การออกแบบและสร้างเครื่องต้นแบบสำหรับการช่วยเหลือและจัดการยาภายในห้องยา

นายสิทธินนท์ จุลนิธิ

มหาวิทยาลัยเทคโนโลยีสุรนารี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเมคคาทรอนิกส์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2563

DESIGN AND PROTOTYPING FOR DRUG SUPPORT AND MANAGEMENT IN DISPENSING ROOMS

Sittinon Julniti

มหาวิทยาลัยเทคโนโลยีสุรนารี

A Thesis Submitted in Partial Fulfillment of the Requirement for the

Degree of Master of Engineering in Mechatronics Engineering

Suranaree University of Technology

Academic Year 2020

การออกแบบและสร้างเครื่องต้นแบบสำหรับการช่วยเหลือและจัดการยาภายในห้องยา

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

(ผศ. ดร. โศรฎา แข็งการ)
ประธานกรรมการ

(รศ. ร.อ. ดร. กนต์ธร ชำนิประศาสน์)
กรรมการ(อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(อ. ดร. พิจิตรา เอื้องไพโรจน์)
กรรมการ

(ดร. สุพัฒน์ กลิ่นเขียว)
กรรมการ

(รศ. ดร. ฉัตรชัย โชติษฐยางกูร)
รักษาการแทนรองอธิการบดีฝ่ายวิชาการและประกันคุณภาพ

(รศ. ดร. พรศิริ จงกล)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

สิทธินนท์ จุลนิธิ : การออกแบบและสร้างเครื่องต้นแบบสำหรับการช่วยเหลือและจัดการยาภายในห้องยา (DESIGN AND PROTOTYPING FOR DRUG SUPPORT AND MANAGEMENT IN DISPENSING ROOMS) อาจารย์ที่ปรึกษา : รองศาสตราจารย์เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์, 65 หน้า

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและสร้างเครื่องต้นแบบสำหรับการช่วยเหลือและจัดการยาภายในห้องจัดยา จากปัญหาการจัดยาให้กับผู้ป่วยที่เข้ารับบริการในโรงพยาบาลใช้เวลานานอันเนื่องมาจากปริมาณยาที่มีอยู่ภายในโรงพยาบาทมีจำนวนมาก ปริมาณยาต่อผู้ป่วย หรือปริมาณของผู้ป่วยที่มีมากในบ้างช่วงเวลา นี้เป็นสาหตุที่ทำเวลาในการจัดยาใช้เวลานานและมีโอกาที่จะผิดพลาดได้ ในงานวิจัยนี้ผู้วิจัยได้ทำการออกแบบและสร้างเครื่องต้นแบบที่ช่วยเหลือในการจัดยา โดยออกแบบเป็นหุ่นยนต์ที่มีโครงสร้างแบบแบบคาร์ทีเซียนที่ประกอบขึ้นจากแกนสำเร็จรุ่น IAI ROBO Cylinder RCP2CR-SA5 ที่มีความยาว 350 มิลลิเมตร และ IAI ROBO Cylinder RCP2CR -SA6C ที่มีความยาว 150 มิลลิเมตรและมี Raspberry Pi เป็นอุปกรณ์รับส่งข้อมูลจากผู้ใช้งานผ่านเครือข่ายไร้สายและจัดการการทำงานของตัวหุ่นยนต์ให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งของช่องเก็บยาด้วยระยะทางที่สั้นที่สุดและใช้เวลาน้อยที่สุด

มหาวิทยาลัยเทคโนโลยีสุรนารี

สาขาวิชา วิศวกรรมเมคคาทรอนิกส์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2563 ลายมือชื่ออาจารย์ที่ปรึกษา_____

SITTINON JULNITI : DESIGN AND PROTOTYPING FOR DRUG SUPPORT AND MANAGEMENT IN DISPENSING ROOMS. THESIS ADVISOR : ASSOC. PROF. FLT. LT. KONTORN CHAMNIPRASART, Ph.D.,65 PP

CARTESIAN ROBOT/SOCKET PROTOCOL/TRAVELING SALESMAN PROBLEM

The objective of this research is to design and build a prototype for aiding and managing medicines in the drug storage room. From the problem of drug delivery to patients who have been hospitalized for a long time due to the large amount of medicines available in the hospital. Dosage per patient or the amount of patients with a lot of time This is the reason why pills are taking so long and there is a chance of error. In this research, the researcher designed and built a prototype that assists in drug administration. It is designed as a robot with a Cartesian structure that is made up of the 350 mm long IAI ROBO Cylinder RCP2CR-SA5 core and the 150 mm long IAI ROBO Cylinder RCP2CR -SA6C with a Raspberry Pi. It is a device that transmits data from the user via a wireless network and manages the robot body to move the robot to the position of the drug compartment with the shortest distance and least time

School of Mechatronics Engineering Student's Signature Sittinon.

Academic Year 2020 Advisor's Signature Kont Chyt

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล หลักสูตรวิศวกรรมเมคคาทรอนิกส์ ผู้จัดทำได้รับการอนุเคราะห์จากบุคคลหลายฝ่ายที่ให้คำปรึกษา การแนะนำแนวคิดการทำวิจัยและการช่วยเหลืออย่างดี ดังนี้

รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ รองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ให้โอกาสทางการศึกษาระดับบัณฑิตศึกษา คอยให้คำแนะนำ คำปรึกษาและแนวคิดต่างๆ ในการแก้ไขปัญหาตลอดการทำงานวิจัยรวมทั้งการตรวจทานและแก้ไขเล่มวิทยานิพนธ์นี้จนเสร็จสมบูรณ์

อาจารย์ อภิลักษณ์ หล่อนกลาง อาจารย์ จีรวรรณ หอมจันทร์ และอาจารย์ ปณิธิ เพ็ชรนอก ที่ได้สละเวลาให้ความรู้และให้คำปรึกษา แนะนำแนวทางการทำวิจัย รวมทั้งตรวจสอบแก้ไขให้งานวิจัยมีความถูกต้องและสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณกลุ่มวิจัย FLYE LAB ที่ให้การสนับสนุนด้านการจัดหาอุปกรณ์และสถานที่สำหรับการทำงานวิจัยตลอดจนงานวิจัยนี้เสร็จสมบูรณ์

ขอขอบคุณสถาบันวิจัยแสงซินโครตรอน (องค์การมหาชน) ที่มอบทุนการศึกษาระดับบัณฑิตศึกษา และบุคลากรทุกท่านที่ให้ความรู้และคำแนะนำในการทำวิจัย และบุคคลอื่นๆ ที่ไม่ได้กล่าวนามทุกท่านที่ให้คำแนะนำ และช่วยเหลือในเรื่องต่างๆ ที่เป็นประโยชน์และสามารถทำให้งานวิจัยนี้สำเร็จลุล่วงไปด้วยดี ทำให้ได้รับความรู้ ประสบการณ์และทักษะต่างๆ ในการทำงานวิจัย

ขอกราบขอบพระคุณบิดา มารดา ที่ให้การอบรมเลี้ยงดูสั่งสอน สนับสนุนทางการศึกษา รวมทั้งให้กำลังใจเป็นอย่างดีมาตลอด จนทำให้ผู้วิจัยประสบความสำเร็จเสมอมา

สิทธินนท์ จุลนิธิ

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช

บทที่

1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของงานวิจัย 1
1.3 ขอบเขตของงานวิจัย 1
1.4 วิธีดำเนินงานวิจัย 2
1.5 สถานที่ทำงานวิจัย 2
1.6 ประโยชน์ที่คาดว่าจะได้รับ 2
2 ปริทัศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง 3
2.1 หุ่นยนต์ 3
2.1.1 โครงสร้างของหุ่นยนต์ 5
2.2 socket communication 12
2.2.1 ชนิดของ socket 12
2.2.2 IP Address 12
2.2.3 Port and Services 13
2.3 ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem, TSP) 14
2.4 ปริทัศน์วรรณกรรมที่เกี่ยวข้อง 16
3 วิธีดำเนินงานวิจัย 19
3.1 กล่าวนำ 19

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ (ต่อ)

หน้า

3.2 การออกแบบและสร้างเครื่องต้นแบบ 19
3.3 โปรแกรมสำหรับรับและส่งข้อมูล 21
3.4 โปรแกรมการคำนวณระยะการเคลื่อนที่ของเครื่องต้นแบบการ 22
3.5 อุปกรณ์สำหรับหยิบตัวอย่างยา 25
4 ผลการทดลองและอภิปรายผล 27
4.1 การทดสอบระบบรับส่งข้อมูล 27
4.2 การทดสอบระบบคำนวณเส้นทางที่สั้นที่สุด 30
4.3 การทดสอบเวลาการทำของเครื่องต้นแบบและความถูกต้องในการเคลื่อนที่ 31
5 สรุปผลการวิจัยและข้อเสนอแนะ 33
5.1 สรุปผลการวิจัย 33
5.2 ข้อเสนอแนะ 34
รายการอ้างอิง 35
ภาคผนวก
ภาคผนวก ก. โปรแกรมที่ได้เขียนขึ้นในงานวิจัย 37
ภาคผนวก ข. บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา 59
ประวัติผู้เขียน 65

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง

ตารางที่ **หน้า**

3.1 รายละเอียดของแกนสำเร็จรุ่น IAI ROBO Cylinder RCP2CR-SA5 20

3.2 รายละเอียดของแกนสำเร็จรุ่น IAI ROBO Cylinder RCP2CR-SA6C 20

3.3 รายละเอียดของกล่องควบคุมแกนสำเร็จ IAI ROBO Cylinder Controller RCP2-C 21

4.1 ตารางแสดงข้อมูลที่ผู้เลือกกับข้อมูลที่เครื่องต้นแบบได้รับ.....28

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป

รูปที่ **หน้า**

2.1 หุ่นยนต์อุตสาหกรรมยูนิเมตจากบริษัทยูนิเมชั่น 3
2.2 หุ่นยนต์ที่ถูกควบคุมด้วยปัญญาประดิษฐ์ 4
2.3 ไบนารีลิงก์(binary link)ที่1และ2 5
2.4 การเชื่อมต่อกันของลิงก์ด้วยข้อต่อ(joint) 5
2.5 ประเภทของข้อต่อและสัญลักษณ์ 6
2.6 ลักษณะของหกองศาอิสระกับกรอบพิกัด 7
2.7 ลักษณะของโครงสร้างแบบคาร์ทีเซียน 8
2.8 ลักษณะของโครงสร้างแบบทรงกระบอก 9
2.9 ลักษณะของโครงสร้างแบบโพล่า 9
2.10 ลักษณะของโครงสร้างแบบอาทิกูเรต 10
2.11 ลักษณะของโครงสร้างแบบสกาล่า 10
2.12 โครงสร้างข้อมือหุ่นยนต์ 11
2.13 เอนด์เอฟเฟกต์ลักษณะต่างๆ 12
2.14 ลักษณะของ IPv4 13
2.15 ลักษณะของ IPv6 13
2.16 Port and Servicesที่ถูกเรียกใช้งานเป็นมาตราฐาน 14
2.17 วิธีการหาคำตอบของปัญหาการเดินทางของพนักงานขาย 16
3.1 การออกแบบเครื่องต้นแบบในโปรแกรมโซลิดเวิร์ค 20
3.2 การส่งข้อมูลหากันระหว่างผู้ใช้งานกับ Raspberry Pi 21
3.3 ลักษณะข้อมูลที่ Raspberry Pi ได้รับ 22
3.4 หน้าต่างโปรแกรมของผู้ใช้งานในการเลือกยา 22
3.5 การเลือกเส้นทางเดินแบบ Nearest-neighbor Heuristic 23
3.6 ตำแหน่งยาในช่องเก็บและพิกัด 23
3.7 คำตอบที่ได้เมื่อผ่านโปรแกรมคำนวนระยะทาง 24
3.8 โครงสร้างของคำสั่งที่ใช้ในการเคลื่อนที่ไปยังตำแหน่งใดๆ 24

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.9 แผนผังการทำงานของโปรแกรมคำนวนระยะทาง....255
3.10 อุปกรณ์สำหรับหยิบตัวอย่างยา....26
4.1 ข้อมูลที่ถูกเลือกโดยผู้ใช้งาน....27
4.2 การรอรับข้อมูลจากผู้ใช้งานของเครื่องต้นแบบ....28
4.3 ตัวอย่างข้อมูลที่ถูกสุ่มขึ้นมาสำหรับการทดสอบ....29
4.4 การทดสอบการส่งข้อมูลที่ได้สุ่มมา....29
4.5 ระยะทางเฉลี่ยเมื่อผ่านโปรแกรมกับค่าเริ่มต้น....30
4.6 เวลาเฉลี่ยเมื่อผ่านโปรแกรมกับค่าเริ่มต้น....31

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

สถิติการเข้ารับการรักษาตัวของผู้ป่วยภายนอกในโรงพยาบาลเฉลี่ยอยู่ที่ 8,000 – 10,000 คนต่อวันตามแต่ละพื้นที่และจำนวนผู้ป่วยภายนอกจะมีประมาณมากที่สุดอยู่ในช่วยเช้าของแต่ละวัน จึงทำให้งานบริการหลายๆอย่างภายในโรงพยาบาลเกิดความล้าช้าและใช้เวลาไปกับการรอคิวเป็นเป็นเวลาเฉลี่ยอยู่ที่ 35 – 360 นาที ตามแต่ละบริการภายในบริการ การรอรับยาเป็นหนึ่งในงานบริการที่เกิดความล้าช้าและความผิดพลาด

ปัญหาของความล้าช้าที่เกิดขึ้นภายในห้องจ่ายยาเกิดขึ้นได้หลากหลายสาเหตุไม่ว่าจะเป็นประมาณของใบสั่งยาที่เข้ามาในช่วงเวลาหนึ่ง จำนวนบุคลากรที่ใช้ในการจัดยา ขนาดของห้องเก็บยา ลำดับขั้นตอนของการจัดยา ปริมาณยารักษาโรค ระยะเวลาที่ผู้ป่วยภายนอกจะต้องรอรับยาจึงอยู่ที่ประมาณ 35 – 60 นาทีเป็นอย่างน้อย และปัญหาความผิดพลาดที่อาจเกิดขึ้นภายในห้องจ่ายยามาจากชื่อที่มีความคล้ายคลึงกันและคุณสมบัติของการรักษาโรคจึงทำให้เกิดความไม่ปลอดภัยต่อผู้ป่วย

งานวิจัยชิ้นนี้จึงได้นำเสนอการออกแบบและสร้างเครื่องต้นแบบสำหรับการช่วยเหลือและจัดการยาเพื่อช่วยลดระยะเวลาและเพิ่มความถูกต้องในการจัดยา โดยการสร้างระบบสั่งจ่ายยาผ่านระบบอินเตอร์เน็ตและนำคอมพิวเตอร์ขนาดเล็ก (Single-board computer) อย่าง Raspberry Pi มารับคำสั่งควบคุมการทำงานงานของเครื่องต้นแบบ

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อออกแบบและสร้างเครื่องต้นสำหรับการช่วยเหลือและจัดการยา

1.2.2 เพื่อลดระยะเวลาและความผิดพลาดในการค้นหายา

1.2.3 เพื่อออกแบบระบบที่สามารถทำงานร่วมกันกับระบบอินเตอร์เน็ต

1.3 ขอบเขตของงานวิจัย

1.3.1 สร้างโปรแกรมสำหรับเครื่องต้นแบบจากภาษา Python

1.3.2 ออกแบบโครงสร้างของเครื่องต้นแบบ โดยใช้โปรแกรม SolidWorks

1.3.3 ออกแบบระบบควบคุมการทำงานของเครื่องต้นแบบจากบอร์ดคอมพิวเตอร์ขนาดเล็ก Raspberry Pi

1.4 วิธีดำเนินงานวิจัย

1.4.1 ศึกษาค้นคว้าองค์ความรู้ ทฤษฎี และเอกสารที่เกี่ยวข้อง

1.4.2 วิเคราะห์การออกแบบกลไกของเครื่องต้นแบบ

1.4.3 ออกแบบโครงสร้างเครื่องต้นแบบ โดยใช้โปรแกรม SolidWorks

1.4.4 จัดหาวัสดุและอุปกรณ์ในการจัดสร้าง

1.4.5 นำเนินการสร้างเครื่องต้นแบบ

1.4.6 ดำเนินการเขียนโปรแกรมสำหรับเครื่องต้นแบบและควบคุมเครื่องต้นแบบ

1.4.7 ทดสอบระบบของเครื่องต้นแบบทุกระบบ และทำการปรับปรุงแก้ไข

1.4.8 ทำการวิเคราะห์ข้อมูลและสรุปผลที่ได้

1.4.9 เขียนรายงานฉบับสมบูรณ์

1.5 สถานที่ทำงานวิจัย

1.5.1 อาคารเครื่องมือวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีสุรนารี

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้รับเครื่องต้นแบบสำหรับช่วยเหลืองานในห้องยา

1.6.2 ลดระยะเวลาและความผิดพลาดจากการจัดยา

1.6.3 ได้ระบบที่สามารถทำงานร่วมกันกับระบบอินเตอร์เน็ต

บทที่ 2
ปริทัศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

2.1 หุ่นยนต์

หุ่นยนต์เป็นเครื่องจักรที่ทำงานซับซ้อนได้โดยอัตโนมัติ คำว่าหุ่นยนต์ถูกใช้ครั้งแรกในอาร์ยูอาร์ ละครแนวดิสโทเปียของเช็กในปี 1920 เป็นละครเกี่ยวกับหุ่นยนต์ที่เข้ามาทำงานแทนที่มนุษย์ในสายพานการผลิตหุ่นยนต์ได้ปรากฏตัวในโลกความเป็นจริง ในปี 1939 หุ่นยนต์ชื่อว่า “อิเล็กโทร”ถูกสร้างขึ้นจากฝันเฟื่องและมอเตอร์ หุ้มด้วยอลูมิเนียม ถูกแสดงในงานแสดงสินค้านานาชาติที่นิวยอร์ก ในปี 1960 นักเขียนนิยายวิทยาศาสตร์ชื่อ ไอแซค อสิมอฟ ได้สร้างกฎควบคุมพฤติกรรมหุ่นยนต์ที่มีอิทธิพลต่อแนวคิดเรื่องหุ่นยนต์ ไปอีกหลายทศวรรษ ได้แก่ กฎข้อที่หนึ่ง หุ่นยนต์ต้องไม่ทำร้ายมนุษย์ กฎข้อที่สองหุ่นยนต์ต้องเชื่อฟังคำสั่งของบุคคลที่เหมาะสมเว้นแต่สิ่งนั้นจะขัดกับกฎข้อที่หนึ่ง กฎข้อสามหุ่นยนต์ต้องป้องกันตนเองได้เว้นแต่การทำเช่นนั้นจะขัดกับกฎข้อที่หนึ่งหรือสอง

แม้กฎทั้งสามพอจะเป็นเพียงสิ่งที่แต่งขึ้นแต่ก็เป็นแรงบันดาลใจให้วิศวกรสร้างหุ่นยนต์ที่สามารถใช้ได้จริงขึ้นมา ในปี 1961 เป็นหุ่นยนต์สำหรับอุตสาหกรรมชื่อ “ยูนิเมต” เครื่องจักรกลสามารถยืดแขนได้สองเมตร และสามารถทำงานได้หลากหลายทั้งในโรงงานและห้องแลป

รูปที่ 2.1 หุ่นยนต์อุตสาหกรรมยูนิเมตจากบริษัทยูนิเมชั่น (ที่มา:https://spectrum.ieee.org)

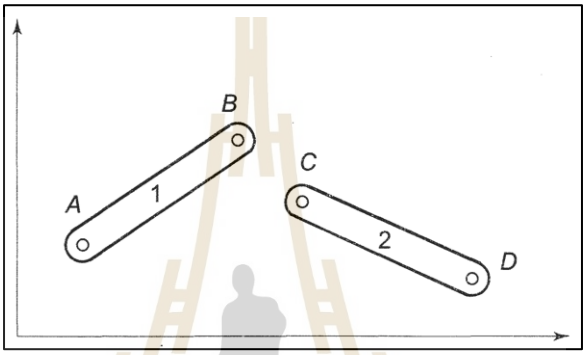
โดยมีทักษะทักเทียมกับมนุษย์แต่ไม่จำเป็นต้องพักผ่อน ยูนิเมตจึงมีอิทธิพลต่ออุตสาหกรรมการผลิตยานยนต์เป็นอย่างมาก ในช่วงปี 1970 ถึง 1981 ประเทศญี่ปุ่นเปิดรับเทคโนโลยีด้านหุ่นยนต์อย่างจริงจังจนมีหุ่นยนต์ที่ทำงานด้านการผลิตยานยนต์มากกว่า 6,000 ตัวทั่วประเทศญี่ปุ่น ในขณะเดียวกันช่วงปี 1960 ถึง 1972 ได้มีการพัฒนาซอฟต์แวร์ที่สามารถเขียนตัวเองได้ เรียกว่า เอไอหรือปัญญาประดิษฐ์ขึ้นมาทดลองในหุ่นยนต์ที่สามารถเคลื่อนที่ได้ที่มีชื่อว่า “เชคีย์” ในการทดสอบเป็นการให้หุ่นยนต์เคลื่อนที่ไปยังจุดหมายผ่านสิ่งกีดขวางหรือสถานการณ์ที่คาดไม่ถึงแต่ในช่วงของการพัฒนาก็ไม่ได้ราบรื่นเนื่องจากความเร็วในการประมวลผลยังค่อนข้างต่ำ จนในช่วงปี 1990 คอมพิวเตอร์มีประสิทธิภาพสูงขึ้นอย่างมากทำให้เอไอมีความฉลาดขึ้นด้วยความสำเร็จสูงสุดในช่วงนั่นคือการที่บริษัทไอบีเอ็มพัฒนาซูเปอร์คอมพิวเตอร์ “ดีปบลู” ขึ้นมาและสอนให้เอไอเล่นหมากรุกจนสามารถชนะแกรนด์มาสเตอร์หมากรุกได้ในปี 1997 ในปี 2000 หุ่นยนต์รูปแบบมนุษย์ชื่อ “อาซิโม” ได้ถูกเปิดตัวขึ้นเป็นครั้งแรก อาซิโมถูกพัฒนาขึ้นโดยบริษัทฮอนด้า ในช่วงนั้นจุดประสงค์ของการพัฒนาหุ่นยนต์รูปแบบมนุษย์เพื่อช่วยเหลือมนุษย์ที่ต้องอยู่อย่างโดดเดี่ยวในชีวิตยุคใหม่ ปี 2013 หุ่นยนต์รูปแบบมนุษย์ที่ถูกพัฒนาโดยบริษัทโตโยต้าชื่อ “คิโรโบ” ได้ถูกส่งขึ้นไปในอวกาศเพื่อเป็นเพื่อนกับนักบินอวกาศในสถานีอวกาศนานาชาติด้วยความช่วยเหลือจากเอไอทำให้หุ่นยนต์สามารถทำงานที่มีความซับซ้อนขึ้นได้มากกว่าการหยิบสิ่งของซ้ำไปซ้ำมาอย่างมากหุ่นยนต์จึงเป็นสิ่งที่อยู่ร่วมกับมนุษย์ไม่ว่าจะเป็นการดูแลเด็กรักษาผู้ป่วยหรือเป็นเพื่อนกับมนุษย์

รูปที่ 2.2 หุ่นยนต์ที่ถูกควบคุมด้วยปัญญาประดิษฐ์ (ที่มา:https://history-computer.com/)

2.1.1 โครงสร้างของหุ่นยนต์

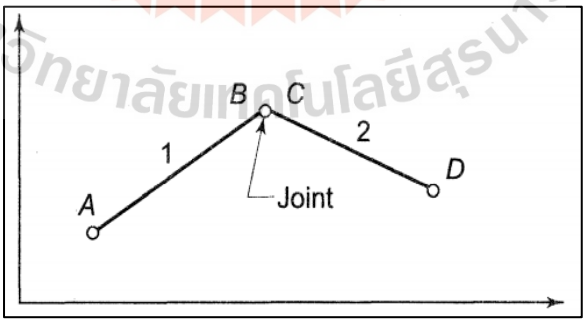
1) ลิงค์

โครงสร้างทางกลของหุ่นยนต์เป็นกลไกที่มีองค์ประกอบเป็นลิงค์หรือบาร์ ลิงค์สามารถเชื่อมต่อกันได้มากกว่าหนึ่งจุด โดยลิงค์ที่จุดเชื่อมต่อกันสองจุดจะเรียกว่า ไบนารีลิงก์ (binary link) รูปที่ 2.3 แสดงไบนารีลิงก์ (binary link) 1 และ 2 แต่ละลิงก์จะมีรูอยู่ที่ปลาย A, Bและ C, D ตามลำดับเพื่อเชื่อมต่อกัน หรือเชื่อมต่อไปยังลิงค์อื่นๆ



รูปที่ 2.3 ไบนารีลิงก์ (binary link) ที่ 1 และ 2 (mittal r. k. และ nagrath i. j.,2003)

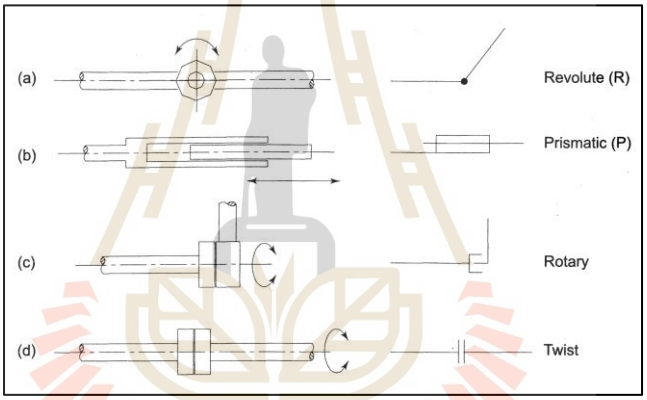
รูปที่ 2.4 แสดงการเชื่อมต่อกันของลิงก์ด้วยข้อต่อ (joint) เป็นการใส่พินผ่านรู B และ C ของลิงก์ที่ 1 กับลิงก์ที่ 2 ข้อต่อที่เกิดขึ้นจะถูกเรียกว่า ข้อต่อแบบหมุน (rotary joint)



รูปที่ 2.4 การเชื่อมต่อกันของลิงก์ด้วยข้อต่อ (joint) (mittal r. k. และ nagrath i. j.,2003)

2) ลักษณะโครงสร้างของข้อต่อ

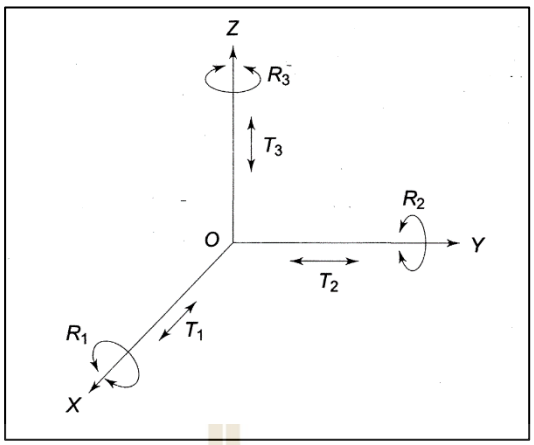
ประเภทของข้อต่อนั้นมีหลายประเภทแต่ประเภทที่นิยมนำมาใช้กับหุ่นยนต์อุตสาหกรรมมีเพียงสองประเภทคือ แบบหมุน (Revolute-R) และ แบบเลื่อน (Prismatic-P) การเคลื่อนที่สัมพัทธ์ของข้อต่อที่อยู่ติดกันขึ้นอยู่กับชนิดของข้อต่อนั้นๆ ตามรูป 2.5 ข้อต่อแบบหมุนในรูป 2.5 (a) ลิงก์ทั้งสองจะถูกเชื่อมด้วยพินหรือเดือยกับแกนของลิงก์ทำให้สามารถหมุนไปด้วยกันได้ ข้อต่อแบบเลื่อน (Prismatic joint)ในรูป 2.5 (b) ลิงก์สองลิงก์ที่อยู่ติดกันลักษณะการเคลื่อนที่เป็นแบบเชิงเส้น ตัวอย่างการเคลื่อนที่เช่น สกรูกับน็อต หรือเหล็กสับเฟืองกับเฟือง เป็นต้น ข้อต่อแบบอื่นๆที่สามารถพบได้แก่ planar เป็นการเคลื่อนที่ของพื้นผิวหนึ่งกับอีกพื้นผิวหนึ่ง cylindrical จะมีหนึ่งลิงก์ที่หมุนรอบลิงก์อื่นได้ 90 องศา ดั้งรูป 2.5 (c) และ spherical ลิงก์สามารถเคลื่อนที่ไปกันกับลิงก์อื่นๆในสามมิติ



รูปที่ 2.5 ประเภทของข้อต่อและสัญลักษณ์ (mittal r. k. และ nagrath i. j.,2003)

3) องศาอิสระ (Degrees of Freedom : DOF)

จำนวนการเคลื่อนไหวอิสระที่วัตถุสามารถทำได้ในพื้นที่ 3 มิติเรียกว่าจำนวนองศาอิสระ (DOF) ดังนั้นวัตถุในอากาศจึงมี 6 องศาอิสระ สามองศาสำหรับการบอกตำแหน่ง (position)และสามองศาสำหรับการบอกการวางแนว (orientation) การเคลื่อนไหวอิสระทั้งหก ในรูปที่ 2.6 ได้แก่ :



รูปที่ 2.6 ลักษณะของหกองศาอิสระกับกรอบพิกัด (mittal r. k. และ nagrath i. j.,2003)

1. สามการเลื่อนตามแนวแกน (translations) (T1, T2, T3) แสดงถึงการเคลื่อนที่เชิงเส้นพร้อมทั้งสามแกนตั้งฉากที่ระบุตำแหน่งของวัตถุในพื้นที่

2. สามการหมุนรอบแกน (rotations) (R1, R2, R3) แสดงถึงการเคลื่อนที่เชิงมุมรอบแกนทั้งสามที่ระบุการวางแนวของวัตถุในพื้นที่

จากเนื้อหาข้างต้น 6 องศาอิสระที่เกิดขึ้นในระนาบ3มิติเมื่อนำมาระบุตำแหน่ง (ตำแหน่งและการวางแนว) จะได้เป็น $2 \times 3 = 6$ อย่างไรก็ตามในระนาบ 2 มิติ วัตถุจะมี 3 องศาอิสระโดยมี 2 การเลื่อนและ 1 การหมุน ตัวอย่างลิงก์ที่ 1 และลิงก์ 2 ในรูปที่ 2.3

4) โครงสร้างของหุ่นยนต์

กลไกของหุ่นยนต์ที่มี 3 องศาอิสระจะขึ้นอยู่กับชนิดของข้อต่อทั้งสามและการจัดเรียงของข้อต่อเพื่อให้ได้ประสิทธิภาพในการเข้าถึงตำแหน่งที่เหมาะสมในพื้นที่สามมิติ การออกแบบหุ่นยนต์จึงต้องมีความสามารถดังนี้ ลิงก์จะต้องมีความยาวมากพอที่จะสามารถเข้าถึงพื้นที่ได้มากที่สุดและการออกแบบต้องมีความแข็งแรงมากพอที่จะรับน้ำหนักของชิ้นงานและน้ำหนักของอุปกรณ์ที่ถูกติดตั้งลงบนหุ่นยนต์

โครงสร้างที่นิยมนำมาสร้างหุ่นยนต์จะด้วยกับทั้งหมด 4 แบบ โดยแต่ละแบบมีการตั้งชื่อมาจากลักษณะการจัดเรียงข้อต่อหรือการรูปทรงของการเคลื่อนที่มีด้วยกันดังต่อไปนี้

1) แบบคาร์ทีเซียน (Cartesian) ข้อต่อแบบเลื่อนทั้งสามข้อ

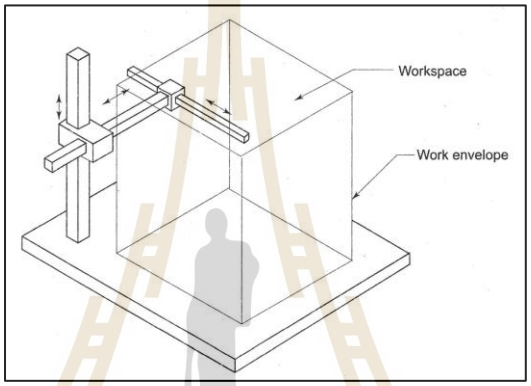
2) แบบทรงกระบอก (Cylindrical) ข้อต่อหมุนหนึ่งข้อและข้อต่อเลื่อนสองข้อ

3) แบบโพล่า (Polar) ข้อต่อแบบหมุนสองข้อและข้อต่อแบบเลื่อนหนึ่งข้อ

4) แบบอาทิกูเรต (Articulated)ข้อต่อทั้งสามเป็นแบบหมุน

รายละเอียดของแต่ละโครงสร้างเป็นไปตามต่อไปนี้

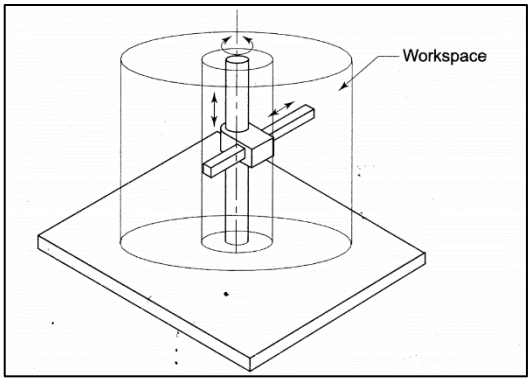
1) แบบคาร์ทีเซียน (Cartesian) เป็นโครงสร้างที่ง่ายที่สุดด้วยทุกข้อต่อเป็นแบบเลื่อน (PPP) ในรูป 2.7 มันถูกสร้างให้มีแกนตั้งฉากซึ่งกันและกัน มีขอบเขตของการเคลื่อนที่ของแต่ละลิงก์ดั้งนั้นจุดที่ปลายของหุ่นยนต์จึงมีพื้นที่ทำงานเป็นรูปทรงลูกบาศก์ พื้นที่ทำงาน (Workspace) จะแสดงถึงส่วนรอบๆฐานของหุ่นยนต์ รูปร่างและขนาดของพื้นที่ทำงานจะขึ้นอยู่กับโครงสร้างของหุ่นยนต์, องศาอิสระ, ขนาดของลิงก์ และ รูปแบบของข้อต่อ แต่เมื่อติดตั้งอุปกรณ์ที่อยู่ปลายของหุ่นยนต์ (end-effector) เพื่อเค้าอาจจะทำให้ พื้นที่ทำงาน เพิ่มขึ้นหรือลดลงก็ได้



รูปที่ 2.7 ลักษณะของโครงสร้างแบบคาร์ทีเซียน (mittal r. k. และ nagrath i. j.,2003)

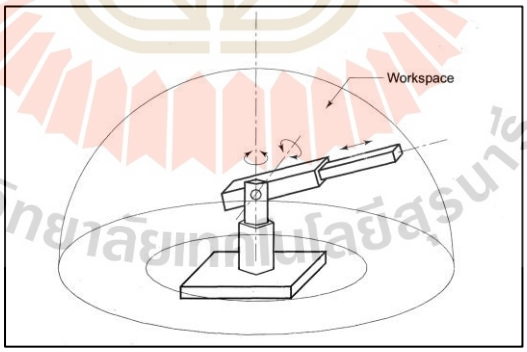
ข้อดีของโครงสร้างแบบคาร์ทีเซียน จะให้ความแม่นยำที่สูง และ ควบคุมการทำงานได้ง่าย แต่ก็ยังมีข้อเสียอยู่ตรง พื้นที่ติดตั้งจะต้องมีที่ขนาดใหญ่หากต้องการแบกน้ำหนักที่มาก และความคล่องตัวของหุ่นยนต์

2) แบบทรงกระบอก (Cylindrical) โครงสร้างจะเป็นไปตามรูป 2.8 จะมีการใช้สองข้อต่อที่เป็นแบบเลื่อน และ หนึ่งข้อต่อที่เป็นแบบหมุน จะต่างจากโครงสร้างแบบคาร์ทีเซียนตรงที่หนึ่งแกนแบบเลื่อนจะถูกเปลี่ยนเป็นแบบหมุน และจะมีการนับข้อต่อแบบหมุนเป็นแกนที่หนึ่ง (RPP) การที่มีหนึ่งแกนเป็นข้อต่อแบบหมุนจะทำให้หุ่นยนต์หมุนรอบตัวเองได้ ขึ้นลงด้วยแกนที่เป็นข้อต่อแบบเลื่อนและเข้าถึงพื้นที่ทำงานผ่านแกนที่ข้อต่อแบบเลื่อน ดังนั้นมีจะทำให้พื้นที่ทำงานของโครงสร้างแบบนี้เป็นรูปทรงกระบอกตามชื่อ โครงสร้างแบบนี้จะมีข้อดีในด้านของความแข็งแรงเชิงกลแต่เมื่อแขนของหุ่นยนต์มีความยาวเพิ่มมากขึ้นจะทำให้ความแม่นยำลดลง งานที่เหมาะกับโครงสร้างแบบนี้จึงเป็นงานเกี่ยวกับการโหลดชิ้นงานเข้าเครื่องจักร



รูปที่ 2.8 ลักษณะของโครงสร้างแบบทรงกระบอก (mittal r. k. และ nagrath i. j.,2003)

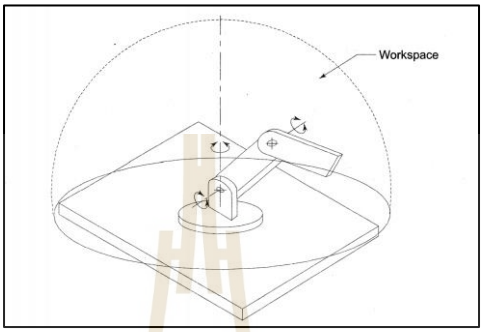
3) แบบโพล่า (Polar) โครงสร้างจะเป็นไปตามรูป 2.9 จะมีสองลิงก์ที่เป็นข้อต่อแบบหมุนถูกติดตั้งอยู่กับฐานของหุ่นยนต์และมีอีกหนึ่งลิงก์ที่ข้อต่อแบบเลื่อนเพื่อปรับความยาวของหุ่นยนต์ (RRP) พื้นที่ทำงานของโครงสร้างแบบนี้จะได้รูปทรงที่เป็นทรงกลมตามการจัดวางของข้อต่อหุ่นยนต์ เนื่องจากการมีข้อต่อแบบหมุนอยู่สองตำแหน่งทำให้ความแข็งแรงของโครงสร้างนี้มีความแข็งแรงน้อยและที่ซับซ้อนกว่าแบบคาร์ทีเซียนและทรงกระบอก งานที่นิยมนำไปใช้จึงเป็นงานประเภท งานประกอบหรืองานพ่นสี



รูปที่ 2.9 ลักษณะของโครงสร้างแบบโพล่า (mittal r. k. และ nagrath i. j.,2003)

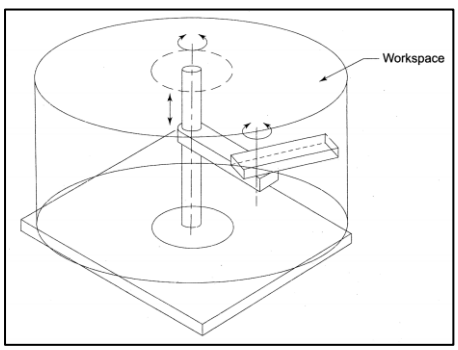
4) แบบอาทิกูเรต (Articulated) เป็นหนึ่งในโครงสร้างที่จำลองแขนของมนุษย์ได้ดีที่สุดและโครงสร้างประเภทนี้มักเรียกกันว่า หุ่นยนต์มนุษย์ มีองค์ประกอบที่คล้ายกับส่วนของมนุษย์ตรง “ปลายแขน” และ “ต้นแขน” และมีข้อต่อแบบหมุนที่ทำหน้าที่ตรงกับ โครงสร้างของมนุษย์บริเวณ “ข้อศอก” และ “หัวไหล่” มีลิงก์ที่อยู่ติดกับฐานคล้ายเอวของมนุษย์ในรูปที่ 2.10

แสดงให้ถึงการจัดเรียงของข้อต่อ ที่มีการใช้ขอตอบแบบหมุนทั้งสามข้อต่อ (RRR) ปริมาณพื้นที่ทำงานของโครงสร้างแบบนี้จะเป็นรูปทรงกลม ที่ขนาดขึ้นอยู่กับความยาวของลิงก์และการออกแบบโครงสร้างแบบนี้นิยมนำไปใช้ในงานอุตสาหกรรม เนื่องจากมีความคล่องแคล่วมากที่สุดและมีความแม่นยำสูงแต่ก็แต่ก็แลกมากับการควบคุมที่ซับซ้อนขึ้น



รูปที่ 2.10 ลักษณะของโครงสร้างแบบอาทิกูเรต (mittal r. k. และ nagrath i. j.,2003)

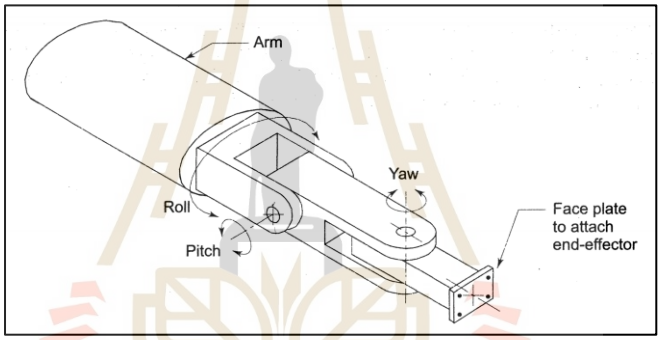
5) แบบสกาล่า (SCARA) เป็นโครงสร้างของหุ่นรูปแบบใหม่ ที่นำเอาข้อดีของแต่ละโครงสร้างมารวมกัน โดยได้มีการเอาโครงสร้างแบบอาทิกูเรต และ แบบทรงกระบอก มารวมกัน เกิดเป็นโครงสร้างใหม่ที่เรียกว่าสกาล่า (SCARA) หรือ Selective Compliance Assembly Robot Arm ในรูปที่ 2.11 โครงสร้างของสกาล่ามีแกนหลักในแนวนอนที่เป็นข้อต่อแบบหมุนช่วยลดแรงเครียด (stress) ที่เกิดจากแรงโน้มถ่วง และแรงเหวี่ยง ได้มากกว่าโครงสร้างที่มีแกนหลักเป็นแนวนอน ข้อดีของโครงสร้างแบบนี้ จะทำให้มีความเร็ว และความแม่นยำที่สูงมาก จึงเหมาะนำไปใช้ในงานประกอบชิ้นส่วน



รูปที่ 2.11 ลักษณะของโครงสร้างแบบสกาล่า (mittal r. k. และ nagrath i. j.,2003)

5) โครงสร้างข้อมือ (Wrist)

ในหัวข้อที่ผ่านมาเป็นการอธิบายถึงโครงสร้างของหุ่นยนต์ที่เป็นการเคลื่อนที่ในส่วนของตำแหน่ง (position) และในหัวข้อนี้จะเป็นการอธิบายการทำงานของข้อมือหุ่นยนต์ที่ติดอยู่กับส่วนปลาย (endpoint) ของหุ่นยนต์ การเคลื่อนที่ของข้อมือจะเป็นการปรับทิศทางของเอนด์เอฟเฟกต์ (end-effector) ให้ทำงานได้อย่างถูกต้อง ตัวอย่างเช่น กริปเปอร์ (gripper) ที่ต้องวางมุมให้เหมาะสมเพื่อเลือกและจับชิ้นงาน ในลักษณะการเคลื่อนที่ของข้อมือจำเป็นจะต้องมี 3 องศาอิสระเพิ่มขึ้นเพื่อให้สามารถเคลื่อนที่ในสามมิติได้อย่างสมบูรณ์ ข้อต่อที่ใช้จะเป็นข้อต่อแบบหมุนเท่านั้นเพื่อใช้ปรับทิศทางของเอนด์เอฟเฟกต์ (end-effector) ทำให้เกิด ลักษณะการเคลื่อนที่ สามแบบที่เรียกว่า โรล (Roll) การเคลื่อนที่ในระนาบตั้งฉากกับปลายแขน, ฟิทช์ (Pitch) การเคลื่อนที่ในระนาบแนวตั้งผ่านแขน, ยอว์ (Yaw) การเคลื่อนไหวในแนวระนาบแนวนอนผ่านแขน



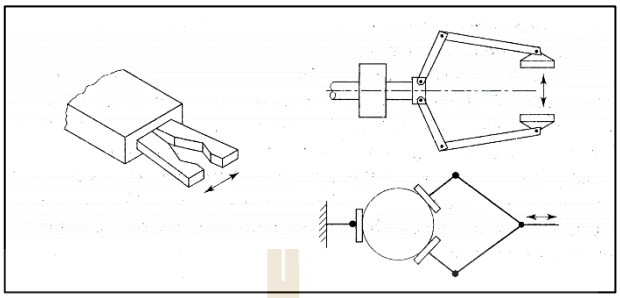
รูปที่ 2.12 โครงสร้างข้อมือหุ่นยนต์ (mittal r. k. และ nagrath i. j.,2003)

6) เอนด์เอฟเฟกต์ (end-effector)

เอนด์เอฟเฟกต์(end-effector) เป็นอุปกรณ์ที่อยู่ภายนอกหุ่นยนต์ และไม่มีการรวมองศาอิสระ (DOF) ของหุ่นยนต์เข้าไปด้วย ในการติดตั้งเอนด์เอฟเฟกต์ (end-effector) ที่ปลายแขนของหุ่นยนต์เพื่อให้เหมาะสมกับงานที่ใช้ โดยสามารถแบ่งออกได้เป็นสองประเภทใหญ่คือ กริปเปอร์ (Grippers), เครื่องมือ (Tools)

กริปเปอร์ (Grippers) เป็นเอนด์เอฟเฟกต์ (end-effector) ชนิดหยิบจับหรือถือชิ้นงานในระหว่างการทำงาน งานที่สามารถนำไปใช้ได้ เป็นขนถ่ายชิ้นงานระหว่างเครื่องจักร จัดเรียงชิ้นงาน หรืองานที่มีลักษณะคล้ายคลึงกัน ลักษณะการทำงานของกริปเปอร์อาจจะเป็นการทำงานเชิงกล แม่เหล็ก หัวดูดสุญญากาศ

ในส่วนของเครื่องมือ (Tools) สำหรับเอนด์เอฟเฟกต์(end-effector) ที่นิยมนำมาใช้กับหุ่นยนต์จะเป็นจำพวก เครื่องมือสำหรับตัด พ่นสี หรือ ประกอบชิ้นงาน



รูปที่ 2.13 เอนด์เอฟเฟกต์ลักษณะต่างๆ (mittal r. k. และ nagrath i. j.,2003)

2.2 socket communication

socket communication คือการแลกเปลี่ยนข้อมูลการภายในอุปกรณ์หรือต่างอุปกรณ์โดยมีการใช้ข้อมูลมาตรฐานแบบยูนิกซ์ (Unix) ในการแลกเปลี่ยนข้อมูลกัน มักพบในชุดคำสั่งเกี่ยวกับ client - server ที่ดำเนินการในฟังชั่นที่ใช้เรียกรับข้อมูลจากลูกข่าย (client) เช่น FTP, SMTP และ POP3 จะทำให้ socket สามารถเริ่มต้น การเชื่อมต่อระหว่างลูกข่าย (client) และแม่ข่าย (server) ในการแลกเปลี่ยนข้อมูลกันได้

2.2.1 ชนิดของ socket

จะมีด้วยการอยู่ทั้งหมด 2ชนิดได้แก่

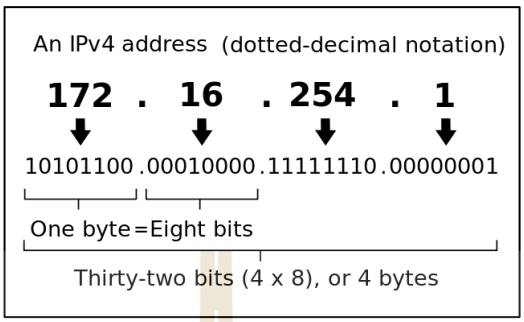
1) Stream Sockets เป็นการส่งข้อมูลในเครือข่ายที่มีการรับประกันความถูกต้องของข้อมูล ถ้าผู้ส่งมีการส่งข้อความ ดังต่อไปนี้ “A”, “B”, “C” ผู้รับก็จะได้รับข้อความตามลำดับเดียวกัน “A”, “B”, “C” Socketsนี้จะใช้ TCP (Transmission Control Protocol) ในการส่งข้อมูลหาผู้รับและถ้ามีการผิดพลาดในการส่งข้อมูลจะมีการส่งข้อผิดพลาดกับห้าผู้ส่งเอง

2) Datagram Sockets เป็นการส่งข้อมูลในเครือข่ายแบบไม่มีการรับประกันความถูกต้องซึ่งมีโอกาสที่ผู้รับจะได้ข้อมูลที่ไม่ถูกต้องแต่ก็แลกมากับความเร็วในการแลกเปลี่ยนข้อมูลที่มากกว่าการส่งข้อมูลด้วย TCP Sockets ชนิดนี้ใช้การส่งแบบUDP (User Datagram Protocol).

2.2.2 IP Address

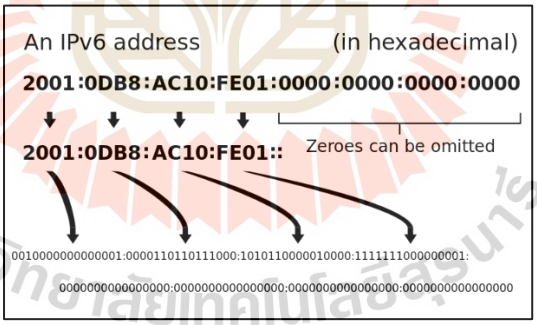
การติดต่อสื่อสารกันบนเครือข่าย IP Address เป็นสิ่งนึงที่เราจำเป็นจะต้องรู้เพราะเปรียบเสมือน บ้านเลขที่ ที่ทุกเครื่องลูกข่าย จำเป็นจะต้องมีในการติดต่อสื่อสารกันในเครือข่าย IP Address อาจจะถูกเรียกในอีกชื่อนึงว่า “แอกเดรส” ในปัจจุบัน IP Address ถูกพัฒนาขึ้นมา ด้วยกัน

ทั้งหมด สองเวอร์ชั่น ในเวอร์ชั่นแรกเรียกว่า IPv4 ที่เป็นชุดตัวเลข สี่ชุด แต่ละชุด จะมีค่า 8 bit ร่วมเป็นทั้งหมด 32 bit ที่สามารถรองรับลูกข่ายได้ประมาณสี่พันล้านหมายเลข



รูปที่ 2.14 ลักษณะของ IPv4 (https://th.wikipedia.org/)

ต่อมาได้มีการพัฒนา เป็นอีกเวอร์ชั่นที่สองที่ เรียกว่า IPv6 เป็นชุดตัวเลขแปดชุด แต่ละชุดจะมีค่า 16 bit ร่วมเป็นทั้งหมด 128 bit สามารถรองรับลูกข่ายได้ประมาณสามร้อยสี่สิบล้านหมายเลข



รูปที่ 2.15 ลักษณะของ IPv6 (https://th.wikipedia.org/)

2.2.3 Port and Services

เป็นการระบุการเข้าถึง host โดยตัวเลขตั้งแต่ 0 ถึง 65535 เพื่อเข้าถึงเซอร์วิส (Services) ที่ต้องการสำหรับ Port ช่วงตั้งแต่ 1 ถึง 1023 จะถูกเรียนกว่า Well known Ports ที่ทาง Internet Assigned Numbers Authority (IANA) เป็นคนกำหนดให้เป็น post สำหรับ Services มาตรฐาน ที่รู้จักกันทั่วไป เช่น TCP Post 23 เป็นการใช้ Service Telnet หรือ UDP Port 69 เป็นการ

ใช้ Service Trivial File transfer Protocol (TFTP) เป็นต้น ส่วน Port ที่อยู่ช่วง 1024 ถึง 65535 จะถูกเรียกว่า Registered Ports เป็นช่วงที่ผู้ใช้งานสามารถกำหนดขึ้นมาเองได้ว่าต้องการให้ทำงานใน Services แบบไหน

Service	Port Number	Service Description
echo	7	UDP/TCP sends back what it receives.
discard	9	UDP/TCP throws away input.
daytime	13	UDP/TCP returns ASCII time.
chargen	19	UDP/TCP returns characters.
ftp	21	TCP file transfer.
telnet	23	TCP remote login.
smtp	25	TCP email.
daytime	37	UDP/TCP returns binary time.
tftp	69	UDP trivial file transfer.
finger	79	TCP info on users.
http	80	TCP World Wide Web.
login	513	TCP remote login.
who	513	UDP different info on users.
Xserver	6000	TCP X windows (N.B. >1023).

รูปที่ 2.16 Port and Servicesที่ถูกเรียกใช้งานเป็นมาตราฐาน (https://www.tutorialspoint.com)

2.3 ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem, TSP)

เป็นการแก้ปัญหาเพื่อใช้ในการตัดสินใจ เช่น ระยะทางที่สั้นหรือเหมาะสมที่สุด โดยมีเมืองหรือ จำนวนจุดที่ต้องเดินทางเป็นจำนวน N เมือง หรือ N จุด เริ่มต้นจากเมืองใดเมืองหนึ่งตามเงื่อนไขที่กำหนด แล้วเลือกทางผ่านทุก N เมืองจนสุดท้ายกลับมาที่เมืองเริ่มต้น ยกตัวอย่างเช่น พนักงานขายต้องการนำเสนอสินค้ากับลูกค้าทั้ง 5 รายที่อยู่ต่างเมืองกัน โดยมีเมืองเริ่มต้นที่เมือง 3 ลักษณะการเดินทางจะเป็น 3 – 4 – 2 – 1 – 5 – 3 ซึ่งจะเห็นว่าจะเริ่มต้นจากเมืองที่กำหนดตามเงื่อนไข แล้วเดินทางต่อไปยังเมือง 4, 2, 1, 5 แล้ววนกลับมาที่เมืองเดิม

การคำนวณหาส้นทางที่เหมาะสมได้จากสมการต้นแบบของการเดินทางของพนักงานขายจะประกอบไปด้วย ดัชนี พารามิเตอร์ และตัวแปรตัดสินใจสามารถแสดงได้ดั้งนี้

ดัชนี:

i, j หมายถึง ลูกค้ารายที่ i หรือ j โดยที่ $i, j = 1...k$

พารามิเตอร์:

C_{ij} หมายถึง ต้นทุนในการเดินทางของลูกค้า i ไปหาลูกค้า j

k หมายถึง จำนวนลูกค้า

S หมายถึง จำนวนลูกค้าที่อยู่ในเส้นทาง

V หมายถึง จำนวนลูกค้าทั้งหมด

ตัวแปรตัดสินใจ:

X_{ij} 1: เมื่อมีการเดินทางจาก i ไป j

0: เมื่อไม่มีการเดินทางจาก i ไป j

แบบจำลองทางคณิตศาสตร์สำหรับกรณีไม่มีข้อจำกัดทางด้านทรัพยากร

สมการเป้าหมาย

$$\sum_{i \neq j} C_{ij} X_{ij} \tag{2.1}$$

สมการขอบข่าย

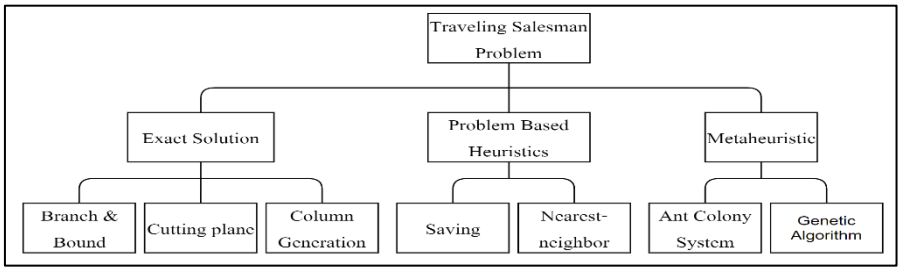
$$\sum_{i=1}^{k} X_{ij} = 1 \qquad i = 1...k \tag{2.2}$$

$$\sum_{j=1}^{k} X_{ij} = 1 \qquad j = 1...k \tag{2.3}$$

$$\sum_{i,j \in S}^{k} X_{ij} \leq |S| - 1 \qquad S \subset V, 2 \leq |S| \leq k - 2 \tag{2.4}$$

สมการที่ (2.1) แสดงต้นทุนการเดินทางจากเมือง i ไปเมือง j สมการที่ (2.2) แสดงการเดินทางออกจากเมือง i ใดๆ ต้องมีค่าเท่ากับ 1 (เมืองใดเมืองหนึ่งเดินทางออกได้เพียงครั้งเดียว) ในขณะที่สมการที่ (2.3) มีการเดินทางเข้าเมือง i ได้เพียงครั้งเดียว เมื่อ (2.4) เป็นสมการป้องกันการเกิดทางเดินย่อย (Subtour) การเกิดทางเดินย่อยหมายความว่า การเดินทางเริ่มต้นจากเมือง ใดเมืองหนึ่งแต่เดินทางไม่ครบทำให้เกิดทางเดินย่อยขึ้น เช่น 1 – 2 – 3 – 4 – 1 และ 5 – 6 – 7 – 8 – 5 ทุกเมืองเดินทางเข้าและออกอย่างละ 1 ครั้งตามสมการ (2.2) และ (2.3) เมื่อมีสมการ (2.4) จะสามารถป้องกัน ปัญหานี้ได้เมื่อในเส้นทางใดๆ ที่มีเมืองน้อยกว่าจำนวน n จะทำให้เกิดเส้นทางการเดินเท่ากับจำนวน เมืองที่มี ทำให้สมการ (2.4) ไม่เป็นจริง

ปัจจุบันปัญหาการเดินทางของพนักงานขาย ได้รับความสนใจแก่ผู้วิจัยเพื่อจำนวนมาก ซึ่งใช้วิธีการแก้ปัญหาด้วยวิธีการที่แตกต่างออกไป ขึ้นกับอยู่เงื่อนไขต่างๆ ในแต่ละปัญหาและมี ข้อดี ข้อเสีย แตกต่างกันไป โดยการแก้ปัญหาการเดินทางของพนักงานขาย แบ่งเป็น 3 กลุ่ม แสดงดังรูปที่ 2.17 อธิบายได้ดังต่อไปนี้



รูปที่ 2.17 วิธีการหาคำตอบของปัญหาการเดินทางของพนักงานขาย
(ชวันลักษณ์ สุวรรณรัศมี,2559)

1) การหาคำตอบที่ดีที่สุด (Exact Solution Algorithm) วิธีนี้จะไม่สามารถ แก้ปัญหาที่ความซับซ้อนด้วยข้อจำกัดของในเรื่องขนาดหรือเวลาในการคำนวณเกิดเป็นคำตอบจึงได้ คำตอบที่มีความกว้างเกินไป ยกตัวอย่างดังต่อไปนี้

1.1) อัลกอริทึมการแตกกิ่งและจำกัดขอบเขต (Branch & Bound Algorithms) เป็นวิธีที่ไม่ยุ่งยาก แต่ไม่สามารถแก้ปัญหาที่ซับซ้อนได้ซึ่งต้องมีการผ่านจุดให้ครบ ทุกจุดโดยไม่สนใจลำดับในการผ่าน โดยอัลกอริทึมจะใช้การลดค่าเมตริกในการคำนวณหาขอบเขต (Bound) เมตริกที่มีค่าขอบเขตที่น้อยที่สุดจะทำการแตกกิ่ง (Branch) ซึ่งวิธีนี้จะสามารถหาเส้นทางที่เหมาะสมที่สุดเหมาะสมกับปัญหาที่มีขนาดเล็กเนื่องจากต้องใช้หน่วยความจำและเวลาในการ คำนวณมาก

1.2) วิธีการแก้ปัญหาด้วยวิธีการตัดระนาบ (Cutting plane) สามารถแก้ปัญหาที่มีขนาดเมือง 49 เมืองและได้คำตอบที่ดีที่สุด

1.3) Column Generation เป็นการคำตอบด้วยวิธี Linear Programming โดยใช้วิธีการซิมเพล็กซ์ในการหาคำตอบ

2) กลุ่มของวิธีการที่สามารถใช้แก้ปัญหาเฉพาะปัญหาเท่านั้น (Problem Based Heuristics) ได้แก่

2.1) Saving Heuristics จะเป็นวิธีที่ทำให้ได้คำตอบที่ดีพอใช้ภายใน ระยะเวลาที่ไม่นานมาก

2.2) Nearest-neighbor Heuristic สามารถเริ่มต้นจาก Node ไหนก็ได้ จากนั้นเชื่อมต่อกับเมืองที่อยู่ใกล้ที่สุด จากนั้นก็เชื่อมจุดนั้นต่อไปยังที่ต่อไปที่อยู่ใกล้ที่สุดเชื่อมต่อแบบนี้ไปเรื่อยๆ จนครบทุกเมือง และสุดท้ายวนมายังจุดเริ่มต้น ซึ่งวิธีการนี้จุดเริ่มต้นจะมีผลต่อคำตอบ

3) เมตาฮิวริสติก (Metaheuristic) เป็นการหาคำตอบโดยการประมาณ (Approximate Algorithm) เป็นวิธีการที่มีความรวดเร็วในการคำนวณ โดยส่วนมากจะให้คำตอบที่ดีกว่ากลุ่ม

Problem Based Heuristic เพราะเป็นวิธีการเพิ่มเติมมาจากกลุ่มที่ 2 แต่วิธีการในกลุ่มนี้จะมีความซับซ้อนกว่าจึงให้คำตอบที่ดีกว่า ตัวอย่างวิธีการในกลุ่มนี้ ได้แก่

3.1) อาณาจักรมด (Ant Colony System) เป็นการได้รับแรง บันดาลใจมาจากพฤติกรรมของมด ผู้ออกแบบวิธีการจึงจำลองพฤติกรรมนี้ โดยในปัญหาการเดินทางของพนักงานขาย (TSP) เริ่มต้นจากการเดินทางจากเมืองหนึ่งไปยังเมืองต่างๆจนครบทุกเมืองจะได้ระยะทางรวมทั้งหมดหากระยะทางเดินทางนี้สั้น จะวางปริมาณฟีโรโมนไว้มาก หากระยะทางที่ได้ยาวจะวางฟีโรโมนไว้น้อย

3.2) วิธีเชิงพันธุกรรม (Genetic Algorithm) เป็นวิธีการหาคำตอบโดยอาศัยรูปแบบกลไกการคัดสรรพันธุกรรมจากธรรมชาติ ซึ่งพันธุกรรมที่ดีเสมือนคำตอบที่ดีจะสามารถอยู่รอดและถูกถ่ายทอดไปสู่รุ่นลูกรุ่นหลานต่อไปได้

2.4 ปริทัศน์วรรณกรรมที่เกี่ยวข้อง

ในการสร้างเครื่องต้นแบบสำหรับการช่วยเหลือและจัดการยาภายในห้องจ่ายยานั้นจะต้องใช้องค์ความรู้ต่างๆ ทั้งในด้านทางทฤษฎีของปัญหาการเดินทางของพนักงานขาย การส่งข้อมูลผ่านเครือข่ายไรสายผ่าน protocol socket เพื่อการนำมาสร้างโปรแกรมที่ใช้ในการทำงานของเครื่องต้นแบบสำหรับการช่วยเหลือและจัดการยาภายในห้องจ่ายยา จึงได้ศึกษางานวิจัยที่เกี่ยวข้องดั้งต่อไปนี้

ชวันลักษณ์ สุวรรณรัศมี (2559) ได้ศึกษาการลดค่าใช้จ่ายเรื่องน้ำมันเชื้อเพลิง ด้วยการลดระยะการเดิทางของพนักขายขายที่จะต้องไปพบลูกค้าในพื้นที่แตกต่างกันทั้งหมด 33 จุด โดยใช้เครื่องมือของโปรแกรมไมโครซอฟท์เอ็กเซล เพื่อสร้างเส้นทางเดินที่สั้นที่สุดจากการสร้างตารางเมตริกระยะทางระหว่างจุด2จุดใดๆ

อภิรักษ์ ขัดวิลาศ (2554) ได้ศึกษาเกี่ยวกับการประยุกต์วิธีเชิงพันธุกรรมซึ่งเป็นหนึ่งในวิธีการหาคำตอบของปัญหาการเดินทางของพนักงานขาย เพื่อหาค่าที่เหมาะสมที่สุด ด้วยการทดลองตามทฤษฎี 5 ขั้นตอนเพื่อหาตำตอบที่ดีที่สุดไว้และกำจัดคำตอบที่ไม่ดีออก

K. A. Radhika และคณะ (2018) ได้ทำการสร้างอุปกรณ์ควบคุมหุ่นยนต์ Pibot จาก Raspberry Pi สองตัวที่อยู่กับ Pibot และผู้ควบคุม ส่งข้อมูลหากันผ่าน TCP/IP Socket communication โดยทางด้านผู้ควบคุมจะส่งข้อมูลจาก joystick ไปยัง Raspberry Pi ที่อยู่กับหุ่นยนต์ให้เคลื่อนที่ตามที่ได้รับคำสั่งและมีการส่งภาพจากหุ่นยนต์กลับไปหาผู้ควบคุมผ่านทาง live video

L. Hui-ping และคณะ (2011) ได้เขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์ผ่าน TCP\IP protocol ในลักษณะ client/server ซึ่งจะมีคอมพิวเตอร์หนึ่งเครื่องเป็น Server คอยสั่งงาน

ให้กับคอมพิวเตอร์ที่อยู่กับหุ่นยนต์ หุ่นยนต์ที่นำมาใช้ในงานวิจัยจะเป็น MOTOMAN-HP3 และ NPT800

H. Belhadj และคณะ (2013) ได้พัฒนาระบบที่ใช้ความคุมหุ่นยนต์ KUKA KR-125 ผ่านสายอินเตอร์เน็ต IEEE 802.3 ในการส่งสัญญาณภาพให้กับระบบปฏิบัติการของหุ่นยนต์ KAKA เพื่อให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งของสิ่งของในภาพได้

บทที่ 3
วิธีดำเนินงานวิจัย

3.1 กล่าวนำ

งานวิจัยนี้นำเสนอการออกแบบและสร้างเครื่องต้นแบบสำหรับช่วยเหลือในห้องจัดยา โดยเริ่มจากการออกแบบและสร้างเครื่องต้นแบบด้วยอุปกรณ์พื้นฐานต่างๆ และชั้นเก็บตัวอย่างยา จากนั้นออกแบบโปรแกรมส่งและรับคำสั่งการทำงาน โปรแกรมการการคำนวณระยะการเคลื่อนที่ของเครื่องต้นแบบ สุดท้ายเป็นการติดตั้งอุปกรณ์สำหรับหยิบตัวอย่างยาออกจากชั้น

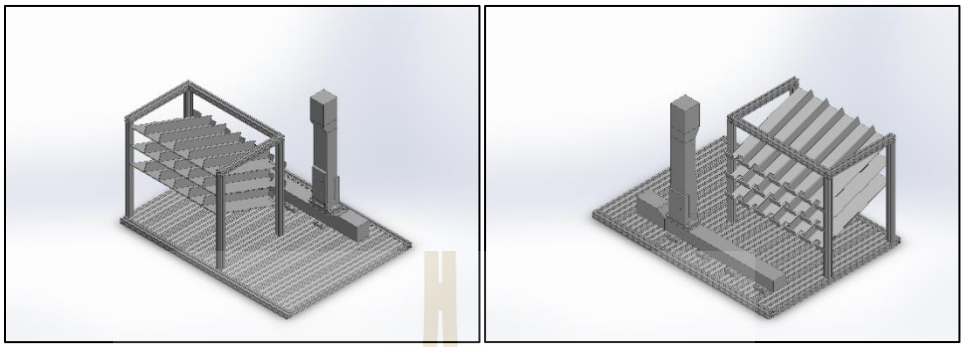
โดยในการสร้างเครื่องต้นแบบได้เลือกนำแกนสำเร็จพร้อมกล่องควบคุมมาใช้เป็นอุปกรณ์ในสร้างเครื่องต้นแบบที่มีความยาวสองขนาดคือ 350 มิลลิเมตร และ 150 มิลลิเมตร เครื่องต้นแบบจะถูกควบคุมด้วยคอมพิวเตอร์บอร์ดเดี่ยว Raspberry Pi โดยการส่งคำสั่งตำแหน่งการเคลื่อนที่ผ่านทางการสื่อสารแบบอนุกรม (Serial communication) รูปแบบการเคลื่อนที่ของเครื่องต้นแบบขจะถูกกำหนดให้มีระยะทางที่สั้นที่สุดผ่านปัญหาการเดินทางของพนักงานขาย

3.2 การออกแบบและสร้างเครื่องต้นแบบ

ในการออกแบบและสร้างเครื่องต้นแบบได้มีการศึกษาทฤษฎีหุ่นยนต์เกี่ยวกับโครงสร้างของหุ่นยนต์มาก่อนการออกแบบ และได้เลือกใช้โครงสร้างแบบคาร์ทีเซียน (Cartesian) ที่มีข้อดีคือการควบคุมตำแหน่งได้ง่ายและโครงสร้างที่ไม่ซับซ้อน มาเป็นโครงสร้างของเครื่องต้นแบบ

เพื่อให้เครื่องต้นแบบทำงานได้อย่างมีประสิทธิภาพจึงเลือกใช้แกนสำเร็จรุ่น IAI ROBO Cylinder RCP2CR-SA5 และ IAI ROBO Cylinder RCP2CR -SA6C มาทำเป็นแกนของเครื่องต้นแบบ โดยที่แกนสำเร็จรุ่น IAI ROBO Cylinder RCP2CR เป็นแกนที่มีส่วนที่เคลื่อนที่เป็นเส้นตรง ถูกควบคุมการทำงานผ่านทางกล่องควบคุมรุ่น IAI ROBO Cylinder Controller RCP2-C เป็นกล่องควบคุมที่สามารถความคุมความเร็ว ตำแหน่งของแกนสำเร็จได้ นิยมนำไปใช้ภาคอุตสาหกรรมที่ต้องการความแม่นยำสูงเพราะเป็นการควบคุมแบบปิด (close loop) หรือการทำคลีนรูม (Cleanroom) ที่ต้องการความสะอาดเนื่องจากตัวแกนสำเร็จมีแผ่นปิดกันรอบตัวแกนทำให้ไม่มีสิ่งสกปรกออกไปนอกตัวแขนได้

เมื่อเลือกอุปกรณ์ที่จะใช้ในการทดลองแล้วจึงได้นำออกแบบโครงสร้าง ชั้นเก็บยาและอุปกรณ์ช่วยในการติดตั้ง ในโปรแกรมโซลิดเวิร์ค



รูปที่ 3.1 การออกแบบเครื่องต้นแบบในโปรแกรมโซลิดเวิร์ค

ตารางที่ 3.1 รายละเอียดของแกนสำเร็จรุ่น IAI ROBO Cylinder RCP2CR-SA5

Motor type	Pulse motor
Encoder type	Incremental
Lead	3 mm
Stork	350 mm
Speed	1-150 mm/s
Power supply voltage	24 VDC
Applicable Controller	RCP2-C

ตารางที่ 3.2 รายละเอียดของแกนสำเร็จรุ่น IAI ROBO Cylinder RCP2CR-SA6C

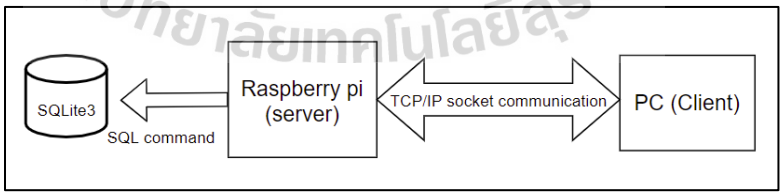
Motor type	Pulse motor
Encoder type	Incremental
Lead	6 mm
Stork	150 mm
Speed	1-300 mm/s
Power supply voltage	24 VDC
Applicable Controller	RCP2-C

ตารางที่ 3.3 รายละเอียดของกล่องควบคุมแกนสำเร็จ IAI ROBO Cylinder Controller RCP2-C

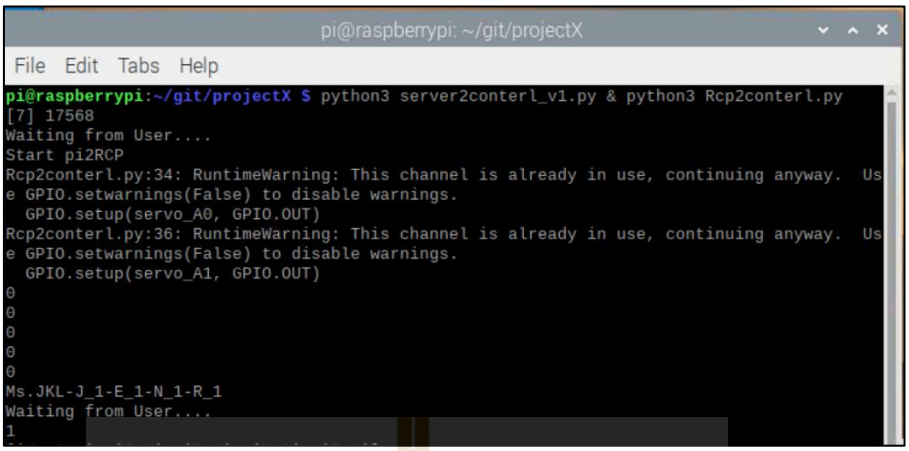
Number of controlled axes	1 axis/unit
Supply voltage	24 VDC ±10%
Control method	Weak field-magnet vector control (patent pending)
Encoder resolution	800 P/rev
Positioning command	Position number specification Direct specification
Position number	Standard 16 points, maximum 64 points
PIO	10 dedicated inputs/10 dedicated outputs. Selectable from five patterns
Communication	RS485 1 channel

3.3 โปรแกรมสำหรับรับและส่งข้อมูล

ถูกเขียนด้วยภาษา python ตัวโปรแกรมจะอยู่บน Raspberry Pi และ บนคอมพิวเตอร์ของผู้ใช้งาน ซึ่งบน Raspberry Pi จะมีโปรแกรมในการรับข้อมูลจากผู้ใช้ที่จะส่งเข้ามาผ่านทาง socket communication และจะถูกแปลงให้อยู่ในรูปแบบของกลุ่มข้อความ (String format) เพื่อจัดเก็บข้อมูลลงฐานข้อมูลบน Raspberry Pi โดยข้อมูลที่จะถูกส่งมาจะประกอบไปด้วย ชื่อของผู้รับยา ชื่อยา และจำนวนของยา



รูปที่ 3.2 การส่งข้อมูลหากันระหว่างผู้ใช้งานกับ Raspberry Pi



รูปที่ 3.3 ลักษณะข้อมูลที่ Raspberry Pi ได้รับ

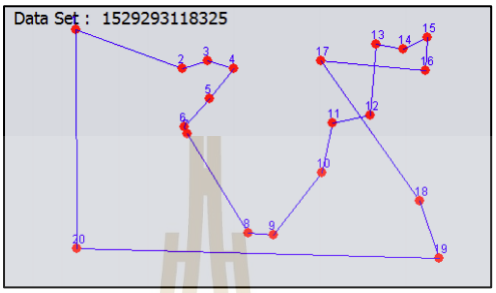
โปรแกรมที่อยู่บนเครื่องของผู้ใช้งานจะเป็นโปรแกรมสำหรับส่งข้อมูลหาเครื่องต้นแบบ โดยจะมีหน้าต่างที่ถูกเขียนขึ้นด้วย python qt-quick เพื่อให้งานต่อการเลือกข้อมูลข้อผู้ใช้

รูปที่ 3.4 หน้าต่างโปรแกรมของผู้ใช้งานในการเลือกยา

3.4 โปรแกรมการคำนวณระยะการเคลื่อนที่ของเครื่องต้นแบบ

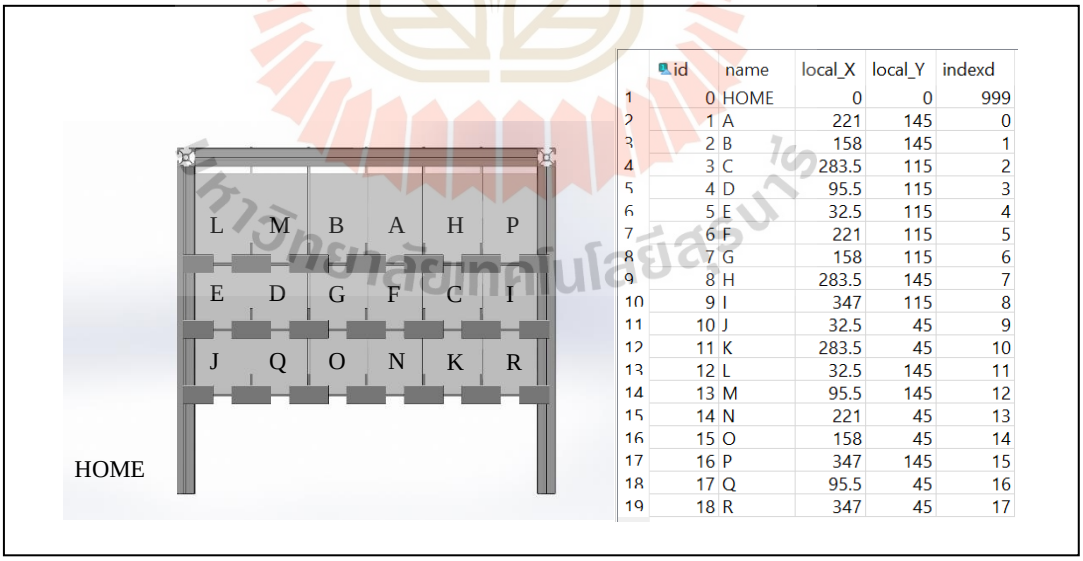
เพื่อใช้ลดระยะทางที่เกิดขึ้นจากการเลือกข้อมูลของผู้ให้ให้สั้นลงและเร็วขึ้นด้วยการผ่านกระบวนการของทฤษฎีปัญหาการเดินทางของพนักงานขาย ด้วยทฤษฎีปัญหาการเดินทางของพนักงานขายมีหลากหลายรูปแบบที่ใช้ในการคิดระยะทาง โดยผู้วิจัยได้เลือกแบบ Nearest-neighbor Heuristic มาใช้ในงานวิจัยนี้ ในการเลือกทฤษฎีปัญหาการเดินทางของพนักงานขายแบบ Nearest-

neighbor Heuristic มาใช้นั้นเป็นวิธีการคิดที่ไม่ซับซ้อนทำให้ความเร็วที่การหาคำตอบออกมาได้เร็ว และมีผลลัพธ์ของระยะทางออกมาได้น่าพอใจแต่ในบางครั้งระยะทางสามารถทำให้สั้นลงได้อีก ด้วยวิธีการนำ Nearest-neighbor Heuristic มาปรับปรุงการคำนวน โดยการหาระยะทางไปจุดต่อไป สลับกับระยะทางที่กลับมาที่จุดเริ่มต้น



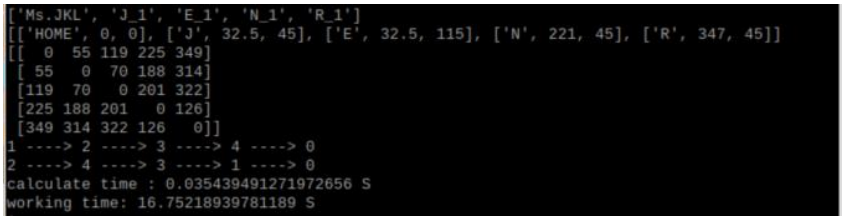
รูปที่ 3.5 การเลือกเส้นทางเดินแบบ Nearest-neighbor Heuristic

การทำงานของโปรแกรมจะทำการนำข้อมูลที่ผู้ใช้งานได้เลือกไปค้นตำแหน่งในฐานข้อมูล เพื่อเก็บตำแหน่งของจุดข้อมูลที่เลือก เมื่อได้ข้อมูลตำแหน่งแล้วจะนำไปคำนวนระยะทาง



	id	name	local_X	local_Y	indexd
1	0	HOME	0	0	999
2	1	A	221	145	0
3	2	B	158	145	1
4	3	C	283.5	115	2
5	4	D	95.5	115	3
6	5	E	32.5	115	4
7	6	F	221	115	5
8	7	G	158	115	6
9	8	H	283.5	145	7
10	9	I	347	115	8
11	10	J	32.5	45	9
12	11	K	283.5	45	10
13	12	L	32.5	145	11
14	13	M	95.5	145	12
15	14	N	221	45	13
16	15	O	158	45	14
17	16	P	347	145	15
18	17	Q	95.5	45	16
19	18	R	347	45	17

รูปที่ 3.6 ตำแหน่งยาในช่องเก็บและพิกัด



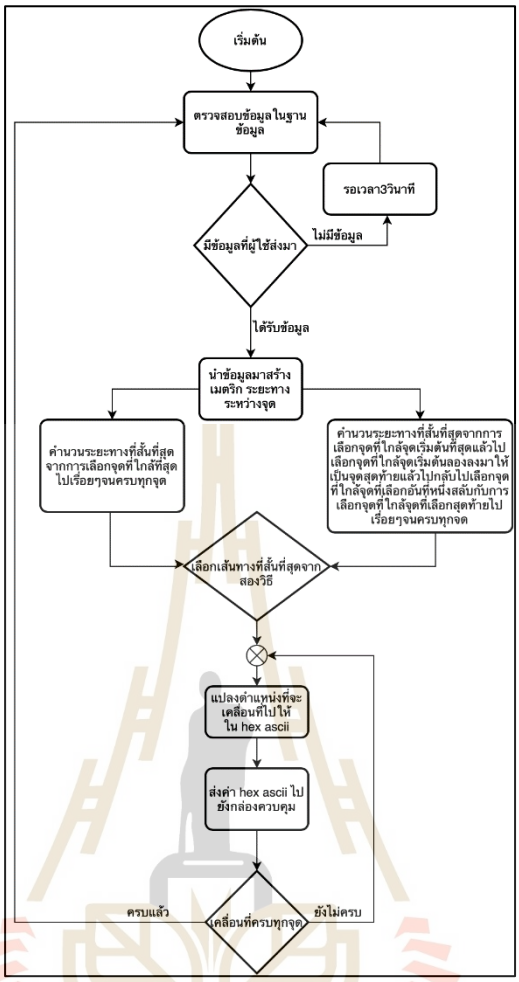
รูปที่ 3.7 คำตอบที่ได้เมื่อผ่านโปรแกรมคำนวนระยะทาง

เมื่อคำนวนระยะห่างของทุกตำแหน่งครบแล้วโปรแกรมจะทำการเตรียมข้อมูลของตำแหน่งให้อยู่ในรูปของ HEX ASCII 16 bit ตามรูปที่ 3.8 เพื่อส่งไปยังกล่องควบคุมให้เคลื่อนที่ไปยังตำแหน่งที่กำหนดผ่านทางการสื่อสารแบบอนุกรม เตรียมข้อมูลจะทำการหาค่าจากช่องที่ขาดไป Axis# จะเป็นลำดับของแกน Position Data (HEX) จะเป็นระยะทางที่ต้องการเคลื่อนที่หารด้วยแปดร้อยที่เป็นความระเอียดของ Encoder แล้วคูณกับระยะพิชของแกน และ BCC เป็นค่าที่เช็คความถูกต้องของคำสั่งหาได้จากการเอาค่าในตำแหน่งที่ 2 ถึง 12 มาบวกกันแล้วเอาสองหลักสุดท้าย

Increment Move	STX	Axis#	'm'	Position Data (HEX)								'0'		BCC		ETX
	02		6D									30	30			03
Inc Move Response	STX	'U'	Axis#	'm'	Status		Alarms		IN		OUT		'0'	BCC		ETX
	02	55		6D									30			03

รูปที่ 3.8 โครงสร้างของคำสั่งที่ใช้ในการเคลื่อนที่ไปยังตำแหน่งใดๆ

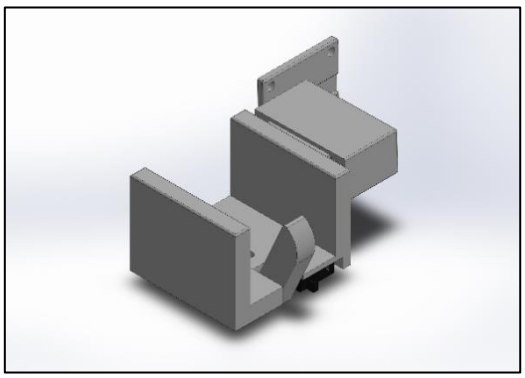
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.9 แผนผังการทำงานของโปรแกรมคำนวนระยะทาง

3.5 อุปกรณ์สำหรับหยิบตัวอย่างยา

เป็นอุปกรณ์ที่ถูกติดตั้งที่ปลายของหุ่นต้นแบบไว้สำหรับนำยาออกจากชั้น โดยในงานวิจัยนี้ได้ออกแบบให้อุปกรณ์สำหรับหยิบตัวอย่างยาตามรูปที่ 3.10 มีขนาด 56.5 x 70 มิลลิเมตรและประกอบไปด้วยอาร์ซีเซอร์โวมอเตอร์ 2 ตัว ที่เป็นรุ่น Tower Pro MG996R และ Tower Pro MG90S โดยซอร์โวมอเตอร์รุ่น Tower Pro MG90S มีหน้าที่ในการควบคุมการเปิดปิดก้านหยิบตัวอย่างยา



รูปที่ 3.10 อุปกรณ์สำหรับหยิบตัวอย่างยา

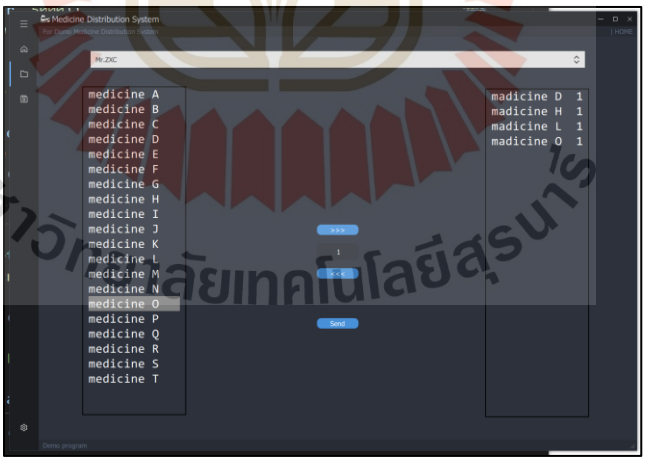
มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 4
ผลการทดลองและอภิปรายผล

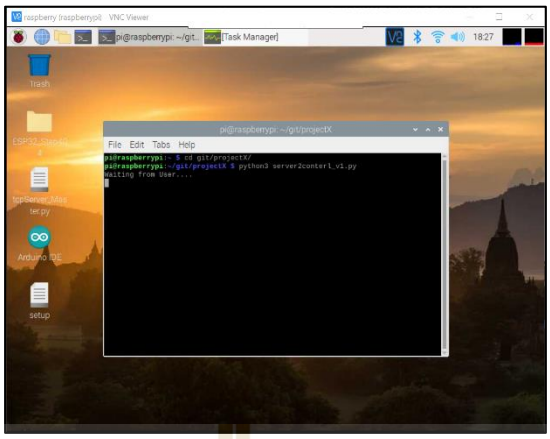
หลังจากที่ได้สร้างเครื่องต้นแบบและโปรแกรมที่ใช้ในการทำงาน เช่น โปรแกรมสำหรับรับส่งข้อมูล โปรแกรมสำหรับคำนวณเส้นทางที่สั้นที่สุด และโปรแกรมสั่งงานแกนสำเร็จพร้อมเซอร์โวมอเตอร์ เรียบร้อยแล้ว ขั้นต้อนถัดมาคือการทดสอบการทำงานของเครื่องต้นแบบกับโปรแกรมในแต่ละส่วนเพื่อทดสอบประสิทธิภาพและความถูกต้องของโปรแกรม

4.1 การทดสอบระบบรับส่งข้อมูล

การทดสอบนี้จะเป็นการตรวจเช็คความถูกต้องของข้อมูลที่ถูกส่งมาจากผู้ใช้กับข้อมูลที่ได้รับบน Raspberry Pi ว่ามีความถูกต้องหรือผิดพลาดเนื่องจากข้อมูลมีการส่งผ่าน TCP/IP socket ที่เป็นเครือข่ายไร้สาย โดยวิธีการทดลองจะทำการส่งข้อมูลที่กำหนดจากหน้าต่างผู้ใช้ไปยังเครื่องต้นแบบจำนวน 5 ครั้ง จากนั้นทดลองสุ่มข้อมูลแล้วส่งไปยังเครื่องต้นแบบอีกครั้ง



รูปที่ 4.1 ข้อมูลที่ถูกเลือกโดยผู้ใช้งาน

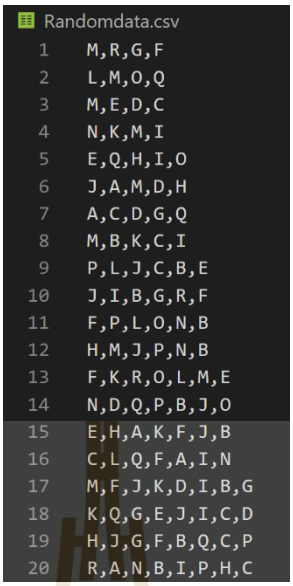


รูปที่ 4.2 การรอรับข้อมูลจากผู้ใช้งานของเครื่องต้นแบบ

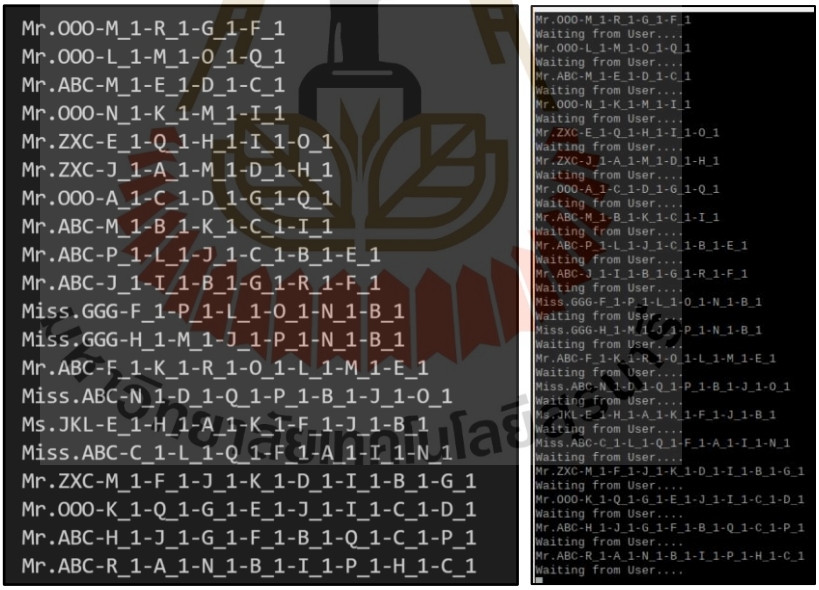
ตารางที่ 4.1 ตารางแสดงข้อมูลที่ผู้เลือกกับข้อมูลที่เครื่องต้นแบบได้รับ

ครั้งที่	ข้อมูลที่ผู้ใช้ส่ง	ข้อมูลที่เครื่องต้นแบบได้รับ
1	Mr.ZXC-D_1-H_1-L_1-O_1	Mr.ZXC-D_1-H_1-L_1-O_1
2	Ms.JKL-E_1-H_1-O_1-S_1-B_1	Ms.JKL-E_1-H_1-O_1-S_1-B_1
3	Mr.OOO-J_1-G_1-C_1-S_1-Q_1-L_1	Mr.OOO-J_1-G_1-C_1-S_1-Q_1-L_1
4	Mr.ABC-N_1-S_1-F_1-B_1-G_1-P_1-T_1	Mr.ABC-N_1-S_1-F_1-B_1-G_1-P_1-T_1
5	Miss.GGG-F_1-I_1-A_1-P_1-T_1-J_1-G_1-E_1	Miss.GGG-F_1-I_1-A_1-P_1-T_1-J_1-G_1-E_1

ทำการสุ่มชุดข้อมูลตำแหน่งของยามาจำนาน 20 ข้อมูลและส่งข้อมูลไปยังเครื่องต้นแบบเพื่อทดสอบความถูกต้องอีกครั้งนึง



รูปที่ 4.3 ตัวอย่างข้อมูลที่ถูกสุ่มขึ้นมาสำหรับการทดสอบ

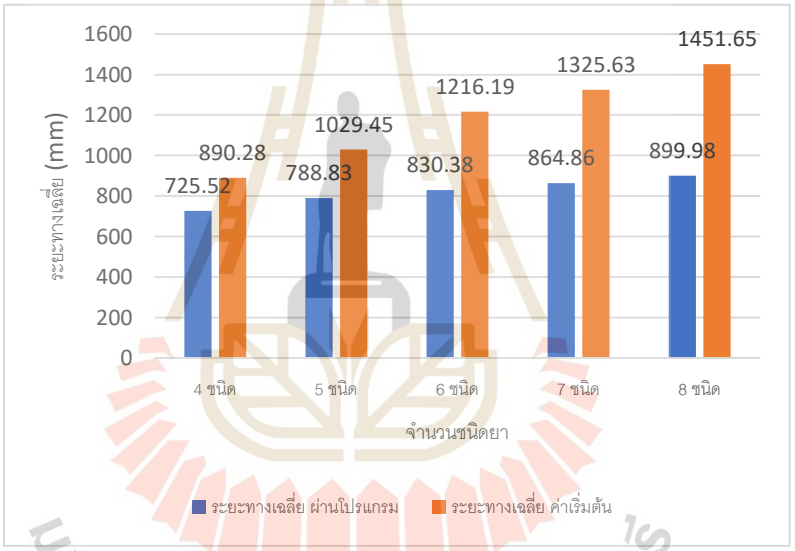


รูปที่ 4.4 การทดสอบการส่งข้อมูลที่ได้สุ่มมา

จากการทดสอบทั้งเลือกชนิดของยาเองและสุ่มตำแหน่งของยามา ข้อมูลที่มีการส่งหาเครื่องต้นแบบมีความถูกต้องทั้งหมด เนื่องจากการในการส่งข้อมูลแบบ TCP/IP socket ที่มีการตรวจสอบความถูกต้องทั้งขณะส่งและหลังส่งข้อมูล

4.2 การทดสอบระบบคำนวณเส้นทางที่สั้นที่สุด

ในการทดสอบนี้จะเป็นการทดสอบประสิทธิภาพของโปรแกรมที่ใช้ในการลดระยะการเคลื่อนที่ของเครื่องต้นแบบให้มีระยะทางที่สั้นลง โดยอัลกอริทึมที่นำมาใช้ในงานวิจัยนี้คือ ปัญหาการเดินทางของพนักงานขายแบบ Nearest-neighbor Heuristic ที่เป็นการมองหาจุดต่อไปที่สั้นที่สุดก่อน และ ปัญหาการเดินทางของพนักงานขายแบบ Nearest-neighbor Heuristicที่มีการปรับปรุงทิศทางการหาคำตอบ ในการทดสอบผู้วิจัยได้ทำการสุ่มข้อมูลมาทั้งหมด 500 ชุด แบ่งเป็นข้อมูลที่มีจำนวนยา 4 ชนิด 100 ชุด จำนวนยา 5 ชนิด 100 ชุด จำนวนยา 6 ชนิด 100 ชุด จำนวนยา 7 ชนิด 100 ชุด และจำนวนยา 8 ชนิด 100 ชุด นำมาผ่านโปรแกรมลดระยะทางเทียบกับระยะที่ไม่ได้ผ่านโปรแกรม ได้ผลการทดลองตามรูปที่ 4.5



รูปที่ 4.5 ระยะทางเฉลี่ยเมื่อผ่านโปรแกรมกับค่าเริ่มต้น

ค่าเฉลี่ยของระยะทางของยาจำนวน 4 ชนิด ค่าเริ่มต้นอยู่ที่ 890.28 มิลลิเมตร และเมื่อผ่านโปรแกรมจะอยู่ที่ 725.52 มิลลิเมตร ระยะทางลดลง 164.76 มิลลิเมตร

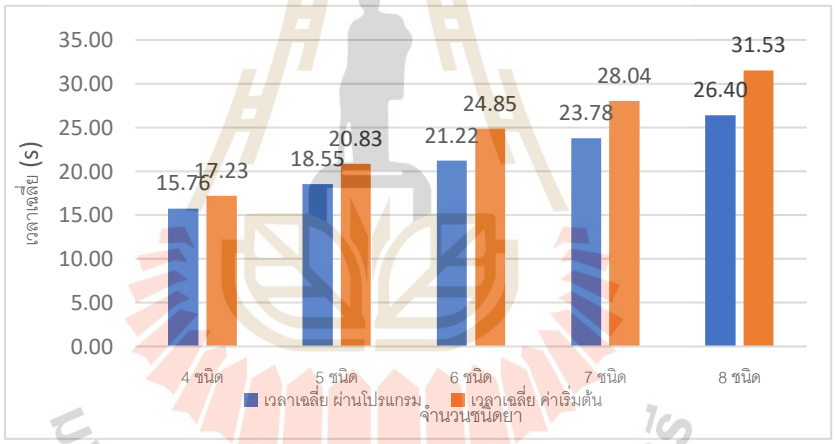
ค่าเฉลี่ยของระยะทางของยาจำนวน 5 ชนิด ค่าเริ่มต้นอยู่ที่ 1029.45 มิลลิเมตร และเมื่อผ่านโปรแกรมจะอยู่ที่ 788.83 มิลลิเมตร ระยะทางลดลง 240.62 มิลลิเมตร

ค่าเฉลี่ยของระยะทางของยาจำนวน 6 ชนิด ค่าเริ่มต้นอยู่ที่ 1216.19 มิลลิเมตร และเมื่อผ่านโปรแกรมจะอยู่ที่ 830.38 มิลลิเมตร ระยะทางลดลง 385.81 มิลลิเมตร

ค่าเฉลี่ยของระยะทางของยาจำนวน 7 ชนิด ค่าเริ่มต้นอยู่ที่ 1325.63 มิลลิเมตร และเมื่อผ่านโปรแกรมจะอยู่ที่ 864.86 มิลลิเมตร ระยะทางลดลง 460.77 มิลลิเมตร
ค่าเฉลี่ยของระยะทางของยาจำนวน 8 ชนิด ค่าเริ่มต้นอยู่ที่ 1451.65 มิลลิเมตร และเมื่อผ่านโปรแกรมจะอยู่ที่ 899.98 มิลลิเมตร ระยะทางลดลง 551.67 มิลลิเมตร

4.3 การทดสอบเวลาการทำของเครื่องต้นแบบและความถูกต้องในการเคลื่อนที่

ในการทดสอบนี้จะเป็นการทดสอบเวลาการทำงานของเครื่องต้นแบบทั้งระบบตั้งแต่การเลือกจำนวนยา ลดระยะทางการเคลื่อนที่ ไปจนถึงการทำงานของอาซีเซอร์โวมอเตอร์ เทียบกับเวลาการทำงานทั้งระบบเหมือนกันจะแตกต่างตรงที่ระยะทางการเคลื่อนที่เป็นค่าเริ่มต้นและความถูกต้องของการเคลื่อนที่ไปยังตำแหน่งต่างๆที่รับคำสั่งไว้ โดนข้อมูลที่ใช้เป็นข้อมูลแบบสุ่มที่เหมือนในหัวข้อที่แล้ว ได้ผลการตามรูปที่ 4.6



รูปที่ 4.6 เวลาเฉลี่ยเมื่อผ่านโปรแกรมกับค่าเริ่มต้น

เวลาเฉลี่ยของการทำงานทั้งระบบที่มีระยะทางเริ่มต้นของยา 4 ชนิด อยู่ที่ 17.23 วินาที และเวลาเฉลี่ยของการทำงานที่มีระยะทางผ่านโปรแกรมอยู่ที่ 15.76 วินาที เวลาเฉลี่ยที่ลดลง 1.47 วินาที
เวลาเฉลี่ยของการทำงานทั้งระบบที่มีระยะทางเริ่มต้นของยา 5 ชนิด อยู่ที่ 20.83 วินาที และเวลาเฉลี่ยของการทำงานที่มีระยะทางผ่านโปรแกรมอยู่ที่ 18.55 วินาที เวลาเฉลี่ยที่ลดลง 2.28 วินาที
เวลาเฉลี่ยของการทำงานทั้งระบบที่มีระยะทางเริ่มต้นของยา 6 ชนิด อยู่ที่ 24.85 วินาที และเวลาเฉลี่ยของการทำงานที่มีระยะทางผ่านโปรแกรมอยู่ที่ 21.22 วินาที เวลาเฉลี่ยที่ลดลง 3.63 วินาที

เวลาเฉลี่ยของการทำงานทั้งระบบที่มีระยะทางเริ่มต้นของยา 7 ชนิด อยู่ที่ 28.04 วินาที และเวลาเฉลี่ยของการทำงานที่มีระยะทางผ่านโปรแกรมอยู่ที่ 23.78 วินาที เวลาเฉลี่ยที่ลดลง 4.26 วินาที เวลาเฉลี่ยของการทำงานทั้งระบบที่มีระยะทางเริ่มต้นของยา 8 ชนิด อยู่ที่ 31.53 วินาที และเวลาเฉลี่ยของการทำงานที่มีระยะทางผ่านโปรแกรมอยู่ที่ 26.40 วินาที เวลาเฉลี่ยที่ลดลง 5.13 วินาที ความถูกต้องของเครื่องต้นแบบที่เคลื่อนที่ไปยังตำแหน่งที่ได้รับมามีความถูกต้องทุกตำแหน่งที่ได้รับ หลังจากการเคลื่อนที่ครบทุกตำแหน่งในแต่ละรอบจะมีการส่งคำสั่งให้เครื่องต้นแบบกลับไปยังจุดเริ่มต้นเพื่อนับค่าเอ็นโค้ดเดอร์ใหม่ให้กลายเป็นศูนย์เพื่อเพิ่มความถูกต้องในการเคลื่อนที่รอบถัดไป

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอออกแบบและสร้างเครื่องต้นแบบสำหรับช่วยเหลือในห้องจัดยาและได้ทำการเขียนโปรกรมสำหรับการส่งข้อมูลผ่านทางเครือข่ายไร้สาย โปรแกรมสำหรับคำนวณเส้นทางที่สั้นที่สุด และโปรแกรมสั่งงานแกนสำเร็จพร้อมเซอร์โวมอเตอร์ โดยมีรายละเอียดของข้อสรุปดั้งนี้

1. เครื่องต้นแบบสำหรับช่วยเหลือในห้องจัดยามีโครงสร้างแบบคาร์ทีเซียน พื้นที่ทำงาน 350 x 150 มิลลิเมตร

2. เครื่องต้นแบบสามารถรับข้อมูลจากผู้ใช้งานผ่านเครือข่ายไร้สาย และแปลงข้อมูลที่ได้เป็นตำแหน่งของการเคลื่อนที่แล้วส่งไปให้แกนสำเร็จเคลื่อนที่

3. เครื่องต้นแบบสามรถลดความผิดพลาดลงได้จากการรับข้อมูลแบบ TCP/IP ที่มีการตรวจสอบข้อมูลก่อนรับส่งข้อมูลและในการเคลื่อนที่ของเครื่องต้นแบบมีการนับค่าระยะทางใหม่ทุกครั้งที่มีการเริ่มรอบการทำงานใหม่

4 ระยะทางและเวลาในการทำงานของเครื่องต้นแบบเมื่อผ่านโปรแกรมคำนวนระยะทางมีค่ามีค่าลดลงตามต่อไปนี้

จำนวนยา 4 ชนิด ระยะทางเฉลี่ยลดลงจากค่าเริ่มต้น 164.76 มิลลิเมตร หรือคิดเป็น 18.5% และเวลาการทำงานลดลง 1.47 วินาที

จำนวนยา 5 ชนิด ระยะทางเฉลี่ยลดลงจากค่าเริ่มต้น 240.62 มิลลิเมตร หรือคิดเป็น 23.4% และเวลาการทำงานลดลง 2.28 วินาที

จำนวนยา 6 ชนิด ระยะทางเฉลี่ยลดลงจากค่าเริ่มต้น 385.81 มิลลิเมตร หรือคิดเป็น 31.7% และเวลาการทำงานลดลง 3.63 วินาที

จำนวนยา 7 ชนิด ระยะทางเฉลี่ยลดลงจากค่าเริ่มต้น 460.77 มิลลิเมตร หรือคิดเป็น 34.8% และเวลาการทำงานลดลง 4.26 วินาที

จำนวนยา 8 ชนิด ระยะทางเฉลี่ยลดลงจากค่าเริ่มต้น 551.67 มิลลิเมตร หรือคิดเป็น 38.0% และเวลาการทำงานลดลง 5.13 วินาที

5.2 ข้อเสนอแนะ

1. เนื่องจากงานวิจัยนี้เป็นการออกแบบเครื่องต้นแบบที่เน้นไปทางด้านการออกแบบระบบการทำงานจึงทำให้มีอุปกรณ์บ้างส่วนมีโครงสร้างยังไม่ค่อยแข็งแรงและอุปกรณ์บ้างชิ้นไม่สามารถหาได้ตามที่ออกแบบไว้

2. การเขียนโปรแกรมหน้าต่างโปรแกรมของผู้ใช้ยังไม่มีเหมาะกับการนำไปใช้งานจริงควรเปลี่ยนเป็นการสร้างหน้าเว็ปไซต์แทนเพื่อเพิ่มความสะดวกในการเข้าถึงข้อมูล

3. การออกแบบอุปกรณ์สำหรับหยิบตัวอย่างยามีขนาดที่ค่อนข้างเล็กเมื่อเทียบกับกล่องใส่ยาที่ใช้ในการทดสอบแก้ไขได้โดยการออกแบบอุปกรณ์ให้มีขนาดที่เหมาะสม

4. การเปลี่ยนกล่องควบคุมให้สามารถสั่งงานแกนสำเร็จได้สองแกนพร้อมกันจะทำการทำงานของเครื่องต้นแบบมีความสมบูรณ์ขึ้น

5. เนื่องจากคุณสมบัติของแกนสองอันมีความแต่ต่างกันในเรื่องของความละเอียดLead จึงทำให้มีผลกับความเร็วในการเคลื่อนที่ของแกนทั้งสองอาจจะต้องมีการปรับคุณสมบัติให้เท่ากันเพื่อให้ได้ความเร็วที่เท่ากัน

6. เครื่องต้นแบบสำหรับช่วยเหลือในห้องจัดยานี้มีแค่ในส่วนของการจัดยาเท่านั้นยังมีส่วนที่สามารถที่จะนำไปพัฒนาต่อยอดได้อีกหลายส่วนไม่ว่าจะเป็นในส่วนของการบรรจุยากลับเข้าไปในชั้นเก็บ หรือส่วนสำหรับลำเลียงยาที่จัดแล้วออกจากห้องยา

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายการอ้างอิง

mittal r. k. & nagrath i. j. (2003). robotics and control, MC GRAW HILL. INDIA

รศ.ดร.ชิต เหล่าวัฒนา (2553). ตามรอยหุ่นยนต์อดีตถึงปัจจุบัน,สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น),กรุงเทพมหานคร

W. Richard Stevens (1994). TCP/IP Illustrated, Volume 1, Addison–Wesley, United States

นคร ไชยวงศ์ศักดา (2558), การจัดเส้นทางการขนส่งโดยใช้เซฟวิ่งอัลกอลิทึมและตัวแบบปัญหาการเดินทางของพนักงานขาย กรณีศึกษาโรงงานน้ำดื่ม, วารสารไทยการวิจัยดำเนินงาน, เล่มที่ 3, ฉบับที่ 11, หน้า 51-60

ระพีพันธ์ ปิตาคะโส (2559), วิธีการวิวัฒนาการโดยใช้ผลต่างสำหรับแก้ปัญหาการขนส่งโลจิสติกส์, โรงพิมพ์มหาวิทยาลัยอุบลราชธานี, อุบลราชธานี

อภิรักษ์ ขัดวิลาศ (2554), การประยุกต์วิธีเชิงพันธุกรรมสำหรับปัญหาการหาค่าที่เหมาะสมที่สุด, วารสารวิชาการและวิจัย มทร.พระนคร,เล่มที่ 12, หน้า 154-163

ระพีพันธ์ ปิตาคะโส (2554), วิธีการเมตาฮิวริสติกเพื่อแก้ไขปัญหาการวางแผนการผลิตและการจัดการโลจิสติกส์, กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น),

G. Dantzig, R. Fulkerson and S. Johnson (1954), Solution of a large-scale traveling-salesman problem, Operation Research, vol. 11, no. 6, pp. 972-989

D. Goldberg (1989), Genetic Algorithms in Search, Optimisation and Machine Learning, Massachusetts, Addison-Wesley.

ชวันลักษณ์ สุวรรณรัศมี (2559), การจัดเส้นทางสำหรับการให้บริการลูกค้าของตัวแทนจำหน่ายอุปกรณ์ทางวิศวกรรมในภาคใต้ของประเทศไทย,มหาวิทยาลัยสงขลานครินทร์.สงขลา

K. A. Radhika, B. L. Raksha, B. R. Sujatha, U. Pruthviraj and K. V. Gangadharan (2018), IoT Based Joystick Controlled Pibot Using Socket Communication, IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 121-125

L. Hui-ping, C. Dai-min and Y. Miao (2011), Communication of multi-robot system on the TCP/IP, International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), pp. 1432-1435

H. Belhadj, S. Ben Hassen, K. Kaâniche and H. Mekki (2013), KUKA robot control based Kinect image analysis, 2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR), 2013, pp. 21-26

ภาคผนวก ก

โปรแกรมที่ได้เขียนขึ้นในงานวิจัย

มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรมหน้าต่างผู้ใช้งาน

```
import sys
import os
import datetime
import socket

from PySide2.QtGui import QGuiApplication
from PySide2.QtQml import QQmlApplicationEngine
from PySide2.QtCore import QObject, Slot, Signal, QTimer, QUrl
from PySide2.QtCore import Property, QStringListModel

class MainWindow(QObject):
    def __init__(self):
        QObject.__init__(self)
        # QTimer - Run Timer
        self.timer = QTimer()
        self.timer.timeout.connect(lambda: self.setTime())
        self.timer.start(1000)
        self._model = QStringListModel()

    # Signal Set Name
    setName = Signal(str)

    # Signal Set Data
    printTime = Signal(str)

    # Signal Visible
    isVisible = Signal(bool)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
# Open File To Text Edit
readText = Signal(str)

# Text String
textField = ""

medicSeclect = ""
dataraw = ""
medicines = []
medicineShow = []
addr = '192.168.1.132'

@Slot(str)
def send2pi(self, text):
    self.dataraw = text
    for i in range(len(self.medicines)):
        self.dataraw = self.dataraw + "-" + self.medicines[i]
    print(self.dataraw)

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    socket.setdefaulttimeout(1)
    s.connect((self.addr, 5000))
    s.send(self.dataraw.encode())
    s.close()

# Show / Hide Rectangle
@Slot(bool)
def showHideRectangle(self, isChecked):
    print("Is rectangle visible: ", isChecked)
    self.isVisible.emit(isChecked)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
# Set Timer Function
def setTime(self):
    now = datetime.datetime.now()
    formatDate = now.strftime("Now is %H:%M:%S %p of %Y/%m/%d")
    print(formatDate)
    self.printTime.emit(formatDate)

# Function Set Name To Label
@Slot(str)
def welcomeText(self, name):
    self.setName.emit("Welcome Docter, " + name)

@Slot(str)
def getMedicine(self, medicine):
    self.medicSeclect = medicine
    print("madicine select :", self.medicSeclect)

@Slot(str)
def medicineList(self, amount):
    self.medicineShow.append("madicine "+self.medicSeclect + "  " + amount)
    self.medicines.append(self.medicSeclect+"_"+amount)

    self._model.setStringList(self.medicineShow)
    print(self.medicines)

@Slot(int)
def medicieDel(self, mDel):
    self.medicines.pop(mDel)
    self.medicineShow.pop(mDel)
    self._model.setStringList(self.medicineShow)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    @Property(QObject, medicineList, medicieDel)
    def model(self):
        return self._model


if __name__ == "__main__":
    app = QGuiApplication(sys.argv)
    engine = QQmlApplicationEngine()

    # Get Context
    main = MainWindow()
    engine.rootContext().setContextProperty("backend", main)

    # Set App Extra Info
    app.setOrganizationName("Wanderson M. Pimenta")
    app.setOrganizationDomain("N/A")

    # Load QML File
    engine.load(os.path.join(os.path.dirname(__file__), "qml/main.qml"))
    if not engine.rootObjects():
        sys.exit(-1)
    sys.exit(app.exec_())
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรมรับข้อมูล

```
import sys
import glob
import re
import codecs
import serial
import time
import sqlite3
import socket

def mainServer():
    host = '192.168.1.132'
    port = 5000

    while(1) :
        soc = socket.socket()
        soc.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR,1)
        soc.bind((host,port))

        soc.listen(1)
        print("Waiting from User....")
        con, addr = soc.accept()
        dataraw = con.recv(1024).decode('utf-8')
        print(dataraw)
        saveData2DB(dataraw)

def saveData2DB(dataraw):
    #print(datafromUser)
    connection = sqlite3.connect('project_test.db')
    cursor = connection.cursor()
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    sql_command = "INSERT INTO dataformUser (dataraw ) VALUES ( ? )"
    cursor.execute(sql_command, (dataraw,))
    connection.commit()
    connection.close()

if __name__ == "__main__":
    mainServer()
```

โปรแกรมคำนวนระยะทาง

```
import sys
import glob
import re
import codecs
import serial
import time
import RPi.GPIO as GPIO
import sqlite3
import numpy as np
import csv

servo_A0 = 27
servo_A1 = 22
inMoveA0 = 23
inMoveA1 = 24

rud = 0
sumq = 0
aryPosition = []
partA =  []
completed = []
n = 0
cost = 0
sortRoute = []

def main2RPC():
   print("Start pi2RCP")
   GPIO.setmode(GPIO.BCM)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
GPIO.setup(servo_A0, GPIO.OUT)
servo1 = GPIO.PWM(servo_A0,50) # pin 27 for servo1
GPIO.setup(servo_A1, GPIO.OUT)
servo2 = GPIO.PWM(servo_A1,50) # pin 22 for servo2
servo1.start(0)
servo2.start(0)

GPIO.setup(inMoveA0, GPIO.IN)
GPIO.setup(inMoveA1, GPIO.IN)
ser = serial.Serial(port = "/dev/ttyUSB0",
                    baudrate = 38400,
                    bytesize = serial.EIGHTBITS,
                    parity=serial.PARITY_NONE,
                    stopbits=serial.STOPBITS_ONE,
                    write_timeout=5.0,
                    inter_byte_timeout=1.0,
                    timeout=1)
starT = time.time()
summ = 0
while(1) :

   if time.time() - starT > 3 :
          tick = time.time()
          starT = time.time()
          connection = sqlite3.connect('project_test.db')
          cursor = connection.cursor()
          sql_command = "SELECT dataraw FROM dataformUser "
          cursor.execute(sql_command)
          results = cursor.fetchall()
          summ =summ+len(results)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
print(summ)
#print("inMoveA0 :",GPIO.input(inMoveA0),"inMoveA1 :",GPIO.input(inMoveA1))

if results is not None :
    sql_command ="DELETE from dataformUser"
    cursor.execute(sql_command)
    connection.commit()

connection.close()

if results is not None :
    for result in results:
        result = str(result)[2:-3]
        result = list(result.split("-"))
        print(result)
        pat = result[0]
        result.pop(0)
        madical = []
        amount = []
        for i in range(len(result)):
            xxx = list(result[i].split("_"))
            madical.append(xxx[0])
            amount.append(xxx[1])
        #print(madical)
        #print(amount)

        madicalDis = select_DB(madical)
        routes,costRoute,initRoute,initSelects = TSP(madicalDis)
        toc = time.time()
        calculate = toc-tick
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
print("calculate time :",calculate,"S")
asciiTextA0 = ""
asciiTextA1 = ""
servoToggle = False
for route in initSelects:
    #print(route[1],route[2])
    distanceA0 = int(route[1])
    distanceA1 = int(route[2])
    distanceA0 = round((distanceA0*800)/3)
    #distanceA1 = round((distanceA1*800)/6)
    if(distanceA1 == 0):
        #print(distanceA1,"A0")
        distanceA1 = round((distanceA1*800)/6)
        servoToggle = False
    elif (distanceA1 < 120):
        #print(distanceA1,"A1")
        distanceA1 = round(((distanceA1-40)*800)/6)
        servoToggle = True
    elif (distanceA1 > 120):
        #print(distanceA1,"A2")
        distanceA1 = round(((distanceA1-10)*800)/6)
        servoToggle = True

    #print("distanceA0 :", distanceA0 ," distanceA1 :", distanceA1 ,"servoToggle:",
servoToggle)
    BCC_A0 = 241 #hextodec Axis0
    BCC_A1 = 242 #hextodec Axis1
    text_A0 = "02 30 61"
    text_A1 = "02 31 61"
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
dis2hex_A0 =  hex(4294967295 - int(distanceA0))
dis2hex_A0 =  dis2hex_A0[2:].upper()
dis2hex_A1 =  format(int(distanceA1),'#010x')
dis2hex_A1 =  dis2hex_A1[2:].upper()

for i in range(len(dis2hex_A0)):
    ha = ord(dis2hex_A0[i])
    ha = hex(ha)
    BCC_A0 = int(ha[2:],16) + BCC_A0
    text_A0 = text_A0 + " " + ha[2:]
for i in range(len(dis2hex_A1)):
    ha = ord(dis2hex_A1[i])
    ha = hex(ha)
    BCC_A1 = int(ha[2:],16) + BCC_A1
    text_A1 = text_A1 + " " + ha[2:]

text_A0 = text_A0 + " 30 30"
text_A1 = text_A1 + " 30 30"
bcc1_A0,bcc2_A0 = bccCal(hex(BCC_A0).upper())
bcc1_A1,bcc2_A1 = bccCal(hex(BCC_A1).upper())

text_A0 = text_A0+" "+ bcc1_A0 +" "+ bcc2_A0 +" 03"
text_A1 = text_A1+" "+ bcc1_A1 +" "+ bcc2_A1 +" 03"
text_A0 = ''.join(text_A0.split())
text_A1 = ''.join(text_A1.split())
text_A0 = hex_to_raw(text_A0)
text_A1 = hex_to_raw(text_A1)
text_A0 = text_A0.encode()
text_A1 = text_A1.encode()
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
ser.write(text_A0)
time.sleep(0.25)
ser.write(text_A1)
time.sleep(0.25)
#msgCount = 0
while (GPIO.input(inMoveA0)==0) or (GPIO.input(inMoveA1)==0):
   pass
if servoToggle :
   servo1.ChangeDutyCycle(10.5)
   time.sleep(0.5)
   servo1.ChangeDutyCycle(0)
   #print(msg)
   time.sleep(0.5)
   distanceA1 = int(route[2])
   distanceA1 = round((distanceA1*800)/6)
   BCC_A1 = 242 #hextodec Axis1
   text_A1 = "02 31 61"
   dis2hex_A1 =  format(int(distanceA1),'#010x')
   dis2hex_A1 =  dis2hex_A1[2:].upper()
   for i in range(len(dis2hex_A1)):
      ha = ord(dis2hex_A1[i])
      ha = hex(ha)
      BCC_A1 = int(ha[2:],16) + BCC_A1
      text_A1 = text_A1 + " " + ha[2:]
   text_A1 = text_A1 + " 30 30"
   bcc1_A1,bcc2_A1 = bccCal(hex(BCC_A1).upper())
   text_A1 = text_A1+" "+ bcc1_A1 +" "+ bcc2_A1 +" 03"

   text_A1 = ''.join(text_A1.split())
   text_A1 = hex_to_raw(text_A1)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
                    text_A1 = text_A1.encode()
                    ser.write(text_A1)
                    while (GPIO.input(inMoveA1)==0):
                        pass
                    #print("A1123")
                    duty = 10
                    while duty >= 7:
                        servo1.ChangeDutyCycle(duty)
                        time.sleep(0.2)
                        duty = duty - 1
                    servo1.ChangeDutyCycle(0)
                    time.sleep(0.1)
                time.sleep(0.25)


            toc = time.time()
            working = toc-tick
            print("working time:",working,"S")
            with open('programRun.csv', 'a', newline='') as file:
                writer = csv.writer(file)

writer.writerow([pathA,cost,pathB,costB,calculate,working,costRoute,initRoute])
            tick = time.time()


def select_DB(datas):#step2
    xxx = []
    datas.insert(0,'HOME')
    for i in range(len(datas)):
        data = datas[i]
        connection = sqlite3.connect('project_test.db')
        cursor = connection.cursor()
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
        sql_command = "SELECT * FROM medicine_data WHERE name = ?;"
        cursor.execute(sql_command,(data, ))
        results = cursor.fetchall()
        for row in results:
            #print (row)
            name = row[1]
            locx = row[2]
            locy = row[3]
            xxx.append([name, locx, locy])

        connection.close()

    print(xxx)
    return xxx

def TSP(listSelects):#step3
    global ary
    global pathA
    global completed
    global cost
    global pathB
    global costB
    global com_RouteB
    pathB = []
    costB = 0
    ary = np.arange(0)
    num1 = []
    num2 = []
    newarr = []
    completed = []
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
pathA = []
cost = 0
for listSelect in listSelects :
    n1 = listSelect[1]
    n2 = listSelect[2]
    newarr.append([n1, n2])
    num1.append(n1)
    num2.append(n2)

ary.resize(len(num1),len(num1))
minn = ary[0][0]
position = 0

global aryPosition
global sumq
global rud
global sortRoute

com_RouteB = []
sortRoute = []
for i in range(len(num1)): # คำนวนระยะห่างระหว่างจุด 2 จุด
    for j in range(len(num1)):
        xi = np.power((num1[i]-num1[j]),2)
        yi = np.power((num2[i]-num2[j]),2)
        d = np.sqrt(xi+(yi))
        ary[i][j] = d
    completed.append(0)
    com_RouteB.append(0)
n = len(num1)
print(ary)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
mincostA(0,n)
mincostB(0,n)
initRoute = 0
initSelect= []
for i in range(n):
    if i + 1  ==  n :
        initRoute = initRoute +  ary[i][0]
        initSelect.append(listSelects[0])

    else:
        initRoute = initRoute +  ary[i][i+1]
        initSelect.append(listSelects[i+1])

for i in range(len(pathA)):
    if i == len(pathA)-1:
        print(pathA[i])
    else :
        print(pathA[i],"---->",end = " ")
    if (cost<=costB):
        sortRoute.append(listSelects[pathA[i]])
for i in range(len(pathB)):
    if i == len(pathB)-1:
        print(pathB[i])
    else :
        print(pathB[i],"---->",end = " ")
    if (costB<cost):
        sortRoute.append(listSelects[pathB[i]])
costRoute = cost if cost <= costB else costB

return sortRoute ,costRoute, initRoute ,initSelect
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
#////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\#

def mincostB(city,n):
    global costB
    ncity = 0
    com_RouteB[city]=1
    #print(pathB)
    #print(com_RouteB)
    if (len(pathB)<2):
        city = 0
    else :
        if(len(pathB)%2==0):
            city =pathB[int(len(pathB)/2)]
        else:
            city =pathB[int(len(pathB)/2)-1]
    ncity  = routeB(city,n)
    if (ncity == 0):
        ncity = pathB[int(len(pathB)/2)]

        city = pathB[int(len(pathB)/2)-1]
        #print(ncity +1)
        costB += ary[city][ncity]
        #print("costB:",costB,"city:",city)
        pathB.append(0)
        return

    mincostB(ncity,n)

def routeB(c,n):
    global pathB
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    global costB
    nc=0
    mint = 999
    for i in range(n):
        #print("i:",i,"c:",c,"ary:", ary[c][i])
        if (ary[c][i]!=0 and com_RouteB[i] == 0):
            if(ary[c][i] <= mint ):
                mint = ary[c][i]
                nc = i
                #print("mint:",mint)

    if(mint != 999):
        #print("costB:",costB)
        costB+=mint
        if(int(len(pathB)%2)==0):
            pathB.insert(int(len(pathB)/2),nc)
        else:
            pathB.insert(int(len(pathB)/2),nc)
    #print("pathB:",len(pathB),"n:",n )
    #print("pathB",len(pathB),"nc:",nc,"costB:",costB)
    return nc

 #////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\#
def mincostA (city,n):
    global cost
    ncity = 0

    completed[city] = 1
    #print("city: ",city,"completed : ",completed)
    #print(city+1,"--->")
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    ncity = least(city,n)
    #print("ncity: ",ncity)
    if (ncity == 0):
        ncity = 0

        #print(ncity +1)
        cost += ary[city][ncity]
        return
    mincostA(ncity,n)

def least(c,n):
    global cost
    global pathA
    nc = 0
    mint = 999
    #print('n',n)
    for i in range(n):
        #print('c :', c, " i :",i,"ary : ",ary[c][i])
        if (ary[c][i]!=0 and completed[i] == 0):
            #print("mint :", mint)
            if(ary[c][i] + ary[i][c] <= mint ):
                mint = ary[i][c] + ary[c][i]
                #print('min :',mint)
                kmin = ary[c][i]
                nc = i

    if(mint != 999):
        cost += kmin
    pathA.append(nc)
    return nc
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
#////////////////////////////\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\#
def bccCal(bcc):
    BCC1 = bcc[-2]
    BCC2 = bcc[-1]

    BCC1_bin = bin(int(BCC1,16))[2:]
    BCC2_bin = bin(int(BCC2,16))[2:]
    #print(len(BCC1_bin),len(BCC2_bin))
    if len(BCC1_bin) != 4:
        for i in range(4- len(BCC1_bin)):
            BCC1_bin = "0" + BCC1_bin
        BCC1_bin = bccinvet(BCC1_bin)
    else:
        BCC1_bin = bccinvet(BCC1_bin)

    #print(BCC1_bin,BCC2_bin)

    if len(BCC2_bin) != 4:
        for i in range(4 - len(BCC2_bin)):
            BCC2_bin = "0" + BCC2_bin
        #print(BCC2_bin)
        BCC2_bin = bccinvet(BCC2_bin)
    else:
        BCC2_bin = bccinvet(BCC2_bin)
    #print(BCC1_bin,BCC2_bin)
    if BCC2_bin != "1111":
        BCC1 = hex(int(BCC1_bin,2)).upper()
        BCC1 = hex(ord(BCC1[2:]))[2:]
        BCC2 = hex(int(BCC2_bin,2)+1).upper()
        BCC2 = hex(ord(BCC2[2:]))[2:]
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
    else:
        BCC2_bin = "0000"
        BCC1 = hex(int(BCC1_bin,2)+1).upper()
        BCC1 = hex(ord(BCC1[2:]))[2:]
        BCC2 = hex(int(BCC2_bin,2)).upper()
        BCC2 = hex(ord(BCC2[2:]))[2:]
    #print(BCC1,BCC2)
    return BCC1,BCC2

def bccinvet(bcc):
    inverse_s = ''
    for i in bcc:
        if i == '0':
            inverse_s += '1'
        else:
            inverse_s += '0'
    return inverse_s

def _chunks(text, chunk_size):
    """Chunk text into chunk_size."""
    for i in range(0, len(text), chunk_size):
        yield text[i:i+chunk_size]

def hex_to_raw(hexstr):
    """Convert a hex encoded string to raw bytes."""
    return ''.join(chr(int(x, 16)) for x in _chunks(hexstr, 2))

if __name__ == "__main__":

    main2RPC()
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ข

บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายชื่อบทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

Sittinon Julnithi, Aphilak Lonklang, Jittima Varagul and Kontorn Chamniprasart (2021), **Robot Teaching Laboratory Development For Effective Remote Delivery**. Proceedings of WRFER International Conference, Phuket, Thailand, 08th May, 2021

มหาวิทยาลัยเทคโนโลยีสุรนารี

ROBOT TEACHING LABORATORY DEVELOPMENT FOR EFFECTIVE REMOTE DELIVERY

[1]SITTINON JULNITHI, [2]APHILAK LONKLANG, [3]JITTIMA VARAGUL, [4]KONTORN CHAMNIPRASART

[1,2,3,4]School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology, Thailand
E-mail:[1]sittinon.julniti@gmail.com,[2]aphilak@sut.ac.th, [3]kontorn@sut.ac.th

Abstract - Teaching Laboratories are the essential learning activities for the engineering education field. During the laboratory period, there are several learning activities, including theoretical review, equipment setup, data or result collecting, and analysis of the results. However, handling laboratory with social distancing during the pandemic of COVID-19 is hard to deal with onsite laboratory. Focusing on the industrial robot laboratories in the mechatronics engineering curriculum of the Suranaree University of Technology, this paper covers a remote robot laboratory design state using a computer programming technique to achieve the industrial robot laboratory and hardware design for robot programming tasks. There are three keys to practicing robot control, jogging, teaching, and programming. Students can remote access the university laboratory from distant areas, and they can book any time during the day and night shift. This paper can help educators deal with robot laboratories or other related laboratories in the social distancing situation.

Keywords - Remote laboratory, Industrial Robot Laboratory, Distance Learning

I. INTRODUCTION

In the context of a Thailand pandemic due to COVID-19, educational institutions, including schools and universities, are advised to enhance their teaching and learning activities in the physical distancing procedure. During the first period of lockdown in Thailand, It is at the end of the semester. Students were in their homes already, so the curfew announcement was delivered. Traveling across provinces was against this law. Consequently, students are advised not to come to university and stay home for online lectures and laboratories. Most teachers had no concrete experience of distance learning and little or no skill in using the technological devices and appropriate software to provide this distance education [1]. There are types of problems of distance learning, and online learning must be concerned.

- Technical problems: difficult or impossible connection for teachers and students
- pedagogical issues: the difficulty of staying focused on learning for hours on-screen communication and mutual assistance hampered by distance, despite the private practice of social networks. It is well known that these daily private practices are not so easily transferred to education [2].

The lecture course remained almost unchanged and delivered in an online classroom, such as in the zoom application. On the other hand, the laboratory courses must be re-designed because it is hard to teach their laboratory lessons or laboratory skills in truly virtual classrooms. The traditional laboratory must be developed to serve engineering students. The process for transforming currently in-person experiential learning activities to those suitable for remote delivery depends on several factors such as the nature of the experiment (simulation heavy vs. hands-on heavy) and the available time and resources. Some educators may adapt hands-on experiments for remote delivery, while others may restructure the activity altogether by concentrating on learning objectives[3].

Focusing on one of the industrial robot laboratories, called the cartesian robot. There are three modes of robot control

- Jogging mode: to perform the robot any position in its workspace by using pulse command
- Teaching mode: after jogging a robot to the desired position. To save the current position of jogging mode, called teaching mode.
- Programming mode: the last process of robot control after the two previous modes. Moving a robot in a sequence of positions, the user needs to program using its programming language.

This research aims to design and develop computer programming to communicate between the controller of the robot and the personal computer of students. Students can access the robot laboratory anywhere they are, and anytime they want. IAI robots are selected to develop the remote laboratory to serve students, who enrolled the industrial robot class.

II. DESIGN OF SYSTEM OVERVIEW

2.1. System Design Concept

The key concept is to design the remote robot laboratory for the industrial robot laboratory class using the assumption that students who enrolled in this laboratory class have their laptops or personal

computers. This laboratory can access through the global network, which means that they can do a laboratory where ever they want. “WebLabs” type option, where the students can control and operatethe equipment remotely[4][5][6].

2.2. System Overview

There are four primary devices in this system, including a personal computer, raspberry pi as a server for communication between robot and computer, an IAI RCP2 robot controller to drive the robot, and a camera for real-time monitoring (Fig.1.). TCP/IP protocol is selected to communicate the controlling signal from a user to the central server. After a user commands the robot on the user interface, the controlling data will be sent to the server via TCP/IP. Then, the server will send the actual commands to the robot controller via serial PIO communication. And a robot will be moved by its controller. We use the camera to monitor the movement of a robot to show that a robot moves to the desired position by broadcasting a real-time video streaming to the web page.

The system layout as see in Fig.2. The main robot (IAI Robot) has the operation stroke of 150 mm and three of output on its. A monitoring camera is located on the top of this plate for catch-up every components to broadcast to Graphic User Interface (GUI) on PC.

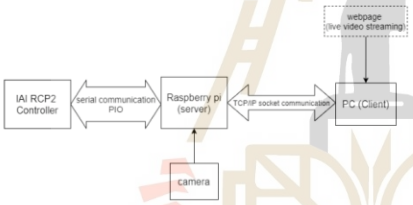


Fig.1 System Overview of Remote Robot Laboratory

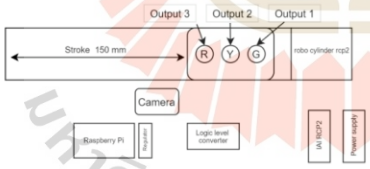


Fig.2 Equipment Layout of Remote Robot Laboratory

III. METHODOGY AND RESULT

3.1. IAI RCP2 Command

Command messages are sent from Raspberry-Pi to RCP2 controller via TCP/IP protocol. String of one message is combined with 16 long ASCII code. The start character is [STX] and end with [ETX]. For the validation of each message, there is the checksum of the last two characters of the continuous messages. The checksum character can be calculated by sum of the shole messages without the start bit character. Then end the message with [ETX] character. It can be seen the way to calculate the two checksum characters in Fig.3.

Status	STX	Axis#	'n'	10 continuous 0's										BCC	ETX
Inquiry	02		6E	30	30	30	30	30	30	30	30	30	30		03

BCC Data

Fig.3Message Detail for RCP2 Controller

3.2. ASCII Messages for RCP2 Controller

There are eight standard messages for using the IAI robot in the laboratory. The First two major commands are HOME and STOP. When the electrical power of the robot was connected, the first command that the student has to send to the controller immediately is a HOME command for homing it and let the robot stayed in its home position before use. Servo ON and Servo OFF is the critical command for active the servo motor of the IAI robot before JOG or movement by increment. Students can move the robot manually by using JOG Forward and JOG Backward functions. While the robot moves, students can monitor its movement via the onboard camera in the robot station. A monitoring camera can show the near real-time situation. The list of messages of the above command is in Table.1. below.

Command	ASCII Characters
HOME	[STX]0o07000000007B[ETX]
Servo On	[STX]0q10000000007E[ETX]
Servo Off	[STX]0q00000000007F[ETX]
Jog Forward	[STX]0aFFFF65430025[ETX]
Jog Backward	[STX]0aFFFFFFFF00DF[ETX]
Stop	[STX]0d00000000008C[ETX]
Position Inquiry	[STX]0R40000740008F[ETX]
Increment Move 80mm	[STX]0aFFFFD6550013[ETX]

Table.1. List of messages for RCP2 Controller Command.

3.3. Sequencing the Robot Command

If the command messages were used separately, there is an essential operation for using the IAI robot in the laboratory. Students have to command the robot in sequence to achieve the robot task. We create the computer programming command to deal with this IAI robot. The flowchart of this programming as can see in the Fig.3. After the program start, users or students have to write the sequence text of operation in one string with the following text.

“MX” stands for moving the robot into a specific position, X represents the position number.

“OX” stands for state changing of the output relay, X represents the output number to change the state ON or OFF.

“TX” stands for delay time for moving to the following command, X represents the time in seconds to delay.

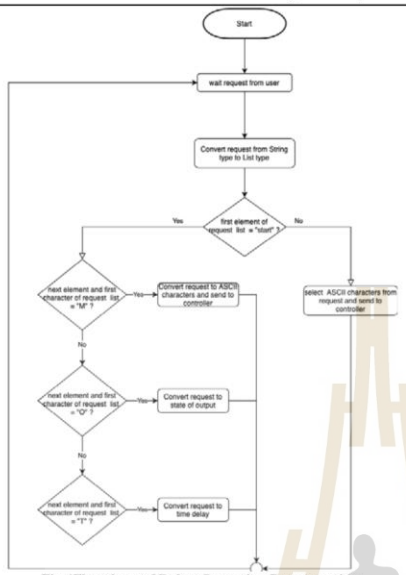


Fig.4Flowchart of Robot Operation Programming

The first exercise of the robot teaching laboratory is to teach the desired position the robot controller. By using the GUI for manually control and then check the "save position" block. Then the current position has been saved to the controller. The example of this operation as see in the Fig.5.

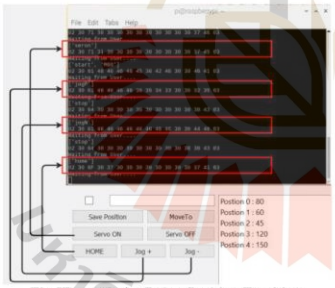


Fig.5Example for Robot Position Teaching

3.4 Laboratory Task

The students will be assigned by the teacher assistant a task. They have to teach and programming the IAI robot to achieve the task, which, combined with moving to some position, changes the state of outputs and waits for the following command.

For example in Fig.6, the series of command is "P0-O11-P4-T3-O10-P1-O31-P3-T3-O30". The steps of this command are following,

1. Move to position number "0"
2. Change state of output number "1" to active high (ON).
3. Move to position number "4"
4. Delay for 3 seconds
5. Change state of output number "1" to active low (OFF).
6. Move to position number "1"
7. Change state of output number "3" to active high (ON).
8. Move to position number "3"
9. Delay for 3 seconds
10. Change state of output number "3" to active low (OFF).

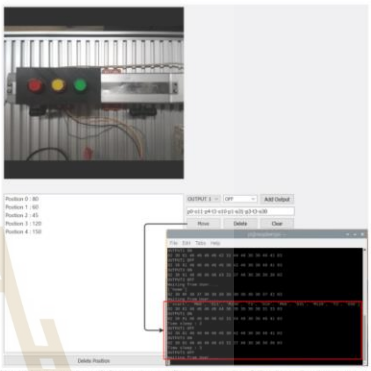


Fig.6 Example of Sequence Command of Robot Laboratory Task

IV. RESULTS

After this remote laboratory has been used for one semester, the students can appreciate the robot with online or distance learning methods. And they can access this laboratory at any time of learning, Because this laboratory open 24 hours a day. Student can book their slots on google sheet and using the user and password to enter them during their period.

As a result of teacher assistance, it is more comfortable to teach the laboratory with real robots in an online situation. Before this remote laboratory was launch, the teacher assistants have to do this robot task by themself without practicing with students

V. CONCLUSION

The results show that this remote laboratory can be used in an industrial robot laboratory for robot class in SUT. These can also help the teacher assistance can teach their students with the actual situation commands and tasks without social distancing or physical distancing problems.

Students can get more flexible of using university laboratory on the period they want. With this condition, they can book their learning period

online and log in to the remote learning page to do their task.

We have to extend this remote platform to every industrial robot laboratory for the near future, for example, Articulated Arm Robot, SCARA Robot, and Cartesian Robt. If the pandemic situation of COVID-19 still makes an impact on our world. The remote laboratory is the key to the success of engineering laboratory in teaching and learning methods.

REFERENCES

[1] Marie Debacq, Giana Almeida,Kevin Lachin,Marie-Laure Lameloise, Jeehyun Lee, Samantha Pagliaro, HediRomdhana, and Stéphanie Roux, "Delivering remote food engineering labs in COVID-19 time," Education for Chemical Engineers., vol. 34, no. 2, pp.9-20,2021.

[2] Laure Endrizzi, "Les technologies numériques dans l'enseignementsupérieur, entre défis et opportunités," Dossier d'actualitéVeille et Analyses n°78, Ifé, 2012.

[3] Vijesh J. Bhute, Pavan Inguva, Umang Shah, and Clemens Brechtelsbauer, "Transforming traditional teaching laboratories for effective remote delivery—A review," Education for Chemical Engineers., vol. 35, no. 14, pp.96-104,2021.

[4] Klein A., and Wozny G., "Web Based Remote Experiments for Chemical Engineering Education: The Online Distillation Column," Education for Chemical Engineers., vol. 1, no. 18, pp.134-138,2006.

[5] Marion Alliet-Gaubert, Galo Carrillo LeRoux, Charles D.F. de Jesus, Antonio J.G. Cruz, David Rouzineau, Roberto C. Giordano, Xavier Joulia, and Claudio A.O. Nascimento, "Cooperative WebLab in chemical engineering between France and Brazil: Validation of the methodology," Education for Chemical Engineers., vol. 7, no. 2, pp.7-13,2012.

[6] Philippe LEPROUX, Denis BARATAUD, Serge BAILLY, and Raphaël NIETO, "LABENVI, un dispositif pour les travaux pratiques à distance," Interfaces numériques., vol. 2, pp 453-467, 2013.

★★★

มหาวิทยาลัยเทคโนโลยีสุรนารี

ประวัติผู้เขียน

นายสิทธินนท์ จุลนิธิ เกิดเมื่อวันที่ 4 พฤษภาคม พ.ศ. 2537 ที่จังหวัดขอนแก่น จบการศึกษาระดับประถมศึกษาจากโรงเรียนเซนต์ยอแซฟกาฬสินธุ์ จังหวัดกาฬสินธุ์ จบการศึกษาระดับมัธยมศึกษาตอนต้นและตอนปลายจากโรงเรียนกาฬสินธุ์พิทยาสรรพ์ จังหวัดกาฬสินธุ์ และสำเร็จการศึกษาได้รับปริญญาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์) หลักสูตร 4 ปี ประจำปีการศึกษา 2558 มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2559 และหลังจากนั้นได้เข้ารับการศึกษาต่อในระดับวิศวกรรมศาสตรมหาบัณฑิต(วิศวกรรมเมคคาทรอนิกส์) ประจำปีการศึกษา 2559 ณ สถาบันเดิม

มหาวิทยาลัยเทคโนโลยีสุรนารี