ่นยนต์ โดยมองผ่านจอที่แสดงภาพจากกล้องตามเวลาได้อย่างสะดวกและง่ายขึ้น จากการทดลองให้ผู้ใช้ควบคุมแขนหุ่นยนต์ผ่านแอปพลิเคชัน โดยควบคุมแบบ MALM-Control สรุปได้ว่าสามารถใช้งานได้จริง และไม่มีผลกระทบมากนักจากปัญหาของการทดลองการเคลื่อนที่ของแขนหุ่นยนต์ในระนาบสองมิติ

5.2 ปัญหาและอุปสรรค

1. เนื่องจากต้นทุนในการสร้างหุ่นยนต์ต้นแบบมีข้อจำกัดด้านต้นทุน จึงทำให้วัสดุอุปกรณ์ต่าง ๆ ทั้งด้านคุณภาพและด้านความสวยงามลดหลั่นลงมา
2. โครงสร้างของหุ่นยนต์ ทั้งบางส่วนของหุ่นยนต์เคลื่อนที่ และเกือบทั้งหมดของแขนหุ่นยนต์ ขึ้นรูปด้วยเครื่องพิมพ์สามมิติ (3D Printer) ทำให้ใช้ระยะเวลาในการขึ้นรูปที่นาน บางชิ้นงานอาจมีการปรับแก้จะต้องขึ้นรูปใหม่
3. แขนหุ่นยนต์หุ่นยนต์ยังหยิบจับได้แค่สิ่งของขนาดเล็กและน้ำหนักเบา เนื่องจากกำลังขับของมอเตอร์รับภาระน้ำหนักมากไม่ได้ เนื่องจากงานวิจัยนี้ยังเป็นเพียงต้นแบบที่เน้นไปที่การงานในระดับเบื้องต้น

5.3 ข้อเสนอแนะและแนวทางในการแก้ไข

งานวิจัยนี้มุ่งเน้นสร้างหุ่นยนต์ที่มีราคาประหยัด ทำให้คนไทยสามารถเข้าถึงเทคโนโลยีหุ่นยนต์ได้ง่ายขึ้น ซึ่งสามารถนำไปพัฒนาต่อยอดปรับปรุงให้หุ่นยนต์ทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น โดยปรับปรุงในส่วนต่าง ๆ เช่น การเซ็นเซอร์ควบคุมหุ่นยนต์สามารถใช้งานแบบ Internet of Things (IoT) พัฒนาส่วน Robot localization ให้หุ่นยนต์สามารถเรียนรู้จดจำตำแหน่งภายในห้องได้ ปรับปรุงโครงสร้างของแขนหุ่นยนต์ออกแบบให้มีน้ำหนักที่เบาลง เปลี่ยนมอเตอร์ที่มีแรงกำลังขับสูงขึ้น และนำไปพัฒนาต่อยอดแขนติดตั้งกับหุ่นยนต์เคลื่อนที่ให้มีความสามารถในการหยิบจับสิ่งของที่หลากหลาย อีกทั้งยังสามารถพัฒนาต่อยอดในส่วนของการควบคุมด้วยสมาร์ตโฟน เป็นการสั่งงานด้วยเสียง หรือวิธีการควบคุมอื่น ๆ ได้ อีกทั้งยังสามารถนำไปประยุกต์ใช้แขนหุ่นยนต์ในด้านอื่น ๆ เช่น งานทางด้านเกษตร งานอุตสาหกรรมการผลิต งานสำรวจ งานสำหรับช่วยเหลือในสภาวะโรคระบาดอย่างโรคโควิด นอกจากนี้ควรเพิ่มงานวิจัยด้าน machine learning และด้านเทคโนโลยีการระบุตำแหน่งตัวเองภายในอาคารเพื่อให้ระบบหุ่นยนต์นี้สมบูรณ์มากขึ้น



บรรณานุกรม



บรรณานุกรม

- Arduino indy. (2018). Arduino MEGA 2560. [Image]. Retrieved from <http://www.arduino-indy.com/product/2/arduino-mega-2560>
- Arduino makerzone. (2020). Servo Motor Components. [Image]. Retrieved from <http://www.arduino-makerzone.com/article>
- Charoenseang, S., & Chuakong, B. (2013). Development of Computer Vision System for Food Feeding Robot Arm for the Disabled Person. *Journal of Information Science and Technology (JIST)*, 4(1), 39-48. <https://doi.org/10.14456/jist.2013.5>
- The Board of Investment of Thailand (2020, November 7). Thai population statistics. Retrieved from <https://www.boi.go.th/index.php?page=demographic&language=th>
- Geeetech Wiki. (2015). Geeetech G2 & G2s pro Building instruction. [Image]. Retrieved from <https://www.geeetech.com/wiki/index.php>
- Inventor. (2018). Components and operation of electric motors. [Image]. Retrieved from <https://www.inventor.in.th/>
- Jung, Y., & Bae, J. (2015). Kinematic Analysis of a 5 DOF Upper-limb Exoskeleton with a Tilted and Vertically Translating Shoulder Joint. *IEEE/ASME Transactions on Mechatronics*, 20(3), 1428-1439. <https://doi.org/10.1109/TMECH.2014.2346767>
- Kosiyaporn, P., & Saengkanong, N. (2004). Behavior-based Mobile Robot and Advanced Software Collaboration. Bangkok: King Mongkut's University of Technology North Bangkok
- Lu, X., & Liu, M. (2016). Optimal Design and Tuning of PID-Type Interval Type-2 Fuzzy Logic Controllers for Delta Parallel Robots. *International Journal of Advanced Robotic Systems*, 13(3). <https://doi.org/10.5772/63941>

- Manikpan, S., Kiatsin, S., & Leelasantitham, A. (2010). A Simulation of 6R Industrial Articulated Robot Arm Using Backpropagation Neural Network. *International Conference on Control, Automation and Systems 2010*. (pp. 823-826). <https://doi.org/10.1109/ICCAS.2010.5670125>
- Ministry of Commerce, Department of Intellectual Property 2019. (2019). Robotics for Aging Society. Retrieved from <http://ipthailand.go.th/images/2284/RoboticAnalysis219.pdf>
- Ovaktechnologies. (2016). Stewart Platform. [Image]. Retrieved from <https://ovaktechnologies.com/stewart-platform/>
- Phanomrattanak, B., & Pornsukvitoon, V. (2017). A Study of Forward and Inverse Kinematic for 6-Link Robot Arm (Staubli RX90). *The Journal of King Mongkut's University of Technology North Bangkok*, 27(2), 241-252. <https://doi.org/10.14416/j.kmutnb.2017.03.013>
- Phoosittisak, A., & Sanposh, P. (2006). Two-Link Planar Robot Arm Control Application: Fully Actuated System by Exact Feedback Linearization Method Case. Bangkok. Kasetsart University. Retrieved from <https://doi.org/10.3390/s17092073>
- Prewthaisong, K. (2013). The Mobile Robot System Development with Fuzzy Logic. *Engineering Journal of Research and Development*, 24(2), 29-35. <https://www.tci-thaijo.org/index.php/eit-researchjournal/article/download/109395/86066/>
- Qian, J., Zi, B., Wang, D., Ma, Y., & Zhang, D. (2017) The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System. *Sensors* 2017, 17(9), 2073. <https://doi.org/10.3390/s17092073>
- Rahul, K., Raheman, H., & Paradkar, V. (2019). Design and development of a 5R 2DOF parallel robot arm for handling paper pot seedlings in a vegetable

- transplanter. *Computers and Electronics in Agriculture*, 166,105014.
<https://doi.org/10.1016/j.compag.2019.105014>
- Robotsiam. (2016). Arduino UNO R3. [Image]. Retrieved from
<https://www.robotsiam.com/product/2/arduino-uno-r3>
- RS Components. (2017). Raspberry Pi 3 B+. [Image]. Retrieved from <https://th-rs-online.com/web/p/raspberry-pi/1373331/>
- Srisawan, C. (2007). *The Development of Mechanical 5 Axis for Commerce*. Bangkok: Chandrakasem Rajabhat University
- Staretu, I. (2021). Robotic Arms with Anthropomorphic Grippers for Robotic Technological Processes. *Proceedings*, 63(1).
<https://doi.org/10.3390/proceedings2020063077>
- Suebsamran, A. (2007). *Development of a Mobile Robot based on Sensory Information for Autonomous Obstacle Avoidance*. Bangkok: King Mongkut's University of Technology North Bangkok
- Sumipol. (2020). know the type of industrial robot and related applications. [Image]. Retrieved from <https://www.sumipol.com/knowledge/types-of-industrial-robot/>
- Thaieasyelec. (2017). How to control RC Servo Motor. [Image]. Retrieved from <https://blog.thaieasyelec.com/example-project-how-to-control-rc-servo-motor-with-arduino/>
- Thestempedia. (2017). Pulse Width Modulation. [Image]. Retrieved from <https://thestempedia.com/tutorials/eve-analog-output-pwm/>
- Tomoya, A., Nakayama, S., Hoshina, A., & Sugaya, M. (2017). A Mobile Robot for Following, Watching and Detecting Falls for Elderly Care. *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st*

International Conference, 112. 1994–2003.

<https://doi.org/10.1016/j.procs.2017.08.125>

Trakankul, C. (1999). Design and Construction of Small Five-axis Robot Arm. Bangkok: Rajamangala University of Technology Phra Nakhon.

Tuchinda, W. (2016). Industrial robot analysis and control. Bangkok: Chulalongkorn University Press.

Waywut, W. (2008). A Designed Object Tracked by Mobile Robot using Image Processing A Designed Object Tracked by Mobile Robot using Image Processing. Bangkok: Rajamangala University of Technology Phra Nakhon.

Zar, M. T., & Ei, W. P. (2017). Point to Point Trajectory Control of 3R Planar Robot Arm. International Journal of Science, Engineering and Technology Research(IJSETR), 6(11), 2278 -7798. <http://ijsetr.org/wp-content/uploads/2017/11/IJSETR-VOL-6-ISSUE-11-1435-1441.pdf>

ภาคผนวก ก

ผลการทดลองการเคลื่อนที่เป็นเส้นตรงของหุ่นยนต์

การทดสอบให้หุ่นยนต์เคลื่อนที่เป็นเส้นตรง ในห้องทดสอบ ให้มีการเคลื่อนที่ทั้งเดินหน้า และถอยหลัง โดยการปรับค่า PWM ของล้อทั้งสองข้าง เริ่มจาก 50 PWM จนถึง 250 PWM โดยเพิ่ม ครั้งละ 50 PWM เคลื่อนที่เป็นเส้นตรงที่ระยะ 5 m แบ่งการทดลองโดยปราศจากน้ำหนักและทดลอง ด้วยมีภาระน้ำหนัก โดยการเพิ่มภาระน้ำหนักให้ตัวหุ่นยนต์เป็น 10 kg และ 20 kg ทดลองเป็นจับ เวลา โดยทดลองซ้ำ 5 ครั้ง เพื่อหาค่าแล้วค่าเบี่ยงเบนมาตรฐาน แสดงผลการทดลองดังตารางที่ 12 ถึงตารางที่ 17 ซึ่งการทดสอบนี้เป็นการทดสอบความเร็วในการเคลื่อนที่ ในทิศทางตรงไปข้างหน้า และเพื่อแยกปัจจัยด้านน้ำหนัก

ตาราง 12 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินไปข้างหน้า ระยะทาง 5 m ไม่มีภาระ น้ำหนักเพิ่มเติม

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 0 kg					ค่าเบี่ยงเบน มาตรฐาน	ความเร็ว เฉลี่ย (m/s)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5		
50	71.5	70.5	72.1	71.1	69.8	0.89	0.07
100	31.6	32.3	32.5	31.8	31.8	0.38	0.16
150	21.4	20.2	21.5	20.3	21.1	0.61	0.24
200	15.6	16.2	14.7	15.3	15.2	0.55	0.32
250	11.7	12.2	12.3	12.1	12.2	0.23	0.41

ตาราง 13 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินถอยหลัง ระยะทาง 5 m ไม่มีภาระ น้ำหนักเพิ่มเติม

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 0 kg					ค่าเบี่ยงเบน มาตรฐาน	ความเร็ว เฉลี่ย (m/s)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5		
50	72.8	72.1	71.8	72.5	73.8	0.77	0.07
100	34.7	33.3	31.8	32.3	32.4	1.14	0.15
150	22.4	20.7	20.3	21.7	20.4	0.91	0.24
200	15.3	15.8	15.2	16.1	15.6	0.37	0.32
250	12.9	11.7	12.7	12.4	12.3	0.46	0.40

ตาราง 14 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินหน้า ระยะทาง 5 m เพิ่มน้ำหนักที่ 10 kg

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 0 kg					ค่าเบี่ยงเบน มาตรฐาน	ความ เร็ว เฉลี่ย (m/s)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5		
50	82.3	80.7	83.2	82.5	80.8	1.10	0.06
100	34.5	33.2	35.1	34	33.2	0.83	0.15
150	20.8	22.1	21.5	21.2	22.9	0.82	0.23
200	16.2	15.5	14.9	16.6	15.8	0.65	0.32
250	12.6	12.3	12.5	11.7	12.9	0.45	0.40

ตาราง 15 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินถอยหลัง ระยะทาง 5 m เพิ่มน้ำหนักที่ 10 kg

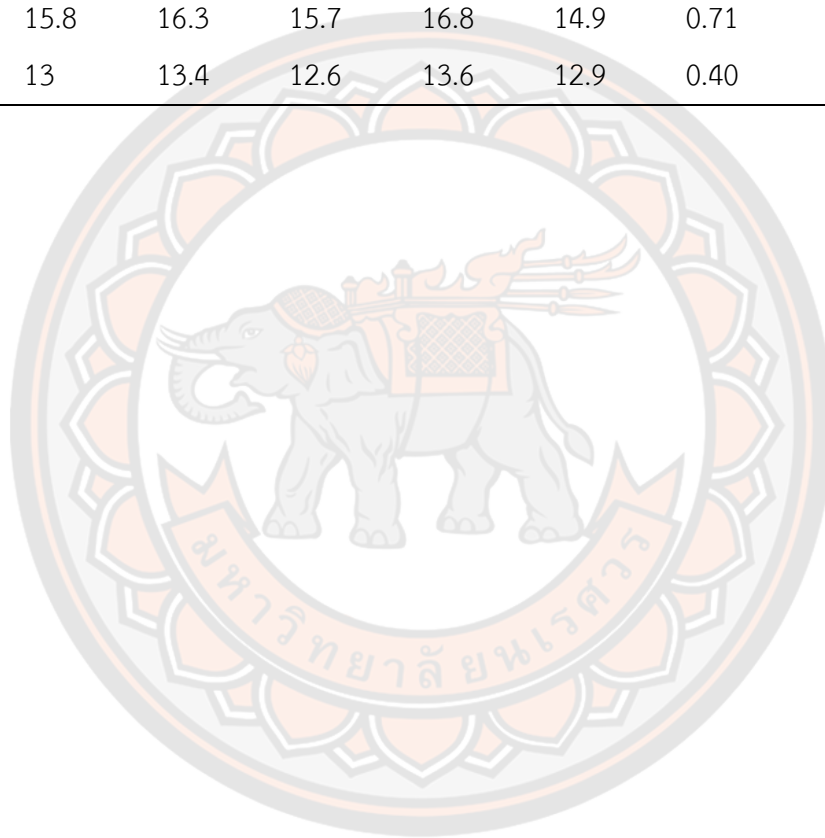
PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 0 kg					ค่าเบี่ยงเบน มาตรฐาน	ความ เร็ว เฉลี่ย (m/s)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5		
50	84.8	83.8	85.1	84.7	84.1	0.53	0.06
100	35.3	34.3	35.6	33.4	34.9	0.87	0.14
150	22.8	23.1	21.7	21.5	21.4	0.79	0.23
200	15.7	16.3	16.5	16.4	15.6	0.42	0.31
250	12.7	12.9	12.4	12.8	13.2	0.29	0.39

ตาราง 16 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินหน้า ระยะทาง 5 m เพิ่มน้ำหนักที่ 20 kg

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 0 kg					ค่าเบี่ยงเบน มาตรฐาน	ความ เร็ว เฉลี่ย (m/s)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5		
50	92.1	92.8	91.9	92.5	93.7	0.71	0.05
100	35	35.4	35.6	35.2	34.8	0.32	0.14
150	23.9	24.3	23.6	22.8	23.9	0.56	0.21
200	15.6	15.3	15.2	16.1	15.3	0.37	0.32
250	12.1	12.6	13.3	11.8	12.2	0.58	0.40

ตาราง 17 การทดสอบหุ่นยนต์เคลื่อนที่เป็นเส้นตรง เดินถอยหลัง ระยะทาง 5 m เพิ่มน้ำหนักที่ 20 kg

PWM	เวลาเฉลี่ย (s) ที่ภาระน้ำหนัก 0 kg					ค่าเบี่ยงเบน มาตรฐาน	ความ เร็ว เฉลี่ย (m/s)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5		
50	95.8	94.5	95.2	95.6	96.4	0.71	0.05
100	37.2	36.5	37.4	38.3	37.1	0.65	0.13
150	21.8	22.3	21.4	22.5	21.5	0.48	0.23
200	15.8	16.3	15.7	16.8	14.9	0.71	0.31
250	13	13.4	12.6	13.6	12.9	0.40	0.38



ภาคผนวก ข

โปรแกรมการควบคุมการเคลื่อนที่ของหุ่นยนต์

การควบคุมการเคลื่อนที่ของหุ่นยนต์ ทางผู้วิจัยได้เลือกใช้บอร์ด microcontroller Arduino มาใช้สำหรับการควบคุมระบบการเคลื่อนที่ของหุ่นยนต์ ในการควบคุมระบบจำเป็นต้องมีการเขียนโปรแกรมภาษา c/c++ สำหรับการควบคุม โดยใช้ Arduino ide ในการเขียน code แสดงรายละเอียดต่าง ๆ ดังนี้

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(12, 13); // TX | RX // สำหรับรับค่าจาก bluetooth

int Motor_Left = 24;
int Speed_Motor_Left = 9;
int Motor_Right = 25;
int Speed_Motor_Right = 10;
int Horn = 52;
int Speed = 0;

//Ultrasonic1

#define Trig1_PIN 6 // Pin connect to Trig pin

#define Echo1_PIN 7 // Pin connect to Echo pin

//Ultrasonic1

#define Trig2_PIN 8 // Pin connect to Trig pin

#define Echo2_PIN 11 // Pin connect to Echo pin
```

```
//Encode

#define encoder0PinA 2

#define encoder0PinB 4

#define encoder0PinC 3

#define encoder0PinD 5

volatile long encoder0Pos = 0;

volatile long encoder0Pos2 = 0;

long newposition;

long oldposition = 0;

long newtime;

long oldtime = 0;

long newposition2;

long oldposition2 = 0;

long newtime2;

long oldtime2 = 0;

long velo = 0;

double newpos = 0;

double oldpos = 0;

long deltatime = 0;

long velo2 = 0;

double newpos2 = 0;
```



```
double oldpos2 = 0;

long deltatime2 = 0;

int counter = 0;

int aState;

int aLastState;

int counter2 = 0;

int aState2;

int aLastState2;

void setup() {

  Serial.begin(9600); // ความเร็วในการสื่อสารสำหรับคอมฯ
  BTSerial.begin(9600); // ความเร็วในการสื่อสารสำหรับ บลูทูธ
  pinMode(Motor_Left, OUTPUT); // กำหนด pin ให้ทำงานแบบ Output
  //pinMode(Motor_Left_reward, OUTPUT); // กำหนด pin ให้ทำงานแบบ Output
  pinMode(Speed_Motor_Left, OUTPUT); // กำหนด pin ให้ทำงานแบบ Output
  pinMode(Motor_Right, OUTPUT); // กำหนด pin ให้ทำงานแบบ Output
  //pinMode(Motor_Right_reward, OUTPUT); // กำหนด pin ให้ทำงานแบบ Output
  pinMode(Speed_Motor_Right, OUTPUT); // กำหนด pin ให้ทำงานแบบ Output
  pinMode(Horn, OUTPUT); // กำหนด pin ให้ทำงานแบบ Output

  //ultrasonic

  pinMode(Trig1_PIN, OUTPUT);
```

```
pinMode(Echo1_PIN, INPUT);

pinMode(Trig2_PIN, OUTPUT);

pinMode(Echo2_PIN, INPUT);

//Encoder

pinMode(encoder0PinA, INPUT_PULLUP);

pinMode(encoder0PinB, INPUT_PULLUP);

pinMode(encoder0PinC, INPUT_PULLUP);

pinMode(encoder0PinD, INPUT_PULLUP);

attachInterrupt(0, doEncoder, CHANGE);

attachInterrupt(1, doEncoder2, CHANGE);

aLastState = digitalRead(encoder0PinA);

aLastState2 = digitalRead(encoder0PinC);

}

void loop()

{

// รับค่าจาก bluetooth

if (BTSerial.available() > 0) // ตรวจสอบว่า bluetooth มีการส่งค่ามาหรือไม่

{

char Control = BTSerial.read(); // อ่านค่าจาก bluetooth มาเก็บไว้ที่ ตัวแปร Control
```

```

Serial.println(Control);          // แสดงค่า จาก bluetooth (ตัวแปร Control) ทาง Serial
Monitor

if (Control == 'F')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'F'

    Forward();                   // หุ่นยนต์เดินหน้า

if (Control == 'B')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'B'

    Reward();                   // หุ่นยนต์ถอยหลัง

if (Control == 'I')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'I'

    Turn_right();               // หุ่นยนต์เลี้ยวขวา

if (Control == 'G')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'G'

    Turn_left();                // หุ่นยนต์เลี้ยวซ้าย

if (Control == 'R')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'R'

    Spin_right();               // หุ่นยนต์หมุนขวา

if (Control == 'L')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'L'

    Spin_left();                // หุ่นยนต์หมุนซ้าย

if (Control == 'H')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'H'

    Reward_turn_left();         // หุ่นยนต์ถอยหลังและเลี้ยวซ้าย

if (Control == 'J')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'J'

    Reward_turn_right();        // หุ่นยนต์ถอยหลังและเลี้ยวขวา

if (Control == 'S')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'S'

    Stop();                     // หุ่นยนต์หยุดเดิน

if (Control == 'V')              // เมื่อค่าที่ bluetooth ส่งมาเป็น 'V'

```

```

digitalWrite(Horn, HIGH);    // แตรดัง

if (Control == 'v')        // เมื่อค่าที่ bluetooth ส่งมาเป็น 'v'

    digitalWrite(Horn, LOW);    // แตรหยุดดัง

    if (Control == '1' || Control == '2' || Control == '3' || Control == '4' || Control == '5' ||
        Control == '6' || Control == '7' || Control == '8' || Control == '9' || Control == 'q') // เมื่อ
        bluetooth ส่งค่า 1-9 และ q มาก

    {

        if (Control == 'q')    // เมื่อค่าที่ bluetooth ส่งมาเป็น 'q'

        {

            Speed = 220;        // กำหนดความเร็ว = 220

        }

        else if (Control == '9') // เมื่อค่าที่ bluetooth ส่งมาเป็น '9'

        {

            Speed = 200;        // กำหนดความเร็ว = 200

        }

        else if (Control == '8') // เมื่อค่าที่ bluetooth ส่งมาเป็น '8'

        {

            Speed = 180;        // กำหนดความเร็ว = 180

        }

        else if (Control == '7') // เมื่อค่าที่ bluetooth ส่งมาเป็น '7'

        {

            Speed = 160;        // กำหนดความเร็ว = 160

```

```
}  
  
else if (Control == '6') // เมื่อค่าที่ bluetooth ส่งมาเป็น '6'  
{  
  
    Speed = 140;        // กำหนดความเร็ว = 140  
  
}  
  
else if (Control == '5') // เมื่อค่าที่ bluetooth ส่งมาเป็น '5'  
{  
  
    Speed = 120;        // กำหนดความเร็ว = 120  
  
}  
  
else if (Control == '4') // เมื่อค่าที่ bluetooth ส่งมาเป็น '4'  
{  
  
    Speed = 100;        // กำหนดความเร็ว = 100  
  
}  
  
else if (Control == '3') // เมื่อค่าที่ bluetooth ส่งมาเป็น '3'  
{  
  
    Speed = 80;         // กำหนดความเร็ว = 80  
  
}  
  
else if (Control == '2') // เมื่อค่าที่ bluetooth ส่งมาเป็น '2'  
{  
  
    Speed = 60;         // กำหนดความเร็ว = 60  
  
}
```

```

else if (Control == '1') // เมื่อค่าที่ bluetooth ส่งมาเป็น '1'

{
    Speed = 40;          // กำหนดความเร็ว = 40

}

}

}

Ultrasonic ();

}

void Forward() // ฟังก์ชันสั่งให้หุ่นยนต์เดินหน้า
{
    // สั่งให้มอเตอร์ซ้ายหมุนไปหน้า
    analogWrite(Speed_Motor_Left, Speed-20); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed
    digitalWrite(Motor_Left, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนไปหน้า
    //digitalWrite(Motor_Left_reward, LOW);

    // สั่งให้มอเตอร์ขวาหมุนไปหน้า

    analogWrite(Speed_Motor_Right, Speed); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร Speed
    digitalWrite(Motor_Right, HIGH); // สั่งให้มอเตอร์ขวาหมุนไปหน้า
    //digitalWrite(Motor_Right_reward, LOW);

    Serial.println("Forward");          // แสดงค่าที่ serial monitor ว่าหุ่นยนต์กำลังเดินหน้า
}

```



```

void Reward() // ฟังก์ชันสั่งให้หุ่นยนต์ถอยหลัง
{
    // สั่งให้มอเตอร์ซ้ายหมุนไปกลับหลัง

    analogWrite(Speed_Motor_Left, Speed-20); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
    Speed

    digitalWrite(Motor_Left, LOW);

    //digitalWrite(Motor_Left_reward, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนกลับหลัง

    // สั่งให้มอเตอร์ขวาหมุนกลับหลัง
    analogWrite(Speed_Motor_Right, Speed); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร Speed
    digitalWrite(Motor_Right, LOW);

    //digitalWrite(Motor_Right_reward, HIGH); // สั่งให้มอเตอร์ขวาหมุนกลับหลัง

    Serial.println("Reward"); // แสดงค่าที่ serial monitor ว่าหุ่นยนต์กำลังถอยหลัง
}

void Turn_left() // ฟังก์ชันสั่งให้หุ่นยนต์เลี้ยวซ้าย
{
    // สำหรับคำนวณความเร็วในการเลี้ยวซ้าย

    int Speed_right = Speed + (Speed / 2);

    if (Speed_right > 255)

        Speed_right = 250;

    analogWrite(Speed_Motor_Left, Speed - 35); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
    Speed

```

```

digitalWrite(Motor_Left, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนไปหน้า

//digitalWrite(Motor_Left_reward, LOW);

analogWrite(Speed_Motor_Right, Speed_right); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed_right

digitalWrite(Motor_Right, HIGH); // สั่งให้มอเตอร์ขวาหมุนไปหน้า

//digitalWrite(Motor_Right_reward, LOW);

Serial.println("Turn_left"); // แสดงค่าที่ serial monitor ว่าหุ่นยนต์กำลังเลี้ยวซ้าย
}

void Turn_right() // ฟังก์ชันสั่งให้หุ่นยนต์เลี้ยวขวา
{
// สำหรับคำนวณความเร็วในการเลี้ยวขวา
int Speed_left = Speed + (Speed / 2);
if (Speed_left > 255)
    Speed_left = 250;

analogWrite(Speed_Motor_Left, Speed_left); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed_left

digitalWrite(Motor_Left, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนไปหน้า

//digitalWrite(Motor_Left_reward, LOW);

analogWrite(Speed_Motor_Right, Speed - 35); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed

digitalWrite(Motor_Right, HIGH); // สั่งให้มอเตอร์ขวาหมุนไปหน้า

// digitalWrite(Motor_Right_reward, LOW);

```

```

Serial.println("Turn_right");          // แสดงค่าที่ serial monitor ว่าหุ่นยนต์กำลังเลี้ยวขวา
}

void Reward_turn_left() // ฟังก์ชันสั่งให้หุ่นยนต์ถอยหลัง
{
    int Speed_trun_left = Speed + (Speed / 2);

    if (Speed_trun_left > 255)

        Speed_trun_left = 250;

    // สั่งให้มอเตอร์ซ้ายหมุนไปกลับหลัง
    analogWrite(Speed_Motor_Left, Speed - 35); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
    Speed
    digitalWrite(Motor_Left, LOW);
    //digitalWrite(Motor_Left_reward, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนกลับหลัง
    // สั่งให้มอเตอร์ขวาหมุนกลับหลัง
    analogWrite(Speed_Motor_Right, Speed_trun_left); // กำหนดความเร็วมอเตอร์ ด้วยตัว
   แปร Speed_trun_left

    digitalWrite(Motor_Right, LOW);

    //digitalWrite(Motor_Right_reward, HIGH); // สั่งให้มอเตอร์ขวาหมุนกลับหลัง

    Serial.println("Reward_turn_left");          // แสดงค่าที่ serial monitor ว่าหุ่นยนต์กำลัง
    ถอยหลัง
}

void Reward_turn_right() // ฟังก์ชันสั่งให้หุ่นยนต์ถอยหลัง

```

```

{

int Speed_trun_right = Speed + (Speed / 2);

if (Speed_trun_right > 255)

    Speed_trun_right = 250;

// สั่งให้มอเตอร์ซ้ายหมุนไปกลับหลัง

analogWrite(Speed_Motor_Left, Speed_trun_right); // กำหนดความเร็วมอเตอร์ ด้วยตัว
แปร Speed_trun_right

digitalWrite(Motor_Left, LOW);

//digitalWrite(Motor_Left_reward, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนกลับหลัง

// สั่งให้มอเตอร์ขวาหมุนกลับหลัง

analogWrite(Speed_Motor_Right, Speed - 35); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed

digitalWrite(Motor_Right, LOW);

//digitalWrite(Motor_Right_reward, HIGH); // สั่งให้มอเตอร์ขวาหมุนกลับหลัง

Serial.println("Reward_turn_right"); // แสดงค่าที่ serial monitor ว่าหุ่นยนต์
กำลังถอยหลัง

}

void Spin_left() // ฟังก์ชันสั่งให้หุ่นยนต์หมุนซ้าย

{

    analogWrite(Speed_Motor_Left, Speed); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
    Speed

    digitalWrite(Motor_Left, LOW);

```

```

//digitalWrite(Motor_Left_reward, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนกลับหลัง

analogWrite(Speed_Motor_Right, Speed); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed

digitalWrite(Motor_Right, HIGH); // สั่งให้มอเตอร์ขวาหมุนไปหน้า

//digitalWrite(Motor_Right_reward, LOW);

Serial.println("Spin_left"); // แสดงค่าที่ serial monitor ว่าหุ่นยนต์กำลังหมุนซ้าย
}

void Spin_right() // ฟังก์ชันสั่งให้หุ่นยนต์หมุนขวา
{

analogWrite(Speed_Motor_Left, Speed); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed

digitalWrite(Motor_Left, HIGH); // สั่งให้มอเตอร์ซ้ายหมุนไปหน้า

//digitalWrite(Motor_Left_reward, LOW);

analogWrite(Speed_Motor_Right, Speed); // กำหนดความเร็วมอเตอร์ ด้วยตัวแปร
Speed

digitalWrite(Motor_Right, LOW);

//digitalWrite(Motor_Right_reward, HIGH); // สั่งให้มอเตอร์ขวาหมุนกลับหลัง

Serial.println("Spin_right"); // แสดงค่าที่ serial monitor ว่าหุ่นยนต์กำลังหมุนขวา
}

void Stop() // ฟังก์ชันสั่งให้หุ่นยนต์หยุดเดิน

{

analogWrite(Speed_Motor_Left, 0); // กำหนดความเร็วมอเตอร์ = 0

```

```

digitalWrite(Motor_Left, LOW); // มอเตอร์หยุดหมุน

//digitalWrite(Motor_Left_reward, LOW); // มอเตอร์หยุดหมุน

analogWrite(Speed_Motor_Right, 0); // กำหนดความเร็วมอเตอร์ = 0

digitalWrite(Motor_Right, LOW); // มอเตอร์หยุดหมุน

//digitalWrite(Motor_Right_reward, LOW); // มอเตอร์หยุดหมุน

}

void Ultrasonic () // เซนเซอร์เตือนสิ่งกีดขวาง
{
  ///// Ultrasonic1

  digitalWrite(Trig1_PIN, LOW);
  delayMicroseconds(5);
  digitalWrite(Trig1_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig1_PIN, LOW);

  unsigned int PulseWidth1 = pulseIn(Echo1_PIN, HIGH);

  unsigned int distance1 = PulseWidth1 * 0.0173681;

  ///// Ultrasonic2

  digitalWrite(Trig2_PIN, LOW);

  delayMicroseconds(5);

  digitalWrite(Trig2_PIN, HIGH);

  delayMicroseconds(10);

```



```
digitalWrite(Trig2_PIN, LOW);

unsigned int PulseWidth2 = pulseIn(Echo2_PIN, HIGH);

unsigned int distance2 = PulseWidth2 * 0.0173681;

BTSerial.print(analogRead(distance1));

// BTSerial.print(analogRead(" / "));

// BTSerial.println(analogRead(distance2));

Serial.print(distance1); Serial.print(" ");

Serial.println(distance2);

if (distance1 <= 40 ) {

    digitalWrite(Horn, HIGH);

    analogWrite(Speed_Motor_Left, 100);

    digitalWrite(Motor_Left, HIGH);

    analogWrite(Speed_Motor_Right, 100);

    digitalWrite(Mot.or_Right, HIGH);

    delay(1000);

    Stop();

    Serial.println(distance1);

}

else if ( distance2 <= 40 ) {

    digitalWrite(Horn, HIGH);

    analogWrite(Speed_Motor_Left, 100);
```

```
digitalWrite(Motor_Left, LOW);

analogWrite(Speed_Motor_Right, 100);

digitalWrite(Motor_Right, LOW);

delay(1000);

Stop();

Serial.println(distance2);
}
else if ((distance1 >= 40) || (distance2 >= 40)) {
digitalWrite(Horn, LOW);
}
}
void doEncoder()
{
aState = digitalRead(encoder0PinA);
if (aState != aLastState) {
if (digitalRead(encoder0PinB) != aState) {
encoder0Pos ++;
} else {
encoder0Pos --;
}
}
}
```

```
aLastState = aState;

noInterrupts();

newposition = encoder0Pos;

interrupts();
}

void doEncoder2()
{
  aState2 = digitalRead(encoder0PinC);
  if (aState2 != aLastState2) {
    if (digitalRead(encoder0PinD) != aState2) {
      encoder0Pos2 ++;
    } else {
      encoder0Pos2 --;
    }
  }
}

aLastState2 = aState2;

noInterrupts();

newposition2 = encoder0Pos2;

interrupts();
}
```

ภาคผนวก ค

โปรแกรมการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์

การควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ ทางผู้วิจัยได้เลือกใช้บอร์ด microcontroller Arduino มาใช้สำหรับการควบคุมระบบการเคลื่อนที่หุ่นยนต์ และใช้บอร์ด shield motor drive สำหรับใช้กับ servo motor ในการควบคุมระบบจำเป็นต้องมีการเขียนโปรแกรมภาษา c/c++ สำหรับการควบคุม โดยใช้ Arduino ide ในการเขียน code แสดงรายละเอียดต่าง ๆ ดังนี้

```
#include <SoftwareSerial.h>

#include <Wire.h>

#include <Adafruit_PWMServoDriver.h>

#include <EEPROM.h>

SoftwareSerial Bluetooth(12, 13);

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

int speedDelay = 20;

int index = 0;

int dataIn;

int m = 0;

int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50], servo05SP[50],
servo06SP[50];

const int servo01 = 0;

const int servo02 = 1;

const int servo03 = 2;
```

```
const int servo04 = 3;  
const int servo05 = 4;  
const int servo06 = 5;  
const int servo07 = 6;
```

```
// Servo Degrees
```

```
int Base_Degree;  
int Shoulder_Degree;  
int Shoulder2_Degree;  
int Elbow_Degree;  
int Wrist_Roll_Degree;  
int Wrist_Pitch_Degree;  
int Grip_Degree;
```

```
// Servo Pulse Length
```

```
int Base_Pulse;  
int Shoulder_Pulse;  
int Shoulder2_Pulse;  
int Elbow_Pulse;  
int Wrist_Roll_Pulse;  
int Wrist_Pitch_Pulse;
```

```
int Grip_Pulse;

const int servoMin = 150;

const int servoMax = 590;

const float positionMax = 28;

const float positionMin = 20;//5.1;

const float positionMax_X = 31.70; //39.6;

const float positionMax_Y = 24.10; //34.6;

const float positionMin_X = 15.30; //13.2;//5.1;

const float positionMin_Y = 7.30; //5;//5;

uint8_t servonum = 0;

float Position_X = 20.00; // x position (cm)

float Position_Y = 20.00; // y position (cm)

float length_a1 = 20.00; // lower joint length (cm)

float length_a2 = 20.00; // upper joint length (cm)

float Cth1;

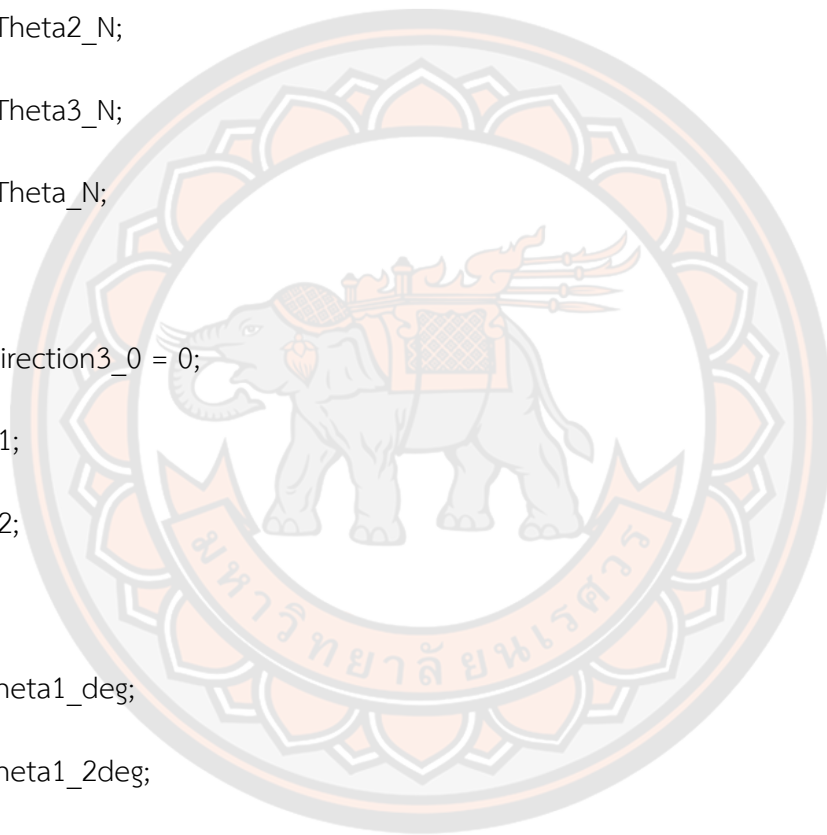
float Cth2;

float C2; // cos theta2

float S2; // sin theta2
```



```
float Theta1;  
  
float Theta1_2;  
  
float Theta2;  
  
float Theta3;  
  
float Theta1_N;  
  
float Theta2_N;  
  
float Theta3_N;  
  
float Theta_N;  
  
float direction3_0 = 0;  
  
float k1;  
  
float k2;  
  
float theta1_deg;  
  
float theta1_2deg;  
  
float theta2_deg;  
  
float theta3_deg;  
  
float theta1_Pulse;  
  
float theta1_2Pulse;  
  
float theta2_Pulse;  
  
float theta3_Pulse;
```



```
// Servo Degrees

float Base_Degree_print;

float Shoulder_Degree_print;

float Shoulder2_Degree_print;

float Elbow_Degree_print;

float Wrist_Roll_Degree_print;

float Wrist_Pitch_Degree_print;

float Grip_Degree_print;

int ModeArm = 1; // เอาไว้สำหรับเป็นโหมดการควบคุมแขน

void setup() {

  Bluetooth.begin(9600);

  Bluetooth.setTimeout(1);

  Serial.begin(9600);

  pwm.begin();

  pwm.setPWMFreq(60);

  Base_Degree = analogRead(servo01);

  Shoulder_Degree = analogRead(servo02);

  Shoulder2_Degree = analogRead(servo07);

  Elbow_Degree = analogRead(servo03);
```

```
Wrist_Roll_Degree = analogRead(servo04);

Wrist_Pitch_Degree = analogRead(servo05);

Grip_Degree = analogRead(servo06);

// Map degree from 0 - 1023, to 0 - 180

Base_Degree = map(Base_Degree, 0, 1023, 0, 180);

Shoulder_Degree = map(Shoulder_Degree, 0, 1023, 0, 180);

Shoulder2_Degree = map(Shoulder2_Degree, 0, 1023, 0, 180);

Elbow_Degree = map(Elbow_Degree, 0, 1023, 0, 180);

Wrist_Roll_Degree = map(Wrist_Roll_Degree, 0, 1023, 0, 180);

Wrist_Pitch_Degree = map(Wrist_Pitch_Degree, 0, 1023, 0, 180);

Grip_Degree = map(Grip_Degree, 0, 1023, 180, 0);

Base_Degree = 90;

Shoulder_Degree = 90;

Shoulder2_Degree = 90;

Elbow_Degree = 140;

Wrist_Roll_Degree = 90;

Wrist_Pitch_Degree = 95;

Grip_Degree = 60;
```

```
// Set Pulse Length

Base_Pulse = map(Base_Degree, 0, 180, servoMin, servoMax);

Shoulder_Pulse = map(Shoulder_Degree, 0, 180, servoMin, servoMax);

Shoulder2_Pulse = map(Shoulder2_Degree, 0, 180, servoMin, servoMax);

Elbow_Pulse = map(Elbow_Degree, 0, 180, servoMin, servoMax);

Wrist_Roll_Pulse = map(Wrist_Roll_Degree, 0, 180, servoMin, servoMax);

Wrist_Pitch_Pulse = map(Wrist_Pitch_Degree, 0, 180, servoMin, servoMax);

Grip_Pulse = map(Grip_Degree, 0, 180, servoMin, servoMax);

pwm.setPWM(servo01, 0, Base_Pulse);

pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);

pwm.setPWM(servo06, 0, Grip_Pulse);

delay(20);

}

void loop() {

Shoulder_Degree_print = map(Shoulder_Pulse, servoMin, servoMax, 0, 180);

Shoulder2_Degree_print = map(Shoulder2_Pulse, servoMin, servoMax, 0, 180);
```

```
Elbow_Degree_print = map(Elbow_Pulse, servoMin, servoMax, 0, 180);

Wrist_Pitch_Degree_print = map(Wrist_Roll_Pulse, servoMin, servoMax, 0, 180);

Serial.print( ModeArm); Serial.print(" , ");

Serial.print( Theta1_N * 57296.00 / 1000.00); Serial.print(" , ");

Serial.print( Theta2_N * 57296.00 / 1000.00); Serial.print(" , ");

Serial.print( Theta3_N * 57296.00 / 1000.00); Serial.print(" , ");

Serial.print( Shoulder_Degree_print); Serial.print(" , ");

Serial.print( Shoulder2_Degree_print); Serial.print(" , ");

Serial.print( Elbow_Degree_print); Serial.print(" , ");

Serial.println( Wrist_Pitch_Degree_print); Serial.print(" , ");

if (Bluetooth.available() > 0) {

    dataIn = Bluetooth.read(); // Read the data

    Serial.println(dataIn);

    if (dataIn == 0) {

        m = 0;

    }

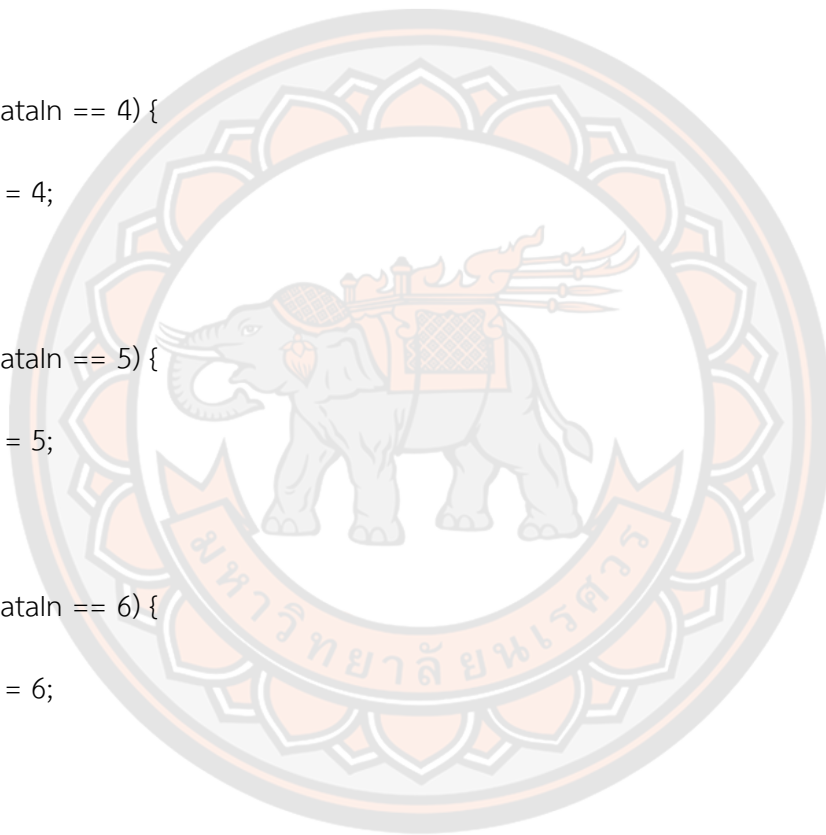
    if (dataIn == 1) {

        m = 1;

    }

}
```

```
if (dataIn == 2) {  
    m = 2;  
}  
  
if (dataIn == 3) {  
    m = 3;  
}  
  
if (dataIn == 4) {  
    m = 4;  
}  
  
if (dataIn == 5) {  
    m = 5;  
}  
  
if (dataIn == 6) {  
    m = 6;  
}  
  
if (dataIn == 7) {  
    m = 7;  
}  
  
if (dataIn == 8) {  
    m = 8;  
}
```




```
if (dataIn == 9) {
```

```
    m = 9;
```

```
}
```

```
if (dataIn == 10) {
```

```
    m = 10;
```

```
}
```

```
if (dataIn == 11) {
```

```
    m = 11;
```

```
}
```

```
if (dataIn == 12) {
```

```
    m = 12;
```

```
}
```

```
if (dataIn == 13) {
```

```
    m = 13;
```

```
}
```

```
if (dataIn == 14) {
```

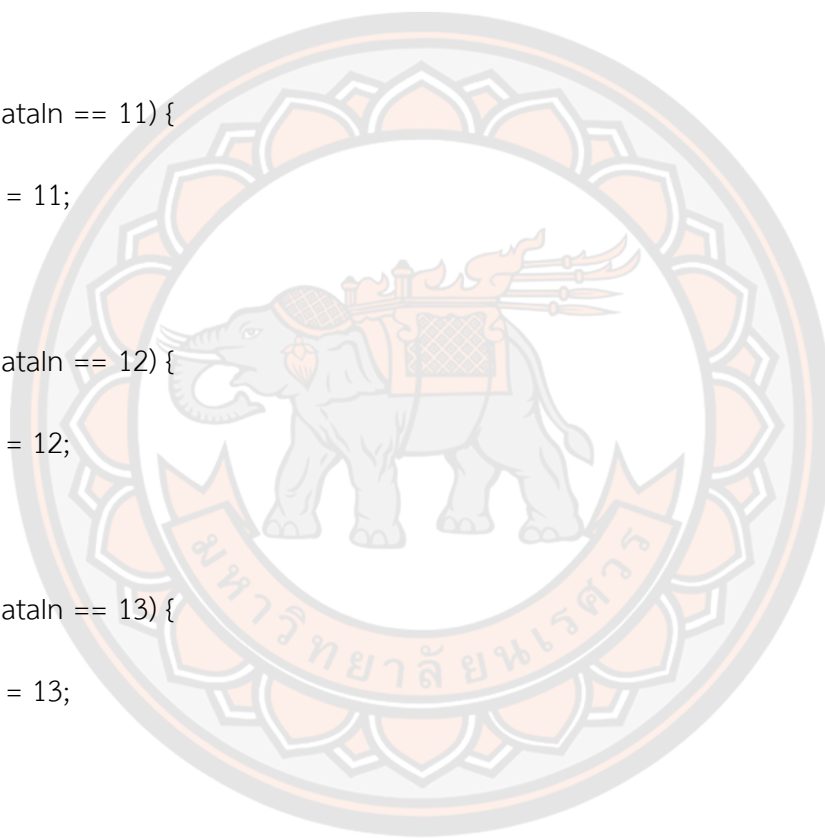
```
    m = 14;
```

```
}
```

```
if (dataIn == 15) {
```

```
    m = 15;
```

```
}
```



```
if (dataIn == 16) {
```

```
    m = 16;
```

```
}
```

```
if (dataIn == 17) {
```

```
    m = 17;
```

```
}
```

```
if (dataIn == 18) {
```

```
    m = 18;
```

```
}
```

```
if (dataIn == 19) {
```

```
    m = 19;
```

```
}
```

```
if (dataIn == 20) {
```

```
    m = 20;
```

```
}
```

```
if (dataIn == 21) {
```

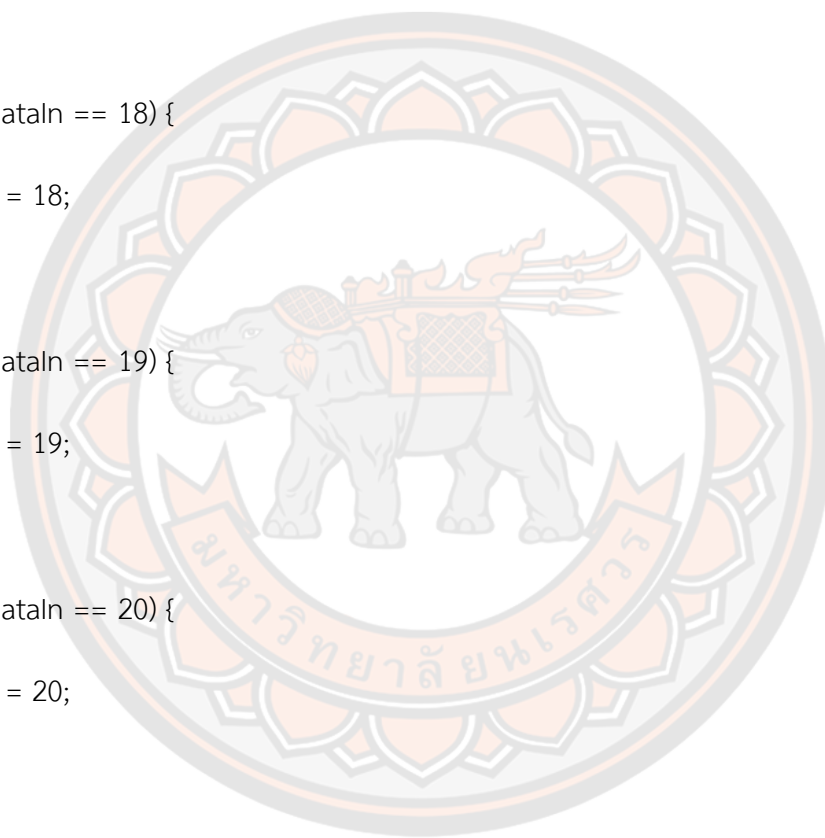
```
    m = 21;
```

```
}
```

```
if (dataIn == 22) {
```

```
    m = 22;
```

```
}
```



```
// Move robot arm

// Move servo 1 in positive direction

while (m == 6) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    ModeArm = 1;

    if (Base_Pulse <= servoMax && Base_Pulse >= servoMin) {

        pwm.setPWM(servo01, 0, Base_Pulse);

        Base_Pulse++;

    }

    else if (Base_Pulse >= servoMax)

    {

        Base_Pulse = servoMax;

        pwm.setPWM(servo01, 0, Base_Pulse);

    }

    else if (Base_Pulse <= servoMin)

    {

        Base_Pulse = servoMin;

        pwm.setPWM(servo01, 0, Base_Pulse);

    }

}
```

```
}  
  
delay(speedDelay);  
}  
  
// Move servo 1 in negative direction  
while (m == 5) {  
  if (Bluetooth.available() > 0) {  
    m = Bluetooth.read();  
  }  
  ModeArm = 1;  
  if (Base_Pulse <= servoMax && Base_Pulse >= servoMin) {  
  
    pwm.setPWM(servo01, 0, Base_Pulse);  
  
    Base_Pulse--;  
  
  }  
  
  else if (Base_Pulse >= servoMax)  
  
  {  
  
    Base_Pulse = servoMax;  
  
    pwm.setPWM(servo01, 0, Base_Pulse);  
  
  }  
}
```

```
else if (Base_Pulse <= servoMin)

{

    Base_Pulse = servoMin;

    pwm.setPWM(servo01, 0, Base_Pulse);

}

delay(speedDelay);

}

// Move servo 2

while (m == 7) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    ModeArm = 1;

    if ((Shoulder_Pulse <= servoMax && Shoulder_Pulse >= servoMin) ||

        (Shoulder2_Pulse <= servoMax && Shoulder2_Pulse >= servoMin)) {

        pwm.setPWM(servo02, 0, Shoulder_Pulse);

        Shoulder_Pulse--;

        pwm.setPWM(servo07, 0, Shoulder2_Pulse);

        Shoulder2_Pulse++;

    }

}
```

```
else if (Shoulder_Pulse >= servoMax && Shoulder2_Pulse >= servoMax)
{
    Shoulder_Pulse = servoMax;
    Shoulder2_Pulse = servoMax;
    pwm.setPWM(servo02, 0, Shoulder_Pulse);
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);
}
else if (Shoulder_Pulse <= servoMin && Shoulder2_Pulse <= servoMin)
{
    Shoulder_Pulse = servoMin;
    Shoulder2_Pulse = servoMin;
    pwm.setPWM(servo02, 0, Shoulder_Pulse);
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);
}
delay(speedDelay);
}

while (m == 8) {
    if (Bluetooth.available() > 0) {
        m = Bluetooth.read();
    }
    ModeArm = 1;
```



```
if ((Shoulder_Pulse <= servoMax && Shoulder_Pulse >= servoMin) ||
(Shoulder2_Pulse <= servoMax && Shoulder2_Pulse >= servoMin)) {

    pwm.setPWM(servo02, 0, Shoulder_Pulse);

    Shoulder_Pulse++;

    pwm.setPWM(servo07, 0, Shoulder2_Pulse);

    Shoulder2_Pulse--;

}

else if (Shoulder_Pulse >= servoMax && Shoulder2_Pulse >= servoMax)
{
    Shoulder_Pulse = servoMax;
    Shoulder2_Pulse = servoMax;
    pwm.setPWM(servo02, 0, Shoulder_Pulse);
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);
}

else if (Shoulder_Pulse <= servoMin && Shoulder2_Pulse <= servoMin)
{
    Shoulder_Pulse = servoMin;
    Shoulder2_Pulse = servoMin;

    pwm.setPWM(servo02, 0, Shoulder_Pulse);

    pwm.setPWM(servo07, 0, Shoulder2_Pulse);

}
```

```
    delay(speedDelay);
}

// Move servo 3
while (m == 10) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    ModeArm = 1;
    if (Elbow_Pulse <= servoMax && Elbow_Pulse >= servoMin) {

        pwm.setPWM(servo03, 0, Elbow_Pulse);

        Elbow_Pulse++;

    }

    else if (Elbow_Pulse >= servoMax)

    {

        Elbow_Pulse = servoMax;

        pwm.setPWM(servo03, 0, Elbow_Pulse);

    }

    else if (Elbow_Pulse <= servoMin)
```

```
{  
  
    Elbow_Pulse = servoMin;  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
}  
  
delay(speedDelay);  
  
}  
  
while (m == 9) {  
    if (Bluetooth.available() > 0) {  
        m = Bluetooth.read();  
    }  
    ModeArm = 1;  
    if (Elbow_Pulse <= servoMax && Elbow_Pulse >= servoMin) {  
        pwm.setPWM(servo03, 0, Elbow_Pulse);  
        Elbow_Pulse--;  
    }  
  
    else if (Elbow_Pulse >= servoMax)  
  
    {  
        Elbow_Pulse = servoMax;  
  
        pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
    }  
  
    else if (Elbow_Pulse <= servoMin)
```

```
{  
  
    Elbow_Pulse = servoMin;  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
}  
  
delay(speedDelay);  
  
}  
  
// Move servo 4  
while (m == 11) {  
  
    if (Bluetooth.available() > 0) {  
  
        m = Bluetooth.read();  
  
    }  
  
    ModeArm = 1;  
  
    if (Wrist_Roll_Pulse <= servoMax && Wrist_Roll_Pulse >= servoMin) {  
  
        pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
  
        Wrist_Roll_Pulse--;  
  
    }  
  
    else if (Wrist_Roll_Pulse >= servoMax)  
  
    {  
  
        Wrist_Roll_Pulse = servoMax;  
  
        pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
  
    }  
  
}
```

```
else if (Wrist_Roll_Pulse <= servoMin)

{

    Wrist_Roll_Pulse = servoMin;

    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

}

delay(speedDelay);

}

while (m == 12) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    ModeArm = 1;

    if (Wrist_Roll_Pulse <= servoMax && Wrist_Roll_Pulse >= servoMin) {

        pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

        Wrist_Roll_Pulse++;

    }

    else if (Wrist_Roll_Pulse >= servoMax)

    {

        Wrist_Roll_Pulse = servoMax;

        pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

    }

}
```

```
else if (Wrist_Roll_Pulse <= servoMin)

{

    Wrist_Roll_Pulse = servoMin;

    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

}

delay(speedDelay);

}

// Move servo 5

while (m == 14) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    ModeArm = 1;

    if (Wrist_Pitch_Pulse <= servoMax && Wrist_Pitch_Pulse >= servoMin) {

        pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);

        Wrist_Pitch_Pulse++;

    }

    else if (Wrist_Pitch_Pulse >= servoMax)

    {

        Wrist_Pitch_Pulse = servoMax;

        pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);

    }

}
```



```
}  
  
else if (Wrist_Pitch_Pulse <= servoMin)  
{  
  
    Wrist_Pitch_Pulse = servoMin;  
  
    pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);  
  
}  
  
delay(speedDelay);  
}  
while (m == 13) {  
    if (Bluetooth.available() > 0) {  
        m = Bluetooth.read();  
    }  
  
    ModeArm = 1;  
  
    if (Wrist_Pitch_Pulse <= servoMax && Wrist_Pitch_Pulse >= servoMin) {  
        pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);  
  
        Wrist_Pitch_Pulse--;  
    }  
  
    else if (Wrist_Pitch_Pulse >= servoMax)  
    {  
  
        Wrist_Pitch_Pulse = servoMax;  
  
        pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);  
    }  
}
```

```
}  
  
else if (Wrist_Pitch_Pulse <= servoMin)  
{  
  
    Wrist_Pitch_Pulse = servoMin;  
  
    pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);  
  
}  
  
delay(speedDelay);  
}  
  
// Move servo 6  
while (m == 16) {  
    if (Bluetooth.available() > 0) {  
        m = Bluetooth.read();  
    }  
  
    ModeArm = 1;  
  
    if (Grip_Pulse <= servoMax && Grip_Pulse >= servoMin) {  
  
        pwm.setPWM(servo06, 0, Grip_Pulse);  
  
        Grip_Pulse++;  
  
    }  
  
    else if (Grip_Pulse >= servoMax)  
  
    {  
  
        Grip_Pulse = servoMax;
```

```
pwm.setPWM(servo06, 0, Grip_Pulse);  
  
}  
  
else if (Grip_Pulse <= servoMin)  
{  
    Grip_Pulse = servoMin;  
    pwm.setPWM(servo06, 0, Grip_Pulse);  
}  
delay(speedDelay);  
}  
while (m == 15) {  
    if (Bluetooth.available() > 0) {  
        m = Bluetooth.read();  
    }  
    ModeArm = 1;  
    if (Grip_Pulse <= servoMax && Grip_Pulse >= servoMin) {  
        pwm.setPWM(servo06, 0, Grip_Pulse);  
        Grip_Pulse--;  
    }  
    else if (Grip_Pulse >= servoMax)  
    {
```

```
Grip_Pulse = servoMax;

pwm.setPWM(servo06, 0, Grip_Pulse);

}

else if (Grip_Pulse <= servoMin)

{

    Grip_Pulse = servoMin;

    pwm.setPWM(servo06, 0, Grip_Pulse);

}

delay(speedDelay);

}

// Function Auto Place
while (m == 17) {

    if (Bluetooth.available() > 0) {

        m = Bluetooth.read();

    }

    ModeArm = 1;

    int s1 = 180;

    int s2 = 90;

    int s7 = 90;

    int s3 = 45;

    int s4 = 88;
```

```
int s5 = 90;

int s6 = 90;

int ss1 = map(s1, 0, 180, servoMin, servoMax);

int ss2 = map(s2, 0, 180, servoMin, servoMax);

int ss7 = map(s7, 0, 180, servoMin, servoMax);

int ss3 = map(s3, 0, 180, servoMin, servoMax);

int ss4 = map(s4, 0, 180, servoMin, servoMax);

int ss5 = map(s5, 0, 180, servoMin, servoMax);

int ss6 = map(s6, 0, 180, servoMin, servoMax);

// s2
if (Shoulder_Pulse < ss2) {
  for (int j = Shoulder_Pulse; j < ss2; j++) {
    pwm.setPWM(servo02, 0, j);

    int i = 590 - j + 150;

    pwm.setPWM(servo07, 0, i);

    Serial.print("A");

    Serial.print(" ");

    Serial.print("i");

    Serial.print(" ");
```

```
Serial.print("j");

Serial.println(" ");

delay(20);

}

}

else if (Shoulder_Pulse > ss2) {

for (int j = Shoulder_Pulse; j > ss2; j--) {

pwm.setPWM(servo02, 0, j);

int i = 590 - j + 150;

pwm.setPWM(servo07, 0, i);

Serial.print("B");

Serial.print(" ");

Serial.print("i");

Serial.print(" ");

Serial.print("j");

Serial.println(" ");

delay(20);

}

}

else {

pwm.setPWM(servo02, 0, ss2);
```

```
pwm.setPWM(servo07, 0, ss7);  
  
delay(20);  
  
}
```

```
//////////-----s3
```

```
if (Elbow_Pulse < ss3) {  
  
    for (int j = Elbow_Pulse; j <= ss3; j++) {  
  
        pwm.setPWM(servo03, 0, j);  
  
        delay(20);  
  
    }  
  
}  
  
else if (Elbow_Pulse > ss3) {  
  
    for (int j = Elbow_Pulse; j >= ss3; j--) {  
  
        pwm.setPWM(servo03, 0, j);  
  
        delay(20);  
  
    }  
  
}  
  
else {  
  
    pwm.setPWM(servo03, 0, ss3);  
  
    delay(20);
```



```
}

//////////-----s4

if (Wrist_Roll_Pulse < ss4) {

  for (int j = Wrist_Roll_Pulse; j <= ss4; j++) {

    pwm.setPWM(servo04, 0, j);

    delay(20);

  }

}

else if (Wrist_Roll_Pulse > ss4) {

  for (int j = Wrist_Roll_Pulse; j >= ss4; j--) {

    pwm.setPWM(servo04, 0, j);

    delay(20);

  }

}

else {

  pwm.setPWM(servo04, 0, ss4);

  delay(20);

}

//////////-----s5
```

```
if (Wrist_Pitch_Pulse < ss5) {  
  
    for (int j = Wrist_Pitch_Pulse; j <= ss5; j++) {  
  
        pwm.setPWM(servo05, 0, j);  
  
        delay(20);  
  
    }  
  
}  
  
else if (Wrist_Pitch_Pulse > ss5) {  
  
    for (int j = Wrist_Pitch_Pulse; j >= ss5; j--) {  
  
        pwm.setPWM(servo05, 0, j);  
  
        delay(20);  
  
    }  
  
}  
  
else {  
  
    pwm.setPWM(servo05, 0, ss5);  
  
    delay(20);  
  
}  
  
//////////-----s1  
  
if (Base_Pulse < ss1) {  
  
    for (int j = Base_Pulse; j <= ss1; j++) {  
  
        pwm.setPWM(servo01, 0, j);  
  
        delay(20);  
  
    }  
  
}
```

```
    }  
}  
  
else if (Base_Pulse > ss1) {  
    for (int j = Base_Pulse; j >= ss1; j--) {  
        pwm.setPWM(servo01, 0, j);  
        delay(20);  
    }  
}  
  
else {  
    pwm.setPWM(servo01, 0, ss1);  
    delay(20);  
}  
  
//////////-----s6  
if (Grip_Pulse < ss6) {  
    for (int j = Grip_Pulse; j <= ss6; j++) {  
        pwm.setPWM(servo06, 0, j);  
        delay(20);  
    }  
}  
  
else if (Grip_Pulse > ss6) {  
    for (int j = Grip_Pulse; j >= ss6; j--) {
```

```
pwm.setPWM(servo06, 0, j);

delay(20);

}

}

else {

    pwm.setPWM(servo06, 0, ss6);

    delay(20);

}

// Set Degree to Potentiometer

Base_Degree = analogRead(servo01); // =ss1

Shoulder_Degree = analogRead(servo02);

Shoulder2_Degree = analogRead(servo07);

Elbow_Degree = analogRead(servo03);

Wrist_Roll_Degree = analogRead(servo04);

Wrist_Pitch_Degree = analogRead(servo05);

Grip_Degree = analogRead(servo06);

// Map degree from 0 - 1023, to 0 - 180

Base_Degree = map(Base_Degree, 0, 1023, 0, 180);

Shoulder_Degree = map(Shoulder_Degree, 0, 1023, 0, 180);

Shoulder2_Degree = map(Shoulder2_Degree, 0, 1023, 0, 180);

Elbow_Degree = map(Elbow_Degree, 0, 1023, 0, 180);
```

```
Wrist_Roll_Degree = map(Wrist_Roll_Degree, 0, 1023, 0, 180);  
  
Wrist_Pitch_Degree = map(Wrist_Pitch_Degree, 0, 1023, 0, 180);  
  
Grip_Degree = map(Grip_Degree, 0, 1023, 0, 180);  
  
Base_Degree = 180;  
  
Shoulder_Degree = 90;  
  
Shoulder2_Degree = 90;  
  
Elbow_Degree = 45;  
  
Wrist_Roll_Degree = 88;  
  
Wrist_Pitch_Degree = 90;  
  
Grip_Degree = 90;  
  
// Set Pulse Length  
  
Base_Pulse = map(Base_Degree, 0, 180, servoMin, servoMax); //ss1  
  
Shoulder_Pulse = map(Shoulder_Degree, 0, 180, servoMin, servoMax);  
  
Shoulder2_Pulse = map(Shoulder2_Degree, 0, 180, servoMax, servoMin);  
  
Elbow_Pulse = map(Elbow_Degree, 0, 180, servoMin, servoMax);  
  
Wrist_Roll_Pulse = map(Wrist_Roll_Degree, 0, 180, servoMin, servoMax);  
  
Wrist_Pitch_Pulse = map(Wrist_Pitch_Degree, 0, 180, servoMin, servoMax);  
  
Grip_Pulse = map(Grip_Degree, 0, 180, servoMin, servoMax);  
  
pwm.setPWM(servo01, 0, Base_Pulse);  
  
pwm.setPWM(servo02, 0, Shoulder_Pulse);
```

```
pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);

pwm.setPWM(servo06, 0, Grip_Pulse);

delay(20);

break;
}

// Set home
while (m == 18) {
  if (Bluetooth.available() > 0) {
    m = Bluetooth.read();
  }
  ModeArm = 1;

  int s1 = 90;

  int s2 = 90;

  int s7 = 90;

  int s3 = 45;

  int s4 = 88;

  int s5 = 95;

  int s6 = 90;
```

```
int ss1 = map(s1, 0, 180, servoMin, servoMax);

int ss2 = map(s2, 0, 180, servoMin, servoMax);

int ss7 = map(s7, 0, 180, servoMax, servoMin);

int ss3 = map(s3, 0, 180, servoMin, servoMax);

int ss4 = map(s4, 0, 180, servoMin, servoMax);

int ss5 = map(s5, 0, 180, servoMin, servoMax);

int ss6 = map(s6, 0, 180, servoMin, servoMax);

//////////-----s1

if (Base_Pulse < ss1) {

  for (int j = Base_Pulse; j <= ss1; j++) {

    pwm.setPWM(servo01, 0, j);

    delay(20);

  }

}

else if (Base_Pulse > ss1) {

  for (int j = Base_Pulse; j >= ss1; j--) {

    pwm.setPWM(servo01, 0, j);

    delay(20);

  }

}

else {
```



```

pwm.setPWM(servo01, 0, ss1);

delay(20);

}

//////////-----s2

if (Shoulder_Pulse < ss2) {

    for (int j = Shoulder_Pulse; j < ss2; j++) {

        pwm.setPWM(servo02, 0, j);

        int i = (590 - j + 150);

        pwm.setPWM(servo07, 0, i);

        delay(20);

    }

}

else if (Shoulder_Pulse > ss2) {

    for (int j = Shoulder_Pulse; j > ss2; j--) {

        pwm.setPWM(servo02, 0, j);

        int i = (590 - j + 150);

        pwm.setPWM(servo07, 0, i);

        delay(20);

    }

}

```

```
else {  
  
    pwm.setPWM(servo02, 0, ss2);  
  
    pwm.setPWM(servo07, 0, ss7);  
  
    delay(20);  
  
}  
  
//////////-----s3  
  
if (Elbow_Pulse < ss3) {  
    for (int j = Elbow_Pulse; j <= ss3; j++) {  
        pwm.setPWM(servo03, 0, j);  
        delay(20);  
    }  
}  
  
else if (Elbow_Pulse > ss3) {  
    for (int j = Elbow_Pulse; j >= ss3; j--) {  
        pwm.setPWM(servo03, 0, j);  
  
        delay(20);  
  
    }  
  
}  
  
else {  
  
    pwm.setPWM(servo03, 0, ss3);  
  
    delay(20);
```

```
}

//////////-----s4

if (Wrist_Roll_Pulse < ss4) {

    for (int j = Wrist_Roll_Pulse; j <= ss4; j++) {

        pwm.setPWM(servo04, 0, j);

        delay(20);

    }

}

else if (Wrist_Roll_Pulse > ss4) {

    for (int j = Wrist_Roll_Pulse; j >= ss4; j--) {

        pwm.setPWM(servo04, 0, j);

        delay(20);

    }

}

else {

    pwm.setPWM(servo04, 0, ss4);

    delay(20);

}

//////////-----s5
```

```
if (Wrist_Pitch_Pulse < ss5) {  
  
    for (int j = Wrist_Pitch_Pulse; j <= ss5; j++) {  
  
        pwm.setPWM(servo05, 0, j);  
  
        delay(20);  
  
    }  
  
}  
  
else if (Wrist_Pitch_Pulse > ss5) {  
  
    for (int j = Wrist_Pitch_Pulse; j >= ss5; j--) {  
  
        pwm.setPWM(servo05, 0, j);  
  
        delay(20);  
  
    }  
  
}  
  
else {  
  
    pwm.setPWM(servo05, 0, ss5);  
  
    delay(20);  
  
}  
  
//////////-----s6  
  
if (Grip_Pulse < ss6) {  
  
    for (int j = Grip_Pulse; j <= ss6; j++) {  
  
        pwm.setPWM(servo06, 0, j);  
  
    }  
  
}
```

```
    delay(20);  
  }  
}  
  
else if (Grip_Pulse > ss6) {  
  for (int j = Grip_Pulse; j >= ss6; j--) {  
    pwm.setPWM(servo06, 0, j);  
    delay(20);  
  }  
}  
else {  
  pwm.setPWM(servo06, 0, ss6);  
  delay(20);  
}  
  
// Set Degree to Potentiometer  
Base_Degree = analogRead(servo01);  
  
Shoulder_Degree = analogRead(servo02);  
  
Shoulder2_Degree = analogRead(servo07);  
  
Elbow_Degree = analogRead(servo03);  
  
Wrist_Roll_Degree = analogRead(servo04);  
  
Wrist_Pitch_Degree = analogRead(servo05);  
  
Grip_Degree = analogRead(servo06);
```

```
// Map degree from 0 - 1023, to 0 - 180

Base_Degree = map(Base_Degree, 0, 1023, 0, 180);

Shoulder_Degree = map(Shoulder_Degree, 0, 1023, 0, 180);

Shoulder2_Degree = map(Shoulder2_Degree, 0, 1023, 0, 180);

Elbow_Degree = map(Elbow_Degree, 0, 1023, 0, 180);

Wrist_Roll_Degree = map(Wrist_Roll_Degree, 0, 1023, 0, 180);

Wrist_Pitch_Degree = map(Wrist_Pitch_Degree, 0, 1023, 0, 180);

Grip_Degree = map(Grip_Degree, 0, 1023, 0, 180);

Base_Degree = 90;

Shoulder_Degree = 90;

Shoulder2_Degree = 90;

Elbow_Degree = 45;

Wrist_Roll_Degree = 88;

Wrist_Pitch_Degree = 95;

Grip_Degree = 90;

// Set Pulse Length

Base_Pulse = map(Base_Degree, 0, 180, servoMin, servoMax);

Shoulder_Pulse = map(Shoulder_Degree, 0, 180, servoMin, servoMax);

Shoulder2_Pulse = map(Shoulder2_Degree, 0, 180, servoMax, servoMin);

Elbow_Pulse = map(Elbow_Degree, 0, 180, servoMin, servoMax);

Wrist_Roll_Pulse = map(Wrist_Roll_Degree, 0, 180, servoMin, servoMax);
```

```

Wrist_Pitch_Pulse = map(Wrist_Pitch_Degree, 0, 180, servoMin, servoMax);

Grip_Pulse = map(Grip_Degree, 0, 180, servoMin, servoMax);

pwm.setPWM(servo01, 0, Base_Pulse);

pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

pwm.setPWM(servo05, 0, Wrist_Pitch_Pulse);

pwm.setPWM(servo06, 0, Grip_Pulse);

delay(20);

break;

}

// Move servo 2 -----
while (m == 19) {

  if (Bluetooth.available() > 0) {

    m = Bluetooth.read();

  }

  if (ModeArm == 1)

  {

    Shoulder_Degree = map(Shoulder_Pulse, servoMin, servoMax, 0, 180);

```



```

Shoulder2_Degree = map(Shoulder2_Pulse, servoMin, servoMax, 0, 180);

Elbow_Degree = map(Elbow_Pulse, servoMin, servoMax, 0, 180);

Wrist_Roll_Degree = map(Wrist_Roll_Pulse, servoMin, servoMax, 0, 180);

Theta_N = Shoulder_Degree;

Theta1_N = Theta_N * 1000.00 / 57296.00;

Theta_N = Elbow_Degree;

Theta2_N = (Theta_N - 140.00) * 1000.00 / 57296.00;

Theta_N = Wrist_Roll_Degree;

Theta3_N = (Theta_N - 90.00) * 1000.00 / 57296.00 ;

//
Position_X = (length_a1 * cos(Theta1_N)) + (length_a2 * cos(Theta1_N +
Theta2_N));

Position_Y = (length_a1 * sin(Theta1_N)) + (length_a2 * sin(Theta1_N +
Theta2_N));

ModeArm = 2;

Serial.print(Position_X); Serial.print(" GG ");

Serial.print(Position_Y); Serial.println(" GG ");

}

if (Position_X <= positionMax_X && Position_X >= positionMin_X) {

Position_Y ;

CalculateServoAngles();

//

```

```
pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

Position_X += 0.2;

}

else if (Position_X >= positionMax_X)

{

Position_X = positionMax_X;

pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

}

else if (Position_X <= positionMin_X)

{

Position_X = positionMin_X;

pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);
```

```
pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
  
}  
  
CalculateServoAngles();  
  
delay(50);  
  
}  
  
while (m == 20) {  
  if (Bluetooth.available() > 0) {  
    m = Bluetooth.read();  
  }  
  if (ModeArm == 1)  
  {  
    Shoulder_Degree = map(Shoulder_Pulse, servoMin, servoMax, 0, 180);  
    Shoulder2_Degree = map(Shoulder2_Pulse, servoMin, servoMax, 0, 180);  
    Elbow_Degree = map(Elbow_Pulse, servoMin, servoMax, 0, 180);  
    Wrist_Roll_Degree = map(Wrist_Roll_Pulse, servoMin, servoMax, 0, 180);  
    Theta_N = Shoulder_Degree;  
    Theta1_N = Theta_N * 1000.00 / 57296.00;  
    Theta_N = Elbow_Degree;  
    Theta2_N = (Theta_N - 140.00) * 1000.00 / 57296.00;  
    Theta_N = Wrist_Roll_Degree;
```



```

Theta3_N = (Theta_N - 90.00) * 1000.00 / 57296.00 ;

//

Position_X = (length_a1 * cos(Theta1_N)) + (length_a2 * cos(Theta1_N +
Theta2_N));

Position_Y = (length_a1 * sin(Theta1_N)) + (length_a2 * sin(Theta1_N +
Theta2_N));

ModeArm = 2;

Serial.print(Position_X); Serial.print(" GG ");

Serial.print(Position_Y); Serial.println(" GG ");
}
if (Position_X <= positionMax_X && Position_X >= positionMin_X) {

Position_Y ;

CalculateServoAngles();

pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

Position_X -= 0.2;

}

else if (Position_X >= positionMax_X)

```

```
{  
  
    Position_X = positionMax_X;  
  
    pwm.setPWM(servo02, 0, Shoulder_Pulse);  
  
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
  
}  
  
else if (Position_X <= positionMin_X)  
{  
  
    Position_X = positionMin_X;  
  
    pwm.setPWM(servo02, 0, Shoulder_Pulse);  
  
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
  
}  
  
CalculateServoAngles();  
  
delay(50);  
  
}  
  
// Move servo 3-----  
  
while (m == 21) {  
  
    if (Bluetooth.available() > 0) {
```

```

    m = Bluetooth.read();

}

if (ModeArm == 1)

{

    Shoulder_Degree = map(Shoulder_Pulse, servoMin, servoMax, 0, 180);

    Shoulder2_Degree = map(Shoulder2_Pulse, servoMin, servoMax, 0, 180);

    Elbow_Degree = map(Elbow_Pulse, servoMin, servoMax, 0, 180);

    Wrist_Roll_Degree = map(Wrist_Roll_Pulse, servoMin, servoMax, 0, 180);

    Theta_N = Shoulder_Degree;

    Theta1_N = Theta_N * 1000.00 / 57296.00;

    Theta_N = Elbow_Degree;

    Theta2_N = (Theta_N - 140.00) * 1000.00 / 57296.00;

    Theta_N = Wrist_Roll_Degree;

    Theta3_N = (Theta_N - 90.00) * 1000.00 / 57296.00 ;

    //

    Position_X = (length_a1 * cos(Theta1_N)) + (length_a2 * cos(Theta1_N +
Theta2_N));

    Position_Y = (length_a1 * sin(Theta1_N)) + (length_a2 * sin(Theta1_N +
Theta2_N));

    ModeArm = 2;

    Serial.print(Position_X); Serial.print(" GG ");

    Serial.print(Position_Y); Serial.println(" GG ");

```

```
}  
  
if (Position_Y <= positionMax_Y && Position_Y >= positionMin_Y) {  
  
    Position_X ;  
  
    CalculateServoAngles();  
  
    pwm.setPWM(servo02, 0, Shoulder_Pulse);  
  
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
  
    Position_Y += 0.2;  
}  
  
else if (Position_Y >= positionMax_Y)  
{  
  
    Position_Y = positionMax_Y;  
  
    pwm.setPWM(servo02, 0, Shoulder_Pulse);  
  
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
  
}  
  
else if (Position_Y <= positionMin_Y)  
  
{
```



```
Position_Y = positionMin_Y;

pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

}

CalculateServoAngles();

delay(50);

}

while (m == 22) {

  if (Bluetooth.available() > 0) {

    m = Bluetooth.read();

  }

  if (ModeArm == 1)

  {

    Shoulder_Degree = map(Shoulder_Pulse, servoMin, servoMax, 0, 180);

    Shoulder2_Degree = map(Shoulder2_Pulse, servoMin, servoMax, 0, 180);

    Elbow_Degree = map(Elbow_Pulse, servoMin, servoMax, 0, 180);

    Wrist_Roll_Degree = map(Wrist_Roll_Pulse, servoMin, servoMax, 0, 180);

    Theta_N = Shoulder_Degree;
```

```

Theta1_N = Theta_N * 1000.00 / 57296.00;

Theta_N = Elbow_Degree;

Theta2_N = (Theta_N - 140.00) * 1000.00 / 57296.00;

Theta_N = Wrist_Roll_Degree;

Theta3_N = (Theta_N - 90.00) * 1000.00 / 57296.00 ;

//

Position_X = (length_a1 * cos(Theta1_N)) + (length_a2 * cos(Theta1_N +
Theta2_N));

Position_Y = (length_a1 * sin(Theta1_N)) + (length_a2 * sin(Theta1_N +
Theta2_N));

ModeArm = 2;

Serial.print(Position_X); Serial.print(" GG ");

Serial.print(Position_Y); Serial.println(" GG ");

}

if (Position_Y <= positionMax_Y && Position_Y >= positionMin_Y) {

Position_X ;

CalculateServoAngles();

pwm.setPWM(servo02, 0, Shoulder_Pulse);

pwm.setPWM(servo07, 0, Shoulder2_Pulse);

pwm.setPWM(servo03, 0, Elbow_Pulse);

pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);

Position_Y -= 0.2;

```

```
}  
  
else if (Position_Y >= positionMax_Y)  
{  
  
    Position_Y = positionMax_Y;  
  
    pwm.setPWM(servo02, 0, Shoulder_Pulse);  
  
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
}  
  
else if (Position_Y <= positionMin_Y)  
{  
  
    Position_Y = positionMin_Y;  
  
    pwm.setPWM(servo02, 0, Shoulder_Pulse);  
  
    pwm.setPWM(servo07, 0, Shoulder2_Pulse);  
  
    pwm.setPWM(servo03, 0, Elbow_Pulse);  
  
    pwm.setPWM(servo04, 0, Wrist_Roll_Pulse);  
}  
  
CalculateServoAngles();  
  
delay(50);  
}  
  
if (dataIn == '1' || dataIn == '2' || dataIn == '3' || dataIn == '4' || dataIn == '5')
```

```
{  
  
  if (dataIn == '5')    // เมื่อค่าที่ bluetooth ส่งมาเป็น '๕'  
  
  {  
  
    speedDelay = 50;      // กำหนดความเร็ว = 220  
  
  }  
  
  else if (dataIn == '4') // เมื่อค่าที่ bluetooth ส่งมาเป็น '๔'  
  
  {  
  
    speedDelay = 40;      // กำหนดความเร็ว = 200  
  
  }  
  
  else if (dataIn == '3') // เมื่อค่าที่ bluetooth ส่งมาเป็น '๓'  
  
  {  
  
    speedDelay = 30;      // กำหนดความเร็ว = 180  
  
  }  
  
  else if (dataIn == '2') // เมื่อค่าที่ bluetooth ส่งมาเป็น '๒'  
  
  {  
  
    speedDelay = 20;      // กำหนดความเร็ว = 160  
  
  }  
  
  else if (dataIn == '1') // เมื่อค่าที่ bluetooth ส่งมาเป็น '๑'  
  
  {  
  
    speedDelay = 10;      // กำหนดความเร็ว = 140  
  
  }  
  
}
```

```
}  
  
Serial.println(speedDelay);  
  
if (m == 1) {  
  
    //if it's initial save, set the steppers position to 0  
  
    servo01SP[index] = Base_Pulse; // save position into the array  
  
    servo02SP[index] = Shoulder_Pulse;  
  
    servo03SP[index] = Elbow_Pulse;  
  
    servo04SP[index] = Wrist_Roll_Pulse;  
  
    servo05SP[index] = Wrist_Pitch_Pulse;  
  
    servo06SP[index] = Grip_Pulse;  
  
    index++; // Increase the array index  
  
    m = 0;  
  
    Serial.println("m1");  
  
}  
  
// If button "RUN" is pressed  
  
if (m == 3) {  
  
    // If button "RESET" is pressed  
  
    if (dataIn != 3) {  
  
        memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0  
  
        memset(servo02SP, 0, sizeof(servo02SP));  
  
        memset(servo03SP, 0, sizeof(servo03SP));  
  
    }  
  
}
```

```

memset(servo04SP, 0, sizeof(servo04SP));

memset(servo05SP, 0, sizeof(servo05SP));

memset(servo06SP, 0, sizeof(servo06SP));

index = 0; // Index to 0

Serial.println("m3");

}

}

}

}

void CalculateServoAngles()
{
//-----Calculate-----

Cth1 = - ( sq(length_a1) + sq(length_a2) - sq(Position_X) - sq(Position_Y) ) ;

Cth2 = 2 * length_a1 * length_a2 ;

C2 = Cth1 / Cth2;

S2 = (sqrt(1 - sq(C2))) * -1;

Theta2 = atan2(S2, C2);

k1 = length_a1 + (length_a2 * C2);

k2 = length_a2 * S2;

Theta1 = atan2(Position_Y, Position_X) - atan2(k2, k1);

Theta3 = direction3_0 - Theta1 - Theta2;

```

```

Shoulder_Degree = (Theta1 * 57296.00 / 1000.00);

Shoulder2_Degree = (Theta1 * 57296.00 / 1000.00);

Elbow_Degree = (Theta2 * 57296.00 / 1000.00) + 140.00; // +135 set ตัวservo + 5
ปรับมุมให้ตรง

Wrist_Roll_Degree = (Theta3 * 57296.00 / 1000.00) + 90.00 ; //+90 set ตัวservo

// Set Pulse Length

Shoulder_Pulse = map(Shoulder_Degree , 0, 180, servoMin, servoMax);

Shoulder2_Pulse = map(Shoulder2_Degree , 0, 180, servoMax, servoMin);

Elbow_Pulse = map(Elbow_Degree , 0, 180, servoMin, servoMax);

Wrist_Roll_Pulse = map(Wrist_Roll_Degree , 0, 180, servoMin, servoMax);

Serial.print(Shoulder_Degree); Serial.print(" :");

Serial.print(Shoulder2_Degree); Serial.print(" :");

Serial.print(Elbow_Degree ); Serial.print(" :");

Serial.print(Wrist_Roll_Degree ); Serial.print(" :");

Serial.print(" : ");

Serial.print(Position_X); Serial.print(" ");

Serial.print(Position_Y); Serial.println(" // ");

// Serial.print(Shoulder_Pulse); Serial.print(" ");

// Serial.print(Shoulder2_Pulse); Serial.print(" ");

// Serial.print(Elbow_Pulse); Serial.print(" ");

// Serial.print(Wrist_Roll_Pulse); Serial.println(" ");

```