

ระบบควบคุมและจำลองแขนหุ่นยนต์
ROBOTIC ARM CONTROL SYSTEM AND SIMULATION

กฤษฎี ดวงมณี
ชัยญาณุช วรรณวัชร
ตรัณ มารุ่งเรือง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ระบบควบคุมและจำลองแขนหุ่นยนต์
ROBOTIC ARM CONTROL SYSTEM AND SIMULATION

กฤษฎี	ดวงมณี
ชัยญาณุช	วรรตน์วัชร
ตรัณ	มารุ่งเรือง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

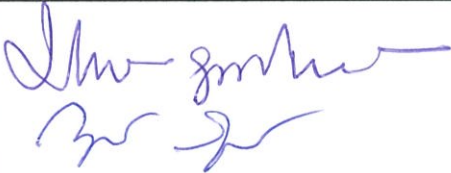
ROBOTIC ARM CONTROL SYSTEM AND SIMULATION

KRIT DUANGMANEE
CHANYANUT WARATWACHARA
TARUN MARUNGRUANG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKRABANG
ACADEMIC YEAR 2014

สาขาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ ระบบควบคุมและจำลองแขนหุ่นยนต์
ROBOTIC ARM CONTROL SYSTEM AND SIMULATION
นักศึกษาผู้จัดทำ นายกฤษฎี ดวงมณี รหัสนักศึกษา 54010061
นางสาวชญญาณุช วรรัตนวัชร รหัสนักศึกษา 54010303
นายตรีน มารุ่งเรือง รหัสนักศึกษา 54010481
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมอัตโนมัติ
ปีการศึกษา 2557

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.ประภาช อุดคภิมาพันธ์ ผศ.กฤษณ์ เสมอพิทักษ์	

หัวข้อปริญญาโท	ระบบควบคุมและจำลองแขนหุ่นยนต์ ROBOTIC ARM CONTROL SYSTEM AND SIMULATION		
นักศึกษาผู้จัดทำ	นายกฤษฎี	ดวงมณี	รหัสนักศึกษา 54010061
	นางสาวชญานุช	วรรัตน์วัชร	รหัสนักศึกษา 54010303
	นายตรัณ	มารุ่งเรือง	รหัสนักศึกษา 54010481
อาจารย์ที่ปรึกษา	รศ.ประภาช อุกคภิมาพันธุ์ ผศ.กฤษณ์ เสมอพิทักษ์		
ปีการศึกษา	2557		

บทคัดย่อ

ปริญญาโทฉบับนี้นำเสนอการออกแบบ และสร้างระบบควบคุมแขนหุ่นยนต์ที่สามารถเคลื่อนที่ตามเส้นทางการเคลื่อนที่ที่สร้างขึ้น โดยระบบควบคุมแขนหุ่นยนต์ที่พัฒนาขึ้นประกอบด้วย ส่วนควบคุมวงจรถับเคลื่อนมอเตอร์ในแต่ละข้อต่อของแขนหุ่นยนต์ที่มีสัญญาณป้อนกลับทำให้การเคลื่อนที่ของแขนหุ่นยนต์มีความแม่นยำมากขึ้น และส่วนตัวควบคุมการประมวลผลการเคลื่อนที่ที่สามารถควบคุมการทำงานผ่านคอมพิวเตอร์ โดยการนำเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์เข้าสมการเคลื่อนที่ของแขนหุ่นยนต์ที่พัฒนาขึ้นจากทฤษฎี Forward และ Inverse Kinematics เพื่อหาขนาดของมุมในแต่ละข้อต่อที่เคลื่อนที่ในทิศทางตามเส้นทางการที่สร้างขึ้น และปริญญาโทฉบับนี้ยังนำเสนอถึงวิธีการออกแบบเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ ที่ซับซ้อนด้วยโปรแกรมในคอมพิวเตอร์

Thesis Title	ROBOTIC ARM CONTROL SYSTEM AND SIMULATION		
Authors	Mr.	Krit	Duangmanee
	Miss	Chanyanut	Waratwachara
	Mr.	Tarun	Marungruang
Thesis Advisor	Assoc. Prof. Prapart Ukakmaparn		
	Asst. Prof. Krit Smerpitak		
Year	2014		

ABSTRACT

The objectives of the thesis were to describe the design, development and testing of Robotic Arm Control System and Simulation and to present the complicated design of the path of motion designed by computer. In addition to the robotic arm control system and simulation, it consisted of two systems. The one was the motor control circuits in each of the joints of the robotic arm that had feedback to control any motions accurately. Another was the processing of motion controller controlled by computer that could be calculated the paths with the equations of motion developed from forward and inverse kinematics theory, in order to find angles of each joints when it moved through any coordinates of the route.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้ นักศึกษาชั้นปีที่ 4 ได้เสนอเป็นส่วนหนึ่งในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอัตโนมัติ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2557 ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์ และช่วยเหลือดูแลเอาใจใส่เป็นอย่างดีจากหลายฝ่าย ทางผู้จัดทำขอขอบพระคุณเป็นอย่างสูง โดยเฉพาะอย่างยิ่ง รศ.ประภาส อุคคิกิมพันธ์ และ ผศ.กฤษณ์ เสมอพิทักษ์ อาจารย์ประจำสาขาวิชาวิศวกรรมอัตโนมัติที่กรุณารับเป็นอาจารย์ที่ปรึกษาปริญญาานิพนธ์ ที่คอยให้คำปรึกษา คำแนะนำ ช่วยหาวิธีการ แก้ไขปัญหาอุปสรรคต่างๆที่เกิดขึ้น ไม่ว่าจะเป็นปัญหาทางด้านวิชาการ หรือปัญหาทางด้าน การดำเนินงาน อีกทั้งเอื้อเฟื้ออุปกรณ์ เครื่องมือต่างในการทำงาน รวมทั้งให้กำลังใจผู้จัดทำด้วยความเมตตา ตั้งแต่เริ่มทำปริญญาานิพนธ์ฉบับนี้จนสำเร็จ

ผู้จัดทำขอขอบพระคุณ ผศ. ดร.ธีรวัฒน์ เทพมณี อาจารย์ประจำสาขาวิชาวิศวกรรมอัตโนมัติ คุณเจตติยะ ศรีพิทักษ์ นักศึกษาปริญญาานิพนธ์แขนกล ปีการศึกษา 2545 คุณอาทิตย์ โชติวีระสถานนท์ อดีตพนักงานบริษัท เอบีบี คุณเจษฎา เจริญวัฒน์โยธิน จากบริษัทไทร์เนอรี่ อินสตรูเมนต์ จำกัด และคณะอาจารย์ประจำสาขาวิชาวิศวกรรมอัตโนมัติ ที่ถ่ายทอดความรู้ความเข้าใจเกี่ยวกับข้อมูลรายละเอียดต่างๆ ให้คำปรึกษาและแนวทางการแก้ไข ชี้แนะแนวทำให้ปริญญาานิพนธ์เสร็จสมบูรณ์มากยิ่งขึ้น

ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่ให้การสนับสนุนและให้กำลังใจ ซึ่งเป็นกำลังใจสำคัญแก่ผู้จัดทำปริญญาานิพนธ์ดำเนินไปสู่ความสำเร็จ

สุดท้ายนี้คณะผู้จัดทำขอขอบคุณตึกสาขาวิชาวิศวกรรมอัตโนมัติที่เป็นสถานที่ทำปริญญาานิพนธ์นี้ ตลอดจนบุคคลต่างๆ ที่ให้ความช่วยเหลือที่ผู้จัดทำไม่ได้กล่าวนามได้ทั้งหมดในที่นี้ รวมถึงเพื่อนๆรุ่นพี่ รุ่นน้องภายในสาขาวิชาวิศวกรรมอัตโนมัติ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยสนับสนุนช่วยเหลือ คอยให้กำลังใจที่ดีเสมอ ผู้จัดทำรู้สึกซาบซึ้งในความกรุณา และความปรารถนาดีของท่านเป็นอย่างยิ่ง

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูปภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริศยานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริศยานิพนธ์	1
1.3 ขอบเขตของปริศยานิพนธ์	2
1.4 รายละเอียดของปริศยานิพนธ์	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 งานทฤษฎีและหลักการที่เกี่ยวข้อง.....	4
2.1 กล่าวนำ.....	4
2.2 หุ่นยนต์สำหรับอุตสาหกรรม (Industrial Robot).....	4
2.2.1 ลักษณะของหุ่นยนต์ (Overview of Robot).....	5
2.2.2 Articulated Robots	8
2.2.3 ข้อดีของ Articulate Robot	8
2.2.4 พื้นที่การทำงาน Articulated Robot.....	8
2.3 Forward Kinematics: The Arm Equation	9
2.3.1 Dot and Cross Product.....	10
2.3.2 โคออร์ดิเนต (Coordinate Frames).....	11
2.3.3 การหมุน (Rotation)	13
2.3.4 พิกัดโฮโมจีเนียส (Homogeneous Coordinates).....	15
2.3.5 พิกัดแขน (Link Coordinate).....	17
2.3.6 สมการแขน (The Arm Equation).....	19

สารบัญ (ต่อ)

	หน้า
2.4 Inverse Kinematics: Solving the Arm Equation.....	21
2.4.1 คุณสมบัติทั่วไปของคำตอบ (General Properties of Solutions)	21
2.4.2 Outline of Inverse Kinematics	22
2.5 การควบคุมการเคลื่อนที่ (Motion Control)	24
2.6 การเคลื่อนที่.....	24
2.6.1 การเคลื่อนที่เชิงเส้น (Linear Motion).....	24
2.6.2 การเคลื่อนที่แบบหมุนรอบแกนตายตัว (Rotation Around a Fixed Axis).....	25
2.7 การควบคุมแบบ PID.....	26
2.7.1 แนวทางคำนวณของ PID.....	26
2.2.2 การควบคุม PID แกนเดียว	26
2.8 ระบบการควบคุมแขนหุ่นยนต์	27
2.8.1 หลักการในการควบคุม	27
2.8.2 โครงสร้างของระบบ.....	28
2.9 แทคโคเจเนอเรเตอร์ (Tacho Generator).....	28
2.10 เอนโคเดอร์ (Encoder).....	29
2.11 CompactRIO	31
2.11.1 LabVIEW.....	32
2.11.2 ตัวควบคุม Real-Time (Real-Time Controller)	33
2.11.3 FPGA (Field Programmable Gate Array).....	33
2.11.4 โมดูล I/O (I/O Modules).....	34
2.12 MATLAB	35
2.13 RobotStudio	36
บทที่ 3 วิธีการดำเนินงาน	37
3.1 กล่าวนำ.....	37
3.2 การศึกษาข้อมูลและค้นหาวิธีการสร้างสมการหา End - Effect.....	38
3.2.1 การศึกษาหาวิธีการสร้างสมการของแขนหุ่นยนต์.....	38

สารบัญ (ต่อ)

	หน้า
3.2.2 การสร้าง Forward Kinematics.....	39
3.2.3 การสร้าง Inverse Kinematics.....	40
3.3 ฮาร์ดแวร์	41
3.3.1 การศึกษาและปรับปรุงแขนหุ่นยนต์.....	41
3.3.2 ตัวควบคุมที่ใช้ในการปรับปรุงซอฟต์แวร์.....	51
3.4 การสร้างเส้นทางการเคลื่อนที่.....	60
3.4.1 การเลือกใช้โปรแกรมในการสร้างเส้นทางการเคลื่อนที่	60
3.4.2 การนำภาษา RAPID ออกมาใช้งาน.....	60
3.5 ระบบคอมพิวเตอร์ซอฟต์แวร์	62
3.5.1 การพัฒนาสมการในโปรแกรม MATLAB.....	62
3.5.2 การสร้างซอฟต์แวร์ในโปรแกรม LabVIEW.....	65
บทที่ 4 ผลการทดลอง	78
4.1 กล่าวนำ.....	78
4.2 สมการที่ใช้ในการเคลื่อนที่ของแขนหุ่นยนต์	78
4.2.1 สมการ Forward Kinematics ของแขนหุ่นยนต์.....	78
4.2.1 สมการ Inverse Kinematics ของแขนหุ่นยนต์.....	80
4.3 พื้นที่ที่เหมาะสมในการใช้งานของแขนหุ่นยนต์.....	81
4.4 เส้นทางการเคลื่อนที่ที่ออกแบบในโปรแกรม RobotStudio	82
4.4.1 Module 1: เส้นทางการเคลื่อนที่แบบวงกลม.....	82
4.4.2 Module 2: เส้นทางการเคลื่อนที่แบบจำลอง Pick-and-Place	83
4.4.3 Module 3: เส้นทางการเคลื่อนที่แบบหลบหลีกสิ่งกีดขวาง	84
4.5 ส่วนการแสดงผลซอฟต์แวร์.....	85
4.5.1 ส่วนประมวลผลบน User Interface VI.....	85
4.5.2 ส่วนการประมวลผลบน Real-Time VI	80
4.6 การทดสอบซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์	89

สารบัญ (ต่อ)

	หน้า
4.6.1 การทดสอบเพื่อหาความถูกต้องของสมการการเคลื่อนที่ของแขนหุ่นยนต์.....	89
4.6.2 การทดลองการหาค่าจำนวนรอบพัลส์ต่อหนึ่งองศา	92
4.6.3 การทดลองหาอัตราการเปลี่ยนแปลงของข้อต่อเมื่ออีกข้อต่อมีการเปลี่ยนแปลง.....	95
3.6.4 การทดสอบซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์.....	97
3.6.5 การทดสอบความแม่นยำในการพิกัดจุด	101
บทที่ 5 สรุปผลการทดลอง.....	105
5.1 สรุปผลการทดลอง	105
5.2 ปัญหาและอุปสรรค.....	105
5.3 ข้อเสนอแนะ	106
เอกสารอ้างอิง	107
ภาคผนวก	108
1. วิธีใช้งานซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์	109
2. วิธีสร้างเส้นทางการเคลื่อนที่ในโปรแกรม RobotStudio	123

สารบัญตาราง

ตารางที่	หน้า
2.1 Kinematics Parameters.....	18
2.2 การถ่ายโอนโครงพิกัด k ไปยังโครงพิกัด $k-1$	19
3.1 ค่า Kinematics Parameters	39
3.2 รายการอุปกรณ์ที่ติดตั้งภายในแขนหุ่นยนต์	45
3.3 การเดินสายจากคอนเนกเตอร์ CN101 ไปอุปกรณ์ภายในแขนหุ่นยนต์	46
3.4 การเดินสายจากคอนเนกเตอร์ CN102 ไปอุปกรณ์ภายในแขนหุ่นยนต์	48
4.1 ค่าเริ่มต้นของเอนโคเดอร์และแขนหุ่นยนต์.....	88
4.2 ผลการทดสอบพิกัดตำแหน่งจาก RobotStudio และผลลัพธ์พิกัดจาก MATLAB.....	91
4.3 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการหมุนข้อต่อ R เมื่อมุมมองเสาเปลี่ยนไป 90 องศา	92
4.4 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการหมุนข้อต่อ L เมื่อมุมมองเสาเปลี่ยนไป 45 องศา	93
4.5 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการหมุนข้อต่อ U เมื่อมุมมองเสาเปลี่ยนไป 30 องศา.....	93
4.6 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการหมุนข้อต่อ θ เมื่อมุมมองเสาเปลี่ยนไป 360 องศา	94
4.7 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการหมุนข้อต่อ ϕ เมื่อมุมมองเสาเปลี่ยนไป 90 องศา.....	94
4.8 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการทดลองหมุนข้อต่อ L.....	95
4.9 ผลการทดลองค่าพัลส์ของ U ที่เกิดขึ้นขณะที่ θ มีค่า 30 องศา.....	96
4.10 ผลการทดสอบการเคลื่อนที่ของ Module 1.....	98
4.11 ผลการทดสอบการเคลื่อนที่ของ Module 2.....	99
4.12 ผลการทดสอบการเคลื่อนที่ของ Module 3.....	100
4.13 ผลการทดสอบการเคลื่อนที่ขนานแกน x ในระยะ y ต่างๆ.....	102
4.14 ผลการทดสอบการเคลื่อนที่ขนานแกน y ในระยะ x ต่างๆ.....	103
4.15 ผลการทดสอบการเคลื่อนที่คงที่แกน z ในระยะ y ต่างๆ.....	103

สารบัญภาพ

รูปที่	หน้า
2.1 Articulate Robot	4
2.2 Degree of Freedom.....	5
2.3 โครงพิกัดอ้างอิงฐาน	6
2.4 โครงพิกัดอ้างอิงข้อต่อ.....	6
2.5 โครงพิกัดอ้างอิง Tool	6
2.6 พื้นที่ความสามารถการหมุนที่เป็นไปได้ของหุ่นยนต์	9
2.7 หุ่นยนต์ที่มีลักษณะเป็นแขน	9
2.8 โครงพิกัดฉากใน R^3	10
2.9 พิกัดของ p เทียบกับโครงพิกัด X.....	12
2.10 การหมุนพื้นฐานใน R^3	13
2.11 การหมุนของ M รอบ f^1 ด้วยมุม ϕ	14
2.12 มุมข้อต่อ θ และระยะข้อต่อ d.....	17
2.13 ความยาวแขน (Link Length) a_k และมุมบิดแขน (Link Twist Angle) α	18
2.14 ตำแหน่งและการหมุนของ Tool ในพิกัดฐาน.....	20
2.15 Forward (Direct) และ Inverse Kinematics	21
2.16 คำตอบสามารถทำให้รูปร่างแขนเป็นได้ทั้ง Elbow up และ Elbow down	22
2.17 การแก้สมการมุมข้อต่อของแขนหุ่นยนต์สอง Links	23
2.18 ตัวควบคุม PID ของแกนเดียว.....	27
2.19 บล็อกไดอะแกรมโดยรวมของระบบควบคุมการทำงานของแขนหุ่นยนต์	27
2.20 แทคโคเจเนอเรเตอร์	28
2.21 หลังการทำงานของแทคโคเจเนอเรเตอร์	29
2.22 ส่วนประกอบของเอนโคเดอร์.....	29
2.23 หลักการทำงานของเอนโคเดอร์.....	30
2.24 สัญญาณที่ออกจากเอนโคเดอร์	30
2.25 สถาปัตยกรรมของระบบฝังตัวที่ปรับแต่งได้	31
2.26 โปรแกรม LabVIEW	32

สารบัญภาพ (ต่อ)

รูปที่	หน้า
2.27 ตัวควบคุม Real-time NI cRIO-9082.....	33
2.28 Reconfigurable FPGA Chassis.....	34
2.29 โมดูล I/O C Serises	34
2.30 โปรแกรม MATLAB.....	35
2.31 โปรแกรม RobotStudio.....	36
3.1 ระบบการทำงานของการควบคุมแขนหุ่นยนต์.....	37
3.2 แบบจำลองแขนหุ่นยนต์ 6 แกน Almega LK.....	38
3.3 การตั้ง Link Coordinates ของแขนหุ่นยนต์.....	39
3.4 มุมที่เกิดขึ้นจากแขนหุ่นยนต์.....	40
3.5 แขนหุ่นยนต์รุ่น Almega LK.....	41
3.6 ตำแหน่งข้อต่อและการหมุนของแขนหุ่นยนต์ในแบบสองมิติ	42
3.7 พื้นที่ความสามารถของแขนหุ่นยนต์มุมมองด้านข้าง.....	43
3.8 พื้นที่ความสามารถของแขนหุ่นยนต์มุมมองด้านบน	43
3.9 ส่วนประกอบต่างๆ ของแขนหุ่นยนต์มุมมองด้านซ้าย.....	44
3.10 ส่วนประกอบต่างๆ ของแขนหุ่นยนต์มุมมองด้านขวา.....	45
3.11 Wiring Diagram ของคอนเนกเตอร์ CN101	47
3.12 Wiring Diagram ของคอนเนกเตอร์ CN102.....	49
3.13 วงจรไฟฟ้ากำลัง ภายในตู้ควบคุม.....	51
3.14 ตู้ควบคุม OSACOM 5000	52
3.15 ส่วนประกอบต่างๆภายในตู้ควบคุมที่ใช้งาน	52
3.16 การเชื่อมต่อสาย PLC ด้าน Input	53
3.17 การเชื่อมต่อสาย PLC ด้าน Output.....	54
3.18 ภาพการใช้ PLC เชื่อมต่อกับแผงวงจรรีเลย์	55
3.19 แผงวงจรขับเคลื่อนมอเตอร์	55
3.20 ชุดวงจรขับเคลื่อน Accelus รุ่น asp-090-36	56
3.21 โมดูล NI ที่ใช้ใน cRIO	56

สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.22 โมดูล NI 9401 (Encoder Group1)	57
3.23 โมดูล NI 9401 (Encoder Group2)	57
3.24 โมดูล NI 9269 (CMD Group1).....	58
3.25 โมดูล NI 9269 (CMD Group2).....	58
3.26 โมดูล NI 9423	59
3.27 โมดูล NI 9474	59
3.28 ภาพการเชื่อมต่อ cRIO กับโมดูล NI	60
3.29 รูปแบบพื้นฐานของภาษา RAPID.....	61
3.30 ข้อมูลภาษา RAPID.....	61
3.31 ข้อมูลพิกัดบน Microsoft Excel.....	62
3.32 โปรแกรมหลังใน MATLAB	63
3.33 ฟังก์ชันที่ใช้ใน MATLAB (1).....	64
3.34 ฟังก์ชันที่ใช้ใน MATLAB (2).....	64
3.35 โครงสร้างการประมวลผลของ LabVIEW.....	65
3.36 แผนผังการทำงานของซอฟต์แวร์ (1)	66
3.37 แผนผังการทำงานของซอฟต์แวร์ (2).....	67
3.38 แผนผังการทำงานของซอฟต์แวร์ (3).....	68
3.39 แผนผังการทำงานของซอฟต์แวร์ (4)	69
3.40 ส่วนซอฟต์แวร์ในการสร้างกราฟเส้นทางการเคลื่อนที่.....	70
3.41 ส่วนซอฟต์แวร์ที่พัฒนาจากโปรแกรม MATLAB	71
3.42 ส่วนการเลือกเปิดไฟล์รูปแบบการเคลื่อนที่จาก RobotStudio.....	72
3.43 ส่วนการนำข้อมูลเข้าสมาการ Kinematics และฟังก์ชันการเคลื่อนที่.....	72
3.44 ส่วนจัดชุดคำสั่งของการเคลื่อนที่	73
3.45 สมการจาก MATLAB ที่นำมาใช้ใน LabVIEW.....	73
3.46 ส่วนการเคลื่อนที่แบบ Manual	74
3.47 ส่วนการเคลื่อนที่แบบ Auto	74

สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.48 ส่วนการเปิดการทำงาน FPGA ใน Real-Time	75
3.49 ส่วนควบคุมการทำงานของเอนโคเดอร์	75
3.50 ส่วนการสร้างสัญญาณพัลส์	76
3.51 ส่วนการจำลองการเคลื่อนที่	76
3.52 ส่วนกำหนดตำแหน่งการเคลื่อนที่	77
3.53 ส่วนควบคุมขนาดสัญญาณให้แก่วงจรขับเคลื่อน	77
4.1 Link Coordinates ของแขนหุ่นยนต์	78
4.2 มุมที่เกิดขึ้นจากแขนหุ่นยนต์	80
4.3 พื้นที่ที่เหมาะสมในการใช้งานของแขนหุ่นยนต์	81
4.4 แบบเส้นทางการเคลื่อนที่ของ Module 1	82
4.5 แบบเส้นทางการเคลื่อนที่ของ Module 1 บน RobotStudio	82
4.6 การวางตำแหน่งวัตถุของ Module 2	83
4.7 แบบเส้นทางการเคลื่อนที่ของ Module 2 บน RobotStudio	83
4.8 การวางตำแหน่งวัตถุกีดขวางของ Module 3	84
4.9 แบบเส้นทางการเคลื่อนที่ของ Module 3 บน RobotStudio	84
4.10 ส่วนหน้าจอในการแสดงการจำลองเส้นทางการเคลื่อนที่ Module ต่างๆ	85
4.11 ส่วนหน้าจอในการแสดงเส้นทางการเคลื่อนที่ Module 1	86
4.12 ส่วนหน้าจอในการแสดงเส้นทางการเคลื่อนที่ Module 2	86
4.13 ส่วนหน้าจอในการแสดงเส้นทางการเคลื่อนที่ Module 3	87
4.14 ส่วนแสดงผลหลักในการใช้งานซอฟต์แวร์	87
4.15 ส่วนการตั้งค่าเริ่มต้นให้กับเอนโคเดอร์	88
4.16 ชุดคำสั่งในการเคลื่อนที่ของแต่ละข้อต่อใน Auto Mode	89
4.17 ผลลัพธ์จาก RobotStudio (ด้านซ้าย) และบนโปรแกรม MATLAB (ด้านขวา) ของ Module 1	90
4.18 ผลลัพธ์ด้านบนจาก RobotStudio และบนโปรแกรม MATLAB ของ Module 1	90

สารบัญภาพ (ต่อ)

รูปที่	หน้า
4.19 ผลลัพธ์จาก RobotStudio (ด้านซ้าย) และบนโปรแกรม MATLAB (ด้านขวา) ของ Module 2.....	90
4.20 ผลลัพธ์ด้านบนจาก RobotStudio และบนโปรแกรม MATLAB ของ Module 2.....	90
4.21 ผลลัพธ์จาก RobotStudio (ด้านซ้าย) และบนโปรแกรม MATLAB (ด้านขวา) ของ Module 3.....	91
4.22 ผลลัพธ์ด้านบนจาก RobotStudio และบนโปรแกรม MATLAB ของ Module 3.....	91
4.23 ภาพการทดสอบการเคลื่อนที่ของ Module 1 (1).....	97
4.24 ภาพการทดสอบการเคลื่อนที่ของ Module 1 (2).....	98
4.25 ภาพการทดสอบการเคลื่อนที่ของ Module 2 (1).....	98
4.26 ภาพการทดสอบการเคลื่อนที่ของ Module 2 (2).....	99
4.27 ภาพการทดสอบการเคลื่อนที่ของ Module 3 (1).....	100
4.28 ภาพการทดสอบการเคลื่อนที่ของ Module 3 (2).....	100
4.29 ภาพการทดสอบการเคลื่อนที่ขนานแกน x.....	101
4.30 ภาพการทดสอบการเคลื่อนที่ขนานแกน y.....	102

บทที่ 1

บทนำ

1.1. ความสำคัญของปริญญานิพนธ์

เป้าหมายหลักในกระบวนการผลิตของโรงงานอุตสาหกรรม คือ การผลิตสินค้าที่มีต้นทุนการผลิตต่ำได้อย่างตรงเวลาและมีคุณภาพสม่ำเสมอ การจะได้ผลลัพธ์เหล่านี้มาจำเป็นต้องมีกระบวนการผลิตที่เสถียร ถูกต้อง และแม่นยำ ซึ่งมนุษย์ไม่สามารถตอบสนองความต้องการทั้งหมดนี้ได้ เนื่องจากบางขั้นตอนของการผลิตเป็นงานที่อาจก่อให้เกิดอันตรายร้ายแรง หรือแรงงานคนยังไม่สามารถทำให้กระบวนการผลิตมีความเสถียรหรือถูกต้องแม่นยำอย่างเพียงพอ เป็นต้น ทำให้เกิดการพัฒนาเครื่องมือหรืออุปกรณ์ต่างๆ เข้ามาใช้ตอบโจทย์ของกระบวนการผลิตที่มีประสิทธิภาพ ดังนั้นแขนหุ่นยนต์ (Robot Arm) จึงเป็นอุปกรณ์ที่ได้รับการพัฒนาและนำมาใช้เพื่อตอบสนองความต้องการในงานอุตสาหกรรมดังกล่าว แขนหุ่นยนต์เหมาะสมกับงานประเภทที่ทำต่อเนื่องเป็นเวลานาน เป็นงานที่ทำซ้ำๆ กันตลอดเวลา งานที่เข้าถึงได้ยาก ทั้งยังตอบสนองต่อการทำงานที่มีความยืดหยุ่น การใช้แขนหุ่นยนต์ทำให้ประสิทธิภาพของการทำงานดีขึ้น มีความแน่นอนของงานที่เกิดขึ้น ความแม่นยำเพิ่มขึ้น ทำงานได้รวดเร็วขึ้น สามารถผลิตสินค้าได้มากขึ้น และมีคุณภาพที่สม่ำเสมอ

ด้วยเหตุนี้ ผู้จัดทำจึงได้จัดทำปริญญานิพนธ์ฉบับนี้ขึ้น โดยการเน้นด้านออกแบบและพัฒนา “ซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์” ขึ้น โดยซอฟต์แวร์ดังกล่าวมีจุดเด่นดังนี้ ประการแรก เป็นซอฟต์แวร์ระบบที่ใช้ควบคุมและกำหนดทิศทางการเคลื่อนที่ให้กับแขนหุ่นยนต์ เพื่อใช้ให้ประโยชน์ในงานด้านอุตสาหกรรมการผลิต อันได้แก่ การเคลื่อนย้ายสินค้า หรือวัสดุต่างๆ ที่อาจก่อให้เกิดอันตรายต่อมนุษย์ ประการที่สอง ซอฟต์แวร์ได้รับการออกแบบให้มีการจำลองการเคลื่อนที่ของแขนหุ่นยนต์ก่อนมีการเคลื่อนที่จริง เพื่อป้องกันความผิดพลาดในการเคลื่อนที่ของแขนหุ่นยนต์ และประการสุดท้าย โปรแกรมได้รับการออกแบบให้ใช้งานได้ง่าย ไม่ซับซ้อน และมีความปลอดภัย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการนำแขนหุ่นยนต์ที่ใช้ในอุตสาหกรรมที่มีความยืดหยุ่นและการทำงานที่เกินขีดจำกัดของมนุษย์
2. ใช้ซอฟต์แวร์ในการคำนวณหาการเคลื่อนที่ของแขนหุ่นยนต์ เพื่อให้แขนหุ่นยนต์เคลื่อนที่ไปตามที่กำหนดได้อย่างถูกต้อง
3. ใช้โปรแกรมจำลองการเคลื่อนที่ของแขนหุ่นยนต์เพื่อการออกแบบ การทดสอบ และการดูความปลอดภัยในการเคลื่อนที่ของแขนหุ่นยนต์
4. เพื่อออกแบบระบบการควบคุมการทำงานของมอเตอร์เพื่อรับคำสั่งการเคลื่อนที่จากซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์

1.3 ขอบเขตของปริญญาานิพนธ์

ซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ที่ออกแบบและสร้างขึ้นมีความสามารถในการทำงานดังต่อไปนี้

1. สามารถสร้างและออกแบบเส้นทางการเคลื่อนที่และจำลองเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ได้จากการสร้างผ่านกราฟิกบนโปรแกรม RobotStudio
2. สามารถแปลงข้อมูลจากการสร้างเส้นทางบนโปรแกรม RobotStudio เป็นข้อมูลที่ต้องการได้ในซอฟต์แวร์การเคลื่อนที่ได้บนโปรแกรม MATLAB
3. สามารถจำลองการเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ได้ในซอฟต์แวร์บนโปรแกรม MATLAB เพื่อตรวจสอบความถูกต้องของการเคลื่อนที่ของแขนหุ่นยนต์
4. สามารถควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ให้มีความถูกต้องตามการออกแบบเส้นทางการเคลื่อนที่ผ่านซอฟต์แวร์การควบคุมการเคลื่อนที่
5. แขนหุ่นยนต์สามารถเคลื่อนที่ได้ตรงตามการออกแบบเส้นทางการเคลื่อนที่โดยผ่านการใช้ซอฟต์แวร์ระบบควบคุมการเคลื่อนที่ทั้งหมด
6. มีการป้องกันอันตรายที่อาจเกิดระหว่างการใช้งานซอฟต์แวร์ระบบควบคุมการเคลื่อนที่ระหว่างการเคลื่อนที่ของแขนหุ่นยนต์ รวมทั้งอันตรายกับแขนหุ่นยนต์เอง

1.4 รายละเอียดของปริญญาานิพนธ์

ในปริญญาานิพนธ์นี้มีเนื้อหา 5 บท โดยแต่ละบทมีเนื้อหาดังนี้

บทที่ 1 บทนำ เป็นการกล่าวถึง ความสำคัญและที่มาของปริญญาานิพนธ์ วัตถุประสงค์ของปริญญาานิพนธ์ รายละเอียดของปริญญาานิพนธ์ แผนการดำเนินงานที่วางไว้ และผลที่คาดว่าจะได้รับจากปริญญาานิพนธ์

บทที่ 2 งานทฤษฎีและหลักการที่เกี่ยวข้อง หุ่นยนต์ในอุตสาหกรรม กล่าวถึงพื้นฐานของความรู้เกี่ยวกับหุ่นยนต์ และประเภทของแขนหุ่นยนต์ที่ใช้ในงานปริญญาานิพนธ์ ทฤษฎีสมการแขนหุ่นยนต์ Forward และ Inverse Kinematics ระบบการควบคุมแขนหุ่นยนต์ หลักการของแอกโคเจเนอเรเตอร์ หลักการของเอนโคเดอร์ ฮาร์ดแวร์ของตัวควบคุม รวมทั้งโปรแกรมที่ใช้ในการออกแบบซอฟต์แวร์ให้กับแขนหุ่นยนต์

บทที่ 3 วิธีการดำเนินงาน กล่าวถึงขั้นตอนต่างที่จะมาเป็นซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ตั้งแต่การตรวจสอบข้อต่อต่างๆ ของแขนหุ่นยนต์ อุปกรณ์ที่ติดตั้ง การเดินสายไฟทั้งภายในแขนหุ่นยนต์ ภายในตู้ควบคุม และการปรับปรุงส่วนที่เสียหายของตู้ควบคุม รวมทั้งวิธีการทดสอบอุปกรณ์ สมการต่างๆ ที่นำมาใช้ในการสร้างซอฟต์แวร์ระบบการเคลื่อนที่ของแขนหุ่นยนต์

บทที่ 4 ผลการทดสอบ เป็นผลการทดสอบอุปกรณ์เพื่อตรวจสอบอุปกรณ์ที่มีอยู่ยังสามารถใช้งานได้ ตรวจสอบผลลัพธ์จากสมการที่ออกแบบขึ้นภายในโปรแกรมต่างๆ และตรวจสอบการทำงานของซอฟต์แวร์ที่ออกแบบขึ้นใช้งานร่วมกับแขนหุ่นยนต์ ด้วยการทดสอบการเคลื่อนที่ของแขนหุ่นยนต์

บทที่ 5 สรุปผลการทดลอง เป็นการสรุปผลการดำเนินงานทั้งหมดเมื่อนำระบบการควบคุมและจำลองการเคลื่อนที่ของแขนหุ่นยนต์มาใช้งานในการดำเนินงานจริง ปัญหาที่พบอุปสรรค และข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

จากการออกแบบและพัฒนาซอฟต์แวร์ระบบดังกล่าวนี้ ผลที่คาดว่าจะได้รับ คือ ซอฟต์แวร์ระบบควบคุมและการจำลองเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ เส้นทางการเคลื่อนที่จากการสร้างเส้นทางในกราฟิกบนโปรแกรม RobotStudio สามารถแปลงข้อมูลเส้นทางการเคลื่อนที่ในโปรแกรม RobotStudio เป็นข้อมูลที่ต้องการใช้ในระบบการเคลื่อนที่ได้ และจำลองในซอฟต์แวร์บนโปรแกรม Matlab เพื่อเป็นการตรวจสอบความถูกต้องของงานที่สร้างขึ้น และทำให้แขนหุ่นยนต์สามารถเคลื่อนที่ได้ตรง และถูกต้องตามการออกแบบเส้นทางการเคลื่อนที่โดยผ่านการใช้ซอฟต์แวร์ระบบควบคุมการเคลื่อนที่ทั้งหมด ซึ่งมีการป้องกันอันตรายที่อาจเกิดระหว่างการใช้งานซอฟต์แวร์ระบบควบคุมการเคลื่อนที่ และระหว่างการเคลื่อนที่ของแขนหุ่นยนต์ รวมทั้งอันตรายกับแขนหุ่นยนต์

บทที่ 2

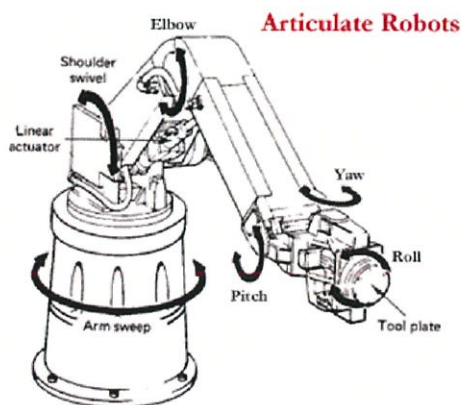
งานทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 กล่าวนำ

หุ่นยนต์ (Robot) เป็นอุปกรณ์ทางกลที่ประกอบด้วยแขน (Link) และข้อต่อ (Joint) นำทางโดยเซ็นเซอร์ (Sensor) ขับเคลื่อนโดยอุปกรณ์ควบคุมสุดท้าย (Actuator) และควบคุมผ่านการเขียนซอฟต์แวร์การจับ และเส้นทางของแขนหุ่นยนต์ วัสดุ เครื่องมือ และอุปกรณ์สำหรับงานต่างๆในสภาพแวดล้อมที่แตกต่างกัน ดังนั้นการออกแบบระบบควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ จะต้องตอบโจทยความต้องการของงานว่า สิ่งที่ต้องการนั้นคืออะไร ต้องการชนิดของหุ่นยนต์ประเภทไหน ต้องการการเคลื่อนที่เป็นเช่นไร และใช้ซอฟต์แวร์อะไรในการสร้างฟังก์ชัน เป็นต้น โดยในบทนี้จะกล่าวถึงทฤษฎี และข้อมูลต่างๆ ที่นำมาใช้ในการออกแบบแขนหุ่นยนต์ ประกอบด้วยส่วนต่างๆต่อไปนี้

2.2 หุ่นยนต์สำหรับอุตสาหกรรม (Industrial Robot) [1]

หุ่นยนต์สำหรับอุตสาหกรรม (Industrial Robot) ถูกวางรูปแบบให้มีลักษณะคล้ายคลึงแขนทางกายภาพของมนุษย์ โดยมีท่อนแขนจากหัวไหล่ถึงข้อศอก (Upper Arm) ท่อนแขนจากข้อศอกกับข้อมือ (Fore arm) และนิ้ว (Fingers) จนสุดแขน ส่วนอก (Chest) และแขนถูกติดเข้ากับฐาน (Base) ของตัวหุ่นยนต์ ส่วนข้อต่อต่างๆของหุ่นยนต์จะมีชื่อเรียกว่า ไหล่ (Shoulder) ข้อศอก (Elbow) และข้อมือ (Wrist) ส่วนนิ้วของหุ่นยนต์จะเรียกว่า End - Effect ซึ่งอาจเป็น Tool หรือตัวจับ (Gripper) ที่ติดอยู่กับหุ่นยนต์ การเคลื่อนไหวของหุ่นยนต์ขึ้นอยู่กับลักษณะทางโครงสร้างของการเขียนโปรแกรมผ่านคอมพิวเตอร์ โดยหุ่นยนต์นี้จะรู้จักกันในนาม “Anthropomorphic หรือ Articulated Robots”



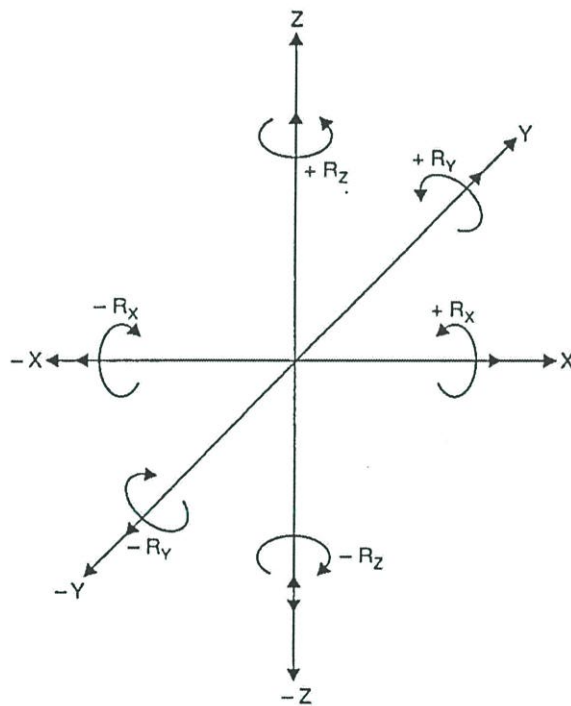
รูปที่ 2.1 Articulate Robot

2.2.1 ลักษณะของหุ่นยนต์ (Overview Of Robots)

ความเข้าใจสำคัญที่เป็นพื้นฐานเกี่ยวกับการนำหุ่นยนต์มาใช้ในการงานอุตสาหกรรม รายละเอียดข้อมูลมีดังต่อไปนี้

2.2.1.1 Degree of Freedom

การเคลื่อนที่ของแขนออกไปในพื้นที่ (Space) เป็นอิสระได้ทั้งหมด 6 ทิศทาง ทั้งทางบวกและทางลบ โดย 6 ทิศทางการหมุนเกี่ยวกับพิสัยคาคิเขียนเป็นตามเข็มนาฬิกา 3 แกน และทวนเข็มนาฬิกาอีก 3 แกน ตามที่แสดงในรูปที่ 2.2

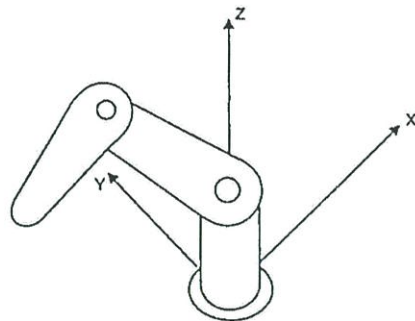


รูปที่ 2.2 Degree of Freedom

2.2.1.2 โครงพิสัยอ้างอิง (The Reference Frames)

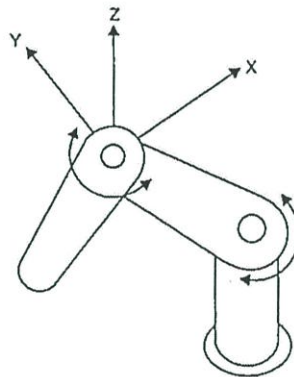
การอ้างอิงมี 3 ประเภท โดยพิจารณาจากโครงสร้างหุ่นยนต์

1. โครงพิสัยอ้างอิงฐาน (Base Reference Frame) เป็นแกนพื้นฐาน x, y, z ซึ่งเป็นแกนของฐาน อาจจะกำหนดไว้หรือหมุนรอบแกน z ตามความต้องการของงาน พิกัดอ้างอิงของฐานนี้เป็นระบบสากลของการอ้างอิงของหุ่นยนต์



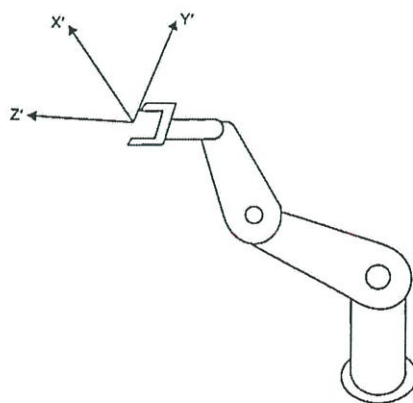
รูปที่ 2.3 โครงพิกัดอ้างอิงฐาน

2. โครงพิกัดอ้างอิงข้อต่อ (Joint Reference Frame) เป็นแกนอ้างอิงที่ชี้จุดข้อต่อของหุ่นยนต์ โดยข้อต่อจะมีการเปลี่ยนตำแหน่งและการหมุนของแกนที่กำหนด



รูปที่ 2.4 โครงพิกัดอ้างอิงข้อต่อ

3. โครงพิกัดอ้างอิง Tool (Tool Reference Frame) เป็นจุดที่อ้างอิงโดยกำหนดที่จุดปลายสุดแขน หรือมือของหุ่นยนต์ โดยจุดปลายหรือโครงพิกัดอ้างอิงนี้จะมีความสัมพันธ์กับโครงพิกัดอ้างอิงของฐานโดยการถ่ายโอนพิกัด (Coordinates Transformation)



รูปที่ 2.5 โครงพิกัดอ้างอิง Tool

2.2.1.3 ข้อต่อหุ่นยนต์ (Robot Joints)

ข้อต่อแขนหุ่นยนต์ทำขึ้นเพื่อสร้างการเคลื่อนไหว อาจเป็นเส้นตรง หมุน หรือเป็นทรงต่างๆ ข้อต่อที่การเคลื่อนที่เป็นเส้นตรงจะเรียกว่า ข้อต่อแบบ Prismatic ส่วนข้อต่อสร้างการเคลื่อนที่แบบหมุนจะเรียกว่า ข้อต่อแบบ Revolute และข้อต่อแบบการหมุนแบบต่างๆจะเรียกว่า ข้อต่อแบบ Spherical

การเคลื่อนไหวของข้อต่อสามารถใช้ในการขับเคลื่อน โดยใช้ระบบไฮดรอลิก (Hydraulic) ระบบนิวแมติก (Pneumatic) หรือระบบไฟฟ้า (Electric Drive) โดยระบบขับเคลื่อนไฮดรอลิกและนิวแมติกสามารถใช้สำหรับการยกวัตถุที่มีน้ำหนักมาก และสำหรับวัตถุน้ำหนักเบาจะใช้ระบบขับเคลื่อนไฟฟ้าก็เพียงพอ ส่วนมากแขนหุ่นยนต์ที่ใช้ในงานอุตสาหกรรมนิยมใช้ข้อต่อแบบ Prismatic หรือแบบ Revolute เพราะสามารถใช้งานง่ายและการควบคุมการขับเคลื่อนเป็นแบบปกติ

2.2.1.4 ส่วนประกอบของหุ่นยนต์ (Robot Component)

สำหรับหุ่นยนต์ทางอุตสาหกรรมเพื่อให้ถูกต้องตรงกับความสามารถ จึงต้องทราบถึงส่วนต่างอย่างแน่นอน ดังต่อไปนี้

1. แขน (Arms) คือ หลายๆแขนและหลายๆข้อต่อนำมารวมกันเป็นร่างของหุ่นยนต์จะถูกเรียกเป็นแมนิพูเลเตอร์ (Manipulator) ซึ่งคือเครื่องจักรกลของหุ่นยนต์
2. Tool หรือ End - Effectors คือ ส่วนที่เชื่อมต่อส่วนปลายของแขนที่เป็นฐานที่ตั้งของมือหุ่นยนต์ โดยมี Tool หัวต่างๆที่มีหน้าที่แตกต่างกันไปตามการใช้งาน อาทิเช่น ฟันสี หัวเชื่อมอินเล็คโทรด ตัวจับ หรือเครื่องมือตามการออกแบบต่าง เป็นต้น
3. อุปกรณ์ควบคุมสุดท้าย (Actuators) คือ ส่วนส่งกำลังไปยังส่วนข้อต่อของหุ่นยนต์ โดยจะสร้างการหมุนหรือการเปลี่ยนการเคลื่อนที่ของแขนต่างๆ ระบบส่งกำลังเป็นได้ทั้งระบบไฮดรอลิก หรือนิวแมติก และมอเตอร์ชนิดเซอร์โว (Servo-motor) หรือมอเตอร์ชนิดสเตปเปอร์ (Stepper Motor)
4. เซ็นเซอร์ (Sensors) คือ ส่วนจดจำตำแหน่งของแขนหุ่นยนต์ทั้งการเคลื่อนที่หรือหยุดนิ่ง โดยระยะการเคลื่อนไหวขึ้นอยู่กับการป้อนกลับของข้อมูลที่เก็บไว้โดยเซ็นเซอร์ เซ็นเซอร์นี้รวมถึงทำหน้าที่เก็บรวมข้อมูลรอบๆ (Work cell) ที่เป็นส่วนการช่วยในการดำเนินการ
5. ตัวควบคุม (Controllers) คือ อุปกรณ์อิเล็กทรอนิกส์ ที่สัญญาณจากเซ็นเซอร์ เป็นตัวจัดการให้ชุดขับเคลื่อนในเข้าใจรูปแบบการเคลื่อนที่ การสร้างการเคลื่อนที่อาจจะเข้ากันหรือไม่เข้ากันกับสิ่งที่ต้องการ ค่าความต่างจะถูกส่งกลับมาในรูปของค่าความผิดพลาด (Error) โดยจะปรับให้เข้ากับ output อ้างอิงส่งไปยังอุปกรณ์ควบคุมสุดท้าย ค่าของการควบคุมและค่าของการป้อนกลับจะเป็นค่าที่อยู่ในระบบควบคุม
6. ซอฟต์แวร์ และฮาร์ดแวร์ (Software and Hardware) คือ ซอฟต์แวร์คอมพิวเตอร์และอุปกรณ์ผ่านโค้ดการคำนวณสู่การขับเคลื่อนแปลงเป็นตำแหน่งที่ต้องการ รวมทั้ง

ความเร็ว และความถูกต้องของการขับเคลื่อนจาก Kinematics Equations โดยการขับเคลื่อนจะเชื่อมต่อกับระบบคอมพิวเตอร์ ภาษาโปรแกรมมีทั้งภาษาระดับล่างที่เหมือนภาษาเครื่องจักร หรือเป็นภาษาขั้นสูงที่นิยมใช้ในปัจจุบัน

2.2.2 Articulated Robots หรือ Anthropomorphic [2]

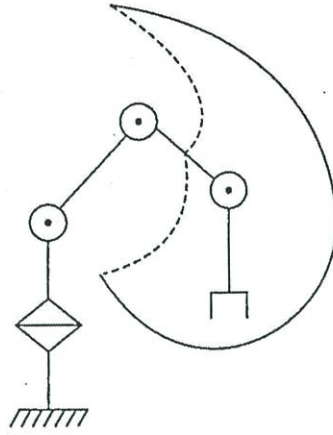
Articulated Robot หรือ Anthropomorphic Robot (3R) คือ หุ่นยนต์ประกอบด้วยข้อต่อเป็นแกนที่มีการเคลื่อนที่เป็นแบบหมุนได้ (Rotary Joint) มีความสามารถหมุนได้หลากหลายไปตามระนาบต่างๆ ซึ่งหุ่นยนต์ประเภทนี้ช่วยเพิ่มความสามารถของหุ่นยนต์ให้เพิ่มสูงขึ้น โดยอาจมีข้อต่อแกนสำหรับการเคลื่อนที่เพียงหนึ่งข้อต่อจนไปถึงหลายๆ ข้อต่อ โดยแต่ละส่วนจะมีความสามารถที่ใช้งานขึ้นอยู่กับกรอบและฟังก์ชันที่ต้องการ หุ่นยนต์ที่มีข้อต่อแกนแบบหมุนนี้นิยมใช้ในงานที่มีความแม่นยำสูงในการเคลื่อนที่ จะพบแขนหุ่นยนต์ประเภทนี้ได้ ในอุตสาหกรรมสายการผลิต โดยจะใช้ประโยชน์จากหุ่นยนต์นี้ในเรื่องความยืดหยุ่นสูงของข้อต่อแกนของหุ่นยนต์ ที่สามารถโค้งงอ และเคลื่อนที่ได้หลายทิศทาง ข้อต่อแกนหลายๆ แกนจะใช้การควบคุมหรือการดำเนินการหลายๆ ส่วนได้ในพร้อมกันเพียงครั้งเดียว ในตัวอย่างเช่น ข้อต่อแกนแบบหมุนที่ทำให้หุ่นยนต์เคลื่อนที่ไปกลับในพื้นระนาบงานที่แตกต่างกัน นักวิจัยที่พัฒนาหุ่นยนต์จะต้องพัฒนาหุ่นยนต์เมื่อต้องการนำ Articulated Robot นี้ มาใช้งานต่างๆ อาทิเช่น การสอนให้หุ่นยนต์เดิน และการพัฒนาการเคลื่อนไหว โดยข้อต่อต่างๆ ของหุ่นยนต์จะถูกโปรแกรมให้สามารถติดต่อกับข้อต่ออื่นๆ ได้ นอกจากนั้นสามารถใช้งานได้อย่างอิสระ มีระดับการใช้ในงานที่ยากขึ้น ในอนาคตหุ่นยนต์ในลักษณะนี้เพราะสามารถในงานในฟังก์ชันที่มีระดับความยากได้มาก

2.2.3 ข้อดีของ Articulated Robot [3]

1. ประสิทธิภาพสูงขึ้นจากเดิม
2. สามารถใช้การเคลื่อนที่เส้นตามทางที่ต่อเนื่อง ตัวอย่างเช่น การประยุกต์ใช้กับการพ่นสี หรือการเชื่อม
3. สามารถใช้กับการเคลื่อนที่ที่ละเอียดๆ โดยไม่เกิดความผิดพลาด

2.2.4 พื้นที่การทำงาน Articulated Robot

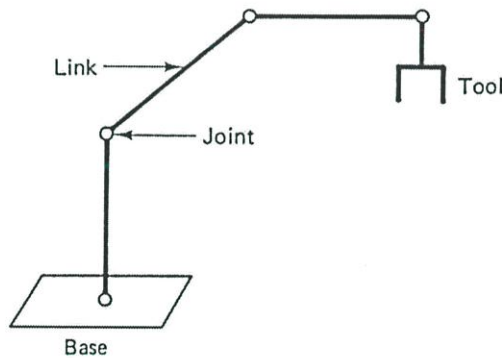
เนื่องจากเป็นแขนหุ่นยนต์แบบข้อต่อหมุนที่มีทั้งหมด 3 ข้อต่อที่สามารถหมุนได้ ซึ่งสามารถเคลื่อนที่หมุนได้ทั้งในแกน x , y , z ดังนั้นรูปร่างการเคลื่อนตามรูป ดังนี้



รูปที่ 2.6 พื้นที่ความสามารถการหมุนที่เป็นไปได้ของหุ่นยนต์

2.3 Forward Kinematics: The Arm Equation [2]

แขนหุ่นยนต์มีลักษณะเป็นวัตถุแข็งเกร็ง (Rigid Body) เป็นแขนหลายๆ อันนำมาเชื่อมต่อเข้าด้วยกันโดยใช้ข้อต่อ ดังรูปที่ 2.7 ปลายข้อหนึ่งของแขน ริมสุดจะติดแน่นกับฐานส่วนปลายของแขน อีกอันจะเคลื่อนที่ได้อย่างอิสระ ปลายข้างที่เคลื่อนที่นี้จะต่อเข้ากับ Tool



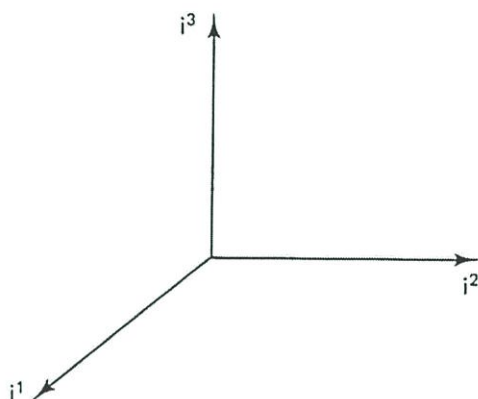
รูปที่ 2.7 หุ่นยนต์ที่มีลักษณะเป็นแขน

การควบคุมทั้งตำแหน่งและการหมุนของ Tool ในระนาบ 3 มิติ เพื่อให้สามารถเขียนโปรแกรมให้ Tool เคลื่อนที่ไปตามเส้นทางที่วางแผน และการจับชิ้นงานเคลื่อนที่ย้ายในพื้นที่ทำงาน ในการเขียนโปรแกรมเพื่อควบคุม Tool นี้ จำเป็นจะต้องคำนวณระหว่างตัวแปรข้อต่อต่างๆ (Joint Variables) ให้สัมพันธ์กับตำแหน่งและการหมุนของ Tool ซึ่งจะเรียกส่วนนี้ว่า ปัญหา Forward Kinematics โดยมีนิยามดังนี้

ปัญหาของ Forward Kinematics คือ การกำหนดเวกเตอร์ตัวแปรข้อต่อต่างๆ ของแขนหุ่นยนต์ แล้วหาพิกัดและการหมุนของ Tool เทียบกับโครงพิกัดที่ติดอยู่กับฐานของแขนหุ่นยนต์

2.3.1 Dot and Cross Product

เวกเตอร์ปริภูมิ n มิติ R^n สามารถแทนด้วยลูกศรพุ่งออกจากจุดกำเนิดดังรูป 2.8 ที่เป็นกรณี $n = 3$



รูปที่ 2.8 โครงงพิกัดฉากใน R^3

นิยามที่ 1 Dot Product ของ 2 เวกเตอร์ X และ Y ใน R^n เขียนเป็น $x \cdot y$

$$x \cdot y \cong \sum_{k=1}^n x_k y_k \quad (2.1)$$

โดยสัญลักษณ์ \cong มีความหมาย นิยามเท่ากับ การ Dot Product จะถูกเรียกว่า Euclidean Inner Product ใน R^n และใช้การ Transpose ของเมตริกซ์ จะได้ดังนี้

$$x \cdot y = x^T y \quad (2.2)$$

คุณสมบัติพื้นฐานของการ Dot Product ให้ $\{x, y, z\}$ เป็นเวกเตอร์ใน R^n และ $\{\alpha, \beta\}$ เป็นสเกลาร์ มีดังนี้

1. $x \cdot y \geq 0$
2. $x \cdot y = 0 \Leftrightarrow x = 0$
3. $x \cdot y = y \cdot x$
4. $(\alpha x + \beta y) \cdot z = \alpha(x \cdot z) + \beta(y \cdot z)$

นิยามที่ 2 การตั้งฉาก เวกเตอร์ x และ y ใน R^n จะตั้งฉากกันก็ต่อเมื่อ $x \cdot y = 0$

นิยามที่ 3 Completeness เป็นเซตของเวกเตอร์ที่ตั้งฉากกัน $\{x^1, x^2, x^3, \dots, x^n\}$ ใน R^n เป็น Complete ก็ต่อเมื่อ

$$y \cdot x^k = 0 \text{ เมื่อ } 1 \leq k \leq n \Rightarrow y = 0 \quad (2.3)$$

จำนวนเวกเตอร์ใน Complete Orthogonal Set เรียกว่ามิติ ดังนั้น R^3 ก็คือปริภูมิสามมิติที่มีรูปทั่วไป คือ R^n

นิยามที่ 4 ขนาด (Norm) ของเวกเตอร์ x ใน R^n เขียนเป็น $\|x\|$ นิยามได้ว่า

$$\|x\| \cong (x \cdot x)^{1/2} = (\sum_{k=1}^n x_k^2)^{1/2} \quad (2.4)$$

โดยเวกเตอร์หนึ่งหน่วยที่ตั้งฉากกันเรียก เรียกว่า Orthogonal

การหมุน (Orientation) ให้ x และ y เป็นเวกเตอร์ใน R^3 และให้ θ เป็นมุมจาก x ไป y ดังนั้น

$$x \cdot y = \|x\| \|y\| \cos \theta \quad (2.5)$$

ดังนั้นการ Dot Product จึงเป็นการวัดการหมุนระหว่างสองเวกเตอร์นั้น นอกจากนี้ยังมีกรกระทำของเวกเตอร์อีกชนิดที่เป็นประโยชน์ในการวิเคราะห์แขนหุ่นยนต์ คือ Cross Product โดยนิยามได้ดังนี้

นิยามที่ 5 Cross Product ของเวกเตอร์ u และ v ใน R^3 คือ เวกเตอร์ $w = u \times v$ ซึ่งตั้งฉากกับกฏมือขวาและ

$$w \cong \det \begin{Bmatrix} i^1 & i^2 & i^3 \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{Bmatrix} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix} \quad (2.6)$$

โดยเวกเตอร์หนึ่งหน่วย $\{i^1, i^2, i^3\}$ เป็นเวกเตอร์ตามแนวแกนของโครงพิกัดฉากตามกฏมือขวา ดังในรูป 2.2 โดย $i^1 = i^2 \times i^3$, $i^2 = i^3 \times i^1$ และ $i^3 = i^1 \times i^2$ ถ้า θ เป็นมุมจาก u ไป v ดังนั้น Cross Product

$$\|u \times v\| = \|u\| \|v\| \sin \theta \quad (2.7)$$

2.3.2 โครงพิกัด (Coordinate Frames)

นิยามที่ 1 พิกัด (Coordinates) ให้ p เป็นเวกเตอร์ใน R^n และให้ $X = \{x^1, x^2, \dots, x^n\}$ เป็น Complete Orthonormal set ของ R^n ดังนั้นพิกัดของ p เทียบกับ X เขียนเป็น $[p]^X$ และนิยามดังนี้

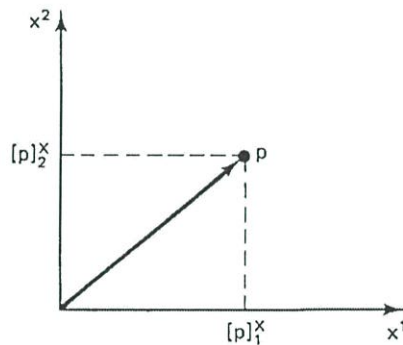
$$P = \sum_{k=1}^n [p]^X_k x^k \quad (2.8)$$

Complete Orthogonal set X บางครั้งจะเรียกว่า โครงสร้างพิกัดหนึ่งหน่วยตั้งฉาก (Orthonormal coordinate frame) ซึ่งการใช้เวกเตอร์โครงสร้างพิกัดเป็นเวกเตอร์หนึ่งหน่วยที่ตั้งฉากกันนี้ ทำให้หาพิกัดของเวกเตอร์อื่นเทียบกับโครงสร้างพิกัดนี้ได้ง่าย โดยใช้ทฤษฎีต่อไปนี้

โครงสร้างพิกัดฉาก (Orthogonal Coordinates) ให้ p เป็นเวกเตอร์ใน R^n และ $[p]^x$ เป็นพิกัดของ p เทียบกับโครงสร้างพิกัดฉาก $x = \{x^1, x^2, x^3, \dots, x^n\}$ ดังนั้น พิกัดที่ k ของ p เทียบกับ x คือ

$$[p]_k^x = p \cdot x^k ; 1 \leq k \leq x \quad (2.9)$$

จะเห็นได้ว่า พิกัดที่ k ของ p เทียบกับโครงสร้างพิกัดฉาก X ก็คือ Dot Product ของ p สมาชิกตัวที่ k ของเซต X โดยสามารถแสดงความสัมพันธ์ระหว่างเวกเตอร์กับพิกัดได้ดังรูป ซึ่งเป็นปริภูมิ 2 มิติ โดย p คือ ผลบวกของภาพฉายตั้งฉากลงบนเวกเตอร์ x^1, x^2 ตามลำดับ



รูปที่ 2.9 พิกัดของ p เทียบกับโครงสร้างพิกัด X

คำตอบของปัญหา Forward Kinematics ในแขนหุ่นยนต์ ต้องแสดงตำแหน่ง และการหมุนของ Tool ที่เคลื่อนที่เทียบกับโครงสร้างพิกัดที่ติดกับฐานที่อยู่หนึ่ง ดังนั้นจึงต้องมีการถ่ายโอนพิกัดจาก Tool ไปยังข้อมือ ข้อมือไปข้อศอก ข้อศอกไปไหล่ และต่อไปเรื่อยๆ การถ่ายโอนพิกัดแต่ละอันสามารถแสดงด้วยเมตริกซ์

การถ่ายโอนพิกัด (Coordinate Transformations) ให้ $F = \{f^1, f^2, f^3, \dots, f^n\}$ และ $M = \{m^1, m^2, m^3, \dots, m^n\}$ เป็นโครงสร้างพิกัดของ R^n และ F เป็นโครงสร้างพิกัด Orthogonal ให้ A เป็นเมตริกซ์ขนาด $n \times n$ โดย $A_{kj} \cong f^k \cdot m^j$ เมื่อ $1 \leq k, j \leq n$ ดังนั้น สำหรับแต่ละจุด p ใน R^n

$$[p]^F = A[p]^M \quad (2.10)$$

ปัญหาของการถ่ายโอนพิกัด คือ การให้พิกัดของ p เทียบกับโครงสร้างพิกัด M มาแล้วให้หาพิกัดของ p เทียบกับโครงสร้างพิกัด F เมตริกซ์ A เรียกว่า เมตริกซ์ถ่ายโอนพิกัด (Coordinate

Transformation Matrix) เป็นคอลัมน์ที่ j ของเมตริกซ์ A คือ พิกัดของเวกเตอร์ลำดับที่ j ของโครง
พิกัดต้นกำเนิด M เทียบกับโครงพิกัดปลายทาง F

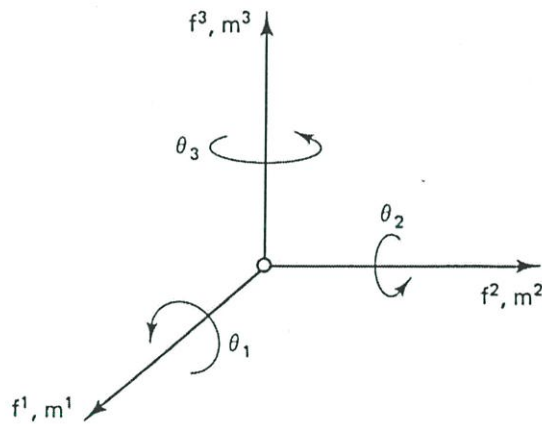
Inverse ของการถ่ายโอนโครงพิกัด (Inverse Coordinate Transformation) ให้ F
และ M เป็นโครงพิกัดฉาก Orthogonal ใน R^n มีจุดกำเนิดเดียวกัน และเป็นเมตริกซ์ถ่ายโอนพิกัด
จาก M ไป F ดังนั้น เมตริกซ์ถ่ายโอนพิกัดจาก F ไป M คือ A^{-1} โดย

$$A^{-1} = A^T \quad (2.11)$$

2.3.3 การหมุน (Rotation)

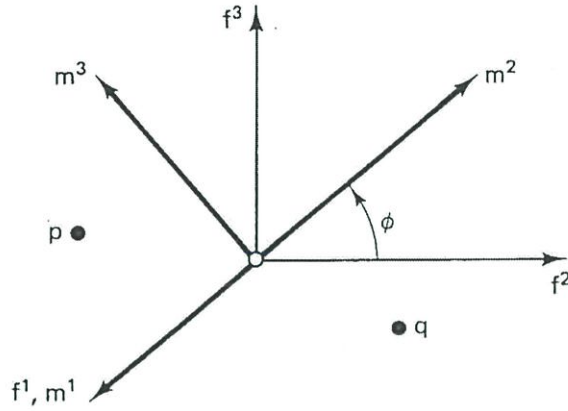
ในการระบุตำแหน่งและการหมุนของ Tool ในเทอมของพิกัดที่ติดอยู่กับฐานที่อยู่หนึ่ง
การถ่ายโอนพิกัดจึงเกี่ยวข้องกับทั้งการหมุน และการเคลื่อนที่เปลี่ยนตำแหน่ง

การหมุนพื้นฐาน (Fundamental Rotations) ถ้าโครงพิกัดเคลื่อนที่ M รอบเวกเตอร์
หนึ่งหน่วยอันหนึ่งของโครงพิกัดอยู่หนึ่ง F ดังนั้นเมตริกซ์การถ่ายโอนพิกัดที่ได้เรียกว่า เมตริกซ์การ
หมุนพื้นฐาน ในปริภูมิ R^3 มีทางเป็นไปได้ 3 ทางดังแสดงในรูปที่ 2.10



รูปที่ 2.10 การหมุนพื้นฐานใน R^3

ถ้าหมุนโครงพิกัดเคลื่อนที่ M รอบแกน f^1 ของโครงพิกัดอยู่หนึ่ง F ให้ ϕ เป็นองศาการ
หมุนวัดตามกฎมือขวา ดังรูปที่ 2.11



รูปที่ 2.11 การหมุนของ M รอบ f^1 ด้วยมุม ϕ

ให้ $R_1(\phi)$ เป็นเมตริกซ์การถ่ายโอนพิกัดเคลื่อนที่ M ไป F (จากการถ่ายโอนพิกัด) จะ
ได้

$$R_1(\phi) = \begin{bmatrix} f^1 \cdot m^1 & f^1 \cdot m^2 & f^1 \cdot m^3 \\ f^2 \cdot m^1 & f^2 \cdot m^2 & f^2 \cdot m^3 \\ f^3 \cdot m^1 & f^3 \cdot m^2 & f^3 \cdot m^3 \end{bmatrix} \quad (2.12)$$

เนื่องจากการหมุนรอบแกน f^1 จากรูป 2.8 จะได้ $f^1 = m^1$ แทนใน $R_1(\phi)$ และจากรูป
ที่ 2.11 เวกเตอร์ $\{f^1, f^2, f^3\}$ และ $\{m^1, m^2, m^3\}$ เป็นเซตของเวกเตอร์หนึ่งหน่วยที่ตั้งฉากกัน ดังนั้น
จะได้

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & f^2 \cdot m^2 & f^2 \cdot m^3 \\ 0 & f^3 \cdot m^2 & f^3 \cdot m^3 \end{bmatrix} \quad (2.13)$$

จากรูปที่ 2.11 เวกเตอร์ $\{f^2, f^3, m^2, m^3\}$ อยู่ในระนาบตั้งฉาก $f^1 = m^1$ จากการหมุน
(Orientation) จะได้ว่า Dot product ของเวกเตอร์เท่ากับค่า cosine ของมุมระหว่าง 2 เวกเตอร์นั้น
จากรูป 2.11 มุมจาก f^2 ไป m^2 และ f^3 ไป m^3 เท่ากับ ϕ มุมจาก f^3 ไป m^3 เท่ากับ $\pi/2 + \phi$ และ
มุมจาก f^2 ไป m^2 เท่ากับ $-\pi/2 + \phi$ ดังนั้นจะได้

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.14)$$

ในทำนองเดียวกัน ถ้า $R_2(\phi)$ และ $R_3(\phi)$ แสดงการหมุนของโครงพิกัด M รอบ f^2
และ f^3 ของ F ตามลำดับจะได้

$$R_2(\phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \quad (2.15)$$

$$R_3(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

2.3.4 พิกัดโฮโมจีเนียส (Homogeneous Coordinates)

การหมุน (Rotation) ใช้อธิบายลักษณะการวางตัว (Orientation) ของ Tool แต่ในการอธิบายตำแหน่งของ Tool ให้สัมพันธ์กับโครงพิกัดอ้างอิงฐานของแขนหุ่นยนต์นั้น จำเป็นต้องใช้การถ่ายโอนอีกแบบหนึ่งซึ่งเรียกว่า การเลื่อนตำแหน่ง (Translation) การเลื่อนตำแหน่งนั้นมีข้อแตกต่างจากการหมุนที่สำคัญ คือ การหมุนจุดกำเนิดของโครงพิกัดที่ถูกหมุนเป็นตำแหน่งเดียวกับจุดกำเนิดของโครงพิกัดดั้งเดิม จึงสามารถแสดงการหมุนในปริภูมิ 3 มิติ ได้ด้วยเมตริกซ์ 3×3 แต่จุดกำเนิดของโครงพิกัดที่เลื่อนตำแหน่งไปนั้นไม่เป็นจุดเดียวกับจุดกำเนิดของโครงพิกัดดั้งเดิม ดังนั้นจึงไม่สามารถแสดงการเลื่อนตำแหน่งใน R^3 ด้วยเมตริกซ์ 3×3 ได้ ในการนี้จะใช้ปริภูมิ 4 มิติเพื่อช่วยในการนิยามโฮโมจีเนียส

2.3.4.1 พิกัดโฮโมจีเนียส (Homogeneous Coordinates)

ให้ q เป็นจุดใน R^3 และ F เป็นโครงพิกัดฉากสำหรับ R^3 ถ้า σ เป็นสเกลาร์เวกเตอร์ที่ไม่เป็นศูนย์ ดังนั้นพิกัดโฮโมจีเนียสของ q เทียบกับ F เขียนเป็น $[q]^F$ นิยามว่า

$$[q]^F \cong \sigma[q_1, q_2, q_3, 1]^T \quad (2.17)$$

ดังนั้น พิกัดโฮโมจีเนียสของจุด q ใน R^3 แสดงได้ด้วยเวกเตอร์ $[q]^F$ ในปริภูมิ 4 มิติ R^4 ในการแปลงพิกัดโฮโมจีเนียส 4 มิติกลับเป็นเวกเตอร์ 3 มิติดั้งเดิม เราใช้ $q = H_\sigma[q]^F$ โดย H_σ เป็นเมตริกซ์แปลงพิกัดโฮโมจีเนียสขนาด 3×4 (Homogeneous Coordinate Conversion Matrix)

$$H_\sigma \cong \frac{1}{\sigma} [I, 0] \quad (2.18)$$

จะสังเกตได้ว่า พิกัดโฮโมจีเนียส $[q]^F$ มีได้หลายค่า ขึ้นอยู่กับสเกลาร์แฟกเตอร์ $\sigma \neq 0$ ที่ใช้ แต่จะให้ผลเป็นเวกเตอร์ 3 มิติ q เดียวกัน ในหุ่นยนต์มักใช้ $\sigma = 1$ ในกรณีนี้สามารถหาพิกัดโฮโมจีเนียสจากพิกัด 3 มิติได้โดยเติม 1 ให้เป็นสมาชิกตัวที่ 4 ในทำนองเดียวกันจะแปลงพิกัดโฮโมจีเนียสจากพิกัด 4 มิติกลับเป็น 3 มิติได้โดยตัดสมาชิกตัวที่ 4 ทิ้ง

ถ้าจุดในปริภูมิ 3 มิติ ถูกแสดงในเทอมของพิกัดโฮโมจีเนียส และต้องการเปลี่ยนโครงพิกัดหนึ่งไปยังโครงพิกัดอื่น จะใช้เมตริกซ์ถ่ายโอนโฮโมจีเนียส (Homogeneous Transformation Matrix) ขนาด 4×4 โดยปกติเมตริกซ์ T นี้ แปรได้เป็นเมตริกซ์ย่อย 4 ส่วน คือ

$$T = \begin{bmatrix} & R & & | & p \\ \hline & & \eta^T & & | & \sigma \\ \hline & & & & & \end{bmatrix} \quad (2.19)$$

โดย R คือ Rotation matrix แสดงการหมุนของโครงพิกัดเคลื่อนที่เทียบกับโครงพิกัดอยู่นิ่ง

p คือ Translation factor แสดงตำแหน่งจุดกำเนิดของโครงพิกัดเคลื่อนที่เทียบกับโครงพิกัดอยู่นิ่ง

σ คือ Scalar factor ที่ไม่เป็นศูนย์ ปกติให้เท่ากับหนึ่ง

η^T คือ Perspective Vector ปกติเท่ากับศูนย์

ในเทอมของแกนหุ่นยนต์ p แสดงตำแหน่งปลาย Tool R แสดงการหมุนของ Tool η แสดงจุดในการมองของอุปกรณ์ เช่น กล้องจับภาพ ในกรณีนี้จะให้ค่าที่ไม่เป็นศูนย์ σ ปกติใช้ 1 เป็นค่ามาตรฐาน

2.3.4.2 การเลื่อนตำแหน่งและการหมุน (Translation and Rotations)

การกระทำพื้นฐานของการเลื่อนตำแหน่งและการหมุน จัดว่าเป็นกรณีพิเศษในเมตริกซ์ถ่ายโอนโฮโมจีเนียส 4x4 เช่น กรณีการหมุนสมมติให้ F และ M เป็นโครงพิกัดฉากที่เริ่มต้นทับกันสนิท ถ้าหมุน M ไป ϕ รอบเวกเตอร์ลำดับที่ k ของ F จะแสดงในเทอมของพิกัดโฮโมจีเนียสเป็นเมตริกซ์ 4x4 Rot (ϕ, k) ได้ว่า

$$\text{Rot}(\phi, k) \cong \begin{bmatrix} & & & & | & 0 \\ & & R_k(\phi) & & | & 0 \\ \hline & & & & & | & 0 \\ 0 & & 0 & & 0 & | & 1 \end{bmatrix} \quad 1 \leq k \leq 3 \quad (2.20)$$

โดย $R_k(\phi)$ เมตริกซ์การหมุนพื้นฐานลำดับที่ k ที่ได้มาจากการนำเมตริกซ์การหมุน Dot product ด้วยเมตริกซ์การถ่ายโอนพิกัด เรียก Rot (ϕ, k) ว่าเป็นเมตริกซ์การหมุนโฮโมจีเนียสพื้นฐานลำดับที่ k จะสังเกตเห็นว่า เวกเตอร์ตำแหน่ง p ในแต่ละกรณีเป็นศูนย์

พิกัดโฮโมจีเนียสยังมีประโยชน์ในการแสดงการเลื่อนตำแหน่งได้ เช่น ให้ F และ M เป็นโครงพิกัดฉากที่ทับกันสนิท สมมติให้เลื่อนจุดกำเนิดของโครงพิกัดเคลื่อนที่ M ไป p_k ตามแนวเวกเตอร์หนึ่งหน่วยด้วยเมตริกซ์ 4x4 Tran (p) เป็น

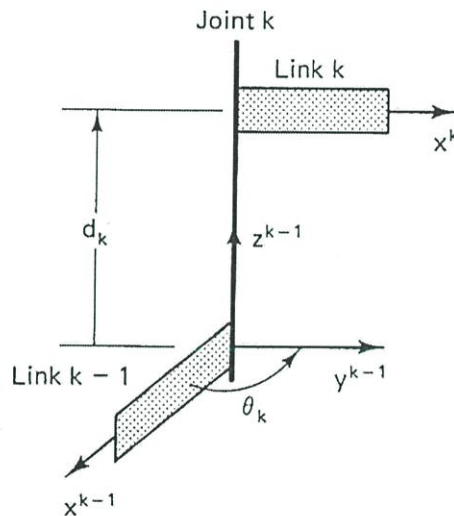
$$\text{Tran}(p) \cong \begin{bmatrix} 1 & 0 & 0 & | & p_1 \\ 0 & 1 & 0 & | & p_2 \\ 0 & 0 & 1 & | & p_3 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (2.21)$$

เรียก Tran (p) ว่าเป็นเมตริกซ์เลื่อนตำแหน่งโฮโมจีเนียสพื้นฐาน (Fundamental Homogeneous Transformation Matrix) จะสังเกตเห็นได้ว่าเวกเตอร์ตำแหน่งใน p ในขณะที่เมตริกซ์การหมุน 3×3 จะเป็นเมตริกซ์เอกลักษณ์ I

2.3.5 พิกัดแขน (Link Coordinate)

ดังได้กล่าวแล้วว่า แขนหุ่นยนต์ประกอบด้วยแขนหลายๆ อันมาเชื่อมต่อกันด้วยข้อต่อ จุดประสงค์ของหัวข้อนี้ คือ การกำหนดโครงพิกัดลงบนแต่ละแขน จากนั้นจึงทำการหาสมการแขน (Arm Equation) ที่แสดงการเคลื่อนที่ของแขนหุ่นยนต์ โดยขั้นแรกจะศึกษาพารามิเตอร์ที่ใช้ในการออกแบบตัวแขนหุ่นยนต์ก่อน

Kinematics Parameters ตำแหน่งและการหมุนสัมพันธ์ของแขนที่อยู่ติดกัน แสดงได้ด้วย Joint parameters 2 ตัว แสดงดังในรูป 2.12



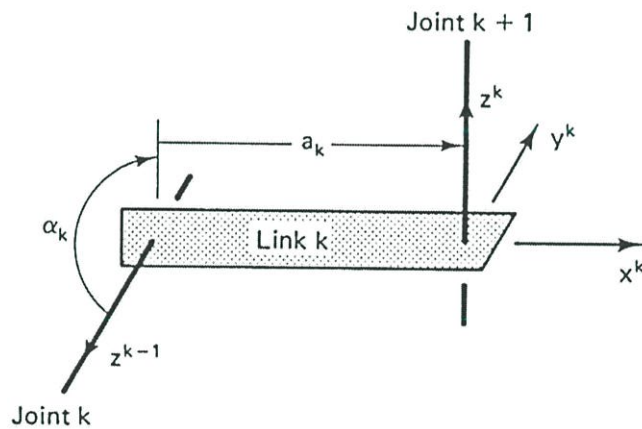
รูปที่ 2.12 มุมข้อต่อ θ และระยะข้อต่อ d

จากรูปข้อต่อ k เชื่อมแขน $k-1$ พารามิเตอร์ที่เกี่ยวข้องกับข้อต่อ k ถูกนิยามเทียบกับแกน z^{k-1} ซึ่งอยู่ในแนวเดียวกับแกนของข้อต่อ k Joint Parameter ตัวแรกเรียกว่า มุมข้อต่อ (Joint Angle); θ_k ซึ่งเป็นการหมุนรอบ z^{k-1} เพื่อให้แกน x^{k-1} ขนานกับแกน x^k Joint Parameter ตัวที่สองเรียกว่า ระยะข้อต่อ (Joint Distance); d_k ซึ่งเป็นการเลื่อนไปตามแกนของข้อต่อ k ตามปกติแต่ละข้อจะมี Joint Parameter หนึ่งตัวมีค่าคงที่ และมีอีกค่าเปลี่ยนแปลงได้ ซึ่งขึ้นอยู่กับชนิดของข้อต่อ ดังแสดงในตารางที่ 2.1 สำหรับข้อต่อหมุนมุมข้อต่อ θ_k จะเปลี่ยนค่าได้ และระยะข้อต่อ d_k มีค่าคงที่ ดังนั้นแขนจะมีการหมุนสัมพันธ์รอบแกนของข้อต่อ k สำหรับข้อต่อเลื่อน ระยะข้อต่อ d_k จะเปลี่ยนค่าได้ และมุมข้อต่อ k สำหรับข้อต่อเลื่อน ระยะข้อต่อ d_k จะเปลี่ยนค่าได้ และมุมข้อต่อ θ_k จะคงที่ ดังนั้นแขน จะมีการเลื่อนตำแหน่งสัมพันธ์กันไปตามแนวแกนของข้อต่อ k

ตาราง 2.1 Kinematics Parameters

Arm Parameter	สัญลักษณ์	ข้อต่อหมุน (R)	ข้อต่อเลื่อน (P)
มุมข้อต่อ (Joint angle)	θ	ตัวแปร	คงที่
ระยะข้อต่อ (Joint distance)	d	คงที่	ตัวแปร
ความยาว link (Link length)	a	คงที่	คงที่
มุมบิด link (Link twist angle)	α	คงที่	คงที่

ข้อต่อจะอยู่ระหว่างแขน 2 อันที่อยู่ติดอยู่ ในทางกลับกันจึงมีแขน 1 อันที่อยู่ระหว่างข้อต่อ 2 อัน ตำแหน่งและการหมุนสัมพันธ์ของแกนของข้อต่อทั้ง 2 แสดงได้ด้วย Link parameters ดังแสดงในรูป 2.13 โดยแขน k เชื่อมข้อต่อ $k+1$ พารามิเตอร์ที่เกี่ยวข้องกับแขน ถูกนิยามเทียบกับ x^k ซึ่งเป็น Common Normal (ตั้งฉาก) ระหว่างแกนของข้อต่อ k และ $k+1$ Link Parameter ตัวแรกเรียกว่า ความยาวแขน (Link Length) a_k ซึ่งเป็นการเลื่อนตำแหน่งไปตามแนว x^k เพื่อให้แกน z^{k-1} ตัดกับ z^k Link Parameter ตัวที่สองเรียกว่า มุมบิดแขน (Link Twist Angle) α_k ซึ่งเป็นการหมุนรอบ x^k เพื่อให้แกน z^{k-1} ขนานกับ z^k

รูปที่ 2.13 ความยาวแขน (Link Length) a_k และมุมบิดแขน (Link Twist Angle) α_k

ตามปกติ Link Parameter ทั้งสองจะมีค่าคงที่ และถูกระบุเป็นส่วนหนึ่งของการออกแบบทางกลไก สำหรับหุ่นยนต์ทางอุตสาหกรรม มุมบิดแขน (Link Twist Angle) α_k มักเป็นจำนวนเท่าของ $\pi/2$ เรเดียน บางครั้งแกนของข้อต่อ k และ $k+1$ จะตัดกันทำให้ความยาวแขนเป็นศูนย์ ดังอย่างเช่น ข้อมือทรงกลม (Spherical Wrist) ของแขนหุ่นยนต์ที่มี $n-3$ แกนสุดท้ายตัดกัน

นอกจากนี้หุ่นยนต์อาจถูกออกแบบให้มี Kinematic Parameters หลายๆ ตัวที่ที่ค่าคงที่และเท่ากับศูนย์ ซึ่งเรียกว่า แขนหุ่นยนต์แบบ Kinematically Simple Manipulators

สำหรับแขนหุ่นยนต์ n แขน ในแต่ละแกนพารามิเตอร์ 3 ตัวจะมีค่าคงที่ และขึ้นอยู่กับ การออกแบบทางกลไก ในขณะที่พารามิเตอร์ตัวที่ 4 จะเป็นตัวแปรข้อต่อ (Joint Variable) ดังตาราง ที่ 2.1 สำหรับหุ่นยนต์พิกัดฉาก ตัวแปรข้อต่อ 3 ตัวแรก จะเป็นมุมข้อต่อทั้งหมด ส่วนหุ่นยนต์พิกัด ทรงกระบอก ทรงกลม และ SCARA ตัวแปร 3 ตัวแรก จะมีมุมข้อต่อและระยะข้อต่อ

2.3.6 สมการแขน (The Arm Equation)

เมื่อกำหนดโครงพิกัดลงบนแขนได้แล้วทำการย้ายโครงพิกัด k ไปยังโครงพิกัด $k-1$ โดยใช้เมตริกซ์การถ่ายโอนพิกัดโฮโมจีเนียส เมื่อนำเมตริกซ์ที่ได้มาคูณกันทั้งหมด จะได้เมตริกซ์ที่แสดง การย้ายพิกัดของ Tool เทียบกับพิกัดฐาน โดยเมตริกซ์ที่ได้มาคูณกันทั้งหมด จะได้เมตริกซ์ที่แสดง การย้ายพิกัดของ Tool เทียบพิกัดฐาน โดยเมตริกซ์ที่ได้นี้เรียกว่า เมตริกซ์แขน (Arm Matrix)

2.3.6.1 เมตริกซ์แขน (Arm Equation)

การสร้างเมตริกซ์การถ่ายโอนพิกัดโฮโมจีเนียสจากโครงพิกัด k ไปยังโครง พิกัด $k-1$ จะมี 4 ขั้นตอน แต่ละขั้นตอนจะเกี่ยวข้องกับ Kinematics Parameter 1 ตัว ในการ พิจารณาจะทำการหมุนและเลื่อนตำแหน่งโครงพิกัด $k-1$ จนทับสนิทกับโครงพิกัด k โดยกระทำตาม ขั้นตอนดังตารางที่ 2.2

ตารางที่ 2.2 การถ่ายโอนโครงพิกัด k ไปยังโครงพิกัด $k-1$

ขั้นตอน (Operation)	คำอธิบาย (Description)
1	หมุน L_{k-1} รอบ z^{k-1} เป็นมุม θ_k
2	เลื่อน L_{k-1} ไปตามแนว z^{k-1} เป็นระยะ d_k
3	เลื่อน L_{k-1} ไปตามแนว x^{k-1} เป็นระยะ a_k
4	หมุน L_{k-1} รอบ x^{k-1} เป็นมุม α_k

จากรูปที่ 2.12 จะได้ว่าขั้นตอนที่ 4 ทำให้แกน x^{k-1} ขนานกับแกน x^k ขั้นตอน ที่ 2 แกน x^{k-1} จะทับกับ x^k ต่อมา จากรูปที่ 2.13 จะได้ว่าขั้นตอนที่ 3 ทำให้จุดกำเนิดของโครง L_{k-1} ทับกับ L_k ขั้นตอนที่ 4 ทำให้แกน z^{k-1} ทับกับ z^k ดังนั้นโครง L_{k-1} และ L_k ทับกันพอดี

การถ่ายโอนโครงพิกัด (Link-Coordinate Transformation) ให้ $\{L_0, L_1, \dots, L_n\}$ เป็นเซตของโครงพิกัด $[q]^k$ และ $[q]^{k-1}$ เป็นพิกัดโฮโมจีเนียสของจุด q เทียบกับโครง L_k และ L_{k-1} สำหรับ $1 \leq k \leq n$ จะได้ $[q]^{k-1} = T_{k-1}^k [q]^k$ โดยที่ T_{k-1}^k คือ การถ่ายโอนโครงพิกัด k ไปยัง $k-1$

$$T_{k-1}^k = \begin{bmatrix} c\theta_k & -c\alpha_k s\theta_k & s\alpha_k s\theta_k & ac\theta_k \\ s\theta_k & c\alpha_k c\theta_k & -s\alpha_k c\theta_k & as\theta_k \\ 0 & s\alpha_k & c\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.22)$$

ดังได้กล่าวมาแล้วว่า Kinematics Parameter 3 ตัวที่ปรากฏใน T_{k-1}^k เป็นค่าคงที่ ตัวที่เหลือจะเป็นตัวแปรข้อต่อ ซึ่งอาจเป็น θ_k (สำหรับข้อต่อหมุน) หรือ d_k (สำหรับข้อต่อเลื่อน)

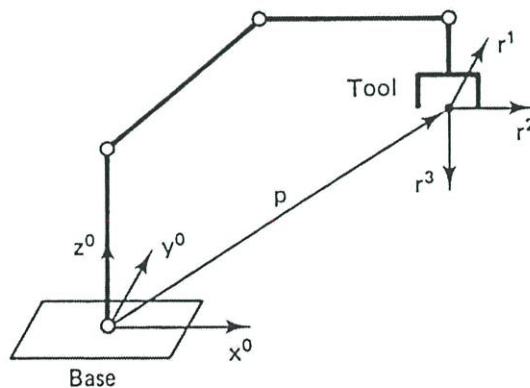
ตามปกติเมตริกซ์ถ่ายโอนโคโนกรังพิกัดโฮโมจีเนียส T_{k-1}^k จะเป็นฟังก์ชันของ q_k สำหรับ $1 \leq k \leq n$ ในการแก้ปัญหา Forward Kinematics ต้องพิจารณาดำแหน่ง และการหมุนของ Tool เทียบกับโคโนกรังพิกัดที่ติดอยู่กับฐาน การถ่ายโอนจากพิกัด Tool ไปยังพิกัดฐานจะเริ่มจากปลาย Tool ถ่ายโอนย้อนกลับไปที่ละโคโนกรังจนถึงฐาน ถ้า T_{base}^{tool} แสดงการถ่ายโอนจากพิกัดปลาย Tool (Link n) ไปยังพิกัดฐาน (Link 0) ดังนั้น

$$T_{base}^{tool}(q) = T_0^1(q_1)T_1^2(q_2) \dots T_{n-1}^n(q_n) = T_0^n(q) \quad (2.23)$$

2.3.6.2 สมการแขน (The Arm Equation)

$$T_{base}^{tool}(q) = \begin{bmatrix} R(q) & p(q) \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

โดย $R(q)$ เป็นเมตริกซ์ย่อย 3×3 แสดงการหมุนของ Tool คอลัมน์ทั้ง 3 ของ R แสดงทิศทางของเวกเตอร์หนึ่งหน่วย $\{r^1, r^2, r^3\}$ ของโคโนกรังเครื่องมือเทียบกับโคโนกรังฐาน โดย $p(q)$ เป็นเมตริกซ์ย่อย 3×1 แสดงตำแหน่งของปลาย Tool เป็นพิกัดปลายของ Tool เทียบกับโคโนกรังฐาน ดังแสดงในรูป 2.14



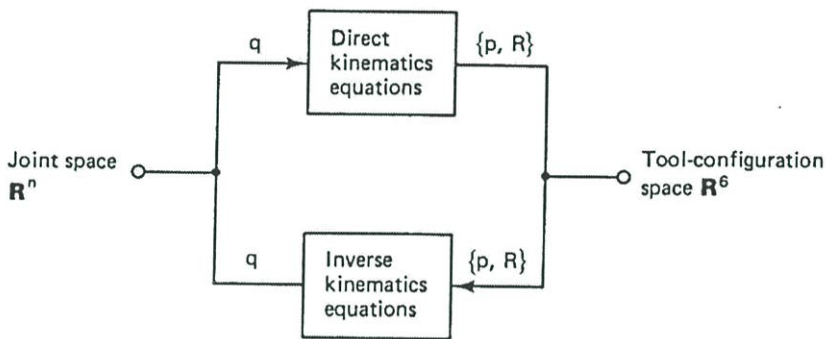
รูปที่ 2.14 ตำแหน่งและการหมุนของ Tool ในพิกัดฐาน

2.4 Inverse Kinematics: Solving the Arm Equation [2]

ในหัวข้อที่แล้วได้ทำการหาตำแหน่งและการหมุนของ Tool จากตัวแปรข้อต่อที่กำหนดให้ แต่ในหัวข้อนี้จะพิจารณาย้อนกลับ คือ การหาตัวแปรข้อต่อจากตำแหน่งและการหมุนของ Tool ที่กำหนดให้ ปัญหา Inverse Kinematics มีความสำคัญและนำไปใช้งานจริงในแขนหุ่นยนต์ ไม่มีขั้นตอนการแก้ปัญหาที่แน่นอนและมักมีหลายคำตอบที่ได้

2.4.1 คุณสมบัติทั่วไปของคำตอบ (General Properties of Solutions)

Inverse Kinematics ดังรูปที่ 2.15 เป็นการกำหนดตำแหน่ง p และการหมุน R ของเครื่องมือเทียบกับโครงฟัดอ้างอิงแล้วทำการคำนวณหาตัวแปรข้อต่อ q ที่สอดคล้องกับสมการแขน



รูปที่ 2.15 Forward (Direct) และ Inverse Kinematics

การมีอยู่ของคำตอบ (Existence of Solutions) พิจารณาในสถานะที่คำตอบของ Inverse Kinematics มีจริง คือ ตำแหน่งปลาย Tool p อยู่ในของเขตการทำงานและการหมุนของ Tool ไม่เกินข้อจำกัดของตัวแปรข้อต่อ จากสมการแขน

$$T_{\text{base}}^{\text{tool}}(q) = \begin{bmatrix} R_{11} & R_{12} & R_{13} & | & p_1 \\ R_{21} & R_{22} & R_{23} & | & p_2 \\ R_{31} & R_{32} & R_{33} & | & p_3 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix} \quad (2.25)$$

เนื่องจากบรรทัดสุดท้ายของ $T_{\text{base}}^{\text{tool}}$ มีค่าคงที่ ดังนั้นสมการแขนจึงเป็นระบบสมการพีชคณิตเชิงเส้น 12 สมการ มี q เป็นตัวแปร n ตัว แต่ทั้ง 12 สมการไม่เป็นอิสระซึ่งกันและกัน

$$R = \{r^1 \quad r^2 \quad r^3\} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (2.26)$$

เมื่อ r^1, r^2, r^3 คือ เวกเตอร์ในแนวแกนโครงพิกัดที่ปลาย Tool เทียบกับโครงพิกัดฐาน

R_{11} คือ องค์ประกอบของ r^1 ในแนวแกน x ของโครงพิกัดฐานอ้างอิง (f^1)

R_{21} คือ องค์ประกอบของ r^2 ในแนวแกน y ของโครงพิกัดฐานอ้างอิง (f^2)

R_{31} คือ องค์ประกอบของ r^3 ในแนวแกน z ของโครงพิกัดฐานอ้างอิง (f^3)

จาก $R^{-1} = R^T$ จะได้ว่า $R^T R = I$ เพื่อพิจารณาผลคูณของเทอมในแนวอนเส้นทแยงมุมของเมตริกซ์ จึงสรุปได้ว่า r^1, r^2, r^3 ตั้งฉากซึ่งกันและกัน

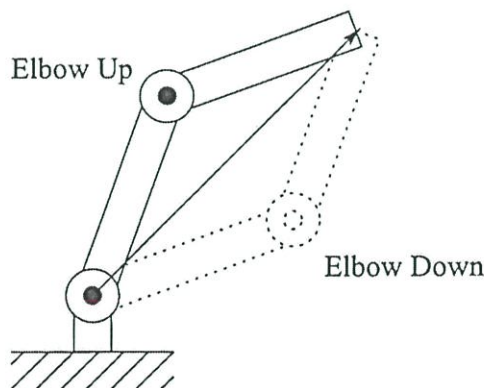
$$\begin{aligned} r^1 \cdot r^2 &= 0 \\ r^1 \cdot r^3 &= 0 \\ r^2 \cdot r^3 &= 0 \end{aligned} \quad (2.27)$$

และพิจารณาผลคูณของเทอมในแนวเส้นทแยงมุม ได้ว่า r^1, r^2, r^3 เป็นเวกเตอร์หนึ่งหน่วยจะได้ว่า $\|r^k\| = 1$ เมื่อ $1 \leq k \leq 3$

ดังนั้นจะได้รับความสัมพันธ์มา 6 สมการ จากระบบสมการ 12 สมการจึงมีเพียง 6 สมการที่เป็นอิสระต่อกัน ถ้าเราต้องการคำตอบทั่วไปจึงควรมีตัวแปรข้อต่อ q อย่างน้อยที่สุด 6 ตัว $q \geq 6$

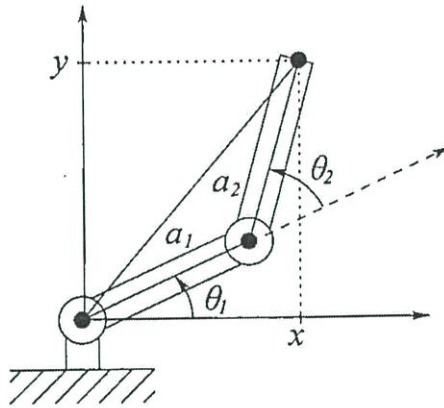
2.4.2 Outline of Inverse Kinematics [4]

ให้มุมข้อต่อเป็น θ_1 และ θ_2 จะสามารถกำหนดจุดของ Tool สัมพันธ์กับ x, y เพื่อที่จะออกคำสั่งให้หุ่นยนต์เคลื่อนไหวไปยังพื้นที่ที่ต้องการ ต้องทำการ Inverse โดยสิ่งที่ต้องการ คือ ตัวแปรข้อต่อ θ_1 และ θ_2 ในเทอมของ x และ y จากสมการ Forward Kinematics ที่เป็น Nonlinear การแก้สมการอาจจะหาได้ยากเพราะการสมการนั้นแตกต่างกัน เห็นได้จาก 2 แขนตรงสองแขนที่อาจจะไม่มีสมการ ตัวอย่างเช่น กำหนดให้ (x, y) เป็นจุดปลายแขนหุ่นยนต์ อาจจะมีคำตอบของสมการตามภาพที่เกิดดังรูปที่ 2.16



รูปที่ 2.16 คำตอบสามารถทำให้รูปร่างแขนเป็นได้ทั้ง Elbow up และ Elbow down

จากรูปที่ 2.17 การใช้กฎของ Cosine จะเห็นได้ว่า



รูปที่ 2.17 การแก้สมการมุมข้อต่อของแขนหุ่นยนต์สองแขน

การหามุม θ_2 ทำได้โดย

$$\cos \theta_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1 a_2} := D \quad (2.28)$$

และกำหนดให้ θ_2 เป็น $\theta_2 = \cos^{-1}(D)$ อย่างไรก็ตามทางที่ดีที่สุดในการหา θ_2 โดยการสังเกตว่าถ้า $\cos(\theta_2)$ จากสมการข้างต้น จะได้ $\sin(\theta_2)$ ดังนี้

$$\sin(\theta_2) = \pm \sqrt{1 - D^2} \quad (2.29)$$

และดังนั้น θ_2 จะได้ดังนี้

$$\theta_2 = \tan^{-1} \frac{\pm \sqrt{1 - D^2}}{D} \quad (2.30)$$

ประโยชน์ของวิธีนี้จะได้ค่าสมการทั้ง Elbow-up และ Elbow-down โดยการเลือกใช้ค่าบวกหรือค่าลบของในสมการข้างต้น จากนั้นจะได้ θ_1 ดังนี้

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1} \left(\frac{a_2 \sin \theta_2}{a_1 + a_1 \cos \theta_2} \right) \quad (2.31)$$

สังเกตได้ว่ามุมของ θ_1 จะขึ้นอยู่กับ θ_2 โดยทั้งนี้ขึ้นอยู่กับทางกลตั้งแต่การคาดการณ์ของค่า θ_1 ที่ขึ้นกับการเลือกใช้ θ_2

2.5 การควบคุมการเคลื่อนที่ (Motion Control) [5]

การควบคุมการเคลื่อนที่ สำหรับการทำงาน อาทิเช่น การจับ และ Pick-and-Place หุ่นยนต์ จำเป็นต้องทำงานเข้าไปมาพิภักที่มีการระบุหรือสอนการเคลื่อนที่ให้กับหุ่นยนต์ ใช้สำหรับงานที่มีความซับซ้อน เช่น งานเชื่อม และงานตักแต่ง (ฟันสี) การเคลื่อนที่ที่ต้องการควบคุมอย่างต่อเนื่องไปตามเส้นทาง โดยการควบคุมเป้าหมาย และความเร็ว

2.6 การเคลื่อนที่

2.6.1 การเคลื่อนที่เชิงเส้น (Linear Motion) [6]

การเคลื่อนที่เชิงเส้นเป็นการเคลื่อนที่ตามแนวเส้นตรง สามารถอธิบายทางคณิตศาสตร์ได้โดยพื้นที่หนึ่งมิติ การเคลื่อนที่เชิงเส้นแบ่งออกเป็น 2 ประเภท คือ การเคลื่อนที่ด้วยความเร็วคงที่หรืออัตราเร่งเป็นศูนย์ และการเคลื่อนที่ด้วยความเร็วไม่คงที่หรืออัตราเร่งไม่เป็นศูนย์ การเคลื่อนที่ของอนุภาคกับเส้นทางสามารถอธิบายได้ด้วยตำแหน่ง x กับ t ที่ต่างๆ (เวลา)

2.6.1.1 การกระจัด (Displacement)

ในการเคลื่อนที่ทั้งหมดวัตถุจะเคลื่อนย้ายในเวลาและระยะทางเดียวกันที่เรียกว่าการเคลื่อนที่แบบเลื่อน ซึ่งการเคลื่อนที่แนวเชิงเส้นจะเป็นการเคลื่อนที่แนวในมิติเดียว ระยะทางที่เดินทางของวัตถุในทิศทางใดจะเรียกว่าการกระจัด โดยมีหน่วย SI เป็นเมตร (Meter) ถ้า x_1 เป็นจุดเริ่มต้นของวัตถุ และ x_2 เป็นจุดสุดท้าย ดังนั้นทางคณิตศาสตร์จะเป็นสมการดังนี้

$$\Delta x = x_2 - x_1 \quad (2.32)$$

ในการเคลื่อนที่แบบหมุนการกระจัดจะเป็นระยะเชิงมุม θ ในการวัดเรเดียน การกระจัดของวัตถุไม่สามารถมากกว่าระยะทางได้

2.6.1.2 ความเร็ว (Velocity)

เป็นอัตราการเปลี่ยนแปลงของการเคลื่อนที่ โดยที่ถูกระบุให้เป็นการกระจัดที่ขึ้นอยู่กับการเคลื่อนที่ โดยมีหน่วย SI เป็นเมตรต่อวินาที (Meter per Second)

ความเร็วเฉลี่ย (Average Velocity) เป็นสัดส่วนของการกระจัดทั้งหมด Δx กับช่วงเวลา Δt จะได้สมการดังนี้

$$v_{av} = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1} \quad (2.33)$$

เมื่อ t_1 คือ เวลาที่วัตถุอยู่ตำแหน่ง x_1
 t_2 คือ เวลาที่วัตถุอยู่ตำแหน่ง x_2

2.6.1.3 ความเร่ง (Acceleration)

เป็นการเปลี่ยนแปลงความเร็วที่ขึ้นอยู่กับเวลา ความเร่งเป็นอนุพันธ์อันดับสองของการกระจัด โดยมีหน่วย SI เป็นเมตรต่อวินาทียกกำลังสอง (Meter per Second Squared)

ความเร่งเฉลี่ย (Average Velocity) เป็นสัดส่วนของความเร็วทั้งหมด Δv กับช่วงเวลา Δt จะได้สมการดังนี้

$$a_{av} = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{t_2 - t_1} \quad (2.34)$$

2.6.1.4 สมการการเคลื่อนที่ (Equation of kinematics)

จากในความเร่งคงที่ จะมีสมการดังนี้

$$v_2 = v_1 + at \quad (2.35)$$

$$s = v_1 t + \frac{1}{2} at^2 \quad (2.36)$$

$$v_2^2 = v_1^2 + 2as \quad (2.37)$$

$$s = \frac{1}{2}(v_2 + v_1)t \quad (2.38)$$

2.6.2 การเคลื่อนที่แบบหมุนรอบแกนตายตัว (Rotation Around a Fixed Axis) [4]

การเคลื่อนที่หมุนรอบแกนตายตัวเป็นรูปแบบหนึ่งของการเคลื่อนที่แบบหมุน โดยแกนนิ่งที่สมมติขึ้นไม่สามารถย้ายหรือเคลื่อนที่ หรือไม่สามารถโยกได้ เมื่อวัตถุแข็งเกร็งเคลื่อนที่เพียงการหมุนในแกนที่ตายตัว ทุกๆจุดที่เกิดขึ้นจะรวมกันเป็นวงกลม ที่จุดศูนย์กลางของวงกลมนี้จะอยู่ที่แกนหมุน ขณะที่วัตถุหมุนจะตั้งฉากกับแกนที่กวาดเป็นมุม θ และมุมนี้จะเหมือนกันทุกจุด ถ้า k เป็นเวกเตอร์หนึ่งหน่วยในทิศทางของการของแกนหมุนดังนั้น ความเร็วเชิงมุมจะได้

$$\omega = \dot{\theta}k \quad (2.39)$$

$\dot{\theta}$ คือ อนุพันธ์เวลาของ θ

ความเร็วเชิงมุมของวัตถุ จากความรู้เชิงเส้นของจุดต่างๆ จะได้สมการ

$$v = \omega \times r \quad (2.40)$$

โดยที่ r เป็นเวกเตอร์จากจุดกำเนิด (ซึ่งในกรณีนี้จะหมายถึงอยู่บนแกนหมุน) ไปถึงจุด ในความเป็นจริงแล้วการคำนวณความเร็ว v นี้เป็นพื้นฐานในหลักสูตรแบบเบื้องต้นของระบบไดนามิกส์ ดังนั้นบทบาทหลักของความเร็วเชิงมุม คือ การเหนี่ยวนำให้เกิดเป็นความเร็วเชิงเส้นของจุดในวัตถุแข็งเกร็ง

2.7 การควบคุมแบบ PID [7]

PID (ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์) เป็นระบบควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับแต่ค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของ PID ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ

2.7.1 แนวทางคำนวณของ PID

ขึ้นอยู่กับสามตัวแปร คือ ค่าสัดส่วน ปริพันธ์ และอนุพันธ์ค่าสัดส่วน กำหนดจากผลของความผิดพลาดในปัจจุบัน ค่าปริพันธ์กำหนดจากผลของผลรวมความผิดพลาดที่ซึ่งพ่วงผ่านไป และค่าอนุพันธ์กำหนดจากผลของอัตราการเปลี่ยนแปลงของค่าความผิดพลาด น้ำหนักที่เกิดจากการรวมกันของทั้งสามนี้จะใช้ในการปรับกระบวนการโดยการปรับค่าคงที่ใน PID ตัวควบคุมสามารถปรับรูปแบบการควบคุมให้เหมาะสมกับที่กระบวนการต้องการได้ การตอบสนองของตัวควบคุมจะอยู่ในรูปของเปลี่ยนแปลงตัวของตัวควบคุมจนถึงค่าความผิดพลาด ค่า Overshoots และค่าแกว่งของระบบ (Oscillation) วิธี PID ไม่รับประกันได้ว่าจะเป็นระบบควบคุมที่เหมาะสมที่สุดหรือสามารถทำให้กระบวนการมีความเสถียรแน่นอน การประยุกต์ใช้งานบางครั้งอาจใช้เพียงหนึ่งถึงสองรูปแบบ ขึ้นอยู่กับกระบวนการเป็นสำคัญ PID บางครั้งจะถูก เรียกว่าการควบคุมแบบ PI, PD, P หรือ I ขึ้นอยู่กับว่าใช้รูปแบบใด

2.7.2 การควบคุม PID แยกเดี่ยว [2]

การควบคุมแบบ PID แยกเดี่ยวนั้น เป็นวิธีการทั่วไปที่ใช้ในการควบคุมหุ่นยนต์ต่างๆ อธิบายกระบวนการทางคณิตศาสตร์ของการควบคุม PID โดย $r(t) \in \mathbb{R}^n$ แสดงถึงจุดอ้างอิงขาเข้า (Reference Input) ซึ่งเป็นตัวแปรข้อต่อที่เราจะต้องปฏิบัติตาม และให้ $e(t)$ แสดงผลต่างค่าความผิดพลาดที่เวลา t ดังนี้

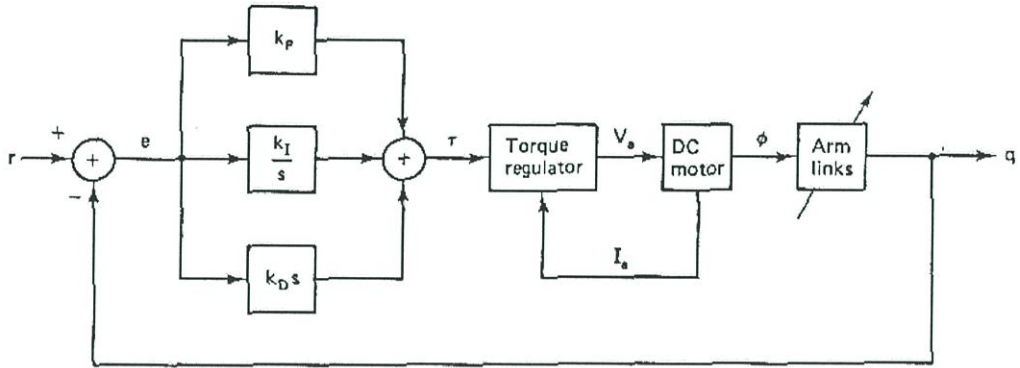
$$e(t) = r(t) - q(t) \quad (2.41)$$

ทางอุดมคติแล้วค่า $e(t)$ จะมีค่าเป็นศูนย์ แต่ในทางปฏิบัติแล้วมันไม่ได้เป็นศูนย์ โดยเฉพาะเมื่อค่า $r(t)$ เปลี่ยนอย่างรวดเร็ว เทคนิคทั่วไปนี้ใช้สำหรับการควบคุมแขนหุ่นยนต์ที่เป็น การควบคุมอิสระแยกเดี่ยว n ดังนี้

$$\tau(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \dot{e}(t) \quad (2.42)$$

ที่ $\{K_p, K_I, K_D\}$ เป็น เมตริกซ์ทแยงมุมขนาด $n \times n$ ซึ่งหมายถึงแต่ละแกนถูกควบคุมแยกออกจากกัน โดยกำหนดให้สัญญาณควบคุม PID ซึ่ง τ ประกอบด้วยทอม ผลความผิดพลาดของ

Proportional ผลความผิดพลาดของ Integral และผลความผิดพลาดของ Derivative ดังนั้น เมตริกซ์ที่แยงมุม $\{K_p, K_i, K_d\}$ ที่ออกแบบจะเรียกว่า Proportional Gain, Integral Gain และ Derivative Gain ตามลำดับ ดังแสดงในรูปที่ 2.18



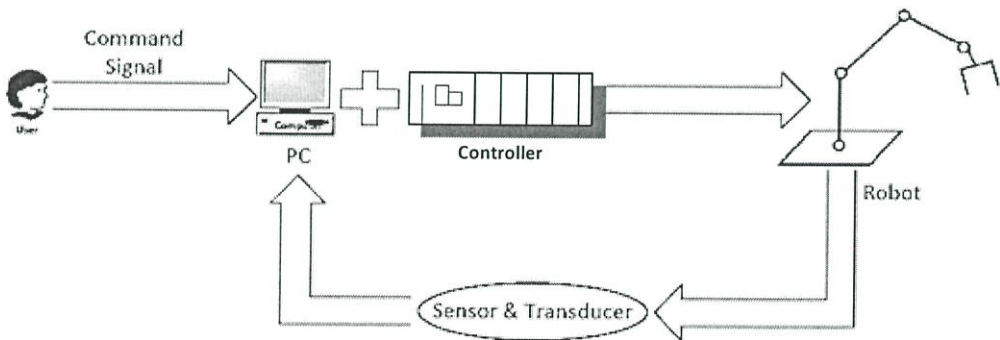
รูปที่ 2.18 ตัวควบคุม PID ของแกนเดียว

2.8 ระบบการควบคุมแขนหุ่นยนต์

2.8.1 หลักการในการควบคุม

ในการควบคุมระบบการเคลื่อนที่ของแขนหุ่นยนต์ให้ได้ตามต้องการนั้น การควบคุมการหมุนของมอเตอร์เป็นส่วนที่สำคัญที่สุด ซึ่งเป็นอุปกรณ์ที่ทำให้แขนหุ่นยนต์แต่ละข้อเคลื่อนที่ตามคำสั่งควบคุม

การควบคุมการทำงาน (การหมุน) ของมอเตอร์นั้นเป็นการควบคุมแบบดิจิทัล (Digital control) ซึ่งประกอบไปด้วย คอมพิวเตอร์ ชุดควบคุม มอเตอร์และเซ็นเซอร์ ดังบล็อกไดอะแกรมรูปที่ 2.19



รูปที่ 2.19 บล็อกไดอะแกรมโดยรวมของระบบควบคุมการทำงานของแขนหุ่นยนต์

2.8.2 โครงสร้างของระบบ

สามารถแบ่งโครงสร้างออกเป็น 3 ส่วนดังนี้

1. ส่วนตรวจจับ เป็นส่วนประกอบที่สำคัญในระบบที่มีการป้อนกลับ เช่น เซ็นเซอร์ และ เอนโคเดอร์ที่ใช้กันมากในระบบควบคุมเช่น Potentiometer เอนโคเดอร์แบบ Optical Encoder Resolver เป็นต้น โดยสัญญาณ Analog จากส่วนตรวจจับนี้จะถูกแปลงเป็นสัญญาณ Digital หลังจากนั้นจะถูกส่งไปยังหน่วยประมวลผลกลางต่อไป

2. หน่วยประมวลผลกลางและควบคุม ได้แก่ คอมพิวเตอร์ และตัวควบคุมที่สามารถป้อนซอฟต์แวร์ที่ออกแบบขึ้นเพื่อใช้ในระบบควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ โดยใช้ โมดูล I/O เป็นตัวเชื่อมต่อกับระบบบัสของคอมพิวเตอร์

3. วงจรขับเคลื่อนมอเตอร์ เป็นวงจรที่ควบคุมการหมุนของมอเตอร์ซึ่งเป็นอุปกรณ์ที่ทำให้แขนหุ่นยนต์เกิดการเคลื่อนไหว มอเตอร์จะถูกติดตั้งที่ข้อต่อต่างๆเพื่อควบคุมการเคลื่อนที่ของแขน

2.9 แทคโคเจเนอเรเตอร์ (Tacho Generator) [8]

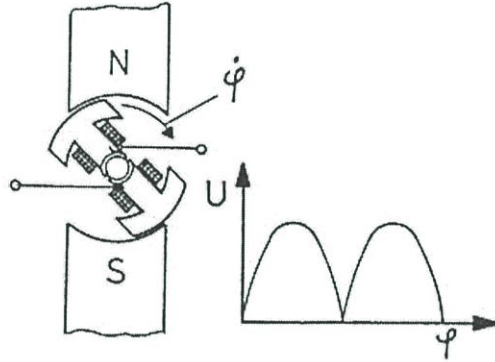
แทคโคเจเนอเรเตอร์ (Tacho Generator) เป็นอุปกรณ์เชิงกลสำหรับเปลี่ยนการหมุนเป็นสัญญาณทางไฟฟ้าที่นิยมใช้งาน เป็นอุปกรณ์เซ็นเซอร์แบบแอคทีฟ (Active) โดยอาศัยหลักการของแรงเคลื่อนไฟฟ้าเหนี่ยวนำในการทำงาน มีทั้งแบบที่ให้สัญญาณเอาต์พุตเป็นสัญญาณไฟฟ้ากระแสตรง และแบบที่เป็นสัญญาณไฟฟ้ากระแสสลับ



รูปที่ 2.20 แทคโคเจเนอเรเตอร์

หลักการการกำเนิดแรงเคลื่อนไฟฟ้าเหนี่ยวนำมีอุปกรณ์สำคัญ 2 ตัว คือ ขดลวด และ แม่เหล็กถาวร โดยถ้าขดลวดหมุนในสนามแม่เหล็กของแม่เหล็กถาวรครบ 1 รอบ จะได้สัญญาณไฟฟ้ารูปคลื่นไซน์ที่สมบูรณ์ 1 ลูก โดยความแรงของสัญญาณขึ้นอยู่กับความเร็วในการหมุนของขดลวด ยิ่งหมุนเร็วมากสัญญาณที่ได้ออกมาก็จะมีความแรงมาก ถ้าต่อแกนของมอเตอร์ที่ต้องการวัดรอบเข้ากับขดลวด ก็จะได้สัญญาณที่แทคโคเจเนอเรเตอร์เปลี่ยนแปลงตามความเร็วของการหมุนของมอเตอร์ตัวนั้นได้ไม่ยาก ส่วนแทคโคเจเนอเรเตอร์แบบไฟฟ้ากระแสตรง ก็มีโครงสร้างพื้นฐานเหมือนมอเตอร์

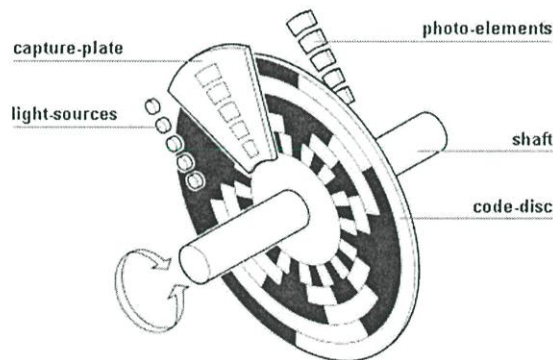
ไฟฟ้ากระแสตรง สามารถทดลองโดยนำมอเตอร์ในรถของเล่นมาป้อน LED ให้ติดสว่างได้ โดยความแรงของสัญญาณจะเปลี่ยนแปลงตามความเร็วของการหมุนเช่นเดียวกับแทคโคเจเนอเรเตอร์แบบกระแสสลับ



รูปที่ 2.21 หลักการทำงานของแทคโคเจเนอเรเตอร์

2.10 เอนโคเดอร์ (Encoder) [8]

เอนโคเดอร์เป็นอุปกรณ์ที่ทำหน้าที่เปลี่ยนสัญญาณเชิงกลในรูปแบบของการหมุนหรือการกระจัดมาเป็นสัญญาณทางไฟฟ้า โดยใช้หลักการทางแสง (Optical) การใช้งานส่วนใหญ่ถูกนำมาใช้เป็นอุปกรณ์นับรอบการหมุนของมอเตอร์หรืออุปกรณ์วัดองศาการหมุนของมอเตอร์ได้ สัญญาณทาง Output ของเอนโคเดอร์มีทั้งแบบที่เป็นสัญญาณ Digital และแบบที่เป็นสัญญาณ Analog

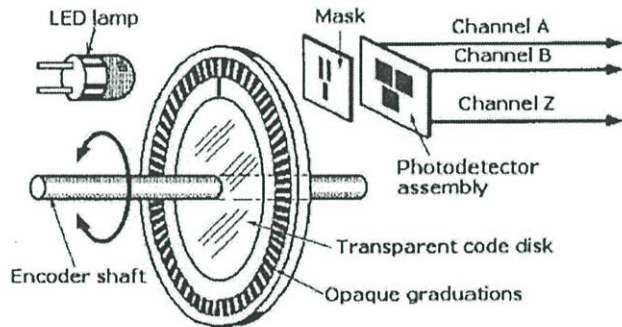


รูปที่ 2.22 ส่วนประกอบของเอนโคเดอร์

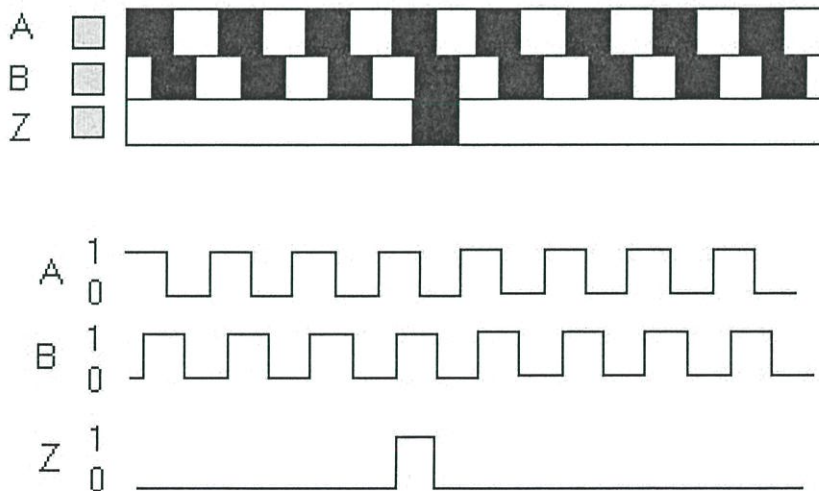
จากรูปที่ 2.22 จะเห็นว่าส่วนประกอบที่สำคัญของเอนโคเดอร์ประกอบด้วย ตัวส่งแสง ตัวรับแสง และจานรหัส โดยส่วนที่เป็นตัวส่งแสงและตัวรับแสงจะติดตั้งคนละด้านโดยมีจานรหัสเป็นตัวคั่นกลาง จานรหัสจะยึดกับมอเตอร์ที่เราต้องการวัดรอบ ลักษณะของจานรหัสจะมีส่วนที่เป็นช่องที่แสงผ่านได้และส่วนที่ทึบแสง เมื่อมอเตอร์หมุนจะทำให้จานรหัสหมุนตามไปด้วย ถ้าจานรหัสหมุน

ส่วนที่รับแสงจะไปกั้นลำแสงทำให้ Output ที่ส่วนรับแสงเป็นลอจิก 0 แต่ถ้าจนวนรหัสหมุนส่วนที่แสงผ่านตรงกับลำแสง ทำให้ตัวรับสามารถรับแสงได้ Output ที่ตัวรับแสงจะเป็นลอจิก 1 สลับกันเป็นเช่นนี้จนครบรอบ เอนโคเดอร์ขนาด 1 บิต มีรหัสเพียงหลักเดียวในการใช้งานจริง ตัวรับและตัวส่งแสงอาจมีมากกว่า 1 บิต เพื่อให้สามารถระบุตำแหน่งของงานได้ละเอียดขึ้น และตัวจนวนรหัสเองก็จะมีหลายขั้นขึ้นขึ้นสอดคล้องกับจำนวนบิตรหัสที่ได้ออกมาจะมีหลายรูปแบบแล้วแต่การออกแบบ โดยทั่วไปรหัสที่ใช้ ได้แก่ Binary code และ BCD code เป็นต้น

หลักการทํางานของเอนโคเดอร์ชนิด Incremental หลักการทํางานอาศัยหลักการ Optical บ่อยครั้งจึงถูกเรียกตามหลักการว่า Optical Encoder หรือบางกรณีก็จะถูกเรียกว่า Digital Encoder ซึ่งมีลักษณะโครงสร้างจะประกอบด้วยตัวกำเนิดแสง ตัวจับแสง ซึ่งถูกคั่นกลางด้วยแผ่นจานกลมๆ ที่มีการทำช่องเจาะไว้รอบๆแผ่น (จำนวนช่องจะขึ้นอยู่กับความละเอียดของเอนโคเดอร์) และหน้ากากแยกช่องของสัญญาณพัลส์ A, B และ Z



รูปที่ 2.23 หลักการทํางานของเอนโคเดอร์



รูปที่ 2.24 สัญญาณที่ออกจากเอนโคเดอร์

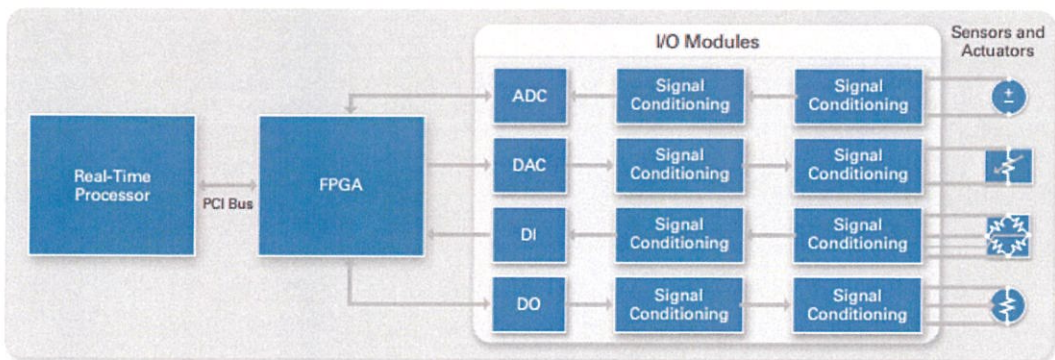
สัญญาณพัลส์ที่ได้จากเอนโคเดอร์ชนิดนี้จะประกอบด้วย 3 แทรค (Tracks) คือ A, B และ Z ดังรูปที่ 2.23 พัลส์ที่เกิดจาก แทรค A และ B จะเกิดการเลื่อนกันมีความต่างเฟสกัน 90 องศา เพื่อทำหน้าที่รายงานผลของความเร็วและทิศทางการหมุนของมอเตอร์ให้ตัวควบคุม ดังนี้

1. กรณีพัลส์ A เกิดขึ้นก่อน B ตัวควบคุมจะรู้ว่ามอเตอร์กำลังหมุนด้วยทิศทางตามเข็มนาฬิกา
2. กรณีพัลส์ B เกิดขึ้นก่อน A ตัวควบคุมจะรู้ว่ามอเตอร์กำลังหมุนด้วยทิศทางทวนเข็มนาฬิกา
3. ส่วนแทรค Z หรือพัลส์อ้างอิงจะเกิดขึ้นพัลส์ในการหมุน 1 รอบ ทำหน้าที่อ้างอิงตำแหน่งของโรเตอร์

Incremental Encoder โดยทั่วไปจะไม่นิยมใช้กับระบบเซอร์โวที่มีการควบคุมตำแหน่ง เนื่องจากไม่สามารถจำตำแหน่งเดิมได้กรณีที่มีการปิดเครื่องหรือไฟดับซึ่งจะต้องทำการหาจุดอ้างอิงใหม่ทุกครั้ง

2.11 CompactRIO [9]

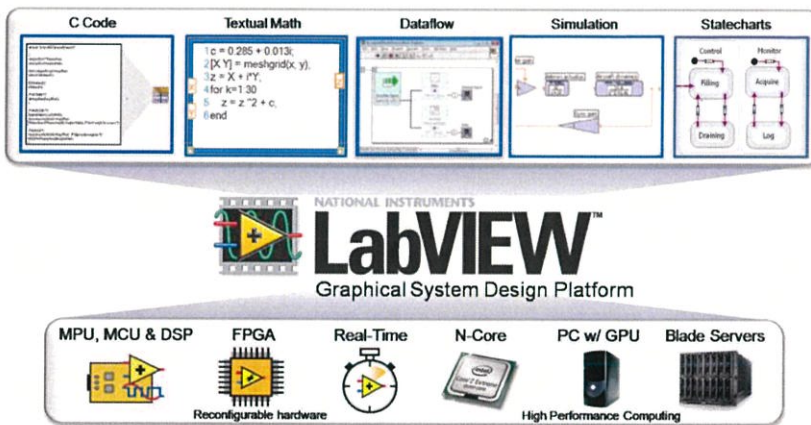
CompactRIO เป็นระบบฝังตัวที่สามารถปรับแต่ง ประกอบไปด้วย 3 ส่วน คือ การประมวลผล RTOS (ระบบปฏิบัติการ Real-Time) FPGA (Field Programmable Gate Array) และ โมดูล I/O ที่รับส่งข้อมูลกับเครื่องมือนัด โดยที่การประมวลผลแบบ Real-Time จะเน้นถึงความน่าเชื่อถือ การคาดคะเนพฤติกรรม การเก็บข้อมูลที่แม่นยำทางคณิตศาสตร์ และการวิเคราะห์ ในขณะที่ FPGA จะเน้นที่งานขนาดเล็กที่ต้องใช้ตรรกะที่มีความเร็วสูงและใช้เวลาที่แม่นยำ บางครั้งการใช้งาน CompactRIO จะใช้งานร่วมกับ HMI (Human Machine Interface) และ GUI (Graphic User Interface) เพื่อแสดงสถานะของระบบ และการตั้งค่าพารามิเตอร์ในการดำเนินงาน



รูปที่ 2.25 สถาปัตยกรรมของระบบฝังตัวที่ปรับแต่งได้

2.11.1 LabVIEW [10]

LabVIEW วัตถุประสงค์หลักเพื่อใช้ในการสร้างระบบอัตโนมัติในการวัด ทดสอบ และควบคุม โดยการใช้การเขียนโปรแกรมด้วยภาษาโค้ดรูปภาพ (Graphical Programming) ทำให้ง่ายต่อการเรียนรู้และใช้เวลาในการเขียนโปรแกรมน้อยลง ชุดซอฟต์แวร์สำหรับพัฒนาจะมาพร้อมกับชุดฟังก์ชันทางคณิตศาสตร์และวิศวกรรมศาสตร์สำหรับการวิเคราะห์ ประมวลผล และแสดงข้อมูลที่มีประสิทธิภาพ รวมถึงความสามารถต่างๆ ในการติดต่อใช้งานร่วมกับฮาร์ดแวร์และซอฟต์แวร์อื่นๆ ได้ดี ทำให้ LabVIEW เป็นที่สนใจและถูกนำไปใช้งานอย่างกว้างขวางในแทบทุกส่วนของอุตสาหกรรมและการวิจัยที่ต้องการของระบบอัตโนมัติในการวัดและการควบคุม อาทิเช่น ในอุตสาหกรรมการผลิต อิเล็กทรอนิกส์ ยานยนต์ การทหารและอากาศยาน พลังงาน การสื่อสาร เครื่องมือแพทย์



รูปที่ 2.26 โปรแกรม LabVIEW

LabVIEW เริ่มต้นจากการรันบนเครื่องแมคอินทอช โดยถูกวางตัวให้เป็นซอฟต์แวร์สำหรับการสร้างเครื่องมือเสมือน หรือ Virtual Instrument (VI) และใช้แนวคิดการเขียนโปรแกรมแบบกราฟิกเพื่อให้เรียนรู้และเข้าใจง่าย ส่วนวิธีการรันโปรแกรมใช้หลักการที่เรียกว่า Dataflow โดยฮาร์ดแวร์ที่ใช้กับ LabVIEW ในช่วงแรกมีเพียงบอร์ด GPIB ที่ใช้สำหรับติดต่อกับเครื่องมือวัดต่างๆ หลังจากนั้นก็เริ่มผลิตบอร์ด Data Acquisition แบบปลั๊กอินออกมาเพื่อให้ LabVIEW เป็นเครื่องวัดแบบ VI ได้ในตัวเองโดยไม่ต้องต่อพ่วงเครื่องมือวัดจากภายนอกก็เกิดฮาร์ดแวร์แพลตฟอร์มแบบใหม่ๆ ที่ใช้กับ LabVIEW อีกหลายอย่าง เช่นใน CompactRIO ซึ่งเป็นเทคโนโลยีที่สามารถนำไปใช้แก้โจทย์ปัญหาที่ซับซ้อนได้มากขึ้น

2.11.2 ตัวควบคุม Real-Time (Real-Time Controller) [9]

ตัวควบคุม Real-Time ประกอบด้วย การประมวลผลที่ต้องการความน่าเชื่อถือ และ กำหนดการทำงานใน LabVIEW การใช้งาน Real-Time และการควบคุมซับซ้อน การติดตามการดำเนินการ การบันทึกข้อมูลลงบอร์ด และการเชื่อมต่อกับอุปกรณ์อื่น ตัวเลือกนอกจากนี้ยังมี

- ไฟเลี้ยงสำรอง 9 VDC to 30 VDC
- Real-Time Clock
- Hardware Watchdog Timers
- พอร์ต Ethernet 2 พอร์ต
- พื้นที่การจัดเก็บข้อมูลมากกว่า 2 GB
- รองรับ USB และ RS232 ในตัว

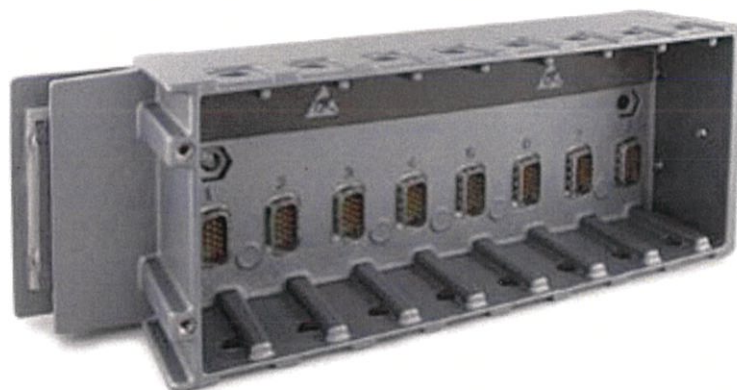


รูปที่ 2.27 ตัวควบคุม Real-Time NI cRIO-9082

2.11.3 FPGA (Field Programmable Gate Array) [9]

โครง FPGA I/O เป็นศูนย์กลางของสถาปัตยกรรมระบบฝังตัว ถูกเชื่อมต่อโดยตรงกับ I/O สำหรับการเข้าถึงประสิทธิภาพสูงของวงจร I/O แต่ละโมดูล รวมทั้งการ Timing การ Triggering และการ Synchronization เป็นเพราะว่าแต่ละโมดูลจะเชื่อมต่อโดยตรงกับ FPGA มากกว่าที่จะผ่านบัลลูนที่มีความล่าช้า โดยค่าเริ่มต้น FPGA นี้จะเชื่อมต่ออัตโนมัติกับโมดูล I/O และกำหนด I/O ที่จะประมวลผลแบบ Real-Time นอกจากนี้ FPGA ยังสามารถโปรแกรมบนตัวควบคุมในการเข้าถึงโมดูล I/O ใช้เวลาน้อยกว่า 500 ns. ของรอบระหว่างลูป หรือจะใช้การโปรแกรม FPGA เพื่อปรับแต่งระบบโดยตรงเพราะความเร็วของ FPGA ที่มักจะถูกใช้เพื่อสร้างระบบควบคุมที่แชร์ร่วมกับบัฟเฟอร์ I/O ความเร็วสูง ลูปควบคุมที่ใช้ความเร็ว หรือการกรองสัญญาณ ตัวอย่างเช่น การใช้ FPGA หนึ่งโครงที่สามารถดำเนินการลูปควบคุม PID ได้มากกว่า 20 ลูปพร้อมกันในอัตรา 100 kHz นอกจากนี้

เนื่องจาก FPGA รั้นโค้ดทั้งหมดในฮาร์ดแวร์ดังนั้นจึงมีความน่าเชื่อถือและการตัดสินใจที่สูงที่สุด ซึ่งเหมาะสำหรับ Hardware-Based Interlocks งานที่ต้องกำหนดใช้เวลาในการออกคำสั่ง เป็นต้น



รูปที่ 2.28 Reconfigurable FPGA Chassis

2.11.4 โมดูล I/O (I/O Modules) [9]

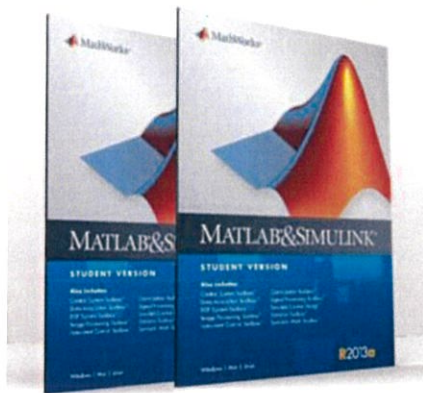
โมดูล I/O ประกอบด้วย การแยก วงจรการแปลง ตัวส่งสัญญาณ และสร้างการเชื่อมต่อส่งตรงไปยังอุปกรณ์ในอุตสาหกรรม เป็นการแสดงถึงความหลากหลายของการเดินสายไฟ และเพิ่มการต่อออกเป็นโมดูล ระบบ CompactRIO จะช่วยลดพื้นที่และค่าใช้จ่ายในการเดินสายไฟ โมดูล I/O C ซีรี่ส์มากกว่า 70 ตัวที่รองรับ CompactRIO เพื่อเชื่อมต่อกับอุปกรณ์อื่นๆ ชนิดของโมดูล



รูปที่ 2.29 โมดูล I/O C Series

2.12 MATLAB [11]

ชื่อของ MATLAB ย่อมาจาก Matrix Laboratory เป็นภาษาคอมพิวเตอร์ชั้นสูง (High-level Language) สำหรับการคำนวณทางเทคนิคที่ประกอบด้วย การคำนวณเชิงตัวเลข กราฟิกที่ซับซ้อน และการจำลองแบบเพื่อให้มองเห็นภาพพจน์ได้ง่ายและชัดเจน MATLAB มีการทำงานลักษณะเชิงโต้ตอบ (ลักษณะคล้ายเครื่องคิดเลข) มีความสามารถคำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็ว มากกว่าภาษาคอมพิวเตอร์สมัยก่อน อาทิเช่น ภาษา C C++ หรือ Fortran ทำให้เราสามารถคำนวณผลลัพธ์ พัฒนาอัลกอริทึม สร้างแบบจำลอง และแอปพลิเคชันได้ง่ายและรวดเร็วมาก เดิมโปรแกรม MATLAB ได้เขียนขึ้นเพื่อใช้ในการคำนวณทาง Matrix หรือเป็น Matrix software เพียงเท่านั้น MATLAB ได้พัฒนาโปรแกรมมาด้วยความสามารถแก้ปัญหาที่หลากหลายๆ ผู้ใช้ประสบพบเจอในการทำงาน เป็นระยะเวลาหลายปี จึงทำให้โปรแกรม MATLAB มีฟังก์ชันต่างๆ ให้เลือกใช้มากมาย



รูปที่ 2.30 โปรแกรม MATLAB

MATLAB สามารถนำไปประยุกต์ใช้ได้หลากหลายสาขาทั้ง การประมวลผลสัญญาณ (Signal Processing) การสื่อสาร (Communication) การประมวลผลภาพและวิดีโอ (Image and Video Processing) ระบบควบคุม (Control System) การวัดและควบคุม (Instrument and Control) การคำนวณทางเศรษฐศาสตร์ (Economic) การคำนวณทางชีววิทยา (Biology) และอื่นๆ ปัจจุบันในหลายๆมหาวิทยาลัยได้ใช้โปรแกรม MATLAB เป็นหลักสูตรโปรแกรมพื้นฐานในการศึกษาทางด้าน คณิตศาสตร์ วิศวกรรมศาสตร์ และวิทยาศาสตร์แขนงต่างๆ ตลอดจนด้านอุตสาหกรรมก็ได้นำโปรแกรม MATLAB มาเป็นเครื่องมือสำหรับใช้ในงานวิจัย พัฒนาและวิเคราะห์งานให้เกิดประโยชน์

2.13 RobotStudio [12]



Image courtesy of ABB

รูปที่ 2.31 โปรแกรม RobotStudio

RobotStudio เป็นโปรแกรม Offline Programming ทางเลือกที่ดีที่สุดเกี่ยวกับระบบหุ่นยนต์ โดยเป็นโปรแกรมจำลอง และเป็นโปรแกรมที่เป็น Offline Programming ของเอบีบีกรุ๊ป (ABB Limited) โดยที่โปรแกรมสามารถเขียนโปรแกรมการทำงานให้สำเร็จบนคอมพิวเตอร์ก่อนที่จะลงสู่การผลิตจริง ช่วยทำให้หุ่นยนต์เพิ่มผลกำไรแก่กระบวนการผลิต โดยภายในโปรแกรมจะมีเครื่องมือที่ช่วยดำเนินการต่างๆ อาทิเช่น การอบรม การเขียนโปรแกรม และเพิ่มประสิทธิภาพโดยไม่ต้องรบกวนการผลิต ซึ่งโปรแกรมนี้อาจช่วยลดความเสี่ยงของการทำงาน เริ่มการทำงานได้รวดเร็วขึ้น ใช้เวลาในการเปลี่ยนแปลงน้อยลง และเพิ่มผลผลิต

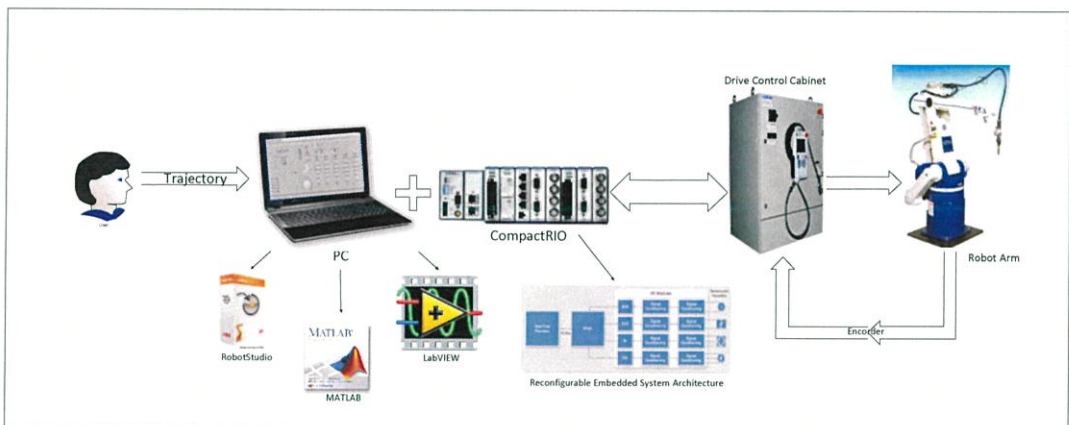
บทที่ 3

วิธีการดำเนินงาน

3.1 กล่าวนำ

ในบทนี้จะกล่าวถึงวิธีการดำเนินงานที่ประกอบด้วย การศึกษาและการออกแบบสมการตามทฤษฎี การปรับปรุง แก้ไข และพัฒนาส่วนต่างๆ การสร้างซอฟต์แวร์การควบคุมที่ทำให้เกิดเป็นระบบแขนหุ่นยนต์ขึ้น โดยมีจุดมุ่งหมายให้แขนหุ่นยนต์จะทำงานได้อย่างมีประสิทธิภาพ รวมทั้งการเลือกใช้โปรแกรมการสร้างเส้นทางการเคลื่อนที่ให้แก่ซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ โดยการทำงานจะแบ่งออกได้ดังต่อไปนี้

1. การออกแบบสมการ Forward และ Inverse Kinematics
2. การปรับปรุง แก้ไข ฮาร์ดแวร์ต่างๆ
3. การเลือกใช้โปรแกรมการสร้างเส้นทางการเคลื่อนที่
4. การออกแบบระบบคอมพิวเตอร์ซอฟต์แวร์

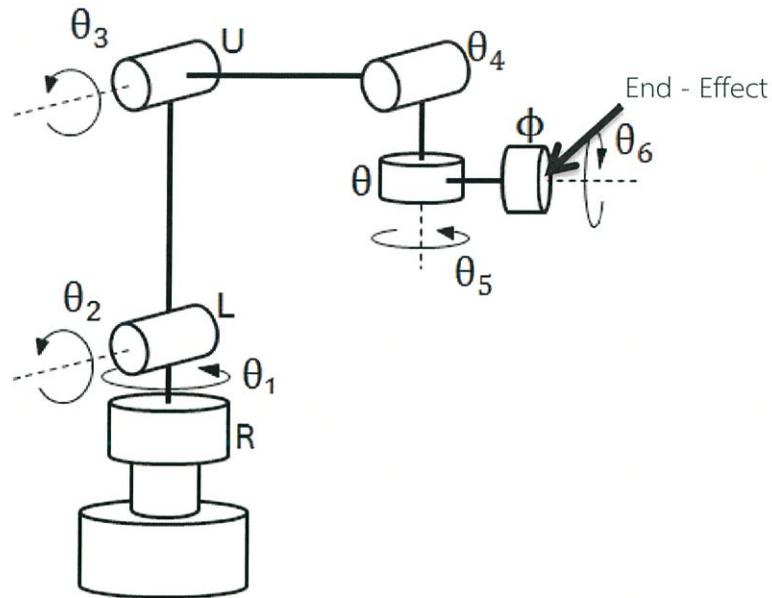


รูปที่ 3.1 ระบบการทำงานของระบบควบคุมแขนหุ่นยนต์

3.2 การศึกษาข้อมูลและค้นหาวิธีการสร้างสมการหา End - Effect

3.2.1 ศึกษาหาวิธีการสร้างสมการของแขนหุ่นยนต์

สมการแขนหุ่นยนต์นั้นมีความสำคัญเป็นอย่างยิ่งที่จะทำให้แขนหุ่นยนต์สามารถขยับได้ตรงกับความต้องการ เนื่องจากสมการนั้นสามารถบอกถึงพิกัดจุดที่ต้องการให้แขนหุ่นยนต์ทำงาน รวมทั้งบอกมุมของข้อต่อที่เปลี่ยนไป มุมข้อต่อของแต่ละข้อต่อและพิกัดจุดที่เปลี่ยนไปของแขนหุ่นยนต์นี้จะทำให้เกิดการเคลื่อนที่ จากที่กล่าวมา End - Effect เป็นจุดปลายสุดของแขนหุ่นยนต์ซึ่งเป็นจุดที่ใช้เชื่อมต่อ Tool ซึ่งเป็นจุดที่บอกพิกัดที่อ้างอิงที่ต้องการ โดยการที่จะทราบถึงตำแหน่งการเคลื่อนที่ มุมที่เกิดขึ้นได้นั้น จะต้องวิเคราะห์รายละเอียดของ Kinematics ของแขนหุ่นยนต์ ซึ่งแขนหุ่นยนต์จะมีสมการ Kinematics แตกต่างกันไปขึ้นอยู่กับขนาดความยาวแขน รูปแบบข้อต่อต่างๆ



รูปที่ 3.2 แบบจำลองแขนหุ่นยนต์ 6 แกน Almega LK

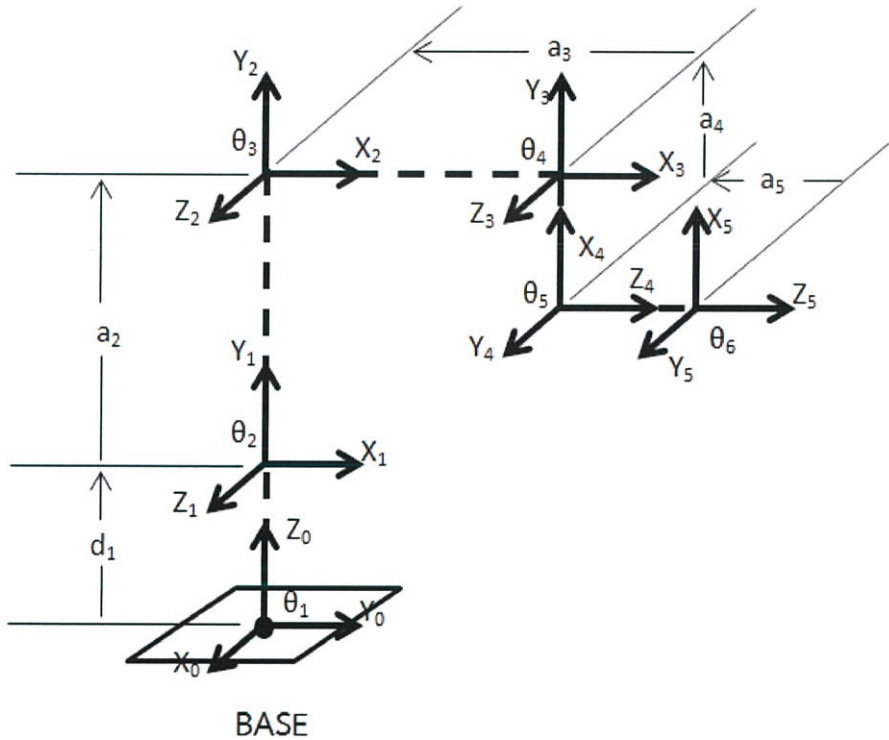
ความยาวของแขนหุ่นยนต์มีความยาวดังนี้

จากฐานถึงแกนหมุน L มีความยาวเท่ากับ	660 mm.
จากแกนหมุน L ถึงแกนหมุน R มีความยาวเท่ากับ	760 mm.
จากแกนหมุน R ถึงแกนหมุน θ_{Link} มีความยาวเท่ากับ	760 mm.
จากแกนหมุน θ_{Link} ถึงแกนหมุน θ มีความยาวเท่ากับ	200 mm.
จากแกนหมุน θ ถึงแกนหมุน ϕ มีความยาวเท่ากับ	120 mm.

3.2.2 การสร้าง Forward Kinematics

การสร้างสมการ Forward Kinematics เป็นการสร้างสมการจากการทราบถึงมุมของข้อต่อที่เกิดขึ้น เพื่อให้ง่ายในการสร้างสมการการเคลื่อนที่จึงนิยมทำให้สมการเป็นเมตริกซ์ เพราะเมตริกซ์สามารถสร้างสมการหลายตัวแปรได้ง่ายมากกว่าสมการแบบทั่วไป

การสร้างสมการจำเป็นต้องมีการทราบ และกำหนดค่าตัวแปรพารามิเตอร์ต่างๆ ที่ใช้ในการสร้างสมการเมตริกซ์ ซึ่งมีค่า a , d , θ และ α ซึ่งเป็นค่าความยาวข้อต่อ ค่าระยะข้อต่อ มุมข้อต่อ และมุมบิด ตามลำดับ โดยจะมีเพียงค่า θ เป็นตัวแปรนอกจากนั้นจะเป็นค่าคงที่ทั้งหมด



รูปที่ 3.3 การตั้ง Link Coordinates ของแขนหุ่นยนต์

ตารางที่ 3.1 ค่า Kinematics Parameters

Axis	θ	d	a	α	Home
R	θ_1	660 mm.	0	-90°	0°
L	θ_2	0	760 mm.	0	90°
U	θ_3	0	760 mm.	0	120°
θ_{Link}	θ_4	0	200 mm.	0	-
θ	θ_5	0	120 mm.	90°	0°
ϕ	θ_6	0	0	0	-

การสร้างสมการ Forward และ Inverse Kinematics มีความสัมพันธ์กันด้วยการนำพิกัดจุดปลายที่ต้องการให้แขนหุ่นยนต์เคลื่อนไปให้ Inverse Kinematics ช่วยในการหามุมที่เกิดขึ้นของแขน แล้วนำมุมของแต่ละข้อที่ได้กลับมาเข้าสมการ Forward Kinematics เพื่อเป็นคำสั่งให้แขนหุ่นยนต์มีการเคลื่อนที่ได้อย่างถูกต้อง จากความสัมพันธ์นี้จึงเกิดเป็นการเคลื่อนที่ตามเส้นทางของแขนหุ่นยนต์

3.3 ฮาร์ดแวร์

ในหัวข้อนี้จะกล่าวถึงอุปกรณ์ชิ้นส่วนที่มีอยู่กับตัวแขนหุ่นยนต์ และอุปกรณ์ชิ้นส่วนต่างๆที่เพิ่มเติมนอกเหนือที่แขนหุ่นยนต์มีอยู่แล้ว อุปกรณ์ที่ติดตั้งในตัวควบคุม และวงจรเพิ่มเติมที่ใช้ตัวควบคุม เนื่องจากแขนหุ่นยนต์ และตัวควบคุมที่นำมาใช้ในการพัฒนาปรุ้งซอฟต์แวร์นั้นไม่ได้ออกแบบสร้างขึ้นใหม่ แต่เป็นการนำแขนหุ่นยนต์ที่เคยใช้งานจริงในอุตสาหกรรมมาปรับปรุง ดังนั้นอุปกรณ์บางอย่างติดอยู่กับแขนหุ่นยนต์นั้นจึงเป็นอุปกรณ์ที่ทางบริษัทได้ออกแบบและปรับปรุงให้มีความเหมาะสมอยู่แล้วจึงไม่ได้ทำการแก้ไขปรับปรุง

3.3.1 การศึกษาและปรับปรุงแขนหุ่นยนต์

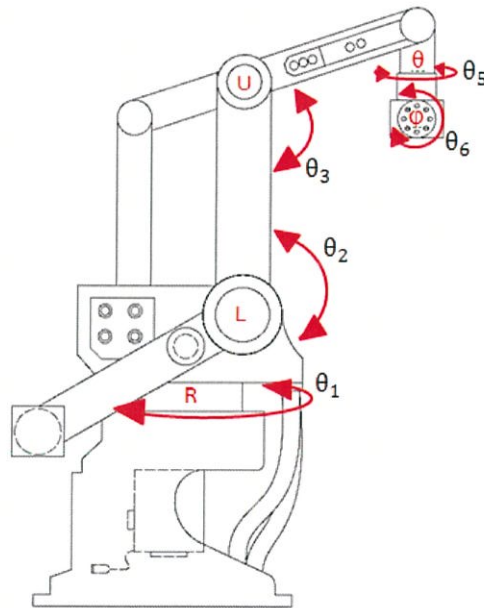
ในหัวข้อนี้ได้นำแขนหุ่นยนต์มาพัฒนาและปรับปรุงซอฟต์แวร์ระบบที่ใช้ในการควบคุมการเคลื่อนที่ใหม่ โดยที่แขนหุ่นยนต์ที่นำมาปรับปรุงนี้เป็นแขนหุ่นยนต์ของ OTC Daihen Inc. แขนหุ่นยนต์รุ่น Almaga LK ผลิตในช่วงปี 1984 – 1988 โดยที่โครงสร้างทางกายภาพของแขนหุ่นยนต์รุ่น Almaga LK เป็นมอเตอร์ที่ฐาน แกนหมุนทั้งหมด 5 แกน และ 1 แกนที่มีลักษณะตั้งฉากกับพิกัดฐานอ้างอิงเสมอไม่สามารถปรับได้



รูปที่ 3.5 แขนหุ่นยนต์รุ่น Almaga LK

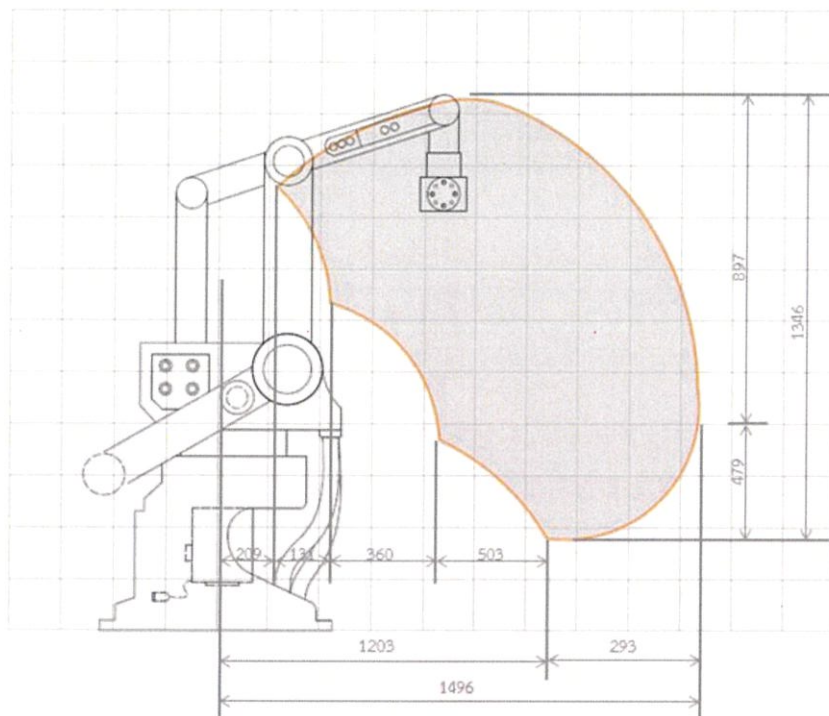
3.3.1.1 ลักษณะของแขนหุ่นยนต์

1. R (Rotation) เป็นข้อหมุนที่ล่างสุดของแขนหุ่นยนต์ติดอยู่กับฐานของแขนหุ่นยนต์ที่ตั้งกับพื้น ข้อส่วนนี้จะหมุนรอบตัวเองในทางตั้งฉากกับโครงอ้างอิงฐาน
2. L (Lower) เป็นข้อต่อแขนที่ติดอยู่ระหว่างข้อ R กับแขนท่อนล่างของแขนหุ่นยนต์ เปรียบเสมือนข้อยกตรงหัวไหล่ ทำหน้าที่ยกส่วนที่สำคัญสำหรับเคลื่อนไหวยังทั้งหมดขึ้น
3. U (Upper) เป็นข้อต่อที่ติดอยู่ระหว่างแขนท่อนล่าง และแขนท่อนบนของแขนหุ่นยนต์ เปรียบเทียบเป็นส่วนข้อศอก ทำหน้าที่ยกแขนท่อนบน และยก Tool ขึ้น
4. θ_{Link} (Zeta Link) เป็นข้อต่อที่ไม่สามารถควบคุมได้ โดยข้อต่อนี้จะทำมุมแขนที่เชื่อมต่อไปยังข้อต่อ θ มีลักษณะตั้งฉากกับฐานอ้างอิง
5. θ (Zeta) เป็นข้อต่อส่วนต่อจากปลายแขนท่อนที่ขนานกับฐานของตัวแขนหุ่นยนต์ตลอดเวลา โดยที่ข้อต่อ θ ทำหน้าที่หมุน Tool
6. ϕ (Phi) เป็นข้อต่อสุดท้ายที่มีจุดปลายเป็น End - Effect ซึ่งเป็นส่วนที่ติดกับ Tool โดยตรง

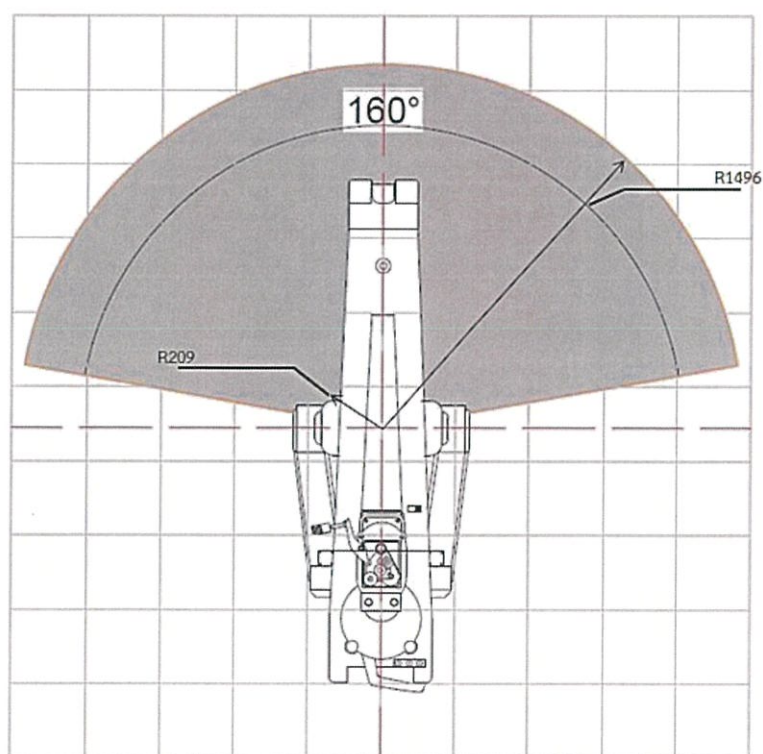


รูปที่ 3.6 ตำแหน่งข้อต่อและการหมุนของแขนหุ่นยนต์ในแบบสองมิติ

จากรูปที่ 3.6 ที่แสดงตำแหน่งข้อต่อแขน และแนวการหมุนของแขนหุ่นยนต์ โดยที่พื้นที่ความสามารถของการเคลื่อนที่แขนหุ่นยนต์นั้นจะขึ้นอยู่กับความกว้างของมุมที่แต่ละข้อสามารถหมุนเคลื่อนที่ไปได้ ณ ตำแหน่งต่างดังต่อไปนี้



รูปที่ 3.7 พื้นที่ความสามารถของแขนหุ่นยนต์มุมมองด้านข้าง

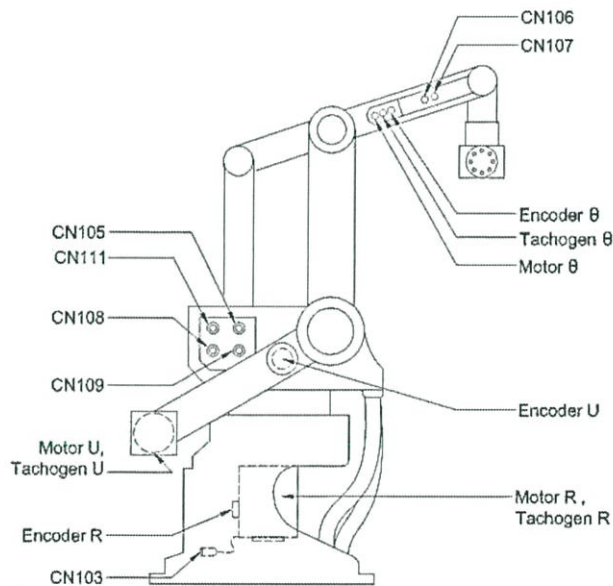


รูปที่ 3.8 พื้นที่ความสามารถของแขนหุ่นยนต์มุมมองด้านบน

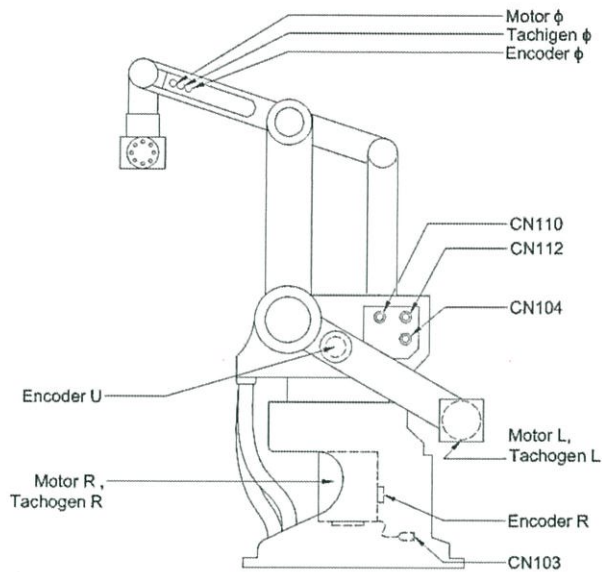
จากรูปที่ 3.7 แสดงพื้นที่ความสามารถของแขนหุ่นยนต์มุมมองด้านข้างนั้น เห็นได้ว่า จะมีบางช่วงที่แขน L และ U ไม่สามารถทำงานได้เพียงข้อเดียวจะต้องปรับมุมของข้อต่อแขนให้มีความเหมาะสมกัน ถึงจะทำให้แขนสามารถเคลื่อนที่ไปได้ จึงทำให้เกิดรัศมีที่แตกต่างกัน ภายในการเคลื่อนที่ที่ใกล้แขนหุ่นยนต์มากที่สุด และจากรูป 3.8 แสดงพื้นที่การทำงานของแขนหุ่นยนต์ด้านบนรัศมีการหมุนของ R โดยภายในรูปหน่วยความยาวที่วัดมีหน่วยเป็นมิลลิเมตร

3.3.1.2 อุปกรณ์ต่างๆที่อยู่ในแขนหุ่นยนต์

ในหัวข้อนี้ได้กล่าวถึงอุปกรณ์ที่ติดตั้งในส่วนต่างๆ อาทิเช่น อุปกรณ์ขับเคลื่อน เซ็นเซอร์ และตัวตรวจวัดต่างๆ ที่ถูกติดตั้งภายในแขนหุ่นยนต์ รวมไปถึงตำแหน่งต่างๆ และจุดเชื่อมต่อ จุดเดินสายไฟที่ถูกติดตั้งภายในแขนหุ่นยนต์ จึงได้ทำการศึกษาเพื่อตรวจสอบสภาพของตัวอุปกรณ์ สภาพการใช้งาน เพื่อประเมินคุณภาพของอุปกรณ์นี้ว่ายังสามารถใช้งานได้อยู่หรือไม่ หรืออุปกรณ์ใดภายในแขนหุ่นยนต์ที่ต้องมีการปรับเปลี่ยน หรือแปลงสภาพ เพื่อความเหมาะสมกับการดำเนินงานต่อไป



รูปที่ 3.9 ส่วนประกอบต่างๆ ของแขนหุ่นยนต์มุมมองด้านซ้าย



รูปที่ 3.10 ส่วนประกอบต่างๆ ของแขนหุ่นยนต์มุมมองด้านขวา

จากรูปที่ 3.9 และรูปที่ 3.10 จะเห็นว่าภายในแขนหุ่นยนต์จะมีอุปกรณ์หลักสำหรับควบคุม เช่น เซอร์ติฟิเคชันสภาวะการทำงาน ตัวเชื่อมต่อการเดินสายต่างๆ และกลไกของแขนหุ่นยนต์ โดยอุปกรณ์ที่ในการใช้งาน แสดงดังตารางที่ 3.2

ตารางที่ 3.2 รายการอุปกรณ์ที่ติดตั้งภายในแขนหุ่นยนต์

อุปกรณ์	จำนวน
มอเตอร์ DC	5 ตัว
เอนโคเดอร์	5 ตัว
ลิมิตสวิตช์	8 ตัว
พรีอกซิมีตี้สวิตช์	5 ตัว
เบรก	3 ตัว

3.3.1.3 แบบการเดินสายของแขนหุ่นยนต์กับอุปกรณ์ต่างๆ

จากหัวข้อข้างต้นที่กล่าวถึงอุปกรณ์ต่างๆที่ติดตั้งอยู่ภายในแขนหุ่นยนต์ ในหัวข้อนี้จะกล่าวถึงการเดินสายเพื่อเชื่อมต่ออุปกรณ์ภายในแขนหุ่นยนต์ และการเดินสายจากแขนหุ่นยนต์เพื่อเข้าสู่ตัวควบคุม โดยการเดินสายภายในแขนหุ่นยนต์จะมี 2 คอนเนกเตอร์ด้วยกัน คือ CN101 และ CN102 ตามลำดับ ซึ่ง CN101 จะเป็นการเดินสายส่งกำลังไปยังมอเตอร์ พรีอกซิมีตี้สวิตช์ (Proximity Switch) ลิมิตสวิตช์ (Limit Switch) และเบรก (Break) ที่อยู่ภายในแขนหุ่นยนต์ ดังตารางที่ 3.3 และรูปที่ 3.11 ส่วนอีกคอนเนกเตอร์ คือ CN102 เป็นการเดินสายเซ็นเซอร์ต่าง คือ เอนโคเดอร์ และแทคโคเจเนอเรเตอร์ ดังตารางที่ 3.4 และรูปที่ 3.12 โดยการทำงานของอุปกรณ์

ทั้งหมดนี้จะขึ้นอยู่กับระบบควบคุมที่ควบคุมการทำงานของวงจรขับเคลื่อนภายในตู้ควบคุม วงจรการเดินไฟของอุปกรณ์ต่างๆในแขนหุ่นยนต์ มีดังนี้

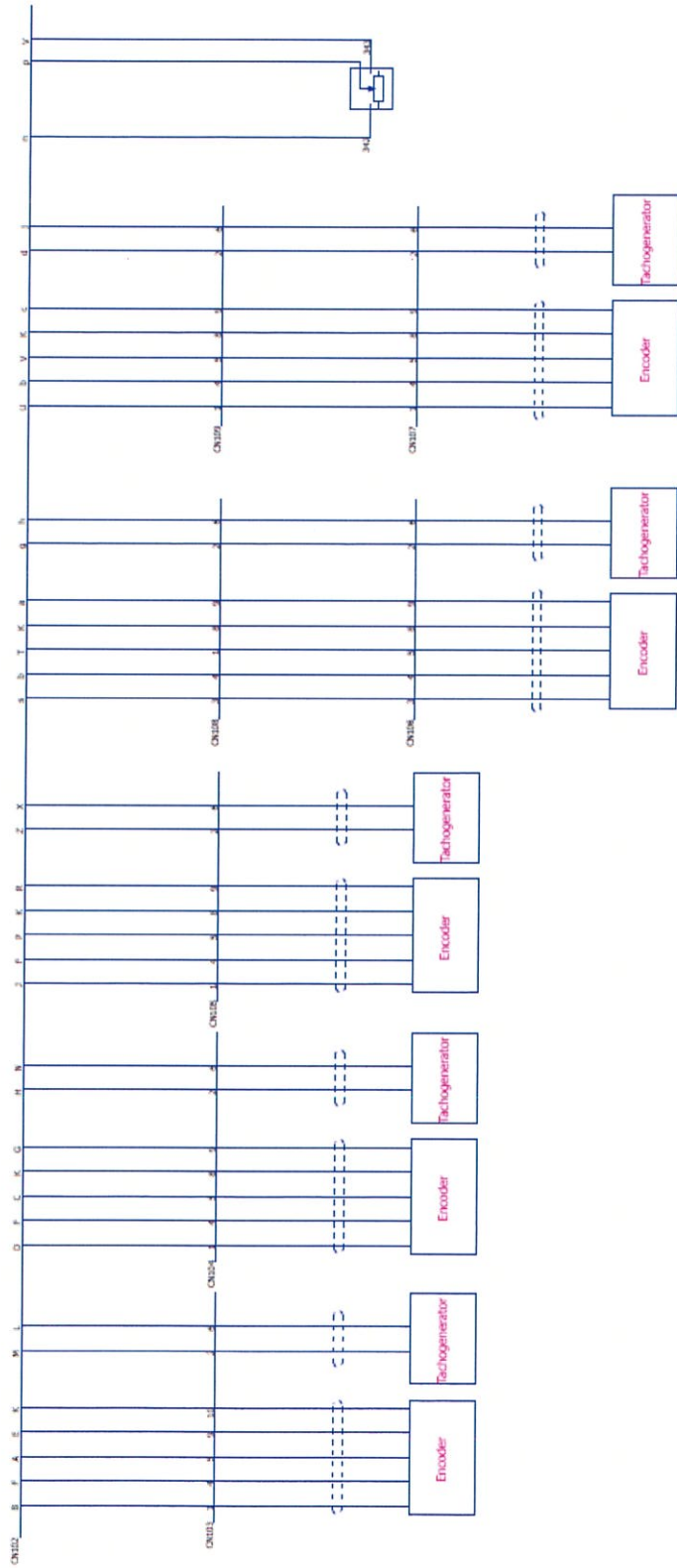
ตารางที่ 3.3 การเดินสายจากคอนเนกเตอร์ CN101 ไปอุปกรณ์ภายในแขนหุ่นยนต์

NO.	COMMENT	Robot Arm						REMARK
		WIRE	TERMINAL	WIRE	CONNECTOR	WIRE	CONNECTOR	
		NO.		NO.	NAME	NO.	NAME	
1	Rotation Motor	330		330		A	CN101	-
		320		320		B		+
2	Rotation Break	102		102		h	CN101	
		104		104		g		
3	Lower Motor	430		1	CN110	C	CN101	-
		420		2		D		+
4	Lower Break	102		4	CN110	N	CN101	
		104		8		d		
5	Upper Motor	530		1	CN111	E	CN101	-
		520		2		F		+
6	Upper Break	102		4	CN111	N	CN101	
		104		8		d		
7	Zeta Motor	630		1	CN112	G	CN101	-
		620		2		H		+
8	Zeta Break	103		7	CN112	e	CN101	
		102		8		N		
9	Phi Motor	730		3	CN112	J	CN101	-
		220		5		K		+
10	Limit Switch θ (P1)	161		12	CN112	V,b	CN101	
		160		159		P2		-
11	Limit Switch θ (P2)	159		14	CN112	f	CN101	
		160		161		P1		-
12	Proximity Switch θ	614		15	CN112	T	CN101	
		101		13		j		
13	Proximity Switch ϕ	714		16	CN112	U	CN101	
		101		13		j		
14	Limit Switch Upper (UL)	157		9	CN112	z	CN101	
		158		4		a		
15	Limit Switch Upper (UR)	159		11	CN112	f	CN101	
		158		4		a		
16	Limit Switch Rotation (RR)	151				W	CN101	
		152		153		RL		-
17	Limit Switch Rotation (RL)	153		16	CN112		CN101	
		101		13		j		
18	Limit Switch Lower (L1)	151				W	CN101	
						L2		-
19	Limit Switch Lower (L2)	159				f	CN101	
						L1		-

ตารางที่ 3.4 การเดินสายจากคอนเนกเตอร์ CN102 ไปอุปกรณ์ภายในแขนหุ่นยนต์

NO.	COMMENT	Robot Arm						REMARK	
			WIRE	CONNECTOR	WIRE	CONNECTOR	WIRE		CONNECTOR
			NO.	NAME	NO.	NAME	NO.		NAME
1	Rotation Encoder		1	CN103			B	CN102	A
			4				F		GND
			5				A		B
			9				E		Z
			10				K		V+
2	Rotation Tachogenerator		2	CN103			M	CN102	+
			6				L		-
			3				-		SHIELD
3	Lower Encoder		1	CN104			D	CN102	A
			4				F		GND
			5				C		B
			8				K		V+
			9				G		Z
4	Lower Tachogenerator		6	CN104			N	CN102	+
			2				H		-
			3				-		SHIELD
5	Upper Encoder		1	CN105			A	CN102	A
			4				F		GND
			5				P		B
			8				K		V+
			9				R		Z
6	Upper Tachogenerator		2	CN105			Z	CN102	+
			6				X		-
			3				-		SHIELD
7	Zeta Encoder		3	CN106	3	CN108	s	CN102	B
			4		4		b		GND
			5		1		T		A
			8		8		K		V+
			9		9		a		Z
8	Zeta Tachogenerator		6	CN106	6	CN108	h	CN102	+
			2		2		g		-
9	Phi Encoder		1	CN107	1	CN109	U	CN102	A
			4		4		b		GND
			5		5		V		B
			8		8		K		V+
			9		9		c		Z
10	Phi Tachogenerator		6	CN107	6	CN109	j	CN102	+
			2		2		d		-
11	Adjust Resistor		342				n	CN102	
			343				p		
			344				v		

CN102 SENSOR WIRING



รูปที่ 3.12 Wiring Diagram ของคอนเนกเตอร์ CN102

Drive Motor	+
Encoder	+
Tachogenerator	+
CN102	+

3.3.1.4 การแก้ไขปรับปรุงแขนหุ่นยนต์ทั้งหมด

เนื่องจากปัญหานิวรณ์ฉบับนี้เป็นการปรับปรุงแขนหุ่นยนต์ทางอุตสาหกรรมที่เคยมีการนำมาใช้ก่อนแล้ว พบว่าอุปกรณ์ต่างๆ ที่ใช้ภายในแขนหุ่นยนต์เป็นอุปกรณ์ที่มีคุณภาพส่วนมากยังสามารถใช้งานได้ อุปกรณ์ วัสดุทางกลใช้สามารถใช้งานได้อยู่ แต่บางส่วนก็มีการเสียหายเกิดจากการใช้งานจากบุคคลก่อน จากการปล่อยทิ้งไว้ไม่ได้นำมาใช้เป็นเวลานาน ซึ่งพบและทำการแก้ไข ดังนี้

1. เบรกคลัชต์เมื่อให้สัญญาณไม่สามารถปลดเบรกได้ เนื่องมาจากอุปกรณ์ที่ไม่ถูกใช้เป็นเวลานาน ทำให้จานเบรกเกิดสนิมขึ้น เมื่อปลดเบรกแล้วเบรกไม่หลุดทำให้ข้อต่อแกนหมุน R ไม่สามารถเคลื่อนที่ได้ เนื่องจากความเร็วที่ใช้ในการทดลองนี้สามารถจำกัดความเร็วที่เกิดขึ้นได้ แกน R จึงหมุนช้าในระดับที่ค่อนข้างมีความปลอดภัย จึงสามารถทำการแก้ไขปัญหาได้ด้วยวิธีการถอดนอตที่ยึดเบรกรอกให้ตัวเบรกหมุนติดไปกับแกนหมุน R ออกจึงทำให้แกนหมุนสามารถหมุนได้ตลอดเวลา

2. เอนโคเดอร์ทำงานไม่เสถียร เกิดจากมีละอองน้ำมันติดบนแผ่นทำให้การอ่านค่าจากเอนโคเดอร์ไม่ตรงกับมุมข้อต่อที่เปลี่ยนไป จึงได้ทำการถอดแล้วนำมาเช็ดให้สะอาดเพื่อให้ตัวเอนโคเดอร์นับค่าได้อย่างถูกต้องทำให้มุมข้อต่อมีความผิดพลาดน้อยที่สุด

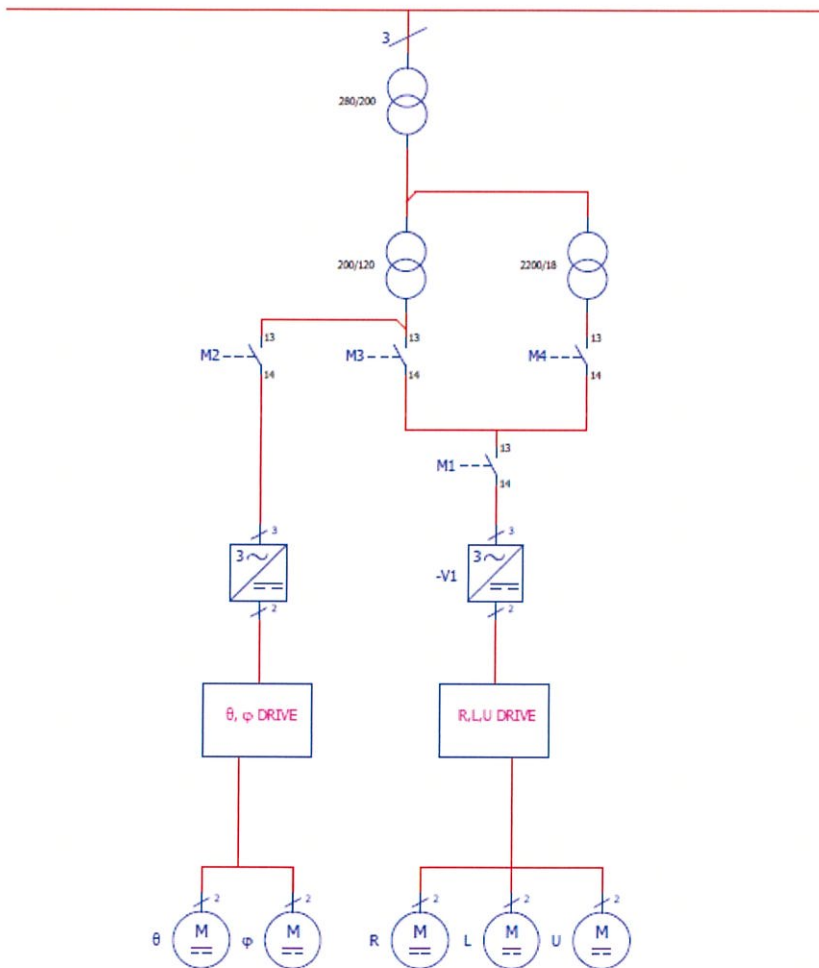
3. การเคลื่อนที่ของข้อต่อแกน L มีผลทำให้ข้อต่อของแกน U หมุนตาม และเคลื่อนที่ของข้อต่อแกน U หมุนเพียงข้อเดียว จะทำให้ข้อต่อแกน θ หมุนตาม การโดยที่ค่าเอนโคเดอร์ไม่เปลี่ยนตาม ทำให้เอนโคเดอร์นับค่าผิดพลาดทำงานไม่ตรงกับความต้องการ จึงได้ทำการแก้ไขปัญหาดังกล่าวด้วยการแก้ไขในซอฟต์แวร์ เพราะเกิดขึ้นโดยเป็นทางแมคคานิกส์ของแขนหุ่นยนต์

4. จากการเคลื่อนย้ายทำให้คอนเนกเตอร์ CN101 ทำให้ขา GND ของลิมิตสวิทช์ไปเชื่อมต่อกับขาที่มีแรงดันไฟฟ้า ทำให้เกิดการลัดวงจร จึงทำการแก้ไขโดยการถอดสายไฟที่ลัดวงจรออก เพื่อป้องกันการลัดวงจร

3.3.2 ตู้ควบคุมที่ใช้ในการปรับปรุงซอฟต์แวร์

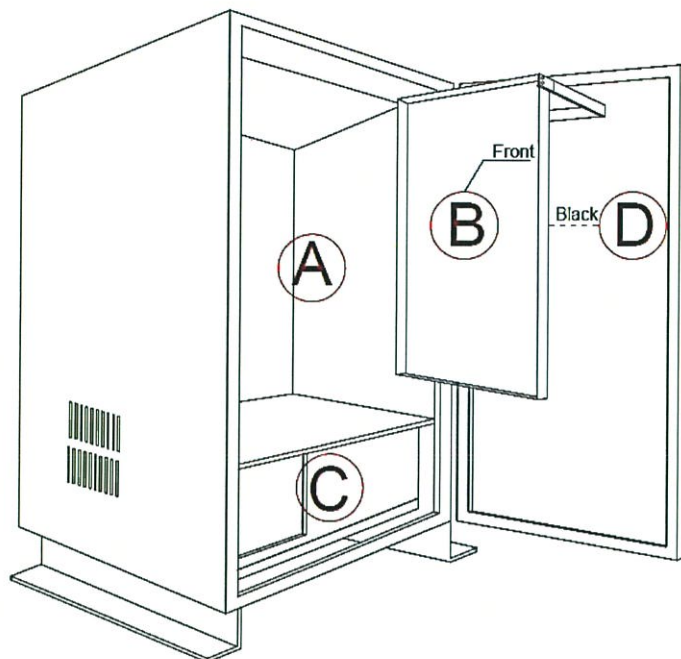
หลังจากทราบการเชื่อมต่อของอุปกรณ์แล้ว ในหัวข้อนี้จะกล่าวถึงส่วนประกอบที่ใช้ในการทำงานของตู้ควบคุมของแขนหุ่นยนต์ ตู้ควบคุมของแขนหุ่นยนต์รุ่นนี้ คือ OSACOM 5000 แรงดันไฟฟ้าที่ใช้ภายในตู้ควบคุมนี้รับไฟฟ้ากระแสสลับ 3 เฟส แรงดัน 200 โวลต์ (200 Vac Three – Phase Electric Power) มีส่วนที่เป็นภาคไฟฟ้ากำลังที่ส่งไฟฟ้าไปควบคุมส่วนต่างๆ ภายในตู้ควบคุมกับแขนหุ่นยนต์ และส่วนที่เป็นการควบคุมการทำงานของแขนหุ่นยนต์

ในปริญญานิพนธ์นี้จะกล่าวถึงส่วนประกอบหลักๆของตู้ควบคุมที่นำมาใช้งานเพียงเท่านั้น ซึ่งเป็นส่วนประกอบที่สำคัญที่ทำให้แขนหุ่นยนต์เคลื่อนที่ได้ แบบการทำงานของตู้ควบคุมนี้จะมีรูปแบบง่ายๆ ดังรูปที่ 3.13

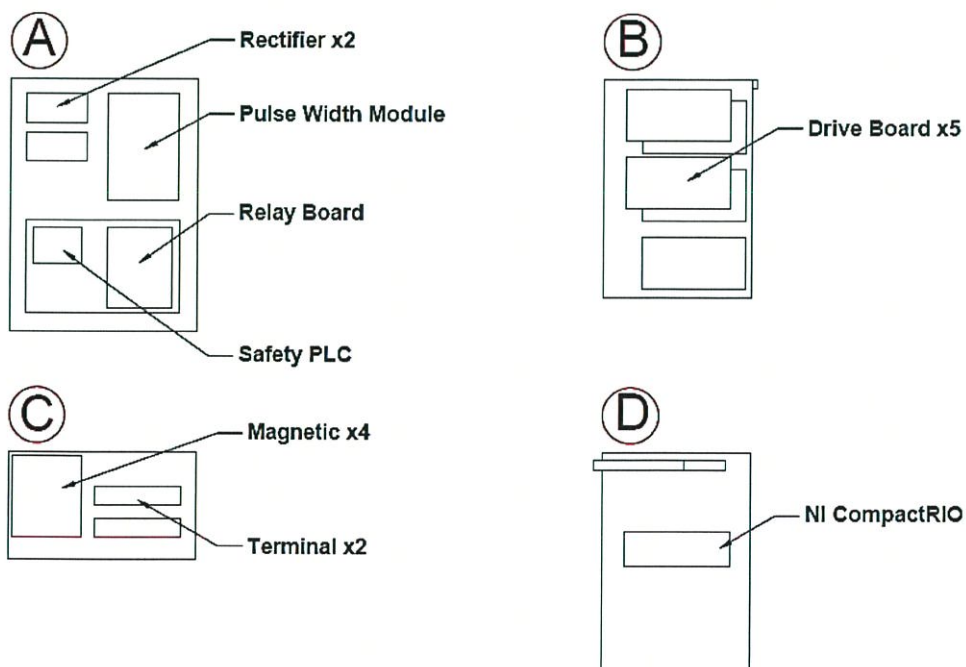


รูปที่ 3.13 วงจรไฟฟ้ากำลังภายในตู้ควบคุม

ภายในของตู้ควบคุมแขนหุ่นยนต์ จะมีส่วนประกอบต่างๆ ที่ติดตั้งอยู่ภายใน แบ่งออก 4 ส่วนตามรูปที่ 3.15 ดังนี้



รูปที่ 3.14 ตู้ควบคุม OSACOM 5000

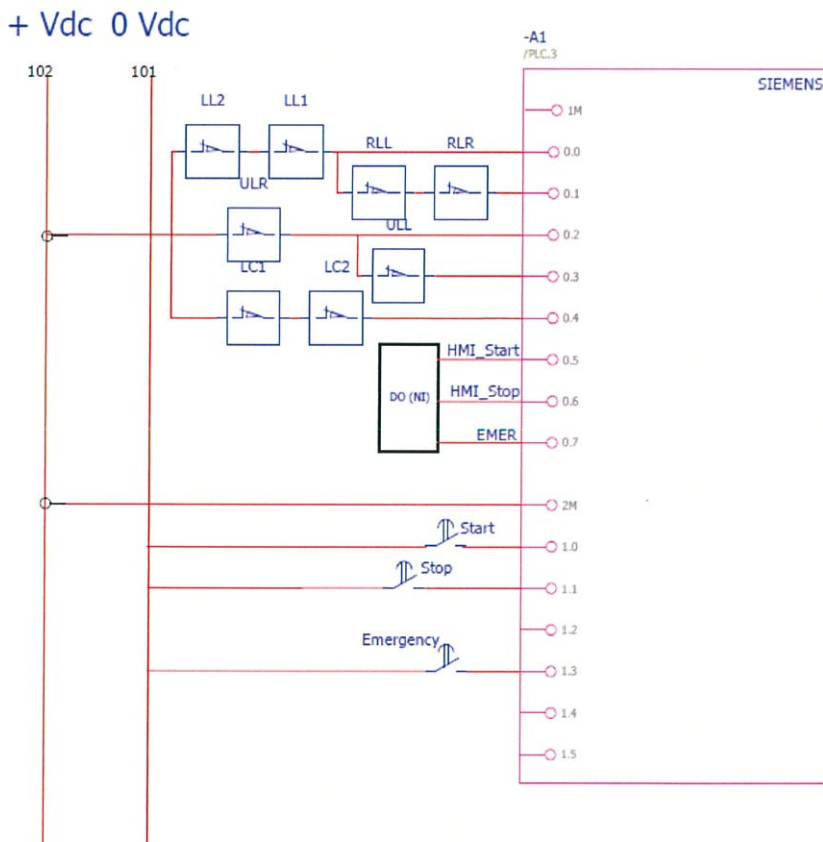


รูปที่ 3.15 ส่วนประกอบต่างๆภายในตู้ควบคุมที่ใช้งาน

3.3.2.1 ส่วน A

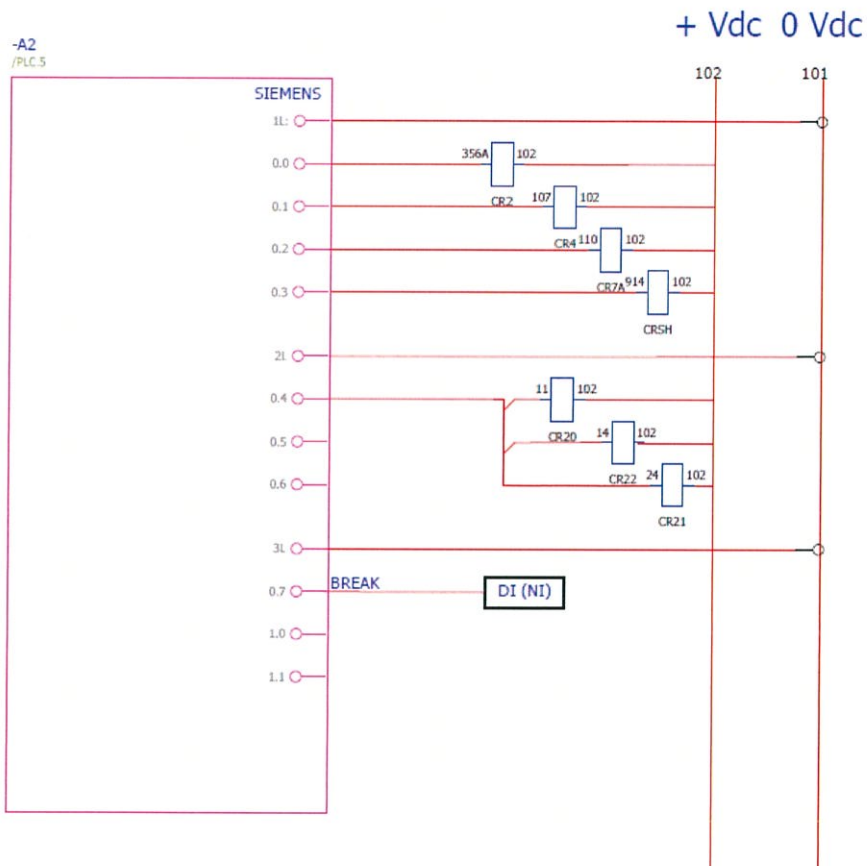
ส่วน A ประกอบด้วย วงจรเรียงกระแส (Rectifier) ที่รับแรงดันไฟฟ้ากระแสสลับมาจัดให้ใช้งานภายในตู้ควบคุมทั้งหมด Pulse Width Module เป็นอุปกรณ์ที่ใช้หลักการ PWM (Pulse Width Modulation) เป็นส่วนการสร้างพัลส์ขนาดต่างๆ ส่งไปให้กับวงจรขับเคลื่อนมอเตอร์ แผงวงจรรีเลย์ และ PLC เป็นส่วนควบคุมการใช้งานอุปกรณ์ต่างๆภายในตู้ควบคุม เช่น แมกเนติกคอนแทคเตอร์ สวิตช์เปิด-ปิด เป็นต้น และเทอร์มินอลที่เป็นจุดรวมสายไฟทั้งหมดในตู้ควบคุมที่ออกไปใช้งานส่วนต่างๆ ทั้งใช้ภายในตู้ควบคุม และใช้ภายนอกให้กับแขนหุ่นยนต์

ในส่วนการใช้งาน PLC ร่วมกับแผงวงจรรีเลย์ ที่เดิมทีแผงวงจรรีเลย์ใช้ร่วมกับแผงวงจร PLC เดิมที่มีขนาดใหญ่ มีการศึกษาการทำงานของแผงวงจรรีเลย์ แล้วเปลี่ยนแผงวงจร PLC ที่ใช้ร่วมกับแผงวงจรรีเลย์เป็น PLC SIEMENS s7-200 ใช้งานแทน เพราะทางผู้จัดทำมีความรู้ด้านการใช้งาน PLC ทำให้ลดเวลาในการศึกษา PLC ใหม่ และเพื่อจะลดเวลาในการศึกษาแผงวงจร PLC เดิม ซึ่งการเชื่อมต่อสายมี ดังนี้



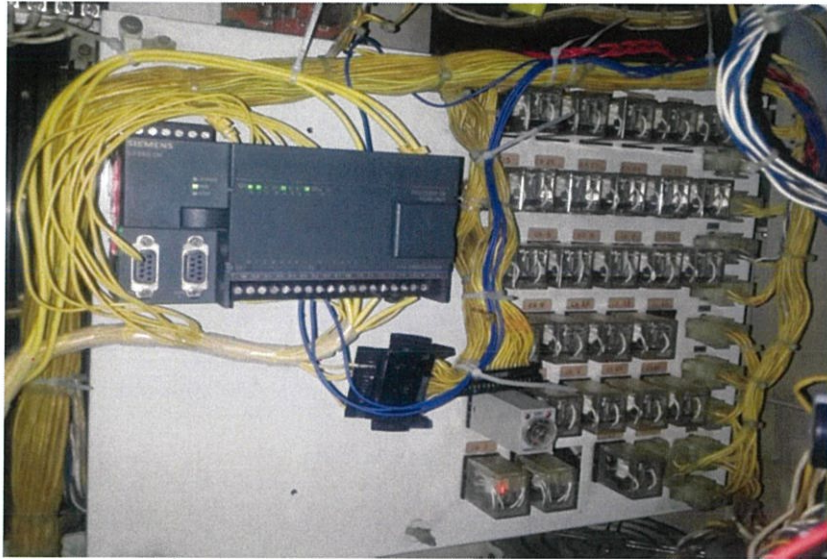
รูปที่ 3.16 การเชื่อมต่อสาย PLC ด้าน Input

จากรูปที่ 3.16 แสดงการเชื่อมต่อสายด้าน Input จะแบ่งออกเป็น 3 ส่วน คือ ส่วนที่เป็นลิมิตสวิตช์ ที่เป็นสัญญาณการป้องกันการดำเนินงานจากแขนหุ่นยนต์ แต่เนื่องจากขนย้ายแขนหุ่นยนต์ทำให้คอนเนกเตอร์ที่เชื่อมขา GND จากลิมิตสวิตช์ชนกับขาที่มีแรงดันไฟฟ้าทำให้ขาเชื่อมต่อกัน จึงหลีกเลี่ยงปัญหาที่เกิดขึ้นจึงตัดการใช้งานของลิมิตสวิตช์ออกป้องกันการลัดวงจร ส่วนที่สอง คือ ส่วน cRIO (NI CompactRIO) รับมาจาก NI 9474 ที่เป็น DO จาก NI และส่วนที่เป็นสวิตช์โดยตรงสำหรับดูดเงินเมื่อมีความผิดปกติหน้าแขนหุ่นยนต์



รูปที่ 3.17 การเชื่อมต่อสาย PLC ด้าน Output

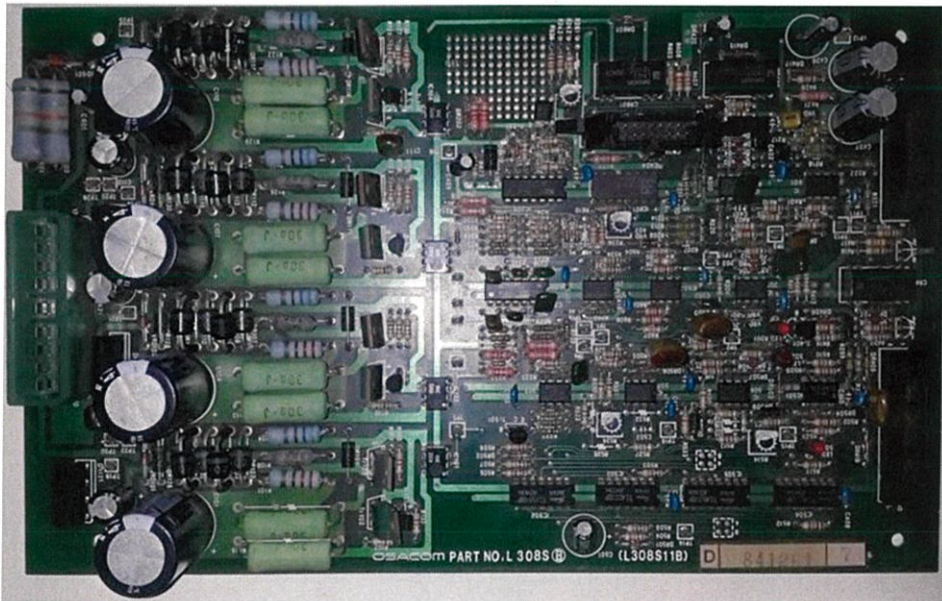
จากรูปที่ 3.17 แสดงการเชื่อมต่อสายด้าน Output แบ่งออกเป็น 2 ส่วน คือ ส่วนส่งการทำงานให้กับเพื่อไปสั่งงานให้ส่วนต่างๆกับแขนหุ่นยนต์ ทั้งการปลดเบรก สั่งแมกเนติกคอนแทรกเตอร์ และสั่งการ Enable แผงวงจรขับเคลื่อน และส่วนสั่งให้กับ cRIO รู้ผลจากการปลดเบรก



รูปที่ 3.18 ภาพการใช้ PLC เชื่อมต่อกับแผงวงจรรีเลย์

3.3.2.2 ส่วน B

จากรูปที่ 3.15 ที่แสดงส่วน B เป็นส่วนที่ติดตั้ง แผงวงจรขับเคลื่อนมอเตอร์ หรือชุดไดรฟ์ (Drive Board) ซึ่งแบ่งออกเป็น 2 ชั้นทั้งหมด 5 แกน ชั้นด้านหน้าประกอบด้วย 2 แกน คือ แกน R และ L ส่วนด้านหลังประกอบด้วย แกน θ แกน ϕ และแกน U



รูปที่ 3.19 แผงวงจรขับเคลื่อนมอเตอร์

จากรูปที่ 3.19 แสดงแผงวงจรขับเคลื่อนมอเตอร์เชื่อมต่อแกน R, L, U ที่ได้ทำการเพิ่มเติม เนื่องจากไม่ทราบเงื่อนไขการใช้ Enable เดิมของแผงวงจร จึงได้ทำการปรับปรุงสร้าง

Enable ใหม่ขึ้นโดยการตัดขาสัญญาณ MOSFET ที่ให้สัญญาณแก่ Pulse Width Module และส่วนของข้อต่อแกน θ และแกน ϕ เปลี่ยนจากแผงวงจรขับเคลื่อนเดิมเป็น Accelus รุ่น asp-090-36 ซึ่งเป็นชุดวงจรขับเคลื่อนมอเตอร์สำเร็จรูป เพราะแผงวงจรขับเคลื่อนมอเตอร์เดิมของสองข้อต่อแกนนี้เมื่อปรับปรุงแล้วมีปัญหาไหม้แรงดันไฟฟ้าย้อนกลับเกิดขึ้น ทำให้ไม่สามารถใช้งานได้ จึงได้นำชุดสำเร็จรูปมาใช้งานแทน



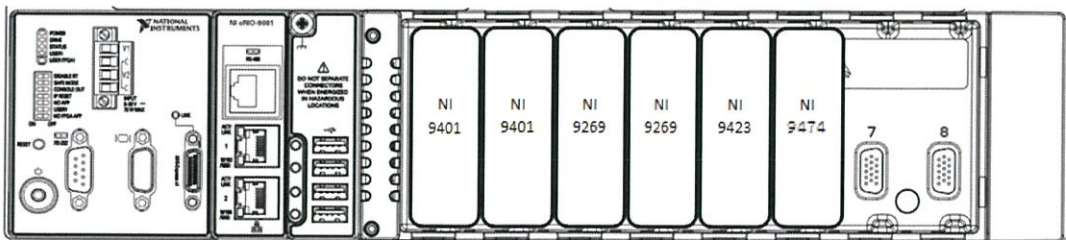
รูปที่ 3.20 ชุดวงจรขับเคลื่อน Accelus รุ่น asp-090-36

3.3.2.3 ส่วน C

ส่วน C จากรูปที่ 3.15 ประกอบด้วย แมกเนติกคอนแทกเตอร์ ที่ทำหน้าที่ตัดต่อกระแสไฟฟ้าในวงจรภายในตู้กับส่วนควบคุมต่างๆ ส่วนของเทอร์มินอลที่ใช้สำหรับเชื่อมต่อการทำงานภายนอกตู้ควบคุมและนอกแขนหุ่นยนต์

3.3.2.4 ส่วน D

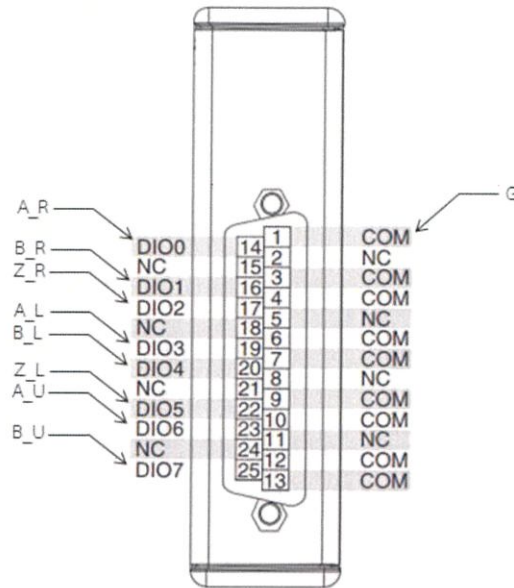
ส่วน D จากรูปที่ 3.15 เป็นส่วนที่ติดตั้งตัวควบคุมในที่นี้ คือ NI (National Instruments) ตัวควบคุม CompactRIO โมเดล NI cRIO-9082 1.33 GHz Dual-Core CPU, LX150 FPGA และระบบปฏิบัติการแบบ Real-time ในตัวควบคุมโมเดลรุ่นนี้มีความสามารถหลักคือ เป็นระบบมัลติคอร์ประสิทธิภาพสูง สามารถแสดงการประมวลผลซับซ้อนได้ และการใช้งานควบคุมต่างๆ ที่ซับซ้อน



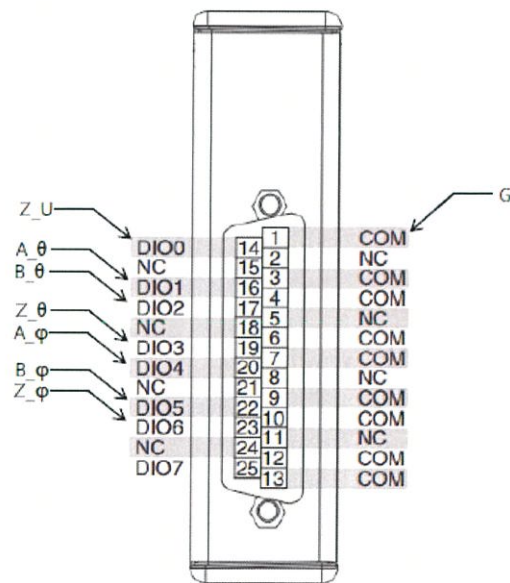
รูปที่ 3.21 โมดูล NI ที่ใช้ใน cRIO

จากรูปที่ 3.21 แสดงโมดูลที่ใช้ใน CompactRIO Chassis มีทั้งหมด 4 โมเดล รวมทั้งหมด 6 โมดูล ซึ่งทำหน้าที่รับและส่งสัญญาณต่างๆ ดังต่อไปนี้

1. NI 9401 เป็นโมดูล Digital Input/output ทำหน้าที่รับสัญญาณแอนโดเตอร์จากเซนเซอร์ที่ส่งมา

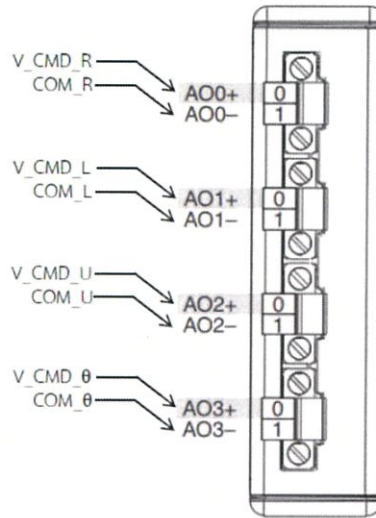


รูปที่ 3.22 โมดูล NI 9401 (Encoder Group 1)

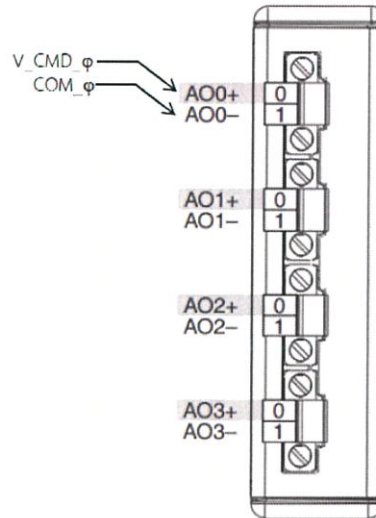


รูปที่ 3.23 โมดูล NI 9401 (Encoder Group 2)

2. NI 9269 เป็นโมดูล Analog Voltage Output ทำหน้าที่ส่งสัญญาณ Command ให้แก่ชุดแผงวงจรขับเคลื่อนแก่เซนเซอร์

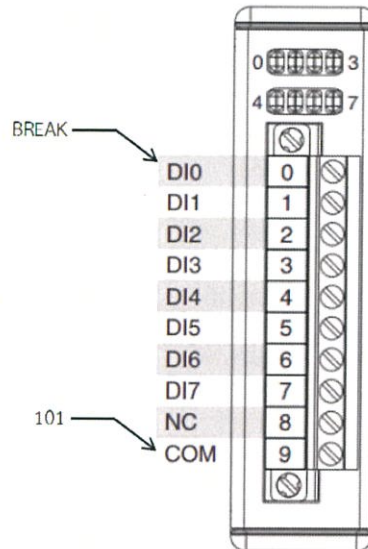


รูปที่ 3.24 โมดูล NI 9269 (CMD Group 1)



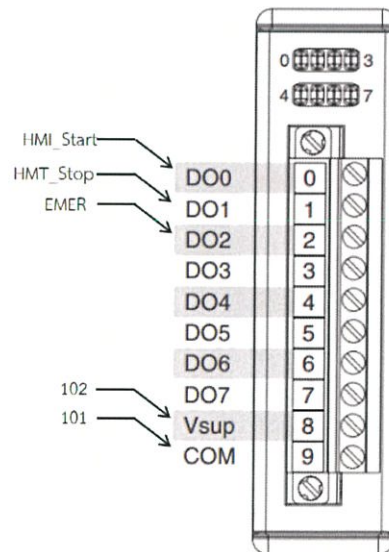
รูปที่ 3.25 โมดูล NI 9269 (CMD Group 2)

3. NI 9423 เป็นโมดูล Digital Input ทำหน้าที่รับสัญญาณจาก PLC เมื่อเบรกถูกปลด หรือ Input อื่นๆ เพื่อไปใช้งานในโปรแกรม LabVIEW



รูปที่ 3.26 โมดูล NI 9423

4. NI 9474 เป็นโมดูล Digital Output ทำหน้าที่ส่งสัญญาณจาก HMI ไปสู่ PLC เมื่อมีการกด Start จาก PLC



รูปที่ 3.27 โมดูล NI 9474



รูปที่ 3.28 ภาพการเชื่อมต่อ cRIO กับโมดูล NI

3.4 การสร้างเส้นทางการเคลื่อนที่

ในหัวข้อนี้กล่าวถึงการสร้างเส้นทางในการเคลื่อนที่ ทัวไปรูปแบบการเคลื่อนที่ของแขนหุ่นยนต์นั้นจะขึ้นอยู่กับการใช้งานของแขนหุ่นยนต์ เช่น การจับและวางสิ่งของกับงานเชื่อม จะมีรูปแบบการเคลื่อนที่แตกต่างกัน โดยที่การจับและวางสิ่งของนั้นการเคลื่อนที่จะเป็นแบบเส้นตรงเป็นมุมฉาก ส่วนการเชื่อมการเคลื่อนที่จะมีวิถีโค้ง มีการหมุนเพิ่มขึ้น เป็นการเคลื่อนที่ซับซ้อนมากขึ้น ในที่นี้ต้องการการเคลื่อนที่ตามเส้นทาง ซึ่งอาจมีเส้นทางที่เป็นวิถีโค้ง ในเส้นทางการเคลื่อนที่

3.4.1 การเลือกใช้โปรแกรมในการสร้างเส้นทางการเคลื่อนที่

ในการสร้างเส้นทางการเคลื่อนที่นั้น ทำโดยวิธีการสอนแขนหุ่นยนต์ให้ทำงาน ในเส้นตรงอาจกำหนดพิกัดจุดเริ่มต้นและจุดปลาย หรือทำได้โดยการให้แขนหุ่นยนต์เคลื่อนที่ไปที่ละจุด แต่ในการสร้างเส้นทางที่เป็นวิถีโค้งการกำหนดจุดจะมีความลำบาก เนื่องจากจะต้องกำหนดพิกัดจุดให้ละเอียดมากเพื่อนำให้ได้เส้นโค้ง ซึ่งจะทำงานใช้เวลาในการดำเนินการนานมากขึ้น ดังนั้นจึงได้นำโปรแกรม RobotStudio เข้ามาช่วยในการสร้างเส้นทางการเคลื่อนที่ที่มีความง่ายขึ้น เนื่องจากความสามารถของโปรแกรม RobotStudio สามารถจำลองแขนหุ่นยนต์ที่ใช้งาน และนำหุ่นยนต์มาสร้างพิกัดเส้นทางการเคลื่อนที่ที่ต้องการ กำหนดความละเอียดในการกำหนดพิกัดที่เคลื่อนที่ได้ และสามารถนำข้อมูลจากโปรแกรมมาใช้งานในระบบควบคุมต่อได้ รวมทั้งการสร้างการเคลื่อนที่ที่สามารถสร้างรูปร่าง รูปแบบต่างๆได้จากโปรแกรม SolidWork จึงได้นำโปรแกรม RobotStudio มาใช้งาน

3.4.2 การนำภาษา RAPID ออกมาใช้งาน

จากการสร้างเส้นทางการเคลื่อนที่ในโปรแกรม RobotStudio จะได้ข้อมูลที่ได้ เป็นภาษา RAPID ซึ่งภาษานี้เป็นภาษาที่ป้อนให้กับตัวควบคุมของแขนหุ่นยนต์ที่เป็นของบริษัท เอบีบี เท่านั้น ภาษา RAPID ที่ใช้จะมีรูปแบบ ดังนี้

	A	B	C	D	E	F	G
1	MODULE						
2		0	778.1793	840	100	0	0
3		0	778.1793	1000	100	0	0
4		-32.7682	1022.98105	1000	100	0	0
5		-51.9465	889.159538	900	100	0	0
6		-59.0933	896.074337	899.009757	100	0	0
7		-65.3271	903.822439	898.020198	100	0	0
8		-70.5515	912.284055	897.030345	100	0	0
9		-74.6856	921.328447	896.040395	100	0	0
10		-77.6658	930.815915	895.050922	100	0	0
11		-79.4458	940.599736	894.060593	100	0	0
12		-79.9983	950.528924	893.071454	100	0	0
13		-79.3146	960.44979	892.080796	100	0	0
14		-77.4053	970.209359	891.091944	100	0	0
15		-74.3	979.656461	890.101007	100	0	0
16		-70.0466	988.64552	889.112395	100	0	0
17		-64.7109	997.037214	888.121228	100	0	0
18		-58.3751	1004.70231	887.132809	100	0	0
19		-51.1376	1011.52193	886.141462	100	0	0
20		-43.1096	1017.39111	885.153189	100	0	0
21		-34.4158	1022.2188	884.161712	100	0	0
22		-25.1898	1025.9307	883.173538	100	0	0
23		-15.575	1028.46923	882.18198	100	0	0
24		-5.71917	1029.79531	881.193859	100	0	0
25		4.224702	1029.88837	880.202271	100	0	0
26		14.10363	1028.74698	879.214156	100	0	0
27		23.7643	1026.38886	878.222585	100	0	0

รูปที่ 3.31 ข้อมูลพิกัดบน Microsoft Excel

จากตารางที่ได้ใน Microsoft Excel จะเห็นได้ว่าค่าที่ได้จากโปรแกรม RobotStudio จะเป็นค่าพิกัดจุด x , y , z ตามรูปแบบการเคลื่อนที่ที่สร้างไว้ในคอลัมน์ B คอลัมน์ C และคอลัมน์ D ส่วนในคอลัมน์ E และคอลัมน์ F เป็นค่าของความเร็วในการเคลื่อนที่ และค่ามุม ϕ ตามลำดับ

3.5 ระบบคอมพิวเตอร์ซอฟต์แวร์

ในส่วนการพัฒนาซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์นี้ ได้นำโปรแกรม MATLAB มาใช้ในส่วนการพัฒนาสมการต่างๆ ที่จะนำไปใช้ในระบบควบคุมแขนหุ่นยนต์ และนำโปรแกรม LabVIEW มาใช้ในการสร้างระบบควบคุมแขนหุ่นยนต์ในการทำให้แขนหุ่นยนต์สามารถเคลื่อนไหว

3.5.1 การพัฒนาสมการในโปรแกรม MATLAB

ในการพัฒนาจะนำไปใช้ในระบบควบคุมแขนหุ่นยนต์ การใช้งานโปรแกรม MATLAB จะใช้ในส่วนการพัฒนาสมการ Forward และ Inverse Kinematics การเคลื่อนที่ของแต่ละข้อต่อแขนหุ่นยนต์เคลื่อนที่สัมพันธ์กัน และสามารถจำลองเส้นทางในการเคลื่อนที่ โดยการสร้าง m-files ซึ่งใช้สำหรับการเขียนสคริปและฟังก์ชัน ในการทำงานของ m-files จะทำงานผ่านการเขียนสคริปในโปรแกรมหลัก จะเรียกใช้ฟังก์ชันต่างๆเพื่อแสดงผลลัพธ์ โดยฟังก์ชันที่โปรแกรมหลักเรียกใช้เป็นสมการแขนหุ่นยนต์ที่สร้างขึ้น

```

1 - clear all
2 - %Length of Link Arm each jointse$
3 - L=[60;760;760;200;120;0;0];
4 - Tx=L(3,1)+L(5,1)+L(7,1);
5 - Ty=0;
6 - Tz=L(1,1)+L(2,1)+L(4,1)+L(6,1);
7 - Tbf=[0;778.1793;840]; %positino beforezFhbf=0;
8 - zFhbf=0;
9 - Zbf=finvkin(Tbf,L,zFhbf);%Zeta before
10 - [t1,t2,t3,t4,t5,t6] = fforckine(Zbf,L);
11 - ch=plotjoint5(t1,t2,t3,t4,t5,t6);
12
13 %Task pathmovement Module
14 - Task = xlsread('C:\Users\YADONG\Desktop\Test_Robot_Project.m\Demol.xlsx','Sheet1','B2:F1002');
15 - m = size(Task);
16 - m=m(1,1);
17 - Vbf=0;
18 - Vibf=[0;0;0;0;0];
19 - m1=1;
20 - while m1<=m
21
22 - Zbf=finvkin(Tbf,L,zFhbf)
23 - Taf=[Task(m1,1):Task(m1,3)];%positino after
24 - Vaf=Task(m1,4); % speed
25 - zFhbf=Task(m1,5);
26 - Zaf=finvkin(Taf,L,zFhbf);%Zeta after
27 - [t1,t2,t3,t4,t5,t6] = fforckine(Zaf,L);
28 - [V1,Z1,a1] = Espeedjoint(Zbf,Taf,Zbf,Zaf,Vbf,Vaf,Vibf);
29 - %pause(.1)
30 - %$m Line MOVE
31
32 - %plot1 Task1
33 - subplot(2,2,1);
34 - view([45 30]);
35 - hold on
36 - plot3([Tbf(1,1),Taf(1,1)],[Tbf(2,1),Taf(2,1)],[Tbf(3,1),Taf(3,1)],'-k','LineWidth', 2)
37
38 - %plot2 Line Tool Robot
39 - subplot(2,2,2);
40 - view([45 30]);
41 - hold on
42 - plot3(t6(1,4),t6(2,4),t6(3,4),'-k','LineWidth', 2)
43 - %%END Line MOVE
44
45 - %command = input('Do You Want simulation move position [Y/N]: ','s');
46 - %if command == ('Y'); command == ('y');

```

รูปที่ 3.32 โปรแกรมหลักใน MATLAB

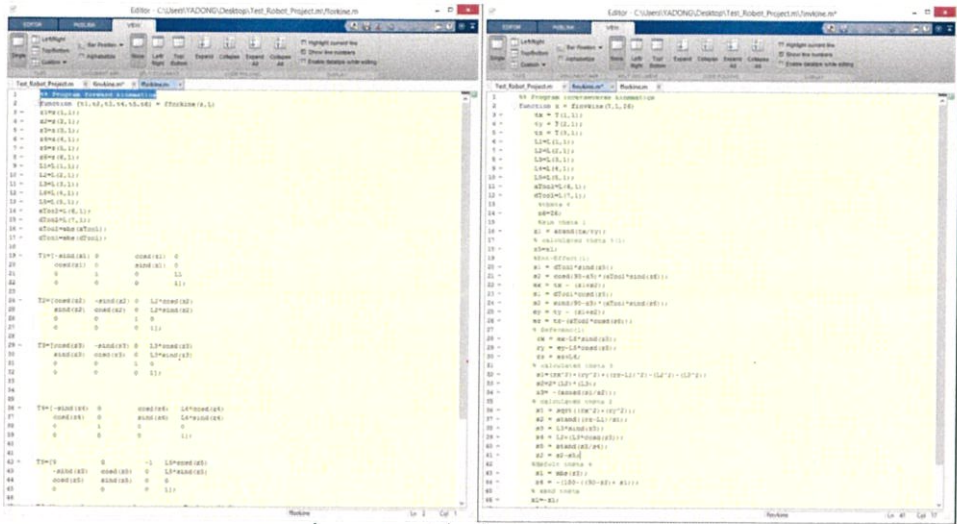
ฟังก์ชันที่ใช้ในโปรแกรม MATLAB ที่ออกแบบมีทั้งหมด 4 ส่วน ดังนี้

3.5.1.1 ฟังก์ชัน Forward Kinematics เป็นส่วนที่รับค่ามุมแต่ละข้อต่อแขนเข้ามา คำนวณหาพิกัดจุดปลายของแขนหุ่นยนต์

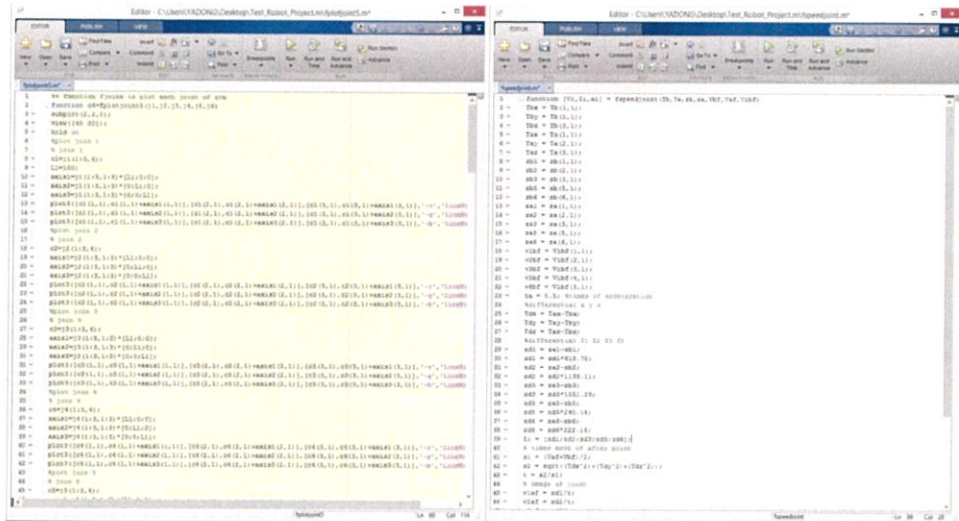
3.5.1.2 ฟังก์ชัน Inverse Kinematics เป็นฟังก์ชันที่รับค่าพิกัดตำแหน่งจุดปลายของแขนหุ่นยนต์เข้าสมการเพื่อหามุมของแต่ละข้อต่อหมุน

3.5.1.3 ฟังก์ชันการเคลื่อนที่ เป็นฟังก์ชันที่รับค่าความแตกต่างของมุมในตำแหน่งปัจจุบันกับตำแหน่งที่แขนหุ่นยนต์จะเคลื่อนที่ต่อไป มาคำนวณหาความเร็วในแต่ละข้อต่อแขนเพื่อให้แขนหุ่นยนต์ที่เคลื่อนที่ประสานกัน

3.5.1.4 ฟังก์ชันจำลองเส้นทางการเคลื่อนที่ เป็นการสร้างเวกเตอร์จากพิกัดตำแหน่งปลายของเส้นทางการเคลื่อนที่



รูปที่ 3.33 ฟังก์ชันที่ใช้ใน MATLAB (1)



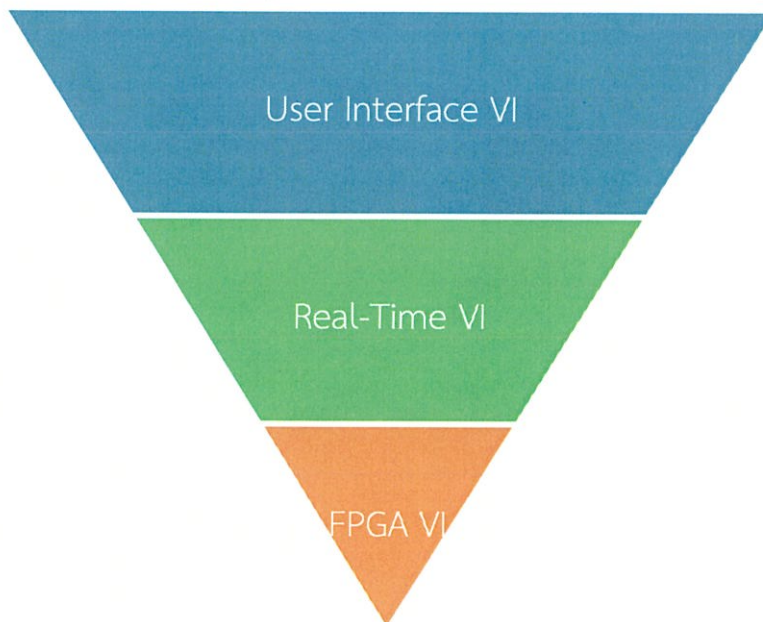
รูปที่ 3.34 ฟังก์ชันที่ใช้ใน MATLAB (2)

3.5.2 การสร้างซอฟต์แวร์ในโปรแกรม LabVIEW

ในการสร้างซอฟต์แวร์การเคลื่อนที่ในปริญญาโทเลือกใช้โปรแกรม LabVIEW การสำหรับสร้างซอฟต์แวร์ โดยทำงานภายใน LabVIEW จะแบ่งออกเป็น 3 ชั้นการทำงาน

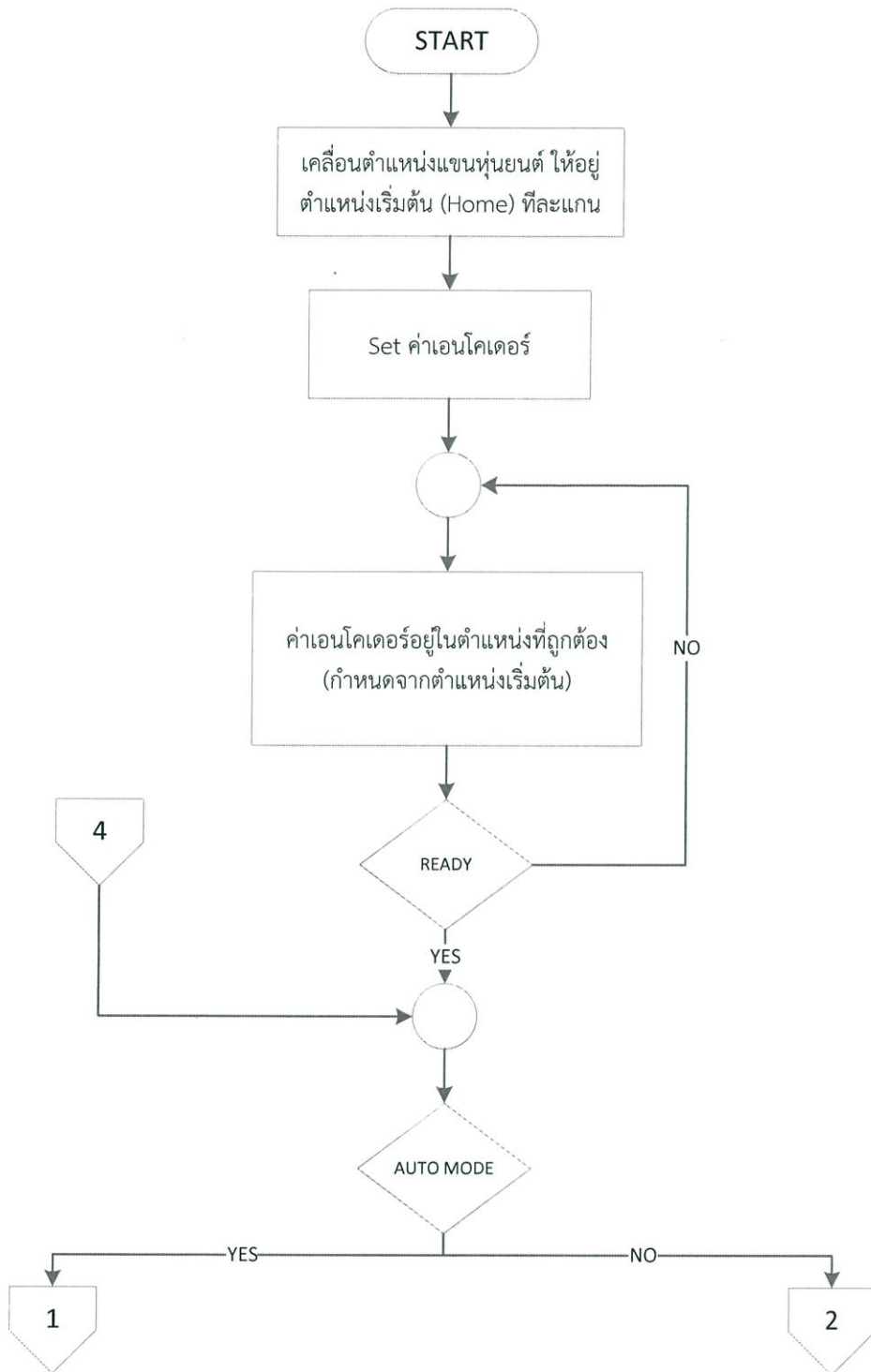
- User Interface VI เป็นส่วนประมวลผลบนเครื่องคอมพิวเตอร์
- RT VI เป็นส่วนประมวลผลบนระบบปฏิบัติการ Real-Time
- FPGA VI เป็นส่วนประมวลผลบนฮาร์ดแวร์

ในแต่ละชั้นการทำงานจะขึ้นอยู่กับความเร็วในการประมวลผล และความซับซ้อนของการทำงาน ซึ่ง FPGA จะมีการประมวลผลเร็วที่สุด แต่ไม่สามารถใช้งานในรูปแบบซับซ้อนได้ โดยการใช้งานเป็นดังรูปที่ 3.35 ต่อไปนี้

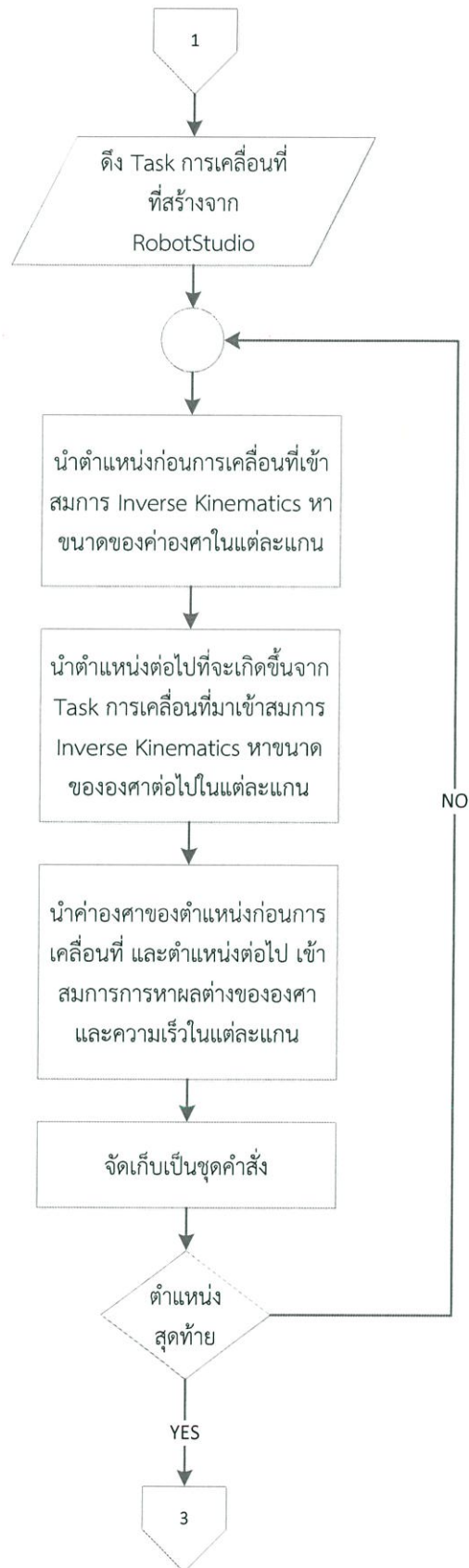


รูปที่ 3.35 โครงสร้างการประมวลผลของ LabVIEW

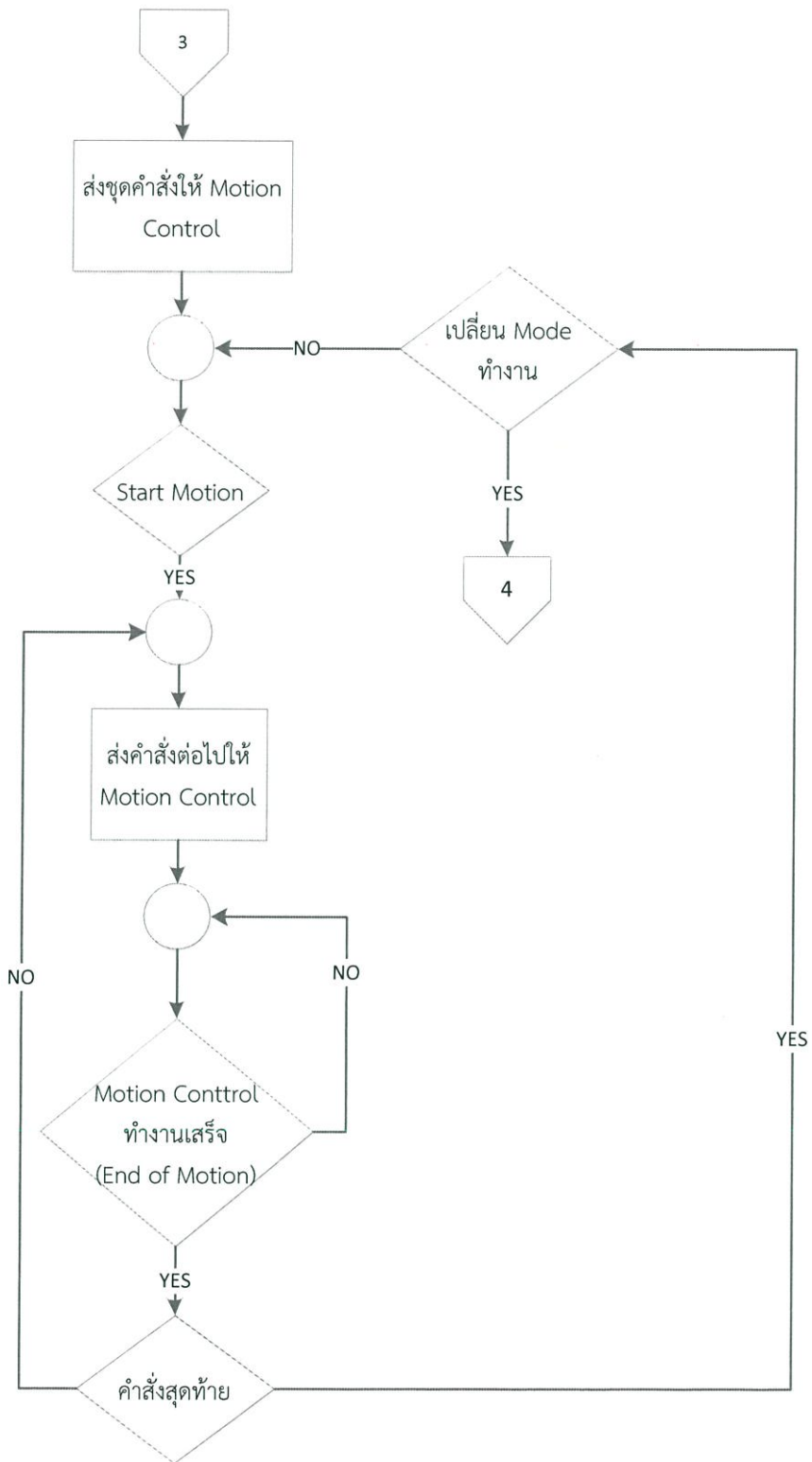
ในระบบการควบคุมที่ได้ทำการพัฒนาขึ้นนี้สามารถทำงานได้ 2 วิธีการ คือ Auto Mode ที่ทำงานสั่งการเคลื่อนที่แขนหุ่นยนต์เป็นชุดคำสั่งที่สร้างขึ้น และ Manual Mode ที่ทำงานตามการสั่งจากการกดเลือก โดยภายใน Manual Mode แบ่งเป็น 2 ส่วน คือ ส่วนเคลื่อนที่ตามแกน x, y, z และส่วนเคลื่อนที่ข้อต่อหมุน โดยเลือกเคลื่อนที่ที่ละข้อต่อหมุน R, L, U, θ , ϕ โดยจะแสดงการทำงานตามแผนผังรูปที่ 3.36 – 3.39 ดังนี้



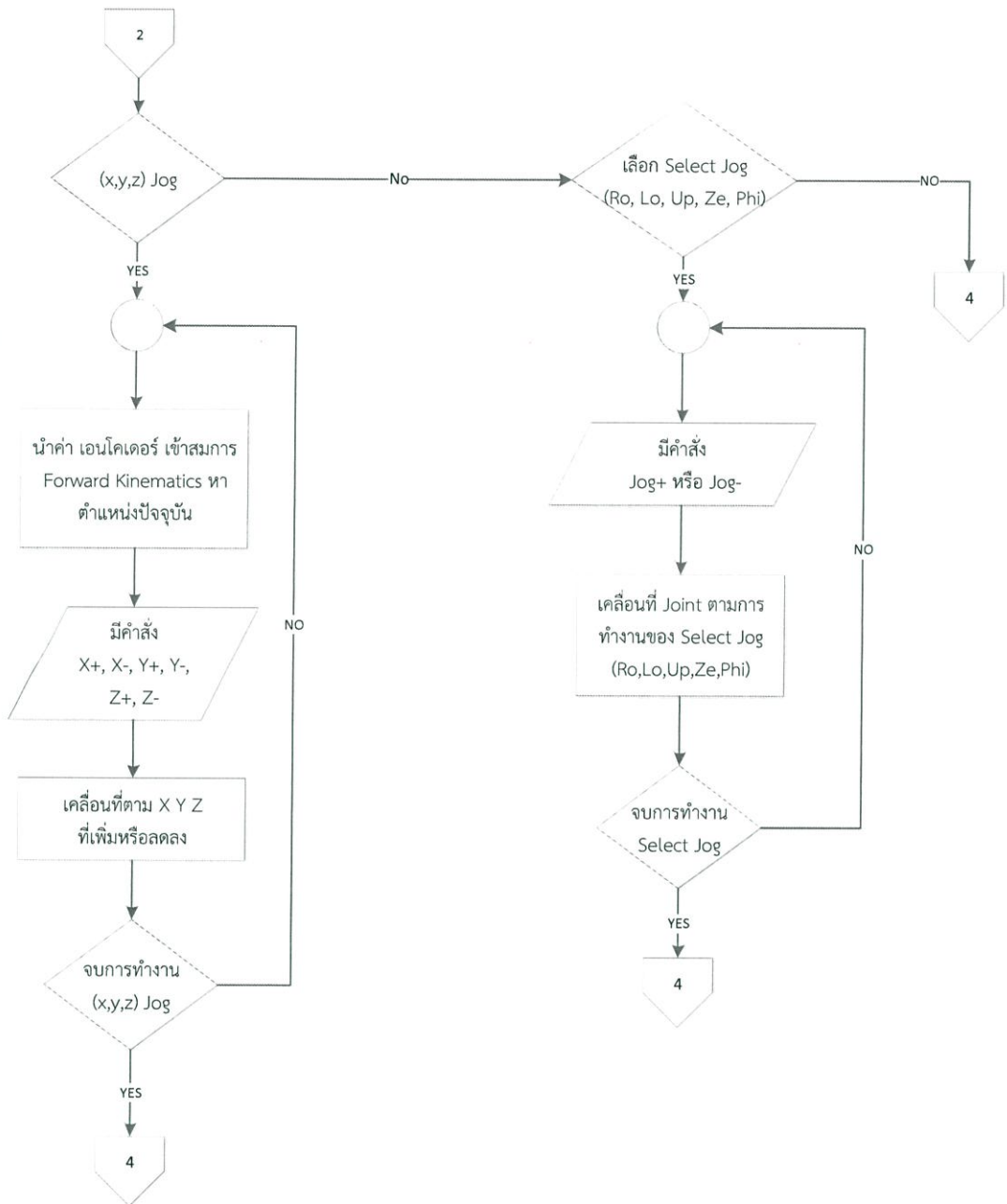
รูปที่ 3.36 แผนผังการทำงานของซอฟต์แวร์ (1)



รูปที่ 3.37 แผนผังการทำงานของซอฟต์แวร์ (2)

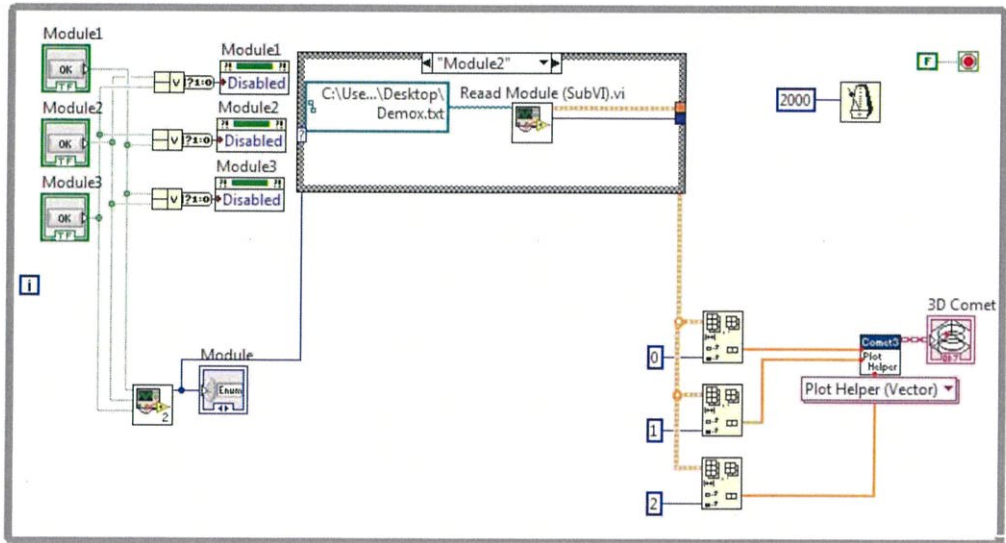


รูปที่ 3.38 แผนผังการทำงานของซอฟต์แวร์ (3)



รูปที่ 3.39 แผนผังการทำงานของซอฟต์แวร์ (4)

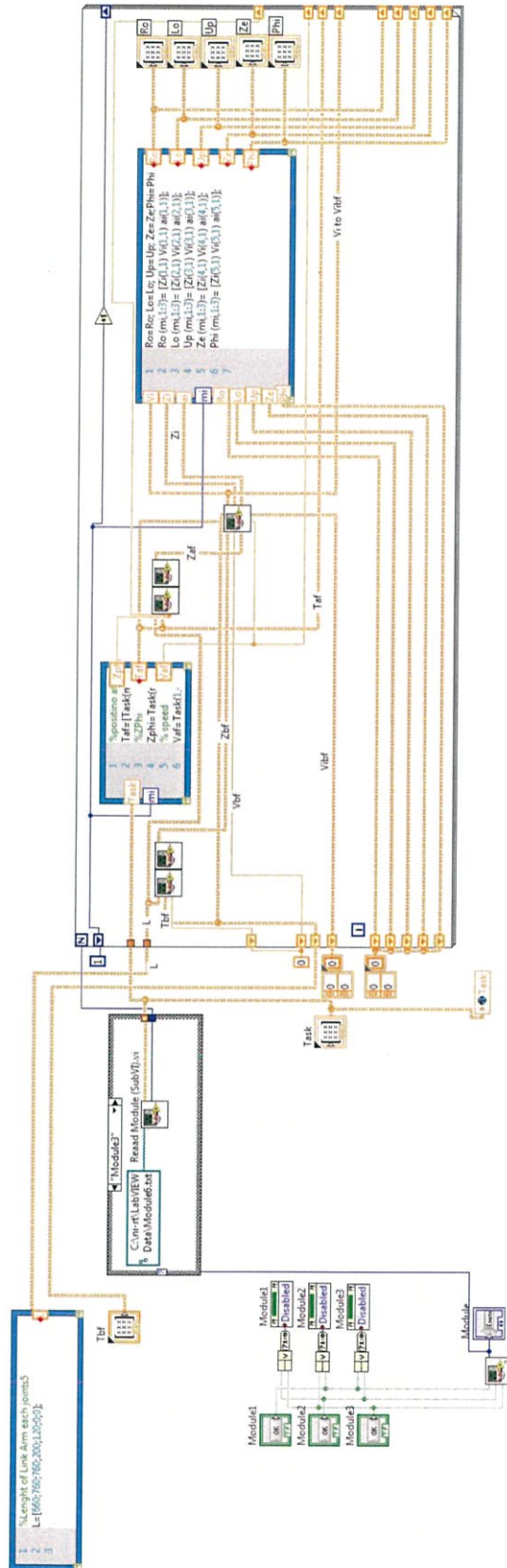
3.5.2.1 ส่วนประมวลผลบน User Interface VI



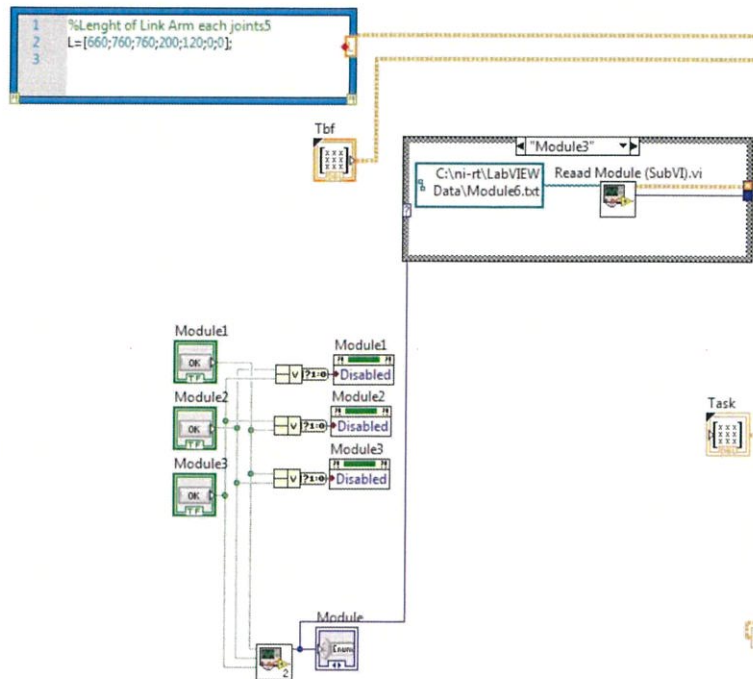
รูปที่ 3.40 ส่วนซอฟต์แวร์ในการสร้างกราฟเส้นทางการเคลื่อนที่

จากรูปที่ 3.40 แสดงซอฟต์แวร์ส่วนนี้เป็นการนำเส้นทางการเคลื่อนที่ที่สร้างจากโปรแกรม RobotStudio มาสร้างกราฟเส้นทางการเคลื่อนที่ เพื่อให้เห็นเส้นทางการเคลื่อนที่ที่จะเกิดขึ้นจริงก่อนสั่งให้แขนหุ่นยนต์ทำงาน

3.5.2.2 ส่วนประมวลผลบน Real-time VI

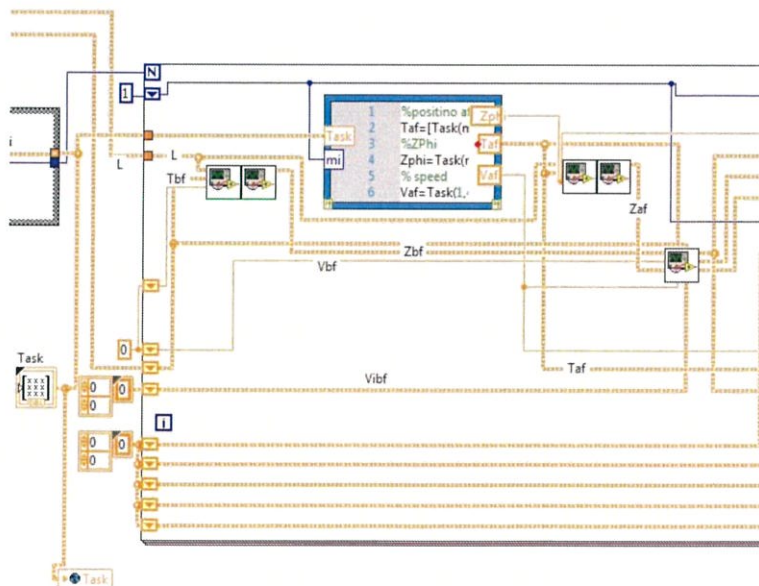


รูปที่ 3.41 ส่วนซอฟต์แวร์ที่พัฒนาจากโปรแกรม MATLAB



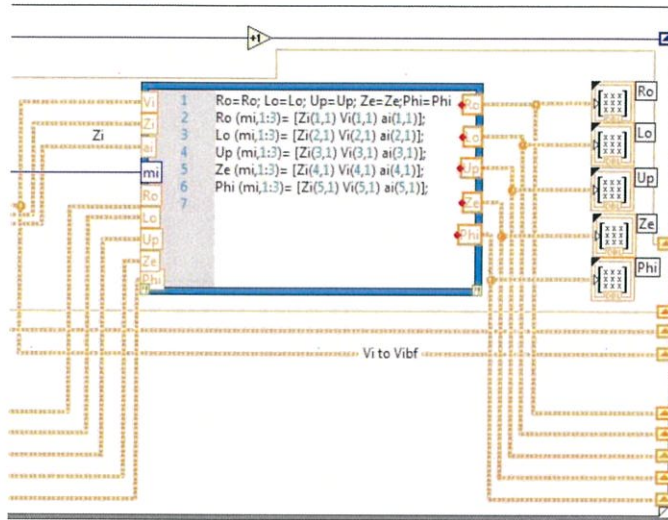
รูปที่ 3.42 ส่วนการเลือกเปิดไฟล์รูปแบบการเคลื่อนที่จาก RobotStudio

การทำงานในส่วนนี้เป็นส่วนการเลือกไฟล์รูปแบบการเคลื่อนที่ที่สร้างขึ้นจากโปรแกรม RobotStudio เพื่อเป็นชุดคำสั่งในการคำนวณใน Inverse Kinematics



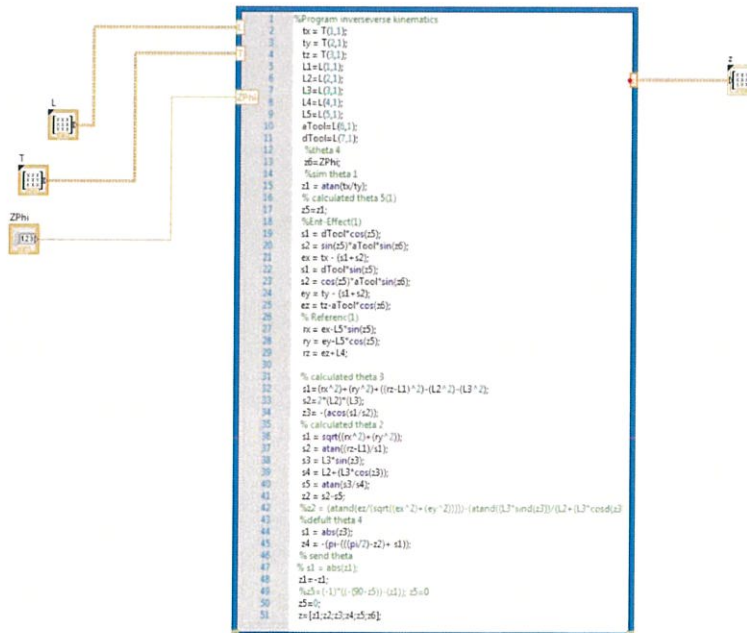
รูปที่ 3.43 ส่วนการนำข้อมูลเข้าสมการ Kinematics และฟังก์ชันการเคลื่อนที่

จากรูปที่ 3.43 จะเป็นส่วนซอฟต์แวร์ที่นำพิกัดจุดก่อนการเคลื่อนที่ และพิกัดจุดที่จะเคลื่อนที่ไปมาคำนวณสมการ Inverse Kinematics เพื่อหาขนาดมุมข้อต่อของแต่ละข้อต่อ ณ ตำแหน่งนั้นๆ และนำค่าที่ได้เข้าสมการการเคลื่อนที่เพื่อหาขนาดมุมของแต่ละข้อต่อ และความเร็วที่ใช้ในการเคลื่อนที่



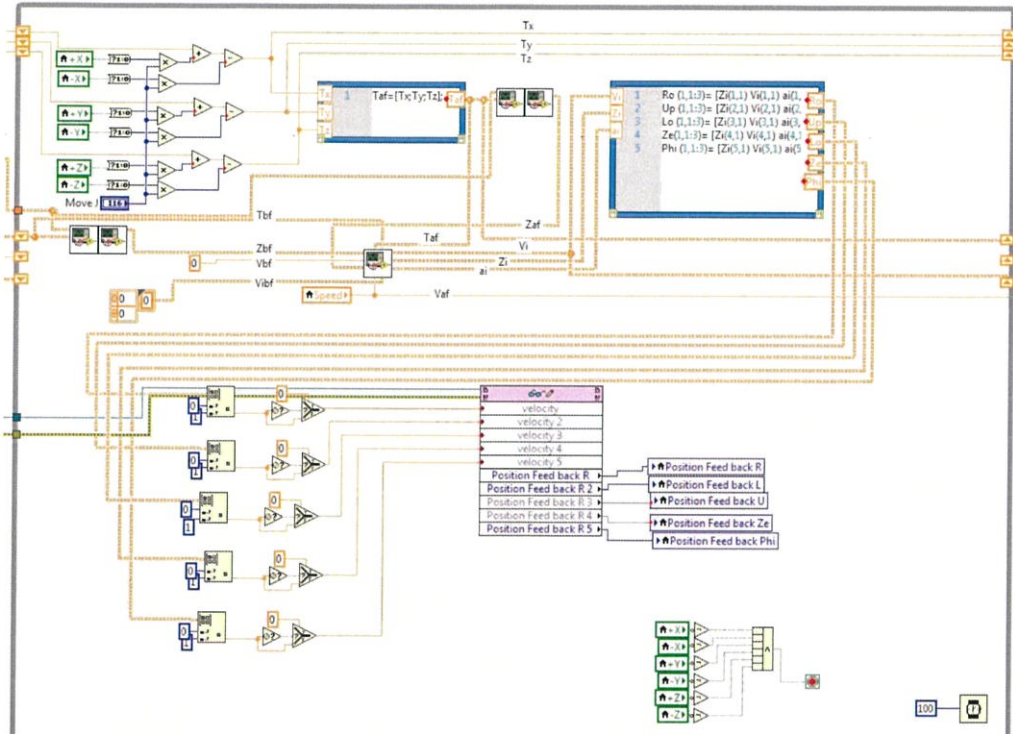
รูปที่ 3.44 ส่วนจัดชุดคำสั่งของการเคลื่อนที่

การทำงานจากรูปที่ 3.44 เป็นการนำผลลัพธ์จากสมการ และการเคลื่อนที่นำมาจัดลำดับเป็นชุดคำสั่งการเคลื่อนที่ของแขนหุ่นยนต์



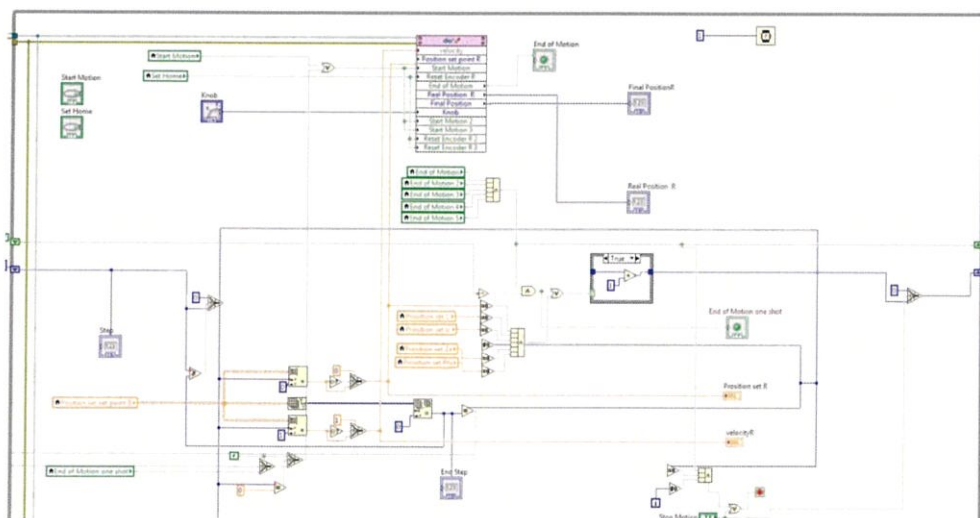
รูปที่ 3.45 สมการจาก MATLAB ที่นำมาใช้ใน LabVIEW

รูปที่ 3.45 เป็นการนำฟังก์ชัน MathScript ที่สามารถนำโค้ดของ MATLAB เขียนลงใน LabVIEW เป็นส่วนในการประมวลผลต่างๆ อาทิเช่น Inverse Kinematic สมการการเคลื่อนที่ และการจัดเรียงชุดคำสั่ง เป็นต้น



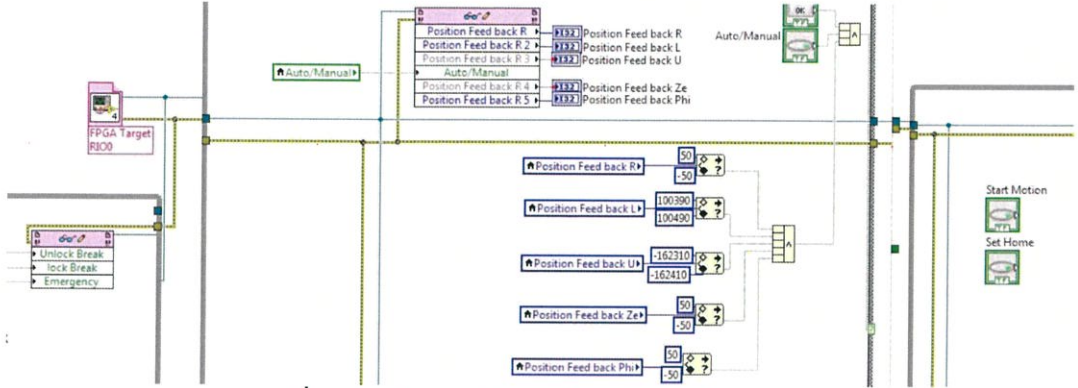
รูปที่ 3.46 ส่วนการเคลื่อนที่แบบ Manual

จากรูปที่ 3.46 ในส่วนนี้เป็นซอฟต์แวร์ส่วน Manual Mode สำหรับการเคลื่อนที่ที่พิกัดจุดปลายตามแกน x, y, z สามารถสั่งการผ่านหน้า GUI (Graphic User Interface)



รูปที่ 3.47 ส่วนการเคลื่อนที่แบบ Auto

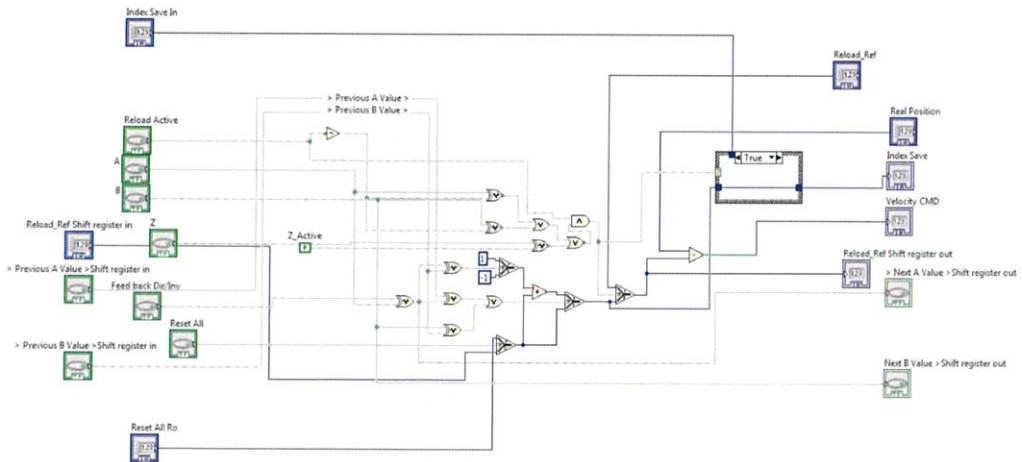
จากรูปที่ 3.47 แสดงถึงซอฟต์แวร์การเคลื่อนที่แบบ Auto Mode ซึ่งเป็นการเคลื่อนที่ตามชุดคำสั่งที่กำหนดไว้จากในรูปที่ 3.42 การทำงานจะทำงานตามลำดับ



รูปที่ 3.48 ส่วนการเปิดการทำงาน FPGA ใน Real-Time

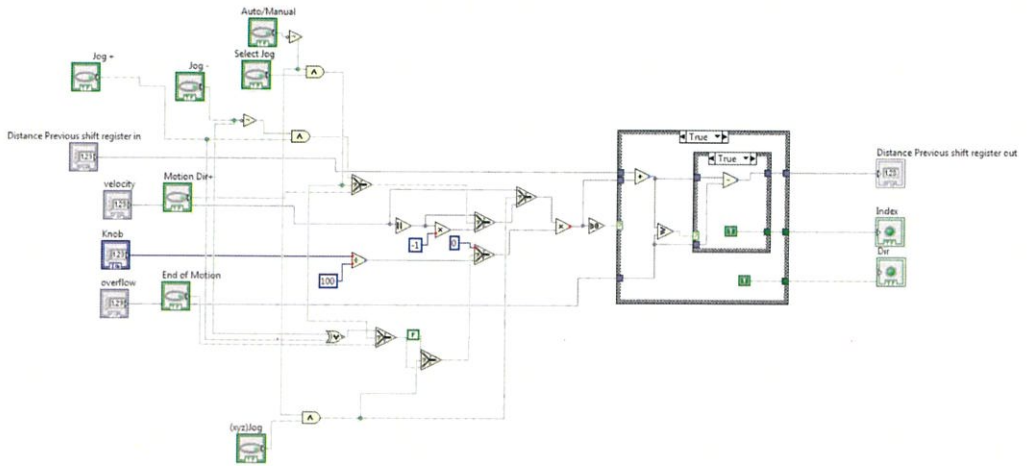
จากรูปที่ 3.48 แสดงส่วนการเปิดการใช้งาน FPGA รวมถึงการอ่าน และเขียนค่าข้อมูลต่างๆ จาก FPGA ที่เป็นส่วนการทำงาน Motion Control

3.5.2.3 ส่วนประมวลผลบน FPGA

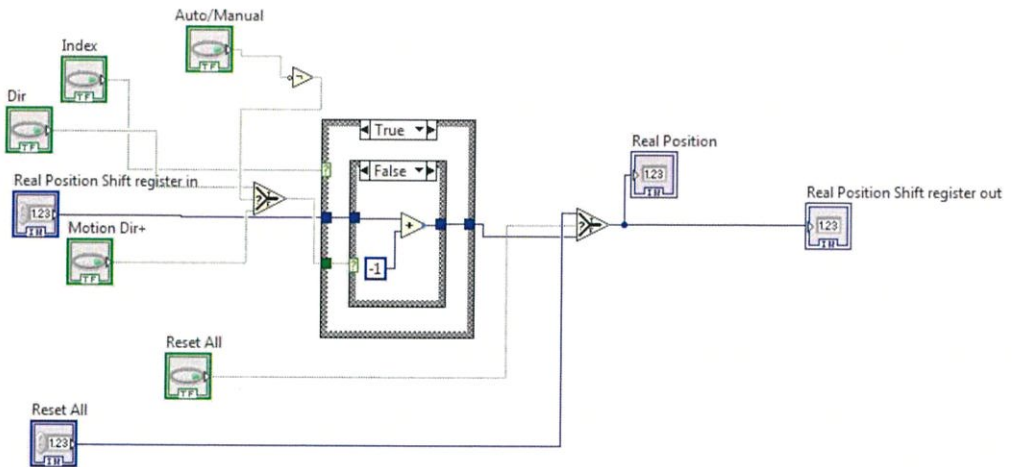


รูปที่ 3.49 ส่วนควบคุมการทำงานของเอนโคเดอร์

จากรูปที่ 3.49 เป็นส่วนควบคุมการนับค่าเอนโคเดอร์ที่เป็นเซ็นเซอร์การเคลื่อนที่ของข้อต่อในแต่ละข้อต่อ และค่าที่นับได้จากเอนโคเดอร์จะส่งเป็นสัญญาณป้อนกลับให้แก่ตัวควบคุมตำแหน่ง

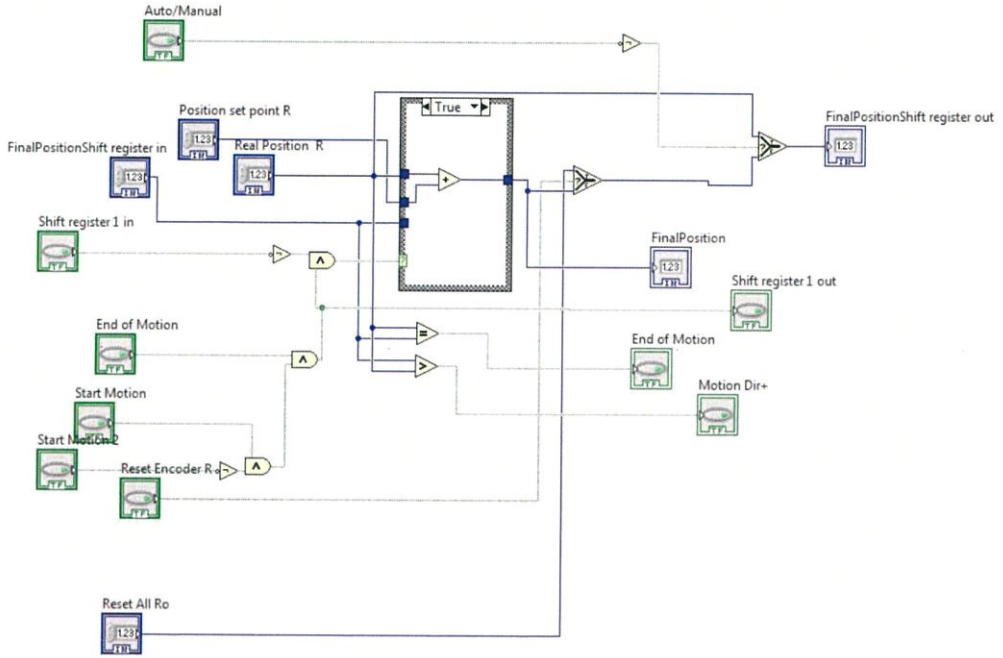


รูปที่ 3.50 ส่วนการสร้างสัญญาณพัลส์



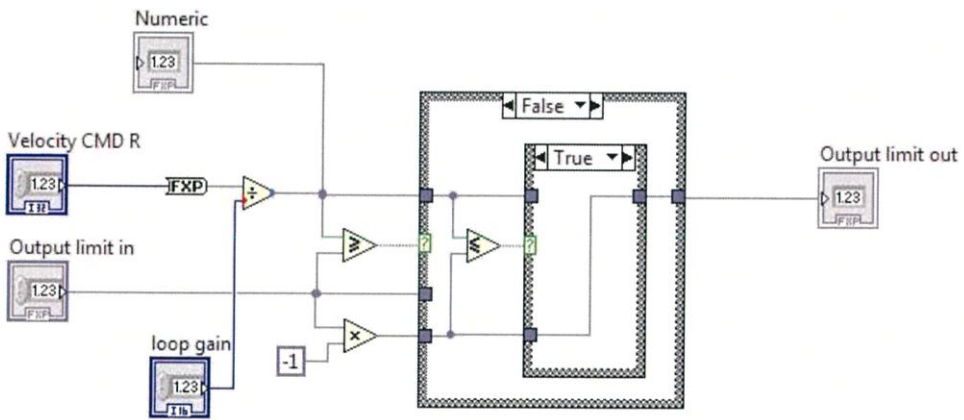
รูปที่ 3.51 ส่วนการจำลองการเคลื่อนที่

จากรูปที่ 3.50 แสดงส่วนการสร้างสัญญาณพัลส์จากค่าความเร็วที่ได้รับจากชุดคำสั่งเส้นทางการเคลื่อนที่ เพื่อเป็นใช้เป็นความเร็วในการจำลองการเคลื่อนที่ สัญญาณพัลส์ที่ได้จะถูกส่งไปยังส่วนการจำลองการเคลื่อนที่ในรูปที่ 3.51



รูปที่ 3.52 ส่วนกำหนดตำแหน่งการเคลื่อนที่

จากรูป 3.52 แสดงซอฟต์แวร์การกำหนดตำแหน่งที่ต้องการเคลื่อนที่ไป โดยการทำงานจะทำงานตามคำสั่งที่ส่งมาจากซอฟต์แวร์บน Real-Time



รูปที่ 3.53 ส่วนควบคุมขนาดสัญญาณให้แก่วงจรขับเคลื่อน

ในรูปที่ 3.53 แสดงส่วนควบคุมสัญญาณให้แก่วงจรขับเคลื่อนเกิดจากความแตกต่างระหว่างสัญญาณป้อนกลับจากเอนโคเดอร์กับค่าการเคลื่อนที่ที่จำลองขึ้น เข้าสู่การควบคุมแบบ PID ซึ่งในซอฟต์แวร์นี้ใช้การควบคุมแบบ P ในการควบคุมเพียงเท่านั้น

บทที่ 4

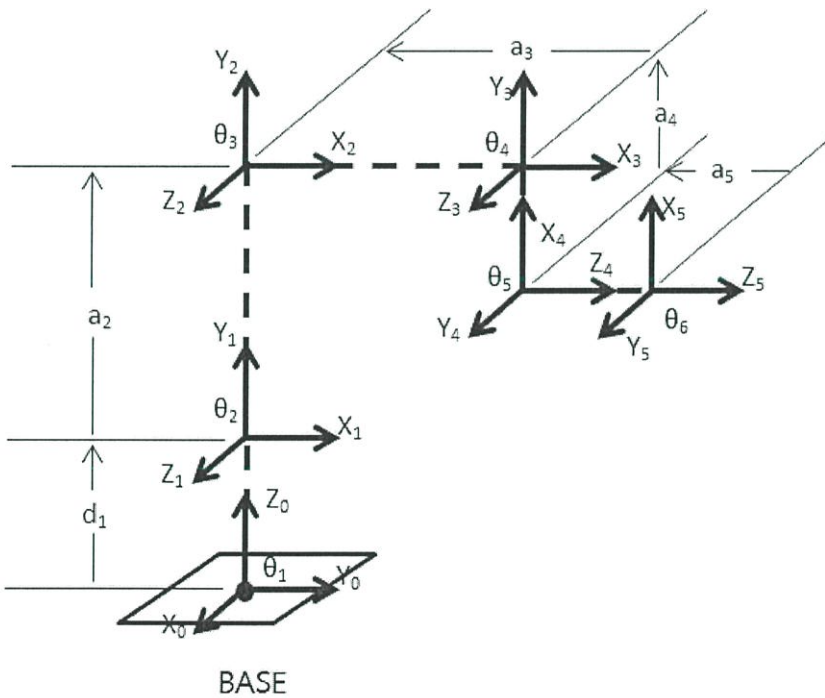
ผลการทดลอง

4.1 กล่าวนำ

ในบทนี้กล่าวถึงผลของการศึกษาและดำเนินการทั้งหมดที่เกิดขึ้น สมการที่ใช้ในการเคลื่อนที่แขนหุ่นยนต์ การปรับปรุงแก้ไขส่วนต่างๆ ทางด้านฮาร์ดแวร์ การสร้างและออกแบบซอฟต์แวร์ที่ใช้งาน รวมทั้งการทดลองต่างๆ ที่ทำให้ซอฟต์แวร์ที่ใช้มีความสมบูรณ์

4.2 สมการที่ใช้ในการเคลื่อนที่ของแขนหุ่นยนต์

4.2.1 สมการ Forward Kinematics ของแขนหุ่นยนต์



รูปที่ 4.1 Link Coordinates ของแขนหุ่นยนต์

จากการศึกษาเมตริกซ์การถ่ายโอนพิกัดโฮโมจีเนียส จะสร้างเมตริกซ์สมการแขนหุ่นยนต์ได้

ดังนี้

$$T_{\text{Base}}^R = T_1 = \begin{bmatrix} -s_1 & 0 & c_1 & 0 \\ c_1 & 0 & s_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$T_R^L = T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$T_L^U = T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$$T_U^{\theta_{\text{Link}}} = T_4 = \begin{bmatrix} -s_4 & 0 & c_4 & a_4 c_4 \\ c_4 & 0 & s_4 & a_4 s_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$$T_{\theta_{\text{Link}}}^{\theta} = T_5 = \begin{bmatrix} 0 & 0 & -1 & a_5 c_5 \\ -s_5 & c_5 & 0 & a_5 s_5 \\ c_5 & s_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

โดยที่ T หมายถึง เมตริกซ์การถ่ายโอนเลื่อนตำแหน่งโฮโมจีเนียส

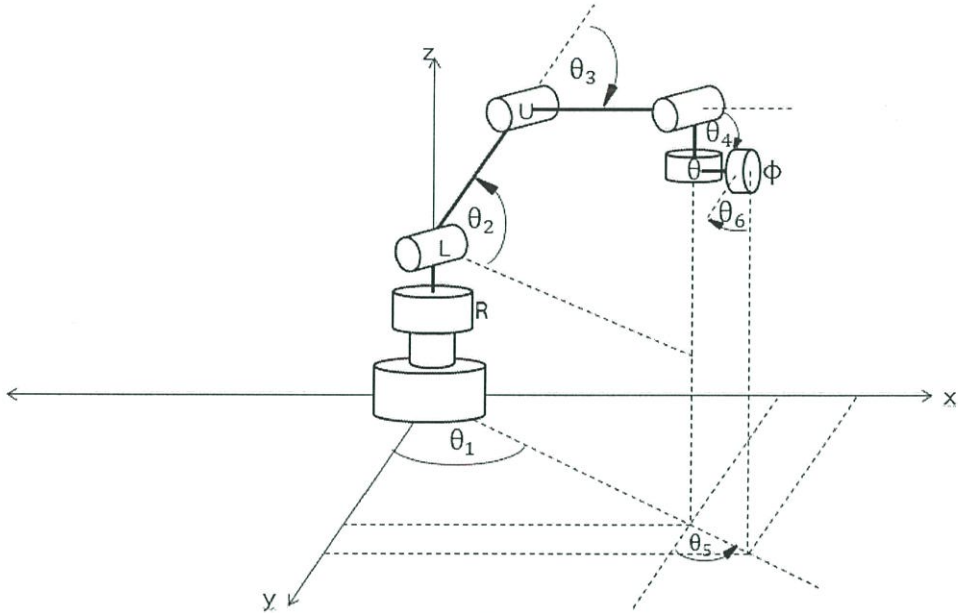
S หมายถึง $\sin \theta$

C หมายถึง $\cos \theta$

การถ่ายโอนเมตริกซ์โครงพิกัดจะเริ่มจากพิกัดปลาย End - Effect ถ่ายโอนย้อนกลับมายังพิกัดเริ่มต้นที่ฐานของแขนหุ่นยนต์ ดังนี้

$$T_{\text{Base}}^{\theta} = (T_{\text{Base}}^R)(T_R^L)(T_L^U)(T_U^{\theta_{\text{Link}}})(T_{\theta_{\text{Link}}}^{\theta}) \quad (4.6)$$

4.2.2 สมการ Inverse Kinematics ของแขนหุ่นยนต์



รูปที่ 4.2 มุมที่เกิดขึ้นจากแขนหุ่นยนต์

กำหนดให้ ex คือ พิกัด End - Effect ที่ตำแหน่งแกน x
 ey คือ พิกัด End - Effect ที่ตำแหน่งแกน y
 ez คือ พิกัด End - Effect ที่ตำแหน่งแกน z

และ $rx = ex - L_5 \sin \theta_5$ (4.7)

$$ry = ey - L_5 \cos \theta_5 \quad (4.8)$$

$$rz = ez + L_4 \quad (4.9)$$

ดังนั้นที่ θ ต่างๆ จะมีมุมตามสมการ ดังนี้

$$\theta_1 = \tan^{-1} \left(\frac{ex}{ey} \right) \quad (4.10)$$

$$\theta_2 = \tan^{-1} \left(\frac{rz - d_1}{\sqrt{rx^2 + ry^2}} \right) - \tan^{-1} \left(\frac{a_3 \sin \theta_3}{a_2 + a_3 (\cos \theta_3)} \right) \quad (4.11)$$

$$\theta_3 = - \left(\cos^{-1} \left(\frac{rx^2 + ry^2 + ((rz - d_1)^2 - a_2^2 - a_3^2)}{2 \cdot a_2 \cdot a_3} \right) \right) \quad (4.12)$$

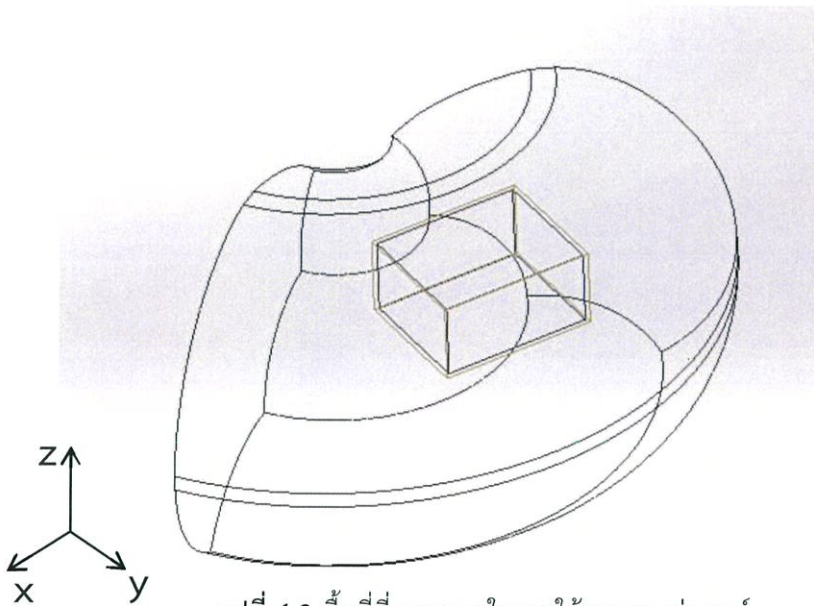
$$\theta_4 = -(180 - (90 - \theta_2) + |\theta_3|) \quad (4.13)$$

$$\theta_5 = \theta_1 \quad (4.14)$$

θ_5 จะมีค่าเท่ากับ θ_1 เสมอในทุกกรณีเมื่อเทียบกับพิกัดอ้างอิงฐาน เนื่องจากเป็นแกนที่หันทางเดียวกับฐานและกำหนดให้หน้าตรงเสมอ เมื่อเทียบกับพิกัดโครงอ้างอิงของ θ จะมีการเปลี่ยนแปลงเป็น 0 องศา และ θ_6 เป็นมุมที่กำหนดเองจากความต้องการใช้งาน

4.3 พื้นที่ที่เหมาะสมในการใช้งานของแขนหุ่นยนต์

เมื่อทราบพื้นที่ความสามารถของการเคลื่อนที่แขนหุ่นยนต์นั้น จึงได้กำหนดพื้นที่การใช้งานของแขนหุ่นยนต์ขึ้นสำหรับใช้งาน (Working Area) ซึ่งเป็นการกำหนดจุดที่เหมาะสมสำหรับการใช้งานมากที่สุด ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 พื้นที่ที่เหมาะสมในการใช้งานแขนหุ่นยนต์

โดย กำหนดขนาดการใช้งานตามแกนต่างๆ ดังนี้

แกน x สามารถใช้งานได้ตั้งแต่ -300 mm. ถึง +300 mm.

แกน y สามารถใช้งานได้ตั้งแต่ +700 mm. ถึง +1200 mm.

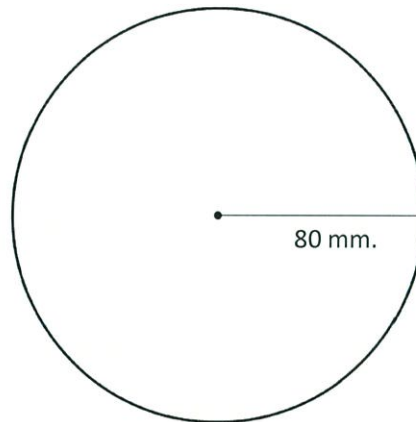
แกน z สามารถใช้งานได้ตั้งแต่ +800 mm. ถึง +1100 mm.

4.4 เส้นทางเคลื่อนที่ที่ออกแบบในโปรแกรม RobotStudio

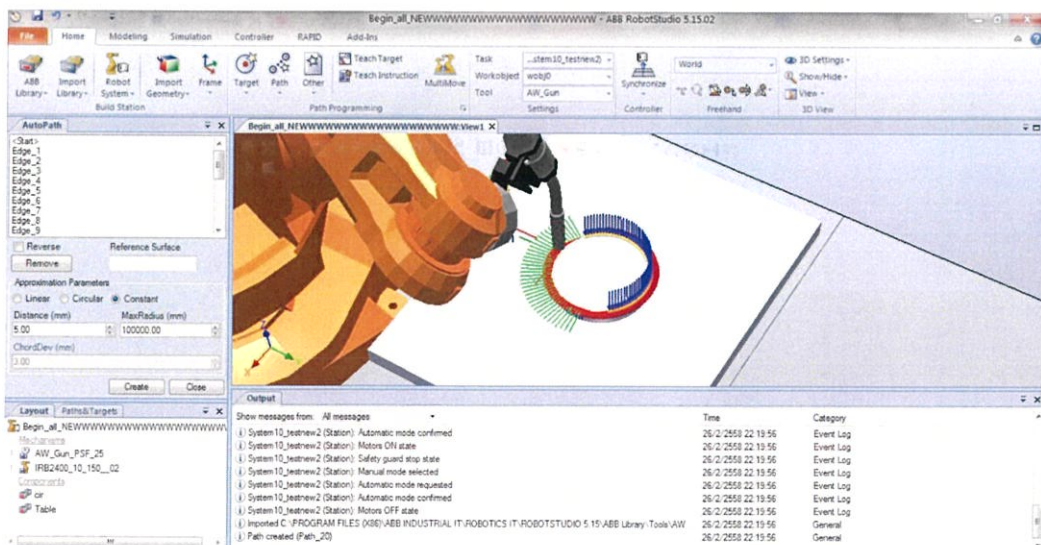
รูปแบบเส้นทางเคลื่อนที่ที่ออกแบบในโปรแกรม RobotStudio ที่ออกแบบขึ้นเพื่อใช้เป็นรูปแบบเส้นทางให้แก่ซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ให้ทำงานตามเส้นทางเคลื่อนที่ที่ออกแบบขึ้น ในปฏิญานิพนธ์นี้ได้ออกแบบตัวอย่างเส้นทางเคลื่อนที่สำหรับซอฟต์แวร์ 3 รูปแบบ ซึ่งเรียกเป็น Module มีดังต่อไปนี้

4.4.1 Module 1: เส้นทางเคลื่อนที่แบบวงกลม

การออกแบบเส้นทางเคลื่อนที่เป็นวงกลมของ Module 1 การเคลื่อนที่ของปลาย Tool จะเคลื่อนที่ตามระนาบ x และ y เพื่อทดสอบการเคลื่อนที่ให้สอดคล้องกัน



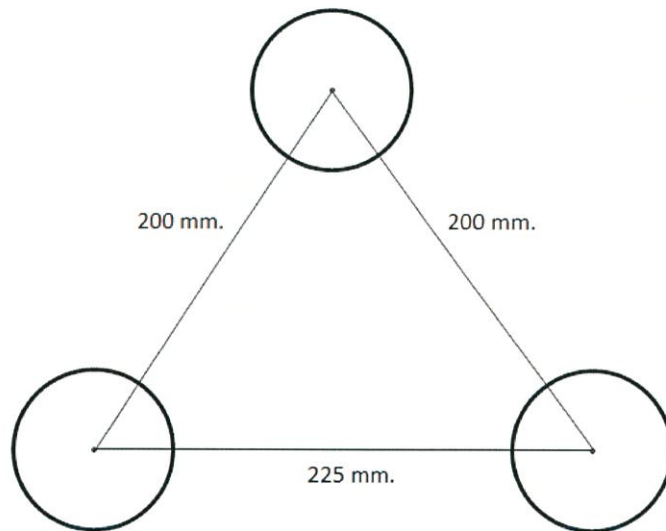
รูปที่ 4.4 แบบเส้นทางเคลื่อนที่ของ Module 1



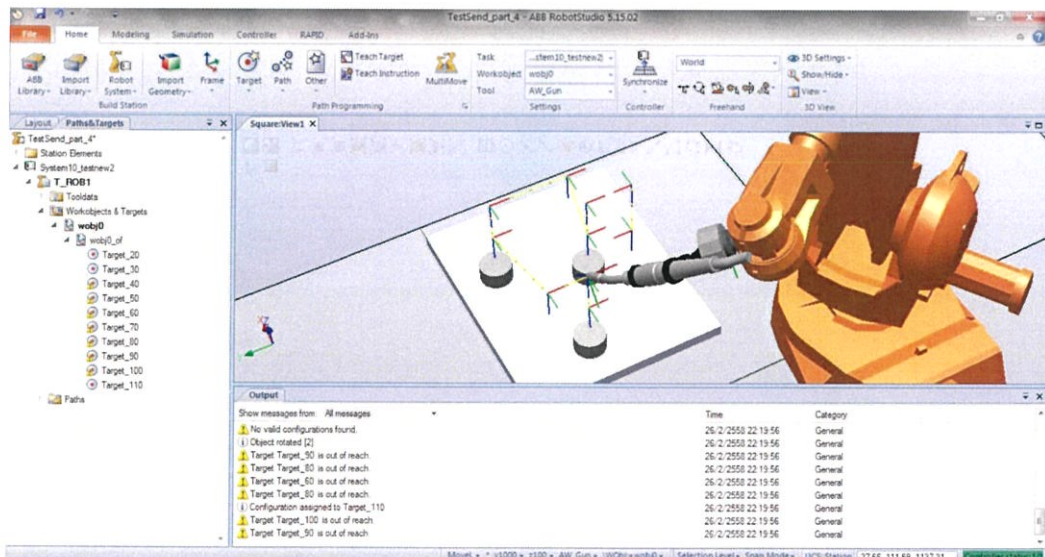
รูปที่ 4.5 แบบเส้นทางเคลื่อนที่ของ Module 1 บน RobotStudio

4.4.2 Module 2: เส้นทางเคลื่อนที่แบบจำลอง Pick-and-Place

การออกแบบเส้นทางเคลื่อนที่ของ Module 2 เป็นเส้นทางเคลื่อนที่เพื่อทดสอบความแม่นยำในการเคลื่อนที่ไปที่พิกัดต่างๆ โดยการกำหนดให้แขนหุ่นยนต์เคลื่อนที่ลงทั้งหมด 3 ตำแหน่ง ดังรูปที่ 4.6



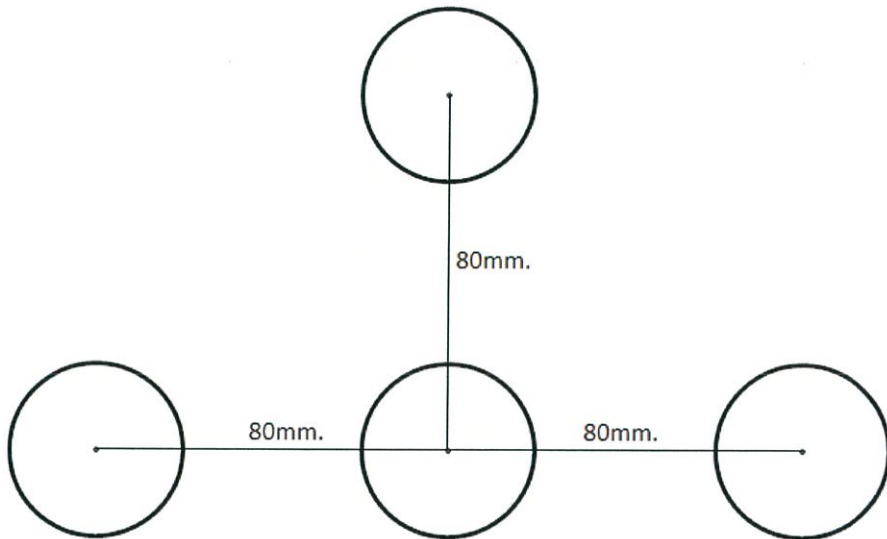
รูปที่ 4.6 การวางตำแหน่งวัตถุของ Module 2



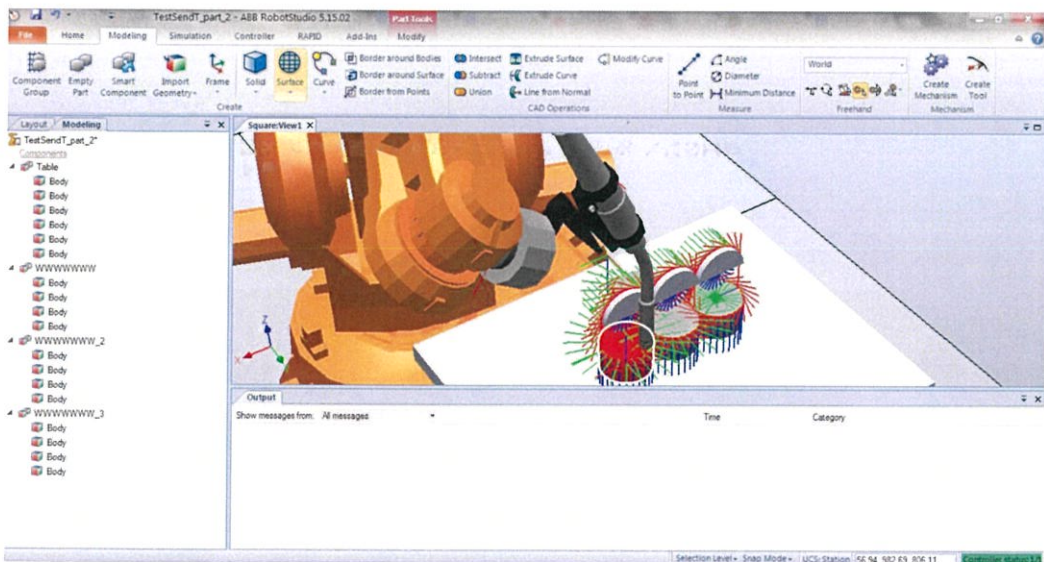
รูปที่ 4.7 แบบเส้นทางเคลื่อนที่ของ Module 2 บน RobotStudio

4.4.3 Module 3: เส้นทางการเคลื่อนที่แบบหลบหลีกสิ่งกีดขวาง

การออกแบบเส้นทางการเคลื่อนที่ของ Module 3 เป็นออกแบบขั้นเพื่อเพิ่มความซับซ้อนของการเคลื่อนที่ มีการเคลื่อนที่ทั้งในระนาบ x, y, z เป็นเส้นโค้งไปมา



รูปที่ 4.8 การวางตำแหน่งวัตถุกีดขวางของ Module 3

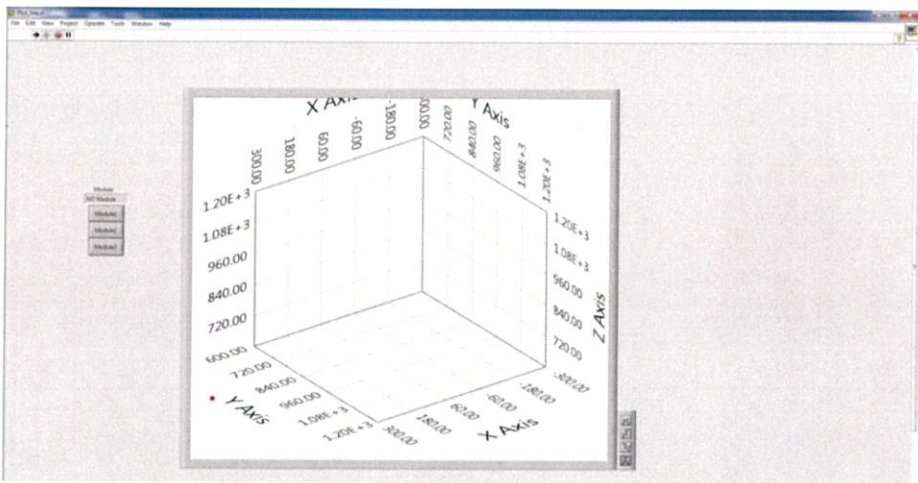


รูปที่ 4.9 แบบเส้นทางการเคลื่อนที่ของ Module 3 บน RobotStudio

4.5 ส่วนการแสดงผลซอฟต์แวร์

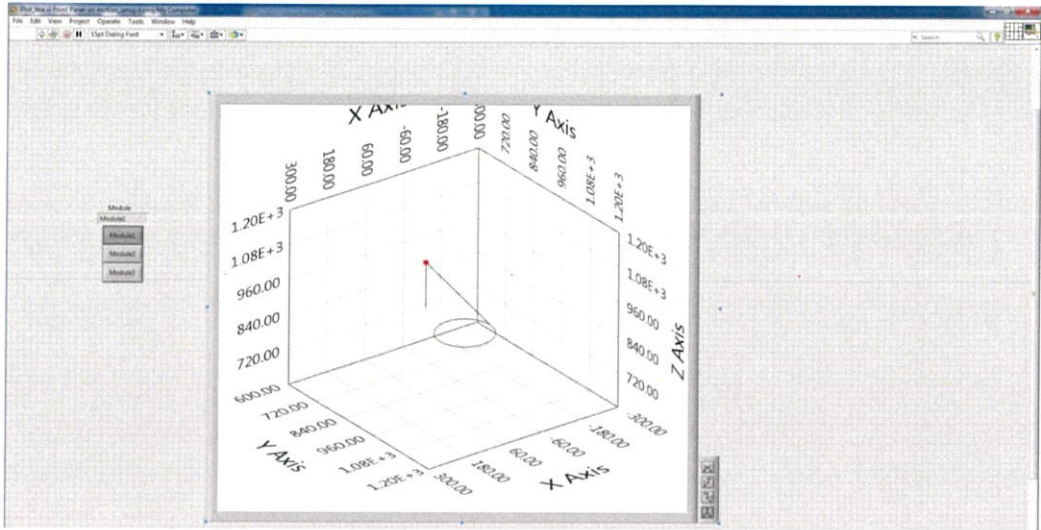
ในหัวข้อนี้จะกล่าวถึงหน้าจอแสดงผล GUI (Graphic User Interface) ที่สำหรับใช้ควบคุมซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ และการจำลองเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ ซึ่งการแสดงผลการใช้งานนี้แสดงผลจากการสร้างซอฟต์แวร์ระบบควบคุมและจำลองแขนหุ่นยนต์ลงในส่วนบล็อกไดอะแกรมผ่านโปรแกรม LabVIEW มีส่วนแสดงผลออกเป็น 2 ส่วนหลัก ดังนี้

4.5.1 ส่วนประมวลผลบน User Interface VI

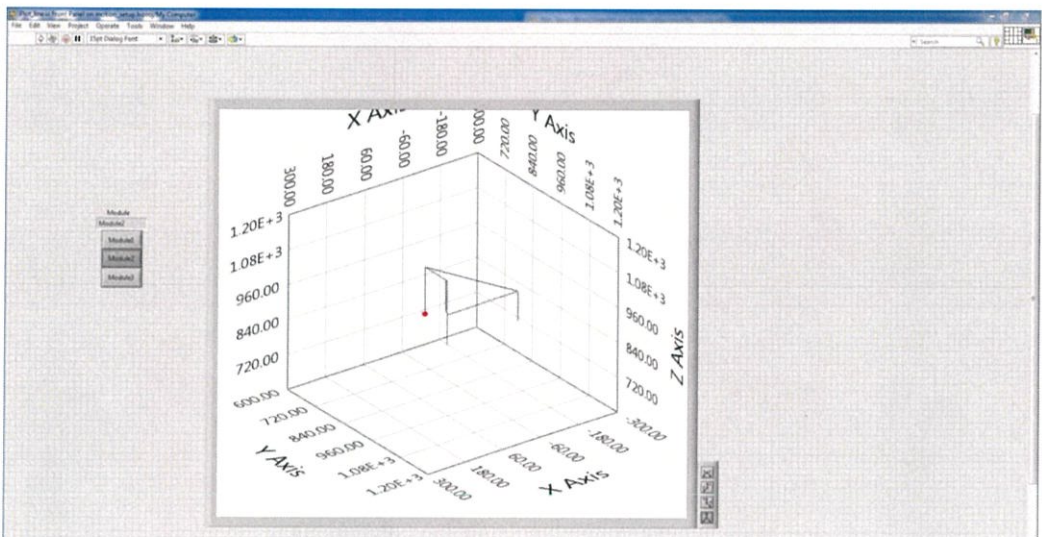


รูปที่ 4.10 ส่วนหน้าจอในการแสดงการจำลองเส้นทางการเคลื่อนที่ Module ต่างๆ

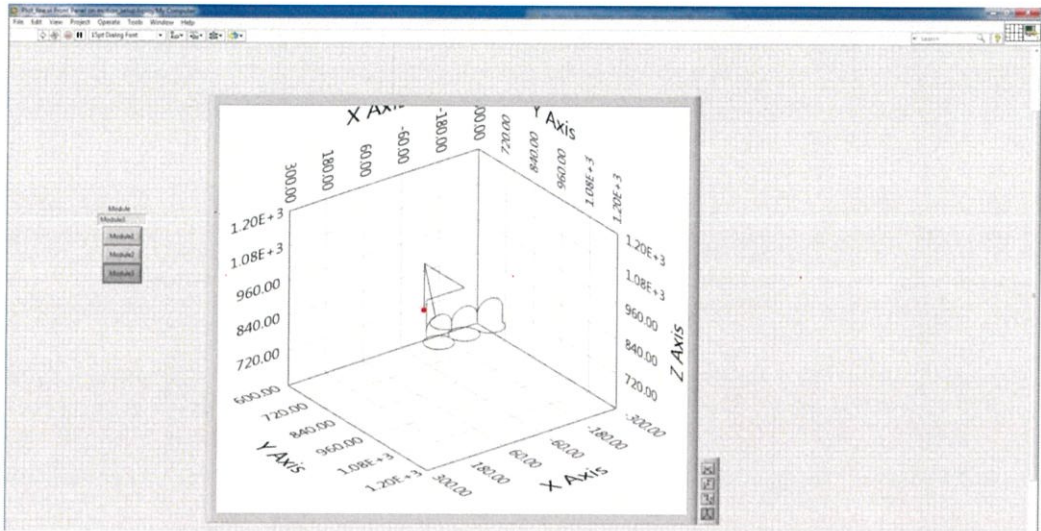
จากรูปที่ 4.10 แสดงหน้าจอส่วนการจำลองเส้นทางการเคลื่อนที่ตามชุดคำสั่งที่สร้างจากโปรแกรม RobotStudio โดยการจำลองเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์จะจำลองตาม Module ที่สร้างขึ้นไว้แล้ว ภาพที่แสดงขึ้นจะเป็นภาพเคลื่อนไหว สามารถเลือกและปรับเปลี่ยนมุมมองเส้นทางตามที่ต้องการแสดงได้ โดยแต่ละ Module ภาพจะแสดงในรูปที่ 4.11 – 4.13



รูปที่ 4.11 ส่วนหน้าจอในการแสดงเส้นทางการเคลื่อนที่ Module 1



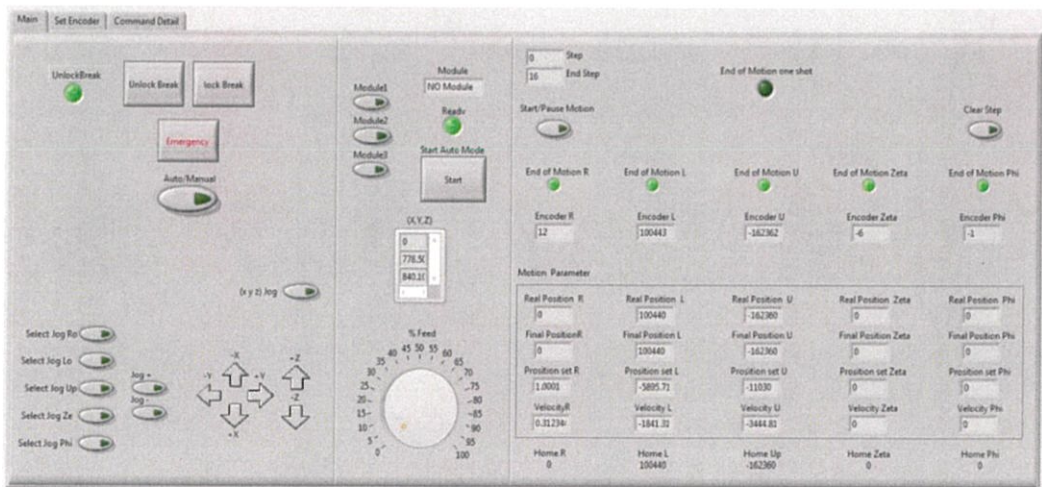
รูปที่ 4.12 ส่วนหน้าจอในการแสดงเส้นทางการเคลื่อนที่ Module 2



รูปที่ 4.13 ส่วนหน้าจอในการแสดงเส้นทางการเคลื่อนที่ Module 3

จากการแสดงการจำลองเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ จะแสดงส่วนที่ปลาย Tool การแสดงการเคลื่อนที่เป็นลักษณะเป็นภาพเคลื่อนไหว แตกต่างจากการจำลองการเคลื่อนที่จากโปรแกรม MATLAB ที่แสดงเป็นจุดทั้งหมด ในการจำลองการเคลื่อนที่ที่จะช่วยทำให้ทราบเส้นทางที่แขนหุ่นยนต์ทำงานที่ถูกต้อง เป็นการตรวจสอบการเคลื่อนที่ และเพื่อความปลอดภัยให้กับการใช้งานแขนหุ่นยนต์

4.5.2 ส่วนประมวลผลบน Real-Time VI

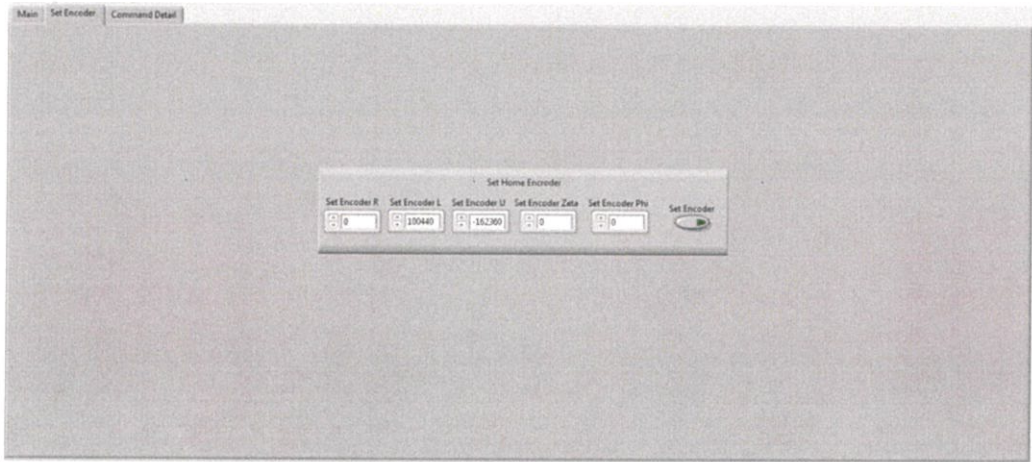


รูปที่ 4.14 ส่วนแสดงผลหลักในการใช้งานซอฟต์แวร์

จากรูปที่ 4.14 เป็นส่วนที่แสดงหน้าจอแสดงผลหลักในการใช้ซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ ซึ่งมีส่วนใช้งานแบ่งออกเป็น Auto Mode และ Manual Mode

Manual Mode ทำหน้าที่ควบคุมการทำงานของแขนหุ่นด้วยผู้ใช้งานเอง โดยสามารถเคลื่อนที่ภายในวิธีการนี้ได้ 2 ลักษณะ ดังนี้

1. Select Jog คือ การเคลื่อนที่ของแขนหุ่นยนต์ทีละข้อต่อ
2. (x, y, z) Jog คือ การเคลื่อนที่ของแขนหุ่นยนต์ไปตามแกน x, y หรือ z



รูปที่ 4.15 ส่วนการตั้งค่าเริ่มต้นให้กับเอนโคเดอร์

จากรูปที่ 4.15 ส่วนนี้แสดงการตั้งค่าพิกัดเริ่มต้น (Home) ให้แก่เอนโคเดอร์ และเป็นการตั้งจุดเริ่มต้นให้กับแขนหุ่นยนต์เช่นกัน ค่าเริ่มต้นของเอนโคเดอร์ที่ตั้งขึ้น ดังตารางที่ 4.1

ตารางที่ 4.1 ค่าเริ่มต้นของเอนโคเดอร์และแขนหุ่นยนต์

ข้อต่อ	ตำแหน่งHome	สมการเปลี่ยนค่ามุมเป็นเอนโคเดอร์	ค่าเอนโคเดอร์
R	0	$(0) \times 1196.343$	0
L	90	$(90) \times 1116$	100440
U	-120	$(-120) \times 1353$	-162360
Zeta	0	$(0) \times 222.14$	0
Phi	0	$(0) \times 240.14$	0

Position set set point R		Position set set point L		Position set set point U		Position set set point Zeta		Position set set point Phi	
0	0	22.2063	1291.38	0	-38.7698	-5746.39	0	0	0
0	0	5919.59	3699.74	0	10903.7	6871.05	0	0	0
0	0	-2.6763	-42.20.03	0	10778.5	4988.45	0	0	0
0	0	179.629	1246.3	0	984.675	11188.1	0	0	0
0	0	127.633	1596.07	0	914.237	11432.7	0	0	0
0	0	73.1449	914.693	0	924.862	11565.6	0	0	0
0	0	16.573	207.249	0	-926.395	-11584.8	0	0	0
0	0	-41.6084	-520.322	0	918.761	11488.3	0	0	0
0	0	100.855	1261.21	0	991.968	11279.3	0	0	0
0	0	-160.349	-2007.7	0	478.107	-10955.9	0	0	0
0	0	-220.007	-2751.24	0	841.358	10523.4	0	0	0
0	0	-278.479	-3481.44	0	797.903	-9979.07	0	0	0
0	0	-335.161	-4191.26	0	-746.372	-9333.54	0	0	0
0	0	-389.21	-4867.15	0	686.954	8590.5	0	0	0
0	0	-439.759	-5499.27	0	620.287	-7756.82	0	0	0
0	0	-485.942	-6076.81	0	547.013	-6840.51	0	0	0
0	0	-528.921	-6599.26	0	467.86	-5850.69	0	0	0
0	0	-561.913	-7026.84	0	387.64	-4797.5	0	0	0
0	0	-590.217	-7380.79	0	295.236	-3691.96	0	0	0
0	0	-611.249	-7643.8	0	203.593	-2545.97	0	0	0
0	0	-624.562	-7810.27	0	-109.708	-1371.92	0	0	0
0	0	-629.866	-7876.61	0	10.6225	132.836	0	0	0
0	0	-627.847	-7841.35	0	-58.6105	-732.936	0	0	0
0	0	-618.166	-7793.28	0	-127.008	-1368.36	0	0	0
0	0			0	80.6474	1008.51	0	0	0
0	0			0	175.015	2188.59	0	0	0

รูปที่ 4.16 ชุดคำสั่งในการเคลื่อนที่ของแต่ละข้อต่อใน Auto Mode

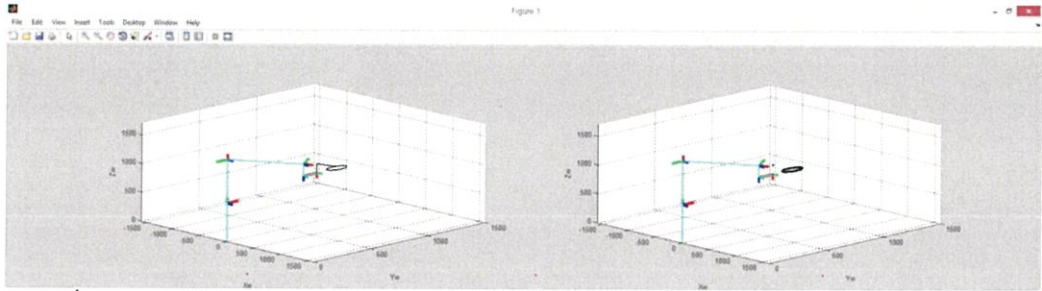
จากส่วนที่แสดงชุดการเคลื่อนที่ของแต่ละข้อต่อในรูปที่ 4.16 จะแสดงค่าพอลัสสำหรับการเคลื่อนที่ของแต่ละข้อต่อ ค่าพอลัสในการเคลื่อนที่ที่จะแสดงก็ต่อเมื่อการใช้งานอยู่ใน Auto Mode ที่ได้จากการแปลงชุดคำสั่งที่ได้จากพิกัดตำแหน่ง

4.6 การทดสอบซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์

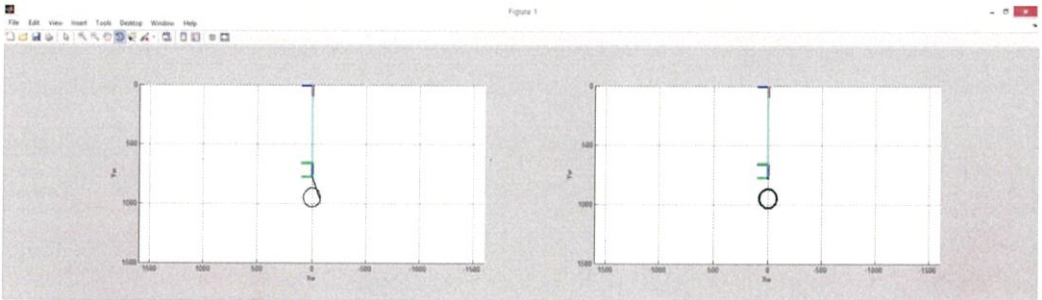
4.6.1 การทดสอบเพื่อหาความถูกต้องของสมการการเคลื่อนที่ของแขนหุ่นยนต์

การทดสอบเพื่อหาความถูกต้องของสมการการเคลื่อนที่ของแขนหุ่นยนต์ เพื่อทดสอบความถูกต้องของสมการการเคลื่อนที่ของแขนหุ่นยนต์ว่าสมการที่ออกแบบขึ้นจากการนำทฤษฎีการสร้างสมการมาใช้กับแขนหุ่นยนต์ ตรงกับงานที่เกิดขึ้น โดยผลลัพธ์ที่เป็นค่ามุมข้อต่อ ได้จาก Inverse Kinematics และผลลัพธ์ที่เป็นตำแหน่งจาก Forward Kinematics และการเปลี่ยนไปยังมุมที่ถูกต้องเมื่อนำสมการเข้าสู่เมตริกซ์

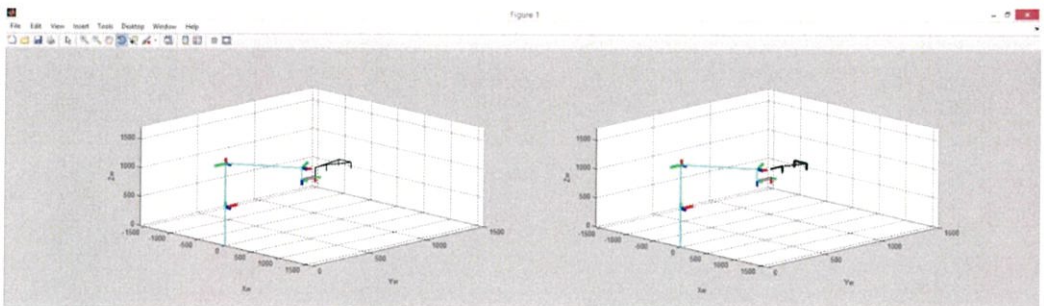
การทดสอบทำโดยการใช้ไฟล์ข้อมูลเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ที่สร้างขึ้นจากโปรแกรม RobotStudio มาเข้าสมการบนโปรแกรม MATLAB ที่สร้างขึ้น เพื่อตรวจสอบเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ที่จำลองขึ้นบนโปรแกรม MATLAB มาเปรียบเทียบกับความถูกต้องกับเส้นทางการออกแบบไว้ในโปรแกรม RobotStudio



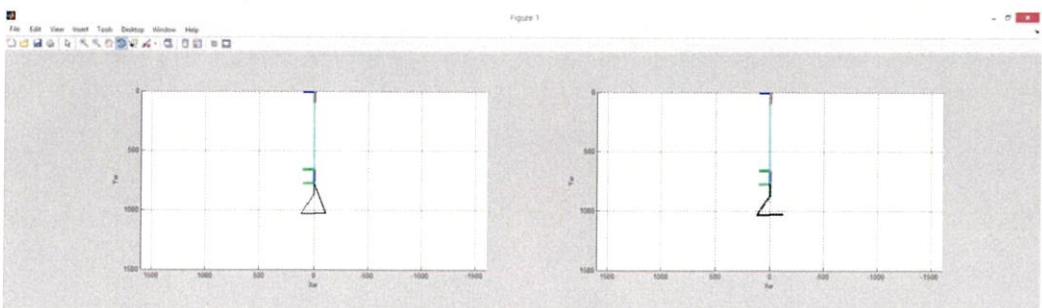
รูปที่ 4.17 ผลลัพธ์จาก RobotStudio (ด้านซ้าย) และบน MATLAB (ด้านขวา) ของ Module 1



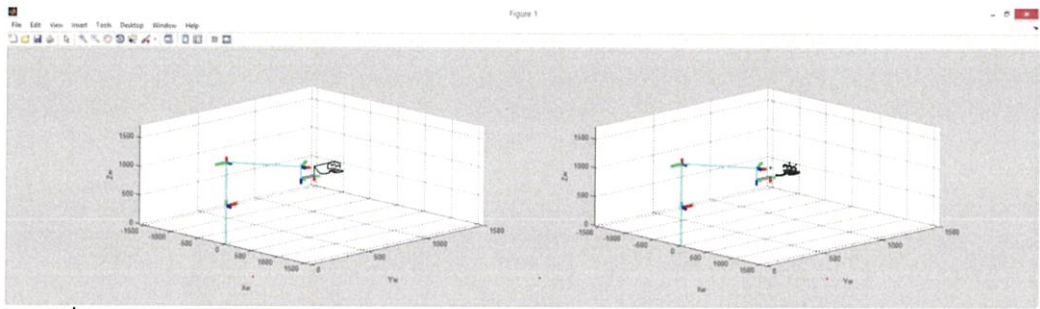
รูปที่ 4.18 ผลลัพธ์ด้านบนจาก RobotStudio และบน MATLAB ของ Module 1



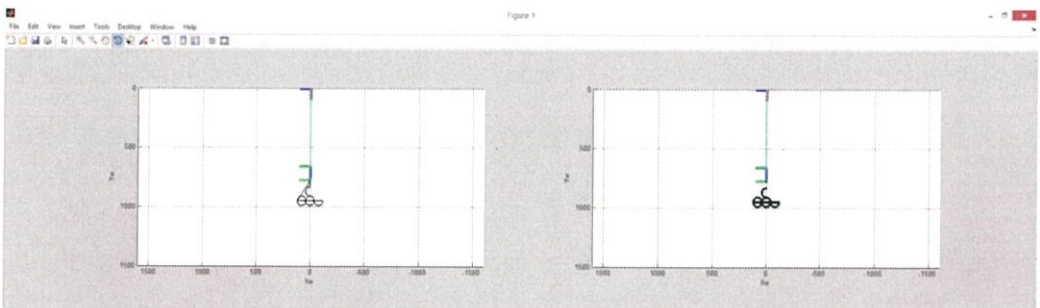
รูปที่ 4.19 ผลลัพธ์จาก RobotStudio (ด้านซ้าย) และบน MATLAB (ด้านขวา) ของ Module 2



รูปที่ 4.20 ผลลัพธ์ด้านบนจาก RobotStudio และบน MATLAB ของ Module 2



รูปที่ 4.21 ผลลัพธ์จาก RobotStudio (ด้านซ้าย) และบน MATLAB (ด้านขวา) ของ Module 3



รูปที่ 4.22 ผลลัพธ์ด้านบนจาก RobotStudio และบน MATLAB ของ Module 3

ตารางที่ 4.2 ผลการทดสอบพิกัดตำแหน่งจาก RobotStudio และผลลัพธ์พิกัดจาก MATLAB

รูปแบบการเคลื่อนที่	ระบุตำแหน่งจาก RobotStudio	ระบุตำแหน่งจาก Kinematics
Module 1	ทำได้ถูกต้อง	ทำได้ถูกต้อง
Module 2	ทำได้ถูกต้อง	ทำได้ถูกต้อง
Module 3	ทำได้ถูกต้อง	ทำได้ถูกต้อง

จากการจำลองพิกัดจุดของเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์จะช่วยให้ทราบถึงความถูกต้องการประมวลผลสมการ Kinematics ว่าสมการที่ออกแบบนั้นใช้งานได้และถูกต้อง เป็นการลดความผิดพลาดที่อาจส่งผลให้แขนหุ่นยนต์เกิดความเสียหายหรือเกิดอันตรายที่เกิดขึ้นต่อบุคคลที่แขนหุ่นยนต์ทำงานผิดพลาด

4.6.2 การทดลองการหาค่าจำนวนรอบพัลส์ต่อหนึ่งองศา

เนื่องจากเอนโคเดอร์เป็นเอนโคเดอร์ที่ไม่สามารถหาข้อมูลได้ ดังนั้นจึงจำเป็นต้องหาค่าจำนวนรอบของพัลส์ต่อมุมหนึ่งองศาเพื่อนำค่าของเอนโคเดอร์มาใช้งานได้อย่างถูกต้อง โดยการนับค่าเอนโคเดอร์เพื่อนับรอบพัลส์ที่เกิดขึ้นในขนาดมุมต่างๆ โดยการใช้วิธีการปลดเบรกข้อต่อแต่ละข้อต่อของแขนหุ่นยนต์ และอ่านค่าพัลส์เอนโคเดอร์ที่เปลี่ยนไปมุมต่างๆ สำหรับหาค่าพัลส์ที่เกิดใน 1 องศา

ตารางที่ 4.3 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการหมุนข้อต่อ R เมื่อมุมองศาเปลี่ยนไป 90 องศา

จำนวนครั้ง	Pulse ที่เกิดขึ้น	Pulse ต่อ 1 องศา
1	108777	1208.633
2	108224	1202.489
3	105993	1177.7
4	106940	1188.222
5	108007	1200.078
6	107966	1199.622
7	107717	1196.856
8	107718	1196.867
9	107738	1197.089
10	107629	1195.878

$$\text{ค่าเฉลี่ยพัลส์ต่อหนึ่งองศา } \bar{x} = \frac{\sum x}{n} = \frac{10512.9}{10} = 1051.29 \text{ พัลส์/องศา}$$

ตารางที่ 4.4 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการทดลองหมุนข้อต่อ L เมื่อมุมมองคาเปลี่ยนไป 45 องศา

จำนวนครั้ง	Pulse ที่เกิดขึ้น	Pulse ต่อ 1 องศา
1	51509	1144.644
2	51365	1141.444
3	51237	1138.6
4	51068	1134.844
5	51315	1140.333
6	51242	1138.711
7	51125	1136.111
8	51376	1141.689
9	51232	1138.489
10	51131	1136.244

$$\text{ค่าเฉลี่ยพัลส์ต่อหนึ่งองศา } \bar{x} = \frac{\sum x}{n} = \frac{11391.11}{10} = 1139.11 \text{ พัลส์/องศา}$$

ตารางที่ 4.5 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการทดลองหมุนข้อต่อ U มุมองศาเปลี่ยนไป 30 องศา

จำนวนครั้ง	Pulse ที่เกิดขึ้น	Pulse ต่อ 1 องศา
1	63150	1052.5
2	63172	1052.867
3	62914	1048.567
4	63175	1052.917
5	63040	1050.667
6	63356	1055.933
7	62938	1048.967
8	62994	1049.9
9	62824	1047.067
10	63211	1053.517

$$\text{ค่าเฉลี่ยพัลส์ต่อหนึ่งองศา } \bar{x} = \frac{\sum x}{n} = \frac{10512.9}{10} = 1051.29 \text{ พัลส์/องศา}$$

ตารางที่ 4.6 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการทดลองหมุนข้อต่อ θ มุมองศาเปลี่ยนไป 360 องศา

จำนวนครั้ง	Pulse ที่เกิดขึ้น	Pulse ต่อ 1 องศา
1	79976	222.1556
2	79957	222.1028
3	79961	222.1139
4	79939	222.0528
5	79935	222.0417
6	79984	222.1778
7	79991	222.1972
8	79992	222.2
9	79983	222.175
10	80003	222.2306

$$\text{ค่าเฉลี่ยพัลส์ต่อหนึ่งองศา } \bar{x} = \frac{\sum x}{n} = \frac{2221.447}{10} = 222.14 \text{ พัลส์/องศา}$$

ตารางที่ 4.7 ผลการทดลองนับพัลส์ที่เกิดขึ้นในการทดลองหมุนข้อต่อ ϕ เมื่อมุมองศาเปลี่ยนไป 90 องศา

จำนวนครั้ง	Pulse ที่เกิดขึ้น	Pulse ต่อ 1 องศา
1	21600	240
2	21599	239.9889
3	21630	240.3333
4	21561	239.5667
5	21587	239.8556
6	21647	240.5222
7	21594	239.9333
8	21588	239.8667
9	21650	240.5556
10	21671	240.7889

$$\text{ค่าเฉลี่ยพัลส์ต่อหนึ่งองศา } \bar{x} = \frac{\sum x}{n} = \frac{2401.411}{10} = 240.14 \text{ พัลส์/องศา}$$

หลังจากการทดสอบ เมื่อนำพัลส์ที่ทดลองได้ไปใช้ในซอฟต์แวร์ พบว่าเมื่อป้อนค่าพัลส์ที่ทดลองไป ข้อต่อที่มีขนาดใหญ่มีความคลาดเคลื่อนเกิดขึ้น ข้อต่อที่มีค่าความคลาดเคลื่อนที่ได้แก่ ข้อต่อ U และข้อต่อ L เมื่อสังเกตจึงได้พบว่าเกิดจากการที่ใช้การยกขึ้นลงอาจทำให้มีความคลาดเคลื่อนไปจากแรงที่พุงทำให้ค่าจากเอนโคเดอร์แกว่งไปมา ดังนั้นจึงได้จึงทดลองได้ใช้ซอฟต์แวร์ทดลองปรับได้ค่าพัลส์ที่ใช้งานได้ดีที่สุดทั้ง 5 ข้อต่อ ดังนี้

ข้อต่อ R เปลี่ยนมุมไป 1 องศา มีค่าพัลส์เท่ากับ 1196 พัลส์/องศา

ข้อต่อ L เปลี่ยนมุมไป 1 องศา มีค่าพัลส์เท่ากับ 1116 พัลส์/องศา

ข้อต่อ U เปลี่ยนมุมไป 1 องศา มีค่าพัลส์เท่ากับ 1353 พัลส์/องศา

ข้อต่อ θ เปลี่ยนมุมไป 1 องศา มีค่าพัลส์เท่ากับ 222 พัลส์/องศา

ข้อต่อ ϕ เปลี่ยนมุมไป 1 องศา มีค่าพัลส์เท่ากับ 240 พัลส์/องศา

4.6.3 การทดลองหาอัตราการเปลี่ยนแปลงของข้อต่อเมื่ออีกข้อต่อมีการเปลี่ยนแปลง

เนื่องจากกลไกของแขนหุ่นยนต์ที่ทำให้การเคลื่อนที่ของข้อต่อแกน L มีผลทำให้ข้อต่อของแกน U หมุนตาม และเคลื่อนที่ของข้อต่อแกน U หมุนเพียงข้อเดียว จะทำให้ข้อต่อแกน θ หมุนตาม แต่การนับของเอนโคเดอร์ไม่เปลี่ยนแปลงตามมุมที่ข้อต่อเปลี่ยนไป ทำให้เอนโคเดอร์นับค่าผิดพลาดทำงานไม่ตรงกับการทำงานจริง ดังนั้นจึงทำการทดสอบหาอัตราส่วนการเปลี่ยนแปลงของทั้งสองข้อต่อเพื่อรักษามุมของข้อต่อไว้ไม่ให้ทำงานผิดพลาด ด้วยการวัดการเปลี่ยนแปลงของมุมทั้งสองได้ ดังนี้

4.6.3.1 การหาอัตราส่วนระหว่างข้อต่อ L กับข้อต่อ U

การทดสอบว่า ถ้าเคลื่อนข้อต่อแกน L ไป 45 องศา ที่ข้อต่อแกน U จะมีมุมเปลี่ยนแปลงไปจากตอนเริ่มต้นเป็นอย่างไร

ตารางที่ 4.8 ผลการทดลองมุมที่เกิดขึ้นในการทดลองหมุนข้อต่อ L

จำนวนครั้ง	มุมที่เกิดขึ้น
1	52
2	52
3	51
4	51
5	50

ดังนั้น เมื่อ L มีมุมเปลี่ยนไป 45 องศา U จะมีมุมเปลี่ยนไปเฉลี่ย $\frac{256}{5} = 51.2$ องศา

เมื่อ L จะมีมุมเปลี่ยนไป 1 องศา ก็ต่อเมื่อ U มีมุมที่ไป $\frac{51.2}{45} = 1.1378$ องศา

จากการทดลองการหาอัตราส่วนระหว่างข้อต่อ L กับข้อต่อ U พบว่าอัตราส่วนที่คำนวณนั้นไม่สามารถใช้งานได้ โดยเมื่อข้อต่อ L มีการเคลื่อนที่ไป 45 องศา U จะมีค่าความผิดพลาดประมาณ 3 องศา ทุกครั้งที่ทำงาน และเมื่อสังเกตพบว่าอัตราส่วนเกิดความผิดพลาดจากการทดลองเนื่องจากต้องสังเกตจากองศาที่เปลี่ยนไปซึ่งเป็นข้อต่อใหญ่ และไม่มีมาตรวัดแสดงให้เห็นชัดเจน การใช้สายตาวัดค่ามุมจึงทำให้ค่ามุมมีความผิดพลาด จึงได้มีการปรับอัตราส่วนใหม่ขึ้นโดยอ้างอิงจากอัตราส่วนแล้วปรับ ได้ค่าอัตราส่วนใหม่ที่เหมาะสมมีค่าเท่ากับ 1.238

4.6.3.2 การหาอัตราส่วนระหว่างข้อต่อ U กับข้อต่อ θ

การทดสอบว่า ถ้าเคลื่อนข้อต่อแกน U ไป 30 องศา ที่ข้อต่อแกน θ จะมีมุมเปลี่ยนแปลงไปจากตอนเริ่มต้นเป็นอย่างไร

ตารางที่ 4.9 ผลการทดลองค่าพัลส์ของ U ที่เกิดขึ้นขณะที่มุม θ มีค่า 30 องศา

จำนวนครั้ง	Pulse ที่เกิดขึ้น
1	26600
2	26548
3	26534
4	26514
5	26586

$$\text{ค่าเฉลี่ยพัลส์ต่อ 30 องศา } \bar{x} = \frac{\sum x}{n} = \frac{132782}{5} = 26556.4$$

$$\text{เมื่อ } \theta \text{ จะเคลื่อนที่ไป 1 องศา ก็ต่อเมื่อ U เคลื่อนที่ไป } \frac{26556.4}{30} = 885.21$$

$$\text{ดังนั้น เมื่อ U เปลี่ยนไป 1 พัลส์ } \theta \text{ จะมีค่าพัลส์เท่ากับ } \frac{222}{885.21} = 0.251$$

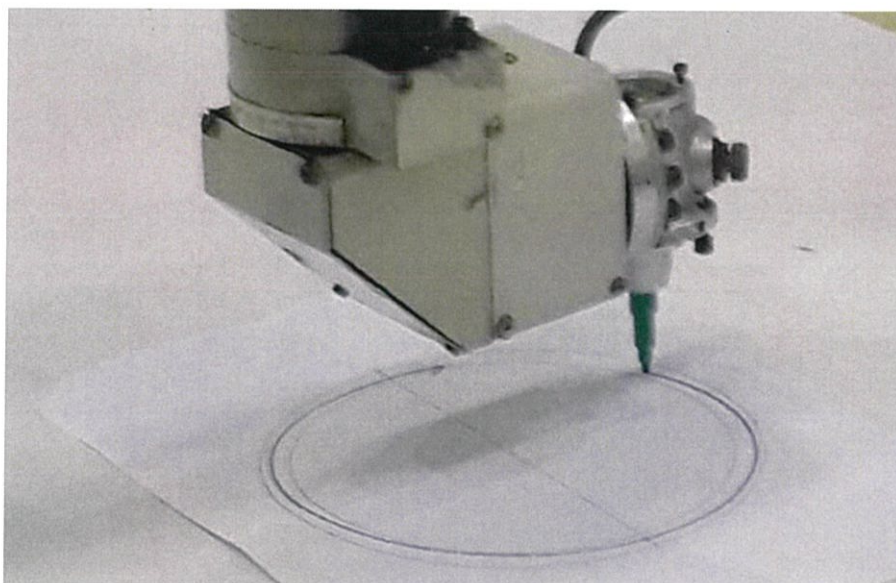
จากการทดลองการหาอัตราส่วนระหว่างข้อต่อ U กับข้อต่อ θ พบว่าเมื่อนำอัตราส่วนที่คำนวณจากการทดลองใช้ในซอฟต์แวร์ระบบการควบคุมการเคลื่อนที่ อัตราส่วนที่คำนวณนั้นสามารถใช้งานได้ ไม่มีความผิดพลาด เนื่องจากเป็นข้อต่อเล็กและมีมาตรวัดชัดเจน ดังนั้นค่าอัตราส่วนที่ใช้ในซอฟต์แวร์ระหว่างข้อต่อ U กับข้อต่อ θ มีค่าเท่ากับ 0.25

4.6.4 การทดสอบซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์

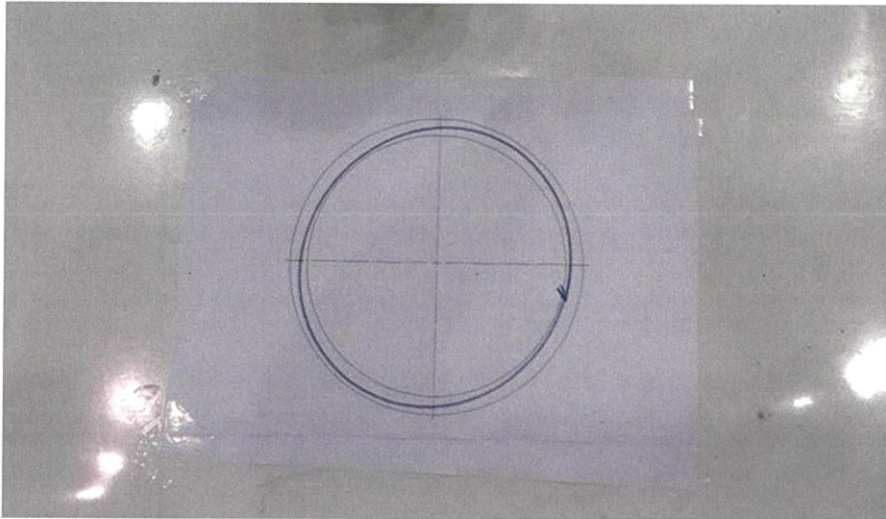
การทดสอบการใช้งานซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ เพื่อทดสอบว่าซอฟต์แวร์ระบบควบคุมที่สร้างขึ้นสามารถทำให้การเคลื่อนที่ของแขนหุ่นยนต์เคลื่อนที่ได้ตามเส้นทางที่ออกแบบไว้ใน RobotStudio ได้จริง ทดสอบด้วยการทำงานซ้ำๆ กันเพื่อดูผลลัพธ์ที่เกิดขึ้น เป็นผลที่ยอมรับได้หรือไม่ โดยแต่ละ Module มีผลดังนี้

4.6.4.1 การทดสอบการเคลื่อนที่แขนหุ่นยนต์ตาม Module 1

การทดสอบการเคลื่อนที่เป็นรูปวงกลมตาม Module 1 ออกแบบโดยการให้แขนหุ่นยนต์ติดปากกาเป็น Tool เคลื่อนที่ตามคำสั่งจากซอฟต์แวร์ซ้ำๆ เพื่อทดสอบความถูกต้องในการเคลื่อนที่ของแขนหุ่นยนต์ ในการเคลื่อนที่ในระนาบ x, y ดูค่าความผิดพลาดที่เกิดขึ้นไม่เกิน ± 5 mm.



รูปที่ 4.23 ภาพการทดสอบการเคลื่อนที่ของ Module 1 (1)



รูปที่ 4.24 ภาพการทดสอบการเคลื่อนที่ของ Module 1 (2)

ตารางที่ 4.10 ผลการทดสอบการเคลื่อนที่ของ Module 1

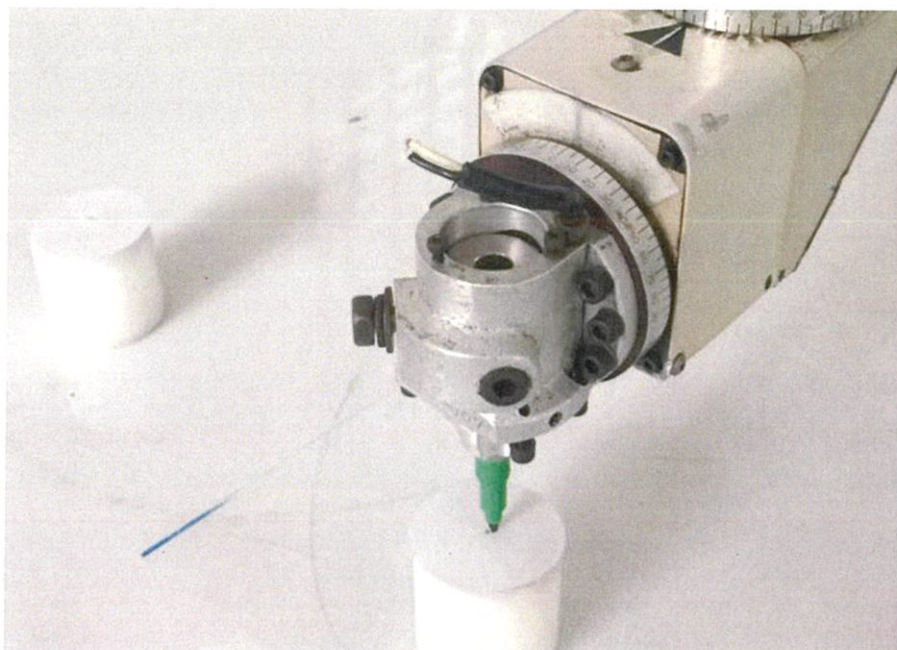
ครั้งที่	1	2	3	4	5	6	7	8	9	10
ผลการทดลอง	ผ่าน	x	ผ่าน	ผ่าน	x	ผ่าน	ผ่าน	ผ่าน	ผ่าน	ผ่าน

4.6.4.2 การทดสอบการเคลื่อนที่แขนหุ่นยนต์ตาม Module 2

ในการทดสอบการเคลื่อนที่แบบจำลอง Pick-and-Place สำหรับ Module 2 ออกแบบโดยการให้แขนหุ่นยนต์ทำงานซ้ำๆ ดูการลงของ Tool ให้ตรงกับตำแหน่งของวัตถุ



รูปที่ 4.25 ภาพการทดสอบการเคลื่อนที่ของ Module 2 (1)



รูปที่ 4.26 ภาพการทดสอบการเคลื่อนที่ของ Module 2 (2)

ตารางที่ 4.11 ผลการทดสอบการเคลื่อนที่ของ Module 2

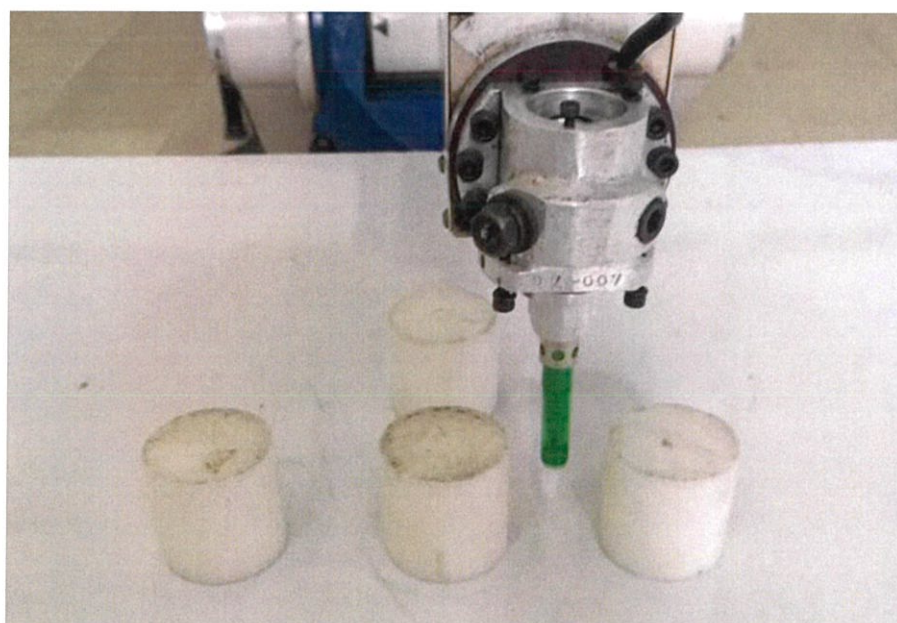
ครั้งที่	1	2	3	4	5	6	7	8	9	10
ผลการทดลอง	ผ่าน	x	ผ่าน	x	ผ่าน	ผ่าน	ผ่าน	ผ่าน	ผ่าน	ผ่าน

4.6.4.3 การทดสอบการเคลื่อนที่แบบหลบหลีกสิ่งกีดขวาง ของ Module 3

การทดสอบการเคลื่อนที่แบบหลบหลีกสิ่งกีดขวาง ของ Module 3 ออกแบบโดยการให้แขนหุ่นยนต์ทำงานซ้ำๆ โดยที่แขนหุ่นยนต์ และปลาย Tool ไม่ชน หรือกระแทกสิ่งกีดขวางที่วางไว้



รูปที่ 4.27 ภาพการทดสอบการเคลื่อนที่ของ Module 3 (1)



รูปที่ 4.28 ภาพการทดสอบการเคลื่อนที่ของ Module 3 (2)

ตารางที่ 4.12 ผลการทดสอบการเคลื่อนที่ของ Module 3

ครั้งที่	1	2	3	4	5	6	7	8	9	10
ผลการทดลอง	ผ่าน	x	ผ่าน	x	ผ่าน	ผ่าน	ผ่าน	ผ่าน	x	ผ่าน

4.6.4.4 สรุปผลการทดสอบซอฟต์แวร์ระบบการเคลื่อนที่แขนหุ่นยนต์

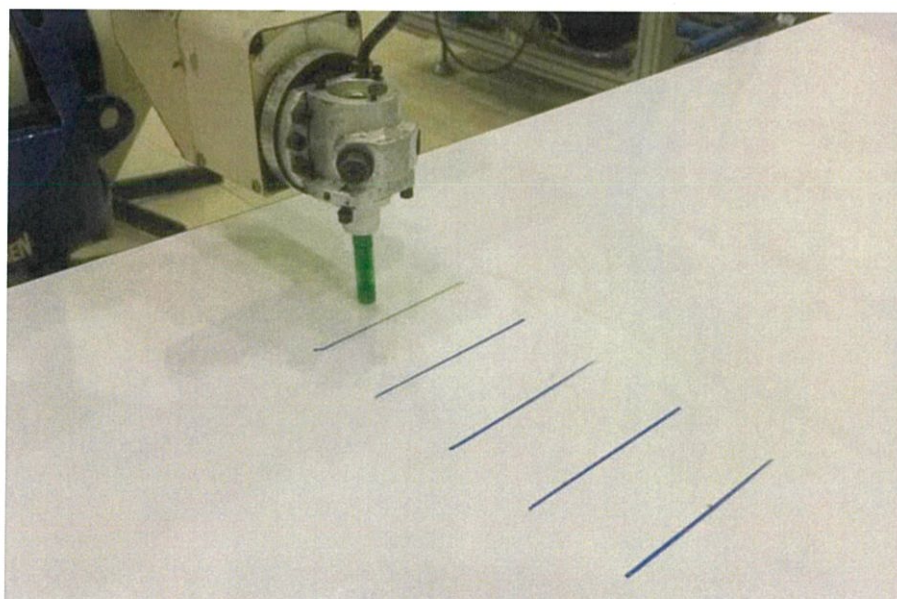
จากการทดสอบซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ทั้ง 3 รูปแบบ จะได้ว่า การเคลื่อนที่ของแขนหุ่นยนต์ตาม Module 1 เคลื่อนที่ที่ได้เป็นวงกลมตามที่ต้องการ ลักษณะที่ได้เป็นเส้นโค้งต่อเนื่องไม่มีเส้นกระตุก ส่วน Module 2 การเคลื่อนที่ที่ได้สามารถลงตรงพิกัดจุดที่ต้องการ โดยที่ปลาย Tool ที่แขนหุ่นยนต์ลงมาไม่มีพิกัดเดิมตรงตามต้องการ และ Module 3 การเคลื่อนที่ของแขนหุ่นยนต์เคลื่อนที่หลบหลีกได้โดยไม่ชนสิ่งกีดขวางที่ออกแบบไว้ โดยความผิดพลาดจากการทดสอบการเคลื่อนที่ทั้ง 3 รูปแบบ เมื่อสังเกตการทำงานของการเคลื่อนที่พบว่าเกิดจากค่าพัลส์ที่สร้างขึ้นให้เอนโคเดอร์มีความผิดพลาด เนื่องจากเป็นค่าที่มาจากทดลองด้วยสายตา และขนาดของแขนแต่ละข้อต่อที่ผิดพลาดมาจากการวัดที่ทำให้สมการมีความผิดพลาด

4.6.5 การทดสอบความแม่นยำในการพิกัดจุด

การทดสอบความแม่นยำในการพิกัดจุด เพื่อตรวจสอบการทำงานของข้อต่อแต่ละแขนในการเคลื่อนที่ไปยังพิกัดต่างๆ ได้อย่างถูกต้องหรือไม่ เนื่องมาจากการเคลื่อนที่ของแขนหุ่นยนต์เกิดจากคำสั่งการเปลี่ยนแปลงค่ามุมมองศา โดยที่ค่ามุมที่เกิดขึ้นได้มาจากค่าเอนโคเดอร์

4.6.5.1 การทดสอบความแม่นยำในการพิกัดจุดขนานแกน x

การออกแบบการทดสอบความแม่นยำในการพิกัดจุดขนานแกน x ทำโดยการออกแบบให้แขนหุ่นยนต์วาดเส้นระยะทาง 100 mm. ทั้งหมด 5 เส้น ที่ระยะ y ต่างๆ



รูปที่ 4.29 ภาพการทดสอบการเคลื่อนที่ขนานแกน x

ตารางที่ 4.13 ผลการทดสอบการเคลื่อนที่ขนานแกน x ในระยะ y ต่างๆ

ครั้งที่	1 (mm.)	2 (mm.)	3 (mm.)	4 (mm.)	5 (mm.)	ค่าเฉลี่ย (mm.)
y = 800	98.65	98.35	98.10	97.95	97.80	98.17
y = 850	99.10	99.10	99.10	981.0	98.50	98.78
y = 900	99.55	98.85	98.50	98.55	98.55	98.80
y = 950	99.00	99.40	98.85	99.00	99.10	99.07
y = 1000	99.90	98.95	98.35	99.55	99.05	99.16

$$\text{ที่ระยะ } y=800 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{98.17-100}{100} \times 100 = -1.83\%$$

$$\text{ที่ระยะ } y=850 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{98.78-100}{100} \times 100 = -1.22\%$$

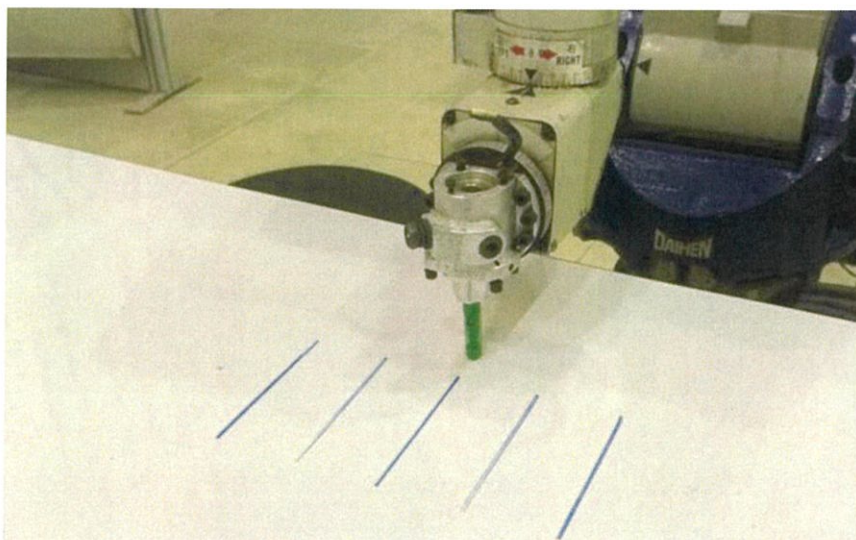
$$\text{ที่ระยะ } y=900 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{98.17-100}{1000} \times 100 = -1.2\%$$

$$\text{ที่ระยะ } y=950 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{99.07-100}{100} \times 100 = -0.93\%$$

$$\text{ที่ระยะ } y=1000 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{99.16-100}{100} \times 100 = -0.84\%$$

4.6.5.2 การทดสอบความแม่นยำในการพิกัดจุดขนานแกน y

การออกแบบการทดสอบความแม่นยำในการพิกัดจุดขนานแกน y ทำโดยการออกแบบให้แขนหุ่นยนต์วาดเส้นระยะทาง 100 mm. ทั้งหมด 5 เส้น ที่ระยะ x ต่างๆ



รูปที่ 4.30 ภาพการทดสอบการเคลื่อนที่ขนานแกน y

ตารางที่ 4.14 ผลการทดสอบการเคลื่อนที่ขนานแกน y ในระยะ x ต่างๆ

ครั้งที่	1 (mm.)	2 (mm.)	3 (mm.)	4 (mm.)	5 (mm.)	ค่าเฉลี่ย (mm.)
x = -100	102.45	102.05	102.40	102.60	102.90	102.48
x = 50	102.60	102.50	102.85	102.90	102.85	102.74
x = 0	102.20	102.45	102.60	102.60	102.95	102.56
x = 50	102.45	102.65	102.70	102.80	102.70	102.66
x = 100	102.80	102.50	102.50	102.40	102.55	102.55

$$\text{ที่ระยะ } x=-100 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{102.48-100}{100} \times 100 = 2.48\%$$

$$\text{ที่ระยะ } x=-50 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{102.74-100}{100} \times 100 = 2.74\%$$

$$\text{ที่ระยะ } x=0 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{102.56-100}{100} \times 100 = 2.56\%$$

$$\text{ที่ระยะ } x=50 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{102.66-100}{100} \times 100 = 2.66\%$$

$$\text{ที่ระยะ } x=100 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{102.55-100}{100} \times 100 = 2.55\%$$

4.6.5.3 การทดสอบความแม่นยำในการพิกัดจุดคงที่แกน z

การออกแบบการทดสอบความแม่นยำในการพิกัดจุดคงที่ที่แกน z ทำโดยการออกแบบให้แขนหุ่นยนต์จุด End - Effect อยู่ที่ 1000 mm หยดที่ระยะ y ต่างๆ

ตารางที่ 4.15 ผลการทดสอบการเคลื่อนที่คงที่แกน z ในระยะ y ต่างๆ

ครั้งที่	1 (mm.)	2 (mm.)	3 (mm.)	4 (mm.)	5 (mm.)	ค่าเฉลี่ย (mm.)
y = 800	999	1000	1000	999	999	999.4
y = 850	999	999	999	998	999	998.8
y = 900	998	997	997	997	996	997
y = 950	996	995	995	994	996	995.2
y = 1000	995	994	994	993	994	994

$$\text{ที่ระยะ } y=800 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{999.4-1000}{1000} \times 100 = -0.006\%$$

$$\text{ที่ระยะ } y=850 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{998.8-1000}{1000} \times 100 = -0.12\%$$

$$\text{ที่ระยะ } y=900 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{997-1000}{1000} \times 100 = -0.3\%$$

$$\text{ที่ระยะ } y=950 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{995.2-1000}{1000} \times 100 = -0.48\%$$

$$\text{ที่ระยะ } y=1000 \text{ มีเปอร์เซ็นต์ความคลาดเคลื่อน} = \frac{994-1000}{1000} \times 100 = -0.6\%$$

4.6.5.4 สรุปการทดสอบความแม่นยำในการพิกัดจุด

จากการทดสอบการเคลื่อนที่แกนทั้ง 3 พบว่ามีความผิดพลาดเกิดขึ้น โดยความผิดพลาดที่เกิดขึ้นนี้เป็นผลมาจากค่าจากเอนโคเดอร์ที่แปลงเป็นค่ามุมมองตามมีความคลาดเคลื่อน ทำให้การเคลื่อนที่มีความผิดพลาดเกิดขึ้น

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากการออกแบบซอฟต์แวร์ระบบควบคุมและจำลองแขนหุ่นยนต์ ได้ทำการศึกษาและสร้างสมการ Forward และ Inverse Kinematics เพื่อหาค่ามุมมองที่เกิดขึ้นกับข้อต่อแต่ละข้อ และพิกัดตำแหน่งปลายของแขนหุ่นยนต์ด้วยหลักการทฤษฎีพื้นฐานแตกแรง ตรีโกณมิติ และทฤษฎีการสร้างเมตริกซ์ของหุ่นยนต์ ผ่านการคำนวณผ่านโปรแกรม Matlab ในการสร้างซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์และการจำลองแขนหุ่นยนต์ ได้สร้างขึ้นผ่านตัวควบคุม NI CompactRIO cRIO-9082 ที่เป็นตัวควบคุมระบบปฏิบัติการ Real-Time และ FPGA ทำให้ประมวลผลได้รวดเร็วขึ้น

ในการพัฒนาซอฟต์แวร์ระบบควบคุมและจำลองแขนหุ่นยนต์ ออกแบบให้ซอฟต์แวร์รับค่าข้อมูลเป็นชุดข้อมูลที่มาจากการสร้างเส้นทางการเคลื่อนที่ผ่านโปรแกรม RobotStudio ที่สามารถสร้างเส้นทางการเคลื่อนที่แบบซับซ้อนได้ง่าย รวดเร็ว และข้อมูลที่ได้จะเข้าสู่ซอฟต์แวร์ทำการคำนวณ Inverse Kinematics เพื่อหามุมที่เกิดแต่ละข้อต่อของแขนหุ่นยนต์ รวมทั้งความเร็วที่ประสานกัน และแปลงชุดข้อมูลที่คำนวณให้เป็นค่าพัลส์เพื่อให้สอดคล้องกับการทำงานของเอนโคเดอร์ที่นับตามคำสั่งที่ส่งให้แก่ระบบควบคุมตามในแต่ละข้อต่อเพื่อสร้างการเคลื่อนที่ให้แขนหุ่นยนต์

5.2 ปัญหาและอุปสรรค

1. เนื่องจากผู้จัดทำไม่มีความรู้พื้นฐานทางด้านการออกแบบหุ่นยนต์ จึงทำให้เกิดความล่าช้าในการศึกษา และออกแบบสมการการเคลื่อนที่ของแขนหุ่นยนต์
2. เนื่องจากแขนหุ่นยนต์ที่ใช้ในปริณิญาณิพนธ์ไม่มีคู่มือ เอกสารประกอบของแขนหุ่นยนต์ ทำให้ไม่มีรายละเอียดขนาดของแขนหุ่นยนต์ รวมทั้งทางกลไก วงจรต่างๆ การเดินสายอุปกรณ์ และข้อมูลรายละเอียดต่างๆ ของแขนหุ่นยนต์
3. ตัวควบคุมแขนหุ่นยนต์ที่ใช้ในปริณิญาณิพนธ์ไม่มีข้อมูลการเดินสายภายในของตู้วงจรกับแขนหุ่นยนต์
4. เกิดอุบัติเหตุคอนเนกเตอร์ CN101 บางส่วนมีความเสียหาย ส่งผลให้ขาภายในเกิดการลัดวงจรขึ้น เนื่องจากการเคลื่อนย้ายแขนหุ่นยนต์และตัวควบคุมแขนหุ่นยนต์ที่มีน้ำหนักมาก
5. เบรกครัชต์เกิดสนิมไม่สามารถปลดเบรกได้
6. เวลาการดำเนินงานไม่เพียงพอ จึงไม่สามารถออกแบบ และสร้างการ์ดอินเตอร์เฟสสำหรับการควบคุมการเคลื่อนที่ได้ทันเวลา
7. จากการเปลี่ยนแปลงการออกแบบระบบควบคุมแขนหุ่นยนต์ใหม่ ทำให้เรียนรู้การใช้งานโปรแกรม LabVIEW ได้ไม่เต็มที่

8. โมดูล I/O ที่ใช้งานสำหรับระบบการควบคุมไม่ตรงกับความต้องการ จึงส่งผลให้ต้องมีการปรับเปลี่ยน และขอเพิ่มโมดูล I/O

5.3 ข้อเสนอแนะ

1. ปรับปรุงสมการแขนหุ่นยนต์ให้เป็นระบบมากยิ่งขึ้น
2. ซ่อมแซมคอนเนกเตอร์ CN101 ในส่วนที่ชำรุด และทำให้ขาที่มีการลัดวงจรสามารถใช้งานได้ตามเดิม
3. ซ่อมแซมเบรกครัชต์ให้สามารถใช้ได้
4. ออกแบบ และสร้างการ์ดอินเตอร์เฟสเพื่อการควบคุมให้มีประสิทธิภาพยิ่งขึ้น
5. ปรับปรุงระบบการทำงานของซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์ให้มีความเสถียร และทำงานได้อย่างมีประสิทธิภาพมากขึ้น

เอกสารอ้างอิง

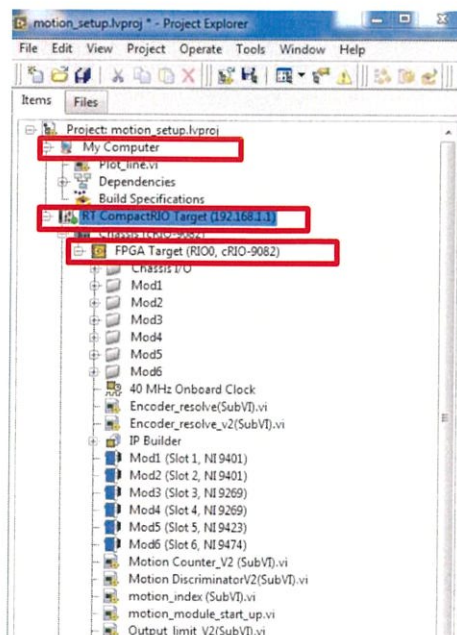
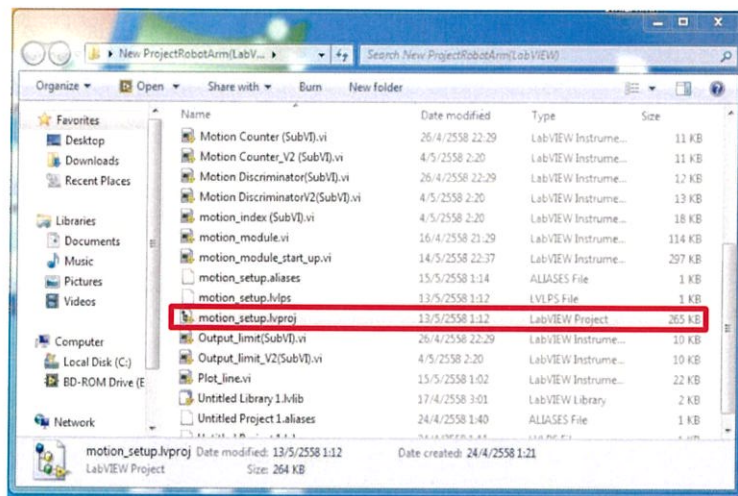
- [1] Ganesh S. Hegde. (2008). INDUSTRIAL ROBOTICS (2nd Ed.). New Delhi: Laxmi Publications (P) LTD.
- [2] Robert J. Schilling. (1990). FUNDAMENTALS OF ROBOTICS. U.S.A.: Prentice-Hall, Inc.
- [3] ApliCAD Co., Ltd. (2012). **หุ่นยนต์อุตสาหกรรม (Industrial Robot Type)**.
[Online]. Available : <http://www.aplicadthai.com/business/articles/industrial-robot-type>
- [4] Mark W. Spong., Seth Hutchinson., & M. Vidyasagar.(2006). **ROBOT MODELING AND CONTROL**. U.S.A.: John Wiley & Sons, Inc.
- [5] Wikipedia The free encyclopedia. **Industrial Robot**.
[Online]. Available : http://en.wikipedia.org/wiki/Industrial_robot
- [6] Wikipedia The free encyclopedia. **Linear Motion**.
[Online]. Available : http://en.wikipedia.org/wiki/Linear_motion
- [7] วิกิพีเดีย สารานุกรมเสรี. **ระบบควบคุมพีไอดี**.
[Online]. Available : <http://th.wikipedia.org/wiki/ระบบควบคุมพีไอดี>
- [8] Sensor_Hall_Encoder.BME. (2011). **เซนเซอร์ตรวจจับระยะการหมุน และความเร็วของมอเตอร์**.
[Online]. Available : http://hallencodersensorbme3.blogspot.com/2011/07/blog-post_25.html
- [9] National Instruments. (2014). **NI LabVIEW for CompactRIO Developer's Guide**.
[Online]. Available : <http://www.ni.com/pdf/products/us/fullcriodevguide.pdf>
- [10] กิจไพบูลย์ ชิวพันธุ์ศรี. (2014). **การสร้างระบบอัตโนมัติด้วย LabVIEW**. กรุงเทพฯ.: บริษัท ซีเอ็ดดูคั่น จำกัด (มหาชน)
- [11] rmutt. **MATLAB คืออะไร**.
[Online]. Available : http://teacher.en.rmutt.ac.th/ktw/04-720-101/intro_matlab.html
- [12] ABB. (2015). **RobotStudio It's just like having the real robot on your PC!**.
[Online]. Available : <http://new.abb.com/products/robotics/robotstudio>

ภาคผนวก

ภาคผนวก

1. วิธีการใช้งานซอฟต์แวร์ระบบควบคุมแขนหุ่นยนต์

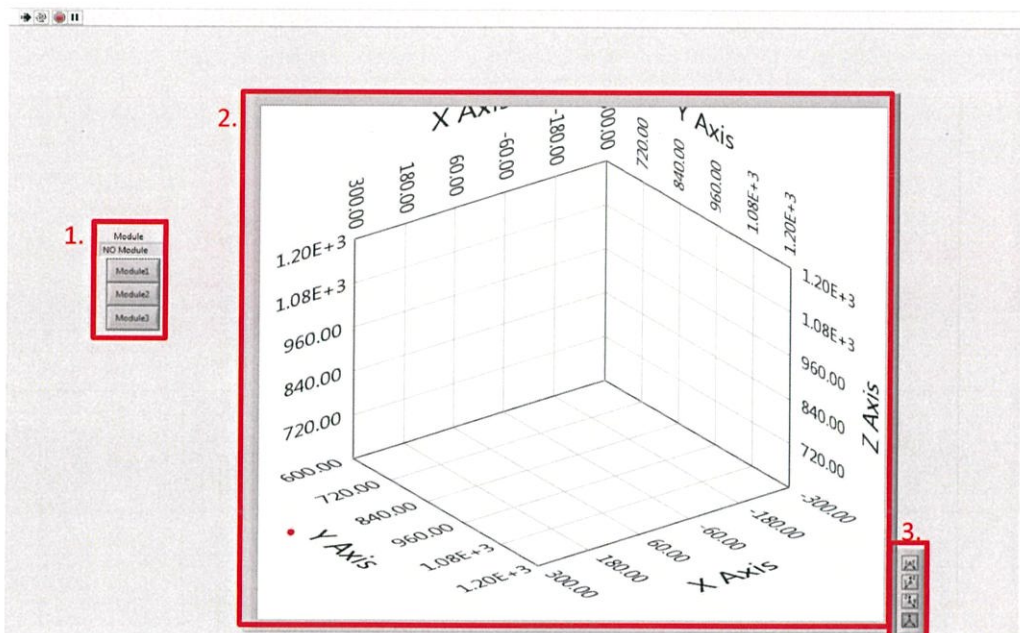
การเปิดใช้งานซอฟต์แวร์ระบบควบคุมและจำลองแขนหุ่นยนต์ เริ่มต้นจากตรวจสอบการเชื่อมต่อระหว่างคอมพิวเตอร์ กับ CompactRIO ผ่านสาย LAN โดย CompactRIO IP Address : 192.168.1.1 จากนั้น จึงเปิดไฟล์โปรเจก motion_setup.lvproj จากไฟล์เตอร์ New ProjectRoboticArm (LabVIEW) จะเข้าสู่โปรแกรม Project Explorer



จะพบส่วนการใช้งานของซอฟต์แวร์แบ่งออกเป็น 3 ส่วน ส่วนต่างๆ ทำหน้าที่ ดังนี้

- 1) ส่วนประมวลผลในคอมพิวเตอร์ ทำหน้าที่เป็นส่วนการจำลองเส้นทางการเคลื่อนที่
- 2) ประมวลผลผ่านระบบปฏิบัติการ Real-time ทำหน้าที่เป็นส่วนประมวลผลเส้นทางการเคลื่อนที่ และ ควบคุมการทำงานของซอฟต์แวร์
- 3) ประมวลผลบน FPGA ทำหน้าที่เป็นส่วนควบคุมการทำงานของ Motion Control

1.1 GUI ของส่วนการจำลองเส้นทางการเคลื่อนที่



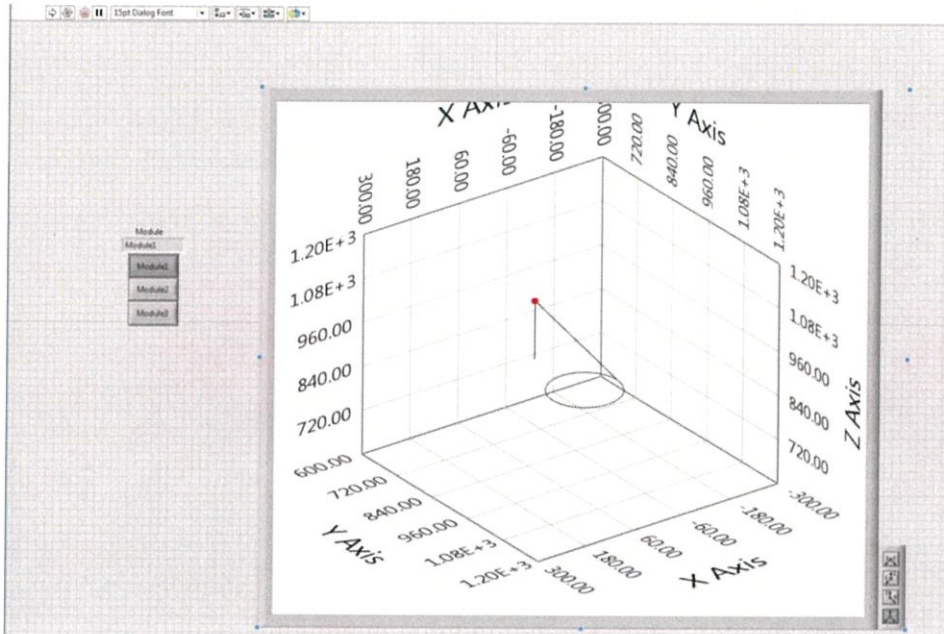
การจำลองเส้นทางการเคลื่อนที่ มีส่วนประกอบต่างๆ ต่อไปนี้

1. ส่วนเลือก Module เส้นทางการเคลื่อนที่
2. ส่วนกราฟแสดงเส้นทางการเคลื่อนที่
3. ส่วนเลือกมุมมองเส้นทางการเคลื่อนที่

การใช้งานหน้า GUI ส่วนการแสดงผลเส้นทางการเคลื่อนที่ของแขนหุ่นยนต์ มีดังนี้

1 กด Run บนแถบเครื่องมือของโปรแกรม LabVIEW เพื่อเริ่มใช้งานซอฟต์แวร์ส่วนจำลองเส้นทางการเคลื่อนที่

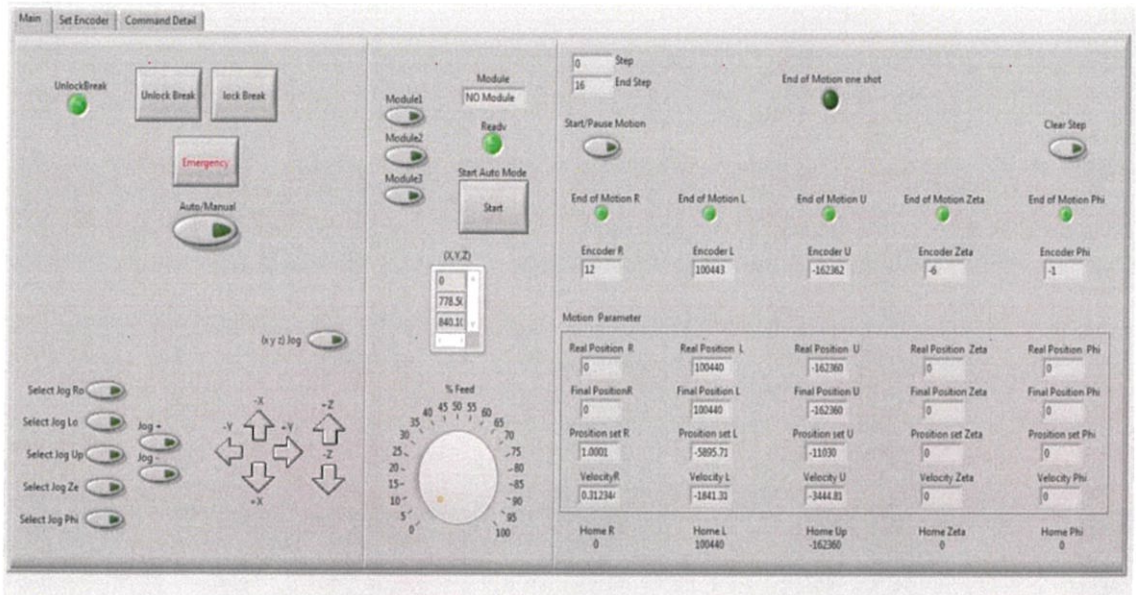
2 เลือก Module ที่ต้องการทราบเส้นทางการเคลื่อนที่ ในที่นี้แสดงตัวอย่างการเลือกแสดงการเคลื่อนที่ของ Module 1



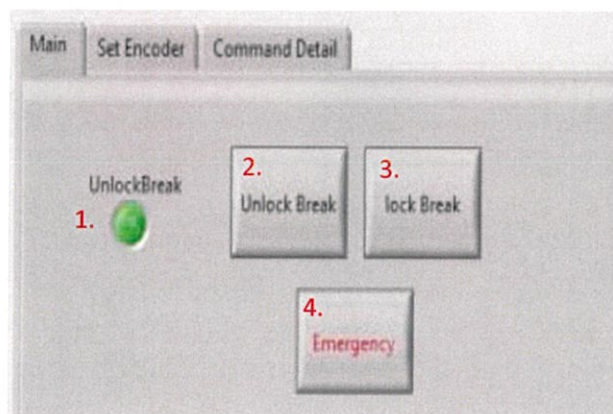
1.2 GUI ของส่วนการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์

GUI ของการใช้งานส่วนการควบคุมการเคลื่อนที่ การทำงานแบ่งออกเป็น 3 หน้า ดังต่อไปนี้

1.2.1 หน้า Main เป็นส่วนการทำงานหลักของซอฟต์แวร์



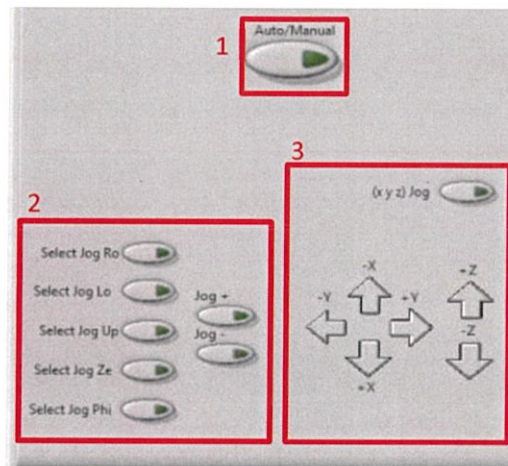
หน้า Main การควบคุมการเคลื่อนที่ มีส่วนประกอบต่างๆ ดังนี้



- ส่วนควบคุมและแสดงการทำงานของ Break

1. ไฟแสดงการทำงานของ Break  สีแดง แสดง Lock Break และ  สีเขียว แสดง Unlock Break

2. ปุ่ม Unlock Break
3. ปุ่ม Lock Break
4. ปุ่ม Emergency

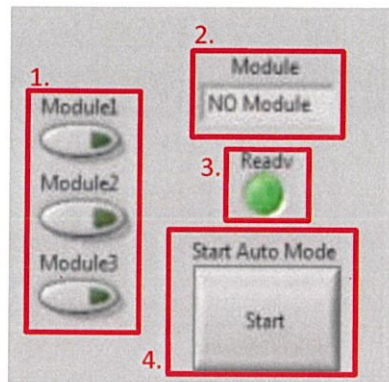


- ส่วนการเลือก Mode การทำงาน และส่วนควบคุมการทำงาน Manual Mode

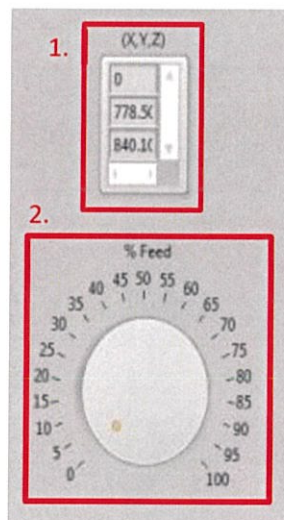
1. ปุ่มเลือก Mode กดปุ่มเป็น  แสดงการทำงาน Auto Mode ไม่กดปุ่ม  แสดงการทำงาน Manual Mode

2. ส่วนควบคุมการเคลื่อนที่ที่ละข้อต่อแขนหุ่นยนต์ ปุ่ม Select Jog ต่างๆ สำหรับเลือกข้อต่อที่จะเคลื่อนที่ และปุ่ม Jog+ กับ Jog- เป็นคำสั่งให้เคลื่อนที่เพิ่มหรือลด มุมองศาของในแต่ละข้อต่อ

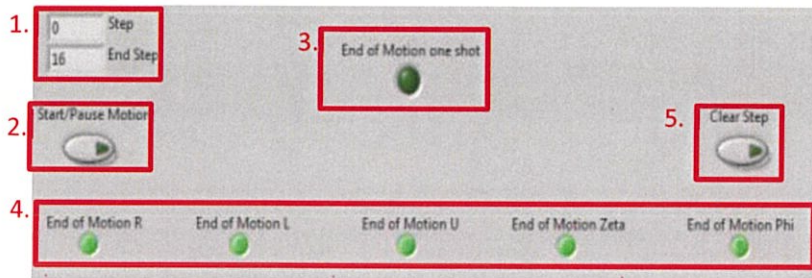
3. ส่วนควบคุมการเคลื่อนที่แบบเพิ่มหรือลดพิสัย x, y, หรือ z การเคลื่อนที่ที่จะเป็นไปตามการเลือกปุ่ม +X, -X, +Y, -Y, +Z หรือ -Z



- ส่วนเลือก Module ในการเคลื่อนที่ และ เริ่มการทำงาน Auto Mode
 1. ปุ่มเลือก Module รูปแบบเส้นทางการเคลื่อนที่
 2. แสดง Module ที่เลือกใช้งาน
 3. ไฟแสดงความพร้อมเริ่มทำงาน
 4. ปุ่มเริ่มทำงานในการทำงาน Auto Mode

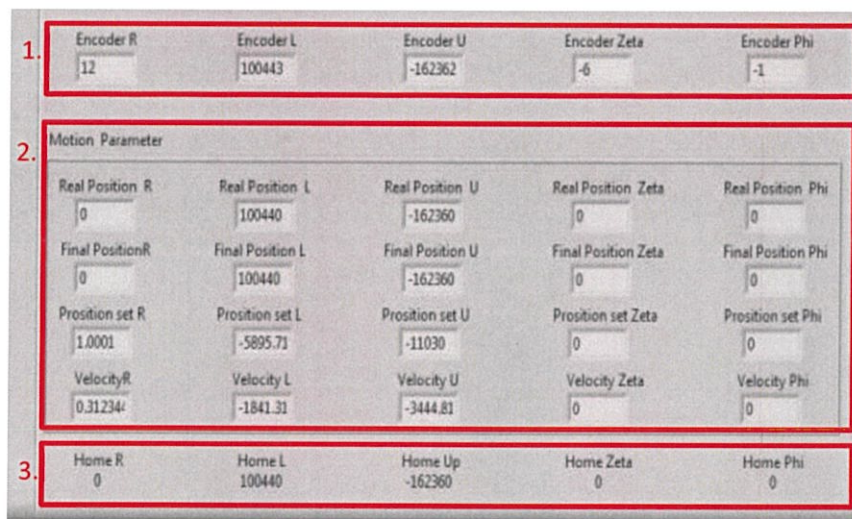


- ส่วนแสดงตำแหน่ง End - Effect และส่วนควบคุมความเร็วในการเคลื่อนที่
 1. แสดงตำแหน่ง End - Effect
 2. ปุ่มควบคุมความเร็วของการเคลื่อนที่



- ส่วนแสดงผลและควบคุมการทำงาน Mode Auto

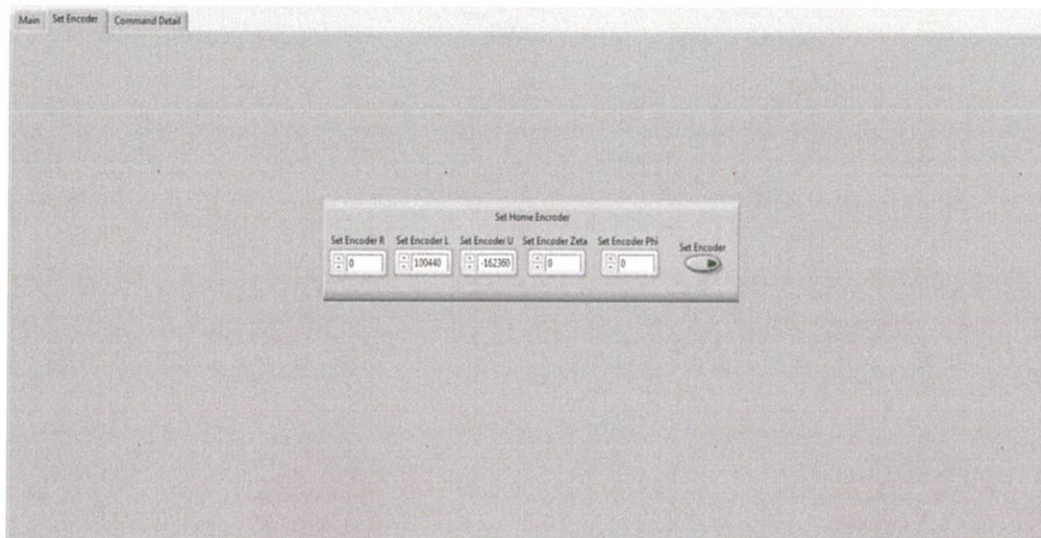
1. แสดงลำดับการเคลื่อนที่ทำงานอยู่และลำดับการเคลื่อนที่ทั้งหมด
2. ปุ่ม Start/Pause Motion
3. ไฟแสดงการเคลื่อนที่ถึงพิคัดในแต่ละลำดับ
4. ไฟแสดงการเคลื่อนที่ถึงพิคัดของแต่ละข้อต่อ
5. ปุ่ม Clear step



- ส่วนแสดงค่าต่างๆในส่วน Motion Control

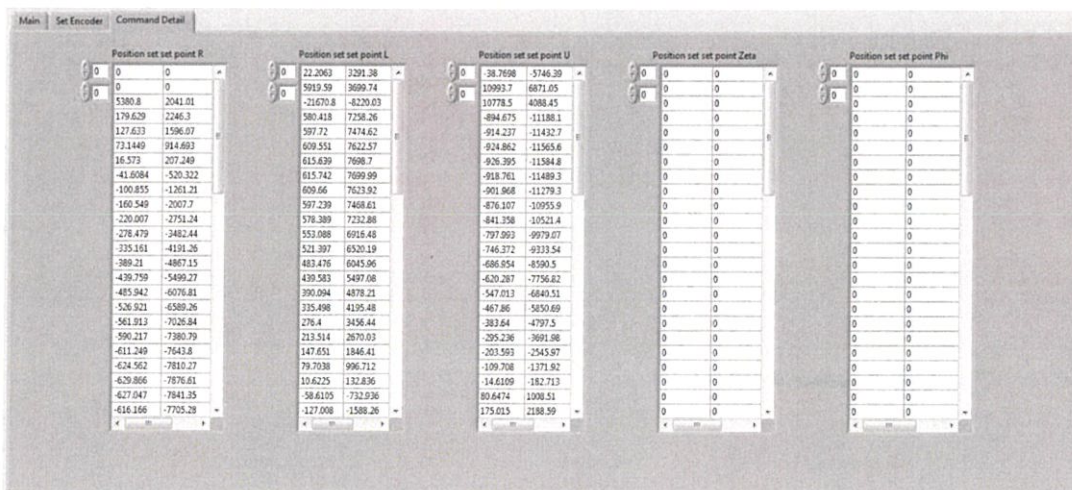
1. ค่าเอนโคเดอร์ปัจจุบัน
2. ค่าต่างๆที่ใช้ในส่วน Motion Control
3. ค่าเอนโคเดอร์ที่ตำแหน่งเริ่มต้น (สามารถเปลี่ยนตำแหน่งได้ตามที่กำหนด)

1.2.2 หน้า Set Encoder เป็นการตั้งค่าเริ่มต้นให้กับ Encoder



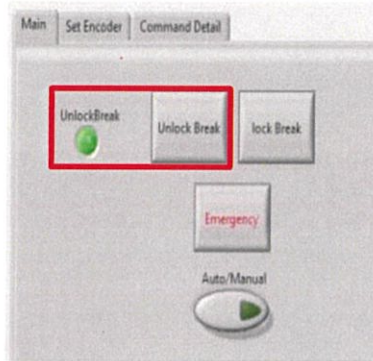
ใช้งานส่วนนี้เมื่อเอนโคเดอร์มีค่าไม่ตรงกับความเป็นจริง

1.2.3 หน้า Command Detail ชุดคำสั่งในการเคลื่อนที่ของแต่ละข้อใน Auto Mode

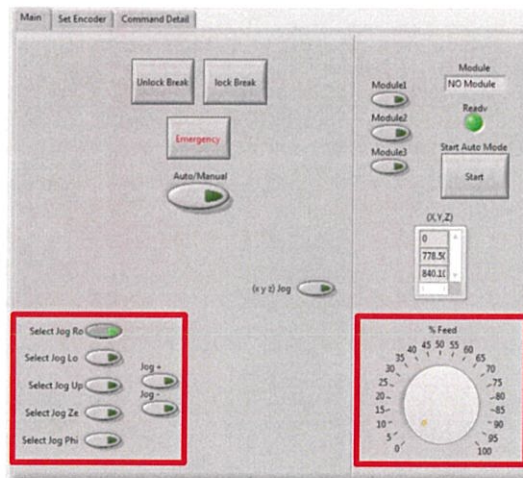


1.3 การตั้งค่าเอนโคเดอร์ก่อนการใช้งาน

1. กด Unlock Break เพื่อพร้อมใช้งานแกนหุ่นยนต์ เมื่อไฟ Unlock Break เปลี่ยนจากสีแดง  เป็นสีเขียว  แสดงว่าแกนหุ่นยนต์พร้อมเคลื่อนที่

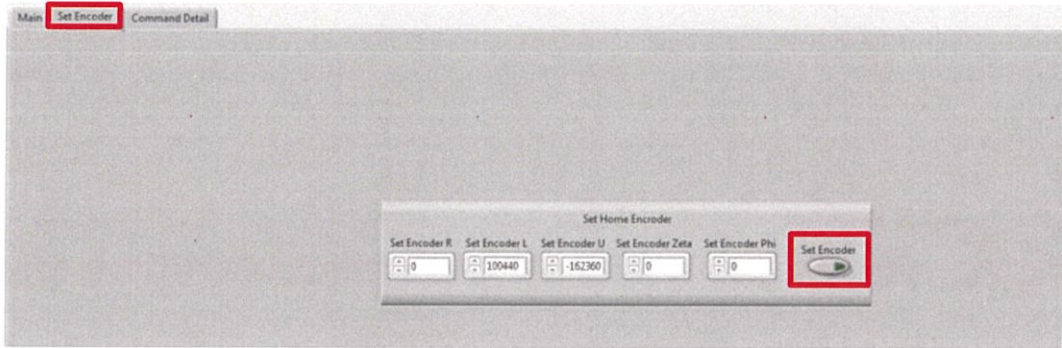


2. เคลื่อนที่แกนหุ่นยนต์ให้อยู่ในตำแหน่งเริ่มต้นที่กำหนด และตั้งค่าเอนโคเดอร์ให้มีค่าที่ถูกต้อง






ในการเคลื่อนที่แกนหุ่นยนต์ให้อยู่ในพิกัดเริ่มต้นที่กำหนดนั้น การเคลื่อนที่จะเคลื่อนที่ด้วยการใช้ Manual Mode โดยการเคลื่อนที่ที่ละเอียดใช้ปุ่ม Select Jog ต่างๆ สำหรับเลือกข้อต่อที่จะเคลื่อนที่ และปุ่ม Jog+ กับ Jog- เป็นคำสั่งให้เคลื่อนที่เพิ่มหรือลด มุมของในแต่ข้อ และสามารถเพิ่มหรือลดความเร็วในการเคลื่อนที่ด้วยปุ่ม % Feed

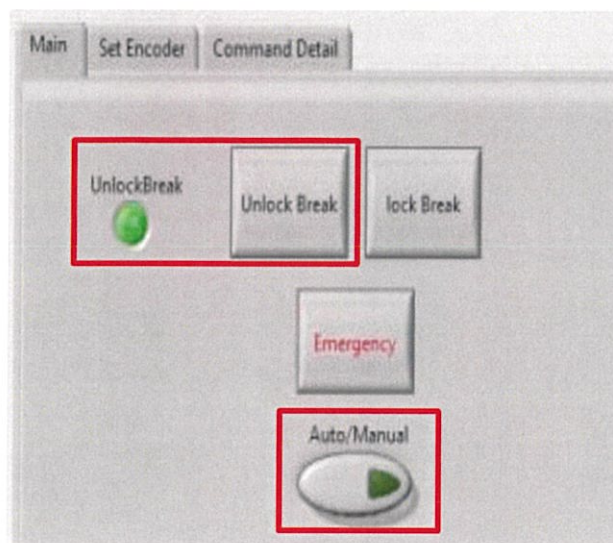
3. เมื่อแขนหุ่นยนต์อยู่ตำแหน่งที่กำหนดให้เปลี่ยนไปที่หน้า Set Encoder กดปุ่ม Set Encoder เพื่อ ตั้งค่าเอนโคเดอร์ให้มีค่าที่ถูกต้อง พร้อมใช้งาน



1.4 การใช้งาน GUI ส่วนประมวลเส้นทางเคลื่อนที่ และควบคุมการทำงานของแขนหุ่นยนต์

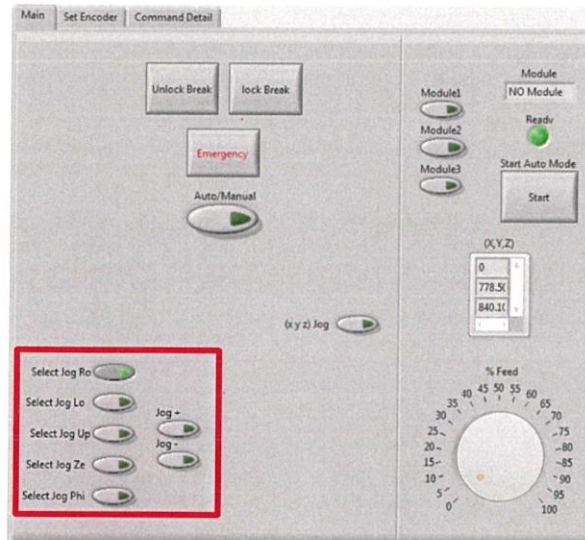
1. กด Unlock Break เพื่อพร้อมใช้งานแขนหุ่นยนต์ เมื่อไฟ Unlock Break เปลี่ยนจากสีแดง  เป็นสีเขียว  แสดงว่าแขนหุ่นยนต์พร้อมเคลื่อนที่ เลือก Mode ที่ต้องการใช้งาน  เป็น Auto Mode หรือไม่กดปุ่ม  เป็น Manual Mode

2. Manual Mode



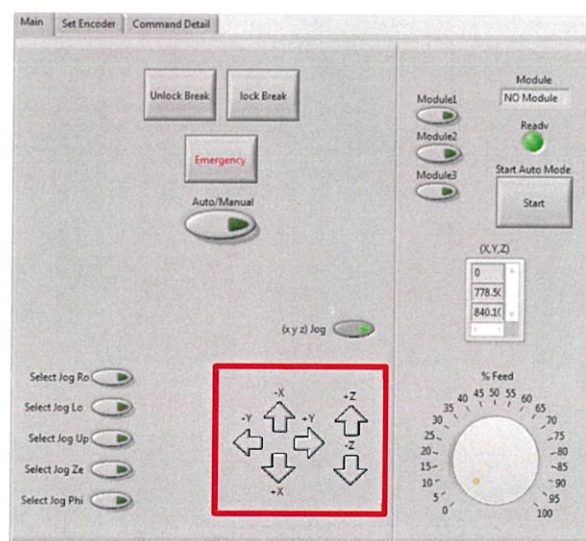
รูปแบบ Manual Mode มีการทำงาน 2 วิธีการ ดังนี้

- การเคลื่อนที่ที่ละเอียด



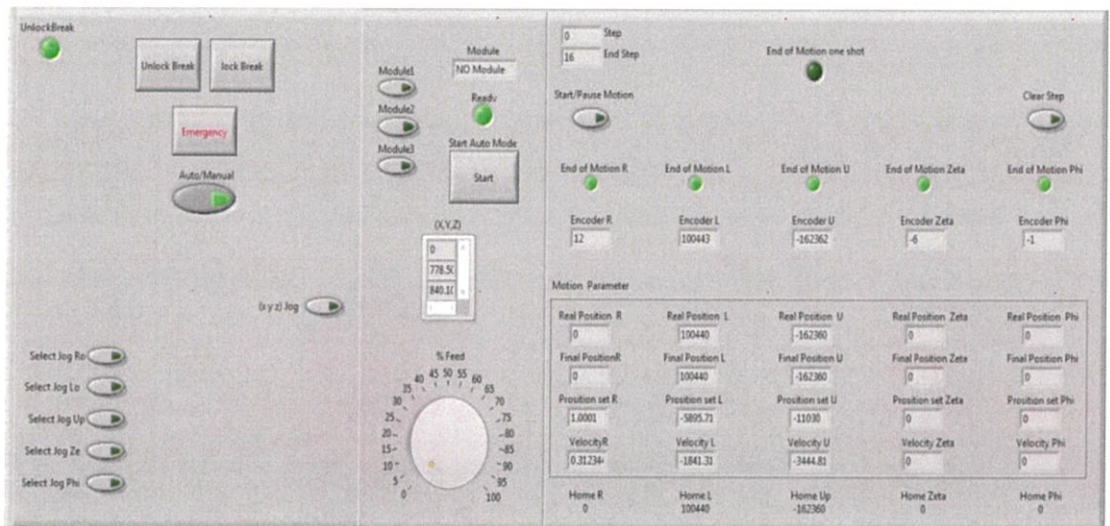
กดปุ่ม Select Jog Ro, Lo, Up, Ze หรือ Phi เพื่อเลือกการเคลื่อนที่ที่ละเอียดการเคลื่อนที่ และเลือกให้ข้อต่อเคลื่อนที่ด้วยปุ่ม Jog+ กับ Jog- เป็นคำสั่งให้เคลื่อนที่เพิ่มหรือลด มุมของแต่ละข้อต่อ และสามารถเพิ่ม หรือลดความเร็วในการเคลื่อนที่ ด้วยปุ่ม % Feed

- การเคลื่อนที่แบบเพิ่มหรือลดพิสัยแกน x, y หรือ z



กดปุ่ม (x y z) Jog เพื่อเลือกการเคลื่อนที่แบบเพิ่มหรือลดพิกัด x, y, หรือ z โดยการเคลื่อนที่จะเป็นไปตาม การกดปุ่ม +X, -X, +Y, -Y, +Z หรือ -Z

3. Auto Mode



เป็นการเคลื่อนที่ตามชุดคำสั่งที่สร้างขึ้น ขั้นตอนการใช้งาน Auto Mode

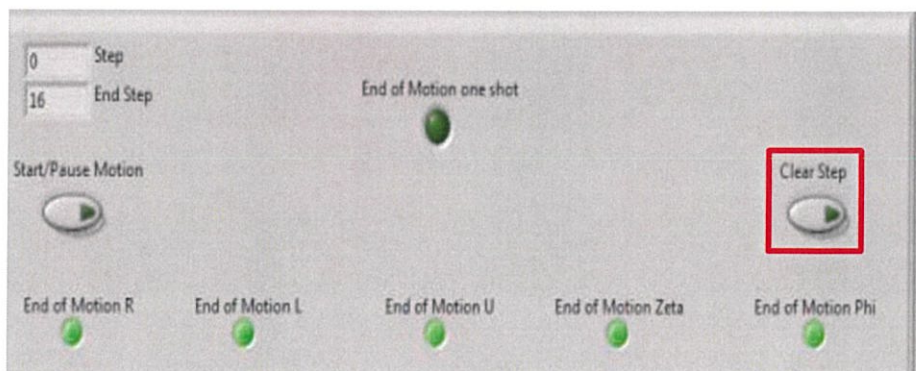
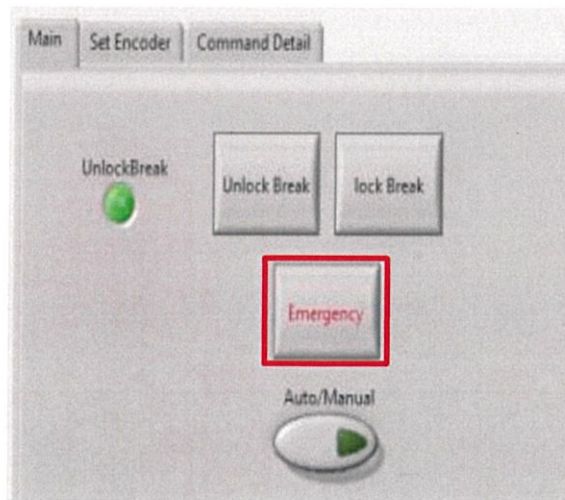
- 1) ตรวจสอบตำแหน่งเริ่มต้นอยู่ให้อยู่ตำแหน่งที่ถูกต้อง แล้วตรวจสอบไฟ Ready
- 2) เลือกรูปแบบเส้นทางการเคลื่อนที่จาก ปุ่ม Module ที่มีอยู่ 3 แบบ
- 3) กดปุ่ม Start เพื่อเริ่มการเคลื่อนที่ตามรูปแบบที่กำหนด
- 4) สามารถเพิ่ม ลด ความเร็วในการเคลื่อนที่ ด้วย ปุ่ม % Feed
- 5) สามารถเคลื่อนที่แต่ละลำดับของเส้นทางการเคลื่อนที่ด้วยการกดปุ่ม Start/Pause

Motion ค้างไว้ การเคลื่อนที่จะหยุดเมื่อจบการทำงานในลำดับนั้นๆ แล้วรอการกดปุ่ม Start/Pause Motion อีกครั้งเพื่อเริ่มทำงานลำดับต่อไป แต่ถ้าต้องการให้กลับมาเคลื่อนที่แบบต่อเนื่อง ต้องกดปุ่ม Start/Pause Motion แล้วปลดออก

1.5 การแก้ไขปัญหาที่เกิดขึ้นในขณะแขนหุ่นยนต์เคลื่อนที่

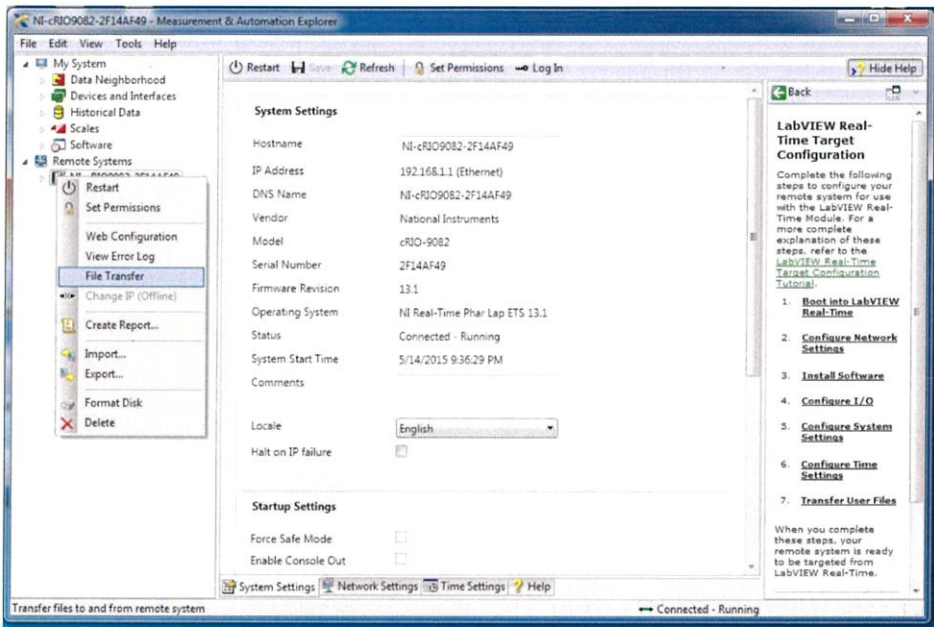
สำหรับ Manual Mode เมื่อแขนหุ่นยนต์เคลื่อนที่ไม่ตรงเส้นทางการเคลื่อนที่ หรือการเคลื่อนที่ของแขนหุ่นยนต์ อาจทำให้เกิดอันตราย สามารถกดปุ่ม Emergency จะทำให้แขนหุ่นยนต์จะหยุดเคลื่อนที่ทันที เมื่อปลดปุ่ม Emergency แขนหุ่นยนต์จะสามารถเคลื่อนที่ต่อไปได้

ส่วนการทำงานใน Auto Mode ถ้ามีการเคลื่อนที่ผิดจากเส้นทางที่สร้างขึ้น หลังจากกดปุ่ม Emergency แล้วต้องกดปุ่ม Clear Step และเปลี่ยน Mode การทำงานกลับมาเป็น Manual Mode เพื่อที่จะเคลื่อนที่แขนหุ่นยนต์ให้อยู่ตำแหน่งพร้อมเคลื่อนที่อีกครั้ง

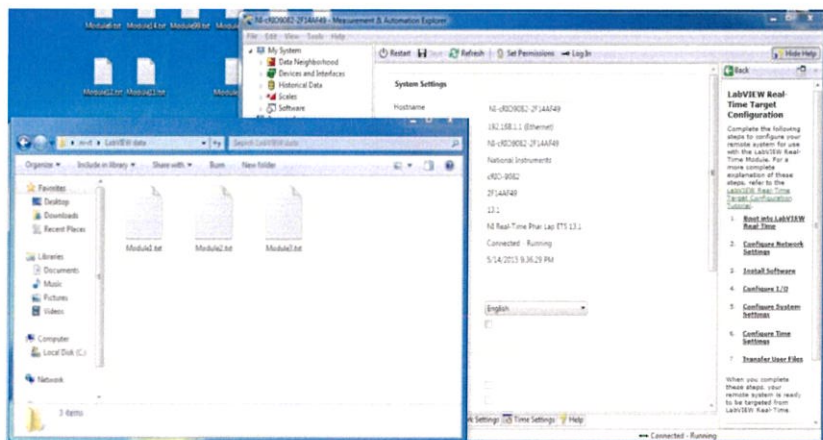


1.6 การเตรียมความพร้อมของเส้นทางการเคลื่อนที่ใหม่

1. เปิดโปรแกรม IN MAX ไปที่ Remote Systems คลิกขวาที่ตัว CompactRIO เลือก File Transfer

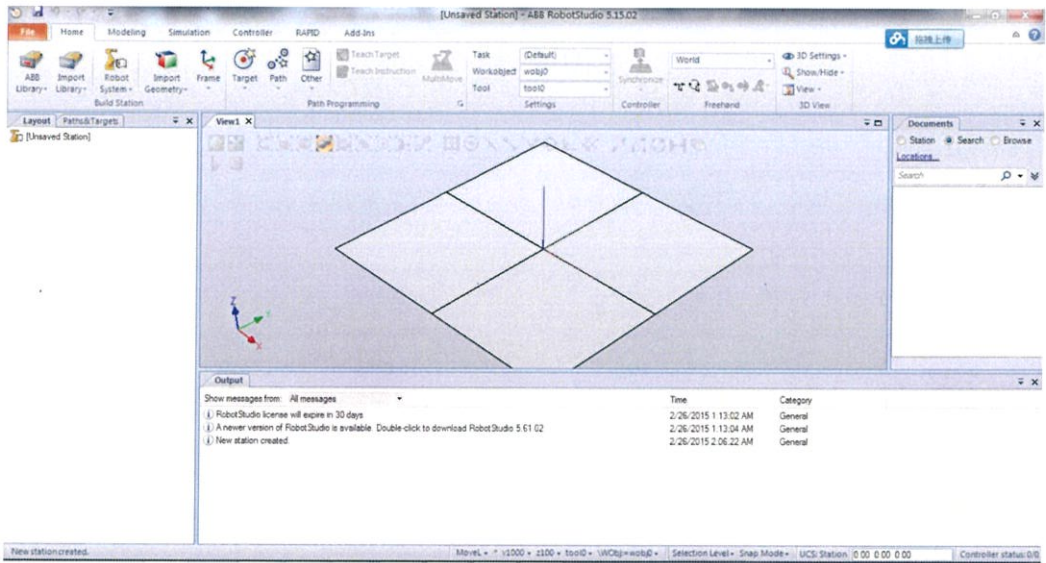


2. เข้าไปที่โฟลเดอร์ ni-rt จากนั้นเข้าโฟลเดอร์ LabVIEW Data นำไฟล์ Text เส้นทางการเคลื่อนที่ใหม่ใส่ในโฟลเดอร์ LabVIEW Data ไฟล์ Text เส้นทางการเคลื่อนที่ใหม่ต้องตั้งชื่อตาม Module ที่ต้องการ เช่น ถ้าต้องการให้เป็น Module1 คือเส้นทางการเคลื่อนที่ใหม่ชื่อไฟล์ Text จะต้องตั้งชื่อว่า Module1

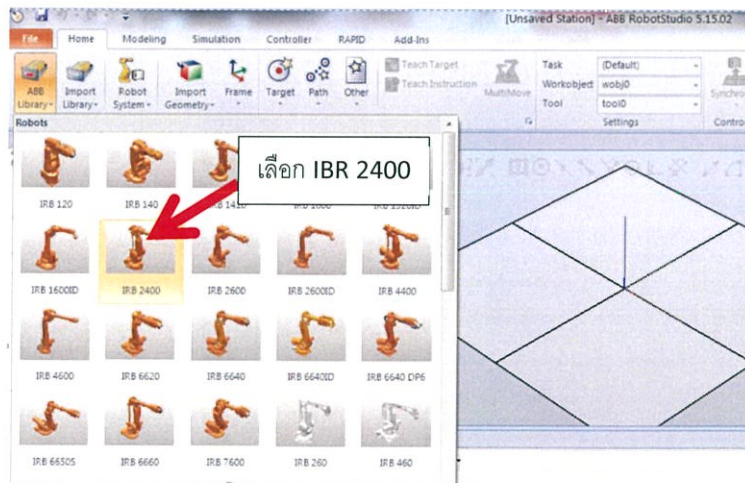


2. วิธีการสร้างเส้นทางการเคลื่อนที่ในโปรแกรม RobotStudio

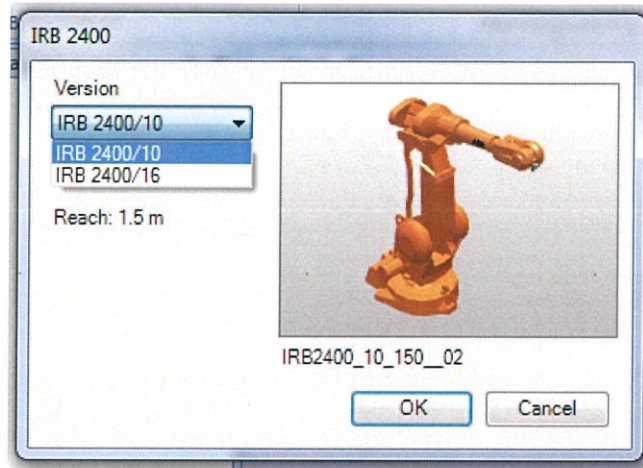
2.1 เปิดโปรแกรม RobotStudio จะพบหน้าต่าง Create a new Station เลือก Empty Station จะเข้าสู่หน้า Station ว่างเปล่า จากนั้นเข้าไปเลือก โมเดลของแขนหุ่นยนต์ ที่ ABB Library ในการใช้ เลือกใช้แขนหุ่นยนต์ในที่นี่เลือกโมเดล IRB2400



หน้า Station ของโปรแกรม RobotStudio

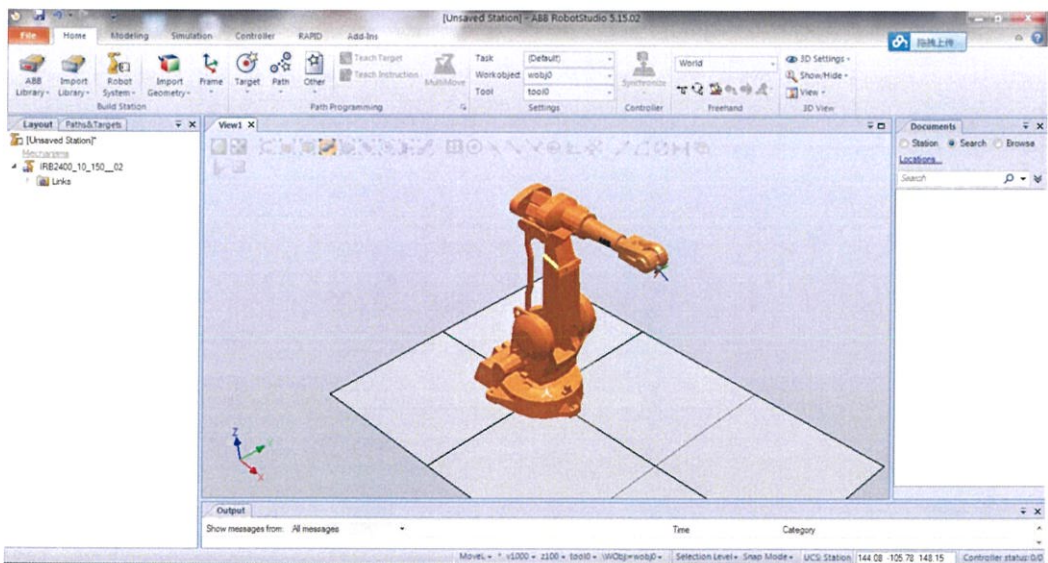


หน้าต่าง ABB Library เลือกโมเดลแขนหุ่นยนต์

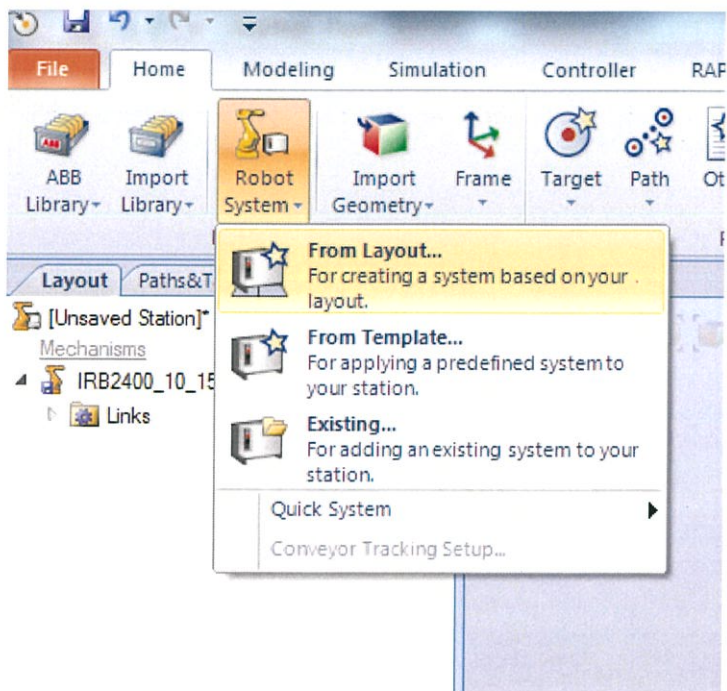


หน้าต่าง Properties ของแขนหุ่นยนต์

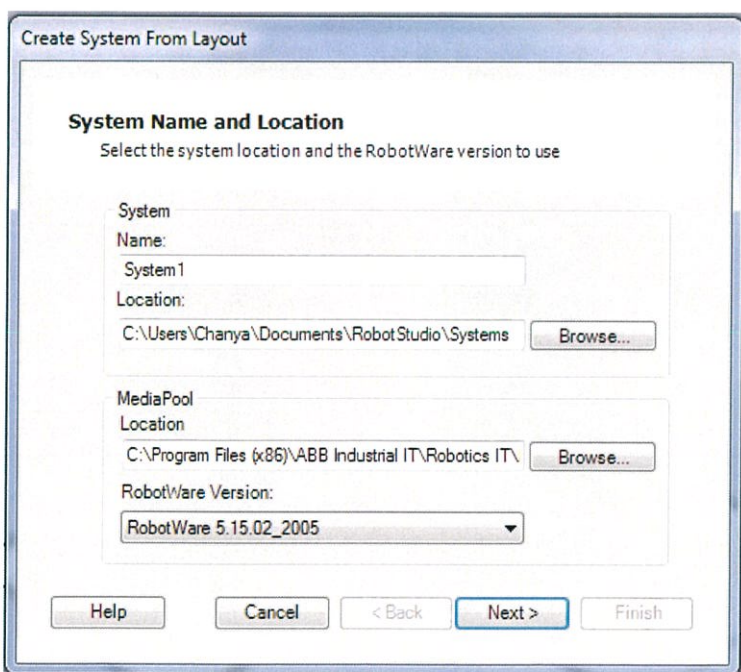
2.2. เมื่อวางแขนหุ่นยนต์เรียบร้อยแล้วจากนั้นจึงมาสร้างตัวควบคุมให้แก่แขนหุ่นยนต์ โดยเลือก Robot System จะปรากฏหน้าต่างให้เลือก From Layout... เมื่อขึ้นหน้าต่าง Create System From Layout ให้กด Next ไปเรื่อยๆ เป็นการสร้าง Location และ Configuration System Options



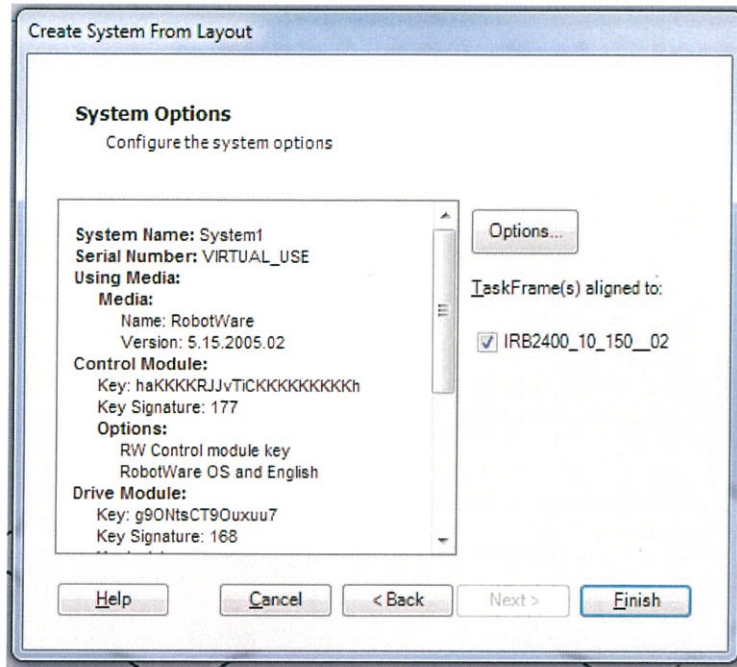
วางโมเดลแขนหุ่นยนต์ลงบน Station



หน้าต่าง Robot Layout... เลือกตัวควบคุม

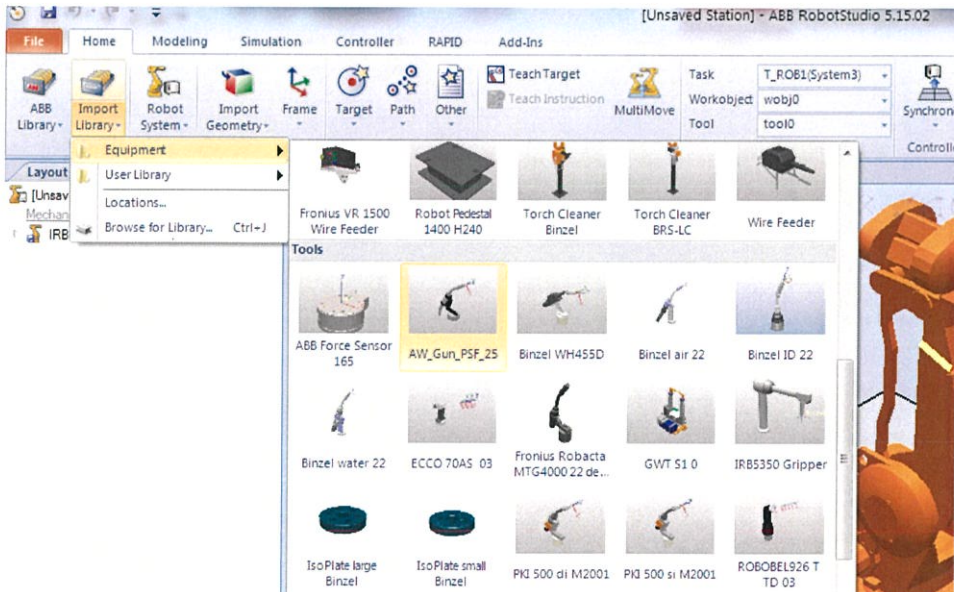


หน้าต่าง Create System From Layout (System Name and Location)



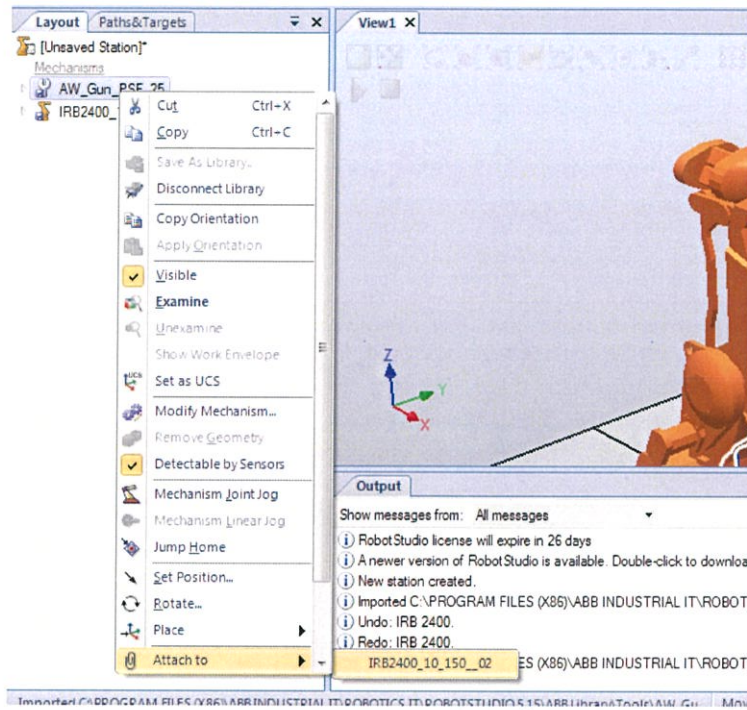
หน้าต่าง Create System From Layout (System Options)

2.3. จากนั้น คลิก Import Library เพื่อเลือก Tool ให้กับแขนหุ่นยนต์ โดยเข้า เลือกใน Equipment ในที่นี้เลือก Tool เป็นหัวเชื่อม AW_Gun_PFS_25



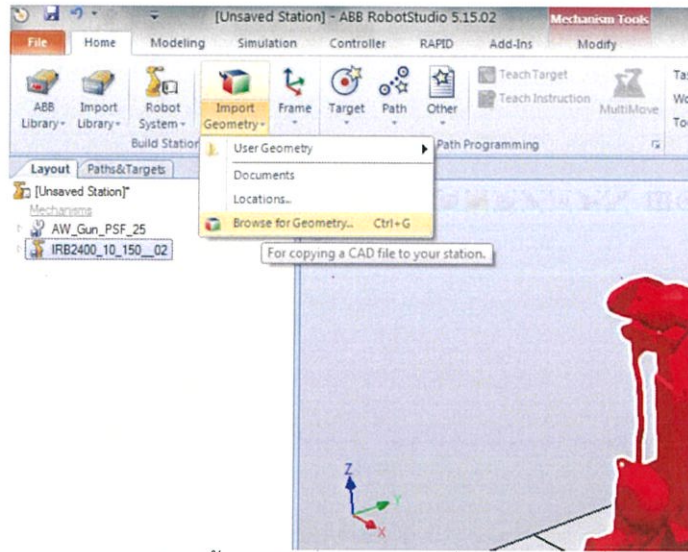
หน้าต่าง Robot Library เลือก Tool

2.4. เมื่อเลือก Equipment เสร็จ จึงเข้ามาติดตั้ง Tool เข้ากับปลายของ End - Effect ใน Station โดยคลิกขวาที่หัว Tool จากนั้น Attach to รุ่นแขนหุ่นยนต์ที่ต้องการ



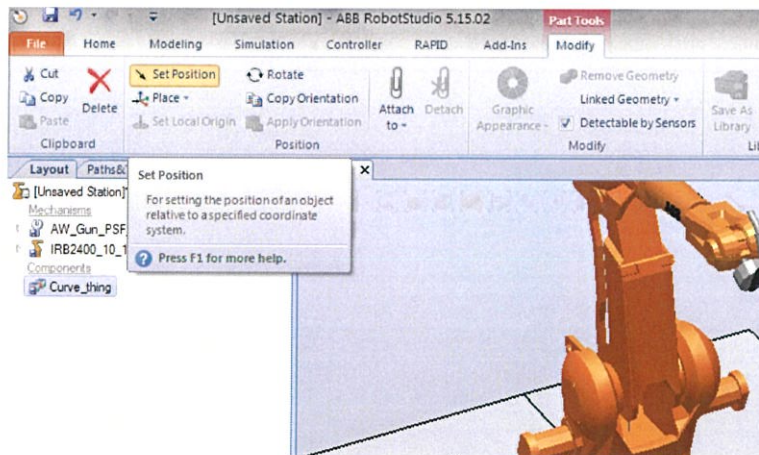
การติด Tool ลงแขนหุ่นยนต์

2.5. ในการสร้างการเคลื่อนที่ สามารถสร้างการเคลื่อนที่รูปชิ้นงานได้จากโปรแกรมอื่นๆ อาทิเช่น Solid Work เป็นต้น รูปชิ้นงานที่สร้างขึ้น จำเป็นต้องสร้างทางด้าน Front View เท่านั้น และสกุลของไฟล์ ต้องบันทึกในรูปแบบ .GIS โดยนำเข้ามาผ่านทาง Import Geometry เลือก Browse for Geometry



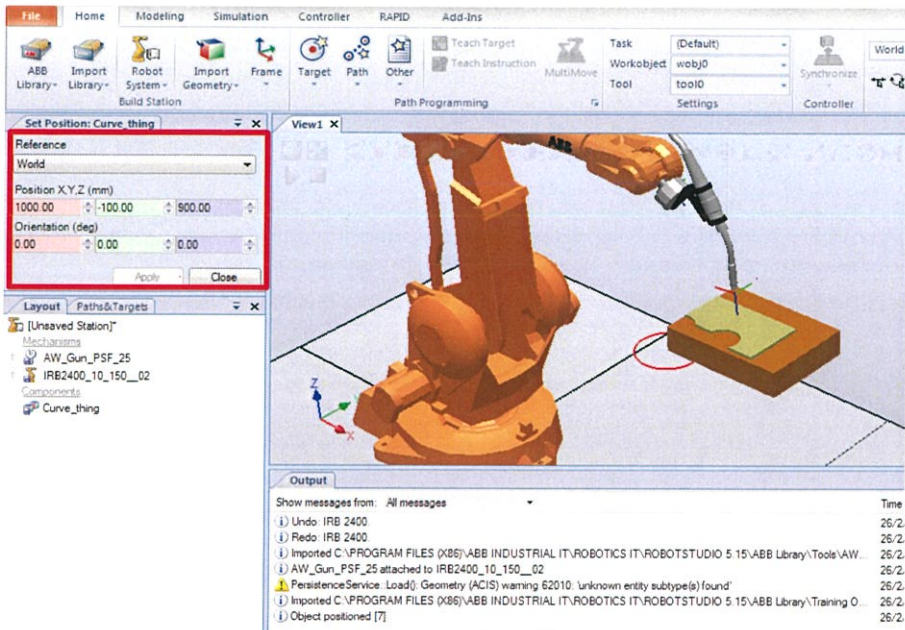
การนำรูปชิ้นงานได้จากโปรแกรมอื่นๆเข้ามา

2.6. เมื่อนำชิ้นงานเข้ามาในหน้า Station สามารถย้ายตำแหน่งของรูปชิ้นงานไปยังพิกัดที่ต้องการได้ โดยการที่คลิกชื่อของรูปชิ้นงานทางซ้ายมือ ให้รูปชิ้นงานเป็นสีแดง เลือก Modify ตรงแถบด้านบน เลือก Set Position



การเคลื่อนพิกัดรูปชิ้นงาน

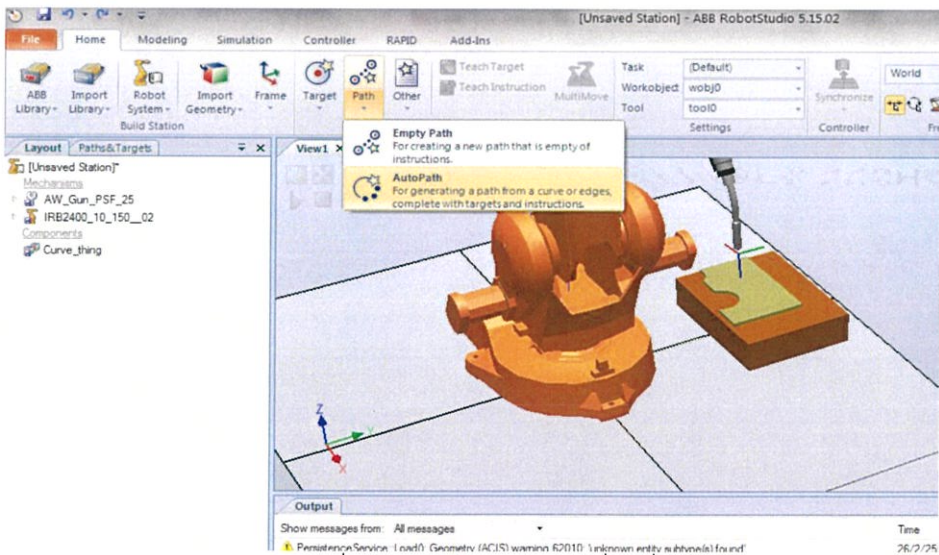
2.7. เมื่อเลือก Set Position จะพบหน้าต่างทางด้านซ้ายเพื่อระบุพิกัดที่ต้องการใหม่ของรูปชิ้นงานที่จุด (x,y,z) ตัวอย่างเช่น $x=1000\text{mm}$, $y=-100\text{ mm}$, $z=900\text{mm}$ เป็นต้น



การระบุพิกัดใหม่ให้รูปชิ้นงาน

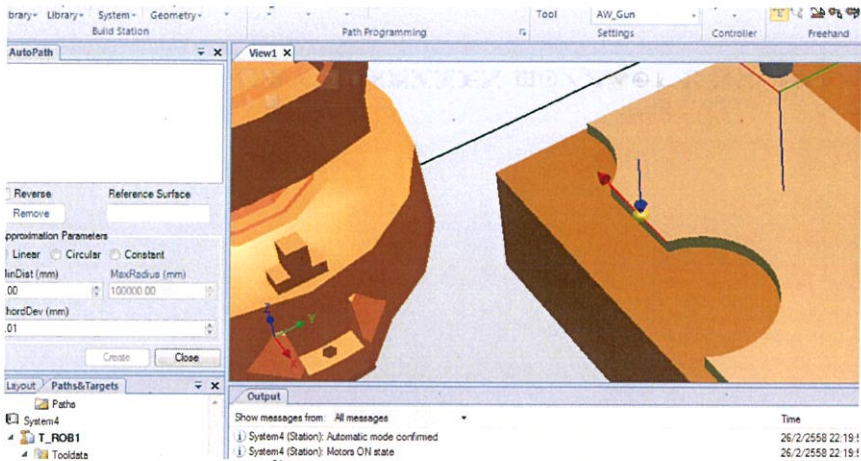
2.8. อีกวิธี คือ เลือกชื่อรูปชิ้นงาน จากนั้นเลือกคำสั่ง  (Move) ทำให้สามารถเลื่อนไปพิกัดที่ต้องการได้ทันที แต่มีข้อเสีย คือ ไม่สามารถระบุพิกัดที่แน่นอนได้

2.9. วิธีการสร้างเส้นทางการเคลื่อนที่ เลือก Path กด Auto Path



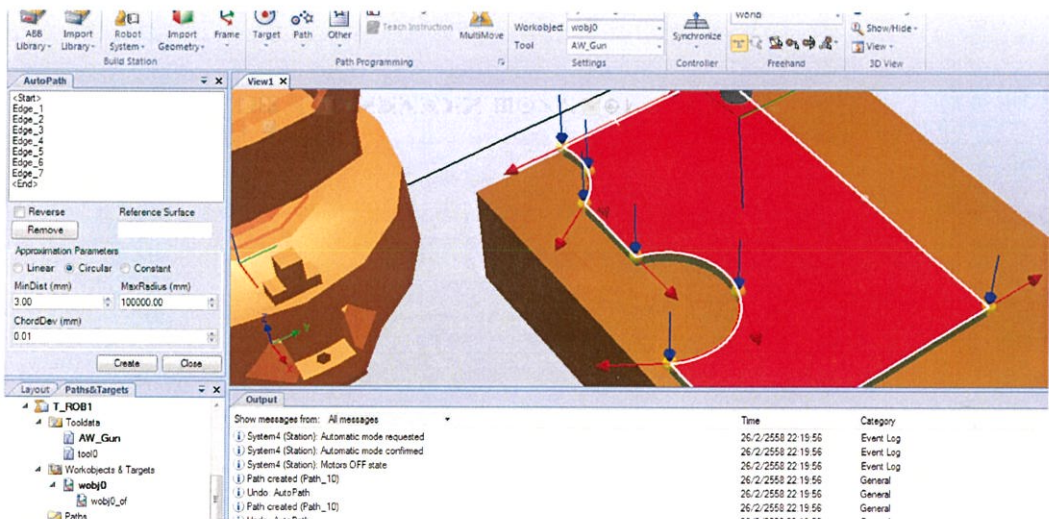
การเริ่มต้นสร้างเส้นทางการเคลื่อนที่

2.10. การเลือกใช้ Auto Path จะต้องเลือกเส้นขอบที่ต้องการสร้างเส้นทางการเคลื่อนที่ โดยจะมี ลูกศรสีน้ำเงินชี้ขึ้นมาดังรูป



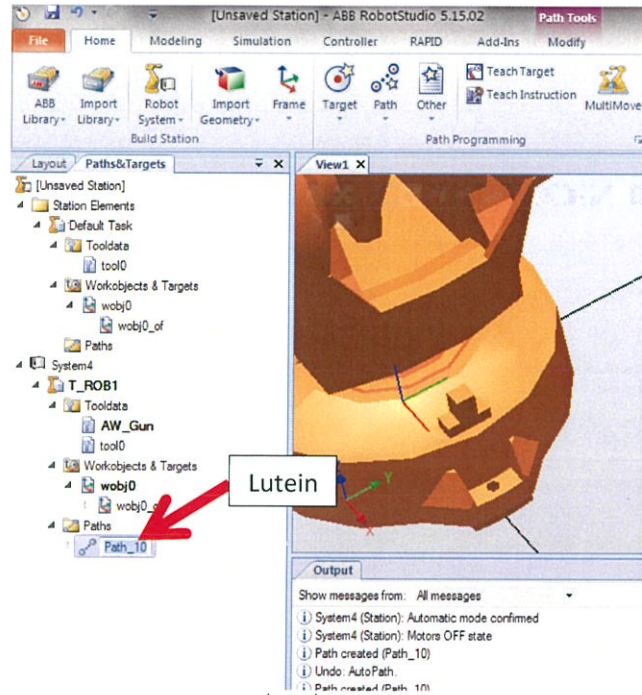
ลูกศรสีน้ำเงินแสดงเส้นขอบที่สร้าง Auto Path

2.11. คลิกซ้ายบนเส้นขอบที่เลือกไว้ จะปรากฏเส้นทางการเดินไว้ตามจุดลูกศรสีน้ำเงินที่มุมซ้าย ในชื่อ Edge_1 เมื่อสร้างเรียบร้อยจนเส้นทางครบมาที่ตำแหน่งเดิม แล้วถึงกด Create



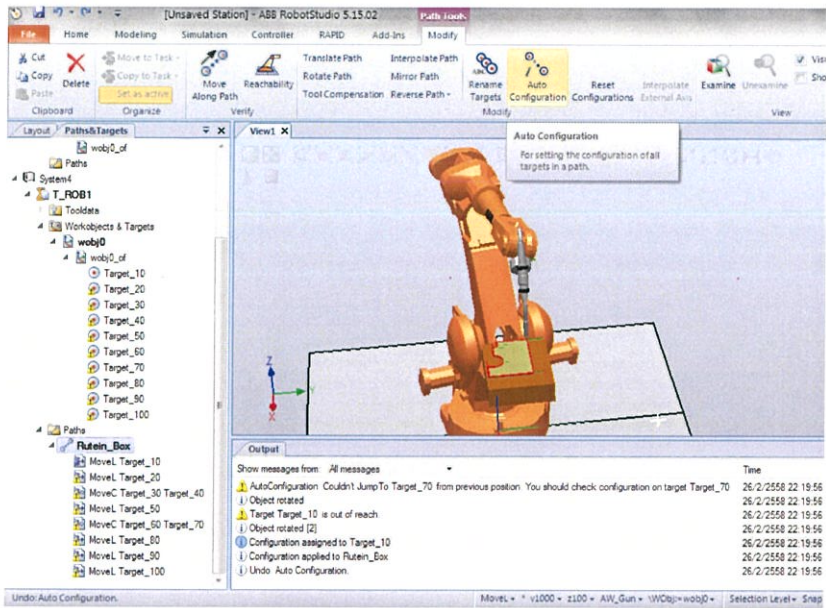
เส้นทางการเดินแบบ Auto Path

2.12. เมื่อกด Create จะปรากฏ Lutein ขึ้น ซึ่งเป็นที่เก็บการเคลื่อนที่ 1 ชุดการเคลื่อนที่ สามารถเปลี่ยนชื่อได้



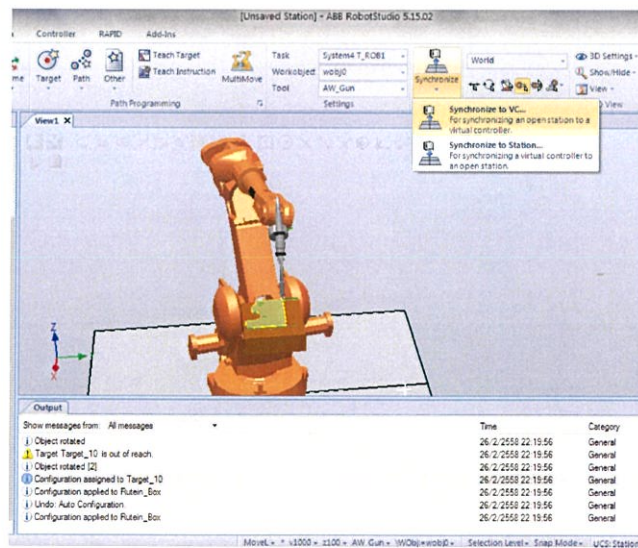
การเปลี่ยนชื่อ Lutein

2.13. คลิกที่ชื่อ Rutein_Box กด Modify คลิก Auto Configuration เพื่อทดสอบการ Configuration ที่ถูกต้อง ถ้าการเคลื่อนที่ถูกต้องโดยที่ไม่ติดปัญหาระยะแกนแขนหุ่นยนต์ไม่สามารถทำงานได้ การ Configuration จะสำเร็จถูกตามเงื่อนไข



การ Configuration

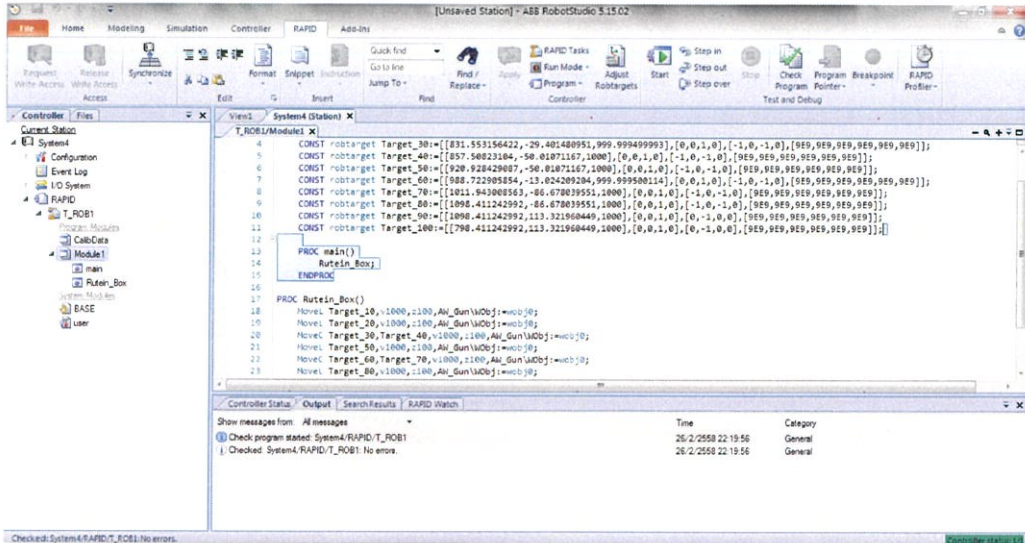
2.14. คลิกที่ Synchroniz to VC เปรียบเสมือนที่เก็บคำสั่ง โดยจะแปรรูปแบบการเคลื่อนที่ทั้งหมดที่ได้สร้างไว้ เป็นภาษา Rapid



การแปลงเป็น Rapid

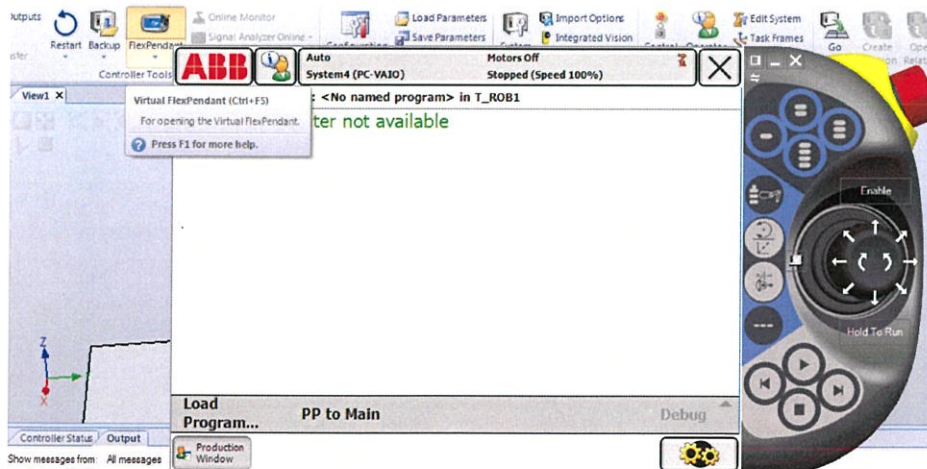
2.15. คลิกที่ Rapid และ Module1 เลือก Rutein_box เพื่อสร้าง Main ให้กับคำสั่ง Robot และ กด Apply ซึ่งใน Main เปรียบเสมือนห้องใส่คำสั่งต่างลงไปเพื่อใช้ในการประมวลผล จากในรูปที่ 18 จะ

พบว่ามีเฉพาะ Rutein_Box เดียวเท่านั้น หากมีหลาย Rutein โปรแกรมจะเข้าไปประมวลผลใน Rutein แรก ก่อนที่จะไปทำใน Rutein ถัดไป

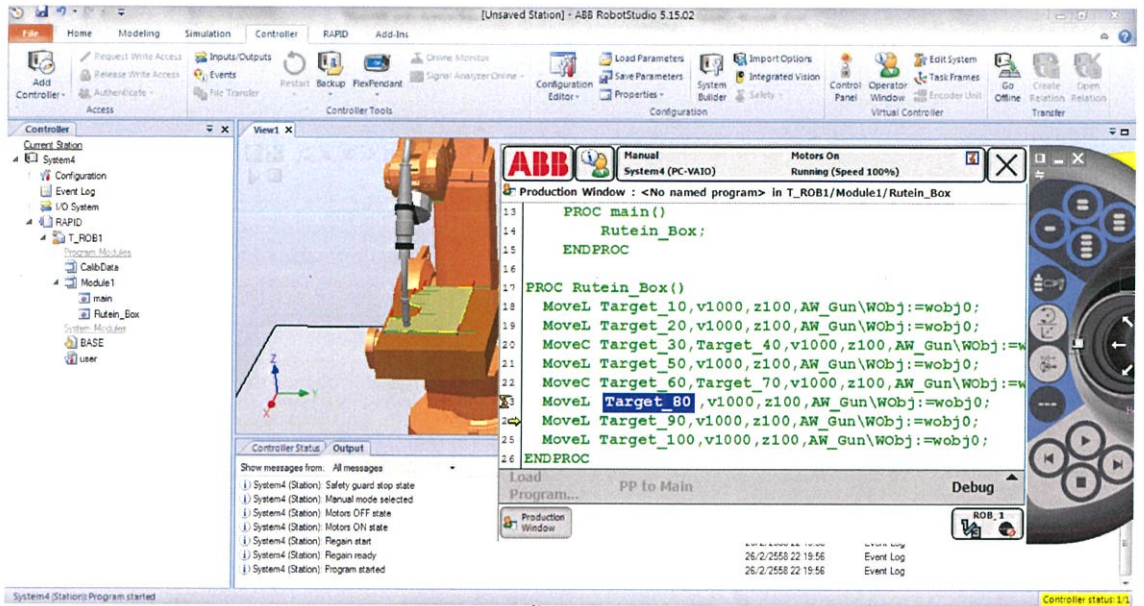


การประมวลผล Lutein

2.16. คลิก Flexpendent จะปรากฏหน้าต่างเครื่องมือขึ้น เครื่องมือชิ้นนี้สามารถทำแทนได้เหมือนหน้าโปรแกรมการทำงาน เปรียบเสมือน Hand Pad ของการควบคุม เมื่อกดปุ่ม Play แขนหุ่นยนต์จะเคลื่อนที่ตามที่เส้นทางที่ได้ Configuration ไว้

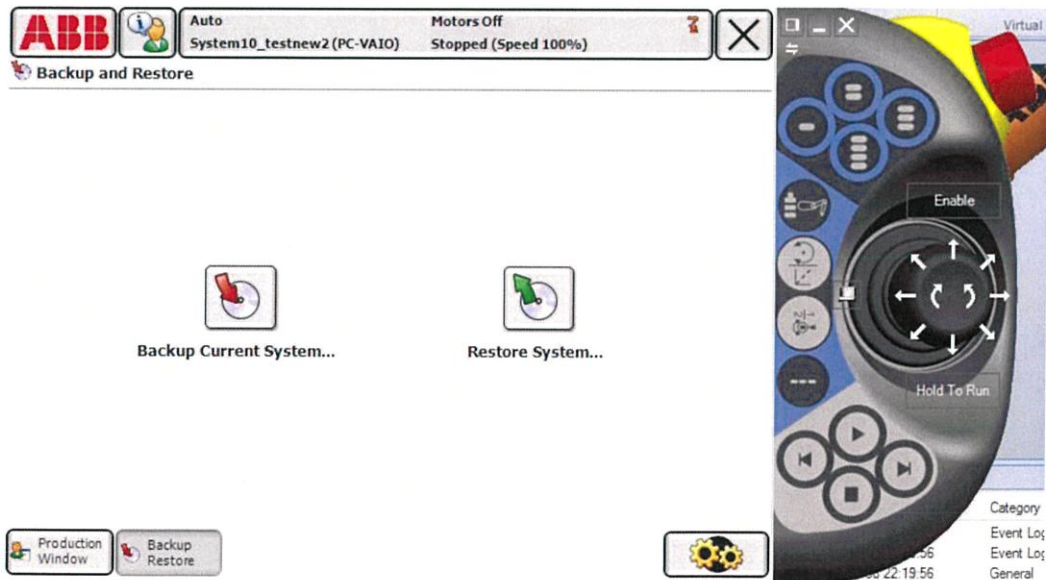


หน้าต่าง Flexpendent

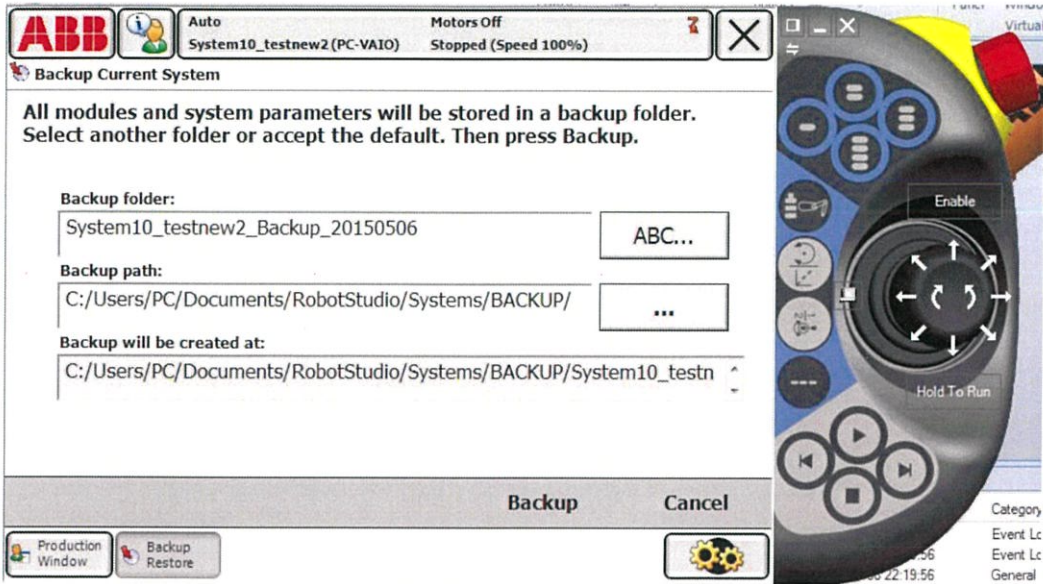


ส่วนการทำงานที่ขึ้นใน Flexpendent

2.17. เมื่อสร้างเส้นทางเคลื่อนที่เรียบร้อยแล้ว ต้องนำข้อมูลออกจากโปรแกรมเพื่อใช้ต่อไป ทำโดยการคลิก Backup Current System..

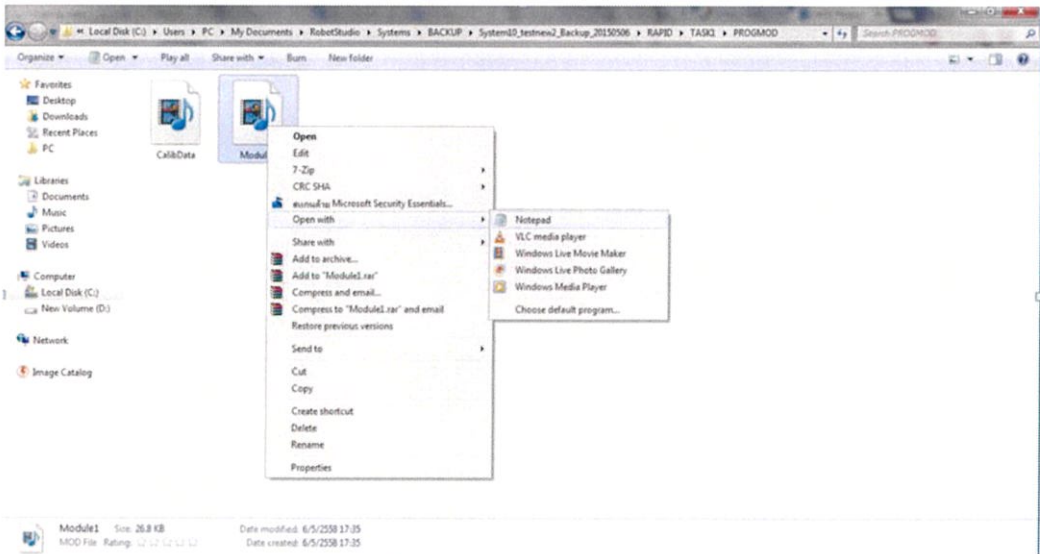


หน้าสำหรับ Backup Current System..



เลือกตำแหน่งเก็บไฟล์ Backup

2.18. การเปิดไฟล์ ให้เข้าไปตำแหน่งที่บันทึกไว้ไฟล์ จากนั้นคลิกขวา Open With เลือก Notepad จะแสดงข้อมูลพิกัดเส้นทางการเคลื่อนที่ทั้งหมดที่สร้างไว้



เลือกไฟล์ในการเปิดภาษา RAPID

