

การประยุกต์ PLC ควบคุม SCARA Robot 4DOF ตามจลนศาสตร์การ
เคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ

APPLICATION OF PLC CONTROL SCARA ROBOT 4DOF
ACCORDING TO FORWARD AND INVERSE KINEMATIC MOTION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION OF PLC CONTROL SCARA ROBOT 4DOF
ACCORDING TO FORWARD AND INVERSE KINEMATIC MOTION



Petfah Chardsomboon

Panuwat Polngam

Apiwat Kamkuen

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 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

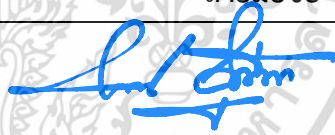
ปริญญานิพนธ์ปีการศึกษา 2565
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ การประยุกต์ PLC ควบคุม SCARA Robot 4DOF ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ
APPLICATION OF PLC CONTROL SCARA ROBOT 4DOF ACCORDING TO FORWARD AND INVERSE KINEMATIC MOTION

นักศึกษาผู้จัดทำ นางสาวเพชรฟ้า ขาติสมบุญ รหัสนักศึกษา 63015127
นายภาณุวัฒน์ พलगาม รหัสนักศึกษา 63015132
นายอภิวัฒน์ คำเชื่อน รหัสนักศึกษา 63015204

ปริญญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2565

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.ดร. สุพรรณ กุลพานิชย์	
ผศ.ดร. นรินทร์ ธรรมารักษ์วัฒนะ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การประยุกต์ PLC ควบคุม SCARA Robot 4DOF ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ	
	APPLICATION OF PLC CONTROL SCARA ROBOT 4DOF ACCORDING TO FORWARD AND INVERSE KINEMATIC MOTION	
นักศึกษาผู้จัดทำ	นางสาวเพชรฟ้า ชาติสมบูรณ์	รหัสนักศึกษา 63015127
	นายภาณุวัฒน์ พลงาม	รหัสนักศึกษา 63015132
	นายอภิวัฒน์ คำเชื่อน	รหัสนักศึกษา 63015204
อาจารย์ที่ปรึกษา	รศ.ดร.สุพรรณ กุลพาณิชย์	
	ผศ.ดร.นรินทร์ ธรรมารักษ์วัฒนะ	
ปีการศึกษา	2565	

บทคัดย่อ

โครงการนี้นำเสนอการประยุกต์ PLC ควบคุมแขนกล SCARA Robot 4DOF โดยศึกษาทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและย้อนกลับ ซึ่งประกอบด้วยองค์ประกอบหลัก 2 ส่วน ได้แก่ ส่วนแรกคือการวิเคราะห์การเคลื่อนที่ของแขนกลตามหลักการของสัญญาณนิวเมติก-อาร์เทนเบอร์กแบบดัดแปลง พีชคณิต และตรีโกณมิติ โดยการกำหนดองศาการเคลื่อนที่ของแต่ละข้อต่อ และความยาวของก้านต่อ (Link) ของแขนกลให้เป็นตัวแปรที่อยู่ในรูปของสมการคณิตศาสตร์ เพื่อสำหรับการคำนวณหาตำแหน่งส่วนปลายของแขนกล (end-effector) ในขณะเดียวกันได้มีการนำสมการคณิตศาสตร์ที่ได้มาทำการแปรผลเพิ่มเติมโดยการสร้างแบบจำลองทางคณิตศาสตร์ เพื่อจำลองการเคลื่อนที่ของแขนกลในรูปแบบของกราฟฟิคคอมพิวเตอร์ด้วยโปรแกรม MATLAB Simulink และสำหรับส่วนที่สองจะเป็นการนำ PLC มาควบคุมแขนกลจริง เพื่อทำการพิสูจน์และเปรียบเทียบความแม่นยำของการเคลื่อนที่กับแบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นว่าการควบคุมแขนกลนั้นมีความถูกต้องมากน้อยเพียงใด ซึ่งจากการทดลองในการหาประสิทธิภาพโดยการควบคุมให้แขนกลเคลื่อนที่ไปยังตำแหน่งเป้าหมายตามจลนศาสตร์การเคลื่อนที่ของแขนกลทั้งแบบไปข้างหน้าและแบบย้อนกลับ และทำการเปรียบเทียบการเคลื่อนที่กับการจำลองบนแบบจำลองทางคณิตศาสตร์พบว่าแขนกลสามารถเคลื่อนที่ไปได้อย่างถูกต้อง และมีข้อผิดพลาดเพียงเล็กน้อยซึ่งอยู่ในเกณฑ์ที่สามารถยอมรับได้

Thesis Title	APPLICATION OF PLC CONTROL SCARA ROBOT 4DOF ACCORDING TO FORWARD AND INVERSE KINEMATIC MOTION	
Authors	Ms. Petfah	Chardsomboon
	Mr. Panuwat	Polngam
	Mr. Apiwat	Kamkuen
Thesis Advisor	Assoc.Prof.Dr.Suphan	Gulpanich
	Asst.Prof.Dr.Narin	Tammarugwattana
Year	2022	

ABSTRACT

This thesis presents the application of PLC to control the SCARA Robot 4DOF by studying the robotic arm's forward and inverse kinematic motion theory. It consists of two main components, the first is an analysis of the motion of the robotic arm based on the principles of modified Denavit-Hartenberg conventions, algebra, and trigonometry. by determining the degree of movement of each joint and the length of the connecting rod (Link) of the robotic arm as a variable in the form of a mathematical equation for calculating the endpoint of the robotic arm (end-effector). At the same time, the obtained mathematical equations were further interpreted by mathematical modeling. To simulate the movement of the robotic arm in the form of computer graphics with MATLAB Simulink program. The second part is an application to use PLC to control the actual SCARA Robot 4DOF. The motion of the robotic arm will be compared with the simulation model to determine the accuracy of the robotic arm movement. From experiments to find efficiency by controlling the robotic arm to move to the target position according to the kinematics motion and comparing the motion with the simulation on a mathematical model, it was found that the robotic arm could move correctly. And there are a few errors that are within acceptable limits.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเนื่องด้วยการช่วยเหลือจากบุคคลหลายท่าน โดยเฉพาะอย่างยิ่งผู้มีพระคุณท่านแรก คณะผู้จัดทำต้องขอขอบคุณ รศ.ดร. สุพรรณ กุลพานิชย์ อาจารย์ที่ปรึกษา ที่ได้ให้ความรู้ ให้คำแนะนำในการปรับปรุงแก้ไขปัญหาต่างๆ รวมถึงสนับสนุนงบประมาณ สำหรับอุปกรณ์ในการจัดทำโครงงาน และได้กรุณาตรวจทานความเรียบร้อยของปริญญานิพนธ์ ด้วยความเอาใจใส่ทุกขั้นตอน จนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ได้อบรมสั่งสอน ให้ความรู้ในด้านต่างๆ คณะกรรมการสอบปริญญานิพนธ์ที่ได้ให้คำแนะนำแก้ไขข้อบกพร่อง ตลอดจนการจัดทำที่ผ่านมา

ขอขอบพระคุณบิดา มารดา และครอบครัวของผู้จัดทำ รวมถึงขอขอบคุณเพื่อนๆ ของผู้จัดทำทุกท่าน ที่ได้คอยให้คำปรึกษาแลกเปลี่ยนความรู้ ช่วยเหลือสนับสนุนในทุกๆด้าน จนโครงงานสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอขอบคุณผู้เขียนหนังสือเอกสารอ้างอิงและเว็บไซต์ต่างๆ ที่คณะผู้จัดทำได้นำมาใช้อ้างอิงประกอบการศึกษา จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงโดยสมบูรณ์ที่สุด

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญ (ต่อ).....	V
สารบัญ (ต่อ).....	VI
สารบัญ (ต่อ).....	VI
สารบัญตาราง.....	VII
สารบัญตาราง (ต่อ).....	VIII
สารบัญรูป.....	IX
สารบัญรูป (ต่อ).....	X
สารบัญรูป (ต่อ).....	XI
สารบัญรูป (ต่อ).....	XII
สารบัญรูป (ต่อ).....	XIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	2
1.3 ขอบเขตของปริญญานิพนธ์.....	2
1.4 ขั้นตอนการศึกษา.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	4
2.1 หลักการพื้นฐานของแขนง.....	4
2.1.1 โครงสร้างของแขนง.....	5
2.1.1.1 ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์.....	5
2.1.1.2 ส่วนประกอบทางกล.....	5
2.1.2 องศาเสรี (Degree of freedom).....	6
2.1.3 พื้นที่การทำงาน (Workspace).....	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.2 ทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล	7
2.2.1 การเคลื่อนที่ของวัตถุแข็งเกร็ง	7
2.2.2 การหมุนเฟรมใน 3 มิติ	8
2.2.3 การเคลื่อนที่จากการเลื่อน	12
2.2.4 การแปลงเอกพันธ์	13
2.2.5 สายโซ่จลนศาสตร์	15
2.2.6 จลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าโดยวิธีการของ.....	15
เดนาวิท-ฮาร์เทนเบอร์เกอร์แบบดัดแปลง	
2.2.6.1 การติดเฟรมให้กับแขนกลตามหลักการของวิธีสัญญาณ.....	16
DH Parameters	
2.2.7 การหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าของ SCARA	17
Robot 4DOFด้วยวิธีสัญญาณ DH Parameter แบบดัดแปลง	
2.2.8 การหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ	20
ของ SCARA Robot 4DOF	
2.3 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC)	22
2.3.1 โครงสร้างและส่วนประกอบของ PLC	22
2.3.1.1 หน่วยประมวลผลกลาง (CPU:Center Processing Unit)	23
2.3.1.2 หน่วยความจำ (Memory Unit)	23
2.3.1.3 หน่วยอินพุต/เอาต์พุต (Input/Output Unit)	24
2.3.1.4 แหล่งจ่ายไฟ (Power Supply Unit)	24
2.3.1.5 เครื่องป้อนโปรแกรม (Programming Device)	25
2.3.2 ภาษาที่ใช้ในการเขียนโปรแกรมให้กับ PLC	26
2.3.2.1 Ladder Diagram (LD)	26
2.3.2.2 Function Block Diagram (FBD)	26
2.3.2.3 Sequential (SFC)	26
2.3.2.4 Instruction List (IL)	26
2.3.2.5 Structure Text Language (ST)	26
2.3.3 การทำงานของ PLC	27
2.3.4 การสแกนอินพุตและเอาต์พุต	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และเผยแพร่ไปยังเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.5 ข้อมูลในหน่วยความจำของ PLC	28
2.4 ระบบเลขฐาน	28
2.4.1 ระบบเลขฐาน 2 (Binary Number System)	28
2.4.2 ระบบเลขฐาน 8 (Octal Number System).....	29
2.4.3 ระบบเลขฐาน 10 (Decimal Number System)	29
2.4.4 ระบบเลขฐาน 16 (Hexadecimal Number System).....	29
2.5 OLE for Process Control (OPC Server).....	30
2.5.1 KEPServerEX6.....	30
2.5.2 Dynamic Data Exchange(DDE).....	31
2.5.2.1 แบบ Hot Link.....	31
2.5.2.2 แบบ Cold Link.....	31
2.5.3 การติดต่อสื่อสารผ่านพอร์ตอนุกรม RS232	32
2.6 มอเตอร์แบบสเต็ป	33
2.6.1 โครงสร้างและหลักการทำงานของมอเตอร์แบบสเต็ป	33
2.6.2 ชนิดของมอเตอร์แบบสเต็ป	34
2.6.2.1 มอเตอร์แบบสเต็ปชนิดไบโพลาร์.....	34
2.6.3 การจ่ายกระแสไฟกระตุ้นให้กับมอเตอร์แบบสเต็ป	34
2.6.3.1 แบบฟูลสเต็ป 1 เฟส (Full Step 1 Phase).....	34
2.6.3.2 แบบฟูลสเต็ป 2 เฟส (Full Step 2 Phase)	35
2.6.3.3 แบบครึ่งสเต็ป (Half Step)	35
2.6.4 วงจรขับมอเตอร์	36
2.7 โฟโตอิเล็กทริกเซนเซอร์	37
2.7.1 หลักการโฟโตอิเล็กทริกเซนเซอร์ที่ใช้ในปริญญานิพนธ์	37
บทที่ 3 วิธีดำเนินการ.....	38
3.1 ภาพรวมการออกแบบการทำงาน.....	38
3.2 การออกแบบโครงสร้างการเชื่อมต่อแขนกล SCARA Robot 4DOF	39
เข้ากับ PLC	
3.3 การออกแบบโปรแกรมแบบจำลองทางคณิตศาสตร์.....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3.1 การออกแบบโมเดลจำลองแขนกล SCARA Robot 4DOF	41
3.3.2 การนำจลนศาสตร์การเคลื่อนที่ของแขนกลมาสร้างเป็น..... แบบจำลองทางคณิตศาสตร์	42
3.4 การออกแบบโปรแกรม PLC ควบคุมการทำงานของแขนกล	45
3.4.1 การเขียนโปรแกรม PLC Ladder ควบคุมการทำงานของมอเตอร์..... แบบสแต็ป	45
3.4.2 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่ตามจลนศาสตร์	46
การเคลื่อนที่ของแขนกลแบบไปข้างหน้า	
3.4.3 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่ตามจลนศาสตร์	49
การเคลื่อนที่ของแขนกลแบบย้อนกลับ	
3.4.4 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่แบบอัตโนมัติ.....	51
3.4.5 การเขียนโปรแกรมในการคำนวณค่าความเร็วที่สั่งให้มอเตอร์	52
3.4.6 ภาพรวมการออกแบบการทำงาน	52
3.5 การเชื่อมต่อ KEPServerEX6 เข้ากับ PLC.....	54
3.5.1 การเชื่อมต่อ Microsoft Excel กับ KEPServerEX6	54
3.5.2 การเชื่อมต่อ KEPServerEX6 กับ CX-Programmer	55
บทที่ 4 ผลการทดลอง	57
4.1 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่	57
ของแขนกลแบบไปข้างหน้า เปรียบเทียบกับการเคลื่อนที่ของแขนกลใน แบบจำลองทางคณิตศาสตร์	
4.2 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่	60
ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลใน แบบจำลองทางคณิตศาสตร์	
4.3 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่แบบจุดต่อจุด	62
(Point to Point)	

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	65
5.1 สรุปผลการทดลอง.....	65
5.2 ปัญหาและอุปสรรคที่พบระหว่างดำเนินการ	66
5.3 ข้อเสนอแนะและแนวทางในการพัฒนา	66
บรรณานุกรม.....	67
ภาคผนวก	68
ภาคผนวก ก แสดงคู่มือการใช้งานแบบจำลองทางคณิตศาสตร์และ HMI.....	69
ก.1 คู่มือการใช้งานแบบจำลองทางคณิตศาสตร์	70
ก.2 คู่มือการใช้งานส่วนเชื่อมต่อกับผู้ใช้งาน (Human machine interface หรือ HMI) ..	71
ก.2.1 การใช้งานการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกล.....	73
แบบไปข้างหน้า	
ก.2.2 การใช้งานการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกล.....	73
แบบย้อนกลับ	
ก.2.3 การใช้งานการควบคุมแบบอัตโนมัติ	74
ภาคผนวก ข แสดงคุณสมบัติทางเทคนิคของแขนกลที่จัดหา 76	
และอุปกรณ์ที่ใช้ในโครงการ	
ภาคผนวก ค โปรแกรมแบบจำลองทางคณิตศาสตร์.....	88
ภาคผนวก ง โปรแกรม PLC ควบคุมแขนกล SCARA Robot 4DOF	102

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดงลักษณะต่างๆที่ใช้ในแขนกล	6
2.2 ตารางแสดง DH Parameters ของ SCARA Robot 4DOF	17
2.3 ตารางแสดงตัวอย่างความสัมพันธ์ระหว่างเลขฐานแต่ละระบบ	30
2.4 ตารางแสดงลำดับการจ่ายกระแสกระตุ้นแบบฟลูสเต็ม 1 เฟส	35
2.5 ตารางแสดงลำดับการจ่ายกระแสกระตุ้นแบบฟลูสเต็ม 2 เฟส	35
2.6 ตารางแสดงลำดับการจ่ายกระแสกระตุ้นแบบครึ่งสเต็ม	36
4.1 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ไปข้างหน้า เมื่อกำหนดอินพุตที่ 20% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์	58
4.2 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ไปข้างหน้า เมื่อกำหนดอินพุตที่ 80% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์	59
4.3 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ไปข้างหน้า เมื่อกำหนดอินพุตที่ -30% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์	59
4.4 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ไปข้างหน้า เมื่อกำหนดอินพุตที่ -70% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์	58
4.5 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในทิศทาง $+x +y +z$ เปรียบเทียบกับแบบ จำลองทางคณิตศาสตร์	60
4.6 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในทิศทาง $+x -y +z$ เปรียบเทียบกับแบบ จำลองทางคณิตศาสตร์	61
4.7 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในทิศทาง $-x +y +z$ เปรียบเทียบกับแบบ จำลองทางคณิตศาสตร์	60
4.8 ตารางทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ ย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในทิศทาง $-x -y +z$ เปรียบเทียบกับแบบ จำลองทางคณิตศาสตร์	60
4.9 ตารางทดลองการควบคุมการเคลื่อนที่ของแขนกลแบบจุดต่อจุด	63
4.10 ตารางแสดงค่าเฉลี่ยในการเคลื่อนที่จริง และค่าเปอร์เซ็นต์ความผิดพลาดในการ เคลื่อนที่ไปในแต่ละตำแหน่ง	64

สารบัญรูป

รูปที่	หน้า
2.1	แสดงส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์ของแขนกล.....5
2.2	แสดงตัวอย่างส่วนประกอบทางกลของแขนกล.....6
2.3	แสดงการเคลื่อนที่ของวัตถุแข็งเกร็งใน 3 มิติ.....7
2.4	แสดงการหมุนของเฟรมใน 3 มิติ 8
2.5	แสดงเฟรมที่เกิดจากการหมุนรอบแกน z 9
2.6	แสดงการติดเฟรมเข้ากับวัตถุแข็งเกร็ง 10
2.7	แสดงการเคลื่อนที่ของเฟรมโดยการเลื่อน 12
2.8	แสดงสัญญาณ DH Parameters แบบดัดแปลง 16
2.9	แสดงองค์ประกอบโครงสร้าง กรอบอ้างอิงและพารามิเตอร์ของแขนกล 17
2.10	แสดงมุมมองด้านบนแสดงความสัมพันธ์ของเฟรมที่ {4} เทียบกับเฟรมอ้างอิง {0}..... 21
2.11	แสดงโครงสร้างสถาปัตยกรรมของ PLC..... 23
2.12	แสดงตัวอย่างการต่อคอมพิวเตอร์เข้ากับ PLC..... 25
2.13	แสดงวงรอบลำดับการทำงานของ PLC..... 27
2.14	แสดงการประยุกต์ใช้ KEPServerEX6 กับงานด้านต่าง ๆ 31
2.15	แสดงส่วนประกอบของระบบ Dynamic Data Exchange (DDE)..... 32
2.16	แสดงตัวอย่างการเชื่อมต่อคอมพิวเตอร์เข้ากับ PLC ผ่านพอร์ตอนุกรม RS232..... 32
2.17	แสดงมอเตอร์แบบสเต็ปขณะหยุดนิ่ง และขณะหมุนไปได้ครึ่งสเต็ป..... 33
2.18	แสดงโครงสร้างของมอเตอร์แบบสเต็ปชนิดไบโพลาร์ 34
2.19	แสดงการต่อใช้งาน Microstep Driver กับเครื่องควบคุม 36
2.20	แสดงไฟโตอิเล็กทรอนิกส์เซนเซอร์ PM-K44..... 37
3.1	แสดงภาพรวมการทำงานของโครงการ..... 38
3.2	แสดง SCARA Robot 4DOF 39
3.3	แสดงแผนผังการเชื่อมต่อวงจรระหว่าง PLC อุปกรณ์อินพุต-เอาต์พุต วงจรขับมอเตอร์..... 40 และมอเตอร์แบบสเต็ปสำหรับข้อต่อ 1 แกน
3.4	แสดงแผนผังการเชื่อมต่อวงจรของ PLC ในการต่อเข้ากับคอนเนคเตอร์เพื่อใช้สำหรับ 40 ต่อเข้ากับวงจรขับมอเตอร์ของมอเตอร์ข้อต่อแต่ละแกน
3.5	แสดงการออกแบบโมเดลจำลองของแขนกล SCARA Robot 4DOF..... 41
3.6	แสดงบล็อกไดอะแกรมบนโปรแกรม MATLAB Simulink ของโมเดลแขนกลจำลอง..... 41
3.7	แสดงโมเดลของแขนกลจำลองบนโปรแกรม MATLAB..... 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตัดข้อความอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.8 แสดง Flow chart เงื่อนไขการทำงานของแบบจำลองทางคณิตศาสตร์	43
3.9 แสดงการเขียนโปรแกรมสำหรับคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกล	44
แบบไปข้างหน้า	
3.10 แสดงการเขียนโปรแกรมสำหรับคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกล	44
แบบย้อนกลับ	
3.11 แสดงหน้าต่าง GUI และโมเดลแขนกลจำลอง	45
3.12 แสดงการตั้งค่า Pulse Output	46
3.13 แสดงโปรแกรมการควบคุมมอเตอร์แบบสแต็ปของแขนกล	46
3.14 แสดง Function Block ของการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบ	47
ไปข้างหน้า	
3.15 แสดง Structure Text ในการคำนวณค่าพิกัดตำแหน่งของส่วนปลายของแขนกล	47
3.16 แสดง Flowchart เงื่อนไขการทำงานของโปรแกรม PLC ควบคุมการทำงานของแขนกล	48
ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า	
3.17 แสดง Function Block ของการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกล	49
แบบย้อนกลับ	
3.18 แสดง Structure Text ในการคำนวณค่ามุมหรือระยะเลื้อนของข้อต่อแขนกล	49
3.19 แสดง Flowchart เงื่อนไขการทำงานของโปรแกรม PLC ควบคุมการทำงานของแขนกล	50
ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ	
3.20 แสดงเงื่อนไขการทำงานของโปรแกรมควบคุมแบบอัตโนมัติ	51
3.21 แสดง Structure Text ในการคำนวณค่าความเร็วของข้อต่อแต่ละแกนในการเคลื่อนที่ ...	52
3.22 แสดง Main Menu	52
3.23 แสดงหน้าต่าง Forward Kinematic Mode	53
3.24 แสดงหน้าต่าง Sequence Control Mode	53
3.25 แสดงหน้าต่าง VIEW DATA	54
3.26 แสดงหน้าต่าง POS SETTING	54
3.27 แสดงกำหนดการเชื่อมต่อฟังก์ชัน DDE	54
3.28 แสดงกำหนดการเชื่อมต่อฟังก์ชัน DDE (ต่อ)	55
3.29 แสดงการเชื่อมต่อระหว่าง KEPServerEX6 กับ CX-Programmer	55
3.30 แสดงการเชื่อมต่อระหว่าง KEPServerEX6 กับ CX-Programmer (ต่อ)	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และดัดแปลงXIอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.31 แสดงตัวอย่าง tag ของโปรแกรม CX-Programmer.....	56
3.32 แสดงตัวอย่าง Link Tag.....	56
4.1 แสดงพิกัดตำแหน่งเป้าหมายทั้ง 4 จุดบนตารางจุดภาคทั้ง 4.....	63
ก-1 แสดงแบบจำลองทางคณิตศาสตร์	70
ก-2 แสดงหน้าต่าง GUI สำหรับผู้ใช้งาน	70
ก-3 แสดงหน้าแรกของส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) และตัวเลือกในการควบคุม	71
ก-4 แสดงตัวอย่างหน้าต่างควบคุมการเคลื่อนที่ของแขนกล	72
ก-5 แสดงหน้าต่าง VIEW DATA	72
ก-6 แสดงการใช้งาน FORWARD KINEMATIC MODE.....	73
ก-7 แสดงการใช้งาน INVERSE KINEMATIC MODE.....	73
ก-8 แสดงการใช้งาน SEQUENCE CONTROL MODE	74
ก-9 แสดงการป้อนค่าพิกัดตำแหน่งเป้าหมายใน POS SETTING.....	75
ก-10 แสดงการป้อนค่าพิกัดตำแหน่งเป้าหมายใน Microsoft Excel.....	75
ข-1 แสดงคุณสมบัติทางเทคนิคของแขนกล SCARA Robot 4DOF.....	77
ข-2 แสดงคุณสมบัติทางเทคนิคของแขนกล SCARA Robot 4DOF (ต่อ).....	78
ข-3 แสดงคุณสมบัติทางเทคนิคของแขนกล SCARA Robot 4DOF (ต่อ).....	79
ข-4 แสดงคุณสมบัติทางเทคนิคของ PLC CP1H	80
ข-5 แสดงคุณสมบัติทางเทคนิคของ PLC CP1H (ต่อ).....	81
ข-6 แสดงคุณสมบัติทางเทคนิคของ PLC CP1H (ต่อ).....	82
ข-7 แสดงคุณสมบัติทางเทคนิคของ HMI Omron NB5QTW00B	83
ข-8 แสดงคุณสมบัติทางเทคนิคของมอเตอร์แบบสแต็ป.....	84
ข-9 แสดงคุณสมบัติทางเทคนิคของวงจรขับเคลื่อนมอเตอร์.....	85
ข-10 แสดงคุณสมบัติทางเทคนิคของวงจรขับเคลื่อนมอเตอร์ (ต่อ).....	86
ข-11 แสดงคุณสมบัติทางเทคนิคของโฟโตอิเล็กทริกเซนเซอร์	87
ค-1 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 1	89
ค-2 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 2	90
ค-3 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 3	91
ค-4 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 4	92
ค-5 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 5	93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ค-6 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 6	94
ค-7 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 7	95
ค-8 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 8	96
ค-9 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 9	97
ค-10 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 10	98
ค-11 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 11	99
ค-12 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 12	100
ค-13 แสดงโปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 13	101
ง-1 แสดงโปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ).....	103
ง-2 แสดงโปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ).....	104
ง-3 แสดงโปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ).....	105
ง-4 แสดงโปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ).....	106
ง-5 แสดงโปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ).....	107
ง-6 แสดงแปลงค่าพัลส์จากFloating to DWord	108
ง-7 แสดงคำนวณค่าความเร็ว	108
ง-8 แสดงแปลงค่าความเร็วจาก Float to DWord.....	109
ง-9 แสดงคำสั่งในการกดสวิทช์เพื่อย้ายข้อมูลที่เป็นพัลส์ไปยังบิตที่กำหนด	110
ง-10 แสดงส่งค่าพัลส์และความเร็วไปขับเคลื่อนมอเตอร์.....	111
ง-11 แสดงส่งค่าพัลส์และความเร็วไปขับเคลื่อนมอเตอร์ (ต่อ).....	112
ง-12 แสดงวัดค่าการเคลื่อนที่ของมอเตอร์	112
ง-13 แสดงคำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ	113
ง-14 แสดงคำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ (ต่อ).....	114
ง-15 แสดงคำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ (ต่อ).....	115
ง-16 แสดงคำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ (ต่อ).....	116
ง-17 แสดงคำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ (ต่อ).....	117

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ประเทศไทยในปัจจุบันกำลังเข้าสู่ยุคที่มีการพัฒนาเทคโนโลยีเพื่อนำไปใช้งานในด้านต่าง ๆ โดยเฉพาะในด้านอุตสาหกรรมที่กำลังมีแนวโน้มในการปรับตัวเข้าให้เป็นอุตสาหกรรมอัตโนมัติ โดยการนำเอาเทคโนโลยีอย่างเช่นแขนกลต่าง ๆ เข้ามาทำงานแทนที่มนุษย์และมีบทบาทที่สำคัญในกระบวนการผลิตมากขึ้น โดยปัจจัยที่ทำให้โรงงานอุตสาหกรรมต่าง ๆ ไม่ว่าจะเป็นอุตสาหกรรมการประกอบชิ้นส่วนยานยนต์ อุตสาหกรรมผลิตชิ้นส่วนอิเล็กทรอนิกส์ หรืออุตสาหกรรมรูปแบบอื่น ๆ ได้มีการนำเอาแขนกลเข้ามาใช้งานนั้นก็เพื่อต้องการยกระดับและเพิ่มขีดความสามารถของกระบวนการผลิตให้สูงขึ้น และลดต้นทุนในการผลิตลง เนื่องจากแขนกลสามารถทำงานได้อย่างต่อเนื่องหรือสามารถทำงานได้มากกว่าขีดจำกัดที่มนุษย์ทำได้ อีกทั้งแขนกลยังสามารถทำงานได้ถูกต้องและมีความเที่ยงตรงที่มากกว่ามนุษย์ ทำให้สามารถลดความผิดพลาดที่เกิดขึ้นในกระบวนการผลิตได้

โดยในการควบคุมการทำงานของแขนกล ให้สามารถเคลื่อนที่ไปยังตำแหน่งเป้าหมายใด ๆ ได้นั้น สามารถแบ่งประเภทของการควบคุมออกได้เป็น 2 วิธี อ้างอิงตามทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล ซึ่งจะประกอบด้วยการควบคุมโดยการป้อนค่ามุมของแต่ละข้อต่อของแขนกล (Joint Angle) ที่ต้องการให้เคลื่อนที่ไป แล้วนำมาคำนวณหาค่าของตำแหน่งสุดท้ายของส่วนปลายแขนกล (x, y, z) ด้วยจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า และการควบคุมอีกรูปแบบหนึ่งคือการป้อนค่าตำแหน่งสุดท้ายของส่วนปลายแขนกล (x, y, z) แล้วทำการคำนวณกลับด้วยจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับเพื่อหาค่าของมุมของแต่ละข้อต่อของแขนกล (Joint Angle) ที่จะทำให้แขนกลเคลื่อนที่ไปยังจุดนั้น ๆ ได้ ดังนั้นจะเห็นได้ว่าทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกลทั้งแบบไปข้างหน้าและแบบย้อนกลับนั้น สามารถนำมาวิเคราะห์เพื่อใช้ในการควบคุมแขนกลให้เคลื่อนที่ไปตามที่กำหนดได้

ทางผู้จัดทำจึงได้จัดทำโครงการการประยุกต์ PLC ควบคุมแขนกล SCARA Robot 4DOF ขึ้นมา เพื่อทำการควบคุมให้แขนกล SCARA Robot 4DOF มีการเคลื่อนที่เป็นไปตามทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ โดยโปรแกรมการทำงานผ่านเครื่องควบคุม PLC ทั้งนี้แขนกล SCARA Robot 4DOF นั้นจะมีพื้นที่การทำงานที่จำกัด เนื่องจากระยะความยาวของแต่ละข้อต่อและขอบเขตขององศาของมุมของแต่ละข้อต่อที่สามารถเคลื่อนที่ไปได้ ทำให้แขนกล SCARA Robot 4DOF จะสามารถเคลื่อนที่ไปได้เฉพาะเมื่อป้อนค่าอินพุตที่อยู่ในขอบเขตของพื้นที่การทำงานเพียงเท่านั้น จากปัญหานี้ ทางผู้จัดทำจึงได้มีการทำแบบจำลองการเคลื่อนที่ทางคณิตศาสตร์ จำลองการเคลื่อนที่ของแขนกล SCARA Robot 4DOF เพื่อให้ทราบถึง

ผลลัพธ์การเคลื่อนที่ของแขนกลในการเคลื่อนที่ไปยังตำแหน่งเป้าหมาย และเป็นตัวตรวจสอบด้วยว่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าอินพุตที่ป้อนนั้นมีค่าที่ถูกต้องและอยู่ในขอบเขตการทำงานของแขนกลหรือไม่ เพื่อป้องกันความเสียหายจากการเคลื่อนที่ที่ผิดพลาดก่อนนำค่าอินพุตไปป้อนให้แขนกลในทางปฏิบัติจริง รวมถึงยังช่วยให้สามารถเปรียบเทียบความถูกต้องของการเคลื่อนที่ เพื่อหาความคลาดเคลื่อนระหว่างแขนกลในแบบจำลอง กับแขนกลในทางปฏิบัติจริงว่ามีความแตกต่างกันมากน้อยเพียงใด

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อประยุกต์ใช้เครื่องควบคุม PLC ให้ทำการควบคุมแขนกล SCARA Robot 4DOF ให้สามารถเคลื่อนที่ได้ ตามทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ
2. เพื่อศึกษาการใช้งานและการควบคุมตำแหน่งของมอเตอร์แบบสเต็ป สำหรับการเคลื่อนที่ของข้อต่อต่าง ๆ ของแขนกล SCARA Robot 4DOF
3. เพื่อศึกษาวิธีการวิเคราะห์ และคำนวณค่าพารามิเตอร์ของทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล ซึ่งประกอบไปด้วยจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ
4. เพื่อศึกษาวิธีในการออกแบบแบบจำลองทางคณิตศาสตร์ เพื่อแสดงผลการเคลื่อนที่ของแขนกล SCARA Robot ในรูปแบบของกราฟฟิกด้วยโปรแกรม MATLAB Simulink

1.3 ขอบเขตของปริญญานิพนธ์

1. จัดหาแขนกล SCARA Robot 4DOF และทำการออกแบบวงจรควบคุมโดยมีชุดขับเป็นมอเตอร์แบบสเต็ป และใช้เครื่องควบคุม PLC ที่มี High Speed O/P ไม่น้อยกว่า 3 แกน พร้อมทั้งมีระบบ Manual ที่อาศัย Encoder 1 ชุด ในการเลือกควบคุมในแต่ละแกนได้
2. มีระบบความปลอดภัยโดยติดตั้ง Sensor กำกับการเคลื่อนที่ทั้ง 3 แกนให้มีการทำงานอยู่ในพื้นที่การทำงานเพียงเท่านั้น
3. การควบคุมตำแหน่งของแขนกล SCARA Robot 4DOF พิสูจน์ได้จากสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและย้อนกลับด้วยวิธีการตรีโกณมิติหรือเมทริกซ์
4. สร้างแบบจำลองทางคณิตศาสตร์ เพื่อจำลองการเคลื่อนที่ของแขนกล SCARA Robot 4DOF ด้วยโปรแกรม MATLAB และ Simulink ที่สอดคล้องกับการเคลื่อนที่ของแขนกลในทางปฏิบัติจริงที่ได้มีการจัดหามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. ศึกษาหลักการทำงานและการเคลื่อนที่ของแขนกล SCARA Robot 4DOF และวิธีในการวิเคราะห์หาสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า และแบบย้อนกลับที่สัมพันธ์กับการเคลื่อนที่ของแขนกล
2. ศึกษาวิธีออกแบบวงจรของชุดควบคุมที่เชื่อมต่อเข้ากับเครื่องควบคุม PLC
3. ศึกษาการคำนวณหาองศาและรอบการหมุนที่มอเตอร์แบบสเต็ปจะเคลื่อนที่ไปหากมีการสั่งให้มีการเคลื่อนที่
4. ศึกษาวิธีในการเขียนโปรแกรม PLC เพื่อควบคุมการทำงานของมอเตอร์แบบสเต็ป ด้วยภาษาแบบ Ladder, Structure Text และ Function Block
5. ศึกษาวิธีในการสร้างแบบจำลองทางคณิตศาสตร์ เพื่อจำลองการเคลื่อนที่ของแขนกล SCARA Robot 4DOF บนโปรแกรม MATLAB และ Simulink

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำการวิเคราะห์ทางจลนศาสตร์ของแขนกลทั้งแบบไปข้างหน้า และแบบย้อนกลับมาประยุกต์ใช้ในการเขียนโปรแกรมเพื่อควบคุมแขนกล SCARA Robot 4DOF ด้วยเครื่องควบคุม PLC ให้เคลื่อนที่ไปยังตำแหน่งเป้าหมายที่กำหนดได้
2. สามารถจำลองการเคลื่อนที่ของแขนกล SCARA Robot 4DOF เพื่อตรวจสอบการเคลื่อนที่ที่มีความถูกต้อง ในแบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นได้
3. สามารถพัฒนาโปรแกรมควบคุมการเคลื่อนที่ของแขนกล SCARA Robot 4DOF ให้มีการเคลื่อนที่เป็นแบบ Automatic และนำองค์ความรู้ที่ได้ไปประยุกต์ในโรงงานอุตสาหกรรม

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ในการจัดทำปริญญานิพนธ์การประยุกต์ PLC ควบคุมแขนกล Scara Robot 4DOF ตาม จลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ ทางคณะผู้จัดทำได้ ทำการศึกษา ทบทวนหนังสือ บทความ ผลงานต่าง ๆ และเอกสารที่เกี่ยวข้องกับการจัดทำปริญญานิพนธ์ทั้งหมด ซึ่งเป็นส่วนอ้างอิงในการกล่าวถึงทฤษฎี หลักการทำงานของอุปกรณ์ต่าง ๆ และความรู้ที่เกี่ยวข้อง ซึ่งเป็นส่วนสำคัญในการจัดทำปริญญานิพนธ์โดยเฉพาะอย่างยิ่งในการออกแบบระบบ เขียนโปรแกรมคำสั่ง รวมไปถึงการทดสอบการทำงาน ซึ่งจะอธิบายและมีเนื้อหาดังต่อไปนี้

- 2.1 หลักการพื้นฐานของแขนกล
- 2.2 ทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล
- 2.3 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC)
- 2.4 ระบบเลขฐาน
- 2.5 OLE for Process Control (OPC Server)
- 2.6 มอเตอร์แบบสแต็ป
- 2.7 โฟโตอิเล็กทริกเซนเซอร์

2.1 หลักการพื้นฐานของแขนกล

ความหมายของแขนกลนั้น คือตัวจัดการ (Manipulator) ที่สามารถทำหน้าที่ได้หลายอย่าง และสามารถโปรแกรมการทำงานใหม่ได้ ที่ออกแบบมาเพื่อใช้งานสำหรับการเคลื่อนย้ายวัสดุ ชิ้นงาน หรือทำงานในด้านอื่น ๆ โดยที่การเคลื่อนที่จะถูกเขียนโปรแกรมเอาไว้ผ่านเครื่องควบคุม ทำให้แขนกลสามารถทำงานได้อย่างมีประสิทธิภาพ และแม่นยำ ซึ่งจากนิยามของคำว่า “การทำหน้าที่ได้หลายอย่าง” และ “การโปรแกรมใหม่ได้” ทำให้แขนกลนั้นมีความแตกต่างไปจากเครื่องจักรอัตโนมัติประเภทอื่นที่มีการควบคุมผ่านคอมพิวเตอร์และเครื่องควบคุมเหมือนกัน เช่น เครื่องซีเอ็นซี หรือ เครื่องพิมพ์ 3 มิติ เพราะว่าแขนกลมีความยืดหยุ่นในการทำงานที่มากกว่า ตัวอย่างเช่น เครื่องซีเอ็นซี ที่มีการทำงานที่เพียงแค่กัดชิ้นงานในพิกัด 3 มิติ และไม่สามารถนำมาใช้งานในด้านอื่น อย่างเช่น การเคลื่อนย้ายสิ่งของได้ แต่แขนกลนั้นสามารถนำมาติดหัวกัดและเขียนโปรแกรมให้กัดชิ้นงาน หรือสามารถที่จะเปลี่ยนจากหัวกัดเป็นหัวเครื่องมืออื่น ๆ และเขียนโปรแกรมใหม่ เพื่อนำไปใช้งานในการเคลื่อนย้ายวัตถุ หรือเชื่อมชิ้นงานได้

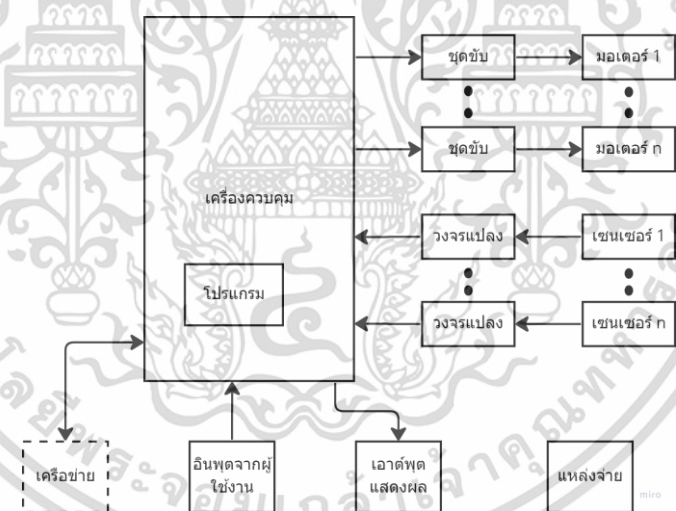
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 โครงสร้างของแขนกล

โครงสร้างของแขนกลประกอบด้วยอุปกรณ์ และองค์ประกอบต่าง ๆ ซึ่งสามารถแบ่งออกได้เป็น 2 ส่วน คือ ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์ และส่วนประกอบทางกลของแขนกล โดยสามารถอธิบายได้ดังนี้

2.1.1.1 ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์

ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์พื้นฐานของแขนกลนั้น ดังรูปที่ 2.1 ซึ่งจะประกอบไปด้วยตัวประมวลผลซึ่งเป็นเครื่องควบคุมที่ทำหน้าที่ประมวลผล และควบคุมการทำงานของแขนกลตามโปรแกรมที่มีการเขียนเอาไว้ โดยที่เครื่องควบคุมจะรับค่าข้อมูล ซึ่งเป็นค่าตำแหน่งต่าง ๆ จากผู้ใช้งานผ่านอุปกรณ์อินพุต เช่น HMI ส่วนมอเตอร์เปรียบเสมือนเป็นส่วนขับเคลื่อนในการเคลื่อนที่เพื่อให้ แขนกลเคลื่อนที่ไปสู่ตำแหน่งเป้าหมาย และแขนกลจะทำงานได้อย่างถูกต้องและแม่นยำได้ต้องมีเซ็นเซอร์หรืออุปกรณ์ตรวจรู้ เพื่อใช้วัดค่าตำแหน่งการเคลื่อนที่ของข้อต่อ หรืออาจจะเป็นเซ็นเซอร์เพื่อป้องกันการเคลื่อนที่ที่ผิดพลาดของแขนกล และส่วนประกอบทั้งหมดในส่วนนี้จะมีแหล่งจ่าย เพื่อให้อุปกรณ์ทุกส่วนนั้นสามารถทำงานได้



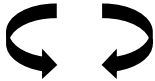
รูปที่ 2.1 ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์ของแขนกล

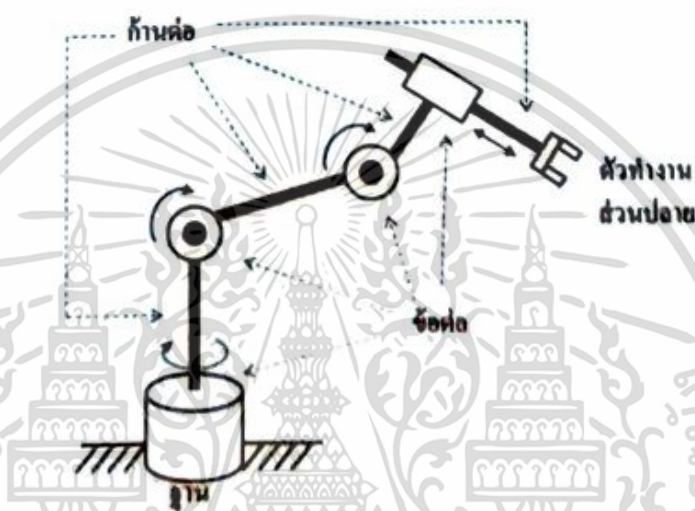
2.1.1.2 ส่วนประกอบทางกล

ส่วนประกอบทางกลเป็นส่วนประกอบที่มีความสำคัญสำหรับการนำไปวิเคราะห์ทางจลนศาสตร์ ซึ่งส่วนประกอบทางกลจะประกอบด้วย 1) ฐาน (Base) ซึ่งเป็นส่วนที่ถูกยึดติดอยู่กับที่ 2) ข้อต่อ (Joints) เป็นจุดเชื่อมต่อระหว่างส่วนต่าง ๆ ในแขนกล และเป็นส่วนที่เคลื่อนที่ได้ของแขนกล ซึ่งข้อต่อจะแบ่งออกเป็น 2 ประเภท แสดงดังตารางที่ 2.1 3) ก้านต่อ (Links) เป็นส่วนที่เชื่อมต่อระหว่างข้อต่อแต่ละจุด 4) ส่วนปลายของแขนกล หรือ เครื่องมือปลายแขน (End Effector) เป็นส่วนที่ติดตั้งเครื่องมือเพื่อใช้งานต่าง ๆ และเป็นส่วนที่สัมผัสกับชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ลักษณะของข้อต่อต่าง ๆ ที่ใช้ในแขนกล

ประเภท	อักษรย่อ	สัญลักษณ์	คำอธิบาย
แบบหมุน (Revolute)	R		การเคลื่อนที่แบบหมุนรอบแกน
แบบเชิงเส้น หรือ เลื่อน (Prismatic)	P		การเคลื่อนที่แบบเชิงเส้นเลื่อนไปตามแกน



รูปที่ 2.2 ตัวอย่างส่วนประกอบทางกลของแขนกล

2.1.2 องศาเสรี (Degree of freedom)

ในการเคลื่อนที่ในระบบ 3 มิติสามารถอธิบายได้โดยพารามิเตอร์ที่เรียกว่าค่าตำแหน่งและมุมการหมุนองศาเสรี (Degree of freedom) ซึ่งเป็นจำนวนของพารามิเตอร์ที่บ่งบอกถึงการเคลื่อนที่ของวัตถุที่เป็นอิสระ โดยค่าองศาเสรีจะมีมากหรือน้อยจะขึ้นอยู่กับโครงสร้างทางกายภาพของวัตถุว่าถูกจำกัดอยู่หรือไม่ ในทางแขนกลนั้นค่าพารามิเตอร์องศาเสรี จะบอกถึงจำนวนของจุดข้อต่อ ที่สามารถเคลื่อนที่ได้ของแขนกล ไม่ว่าจะเป็นการเคลื่อนที่แบบหมุน หรือแบบเลื่อนก็ตาม

2.1.3 พื้นที่การทำงาน (Workspace)

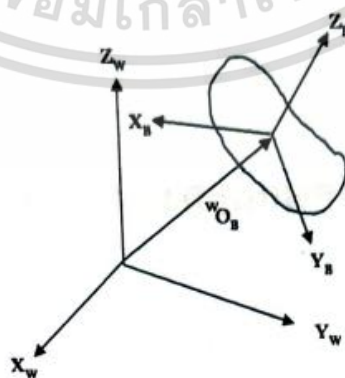
พื้นที่การทำงานของแขนกล (Workspace) คือ พื้นที่ หรือ ปริมาตร ทั้งหมดที่ส่วนปลายของแขนกลนั้นสามารถเข้าถึงได้ ซึ่งก็คือพื้นที่ที่แขนกลสามารถเคลื่อนที่ หรือ ทำงาน ไปยังจุดนั้นได้ โดยที่ไม่ถูกขวางกั้น หรือเกิดความเสียหาย

2.2 ทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล

ทฤษฎีจลนศาสตร์ เป็นการอธิบายถึงการเคลื่อนที่ของวัตถุโดยที่ไม่พิจารณาแรงที่มากระทำกับวัตถุและพลวัต ทำให้ตำแหน่งทางจลนศาสตร์จะขึ้นอยู่กับตำแหน่งทางเรขาคณิตของวัตถุ ในส่วนของการคิดจลนศาสตร์ของแขนกลนั้นจะขึ้นอยู่กับโครงสร้างของข้อต่อและก้านต่อ ซึ่งจลนศาสตร์ของแขนกลนั้น สามารถแบ่งออกได้เป็น 2 ส่วน คือ จลนศาสตร์แบบไปข้างหน้า (Forward Kinematics) ซึ่งเป็นการหาพิกัดและทิศทางในรูปของการแปลงเอกพันธ์ของส่วนปลายของแขนกล โดยที่ทราบตำแหน่งและมุมของข้อต่อทั้งหมด ในทางกลับกัน จลนศาสตร์แบบย้อนกลับ (Inverse Kinematics) จะเป็นการหาค่าตำแหน่งและมุมของข้อต่อ โดยที่ทราบพิกัดตำแหน่ง หรือเมทริกซ์เอกพันธ์ของส่วนปลายของแขนกลเมื่อเทียบกับเฟรมฐาน ซึ่งหลักการและวิธีการได้มาซึ่งสมการจลนศาสตร์ทั้ง 2 แบบ จะอธิบายในส่วนต่อไป

2.2.1 การเคลื่อนที่ของวัตถุแข็งเกร็ง

การเคลื่อนที่ของแขนกลจะมีลักษณะการเคลื่อนที่ที่เป็นแบบการเคลื่อนที่ของวัตถุแข็งเกร็ง กล่าวคือ วัตถุแข็งเกร็งจะไม่เปลี่ยนแปลงในขณะที่ยกแขนกลนั้นเคลื่อนที่ โดยที่ตำแหน่งสัมพัทธ์ระหว่างอนุภาคภายในไม่มีการเปลี่ยนแปลง การเคลื่อนที่ของวัตถุแข็งเกร็งนั้นจะประกอบด้วย การหมุน (Rotation) และการเลื่อน (Translation) ซึ่งก็คือการเคลื่อนที่ของข้อต่อแขนกลนั่นเอง ดังนั้น การศึกษาวิเคราะห์การเคลื่อนที่ของแขนกล จึงนำวิธีการวิเคราะห์การเคลื่อนที่ของวัตถุแข็งเกร็งมาใช้ ซึ่งวิธีการในการวิเคราะห์การเคลื่อนที่ของวัตถุแข็งเกร็งคือการติดเฟรมพิกัด (Coordinate frame) หรือเฟรม $\{i\}$ ที่วัตถุ ณ ตำแหน่งนั้น ๆ โดย i คือ หมายเลขหรือชื่อเฟรม ซึ่งเฟรมก็คือพิกัดคาร์ทีเซียนที่ประกอบไปด้วยแกน x , y , z ที่ตั้งฉากกันตามกฎมือขวา โดยเฟรมนี้จะยึดติดและเคลื่อนที่ไปพร้อมวัตถุ ดังนั้นสามารถอธิบายได้ว่าการเคลื่อนที่ของวัตถุแข็งเกร็งนั้นเป็นการหมุนและเลื่อนของเฟรม ณ ตำแหน่งนั้น ๆ เทียบกับเฟรมอ้างอิง



รูปที่ 2.3 การเคลื่อนที่ของวัตถุแข็งเกร็งใน 3 มิติ

2.2.2 การหมุนเฟรมใน 3 มิติ

จากรูปที่ 2.4 การหมุนของเฟรมใน 3 มิติทำได้โดยการฉายแกน x, y, z ของเฟรม $\{1\}$ ที่มีการหมุนลงบนแกนหลักของเฟรมอ้างอิง $\{0\}$ โดยใช้วิธีของผลคูณจุด (Dot product) และกำหนดให้แกน x, y, z ของเฟรมทั้ง 2 เป็นเวกเตอร์หนึ่งหน่วย (Unit vector) จะได้เมทริกซ์การหมุน (Rotation Matrix) ที่สัมพันธ์กับการหมุนระหว่าง 2 เฟรมได้ว่า

$${}^0R_1 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} \quad (2.1)$$

รูปที่ 2.4 การหมุนของเฟรมใน 3 มิติ

ในทางกลับกันหากต้องเริ่มจากเฟรม $\{1\}$ และหมุนให้เป็นเฟรม $\{0\}$ จะได้เมทริกซ์การหมุนเป็น

$${}^0R_1 = \begin{bmatrix} x_0 \cdot x_1 & y_0 \cdot x_1 & z_0 \cdot x_1 \\ x_0 \cdot y_1 & y_0 \cdot y_1 & z_0 \cdot y_1 \\ x_0 \cdot z_1 & y_0 \cdot z_1 & z_0 \cdot z_1 \end{bmatrix} \quad (2.2)$$

โดยคุณสมบัติสลับที่ของผลคูณจุด ($x_i \cdot y_j = x_j \cdot y_i$) จะได้เป็น

$${}^0R_1 = {}^0R_1^T \quad (2.3)$$

ซึ่งในมุมมองทางเรขาคณิตและตรีโกณมิติ การหมุนเฟรมจากเฟรม $\{1\}$ เป็นเฟรม $\{0\}$

เป็นการดำเนินการผกผันกับเมทริกซ์การหมุนโดยการนำทรานสโพสของสมการเมทริกซ์การหมุนคูณเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับตัวเอง ดังสมการที่ (2.4) ซึ่งเมทริกซ์ที่มีคุณสมบัติเช่นนี้จะถูกเรียกว่า เมทริกซ์เชิงตั้งฉาก (Orthogonal Matrix) หรือ $R \in SO(n)$ ซึ่ง $SO(n)$ คือกลุ่มเชิงตั้งฉากพิเศษ (Special Orthogonal group) จำนวน n

$${}^0R_1^T = {}^0R_1^{-1} \quad (2.4)$$

โดยในลำดับถัดไปจะเป็นการหาเมทริกซ์การหมุนรอบแกน z จากรูปที่ 2.5 เฟรม {1} เกิดจากการหมุนรอบแกน z_0 ของเฟรม {0} เป็นมุม θ โดยใช้สมการความสัมพันธ์ของผลคูณจุดกับเวกเตอร์ใน 3 มิติ จะได้ว่า

$$v_0 \cdot v_1 = \|v_0\| \|v_1\| \cos(\theta) \quad (2.5)$$

รูปที่ 2.5 เฟรมที่เกิดจากการหมุนรอบแกน z

จากนั้นทำการแทนค่าสมการที่ (2.5) ในตัวแปรแต่ละตัวในสมการที่ (2.1) และจัดรูปสมการให้เหลือเพียงแค่ตัวแปร θ ตัวเดียวโดยใช้การแทรกค่าตรีโกณมิติจาก

$$\begin{aligned} x_0 &= \cos(\theta)x_1 - \sin(\theta)y_1 \\ y_0 &= \sin(\theta)x_1 + \cos(\theta)y_1 \\ z_0 &= z_1 \end{aligned} \quad (2.6)$$

และนอร์มของเวกเตอร์หนึ่งหน่วยมีค่าเท่ากับ 1 ดังนั้นจะได้เมทริกซ์การหมุนจะมีค่าเท่ากับ

$${}^0R_1 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเมทริกซ์การหมุนรอบแกนในสมการที่ (2.7) คือ เมทริกซ์การหมุนพื้นฐาน (pure rotation matrix) โดยในสมการที่ (2.7) นั้นเป็นการหมุนรอบแกน z ซึ่งสามารถแทนด้วยสัญลักษณ์ $R_{z,\theta}$ จะได้สมการเมทริกซ์การหมุนพื้นฐานรอบแกน z เป็น

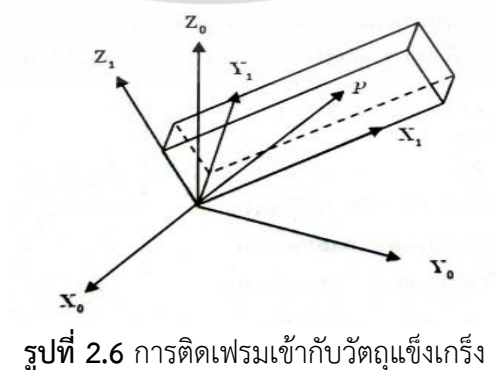
$$R_{z,\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

ในทำนองเดียวกันสามารถหาเมทริกซ์การหมุนพื้นฐานรอบแกน x และ y ได้ดังนี้

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (2.9)$$

$$R_{y,\beta} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2.10)$$

ในการประยุกต์เมทริกซ์การหมุนเพื่อใช้ในการวิเคราะห์การเคลื่อนที่ของแขนกล ทำได้โดยการติดเฟรมเข้ากับข้อต่อและก้านต่อของแขนกล และทำการหาความสัมพันธ์ของเฟรมที่หมุนเปรียบเทียบกับเฟรมอ้างอิง จากรูปที่ 2.4 วัตถุแข็งเกร็งมีมุมหนึ่งติดกับจุดกำเนิดของเฟรมอ้างอิง $\{0\}$ ซึ่งมีการหมุนได้รอบทิศทาง และถูกยึดติดอยู่กับที่ไม่ให้เลื่อนออกจากจุดกำเนิดของเฟรม $\{0\}$ และติดเฟรม $\{1\}$ เข้ากับวัตถุ โดยกำหนดให้มีจุดกำเนิดเดียวกันกับเฟรม $\{0\}$ และหมุนไปพร้อมกับวัตถุ โดยมีจุด P ทำหน้าที่ในการบอกตำแหน่งของวัตถุที่สัมพันธ์กับเฟรม $\{0\}$



รูปที่ 2.6 การติดเฟรมเข้ากับวัตถุแข็งเกร็ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการพิจารณาที่จุด P ซึ่งถูกชี้โดยเวกเตอร์ p สมมติว่า ${}^1p$ เป็นตำแหน่งของจุด P เทียบกับเฟรม $\{1\}$ จะได้

$$p = ux_1 + vy_1 = wz_1 \quad (2.11)$$

สามารถหาพิกัดของจุด P เทียบกับ ${}^0p$ ดังนี้

$${}^0p = \begin{bmatrix} p \cdot x_0 \\ p \cdot y_0 \\ p \cdot z_0 \end{bmatrix} \quad (2.12)$$

เมื่อทำการแทนค่าสมการที่ (2.11) ลงในสมการที่ (2.12) และเปรียบเทียบกับนิยามของเมทริกซ์การหมุนตามสมการที่ (2.1) จะเขียนใหม่ได้เป็น

$${}^0p = \begin{bmatrix} x_1x_0 & y_1x_0 & z_1x_0 \\ x_1y_0 & y_1y_0 & z_1y_0 \\ x_1z_0 & y_1z_0 & z_1z_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \text{หรือ} \quad {}^0p = {}^0R_1 {}^1p \quad (2.13)$$

ในกรณีที่มีการหมุนของเฟรมมากกว่า 1 ครั้ง สามารถนำเมทริกซ์ของการหมุนในแต่ละครั้งมาทำการคูณกัน โดยจะต้องพิจารณาตามลำดับการหมุนแต่ละครั้งเทียบกับเฟรมสุดท้าย จากสมการที่ (2.13) เป็นเมทริกซ์การหมุนจากเฟรม $\{0\}$ เป็น $\{1\}$ สมมติว่ามีการหมุนเฟรมอีกครั้งจากเฟรม $\{1\}$ ไป $\{2\}$ สามารถเขียนความสัมพันธ์ของจุด p เมื่อเทียบกับเฟรม $\{0\}$, $\{1\}$, $\{2\}$ ดังนี้

$${}^0p = {}^0R_2 {}^2p \quad (2.14)$$

$${}^0p = {}^0R_2 {}^2p \quad (2.15)$$

$${}^0p = {}^0R_1 {}^1R_2 {}^2p \quad (2.16)$$

ดังนั้น

$${}^0R_2 = {}^0R_1 {}^1R_2 \quad (2.17)$$

ในการหาเมทริกซ์การหมุนรวมรอบแกน x, y, z เมื่อมีการหมุนเฟรมหลายครั้ง ลำดับของการหมุนจะพิจารณาจากชนิดของเฟรมที่มีการหมุน โดยหากเป็นการหมุนเทียบเฟรมปัจจุบันให้คูณเมทริกซ์การหมุนทางด้านหลัก แต่ถ้าเป็นการหมุนเทียบเฟรมอ้างอิงให้คูณเมทริกซ์การหมุนทางด้านหน้า ดังนั้นการคูณเมทริกซ์การหมุนรอบแกน x, y, z จะมีลำดับดังนี้

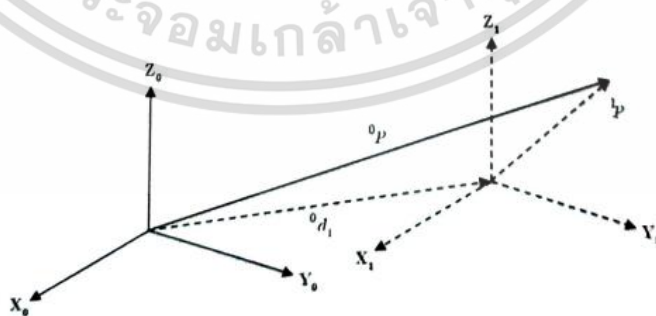
$$R = R_{x,\delta} R_{z,\alpha} R_{x,\theta} R_{z,\phi} R_{y,\beta} \quad (2.18)$$

เมื่อ δ และ α คือมุมของการหมุนรอบแกนของเฟรมอ้างอิง ส่วน θ, ϕ และ β คือมุมของการหมุนรอบแกนของเฟรมปัจจุบัน

2.2.3 การเคลื่อนที่จากการเลื่อน

การเคลื่อนที่ของแขนกลจะเป็นการผสมระหว่างการเคลื่อนที่แบบเลื่อนและแบบหมุน ในการพิจารณาการเลื่อนของเฟรม ดังรูปที่ 2.7 เฟรม {1} เกิดจากการเลื่อนเฟรม {0} ไปยังตำแหน่ง P โดยที่ทั้ง 2 เฟรมนั้นมีทิศทางที่เหมือนกัน เมื่อ ${}^1p$ คือตำแหน่งของจุด P เทียบกับเฟรม {1} การหา ${}^0p$ หรือตำแหน่งของจุด P เทียบกับเฟรม {0} จะหาได้จากการบวกเวกเตอร์ ${}^1p$ กับเวกเตอร์ที่ชี้ไปยังจุดกำเนิดของเฟรม {1} ดังสมการ

$${}^0p = {}^1p + {}^0d_1 \quad (2.19)$$



รูปที่ 2.7 การเคลื่อนที่ของเฟรมโดยการเลื่อน

แต่ในการเลื่อนจากเฟรม $\{0\}$ เป็น $\{1\}$ นั้นไม่จำเป็นที่จะต้องมีการกำหนดทิศทางให้มีความทิศทางเดียวกัน ซึ่งการหมุนใด ๆ ของเฟรม $\{1\}$ เทียบกับเฟรม $\{0\}$ สามารถแทนได้ด้วยเมทริกซ์การหมุน ${}^0R_1$ ดังนั้น

$${}^0p = {}^0R_1 {}^1p + {}^0d_1 \quad (2.20)$$

วิธีข้างต้นนี้สามารถใช้สำหรับการเปรียบเทียบพิกัดของระบบที่มีจำนวนหลายเฟรมได้ แต่จะมีความยาวและซับซ้อนมากขึ้นตามจำนวนของเฟรมที่มี เนื่องจากรูปของสมการมีการพจน์ของการคูณและการบวกซ้อนกัน

2.2.4 การแปลงเอกพันธ์

จากการวิเคราะห์การหมุนใน 3 มิติที่มีการหมุนเฟรมหลาย ๆ ครั้งนั้นทำให้ได้สมการซึ่งเป็นการคูณเมทริกซ์การหมุนทั้งหมดเข้าด้วยกัน ดังสมการที่ (2.18) ขณะที่การวิเคราะห์การเลื่อนของเฟรมจะใช้วิธีการบวกเวกเตอร์เวกเตอร์ที่ชี้ไปยังจุดกำเนิดของเฟรมปัจจุบันคูณกับเมทริกซ์การหมุน ดังสมการที่ (2.19) ซึ่งทั้ง 2 สมการนี้สามารถใช้ในการอธิบายการหมุนและเลื่อนของเฟรมที่ติดอยู่กับวัตถุได้ เพียงแต่ต้องมีการจัดรูปสมการใหม่ ซึ่งสามารถเขียนสมการใหม่ให้อยู่ในรูปเมทริกซ์ได้เป็น

$$T = \begin{bmatrix} R & d \\ 0 & 0 & 0 & 1 \end{bmatrix}, R \in SO(3), d \in R^3 \quad (2.21)$$

เมทริกซ์ T ตามสมการที่ (2.21) คือเมทริกซ์การแปลงเอกพันธ์ (Homogeneous Transformation) ซึ่งเป็นเมทริกซ์ที่รวมเมทริกซ์การหมุนและเมทริกซ์การเลื่อนเข้าด้วยกัน เพื่อลดรูปสมการให้สั้นลงและเหมาะสมกว่าสมการที่ (2.18) และสมการที่ (2.19) โดยการเมทริกซ์ของการแปลงเอกพันธ์ จะอยู่ในรูปสมการดังนี้

$${}^{i-1}T_i = \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & o & a & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.22)$$

เมื่อ $n = [n_x \ n_y \ n_z]^T$, $o = [o_x \ o_y \ o_z]^T$, $a = [a_x \ a_y \ a_z]^T$ คือเวกเตอร์แทนทิศทางของ x, y, z ตามลำดับ เทียบกับเฟรม $\{i\}$ ส่วน $d = [d_x \ d_y \ d_z]^T$ คือเวกเตอร์บอกพิกัดของจุด

กำเนิดของเฟรม $\{i\}$ เทียบกับเฟรม $\{0\}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงเอกพันธ์พื้นฐานจะประกอบด้วยเมทริกซ์การเลื่อนและเมทริกซ์การหมุนรวม
6 สมการ ดังนี้

$$\begin{aligned}
 Transl_{x,a} &= \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & Trot_{x,\phi} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\phi & -s_\phi & 0 \\ 0 & s_\phi & c_\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 Transl_{y,b} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & Trot_{y,\theta} &= \begin{bmatrix} c_\theta & 0 & s_\theta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\theta & 0 & c_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 Transl_{z,c} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} & Trot_{z,\psi} &= \begin{bmatrix} c_\psi & -s_\psi & 0 & \alpha \\ s_\psi & c_\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2.23}$$

ซึ่งเมทริกซ์ $Transl_{x,a}$, $Transl_{y,b}$, $Transl_{z,c}$ คือเมทริกซ์การเลื่อนในแนวแกน x, y, z ตามลำดับ และเมทริกซ์ $Trot_{x,\phi}$, $Trot_{y,\theta}$, $Trot_{z,\psi}$ คือเมทริกซ์การหมุนรอบแกน x, y, z ตามลำดับ

2.2.5 สายโซ่จลนศาสตร์

การนำหลักการวิเคราะห์การเคลื่อนที่ของวัตถุแข็งเกร็ง และการแปลงเอกพันธ์ที่ได้มีการกล่าวถึงในหัวข้อที่ผ่านมาข้างต้นนำมาประยุกต์ใช้ในการวิเคราะห์หาจลนศาสตร์การเคลื่อนที่ของแขนกลนั้น ทำได้โดยการติดเฟรมให้กับข้อต่อทั้งหมดจากฐานไปสู่ส่วนปลายของแขนกล ซึ่งจากการศึกษาของสาเสรีใน 3 มิติ พบว่าการติดเฟรมให้แขนกลโดยทั่วไปจะทำให้แขนกลมีองศาเสรี 6 ตัว แปรอิสระ (3 ตัวแปรตำแหน่งและ 3 ตัวแปรทิศทาง) ทำให้ในเมทริกซ์เอกพันธ์มีตัวแปรที่ไม่ทราบค่าจำนวนมาก และตัวแปรจะมีค่าเพิ่มขึ้นตามจำนวนของก้านต่อแขนกล ดังนั้นจึงทำการลดจำนวนของตัวแปรลง โดยตั้งสมมติฐานกำหนดให้แต่ละข้อต่อมีองศาเสรีเท่ากับ 1 ทำให้ฟังก์ชันของตัวแปรเหลือเพียง 1 ตัวแปรไม่ทราบค่า เรียกว่า ตัวแปรข้อต่อ (Joint variable) ซึ่งจะใช้สัญลักษณ์ q_i เมื่อ q_i จะมีค่าเท่ากับ θ_i ซึ่งเป็นองศาการหมุนสำหรับข้อต่อแบบหมุน หรือ q_i จะมีค่าเท่ากับ d_i ซึ่งเป็นระยะทางที่เคลื่อนที่ของข้อต่อแบบเลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวิธีการลดตัวแปรในข้างต้นสามารถกำหนดเมทริกซ์เอกพันธ์ที่อธิบายการเคลื่อนที่ของเฟรมสำหรับแต่ละข้อต่อได้ดังนี้

$$A_i = A_i(q_i) \quad (2.24)$$

ดังนั้นสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า จะเขียนได้เป็น

$${}^0T_n = A_1(q_1)A_2(q_2)\dots A_n(q_n) \quad (2.25)$$

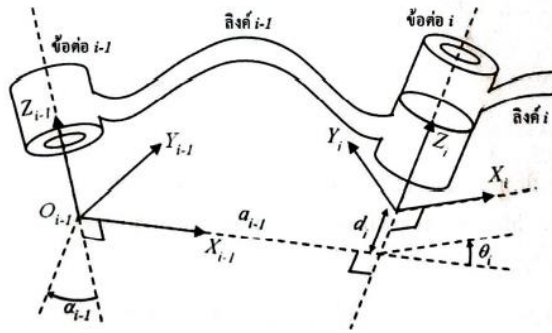
2.2.6 จลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าโดยวิธีการของเดนาวิท-ฮาร์เทนเบอร์กแบบดัดแปลง

สัญนิยมของเดนาวิท-ฮาร์เทนเบอร์ก (Denavit-Hartenberg convention) หรือ สัญนิยม DH Parameters เป็นวิธีหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าโดยการนำเฟรมของข้อต่อแต่ละเฟรมมาต่ออนุกรมกัน ทำให้เกิดเป็นฟังก์ชันของตัวแปรที่ไม่ทราบค่าเพียงจำนวน 4 ตัวแปร ซึ่ง สัญนิยม DH Parameters แบบดัดแปลงจะมีความแตกต่างจากสัญนิยม DH Parameters แบบปกติ ตรงที่ลำดับในการคูณกันของการแปลงพื้นฐานจะไม่เหมือนกัน โดยสัญนิยม DH Parameters แบบดัดแปลงจะกำหนดการแปลงเอกพันธ์ A_i แต่ละสมการแทนด้วยผลคูณของการแปลงพื้นฐานตามลำดับ ดังนี้

$$A_i = \text{Transl}_{x,a_i} \text{Trot}_{x,\alpha_i} \text{Transl}_{z,d_i} \text{Trot}_{z,\theta_i}$$

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & d_i \\ \sin(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\cos(\alpha_i) & -\sin(\alpha_i) & -\alpha_i\sin(\alpha_i) \\ \sin(\theta_i)\sin(\alpha_i) & \cos(\theta_i)\sin(\alpha_i) & \cos(\alpha_i) & \alpha_i\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

โดย พารามิเตอร์ DH จะประกอบไปด้วย ตัวแปร 4 ตัว ได้แก่ a_i คือความยาวก้านต่อ (link length), α_i คือมุมบิดก้านต่อ (link twist), d_i คือออฟเซตก้านต่อ (link offset) และ θ_i คือมุมข้อต่อ (joint angle) ซึ่งจะมีเพียงพารามิเตอร์เดียวที่เป็นตัวแปรของข้อต่อโดยที่ θ_i ที่เป็นตัวแปรสำหรับข้อต่อแบบหมุน และ d_i สำหรับข้อต่อแบบเลื่อน และพารามิเตอร์ที่เหลืออีก 3 ตัวจะกำหนดเป็นค่าคงที่



รูปที่ 2.8 สัญนิยม DH Parameters แบบดัดแปลง

2.2.6.1 การติดเฟรมให้กับแขนกลตามหลักการของวิธีสัญนิยม DH Parameters

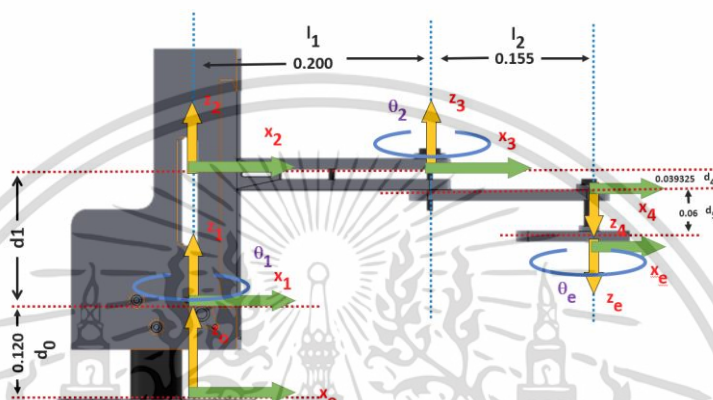
ในหัวข้อนี้จะอธิบายการติดเฟรมให้กับแขนกลตามหลักการของสัญนิยม DH Parameters ซึ่งจะมีกฎเกณฑ์และขั้นตอนตามลำดับดังนี้

- 1) ติดแกน z_i ให้กับทุกข้อต่อทั้งหมด โดยเริ่มจากข้อต่อแรกจากฐานไปยังข้อต่อสุดท้ายของส่วนปลายแขนกล ซึ่งการติดแกน z_i จะขึ้นอยู่กับชนิดของข้อต่อ โดยข้อต่อแบบหมุนการติดแกน z_i เป็นแกนหมุนของข้อต่อ ส่วนข้อต่อแบบเลื่อนจะติดแกน z_i ตามแนวเคลื่อนที่ของแกน
- 2) กำหนดตำแหน่งของเฟรมฐาน หรือเฟรม {0} บนแกน z_0 ซึ่งโดยทั่วไปจะเลือกจุดที่ยึดติดกับพื้น จากนั้นกำหนดจุดกำเนิด แล้วจึงติดแกน x_0 และ y_0 โดยอ้างอิงตามกฎมือขวา
- 3) การกำหนดตำแหน่งและทิศทางของเฟรมข้อต่อ ซึ่งจะเป็นการหาตำแหน่งของจุดกำเนิดและทิศทางของแกนสำหรับเฟรมทุกเฟรมที่ถูกติดตั้งลงบนข้อต่อของแขนกล ซึ่งจะมีอยู่ 3 กรณี ได้แก่
 - กรณีที่ 1 แกน z_{i-1} และ z_i ไม่ได้อยู่แนวเดียวกัน ให้ทำการหาเส้นตั้งฉากที่สั้นที่สุด และหาจุดตัดระหว่าง z_{i-1} และ z_i เพื่อกำหนดเป็นจุดกำเนิด ส่วนแกน x_i และ y_i จะเป็นไปตามกฎมือขวา
 - กรณีที่ 2 แกน z_{i-1} และ z_i ขนานกัน สามารถกำหนดจุดกำเนิด และแกน x_i ได้ตามแนวเส้นตั้งฉากระหว่างแกนทั้ง 2 โดยที่ y_i จะเป็นไปตามกฎมือขวา
 - กรณีที่ 3 แกน z_{i-1} และ z_i ตัดกัน กำหนดจุดกำเนิดที่จุดตัดระหว่างแกนทั้ง 2 และเลือกแกน x_i ตั้งฉากกับระนาบระหว่างแกน และ y_i จะเป็นไปตามกฎมือขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทำการติดเฟรมให้กับข้อต่อของแขนกลทั้งหมดแล้ว ในส่วนถัดไปจะเป็นการหาจลนศาสตร์การเคลื่อนที่ของแขนกล SCARA Robot 4DOF โดยการนำเอาทฤษฎีการวิเคราะห์การโดยวิธีสัญญาณ DH Parameters แบบดัดแปลงมาประยุกต์ใช้

2.2.7 การหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าของ SCARA Robot 4DOF ด้วยวิธีสัญญาณ DH Parameters แบบดัดแปลง



รูปที่ 2.9 แสดงองค์ประกอบโครงสร้าง กรอบอ้างอิง และพารามิเตอร์ของแขนกล

จากรูป 2.9 เป็นการวิเคราะห์หาค่า DH Parameters แบบดัดแปลงโดยการเริ่มพิจารณาจากโครงสร้าง องค์ประกอบ และการวางตัวของข้อต่อต่าง ๆ จากนั้นทำการติดเฟรมพิกัดเข้าที่ข้อต่อทั้งหมดโดยใช้กฎการติดเฟรมดังหัวข้อที่ 2.2.6.1 ซึ่งความสัมพันธ์ของเฟรมแต่ละเฟรมสามารถนำมาเขียนเป็นตาราง DH Parameters ได้ดังนี้

ตารางที่ 1 DH parameters ของ SCARA Robot 4DOF

$i-1$	i	a_i	α_i	d_i	θ_i
0	1	0	0	d_0	θ_1
1	2	0	0	d_1	0
2	3	l_1	0	0	θ_2
3	4	l_2	$\frac{\pi}{2}$	d_4	θ_3

เมื่อ a_i คือค่าคงที่ของระยะตามแนวแกน x_i จากจุด o_i ถึงจุดตัดระหว่างแกน x_i และ z_{i-1}

d_i คือค่าระยะเลื่อนตามแนวแกน z_{i-1} จากจุด o_{i-1} ถึงจุดตัดระหว่างแกน x_i เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ z_{i-1} โดยจะเป็นตัวแปรของข้อต่อแบบเลื่อน และเป็นค่าคงที่ของข้อต่อแบบหมุน

α_i คือค่าคงที่ของมุมระหว่างแกน z_{i-1} และ z_i วนรอบแกน x_i

θ_i คือค่าของมุมระหว่างแกน x_{i-1} และ x_i วนรอบแกน z_{i-1} ซึ่งค่านี้จะเป็นตัวแปรสำหรับข้อต่อแบบหมุน และเป็นค่าคงที่สำหรับข้อต่อแบบเลื่อน

จากค่า DH Parameters ในตารางที่ 2.2 สามารถแทนลงในสมการที่ (2.26) เพื่อคำนวณหาเมทริกซ์การแปลงเอกพันธ์จากเฟรมแรกไปยังเฟรมสุดท้ายได้ดังนี้

เมทริกซ์การแปลงจากเฟรม {0} ไปยัง {1}

$${}^0A_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.27)$$

เมทริกซ์การแปลงจากเฟรม {1} ไปยัง {2}

$${}^1A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.28)$$

เมทริกซ์การแปลงจากเฟรม {2} ไปยัง {3}

$${}^2A_3 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_1 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.29)$$

เมทริกซ์การแปลงจากเฟรม {3} ไปยัง {4}

$${}^3A_4 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_2 \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & 0 \\ 0 & 0 & -1 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.30)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการศึกษาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมทริกซ์การแปลงจากเฟรม {4} ไปยังส่วนปลายของแขนกล {e}

$${}^4A_e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.31)$$

ดังนั้นจลนศาสตร์การเคลื่อนที่ของแขนกล SCARA Robot 4DOF แบบไปข้างหน้า จะได้เป็น

$${}^0T_e = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_e$$

$${}^0T_e = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1) & -\cos(\theta_1) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ 0 & 0 & -1 & d_0 + d_1 - d_4 - d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

จากสมการที่ (2.32) จะได้ค่าพิกัด x, y, z ของจุดกำเนิดของส่วนปลายของแขนกล ดังต่อไปนี้

$$x_e = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \quad (2.33)$$

$$y_e = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \quad (2.34)$$

$$z_e = d_0 + d_1 - d_4 - d_5 \quad (2.35)$$

ซึ่งเมื่อทำการกำหนดค่ามุมของข้อต่อและแทนค่าตัวแปรลงในสมการที่ (2.33) ถึงสมการที่ (2.35) จะทำให้สามารถคำนวณเป็นค่าตำแหน่งพิกัด x, y, z ของส่วนปลายของแขนกลสำหรับจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าได้

2.2.8 การหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับของ SCARA Robot 4DOF

จลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ จะใช้วิธีการคำนวณค่าตัวแปรของแต่ละข้อต่อโดยกำหนดตำแหน่งที่ปลายของแขนกล โดยวิธีการคำนวณจลนศาสตร์แบบย้อนกลับนี้ในบางครั้งมีได้หลายคำตอบ (Several Solution) หรือไม่สามารถหาคำตอบ (No Answer) ได้ จากข้อจำกัดในเรื่องของพื้นที่การทำงานของแขนกลที่จะสามารถเคลื่อนที่ไปได้ สำหรับในปริภูมิงานฉบับนี้การวิเคราะห์และคำนวณหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบผกผันจะใช้วิธีรูปแบบปิด (Closed Form) ซึ่งจะเป็นการใช้พีชคณิตและตรีโกณมิติในการคำนวณหา ซึ่งสามารถอธิบายหลักการในการคำนวณหาได้ดังนี้

จากสมการที่ (2.33) และ (2.34) สามารถนำมายกกำลัง 2 และบวกรวมเข้าด้วยกันจะได้เป็น

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 \cos(\theta_2) \quad (2.36)$$

ทำการจัดรูปสมการใหม่ จะได้เป็น

$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (2.37)$$

ซึ่ง $\cos(\theta_2)$ จะมีค่าอยู่ในช่วง ± 1 ซึ่งหากทำการแทนค่าตัวแปรและคำนวณตามสมการที่ (2.37) แล้วทำให้เทอมของสมการในด้านขวามีค่าอยู่นอกช่วงของ ± 1 แสดงว่าค่าตำแหน่งพิกัด x, y ที่กำหนดนั้นมีค่าอยู่นอกพื้นที่การทำงานของแขนกล ทำให้ไม่สามารถหาคำตอบที่เป็นไปได้

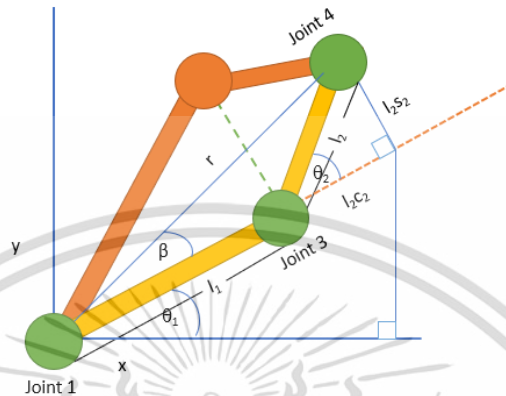
จากสมการที่ (2.37) สามารถหาคำตอบของสมการได้โดยการทำ $\cos^{-1}$ ซึ่งการทำด้วยวิธีนี้จะทำให้ได้คำตอบเดียว แต่สำหรับการเคลื่อนที่ของแขนกลนั้นจะมีการเคลื่อนที่ไปในทิศทางทั้ง 4 จตุภาค ทำให้คำตอบของจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับอาจมีได้หลายคำตอบ ดังนั้นจากความสัมพันธ์ของตรีโกณมิติ

$$\sin^2(\theta_2) + \cos^2(\theta_2) = 1$$

$$\sin(\theta_2) = \pm \sqrt{1 - \cos^2(\theta_2)} \quad (2.38)$$

ดังนั้นสามารถหาค่ามุมของข้อต่อที่ 3 หรือ θ_2 ได้จาก

$$\theta_2 = \text{Atan2}(\sin(\theta_2), \cos(\theta_2)) \quad (2.39)$$



รูปที่ 2.10 มุมมองด้านบนแสดงความสัมพันธ์ของเฟรมที่ {4} เทียบกับเฟรมอ้างอิง {0}

จากรูปที่ 2.10 แสดงให้เห็นว่าค่า θ_2 ที่ได้จากการคำนวณตามสมการที่ (2.39) นั้นสามารถมีคำตอบของสมการได้ 2 คำตอบ ซึ่งเกิดจากค่าของ $\cos(\theta_2)$ ที่ทำให้เกิด $\sin(\theta_2)$ จำนวน 2 ค่า ซึ่งเกิดจากค่าของ $\cos(\theta_2)$ ที่ทำให้เกิด $\sin(\theta_2)$ จำนวน 2 ค่า ซึ่งคำตอบที่ได้มานั้นจะเป็นการระบุท่าทางของแขนกลในกรณีที่แขนกลเคลื่อนที่ไปทางด้านบนหรือเคลื่อนที่ไปทางด้านลบ

ต่อไปจะเป็นการหาค่าของมุมข้อต่อที่ 1 หรือ θ_1 จากรูปที่ 2.10 แสดงความสัมพันธ์ของมุมได้เป็น

$$\tan(\beta) = \frac{l_2 \sin(\theta_2)}{l_1 + l_2 \cos(\theta_2)}$$

$$\beta = \text{Atan2}(l_2 \sin(\theta_2), l_1 + l_2 \cos(\theta_2)) \quad (2.40)$$

เมื่อ

$$\beta + \theta_1 = \text{Atan2}(y, x) \quad (2.41)$$

ดังนั้นมุมของข้อต่อที่ 1 จะมีค่าเท่ากับ

$$\theta_1 = \text{Atan2}(y, x) - \beta$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan2}(l_2 \sin(\theta_2), l_1 + l_2 \cos(\theta_2)) \quad (2.42)$$

การหาค่าของข้อต่อที่ 2 หรือ d_1 ซึ่งเป็นข้อต่อแบบเลื่อน ซึ่งสามารถคำนวณได้โดยการรวมความสูงของส่วนปลายของแขนกลเทียบกับฐาน จะคำตอบของสมการเป็น

$$d_1 = z + d_4 - d_5 - d_0 \quad (2.43)$$

จากสมการที่ (2.39), (2.42) และ (2.43) เมื่อทำการป้อนอินพุตซึ่งเป็นค่าตำแหน่งพิกัดของส่วนปลายของแขนกล และทำการแทนค่าค่าที่ลงในตัวแปรทั้งหมด จะสามารถคำนวณหาค่าของมุมของข้อต่อต่าง ๆ สำหรับจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับได้

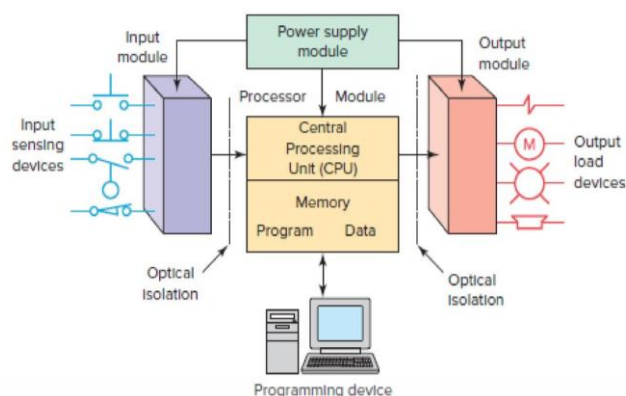
2.3 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC)

โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ หรือที่นิยมเรียกกันว่า PLC เป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักรหรือกระบวนการทำงานต่าง ๆ ซึ่ง PLC ถูกสร้างขึ้นมาเพื่อใช้ทดแทนวงจรรีเลย์แบบเก่าที่มีข้อจำกัดในเรื่องของการเดินสายไฟเพื่อเปลี่ยนแปลงเงื่อนไขของการควบคุมที่มีความยุ่งยาก และเสถียรภาพของการควบคุมจะลดลงเมื่อมีการใช้งานไปในระยะเวลานาน ดังนั้นจึงมีการนำเอา PLC เข้ามาทำงานทดแทน โดยที่ PLC วงจรภายในของ PLC จะมีไมโครโพรเซสเซอร์ที่ทำหน้าที่เป็นหน่วยประมวลผลและสั่งการ ทำให้มีความสามารถเปลี่ยนเงื่อนไขของการควบคุมใหม่ได้ โดยเพียงการปรับแต่งโปรแกรมการทำงานใหม่ หรือเขียนโปรแกรมใหม่ นอกจากนั้น PLC จะมีส่วนที่เชื่อมต่อกับอินพุตและเอาต์พุตใช้งานได้โดยตรง โดยอุปกรณ์ตรวจรู้ เซนเซอร์ หรือสวิตช์ต่าง ๆ จะต่อเข้ากับอินพุต ส่วนเอาต์พุตจะต่อออกไปควบคุมการทำงานของอุปกรณ์หรือเครื่องจักรที่เป็นเป้าหมาย และ PLC ยังสามารถใช้งานร่วมกับอุปกรณ์อื่นได้ ตัวอย่างเช่น เครื่องอ่านบาร์โค้ด (Barcode Reader), เครื่องพิมพ์ (Printer) หรือระบบ RFID เป็นต้น โดย PLC นั้นสามารถแบ่งการใช้งานออกเป็น 2 รูปแบบ ตามรูปแบบของการต่อใช้งาน ซึ่งก็คือการใช้งานแบบเดี่ยว (Standalone) และการใช้งานแบบต่อหลาย ๆ ตัวเข้าด้วยกันรวมเป็นระบบเครือข่าย (Network) ซึ่งจะเห็นได้ว่า PLC นั้นค่อนข้างที่จะมีความยืดหยุ่นในการใช้งานและถือเป็นอุปกรณ์ควบคุมที่สำคัญอย่างยิ่งในโรงงานอุตสาหกรรม สำหรับการควบคุมเครื่องจักรให้มีความทำงานที่เป็นแบบอัตโนมัติ เพื่อเพิ่มประสิทธิภาพของกระบวนการผลิตให้มากขึ้น

2.3.1 โครงสร้างและส่วนประกอบของ PLC

โครงสร้างของเครื่องควบคุมโปรแกรมเมเบิลลอจิกคอนโทรล หรือ PLC มีส่วนประกอบที่สำคัญ ซึ่งสามารถแบ่งออกได้เป็น 5 ส่วน ซึ่งประกอบด้วย หน่วยประมวลผลกลาง หน่วยความจำ หน่วยอินพุต/เอาต์พุต แหล่งจ่ายไฟ และเครื่องป้อนโปรแกรม

เอกสารนี้เป็นเอกสารลิขสิทธิ์ส่วนบุคคล ห้ามการเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 โครงสร้างสถาปัตยกรรมของ PLC

2.3.1.1 หน่วยประมวลผลกลาง (CPU : Center Processing Unit)

หน่วยประมวลผลกลาง หรือ CPU จะมีวงจรที่ประกอบด้วยวงจรลอจิกเกตหลาย ๆ ชนิดและไมโครโพรเซสเซอร์อยู่ภายใน ซึ่งมีทำหน้าที่ในการควบคุม ประมวลผลชุดคำสั่งข้อมูล หรือโปรแกรมตามที่ได้รับมาจากอุปกรณ์ส่วนอินพุต และเก็บข้อมูลทั้งหมดนั้นไว้ในหน่วยความจำ หลังจากนั้นเมื่อทำการประมวลผลเสร็จสิ้น ข้อมูลที่เป็นผลของการประมวลก็จะถูกส่งไปยังส่วนของอุปกรณ์เอาต์พุตตามเงื่อนไขของโปรแกรมที่ได้รับ โดยที่เวลาในการประมวลผลนั้นจะขึ้นอยู่กับเวลาสแกน (Scan Time) ซึ่งจะช้าหรือเร็วมีสาเหตุมาจากหลากหลายปัจจัย เช่น จำนวนข้อมูลของโปรแกรมที่ต้องประมวลผล หรือ ความยาวและลำดับขั้นของเงื่อนไขของโปรแกรมคำสั่ง เป็นต้น

2.3.1.2 หน่วยความจำ (Memory Unit)

หน่วยความจำ (Memory Unit) ทำหน้าที่ในการเก็บบันทึกโปรแกรมและข้อมูลต่าง ๆ ซึ่งในระหว่างที่ PLC มีการทำงานก็จะมีการดึงข้อมูลต่าง ๆ และโปรแกรมที่ได้มีการเก็บไว้ในหน่วยความจำออกมาเพื่อทำการประมวลผลการทำงาน โดยขนาดของหน่วยความจำจะถูกแบ่งออกเป็นบิตข้อมูล (Data Bit) โดยภายในหน่วยความจำ 1 บิตจะมีค่าสถานะทางลอจิกเป็น 0 หรือ 1 หน่วยความจำของ PLC จะประกอบไปด้วยหน่วยความจำ 2 ชนิด คือ หน่วยความจำชนิดชั่วคราว และหน่วยความจำถาวร ซึ่งสามารถอธิบายหน่วยความจำแต่ละชนิดได้ดังนี้

- 1) หน่วยความจำชั่วคราว (RAM : Random Access Memory) ทำหน้าที่เป็นหน่วยความจำในการเก็บโปรแกรมของผู้ใช้ และข้อมูลที่ใช้ในการทำงานของ PLC ซึ่งหน่วยความจำชั่วคราวนี้ถือว่าเป็นหน่วยความจำมาตรฐานของ PLC ที่ใช้งานในการเก็บโปรแกรมที่มีการแก้ไขบ่อย ๆ ซึ่งหน่วยความจำประเภทนี้จะมีแบตเตอรี่สำรองขนาดเล็ก ๆ ต่อเอาไว้เพื่อใช้เลี้ยงข้อมูลที่ถูกบันทึกไว้และป้องกันข้อมูลที่ถูกเก็บไว้ในหน่วยความจำนี้หายไป เมื่อเกิดไฟดับหรือไม่มีไฟเลี้ยงจ่ายให้กับ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) หน่วยความจำถาวร (ROM : Read Only Memory) ทำหน้าที่เป็นหน่วยความจำในการเก็บโปรแกรมระบบของ PLC และใช้เป็นพื้นที่สำรองข้อมูลเพื่อป้องกันกรณีที่โปรแกรม หรือข้อมูลในหน่วยความจำชั่วคราวหายไปซึ่งหน่วยความจำนี้ไม่จำเป็นต้องมีแบตเตอรี่สำรองข้อมูล แต่มีข้อจำกัดคือหน่วยความจำชนิดนี้สามารถทำการโปรแกรมใหม่ได้ แต่ไม่สามารถลบได้ อีกทั้งหากมีการเสียขารุดแล้วไม่สามารถซ่อมได้ และมีความเร็วในการเปิดข้อมูล (Time Access) ช้ากว่าหน่วยความจำแบบชั่วคราว จึงจำเป็นที่จะต้องมีการใช้หน่วยความจำทั้ง 2 ชนิดร่วมกัน เพื่อให้การทำงานมีความถูกต้องสมบูรณ์

2.3.1.3 หน่วยอินพุต/เอาต์พุต (Input/Output Unit)

หน่วยอินพุต/เอาต์พุต (Input/Output Unit) จะถูกต่อเข้ากับชุดควบคุมเพื่อรับและส่งค่าสถานะและค่าสัญญาณต่าง ๆ ไปยังอุปกรณ์ที่มาต่อใช้งาน ซึ่งสามารถแบ่งส่วนของการทำงานออกเป็น 2 ส่วนได้ดังนี้

- 1) หน่วยอินพุตของ PLC จะทำหน้าที่รับสัญญาณอินพุตเข้ามาแล้วแปลงสัญญาณเพื่อที่จะส่งเข้าไปภายใน PLC ซึ่งอุปกรณ์อินพุต (Input Device) ต่าง ๆ ที่จะนำมาต่อกับภาคอินพุตได้นั้น เช่น Relay, Limit Switch, Inverter, Encoder, Temperature Controller, Photoelectric Sensor เพื่อส่งไปยัง CPU เพื่อประมวลผลตามโปรแกรมคำสั่งของผู้ใช้ โดยหน่วยอินพุตจะมีหน้าที่การทำงาน ดังต่อไปนี้
 - แปลงระดับสัญญาณเข้าไปให้เป็นระดับสัญญาณดิจิทัลหรือแอนะล็อกที่เหมาะสมให้กับระบบการทำงานของหน่วยประมวลผลกลาง
 - แบ่งสัญญาณภายนอกและภายในออกจากกัน (Isolate) เพื่อที่จะป้องกันไม่ใหหน่วยประมวลผลกลางได้รับความเสียหาย
 - แก้ปัญหาการสั้นสละเพื่อนของหน้าสัมผัส
- 2) หน่วยเอาต์พุตของ PLC ทำหน้าที่ส่งสัญญาณออกไปขับโหลดชนิดต่าง ๆ ตามเงื่อนไขที่ได้มีการเขียนโปรแกรมเอาไว้ ซึ่งหน่วยเอาต์พุตทำหน้าที่รับข้อมูลจากหน่วยประมวลผลกลางแล้วส่งต่อข้อมูล ไปควบคุมอุปกรณ์ภายนอก เช่น ควบคุมหลอดไฟ มอเตอร์ และวาล์ว เป็นต้น

2.3.1.4 แหล่งจ่ายไฟ (Power Supply Unit)

แหล่งจ่ายไฟ (Power Supply) มีหน้าที่ในการจ่ายกำลังไฟฟ้าและรักษาระดับแรงดันให้กับส่วนต่าง ๆ ของ PLC ซึ่งแหล่งจ่ายไฟของ PLC นั้นจะแบ่งออกเป็น 2 ชุด ประกอบด้วยชุดแรกจะมีไว้สำหรับอุปกรณ์และวงจรภายใน และชุดที่สองจะมีไว้สำหรับการต่อวงจรของส่วนอินพุต

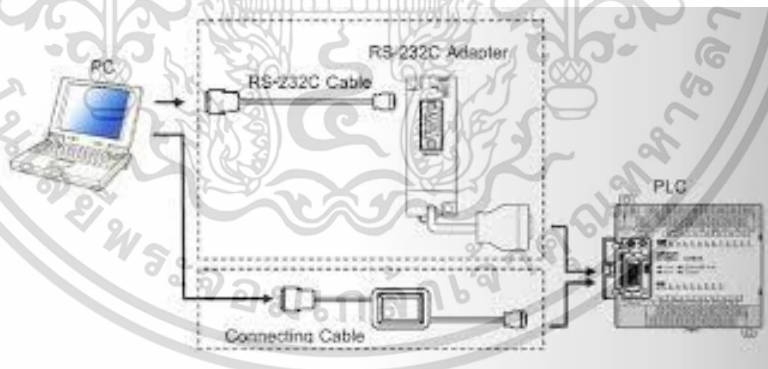
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเอาต์พุต โดยระดับแรงดันของ PLC ที่มีใช้งานนั้น จะมีอยู่ด้วยกัน 3 ระดับ คือ 24Vdc, 120Vac และ 240Vac ซึ่งจะมีหลักในการเลือกโดยเลือกใช้งานพิจารณาจากชนิด และจำนวนของอินพุต/เอาต์พุตที่ใช้

2.3.1.5 เครื่องป้อนโปรแกรม (Programming Device)

ในการสั่งงานให้ PLC มีการทำงาน ผู้ใช้จะต้องมีการป้อนโปรแกรมคำสั่งให้กับ PLC เพื่อให้ PLC ทำการประมวลผลโปรแกรมตามข้อมูลที่ได้รับให้เป็นไปตามเงื่อนไข และส่งสัญญาณไปขับเคลื่อนเอาต์พุตให้ทำงานตามที่ต้องการ ซึ่งในการป้อนโปรแกรมคำสั่งจะอาศัยการป้อนผ่านอุปกรณ์ที่เชื่อมต่อกับ PLC ผ่านการเชื่อมต่อในรูปแบบต่าง ๆ ไม่ว่าจะเป็นการเชื่อมต่อผ่านพอร์ตอนุกรม หรือการเชื่อมต่อผ่านเครือข่าย โดยในปฏิญานพนธ์ฉบับนี้จะมีอุปกรณ์ที่ใช้ในการเชื่อมต่อกับ PLC เพื่อใช้สำหรับการป้อนโปรแกรมคำสั่งอยู่ด้วยกัน 2 ประเภท คือ คอมพิวเตอร์ส่วนบุคคล และ Human Machine Interface (HMI)

1) คอมพิวเตอร์ส่วนบุคคล (Personal Computer) คือการใช้ซอฟต์แวร์คอมพิวเตอร์ ในที่นี้คือโปรแกรม CX-Programmer สำหรับการติดต่อระหว่าง PLC กับคอมพิวเตอร์ส่วนบุคคล ในการควบคุม ป้อนหรือแก้ไขโปรแกรมคำสั่ง กำหนดสัญญาณอินพุต/เอาต์พุต กำหนดลักษณะการตั้งค่า และกำหนดพื้นที่หน่วยความจำ ให้กับ PLC ให้สามารถทำตามเงื่อนไขที่มีการกำหนดได้



รูปที่ 2.12 ตัวอย่างการต่อคอมพิวเตอร์เข้ากับ PLC

2) Human Machine Interface (HMI) คือการควบคุมผ่านระบบอินเตอร์เฟซ หรือผ่านหน้าจอ ที่สามารถแสดงผล สั่งการ และจัดการข้อมูลต่าง ๆ ผ่านหน้าจอแสดงผล โดยใช้การติดต่อระหว่าง PLC กับ HMI เพื่อรับส่งค่าสัญญาณหรือข้อมูลต่าง ๆ ผ่านหน่วยประมวลผลกลางแล้วนำไปควบคุมอุปกรณ์เอาต์พุตต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ภาษาที่ใช้ในการเขียนโปรแกรมให้กับ PLC

การเขียนโปรแกรมคำสั่งให้ PLC ทำงานตามเงื่อนไขที่ต้องการ ตามมาตรฐาน IEC-31131-3 ได้มีการกำหนดภาษาที่ใช้ในการเขียนโปรแกรมของ PLC ไว้ทั้งหมด 5 ภาษาดังต่อไปนี้

2.3.2.1 Ladder Diagram (LD)

เป็นภาษาที่เขียนอยู่ในรูปแบบของกราฟิก ซึ่งมีพื้นฐานมาจากวงจรควบคุมแบบรีเลย์และวงจรไฟฟ้า ซึ่ง Ladder Diagram จะประกอบด้วย ราง (Rail) เพื่อใช้สำหรับเชื่อมต่อกับอุปกรณ์อินพุต ซึ่งเป็นสวิตช์หน้าสัมผัสและเป็นทางผ่านของกระแส และขดลวดหรือคอยล์ (Coil) เพื่อใช้สำหรับเชื่อมต่อกับอุปกรณ์ที่เป็นเอาต์พุต

2.3.2.2 Function Block Diagram (FBD)

เป็นภาษาที่มีรูปแบบการเขียน และฟังก์ชันการทำงานเช่นเดียวกับภาษา Ladder Diagram แต่จะมีการเชื่อมต่อกันเป็นโครงข่าย โดยวิธีการเขียนนั้นจะมีลักษณะเป็นฟังก์ชันบล็อกไดอะแกรมที่มีพื้นฐานมาจากลอจิกไดอะแกรม

2.3.2.3 Sequential Flow Chart (SFC)

เป็นภาษาที่มีโครงสร้างการทำงานเป็นแบบลำดับขั้น ซึ่งจะประกอบด้วย ส่วนประกอบ 2 ส่วน ได้แก่ ลำดับขั้น (Step) ของคำสั่งในการปฏิบัติงานในแต่ละขั้นตอน และเงื่อนไข (Condition) ที่กำหนดให้กระทำคำสั่งในแต่ละขั้นตอน ซึ่งสามารถกำหนดลักษณะการทำงานของโปรแกรมให้เป็นไปตามที่ต้องการได้

2.3.2.4 Instruction List (IL)

เป็นภาษาที่เขียนอยู่ในรูปของข้อความ ซึ่งจะมีลักษณะคล้ายกับภาษา Assembly หรือภาษาเครื่อง (Machine Code) โดยที่ภายในหนึ่งคำสั่งควบคุมจะประกอบด้วย 2 ส่วน ได้แก่ ส่วนปฏิบัติการ (Operator) และส่วนที่ถูกดำเนินการ (Operand)

2.3.2.5 Structure Text Language (ST)

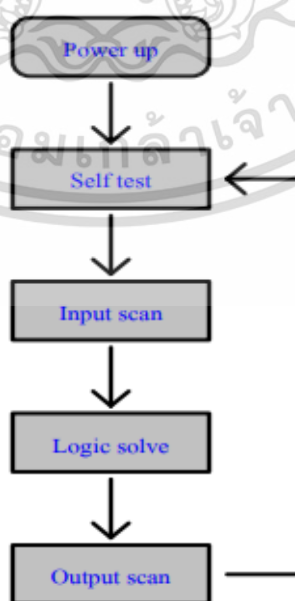
เป็นภาษาระดับสูงที่มีพื้นฐานกำหนดมาจากภาษา Pascal ซึ่งจะประกอบไปด้วยนิพจน์ และคำสั่ง ซึ่งคำสั่งโดยทั่วไปในการเขียนด้วย Structure Text จะอยู่ในรูปของคำสั่งที่เกี่ยวข้องกับการคำนวณค่าพารามิเตอร์ต่าง ๆ ด้วยสมการทางคณิตศาสตร์, การเลือกเงื่อนไขการทำงาน เช่น IF....THEN....ELSE เป็นต้น และคำสั่งในรูปของการทำงานแบบวนซ้ำ เช่น FOR, WHILE เป็นต้น

ซึ่งในการเขียนโปรแกรมคำสั่ง เพื่อควบคุมการทำงานของแขนกล SCARA Robot 4DOF ผู้จัดทำได้มีการเขียนโปรแกรมคำสั่งให้การทำงานที่เป็นลำดับขั้นตอน มีการคำนวณ

ค่าพารามิเตอร์ และมีการทำงานตามเงื่อนไข ซึ่งจะประยุกต์ใช้การเขียนในรูปแบบของภาษา Ladder Diagram, Function Block Diagram และ Structure Text เป็นหลัก

2.3.3 การทำงานของ PLC

การทำงานของ PLC ส่วนใหญ่แล้วจะมีลำดับขั้นของการทำงานพื้นฐานอยู่ 4 ขั้นตอน และจะมีการทำงานวนซ้ำ ๆ กันหลายครั้งภายในเวลาหนึ่งวินาที การทำงานของ PLC จะเริ่มต้นเมื่อมีการจ่ายแรงดันให้กับ PLC ซึ่ง PLC จะทำการตรวจสอบสถานะการทำงานของอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์เพื่อที่จะทำการตรวจหาข้อผิดพลาดต่าง ๆ หากไม่มีปัญหาใด ๆ PLC จะนำเอาสัญญาณอินพุต ที่เป็นข้อมูลต่าง ๆ จากอุปกรณ์ในหน่วยอินพุต เข้ามาเก็บไว้ในหน่วยความจำ ซึ่งขั้นตอนนี้จะถูกเรียกว่า สแกนอินพุต (Input Scan) หลังจากนั้น PLC จะทำการประมวลผลตามโปรแกรมคำสั่งที่ผู้ใช้มีการเขียนเอาไว้ โดยใช้ข้อมูลจากหน่วยความจำประกอบ ขั้นตอนการประมวลผลนี้เรียกว่า สแกนลอจิก (Logic Scan) ในขณะที่ PLC ทำการประมวลผลตามเงื่อนไขของโปรแกรมคำสั่งนั้น ค่าเอาต์พุตของ PLC ก็เปลี่ยนแปลงไปตามเงื่อนไขต่าง ๆ ของโปรแกรม โดยที่การเปลี่ยนแปลงของค่าสถานะทางเอาต์พุตนี้จะถูกเก็บไว้ในหน่วยความจำเช่นกัน และจะเก็บไว้ที่หน่วยความจำชั่วคราวเพียงเท่านั้น เมื่อ PLC ทำการสแกนลอจิกเสร็จสิ้นแล้ว ข้อมูลเอาต์พุตในหน่วยความจำชั่วคราวนี้จะถูกส่งไปที่หน่วยเอาต์พุต เพื่อให้อุปกรณ์หรือโหลดที่ต่ออยู่นั้นมีสถานะทำงานหรือไม่ทำงานตามเงื่อนไขและผลลัพธ์ที่ได้จากการประมวลผล ซึ่งเรียกขั้นตอนนี้ว่า สแกนเอาต์พุต (Output Scan) ซึ่งเมื่อสิ้นสุดการสแกนเอาต์พุต PLC ก็จะกลับไปเริ่มต้นการทำงานใหม่ ซึ่งกระบวนการดังกล่าวนี้จะใช้เวลาอยู่ที่ 5 – 10 ms หรือเร็วกว่าขึ้นอยู่กับความสามารถและความเร็วในการทำงานของหน่วยประมวลผลกลางของ PLC ด้วย



รูปที่ 2.13 วนรอบลำดับการทำงานของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ยืมได้ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 การสแกนอินพุตและเอาต์พุต

เมื่อถึงขั้นตอนของการ สแกนอินพุต อุปกรณ์อินพุตต่าง ๆ ที่ต่อเข้ากับ PLC นั้นถูกสแกน ค่าข้อมูลหรือค่าสถานะต่าง ๆ จะถูกนำไปเก็บไว้ในหน่วยความจำ ในทางกลับกันเมื่อถึงขั้นตอนของการสแกนเอาต์พุต อุปกรณ์เอาต์พุตที่ต่อกับ PLC ที่ถูกสแกนจะทำการคัดลอกข้อมูลจากในหน่วยความจำ ส่งออกไปให้กับอุปกรณ์เอาต์พุต ซึ่งการสแกนลอจิกหรือในการประมวลผลโปรแกรม PLC จะใช้ค่าหรือข้อมูลที่อยู่ภายในหน่วยความจำเท่านั้น โดยที่จะไม่สนใจค่าหรือข้อมูลจริงของอินพุตและเอาต์พุตในขณะนั้นเลย โดยที่ต้องเป็นเช่นนั้นก็เพราะว่า ถ้าค่าข้อมูลที่ใช้ในการประมวลผลทั้งทางด้านอินพุตและเอาต์พุตมีการเปลี่ยนแปลงไปตลอดเวลา จะทำให้ PLC มีการทำงานที่ช้ามาก เพราะแทนที่ PLC จะประมวลผลโปรแกรมในแต่ละคำสั่งให้จบทั้งโปรแกรมไป แต่กลับที่จะต้องมาคัดลอกข้อมูลใหม่ที่มีการเปลี่ยนแปลงทุกครั้ง ซึ่งทำให้การประมวลผลโปรแกรมไม่สามารถทำได้โดยง่าย

2.3.5 ข้อมูลในหน่วยความจำของ PLC

PLC จะมีการเก็บข้อมูลของค่าสถานะ ค่าจากการวัดของอุปกรณ์ตรวจรู้หรือเซนเซอร์ หรือค่าสัญญาณต่าง ๆ จากหน่วยอินพุตและหน่วยเอาต์พุตเอาไว้ในหน่วยความจำของ PLC ซึ่งข้อมูลที่ถูกจัดกันั้นจะถูกแบ่งออกเป็น 3 ประเภท โดยอ้างอิงตามขนาดของข้อมูล ซึ่งได้แก่ บิต (Bit), ดิจิต (Digit), ไบต์ (Byte) และเวิร์ด (Word) ซึ่งสามารถอธิบายความสัมพันธ์ได้ดังนี้

1 เวิร์ด	=	2 ไบต์
1 ไบต์	=	2 ดิจิต
1 ดิจิต	=	4 บิต

2.4 ระบบเลขฐาน

เนื่องจากในปริญญานิพนธ์ฉบับนี้ใช้ PLC เป็นเครื่องควบคุมหลัก ซึ่งการประมวลผลข้อมูลจะเป็นการประมวลผลที่อยู่ในรูปของข้อมูลแบบดิจิตอล เช่นเดียวกับคอมพิวเตอร์ หรือเครื่องควบคุมอื่น ๆ ที่มีใช้งานอยู่ในปัจจุบัน ดังนั้นระบบเลขฐานจึงเป็นพื้นฐานที่สำคัญในการแปลงข้อมูลต่าง ๆ ให้สามารถเข้าถึงร่วมกันได้ ซึ่งระบบเลขฐานที่ถูกนำมาใช้งานบน PLC ในการเขียนโปรแกรมคำสั่งควบคุม หรือใช้งานในด้านต่าง ๆ นั้น จะมีอยู่ด้วยกัน 4 ระบบ ได้แก่

2.4.1 ระบบเลขฐาน 2 (Binary Number System)

ข้อมูลสำหรับระบบเลขฐาน 2 จะประกอบไปด้วยเลข 2 ตัว คือ 0 กับ 1 ซึ่งเป็นเลขฐานที่ใช้สำหรับการประมวลผลในภาษาคอมพิวเตอร์ เพราะจากการที่อุปกรณ์ทางไฟฟ้าหรือวงจรดิจิตอลนั้นมีสถานะการทำงานอยู่ด้วยกัน 2 สถานะคือ เปิด กับ ปิด ซึ่งเทียบได้กับข้อมูล 0 กับ 1 ของเลขฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ตามลำดับ ซึ่งการใช้เลขฐาน 10 อาจไม่เป็นที่นิยมนักในสำหรับใช้งานกับคอมพิวเตอร์ และอาจเกิดปัญหาอื่น ๆ ขึ้นได้ หรือแม้แต่ในอุปกรณ์ทางไฟฟ้าก็อาจจะต้องทำให้มีการแบ่งสถานะออกเป็น 10 สถานะ ซึ่งทำให้เกิดความยุ่งยาก ซึ่งในการเก็บข้อมูลด้วยระบบเลขฐาน 2 นี้ จะมีการจัดเก็บโดยการเก็บเป็นกลุ่มเลขฐาน 2 จำนวนหลาย ๆ บิต นำมาต่อกัน โดยที่จะมีความยาวตามขนาดของข้อมูลที่ต้องการจัดเก็บ

2.4.2 ระบบเลขฐาน 8 (Octal Number System)

เป็นระบบเลขฐานที่ประกอบด้วยตัวเลขจำนวน 8 ตัว ประกอบด้วยเลข 0, 1, 2, 3, 4, 5, 6, 7 ซึ่งระบบเลขฐาน 8 นั้น เป็นระบบเลขฐานที่ถูกนำไปใช้แทนระบบเลขฐาน 2 สำหรับในการเก็บข้อมูลที่มีขนาดใหญ่ขึ้น

2.4.3 ระบบเลขฐาน 10 (Decimal Number System)

เป็นระบบเลขฐานที่ถูกนำมาใช้เป็นมาตรฐานที่สามารถพบเห็นได้ทั่วไปในชีวิตประจำวัน ซึ่งประกอบด้วยเลขจำนวน 10 ตัว ประกอบด้วยเลข 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ซึ่งระบบเลขฐาน 10 นี้เป็นระบบเลขฐานที่สามารถนำไปคำนวณค่าทางคณิตศาสตร์ได้ง่ายกว่าระบบเลขฐานทั้งหมด

2.4.4 ระบบเลขฐาน 16 (Hexadecimal Number System)

เป็นระบบเลขฐานที่ถูกนำมาใช้แสดงแทนระบบเลขฐาน 2 ในการเก็บข้อมูลที่มีจำนวนมาก สำหรับการประมวลผลทางคอมพิวเตอร์ เพราะถ้าใช้ระบบเลขฐาน 2 ในการเก็บข้อมูลที่มีจำนวนมากๆ แล้วนั้น จะทำให้ต้องเขียนแทนด้วยเลขฐาน 2 ที่มีขนาดหลายบิต แต่หากเขียนแทนด้วยเลขฐาน 16 จะเป็นการลดรูปทำให้เหลือเพียงจำนวนไม่กี่หลัก ซึ่งใน PLC ก็ได้มีการใช้ระบบเลขฐาน 16 เป็นหลักในการแสดงค่าของข้อมูลที่ถูกเก็บไว้ในหน่วยความจำ เพื่อแทนเลขฐาน 2 ที่ใช้ในการประมวลผลด้วยเช่นกัน ระบบเลขฐาน 16 จะประกอบไปด้วยตัวเลขจำนวน 10 เลข และตัวอักษรอีก 6 ตัว ซึ่งประกอบด้วย 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 และ A, B, C, D, E, F

ตารางที่ 2.3 ตารางตัวอย่างความสัมพันธ์ระหว่างเลขฐานแต่ละระบบ

เลขฐาน 2	เลขฐาน 8	เลขฐาน 10	เลขฐาน 16
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

2.5 OLE for Process Control (OPC Server)

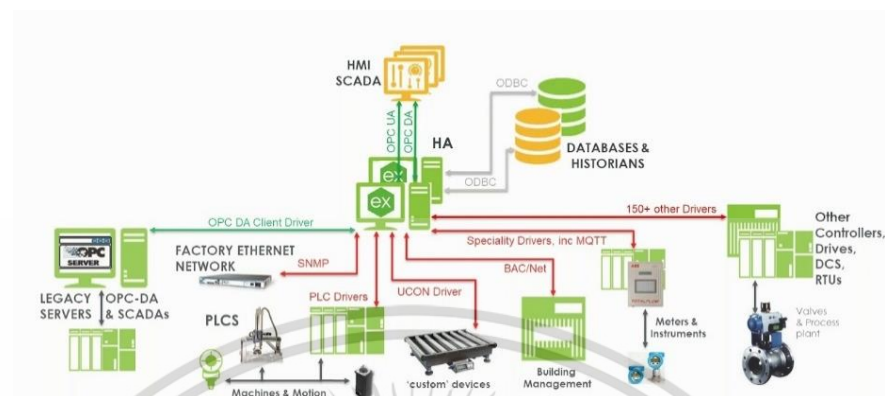
OLE for Process Control หรือ OPC Server ทำหน้าที่ในการเป็นตัวกลางในการติดต่อสื่อสารข้อมูลระหว่างอุปกรณ์ทางด้านอุตสาหกรรม เช่น เครื่องจักร HMI หรือ PLC กับระบบของผู้ใช้ ซึ่งใช้มาตรฐานการสื่อสารในรูปแบบของ server และ client โดย OPC Server จะทำการแปลงโปรโตคอลการเชื่อมต่อของอุปกรณ์ต่าง ๆ ที่เชื่อมต่ออยู่ให้เป็นโปรโตคอลเดียวกัน คือ OPC โปรโตคอล เพื่อให้สามารถรับ-ส่ง และใช้ข้อมูลต่าง ๆ ร่วมกันได้

2.5.1 KEPServerEX6

ในปริญญานิพนธ์ฉบับนี้ จะมีการใช้ OPC Server เชื่อมต่อระหว่าง PLC กับคอมพิวเตอร์ สำหรับการรับส่งข้อมูลตำแหน่งส่วนปลายของแขนกลที่อยู่ในรูปของ G-Code บนฐานข้อมูลซึ่งอยู่บนตารางของโปรแกรม Microsoft Excel กับหน่วยความจำของ PLC เพื่อใช้ในการป้อนโปรแกรมคำสั่งสำหรับการควบคุมแขนกล SCARA Robot 4DOF ให้มีการเคลื่อนที่ไปในหลาย ๆ

ตำแหน่ง โดยเลือกใช้ KEPServerEX6 เป็น OPC Server ซึ่งในการใช้งานจะต้องทราบถึงโครงสร้างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของอุปกรณ์ที่เชื่อมต่ออยู่ และทำการสร้าง tag เชื่อมต่อกับที่อยู่ของหน่วยความจำ PLC ที่จะทำให้การเชื่อมต่อเพื่อรับ-ส่งข้อมูล ลงใน server เพื่อให้สามารถทำการควบคุม หรือมอนิเตอร์ข้อมูลได้



รูปที่ 2.14 การประยุกต์ใช้ KEPServerEX6 กับงานด้านต่าง ๆ

2.5.2 Dynamic Data Exchange (DDE)

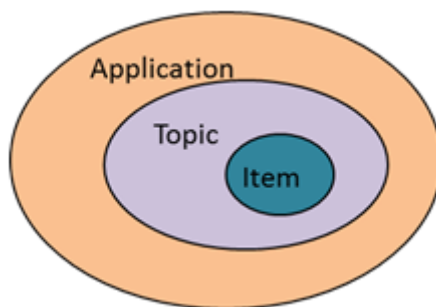
ในการดึงข้อมูลจากฐานข้อมูลบนโปรแกรม Microsoft Excel นั้นจะไม่สามารถเชื่อมต่อกับ OPC Server โดยตรงได้ ซึ่งจะต้องมีเครื่องมือที่เป็นตัวกลางในเชื่อมโยงข้อมูลจากบนโปรแกรม Microsoft Excel ไปยัง OPC Server เพื่อเก็บข้อมูลไปที่หน่วยจำ PLC หรือบนโปรแกรม CX-Programmer แบบเรียลไทม์ ซึ่ง Dynamic Data Exchange หรือ DDE นั้นเป็นมาตรฐานในการติดต่อสื่อสารระหว่างโปรแกรมที่ทำงานบนระบบปฏิบัติการ Windows ซึ่งการแลกเปลี่ยนข้อมูลจะประกอบด้วยส่วนประกอบที่สำคัญ 3 ส่วน ได้แก่ โปรแกรมที่ติดต่อด้วย, กลุ่มที่ข้อมูลอยู่ และข้อมูลที่ต้องการ และสามารถแบ่งชนิดของการเชื่อมต่อได้ 2 วิธี คือ

2.5.2.1 แบบ Hot Link

เป็นการเชื่อมต่อผ่าน Dynamic Data Exchange (DDE) เพื่อใช้สำหรับอ่านข้อมูลเพียงอย่างเดียว (Read only) ซึ่งจะมีการเชื่อมต่ออยู่ตลอดเวลาตลอดการใช้งาน

2.5.2.2 แบบ Cold Link

เป็นการเชื่อมต่อผ่าน Dynamic Data Exchange (DDE) ที่สามารถอ่านและเขียนข้อมูลลงไปได้ ซึ่งจะถูกกำหนดเงื่อนไขหรือเวลาในการเชื่อมต่อเอาไว้

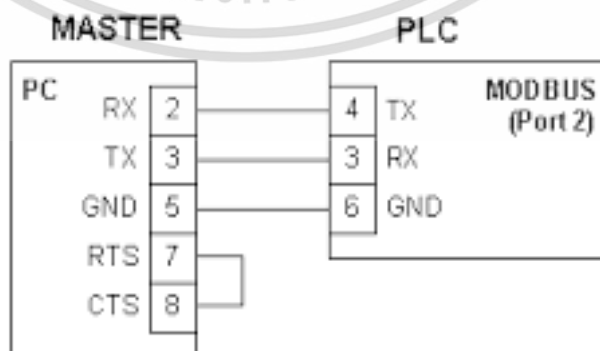


รูปที่ 2.15 ส่วนประกอบของระบบ Dynamic Data Exchange (DDE)

2.5.3 การติดต่อสื่อสารผ่านพอร์ตอนุกรม RS232

สำหรับการเชื่อมต่อคอมพิวเตอร์ เพื่อใช้ในการรับ-ส่งข้อมูลกับ PLC ผ่าน OPC Server นั้น จะใช้การติดต่อสื่อสารผ่านพอร์ตอนุกรมโดยใช้ RS232 ซึ่ง RS232 คือมาตรฐานการรับ-ส่งข้อมูลดิจิทัลแบบอนุกรม (Serial Communication) ที่มีการรับส่งข้อมูลกันเป็นแบบอะซิงโครนัส 2 ทิศทาง หรือ Full duplex กล่าวคืออุปกรณ์ที่ติดต่อสื่อสารกันอยู่นั้น สามารถรับและส่งข้อมูลได้พร้อม ๆ กันในเวลาเดียวกัน ซึ่งการรับ-ส่งข้อมูล สำหรับสถานะลอจิก 0 นั้นจะมีระดับสัญญาณตั้งแต่ 3 โวลต์ ไปจนถึง 15 โวลต์ และสำหรับสถานะลอจิก 1 จะมีระดับแรงดันที่ -3 โวลต์ ไปจนถึง -15 โวลต์ ซึ่งการนำไปใช้งานกับวงจรดิจิทัลนั้นจึงต้องมีอุปกรณ์ในการแปลงระดับแรงดันให้เหลือเพียง 0-5 โวลต์ สำหรับภาคส่งข้อมูล และสำหรับภาครับข้อมูลก็ต้องเปลี่ยนระดับแรงดันให้สูงกว่า +3 หรือต่ำกว่า -3 ซึ่งในการรับ-ส่งข้อมูลจะประกอบไปด้วยสายเชื่อมต่อทั้งหมด 3 เส้น ดังรูปที่ 2.16 ซึ่งสามารถอธิบายได้ ดังนี้

- 1) Tx (Transmit Data) คือ สายที่ใช้สำหรับการส่งข้อมูล
- 2) Rx (Receive Data) คือ สายที่ใช้สำหรับรับข้อมูล
- 3) GND (Signal Ground) คือ สายกราวด์ หรือสายอ้างอิงในการเทียบศักย์แรงดันไฟฟ้า ที่ 0 โวลต์



รูปที่ 2.16 ตัวอย่างการเชื่อมต่อคอมพิวเตอร์เข้ากับ PLC ผ่านพอร์ตอนุกรม RS232

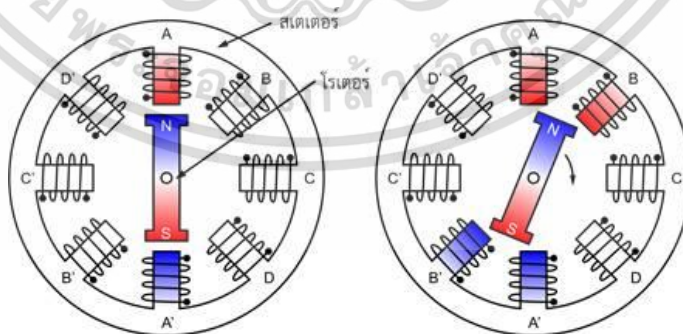
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 มอเตอร์แบบสเต็ป

มอเตอร์แบบสเต็ป หรือที่เรียกว่าสเต็ปมิ่งมอเตอร์ (Stepping Motor) คือมอเตอร์ไฟฟ้าชนิดหนึ่ง ที่ถูกนำมาประยุกต์ใช้สำหรับการควบคุมที่มีการกำหนดตำแหน่ง และทิศทางให้กับกลไกการทำงาน of เครื่องจักรที่ต้องการความแม่นยำและความแน่นอนที่สูง ซึ่งการหมุนของมอเตอร์ชนิดนี้ จะมีความแตกต่างจากมอเตอร์ชนิดอื่น ๆ นั่นก็คือการหมุนของมอเตอร์แบบสเต็ปนั้นจะมีการหมุนที่แบ่งออกเป็นทีละขั้น หรือสเต็ป โดยการหมุนไปแต่ละสเต็ปนั้นจะมีระยะที่เท่า ๆ กัน เช่น หมุนไปสเต็ปละ 0.9 องศา, 1.8 องศา, 3.6 องศา เป็นต้น ซึ่งหลักการทำงานของมอเตอร์ชนิดนี้จะใช้สัญญาณพัลส์ในการขับเคลื่อนมอเตอร์ให้มีการหมุน มอเตอร์ชนิดนี้นั้นสามารถหมุนได้รอบแกน 360 องศา โดยที่มุมในการหมุนและความเร็วในการหมุนนั้นจะมีค่าตามสัดส่วนที่สัมพันธ์กับสัญญาณพัลส์ ที่จ่ายให้กับมอเตอร์

2.6.1 โครงสร้างและหลักการทำงานของมอเตอร์แบบสเต็ป

โครงสร้างภายในของมอเตอร์แบบสเต็ป จะแสดงดังรูปที่ 2.X ซึ่งประกอบด้วยสเตเตอร์ (Stator) ที่มีขดลวดพันอยู่ และส่วนที่เป็นแกนหมุนหรือโรเตอร์ (Rotor) เป็นแม่เหล็กถาวร โดยหลักการทำงานของมอเตอร์แบบสเต็ปคือ เมื่อจ่ายกระแสให้ไหลผ่านขดลวดสเตเตอร์จะทำให้เกิดการเหนี่ยวนำของสนามแม่เหล็กขึ้น แรงดึงดูดที่เกิดขึ้นจากสนามแม่เหล็กนี้จะทำให้เกิดการหมุนของโรเตอร์ ซึ่งกระแสที่จ่ายให้กับมอเตอร์แบบสเต็ปนั้นจะอยู่ในรูปแบบของสัญญาณพัลส์ ทำให้ต้องใช้วงจรอิเล็กทรอนิกส์หรือวงจรขับมอเตอร์ในการกำเนิดสัญญาณก่อนป้อนให้กับมอเตอร์ โดยเมื่อทำการจ่ายกระแสไปยังขดลวดแต่ละขดอย่างต่อเนื่องจะมีผลทำให้มอเตอร์แบบสเต็ปนั้นมีการหมุนอย่างต่อเนื่อง และหากต้องการให้มอเตอร์มีการหมุนไปตามจำนวนของสเต็ปที่กำหนดนั้น สามารถทำได้โดยการจ่ายกระแสที่เป็นสัญญาณพัลส์ผ่านขดลวดตามจำนวนสเต็ปที่ต้องการ



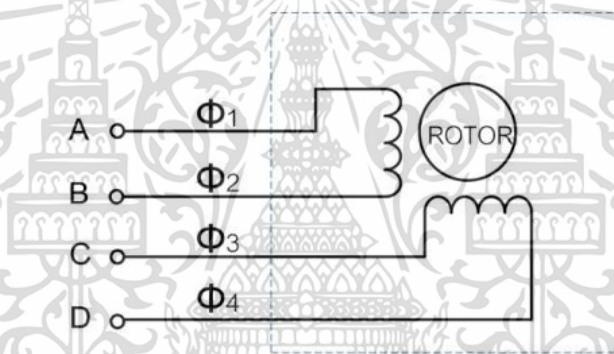
รูปที่ 2.17 มอเตอร์แบบสเต็ปขณะหยุดนิ่ง และขณะหมุนไปได้ครึ่งสเต็ป

2.6.2 ชนิดของมอเตอร์แบบสเต็ป

มอเตอร์แบบสเต็ปสามารถแบ่งออกได้เป็น 2 ชนิด คือ ชนิดยูนิโพลาร์ (Unipolar) และ ชนิดไบโพลาร์ (Bipolar) ซึ่งทั้งสองชนิดเป็นมอเตอร์แบบ 2 เฟส โดยแต่ละแบบจะมีโครงสร้างและจุดเชื่อมต่อที่แตกต่างกัน โดยในปริิณญาณิพนธ์ฉบับนี้จะขออธิบายเพียงเฉพาะมอเตอร์แบบสเต็ปชนิดไบโพลาร์ ซึ่งมีรายละเอียด ดังนี้

2.6.2.1 มอเตอร์แบบสเต็ปชนิดไบโพลาร์

มอเตอร์แบบสเต็ปชนิดไบโพลาร์จะมีโครงสร้างที่มีขดลวดภายในจำนวน 2 ชุด แบ่งออกเป็นชุด A และชุด B และไม่มีจุดต่อสำหรับต่อแรงดันไฟบวก การหมุนของมอเตอร์นั้นจะ ใช้การจ่ายกระแสกระตุ้นผ่านเฟสแต่ละเฟสของขดลวดทั้ง 2 ชุด ทำให้จึงมีสายต่อออกมาใช้งานเพียงจำนวน 4 เส้น โดยโครงสร้างของมอเตอร์แบบสเต็ปชนิดไบโพลาร์ แสดงดังรูปที่ 2.X



รูปที่ 2.18 โครงสร้างของมอเตอร์แบบสเต็ปชนิดไบโพลาร์

2.6.3 การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป

การควบคุมและสั่งการให้มอเตอร์แบบสเต็ปมีการหมุนหรือเคลื่อนที่ไปที่ละสเต็ป ๆ ได้นั้น สามารถทำได้โดยการจ่ายกระแสเข้าไปที่ขดลวดสเตเตอร์อย่างทีกล่าวไปข้างต้น ซึ่งการจ่ายกระแสนั้นจะทำในลักษณะที่เป็นลำดับและมีการวนรอบการทำตามเดิม โดยวิธีการจ่ายกระแสนั้นสามารถแบ่งออกเป็น 3 แบบ ได้แก่ แบบฟูลสเต็ป 1 เฟส, แบบฟูลสเต็ป 2 เฟส และแบบครึ่งสเต็ป

2.6.3.1 แบบฟูลสเต็ป 1 เฟส (Full Step 1 Phase)

การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป แบบฟูลสเต็ป 1 เฟส จะเป็นการจ่ายกระแสเพื่อขับมอเตอร์ไปที่ละขด โดยเรียงลำดับจากขดลวดที่ 1 2 3 4 หรือ 4 3 2 1 ตามทิศทางการหมุน ซึ่งลำดับการกระตุ้นสามารถดูได้จากตารางที่ 2.4

ตารางที่ 2.4 ลำดับการจ่ายกระแสกระตุ้นแบบฟูลสเต็ป 1 เฟส

สเต็ปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน
5	ทำงาน	-	-	-
6	-	ทำงาน	-	-

2.6.3.2 แบบฟูลสเต็ป 2 เฟส (Full Step 2 Phase)

การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป แบบฟูลสเต็ป 2 เฟส นั้นจะ เป็นการกระตุ้นไปที่ละขดลวดเช่นเดียวกับแบบ 1 เฟส แต่จะแจกจ่ายกันตรงที่แบบ 2 เฟสนั้นจะ กระตุ้นขดลวดที่ละ 2 ขด ที่อยู่ลำดับใกล้เคียงกัน เช่น 1 และ 2 2 และ 3 3 และ 4 4 และ 1 หรือ 1 และ 4 4 และ 3 3 และ 2 2 และ 1 ตามทิศทางการหมุนของมอเตอร์ ลำดับการกระตุ้นสามารถดูได้จากตารางที่ 2.5

ตารางที่ 2.5 ลำดับการจ่ายกระแสกระตุ้นแบบฟูลสเต็ป 2 เฟส

สเต็ปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน
5	ทำงาน	ทำงาน	-	-
6	-	ทำงาน	-	-

2.6.3.3 แบบครึ่งสเต็ป (Half Step)

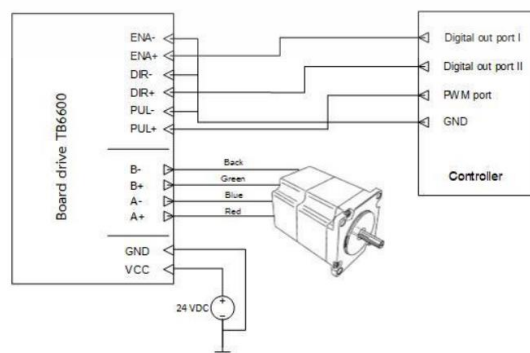
การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป แบบนี้นั้นจะเป็นการจ่าย กระแสไปที่ขดลวดที่ 1 และขดลวดที่ 2 สลับกันไป โดยการกระตุ้นแบบนี้จะเป็นการเพิ่มจำนวนรอบ ของการกระตุ้นให้มากขึ้น ทำให้มอเตอร์แบบสเต็ปหมุนเร็วขึ้น และมีความแม่นยำที่มากขึ้น ซึ่งลำดับ การกระตุ้นดูได้จากตารางที่ 2.6

ตารางที่ 2.6 ลำดับการจ่ายกระแสกระตุ้นแบบครึ่งสเต็ป

สเต็ปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	-
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน
9	ทำงาน	-	-	ทำงาน
10	ทำงาน	ทำงาน	-	-

2.6.4 วงจรขับมอเตอร์

ในการขับมอเตอร์โดยปกติจะป้อนแรงดันไฟตรงให้โดยตรง ทำให้มอเตอร์ทำงานเต็มกำลัง ซึ่งอาจมีความเร็วมากเกินไป ส่งผลให้ไม่เกิดความไม่เสถียรในการทำงาน และอาจทำให้เกิดความยากและความผิดพลาดในการควบคุมองศาการหมุนของมอเตอร์ ดังนั้นการปรับความเร็วของมอเตอร์จึงใช้วิธีลดแรงดันไฟฟ้าที่ป้อนให้กับมอเตอร์ วิธีที่นิยมคือ การป้อนพัลส์ไปขับมอเตอร์แทน แล้วปรับความกว้างพัลส์ช่วงบวก เพื่อให้ได้ค่าแรงดันเฉลี่ยตามต้องการ วิธีการนี้เรียกว่า พัลส์วidthมอดูเลเตอร์ (PWM) โดยการควบคุมอาศัยหลักการทำงานของอุปกรณ์สารกึ่งตัวนำต่างๆ เช่น ทรานซิสเตอร์ ไอซี หรือมอสเฟต เป็นต้น ซึ่งวงจรขับมอเตอร์ที่ใช้ในปริญญาานิพนธ์ฉบับนี้ คือ Microstep driver เป็นไดเวอร์สเต็ปเปอร์มอเตอร์แบบสองเฟส สร้างขึ้นจาก IC TB6600HG ทำหน้าที่ในการควบคุมความเร็วและทิศทางของมอเตอร์ ที่สามารถจ่ายกระแสได้ 0-4 แอมป์ และจ่ายแรงดันไฟกระแสตรงได้ 9-24 โวลต์



รูปที่ 2.19 การต่อใช้งาน Microstep Driver กับเครื่องควบคุม

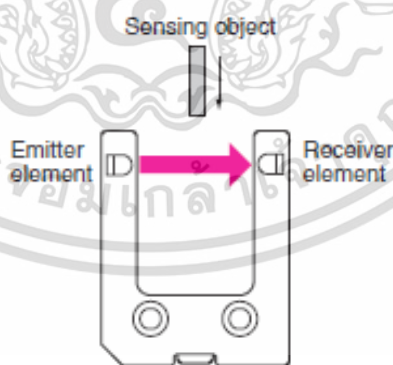
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ขออนุญาต การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาต ถือว่าผิดกฎหมาย และผู้จัดทำขอสงวนสิทธิ์ในการแก้ไขเอกสารฉบับนี้โดยไม่ต้องแจ้งให้ทราบล่วงหน้า

2.7 โฟโตอิเล็กทริกเซนเซอร์

โฟโตอิเล็กทริกเซนเซอร์ คือเซนเซอร์ที่ใช้สำหรับการตรวจจับการเคลื่อนไหวของวัตถุ หรือตรวจสอบขนาดรูปร่างของวัตถุ โดยที่เซนเซอร์ชนิดนี้ทำงานโดยที่ไม่ต้องมีการสัมผัสตัววัตถุ แต่จะอาศัยหลักการในการส่งและรับแสงระหว่างภาคส่งแสง (Emitter) ภาคตัวรับแสง (Receiver) ลักษณะของการตรวจจับนั้นเกิดจากการที่ลำแสงจากตัวส่งแสงส่งไปสะท้อนกับวัตถุหรือถูกขวางกั้นด้วยวัตถุ ส่งผลให้ตัวรับแสงนั้นมีสภาวะการทำงานที่เปลี่ยนไป และทำให้เกิดสัญญาณเอาต์พุตเพื่อนำไปใช้งานต่อไป

2.7.1 หลักการโฟโตอิเล็กทริกเซนเซอร์ที่ใช้ในปริภูมิพนธ์

โฟโตอิเล็กทริกเซนเซอร์ที่นำมาใช้ในปริภูมิพนธ์ฉบับนี้คือ PM-K44 เป็นโฟโตอิเล็กทริกเซนเซอร์ที่มีขนาดเล็ก มีวงจรรขยายสัญญาณในตัว ลักษณะโครงสร้างเป็นรูปตัว U หรือก้ามปู ซึ่งมีวงจรควบคุมที่เป็นทั้งภาคส่งแสง (Emitter) และภาครับแสง (Receiver) หลักการทำงานของเซนเซอร์นั้นจะอาศัยหลักการเดียวกันกับโฟโตอิเล็กทริกเซนเซอร์ชนิดลำแสงผ่านตลอด (Through Beam Photoelectric Sensor) ซึ่งมีหลักการคือให้ภาคส่งแสง และภาครับแสงนั้นอยู่ตรงข้ามกัน ในขณะที่มีสภาวะปกติภาครับแสง จะสามารถรับลำแสงที่ส่งออกมาจากตัวส่งได้ตลอดเวลา ซึ่งเมื่อมีวัตถุหรือชิ้นงานผ่านหน้าเซนเซอร์ และไปขวางลำแสงที่ส่งมาจากภาคส่งแสง ทำให้ภาครับแสงไม่สามารถรับสัญญาณได้ ก็จะเป็นการทำให้วงจรภายในของเซนเซอร์นั้นตรวจรู้ได้ว่ามีวัตถุหรือชิ้นงานขวางอยู่ และจะทำให้สถานะเอาต์พุตของภาครับแสงเปลี่ยนไป ซึ่งจะเรียกลักษณะการทำงานแบบนี้ว่า Dark On หรือ Dark Operate



รูปที่ 2.20 การทำงานของโฟโตอิเล็กทริกเซนเซอร์แบบ U-type

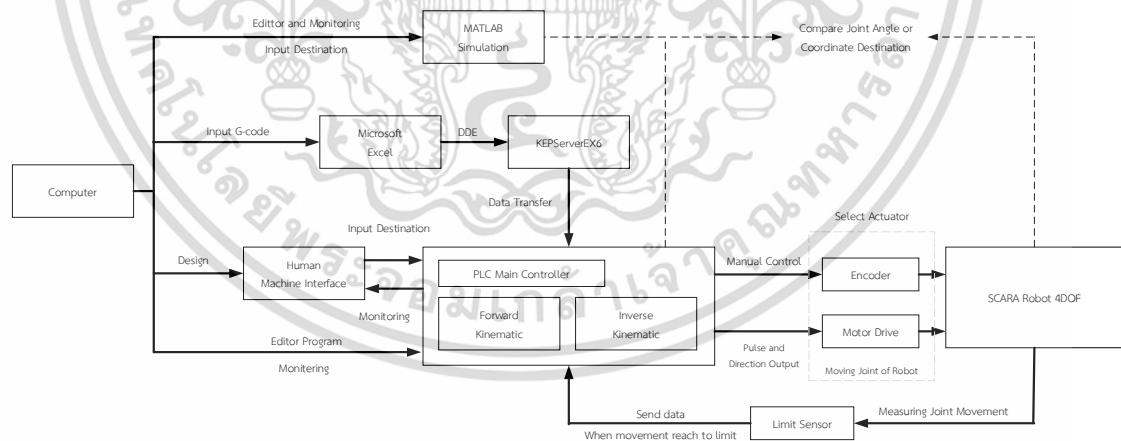
บทที่ 3

วิธีการดำเนินงาน

บทนี้จะกล่าวถึงวิธีการดำเนินงานจัดทำปริญญาณิพนธ์ซึ่งจะมีเนื้อหาที่เกี่ยวข้องกับกระบวนการในการทำงาน ขั้นตอนและวิธีการในการจัดทำ ตลอดจนการประยุกต์ใช้หลักการและทฤษฎีต่าง ๆ เพื่อให้สามารถจัดทำโครงการประยุกต์ใช้ PLC ในการควบคุมการเคลื่อนที่ของแขนกล SCARA Robot 4DOF ให้เป็นไปตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ ให้บรรลุวัตถุประสงค์ตามที่ได้กำหนดไว้ โดยสามารถจำแนกขั้นตอนในการดำเนินงานได้ดังต่อไปนี้

- 3.1 ภาพรวมการออกแบบการทำงาน
- 3.2 การออกแบบโครงสร้างการเชื่อมต่อแขนกล SCARA Robot 4DOF เข้ากับ PLC
- 3.3 การออกแบบโปรแกรมแบบจำลองทางคณิตศาสตร์
- 3.4 การออกแบบโปรแกรม PLC ควบคุมการทำงานของแขนกล
- 3.5 การเชื่อมต่อ OPC Server KEPServerEX6 เข้ากับ PLC

3.1 ภาพรวมการออกแบบการทำงาน



รูปที่ 3.1 ภาพรวมการทำงานของโครงการ

ในส่วนแรกจะเป็นการกล่าวถึงภาพรวมการทำงานทั้งหมดของโครงการการประยุกต์ใช้ PLC ควบคุม SCARA Robot 4DOF ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ จากรูปที่ 3.1 จะประกอบด้วยส่วนสำคัญหลัก 2 ส่วน ในส่วนแรกจะเป็นการประยุกต์เอา PLC มาใช้สำหรับการควบคุมการเคลื่อนที่ของแขนกล SCARA Robot 4DOF โดยที่ภายใน PLC จะมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมในการคำนวณค่าตามจลนศาสตร์การเคลื่อนที่ของแขนกล ซึ่งค่าเอาต์พุตจาก PLC ที่เป็นสัญญาณพัลส์และทิศทางการหมุนของมอเตอร์จะถูกส่งไปยังวงจรขับเคลื่อนมอเตอร์เพื่อขับเคลื่อนให้ข้อต่อแขนกลเคลื่อนที่ไปยังตำแหน่งเป้าหมายที่กำหนด โดยที่การเคลื่อนที่ของแขนกลนั้นจะถูกกำกับขอบเขตในการเคลื่อนที่เอาไว้ด้วยเซ็นเซอร์เพื่อป้องกันกรณีที่เกิดการทำงานผิดพลาด ซึ่งในส่วนของการประยุกต์ PLC เพื่อควบคุมแขนกลข้างต้นนี้จะมีการทำงานร่วมกับส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) ซึ่งมีการออกแบบมาเพื่อทำหน้าที่ให้ผู้ใช้งานได้ทำการกำหนดค่าตำแหน่งเป้าหมายของการควบคุมแขนกล และแสดงค่าเอาต์พุตของการคำนวณและประมวลผลให้ผู้ใช้งานได้ทำการบันทึกค่า สำหรับส่วนที่สองนั้นจะเป็นการสร้างแบบจำลองทางคณิตศาสตร์ในรูปแบบของกราฟฟิคคอมพิวเตอร์ผ่านการสร้าง Graphical user interface หรือ GUI บนโปรแกรม MATLAB เพื่อนำมาใช้ในการจำลองการเคลื่อนที่ของแขนกลในการเคลื่อนที่ไปยังตำแหน่งต่าง ๆ อีกทั้งยังใช้สำหรับเป็นเครื่องมือในการตรวจสอบความถูกต้องของการเคลื่อนที่ของแขนกลที่มีการเคลื่อนที่ไป ซึ่งในส่วนนี้จะมีการอภิปรายเพิ่มเติมในบทที่ 4 เป็นลำดับต่อไป นอกจากนี้ยังมีการสร้างการเชื่อมต่อกับ OPC Server ผ่าน KEPServerEX6 เพื่อใช้ในการรับส่งชุดข้อมูลของพิกัดตำแหน่งเพื่อใช้ในการควบคุมแขนกลให้มีการทำงานแบบอัตโนมัติในการเคลื่อนที่ไปยังตำแหน่งใดๆ ตามที่มีการกำหนดไว้

3.2 การออกแบบโครงสร้างการเชื่อมต่อแขนกล SCARA Robot 4DOF เข้ากับ PLC

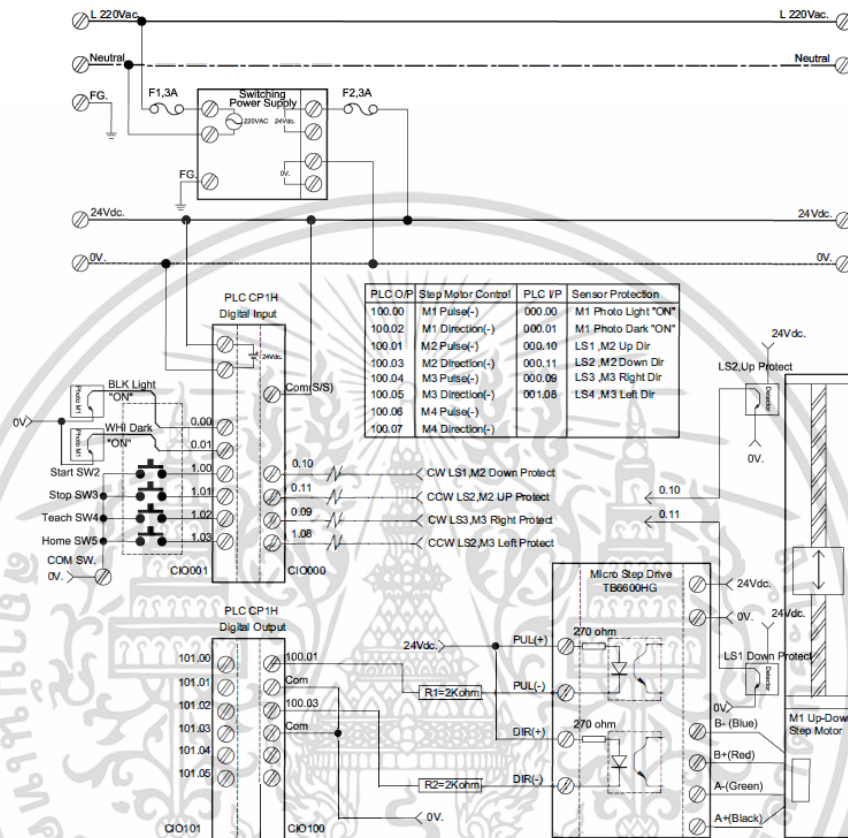
แขนกลที่นำมาใช้ในโครงการคือแขนกล SCARA Robot ซึ่งประกอบด้วยข้อต่อจำนวน 4 แกน ได้แก่ แกนที่ 1 (base), แกนที่ 3 (elbow) และแกนที่ 4 (end-effector) จะเป็นข้อต่อแบบหมุน ซึ่งมีการเคลื่อนที่โดยการหมุนรอบแกน z และแกนที่ 2 (shoulder) จะเป็นข้อต่อแบบเลื่อนที่มีลักษณะการเคลื่อนที่แบบขึ้นและลง



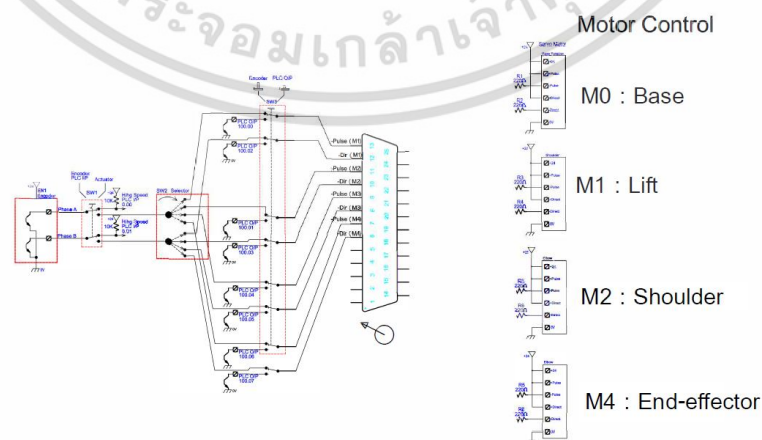
รูปที่ 3.2 SCARA Robot 4DOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อต่อของแขนกลแต่ละแกนจะมีการต่อเข้ากับมอเตอร์แบบสเต็ปเพื่อควบคุมการเคลื่อนที่ โดยมอเตอร์แบบสเต็ปนั้นจะถูกต่อเข้ากับวงจรขับเคลื่อนมอเตอร์ ซึ่งวงจรขับเคลื่อนมอเตอร์จะรับสัญญาณควบคุมมาจาก PLC เพื่อควบคุมให้มอเตอร์มีการหมุนไปตามที่ได้กำหนด โดยแผนผังการเชื่อมต่อวงจรระหว่าง PLC วงจรขับเคลื่อนมอเตอร์ และมอเตอร์แบบสเต็ป จะเป็นดังรูปที่ 3.3 และ 3.4 ดังนี้



รูปที่ 3.3 แสดงแผนผังการเชื่อมต่อวงจรระหว่าง PLC อุปกรณ์อินพุต-เอาต์พุต วงจรขับเคลื่อนมอเตอร์ และมอเตอร์แบบสเต็ปสำหรับข้อต่อ 1 แกน



รูปที่ 3.4 แสดงแผนผังการเชื่อมต่อวงจรของ PLC ในการต่อเข้ากับคอนเนคเตอร์เพื่อใช้สำหรับต่อเข้ากับวงจรขับเคลื่อนมอเตอร์ของข้อต่อแต่ละแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบโปรแกรมแบบจำลองทางคณิตศาสตร์

การสร้างแบบจำลองทางคณิตศาสตร์เพื่อทำการจำลองการเคลื่อนที่ของแขนกลไปยังตำแหน่งที่กำหนด ได้ทำการดำเนินงานในหลาย ๆ ส่วน ซึ่งเริ่มจากการออกแบบและสร้างโมเดลจำลองของแขนกล ไปจนถึงการนำสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับมาประยุกต์ใช้ในการสร้าง GUI ของแบบจำลอง ซึ่งสามารถอธิบายในแต่ละส่วนได้ดังนี้

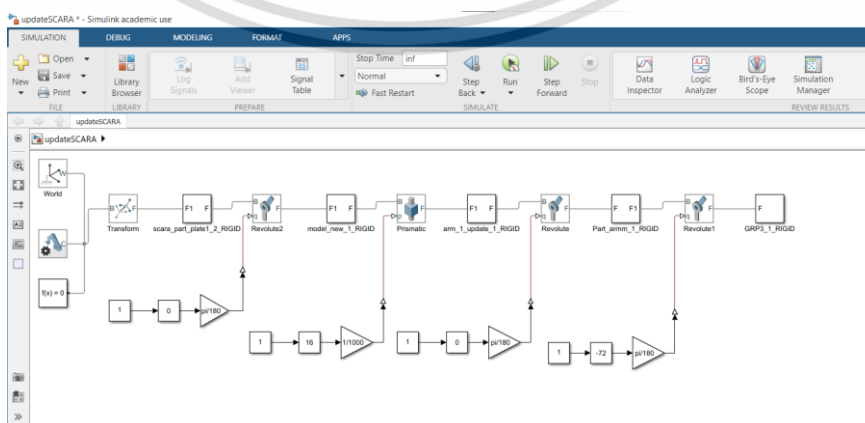
3.3.1 การออกแบบโมเดลจำลองแขนกล SCARA Robot 4DOF

เพื่อทำการจำลองการเคลื่อนที่ของแขนกลในรูปแบบของกราฟฟิคคอมพิวเตอร์ ได้มีการออกแบบและสร้างโมเดลของแขนกล SCARA Robot 4DOF โดยอ้างอิงขนาดและส่วนประกอบของ แขนกลตามจริงบนโปรแกรม SolidWorks ดังรูปที่ 3.5



รูปที่ 3.5 การออกแบบโมเดลจำลองของแขนกล SCARA Robot 4DOF

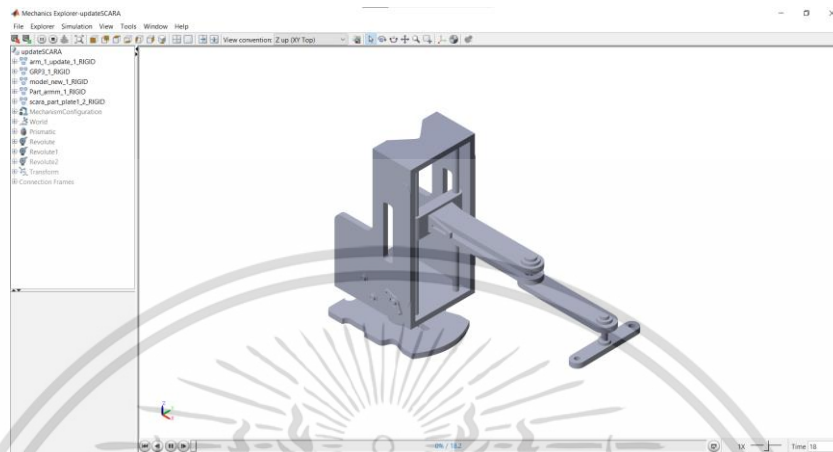
เมื่อได้โมเดลจำลองของแขนกลดังรูปที่ 3.5 แล้ว จากนั้นจะนำเอาเครื่องมือ Simscape Multibody Link มาทำการแปลงให้โมเดลของแขนกลจำลองที่ได้ออกแบบไว้ให้กลายเป็นบล็อกไดอะแกรมที่สามารถทำงานได้บน MATLAB Simulink เพื่อใช้เป็นโมเดลแขนกลสำหรับแบบจำลองทางคณิตศาสตร์ต่อไป



รูปที่ 3.6 บล็อกไดอะแกรมบนโปรแกรม MATLAB Simulink ของโมเดลแขนกลจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.6 เป็นการบล็อกไดอะแกรมบนโปรแกรม MATLAB Simulink ที่เกิดจากการแปลงโมเดลจำลองของแขนกล SCARA Robot 4DOF ซึ่งจะประกอบไปด้วยโครงสร้างของแขนกล และในแต่ละข้อต่อจะถูกกำกับกับการเคลื่อนที่ด้วย Slider gain ที่ทำหน้าที่คล้ายกับมอเตอร์ในแขนกลจริง



รูปที่ 3.7 โมเดลของแขนกลจำลองบนโปรแกรม MATLAB

3.3.2 การนำจลนศาสตร์การเคลื่อนที่ของแขนกลมาสร้างเป็นแบบจำลองทางคณิตศาสตร์

จากการศึกษาทฤษฎีการวิเคราะห์จลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า และแบบย้อนกลับในบทที่ 2 ทำให้ได้สมการจลนศาสตร์การเคลื่อนที่ของแขนกล SCARA Robot 4DOF แบบไปข้างหน้า ดังสมการที่ (2.33), (2.34) และ (2.35) และจลนศาสตร์การเคลื่อนที่แบบย้อนกลับดังสมการที่ (2.39), (2.42) และ (2.43) ซึ่งเมื่อทำการแทนค่าของความยาวก้านต่อลงในสมการจลนศาสตร์การเคลื่อนที่ของแขนกลทั้ง 2 แบบ โดยที่ d_0 มีค่าเป็น 0.120 m, d_4 มีค่าเป็น -0.039325 m, d_5 มีค่าเป็น -0.06 m, l_1 มีค่าเป็น 0.200 m และ l_2 มีค่าเป็น 0.155 m จะทำให้ได้สมการใหม่เป็น

สมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า จะได้

$$x_e = 0.2 \cdot \cos(\theta_1) + 0.155 \cdot \cos(\theta_1 + \theta_2) \quad (3.1)$$

$$y_e = 0.2 \cdot \sin(\theta_1) + 0.155 \cdot \sin(\theta_1 + \theta_2) \quad (3.2)$$

$$z_e = d_1 + 0.099325 \quad (3.3)$$

สมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ จะได้

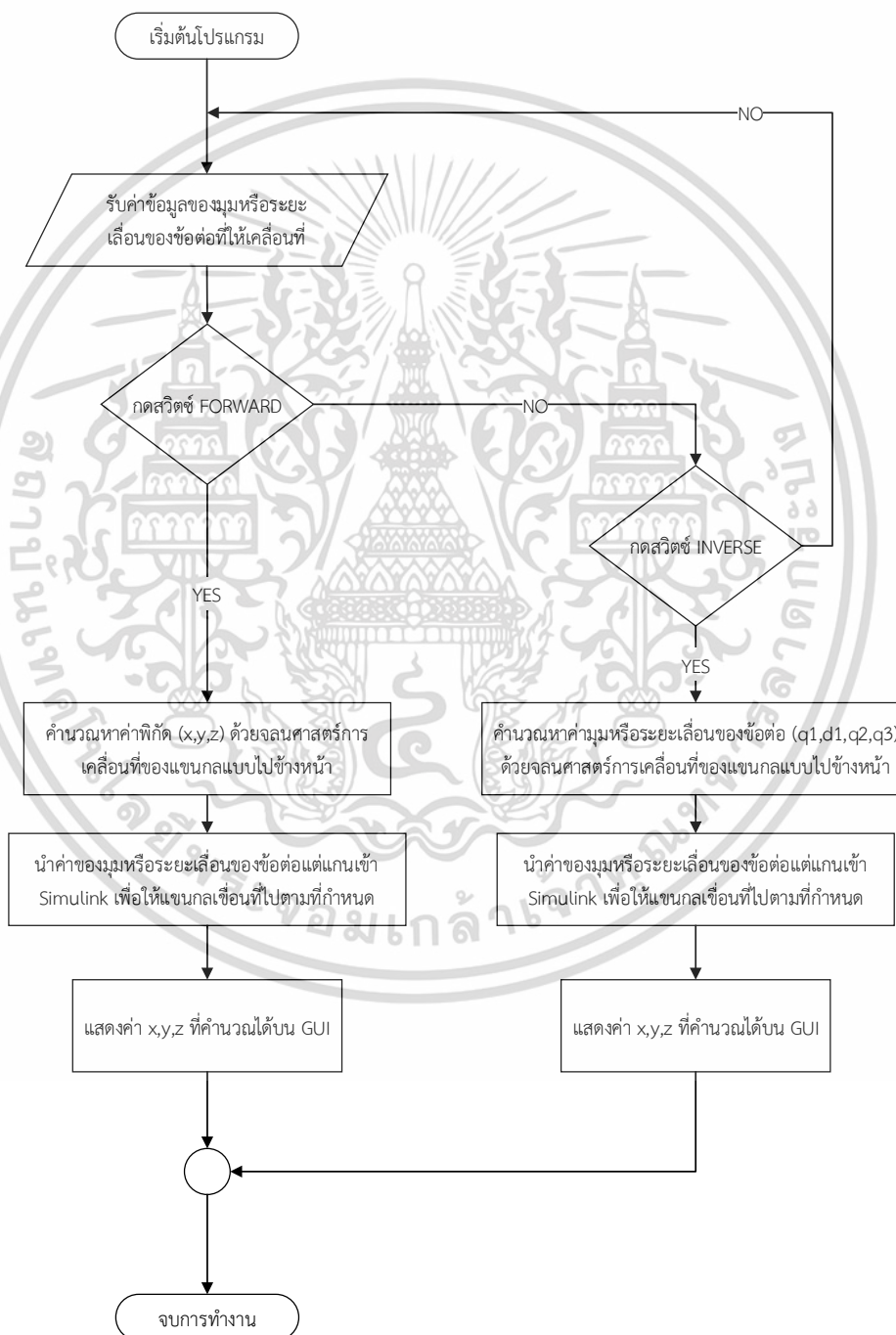
$$\theta_2 = \text{Atan2}\left(\pm \sqrt{1 - \cos^2(\theta_2)}, \frac{x^2 + y^2 - (0.200)^2 - (0.155)^2}{0.062}\right) \quad (3.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan2}(0.2 \cdot \sin(\theta_2), (0.2 + 0.155)\cos(\theta_2)) \quad (3.5)$$

$$d_1 = z - 0.219325 \quad (3.6)$$

จากนั้นได้ทำการสร้าง GUI โดยนำเอาสมการที่ (3.1) จนถึง (3.6) มาประยุกต์ใช้ โดยกำหนดให้มีเงื่อนไขในการทำงาน ดังนี้



รูปที่ 3.8 แสดง Flow chart เงื่อนไขการทำงานของแบบจำลองทางคณิตศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ModelName = 'updateSCARA';
global var;
t1=get(handles.edit1,'string');
t1 = str2num(t1);
if ((t1 <= -180)&(t1 >=180))
    msgbox('Please Select t1 Value In Range -180-->180');
end
d1=get(handles.edit2,'string');
d1 = str2num(d1);
if ((d1 <= 20)&(d1 >=180))
    msgbox('Please Select d1 Value In Range 20-->180');
end
t2=get(handles.edit3,'string');
t2 = str2num(t2);
t2 = t2*(-1);
if ((t2 <= -145)&(t2 >=145))
    msgbox('Please Select t2 Value In Range -145-->145');
end
t3=get(handles.edit4,'string');
t3 = str2num(t3);
if ((t3 <= -180)&(t3 >=180))
    msgbox('Please Select t3 Value In Range -180-->180');
end
set_param([ModelName '/Slider Gain'], 'Gain', num2str(t1))
set_param([ModelName '/Slider Gain1'], 'Gain', num2str(d1))
set_param([ModelName '/Slider Gain2'], 'Gain', num2str(t2))
set_param([ModelName '/Slider Gain3'], 'Gain', num2str(t3))
set(handles.edit4, 'string', num2str(t3));
%%Kinematics Forward Model
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
H01 = [cosd(t1) -sind(t1) 0 0; sind(t1) cosd(t1) 0 0; 0 0 1 d0; 0 0 0 1];
H12 = [1 0 0 0; 0 1 0 0; 0 0 1 d1/1000; 0 0 0 1];
H23 = [cosd(t2) -sind(t2) 0 L2; sind(t2) cosd(t2) 0 0; 0 0 1 0; 0 0 0 1];
H34 = [cosd(t3) -sind(t3) 0 L3; -sind(t3) cosd(t3) 0 0; 0 0 -1 -d4; 0 0 0 1];
H4E = [1 0 0 0; 0 1 0 0; 0 0 1 -d5; 0 0 0 1];
H0E = H01*H12*H23*H34*H4E;
px = H0E(1,4)*1000;
py = H0E(2,4)*1000;
pz = H0E(3,4)*1000;
set(handles.edit5, 'string', num2str(px));
set(handles.edit6, 'string', num2str(py));
set(handles.edit7, 'string', num2str(pz));

```

รูปที่ 3.9 แสดงการเขียนโปรแกรมสำหรับคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า

```

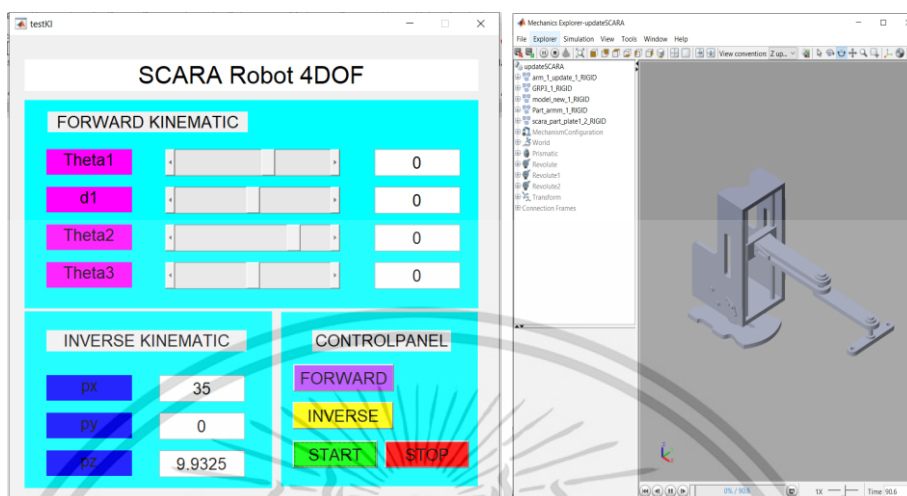
ModelName = 'updateSCARA';
global var;
px=get(handles.edit5,'string');
px = str2num(px);
if ((px <= -350)&(px >=350))
    msgbox('Please Select PX Value In Range -350-->350');
end
py=get(handles.edit6,'string');
py = str2num(py);
if ((py <= -350)&(py >=350))
    msgbox('Please Select PY Value In Range -350-->350');
end
pz=get(handles.edit7,'string');
pz = str2num(pz);
if ((pz <= 20)&(pz >= 180))
    msgbox('Please Select PZ Value In Range 20-->180');
end
pxi = px/1000;
pyi = py/1000;
pzi = pz/1000;
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
C2 = ((pxi)^2 + (pyi)^2 - (L2)^2 - (L3)^2)/(2*L2*L3);
C2f = round(C2*1000)/1000;
S2 = 1*sqrt(1-(C2f)^2);
if (S2 <= -1)
    msgbox('Robot Crash Close Program to Restart');
end
t2 = atan2d(S2, C2f)*(-1);
t1n = t2*(-1);
t1 = atan2d((pyi), (pxi))-atan2d(L3*S2, L2+(L3*C2f));
d1 = (pzi + d4 - d5 - d0)*1000;
t3 = 0;
guidata(hObject, handles);
set(handles.edit1, 'string', t1);
set_param([ModelName '/Slider Gain'], 'Gain', num2str(t1))
set(handles.edit2, 'string', d1);
set_param([ModelName '/Slider Gain1'], 'Gain', num2str(d1))
set(handles.edit3, 'string', t2);
set_param([ModelName '/Slider Gain2'], 'Gain', num2str(t1n))
set(handles.edit4, 'string', t3);
set_param([ModelName '/Slider Gain3'], 'Gain', num2str(t3))

```

รูปที่ 3.10 แสดงการเขียนโปรแกรมสำหรับคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะได้ผลสุดท้ายของแบบจำลองทางคณิตศาสตร์ที่ใช้สำหรับการจำลองการเคลื่อนที่ของแขนกล เป็น GUI เชื่อมต่อกับผู้ใช้งาน ดังรูปที่ 3.11



รูปที่ 3.11 แสดงหน้าต่าง GUI และโมเดลแขนกลจำลอง

จากรูปที่ 3.11 จะแสดงหน้าต่างของ GUI แบบจำลองทางคณิตศาสตร์ ซึ่งจะแบ่งการทำงานออกเป็น 2 ส่วน ซึ่งได้แก่ส่วนแรกที่เป็นส่วนสำหรับควบคุมโดยผู้ใช้งานมีไว้สำหรับการทำกระบวนการคำนวณหรือการเคลื่อนที่ของข้อต่อ ($\theta_1, d_1, \theta_2, \theta_3$) สำหรับการจำลองจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า และค่าพิกัดตำแหน่งเป้าหมาย (x, y, z) สำหรับการจำลองจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ และส่วนที่สองคือส่วนของโปรแกรมในการรับส่งค่าข้อมูลจากส่วนสำหรับควบคุมที่ผ่านการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลเรียบร้อยแล้ว มาทำให้โมเดลของแขนกลจำลองนั้นเกิดการเคลื่อนที่ไปตามที่กำหนด

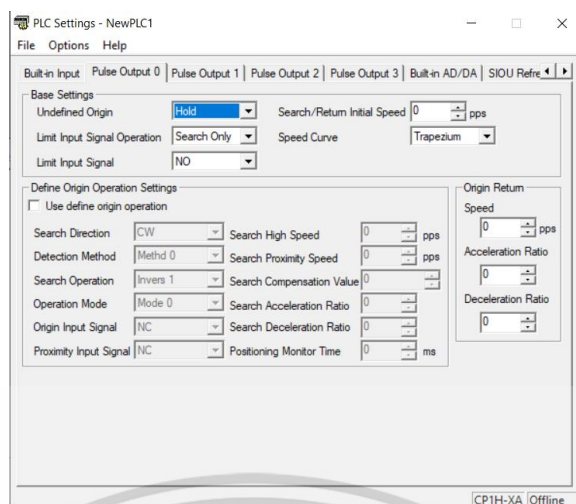
3.4 การออกแบบโปรแกรม PLC ควบคุมการทำงานของแขนกล

การออกแบบโปรแกรมของการควบคุมแขนกล SCARA Robot 4DOF ได้มีการกำหนดเงื่อนไขในการทำงาน และเขียนโปรแกรมควบคุมให้กับ PLC รวมไปถึงการออกแบบส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) ซึ่งสามารถอธิบายวิธีการดำเนินการได้ดังนี้

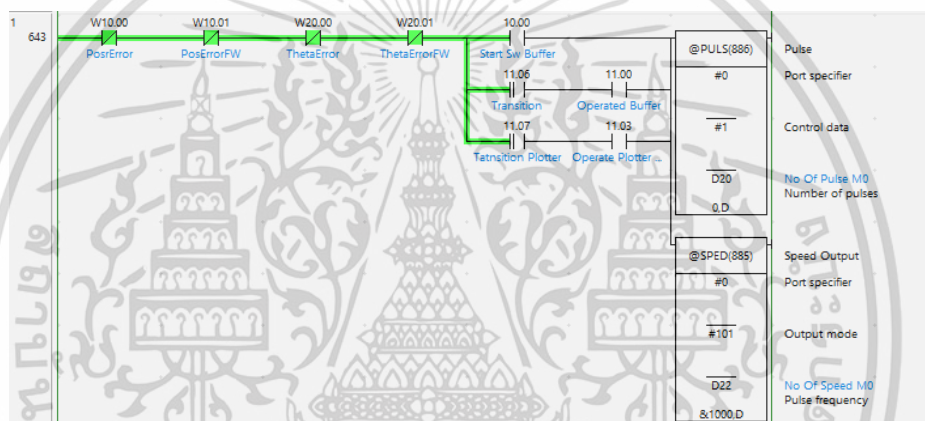
3.4.1 การเขียนโปรแกรม PLC Ladder ควบคุมการทำงานของมอเตอร์แบบสเต็ป

การเขียนโปรแกรมเพื่อควบคุมการทำงานของมอเตอร์จะเริ่มจากการตั้งค่า Pulse Output ให้ตรงกับลักษณะการทำงานของมอเตอร์ จากนั้นใช้ฟังก์ชัน Pulse(886) ซึ่งเป็นฟังก์ชันที่ใช้สำหรับการป้อนระยะทางในการเคลื่อนที่ของมอเตอร์ และจะใช้ฟังก์ชัน SPED(885) ในการกำหนดความเร็วและทิศทางของมอเตอร์ที่จะเคลื่อนที่ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



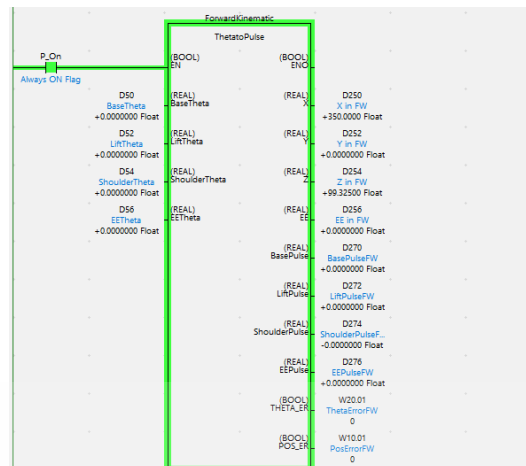
รูปที่ 3.12 การตั้งค่า Pulse Output



รูปที่ 3.13 โปรแกรมการควบคุมมอเตอร์แบบสแต็ปของแกนกล

3.4.2 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า

การควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้านั้นได้มีการเขียนโปรแกรมด้วยภาษา Structure Text ในการนำเอาสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าตามสมการที่ (3.1) ถึงสมการที่ (3.3) มาทำการคำนวณเพื่อหาค่าของพิกัดตำแหน่งของส่วนปลายของแขนกล จากอินพุตที่เป็นค่ามุมหรือระยะเลื้อนของมุมที่ต้องการให้ข้อต่อแขนกลเคลื่อนไป



รูปที่ 3.14 Function Block ของการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า

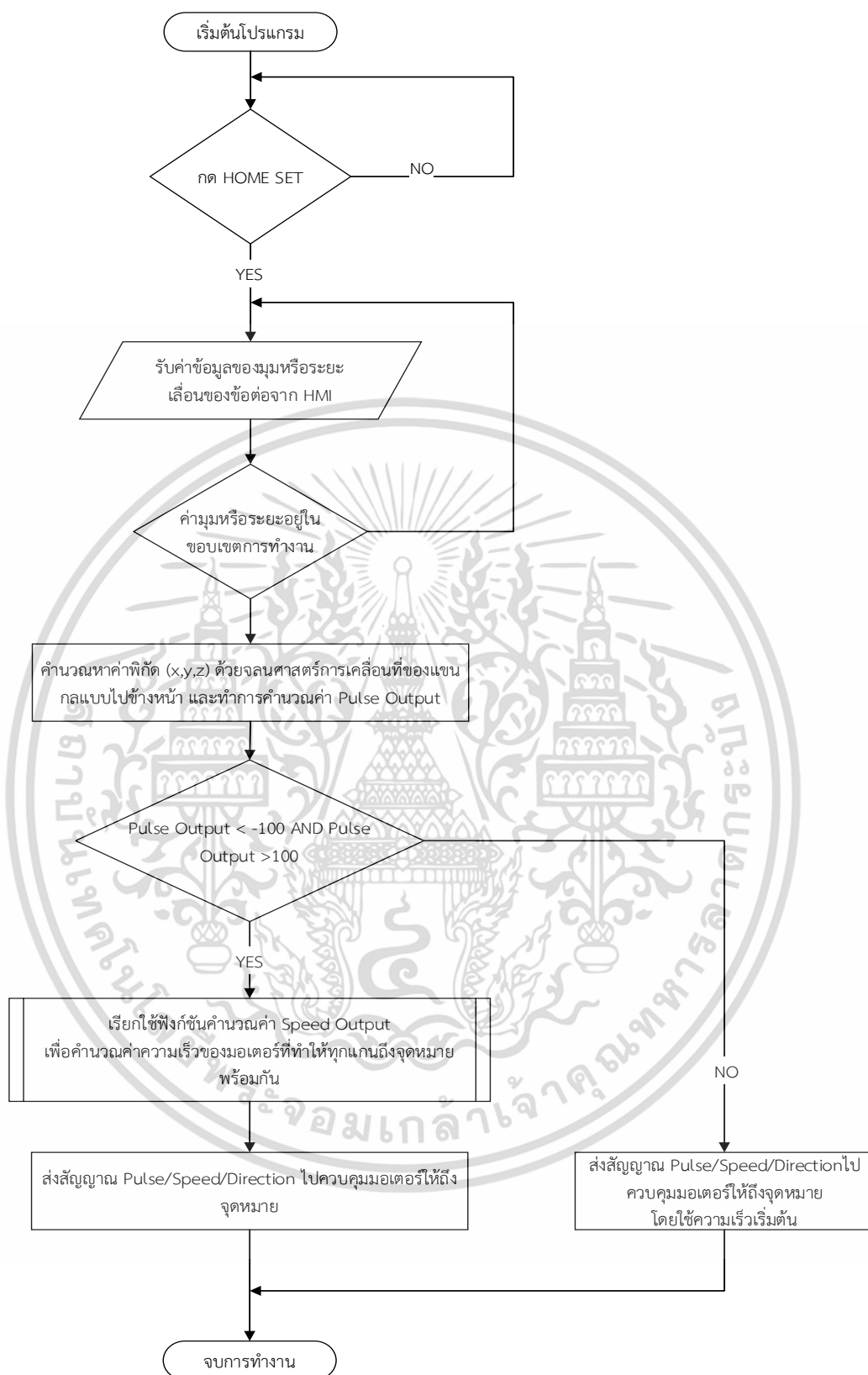
```

1 RadBase := DEG_TO_RAD(BaseTheta);
2 RadShoulder := DEG_TO_RAD(ShoulderTheta);
3 RadEE := DEG_TO_RAD(EETTheta);
4 mmLIR := LIRTheta/1000.0;
5
6 Xe := ((0.20*(COS(RadBase)))+(0.15*(COS(RadBase+RadShoulder))));
7 Ye := ((0.20*(SIN(RadBase)))+(0.15*(SIN(RadBase+RadShoulder))));
8 Ze := (0.12 + mmLIR) - ((0.039325) - (-0.06));
9
10 X := Xe*1000.0;
11 Y := Ye*1000.0;
12 Z := Ze*1000.0;
13 EE := EETTheta;
14
15 IF (BaseTheta > BaseThetaMax OR BaseTheta < BaseThetaMin OR LIRTheta > LIRMax OR LIRTheta < LIRMin OR ShoulderTheta > ShoulderThetaMax OR ShoulderTheta < ShoulderThetaMin OR (EETTheta > EEMax OR EETTheta < EEMin)) THEN
16   THETA_ER := TRUE;
17 ELSE
18   THETA_ER := FALSE;
19   BasePulse := BaseTheta * 90.0;
20   LIRPulse := LIRTheta * 200.0;
21   ShoulderPulse := ShoulderTheta * 52.941 * (-1.0);
22   EEPulse := EETTheta * 17.64;
23 END_IF;
24
25 IF (X > 265.0 OR X < -265.0 OR Y > 265.0 OR Y < -265.0 OR Z >= 175.0 OR Z < 20.0) THEN
26   POS_ER := TRUE;
27 ELSE
28   POS_ER := false;
29 END_IF;
30

```

รูปที่ 3.15 Structure Text ในการคำนวณหาตำแหน่งของส่วนปลายของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

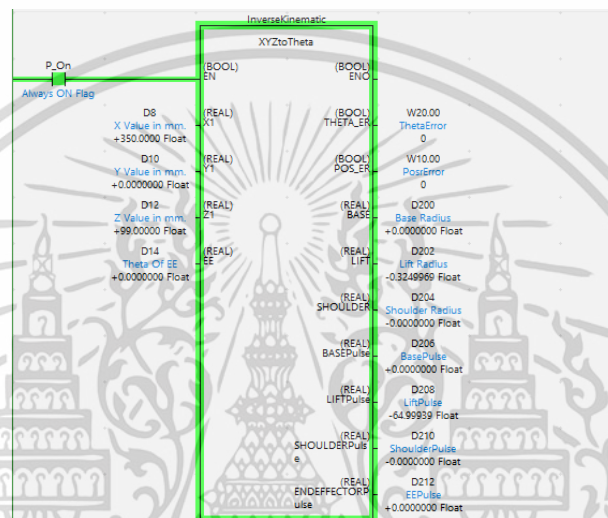


รูปที่ 3.16 แสดง Flowchart เงื่อนไขการทำงานของโปรแกรม PLC ควบคุมการทำงานของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ

การเขียนโปรแกรมเพื่อควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ ย้อนกลับจะมีการเขียนโปรแกรม Structure Text เพื่อสร้าง Function Block ในการคำนวณค่าจลนศาสตร์การเคลื่อนที่ของแขนกลเช่นเดียวกัน โดยนำเอาสมการที่ (3.4) ถึงสมการที่ (3.6) ได้จากการแทนค่ามาประยุกต์ใช้ เพื่อหาค่าของมุมหรือข้อต่อของแขนกลที่จะเคลื่อนที่ไปยังพิกัดตำแหน่งที่กำหนด



รูปที่ 3.17 Function Block ของการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ

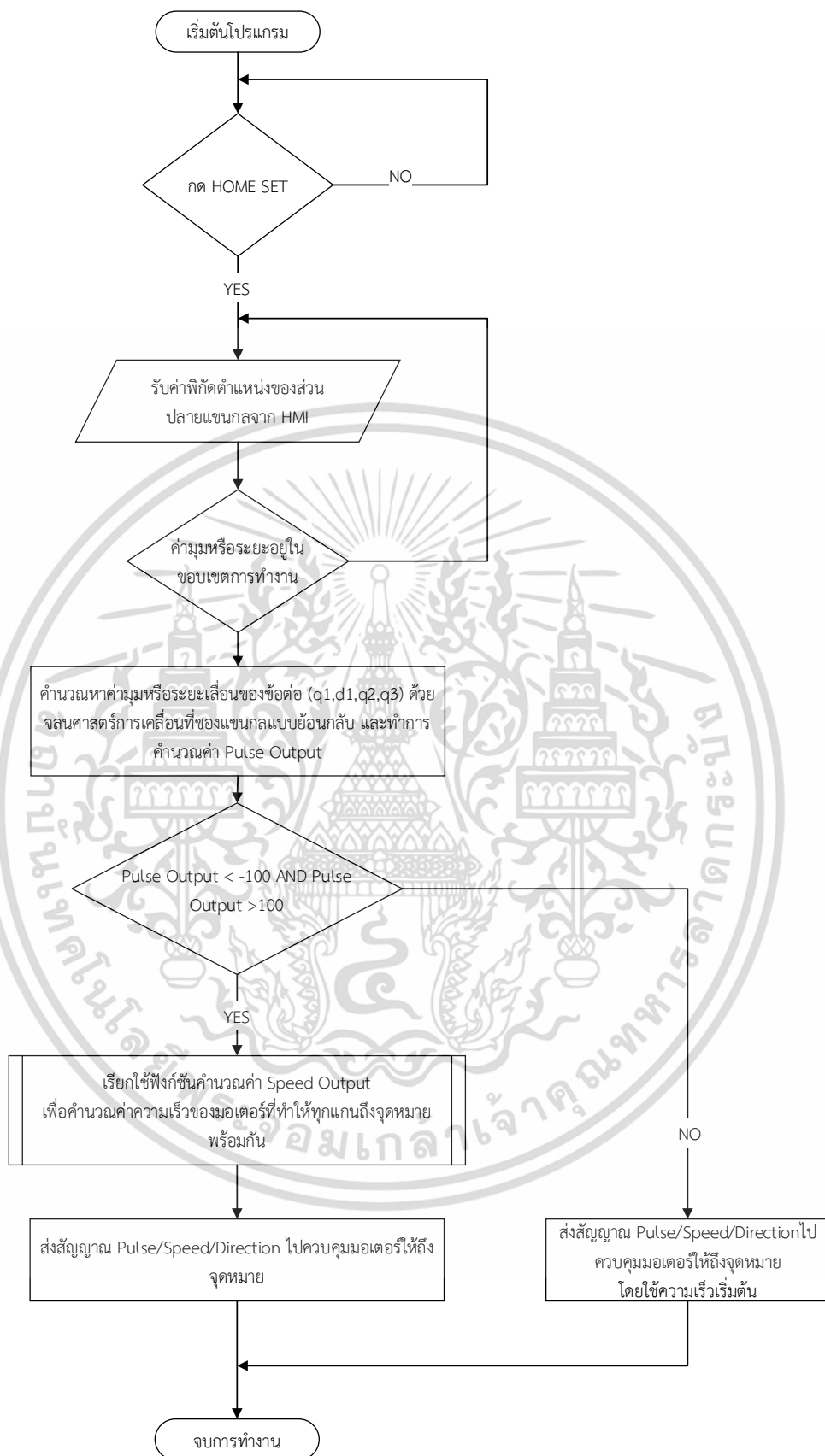
```

1  RADIUS := SQRT((X1**2)+(Y1**2)+(Z1**2));
2  IF (RADIUS > 405.0 OR RADIUS < 150.0) OR (X1 > X1max OR X1 < X1min) OR (Y1 > Y1max OR Y1 < Y1min) OR (Z1 > Z1max OR Z1 < Z1min) THEN
3  POS_ER := true;
4  ELSE
5  POS_ER := false;
6  END_IF;
7
8  s1 := (X1**2)+(Y1**2)-(L2**2)-(L3**2);
9  s2 := 2.0*L2*L3;
10 s3 := s1/s2;
11 h1 := 1.0*(SQRT(1.0-s3**2));
12 SHOULDER := (-1.0)*RAD_TO_DEG(ACOS(s3));
13
14 s4 := L3*s1;
15 s5 := L2*(L3*s3);
16 s6 := s4/s5;
17 IF (X1 > (-1.0) AND Y1 > (-1.0)) THEN
18 BASE := RAD_TO_DEG(ATAN(Y1/X1)+ATAN(s6));
19 END_IF;
20 IF (X1 < (-1.0) AND Y1 > (-1.0)) THEN
21 BASE := 90.0 + (90.0 - ABS(RAD_TO_DEG(ATAN(Y1/X1)+ATAN(s6))));
22 END_IF;
23 IF (X1 < (-1.0) AND Y1 < (-1.0)) THEN
24 BASE := (-1.0)*ABS(RAD_TO_DEG(ATAN(Y1/X1)+ATAN(s6)));
25 END_IF;
26 IF (X1 < (-1.0) AND Y1 < (-1.0)) THEN
27 BASE := (-1.0)*(90.0 + (90.0 - ABS(RAD_TO_DEG(ATAN(Y1/X1)+ATAN(s6))));
28 END_IF;
29
30 LIFT := Z1 + dd4 - dd5 - dd0;
31
32 IF (BASE > BaseThetaMax OR BASE < BaseThetaMin) OR (LIFT > LiftMax OR LIFT < LiftMin) OR (SHOULDER > ShoulderThetaMax OR SHOULDER < ShoulderThetaMin) OR (EE > EEMax OR EE < EEMin) THEN
33 THETA_ER := TRUE;
34 ELSE
35 THETA_ER := FALSE;
36 BASEPulse := BASE*90.0;
37 LIFTpulse := LIFT*200.0;
38 SHOULDERPulse := SHOULDER*52.841;
39 ENDEFFECTORPulse := EE*17.64;
40 END_IF;

```

รูปที่ 3.18 Structure Text ในการคำนวณหาค่ามุมหรือระยะเลื่อนของข้อต่อแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

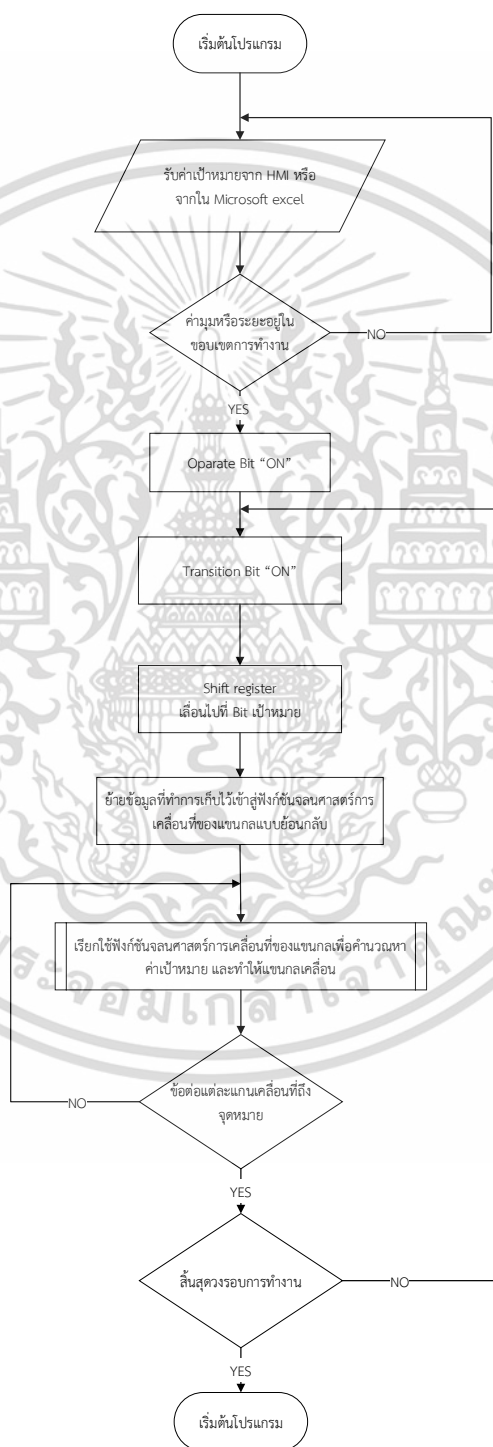


รูปที่ 3.19 แสดง Flowchart เงื่อนไขการทำงานของโปรแกรม PLC ควบคุมการทำงานของแขนกล ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่เป็นแบบอัตโนมัติ

การประยุกต์ใช้แขนกลเพื่องานอุตสาหกรรมนั้นจะมีความทำงานที่มีการเคลื่อนที่ไปในหลาย ๆ จุด ดังนั้นจึงได้มีการออกแบบโปรแกรมเพื่อควบคุมแขนกลโดยให้ผู้ใช้สามารถป้อนค่าอินพุตได้หลายค่า และให้แขนกลมีการเคลื่อนที่เป็นวงรอบ ซึ่งจะใช้การเขียนด้วยภาษา Ladder และใช้ Shift register เป็นตัวชี้ตำแหน่ง โดยสามารถเขียนลักษณะเงื่อนไขของโปรแกรมได้ดังนี้



รูปที่ 3.20 เงื่อนไขการทำงานของโปรแกรมควบคุมแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 การเขียนโปรแกรมในการคำนวณค่าความเร็วที่สั่งให้มอเตอร์

ในการเคลื่อนที่ของแขนกลไปยังตำแหน่งเป้าหมายใด ๆ การเคลื่อนที่ของข้อต่อแต่ละแกนนั้นจะต้องมีความสัมพันธ์กัน กล่าวคือข้อต่อแต่ละแกนนั้นจะต้องเคลื่อนที่ไปยังตำแหน่งเป้าหมายโดยใช้ระยะเวลาที่เท่า ๆ กัน ซึ่งได้มีการเขียนโปรแกรมด้วยภาษา Structure Text ในการคำนวณหาความเร็วของข้อต่อแต่ละแกนไว้ดังนี้

```

1 BaseSpeed_Real := BaseSpeed;
2 IF(BaseNo_Of_Pulse > 100.0 OR BaseNo_Of_Pulse < -100.0) THEN
3   LiftSpeed_Real := ABS((LiftNo_Of_Pulse)/(BaseNo_Of_Pulse)) * BaseSpeed_Real;
4   ShoulderSpeed_Real := ABS((ShoulderNo_Of_Pulse)/(BaseNo_Of_Pulse)) * BaseSpeed_Real;
5   EndeffectorSpeed_Real := ABS((EndeffectorNo_Of_Pulse)/(BaseNo_Of_Pulse)) * BaseSpeed_Real;
6 IF(LiftNo_Of_Pulse = 0.0) THEN
7   LiftSpeed_Real := BaseSpeed;
8 END_IF;
9 IF(ShoulderNo_Of_Pulse = 0.0) THEN
10  ShoulderSpeed_Real := BaseSpeed;
11 END_IF;
12 IF(EndeffectorNo_Of_Pulse = 0.0) THEN
13  EndeffectorSpeed_Real := BaseSpeed;
14 END_IF;
15 END_IF;

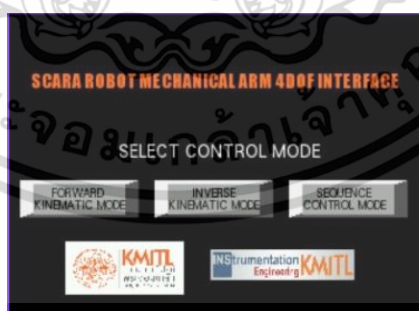
```

รูปที่ 3.21 Structure Text ในการคำนวณหาความเร็วของข้อต่อแต่ละแกนในการเคลื่อนที่

3.4.6 การออกแบบส่วนเชื่อมต่อกับผู้ใช้งาน

ในการออกแบบส่วนเชื่อมต่อกับผู้ใช้งานได้มีการออกแบบเพื่อให้ผู้ใช้งานทำการป้อนค่าอินพุตซึ่งเป็นค่าตำแหน่งเป้าหมายของการควบคุมแขนกล ซึ่งแบ่งออกได้เป็น 6 หน้าต่างดังนี้

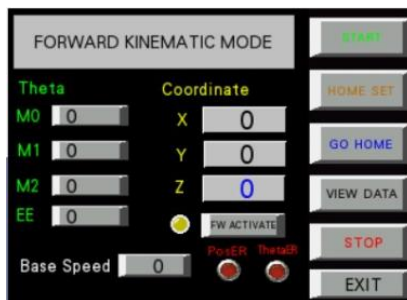
- 1) Main Menu เป็นหน้าต่างสำหรับเลือกรูปแบบการควบคุม ซึ่งจะประกอบไปด้วยการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า, การควบคุมตามจลนศาสตร์การเคลื่อนที่แบบย้อนกลับ และการควบคุมแบบอัตโนมัติ



รูปที่ 3.22 Main Menu

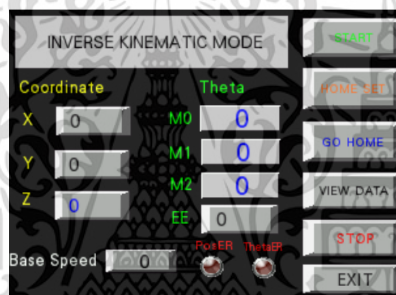
- 2) Forward Kinematic Mode เป็นหน้าต่างสำหรับการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า ซึ่งจะประกอบไปด้วยส่วนของการป้อนค่ามุมหรือระยะเลื้อนของข้อต่อแต่ละข้อต่อ และส่วนแสดงผลค่าพิกัดตำแหน่งของส่วนปลายแขนกลที่ประมวลผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



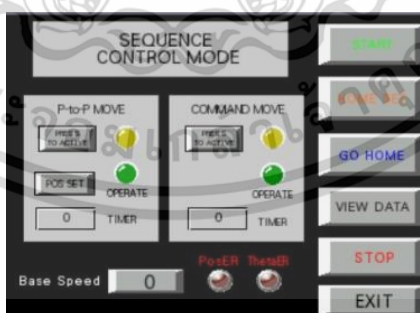
รูปที่ 3.23 หน้าต่าง Forward Kinematic Mode

- 3) Inverse Kinematic Mode เป็นหน้าต่างสำหรับการควบคุมตามจลนศาสตร์ การเคลื่อนที่ของแขนกลแบบย้อนกลับ ซึ่งจะประกอบไปด้วยส่วนของการป้อนค่า พิกัดตำแหน่งของส่วนปลายแขนกล และส่วนแสดงผลค่ามุมหรือระยะเลื่อนของแต่ละข้อต่อที่ประมวลผลได้



รูปที่ 3.24 หน้าต่าง Inverse Kinematic Mode

- 4) Sequence Control Mode เป็นหน้าต่างสำหรับการควบคุมแบบอัตโนมัติ



รูปที่ 3.24 หน้าต่าง Sequence Control Mode

- 5) VIEW DATA เป็นหน้าต่างสำหรับการดูเอาต์พุตที่ได้จากการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 หน้าต่าง VIEW DATA

- 6) POS SETTING เป็นหน้าต่างสำหรับการป้อนค่าพิกัดตำแหน่งสำหรับการเคลื่อนที่แบบ Point to Point

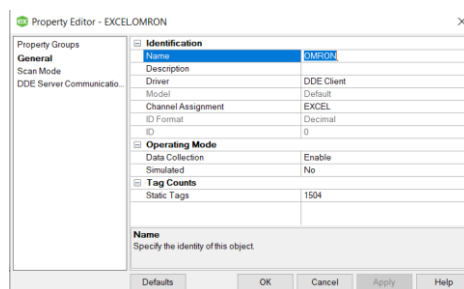


รูปที่ 3.26 หน้าต่าง POS SETTING

3.5 การเชื่อมต่อ KEPSeverEX6 เข้ากับ PLC

3.5.1 การเชื่อมต่อ Microsoft Excel กับ KEPSeverEX6

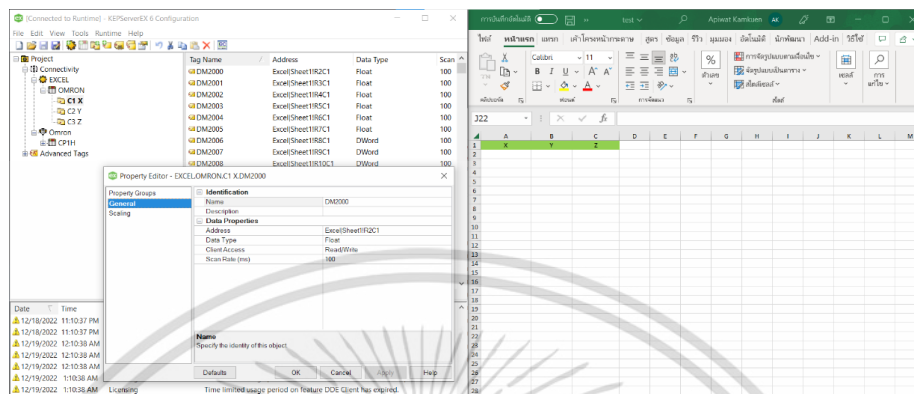
ทำการเชื่อมต่อ Microsoft Excel กับ KEPSeverEX6 ซึ่งในการเชื่อมต่อนั้นใช้ฟังก์ชัน DDE (Dynamic Data Exchange) ของ KEPSeverEX6 ในการอ่านข้อมูลจากเซลล์ที่ระบุใน Microsoft Excel เวอร์กชีตที่ได้กำหนดไว้ โดยผู้จัดทำได้ทำการสร้าง Device โดยผู้จัดทำตั้งชื่อ "OMRON" โดยมีรายละเอียดดังนี้



รูปที่ 3.27 กำหนดการเชื่อมต่อฟังก์ชัน DDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อได้เผยแพร่ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

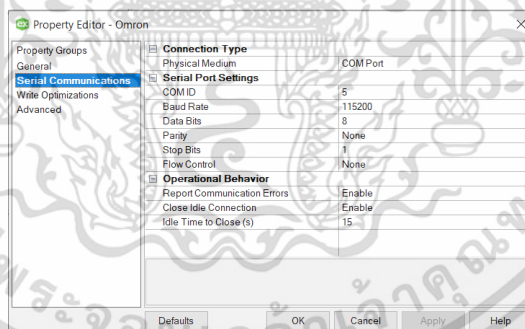
ผู้จัดทำต้องการเชื่อมต่อข้อมูลระหว่าง Microsoft Excel กับ DDE Server โดยการสร้าง tag ระบุแถวและคอลัมน์ เช่น “R1C2” จากนั้นฟังก์ชัน DDE จะร้องข้อมูลจากเซลล์ที่แถว 1 คอลัมน์ 2 ในเวิร์กชีต Excel โดย Address = DDE Service Name|Topic Name|Item Name ดังรูป



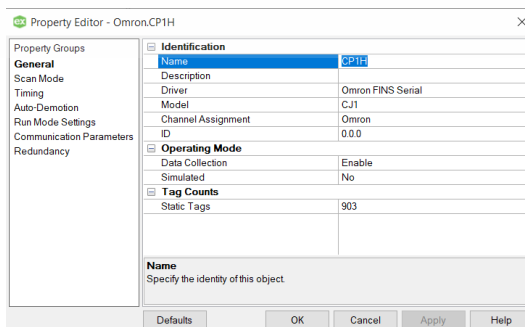
รูปที่ 3.28 กำหนดการเชื่อมต่อฟังก์ชัน DDE (ต่อ)

3.5.2 การเชื่อมต่อ KEPServerEX6 กับ CX-Programmer

ก่อนจะเริ่มใช้งาน KEPServerEX6 ต้องทราบถึงโครงสร้างฮาร์ดแวร์ที่เชื่อมต่อและ tag ใน address ที่ต้องการเชื่อมต่อ ซึ่งการเชื่อมต่อการส่งข้อมูลระหว่าง PLC Omron CP1H โดยใช้ซอฟต์แวร์ KEPServerEX6 ซึ่งมีขั้นตอนการตั้งค่าดังนี้



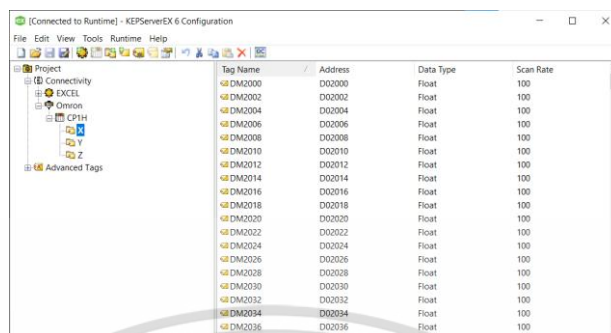
รูปที่ 3.29 แสดงการเชื่อมต่อระหว่าง KEPServerEX6 กับ CX-Programmer



รูปที่ 3.30 แสดงการเชื่อมต่อระหว่าง KEPServerEX6 กับ CX-Programmer (ต่อ)

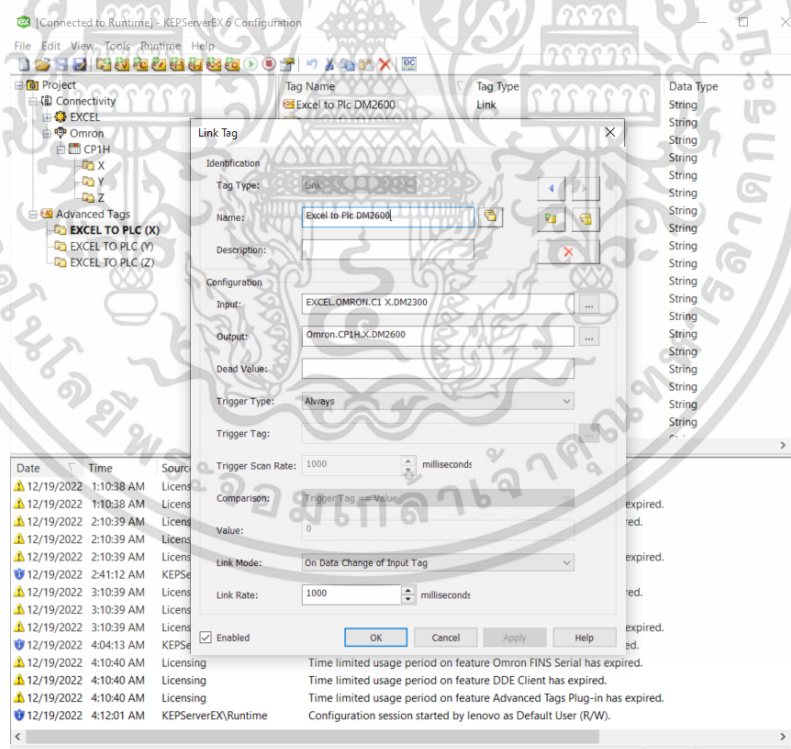
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการสร้าง OPC tag ใน address จากจากโปรแกรม CX-Programmer เพื่อเชื่อมต่อไปยังอินพุตและเอาต์พุตของ PLC ดังรูป



รูปที่ 3.31 ตัวอย่าง tag ของโปรแกรม CX-Programmer

เมื่อสร้าง OPC tag เรียบร้อยแล้ว ต่อไปทำการ Link Tag ระหว่าง Excel กับ CX-Programmer โดยใช้ฟังก์ชัน Advanced Tags ในการเชื่อมต่อที่ใช้ในการอ่าน-เขียน ข้อมูลลงใน Data-memory ของ PLC



รูปที่ 3.32 ตัวอย่าง Link Tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้จะเป็นการทดสอบและหาประสิทธิภาพของการประยุกต์ PLC ควบคุมแขนกล SCARA Robot 4DOF ในการเคลื่อนที่ไปยังตำแหน่งเป้าหมาย ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับให้เป็นไปตามวัตถุประสงค์ที่ได้มีการกำหนดไว้ ซึ่งสามารถแบ่งผลของการทดลองออกเป็น 3 หัวข้อดังนี้

- 4.1 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์
- 4.2 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์
- 4.3 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่แบบจุดต่อจุด (Point to Point)

4.1 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์

การทดลองนี้จะเป็นการควบคุมแขนกลให้เคลื่อนที่โดยการกำหนดอินพุตซึ่งเป็นค่าของตำแหน่งเป้าหมายที่เป็นค่ามุมหรือระยะเลื่อนของข้อต่อแต่ละแกนที่ต้องการให้มีการเคลื่อนที่ไปให้กับส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) จากนั้นจะนำผลของการเคลื่อนที่ของแขนกลจริง ซึ่งเป็นค่าของพิกัดตำแหน่ง (x, y, z) สุดท้ายของส่วนปลายของแขนกล (End effector) ที่ตำแหน่งใด ๆ ซึ่งเกิดจากการเคลื่อนที่ของข้อต่อ มาเปรียบเทียบกับผลของการจำลองการเคลื่อนที่จากในแบบจำลองทางคณิตศาสตร์ เพื่อให้ทราบถึงความถูกต้องในการเคลื่อนที่ของแขนกลจริง ทั้งในเรื่องของทิศทางในการเคลื่อนที่ไป หรือตำแหน่งเป้าหมายของแขนกลที่ควรเคลื่อนที่ไป และการทดลองในการเคลื่อนที่ไปในแต่ละตำแหน่งจะทดลองทั้งหมด 5 ครั้งเพื่อหาความสามารถในการทำซ้ำได้ (Repeatability) ของแขนกลในการเคลื่อนที่ไปยังตำแหน่งเป้าหมายตามที่กำหนด โดยเงื่อนไขในการทดลองจะกำหนดอินพุตของค่ามุมหรือระยะเลื่อนของแต่ละข้อต่อให้มีค่าอยู่ที่ 20%, 80%, -30% และ -70% ของการเคลื่อนที่สูงสุดในแต่ละข้อต่อ ยกเว้นข้อต่อของส่วนปลายของแขนกล (θ_e) ที่ไม่เกิดการเคลื่อนที่ใดๆ ซึ่งจะให้ผลการทดลองเป็นดังนี้

ตารางที่ 4.1 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า เมื่อกำหนดอินพุตที่ 20% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่ามุมหรือระยะเลื่อนของข้อต่อที่ 20% (องศา หรือ มม.)				ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่ได้จากในแบบจำลอง (มม.)			ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่วัดจริง (มม.)			ทิศทางการเคลื่อนที่
	θ_1	d_1	θ_2	θ_e	x	y	z	x	y	z	
1	36	16	29	0	310.6	135.8	115.3	311.9	136	115.5	ถูกต้อง
2	36	16	29	0	310.6	135.8	115.3	311.2	136	115.1	ถูกต้อง
3	36	16	29	0	310.6	135.8	115.3	311.4	136.1	115.3	ถูกต้อง
4	36	16	29	0	310.6	135.8	115.3	311.2	135.9	115.8	ถูกต้อง
5	36	16	29	0	310.6	135.8	115.3	311.4	135.7	115.3	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่					310.6	135.8	115.3	311.4	135.9	115.4	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

ตารางที่ 4.2 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า เมื่อกำหนดอินพุตที่ 80% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่ามุมหรือระยะเลื่อนของข้อต่อที่ 80% (องศา หรือ มม.)				ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่ได้จากในแบบจำลอง (มม.)			ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่วัดจริง (มม.)			ทิศทางการเคลื่อนที่
	θ_1	d_1	θ_2	θ_e	x	y	z	x	y	z	
1	144	64	116	0	-29.3	187.9	163.3	-29	188.5	163.9	ถูกต้อง
2	144	64	116	0	-29.3	187.9	163.3	-29.4	187	163.1	ถูกต้อง
3	144	64	116	0	-29.3	187.9	163.3	-29.4	187	163.1	ถูกต้อง
4	144	64	116	0	-29.3	187.9	163.3	-29.9	187.8	163.5	ถูกต้อง
5	144	64	116	0	-29.3	187.9	163.3	-29.3	187.4	163.5	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่					-29.3	187.9	163.3	-29.4	187.5	163.4	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

ตารางที่ 4.3 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า เมื่อกำหนดอินพุตที่ -30% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่ามุมหรือระยะเลื่อนของข้อต่อที่ -30% (องศา หรือ มม.)				ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่ได้จากในแบบจำลอง (มม.)			ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่วัดจริง (มม.)			ทิศทางการเคลื่อนที่
	θ_1	d_1	θ_2	θ_e	x	y	z	x	y	z	
1	-54	-24	-44	0	265.2	-187.8	75.3	264.1	-187.5	74.9	ถูกต้อง
2	-54	-24	-44	0	265.2	-187.8	75.3	265	-187.5	75.2	ถูกต้อง
3	-54	-24	-44	0	265.2	-187.8	75.3	265.2	-188.1	75.2	ถูกต้อง
4	-54	-24	-44	0	265.2	-187.8	75.3	265	-187.7	75.3	ถูกต้อง
5	-54	-24	-44	0	265.2	-187.8	75.3	265.5	-187.1	75.0	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่					265.2	-187.8	75.3	264.9	187.6	75.1	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

ตารางที่ 4.4 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า เมื่อกำหนดอินพุตที่ -70% เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่ามุมหรือระยะเลื่อนของข้อต่อที่ -30% (องศา หรือ มม.)				ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่ได้จากในแบบจำลอง (มม.)			ค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่วัดจริง (มม.)			ทิศทางการเคลื่อนที่
	θ_1	d_1	θ_2	θ_e	x	y	z	x	y	z	
1	-126	-56	-102	0	19.4	-222.8	43.3	19.3	-222.9	43.3	ถูกต้อง
2	-126	-56	-102	0	19.4	-222.8	43.3	19.3	-222.8	43.2	ถูกต้อง
3	-126	-56	-102	0	19.4	-222.8	43.3	19.1	-222.6	43.1	ถูกต้อง
4	-126	-56	-102	0	19.4	-222.8	43.3	19	-222.8	43.3	ถูกต้อง
5	-126	-56	-102	0	19.4	-222.8	43.3	19.2	-222.7	43.1	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่					19.4	-222.8	43.3	19.2	-222.7	43.2	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

4.2 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์

ในการทดลองนี้จะเป็นการควบคุมแขนกลให้เคลื่อนที่โดยกำหนดอินพุตที่ป้อนให้กับส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) นั้นเป็นค่าของพิกัดตำแหน่ง (x, y, z) ของส่วนปลายของแขนกลที่ต้องการให้แขนกลเคลื่อนที่ไป จากนั้นนำผลของการเคลื่อนที่ของข้อต่อแขนกลที่มีการเคลื่อนที่ไป นำมาเปรียบเทียบกับผลการจำลองการเคลื่อนที่ของแขนกลบนแบบจำลองทางคณิตศาสตร์ เพื่อตรวจสอบความถูกต้องในการเคลื่อนที่ จากนั้นทำการหาความสามารถในการทำซ้ำได้ของแขนกลด้วยการทดลองซ้ำในการทดลองแต่ละครั้งเป็นจำนวน 5 ครั้ง โดยเงื่อนไขในการทดลองจะกำหนดให้แขนกลมีการเคลื่อนที่ไปในจุดภาค (Quadrant) ทั้ง 4 ซึ่งประกอบไปด้วยพิกัด $+x +y$, $+x -y$, $-x +y$ และ $-x -y$ ตามลำดับ และแกน z จะกำหนดให้มีการเคลื่อนที่ตามความเหมาะสม ซึ่งจะได้ผลการทดลองเป็นดังนี้

ตารางที่ 4.5 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในพิกัด $+x +y$ และ $+z$ เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่าพิกัดตำแหน่งของส่วนปลายแขนกล (มม.)			ค่ามุมหรือระยะเลื่อนของข้อต่อที่ได้จากในแบบจำลอง (องศา หรือ มม.)				ค่ามุมหรือระยะเลื่อนของข้อต่อที่วัดได้จริง (องศา หรือ มม.)				ทิศทางการเคลื่อนที่
	x	y	z	θ_1	d_1	θ_2	θ_e	θ_1	d_1	θ_2	θ_e	
1	70	250	120	39	21	-85	0	38	21	-86	0	ถูกต้อง
2	70	250	120	39	21	-85	0	39	21	-85	0	ถูกต้อง
3	70	250	120	39	21	-85	0	39	21	-85	0	ถูกต้อง
4	70	250	120	39	21	-85	0	38	21	-84	0	ถูกต้อง
5	70	250	120	39	21	-85	0	40	21	-85	0	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่				39	21	-85	0	38.8	21	-85	0	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

ตารางที่ 4.6 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในทิศทาง $+x -y$ และ $+z$ เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่าพิกัดตำแหน่งของส่วนปลายแขนกล (มม.)			ค่ามุมหรือระยะเลื่อนของข้อต่อที่ได้จากในแบบจำลอง (องศา หรือ มม.)				ค่ามุมหรือระยะเลื่อนของข้อต่อที่วัดได้จริง (องศา หรือ มม.)				ทิศทางการเคลื่อนที่
	x	y	z	θ_1	d_1	θ_2	θ_e	θ_1	d_1	θ_2	θ_e	
1	250	-150	120	-59	21	-68	0	-59	21	-67	0	ถูกต้อง
2	250	-150	120	-59	21	-68	0	-58	22	-67	0	ถูกต้อง
3	250	-150	120	-59	21	-68	0	-59	21	-67	0	ถูกต้อง
4	250	-150	120	-59	21	-68	0	-59	21	-66	0	ถูกต้อง
5	250	-150	120	-59	21	-68	0	-58	21	-67	0	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่				-59	21	-68	0	-58.6	21.2	-66.8	0	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

ตารางที่ 4.7 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในทิศทาง $-x +y$ และ $+z$ เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่าพิกัดตำแหน่งของส่วนปลายแขนกล (มม.)			ค่ามุมหรือระยะเลื่อนของข้อต่อที่ได้จากในแบบจำลอง (องศา หรือ มม.)				ค่ามุมหรือระยะเลื่อนของข้อต่อที่วัดได้จริง (องศา หรือ มม.)				ทิศทางการเคลื่อนที่
	x	y	z	θ_1	d_1	θ_2	θ_e	θ_1	d_1	θ_2	θ_e	
1	-250	150	60	120	-39	-68	0	121	-40	-69	0	ถูกต้อง
2	-250	150	60	120	-39	-68	0	121	-39	-69	0	ถูกต้อง
3	-250	150	60	120	-39	-68	0	122	-40	-67	0	ถูกต้อง
4	-250	150	60	120	-39	-68	0	120	-40	-68	0	ถูกต้อง
5	-250	150	60	120	-39	-68	0	121	-39	-69	0	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่				120	-39	-68	0	121	-39.6	-68.4	0	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

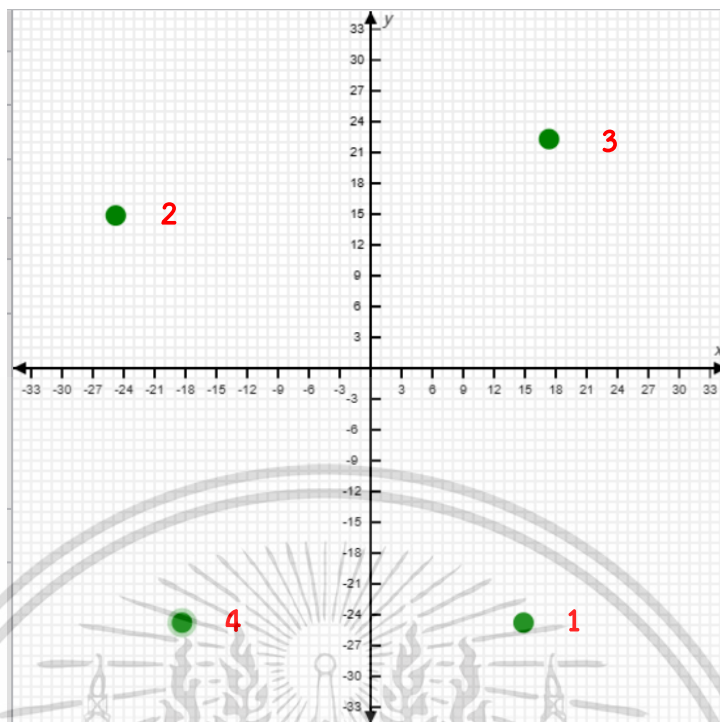
ตารางที่ 4.8 การทดลองการเคลื่อนที่ของแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เมื่อกำหนดอินพุตที่ให้เคลื่อนที่ไปในทิศทาง $-x$ $-y$ และ $+z$ เปรียบเทียบกับแบบจำลองทางคณิตศาสตร์

ครั้งที่	อินพุตค่าพิกัด ตำแหน่งของส่วน ปลายแขนกล (มม.)			ค่ามุมหรือระยะเลื่อน ของข้อต่อที่ได้จากใน แบบจำลอง (องศา หรือ มม.)				ค่ามุมหรือระยะเลื่อนของข้อ ต่อที่วัดได้จริง (องศา หรือ มม.)				ทิศ ทางการ เคลื่อน ที่
	x	y	z	θ_1	d_1	θ_2	θ_e	θ_1	d_1	θ_2	θ_e	
1	-50	-300	60	-125	-39	-60	0	-124	-39	-60	0	ถูกต้อง
2	-50	-300	60	-125	-39	-60	0	-124	-39	-60	0	ถูกต้อง
3	-50	-300	60	-125	-39	-60	0	-124	-40	-61	0	ถูกต้อง
4	-50	-300	60	-125	-39	-60	0	-125	-39	-60	0	ถูกต้อง
5	-50	-300	60	-125	-39	-60	0	-124	-40	-60	0	ถูกต้อง
ค่าเฉลี่ยในการเคลื่อนที่				-125	-39	-60	0	-124.2	39.4	-60.2	0	

หมายเหตุ : ทิศทางการเคลื่อนที่ คือลักษณะของการเคลื่อนที่ในการเคลื่อนที่จริงเปรียบเทียบกับแบบจำลอง

4.3 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่แบบจุดต่อจุด (Point to Point)

การทดลองนี้จะเป็นการควบคุมให้แขนกลมีการเคลื่อนที่แบบต่อเนื่องโดยกำหนดเงื่อนไขของการทดลองให้แขนกลมีการเคลื่อนที่จากจุดเริ่มต้นไปยังพิกัดตำแหน่งเป้าหมายจุดที่ 1 จากนั้นทำการป้อนอินพุตอีกครั้งเพื่อให้แขนกลเคลื่อนที่ไปยังจุดที่ 2 และเคลื่อนที่จากจุดที่ 2 ไปยังจุดที่ 3 ตามลำดับ ซึ่งแขนกลจะมีการเคลื่อนที่แบบจุดต่อจุดไปจนถึงพิกัดตำแหน่งเป้าหมายสุดท้าย โดยในการทดลองนี้จะกำหนดให้มีการเคลื่อนที่ 4 จุด โดยการเคลื่อนที่ไปในแต่ละจุดจะถูกจำลองด้วยแบบจำลองทางคณิตศาสตร์เพื่อที่จะตรวจสอบความถูกต้องในการเคลื่อนที่



รูปที่ 4.1 แสดงพิกัดตำแหน่งเป้าหมายทั้ง 4 จุดบนตารางจตุภาคทั้ง 4

ตารางที่ 4.9 การทดลองการควบคุมการเคลื่อนที่ของแขนกลแบบจุดต่อจุด

ครั้งที่	พิกัดตำแหน่ง (x,y,z) ของส่วนปลายแขนกล ตำแหน่งที่ 1 (150,-250,45)		พิกัดตำแหน่ง (x,y,z) ของส่วนปลายแขนกล ตำแหน่งที่ 2 (-250,150,60)		พิกัดตำแหน่ง (x,y,z) ของส่วนปลายแขนกล ตำแหน่งที่ 3 (175,225,45)		พิกัดตำแหน่ง (x,y,z) ของส่วนปลายแขนกล ตำแหน่งที่ 4 (-185,-250,35)	
	ค่าจำลอง	ค่าจริง	ค่าจำลอง	ค่าจริง	ค่าจำลอง	ค่าจริง	ค่าจำลอง	ค่าจริง
1	$X_{s1} = 150$ $Y_{s1} = -250$ $Z_{s1} = 45$	$X = 146$ $Y = -256$ $Z = 46$	$X_{s2} = -250$ $Y_{s2} = 150$ $Z_{s2} = 60$	$X = -250$ $Y = 145$ $Z = 60$	$X_{s3} = 175$ $Y_{s3} = 225$ $Z_{s3} = 45$	$X = 168$ $Y = 227$ $Z = 45$	$X_{s4} = -185$ $Y_{s4} = -250$ $Z_{s4} = 35$	$X = -185$ $Y = -249$ $Z = 35$
2	$X_{s1} = 150$ $Y_{s1} = -250$ $Z_{s1} = 45$	$X = 142$ $Y = -256$ $Z = 45$	$X_{s2} = -250$ $Y_{s2} = 150$ $Z_{s2} = 60$	$X = -250$ $Y = 145$ $Z = 60$	$X_{s3} = 175$ $Y_{s3} = 225$ $Z_{s3} = 45$	$X = 180$ $Y = 224$ $Z = 45$	$X_{s4} = -185$ $Y_{s4} = -250$ $Z_{s4} = 35$	$X = -179$ $Y = -254$ $Z = 35$
3	$X_{s1} = 150$ $Y_{s1} = -250$ $Z_{s1} = 45$	$X = 147$ $Y = -251$ $Z = 45$	$X_{s2} = -250$ $Y_{s2} = 150$ $Z_{s2} = 60$	$X = -251$ $Y = 150$ $Z = 60$	$X_{s3} = 175$ $Y_{s3} = 225$ $Z_{s3} = 45$	$X = 174$ $Y = 224$ $Z = 45$	$X_{s4} = -185$ $Y_{s4} = -250$ $Z_{s4} = 35$	$X = -179$ $Y = -254$ $Z = 35$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 แสดงค่าเฉลี่ยในการเคลื่อนที่จริง และค่าเปอร์เซ็นต์ความผิดพลาดในการเคลื่อนที่ไปในแต่ละตำแหน่ง

	ตำแหน่งที่ 1 (150,-250,45)	ตำแหน่งที่ 2 (-250,150,60)	ตำแหน่งที่ 3 (175,225,45)	ตำแหน่งที่ 4 (-185,-250,35)
ค่าเฉลี่ยการเคลื่อนที่	X = 145 Y = -254.33 Z = 45.33	X = -250.33 Y = 146.66 Z = 60.17	X = 174 Y = 225 Z = 45	X = -181 Y = -252.33 Z = 34.67
ค่า % ความผิดพลาด ของการเคลื่อนที่ สะสม	X = 3.333% Y = 1.732% Z = 0.733%	X = 0.132% Y = 2.226% Z = 0.283%	X = 0.571% Y = 0% Z = 0%	X = 2.162% Y = 0.932% Z = 0.943%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ปริญญานิพนธ์ฉบับนี้ได้ทำการประยุกต์ PLC เพื่อควบคุมแขนกล SCARA Robot 4DOF ให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ ซึ่งได้มีการออกแบบโปรแกรมการทำงานซึ่งมีการรับค่าข้อมูลอินพุตจากผู้ใช้งาน จากนั้นมาทำการคำนวณเพื่อหาตำแหน่งของส่วนปลายแขนกล หรือค่ามุมของข้อต่อแขนกล ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ โดยค่าที่ได้ก็จะถูกนำมาคำนวณเพื่อหาค่าของสัญญาณพัลส์และความเร็วของมอเตอร์ เพื่อนำไปขับมอเตอร์ให้แขนกลเกิดการเคลื่อนที่ไปยังตำแหน่งเป้าหมายตามที่กำหนด และจากการทดลองในบทที่ 4 ได้มีการทำการทดสอบและหาประสิทธิภาพของการประยุกต์ PLC ในการควบคุมแขนกล โดยใช้วิธีในการตรวจสอบความถูกต้องของแขนกลที่การเคลื่อนที่ไปยังตำแหน่งใด ๆ ด้วยการเปรียบเทียบกับผลการจำลองการเคลื่อนที่บนแบบจำลองทางคณิตศาสตร์ ซึ่งจากตารางการทดลองที่ 4.1 ถึงตารางที่ 4.4 เป็นการควบคุมแขนกลให้มีการเคลื่อนที่โดยการป้อนอินพุตเป็นค่ามุมหรือระยะเลื่อนของข้อต่อ และให้แขนกลเคลื่อนที่ไปยังพิกัดตำแหน่งที่ต้องการ ส่วนตารางการทดลองที่ 4.5 ถึงตารางที่ 4.8 จะควบคุมแขนกลเคลื่อนที่ไปยังพิกัดตำแหน่งที่ต้องการโดยการป้อนค่าของพิกัดตำแหน่งเป้าหมายเป็นอินพุต และให้ข้อต่อในแต่ละแกนเคลื่อนที่ไปเพื่อให้แขนกลไปอยู่ ณ ตำแหน่งเป้าหมาย โดยจากการทดลองทั้งหมดนั้น ได้มีการทำการทดลองซ้ำ ๆ ในแต่ละการทดลอง พบว่าในการควบคุมแขนกลให้มีการเคลื่อนที่จริงมีความถูกต้องและได้ค่าเอาต์พุตที่ใกล้เคียงกับการทำการจำลองการเคลื่อนที่บนแบบจำลองทางคณิตศาสตร์ อีกทั้งยังได้มีการทดลองในส่วนของการควบคุมการเคลื่อนที่แบบจุดต่อจุด (Point to Point) ซึ่งเป็นการเคลื่อนที่แบบอัตโนมัติแทนการใช้งานจริงของแขนกลในอุตสาหกรรม ซึ่งพบว่าแขนกลได้มีการเคลื่อนที่ไปยังพิกัดตำแหน่งเป้าหมายได้อย่างถูกต้อง และมีข้อผิดพลาดเพียงเล็กน้อย ทำให้สามารถสรุปได้ว่าการประยุกต์ใช้ PLC ควบคุมแขนกล SCARA Robot 4DOF ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับนั้นสามารถบรรลุตามวัตถุประสงค์ตามที่ได้มีการกำหนดไว้

5.2 ปัญหาและอุปสรรคที่พบในระหว่างดำเนินการ

1) ปัญหาเกี่ยวกับการคำนวณหาความเร็วของมอเตอร์ที่จะทำให้มอเตอร์มีการเคลื่อนที่ไปถึงตำแหน่งเป้าหมายได้พร้อมกันสำหรับการเคลื่อนที่แบบจุดต่อจุด ซึ่งข้อต่อของแขนกลนั้นจะสามารถเคลื่อนที่ไปถึงได้พร้อมกันเมื่อการเคลื่อนที่ที่มีการเริ่มต้นจากจุดเริ่มไปยังจุดที่ 1 ในขณะที่การเคลื่อนที่จากจุดที่ 1 ไปยังจุดที่ 2 นั้นในบางครั้งไม่ข้อต่อไม่สามารถเคลื่อนที่ไปได้อย่างพร้อม ๆ กัน ถึงแม้ว่าจะทำการนำค่าของ PRV นำมาลบกับค่า Pulse Output เพื่อหาข้อต่อที่มีการเคลื่อนที่ที่ใกล้เคียงที่สุด เพื่อนำมาเป็นตัวตั้งสำหรับการคำนวณหาความเร็วแล้วก็ตาม

2) ในช่วงแรกของการเชื่อมต่อระหว่าง Microsoft Excel เข้ากับ Data Memory ของ PLC ผ่าน KEPServerEX6 ซึ่งได้มีการกำหนดชนิดของข้อมูลในการรับส่งระหว่างกันเป็นชนิด DWORD ซึ่งเมื่อนำมาใช้สำหรับการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลนั้นจะเกิดความยุ่งยากที่จะต้องมีการแปลงชนิดของข้อมูลจาก DWORD ให้กลายเป็น FLOATING เพื่อนำไปคำนวณ ซึ่งจะทำให้การเขียนโปรแกรมมีความยาวขึ้น ดังนั้นจึงมีการแก้ไขในการเปลี่ยนให้มีการรับส่งด้วย FLOATING โดยตรง

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

แนวทางในการพัฒนาสำหรับการประยุกต์ PLC ควบคุม SCARA Robot 4DOF ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ ทางผู้จัดทำได้เล็งเห็นว่าสามารถเพิ่มในส่วนของมือจับชิ้นงานหรือ Gripper ที่ใช้สำหรับการหยิบวัสดุ หรือ ชิ้นงานจากจุดหนึ่งไปวางในอีกจุดหนึ่ง เพื่อให้การทำงานของแขนกลนั้นมีความใกล้เคียงกับแขนกลที่มีการทำงานในอุตสาหกรรมที่มากขึ้น และการเพิ่มส่วนของ Robot Vision เพื่อเข้ามาช่วยเป็นอุปกรณ์สำหรับการตรวจสอบกลับในการเคลื่อนที่ของแขนกลไปยังตำแหน่งเป้าหมายในตำแหน่งต่าง ๆ ซึ่งจะช่วยให้การควบคุมแขนกลมีประสิทธิภาพ และมีความถูกต้องแม่นยำที่มากขึ้น นอกจากนั้นในปริญญานิพนธ์นี้ได้มีการใช้งาน OPC Server ในการรับส่งชุดข้อมูลของพิกัดตำแหน่งในหลาย ๆ ชุด เพื่อใช้สำหรับการเคลื่อนที่แบบจุดต่อจุด ซึ่งสามารถนำไปต่อยอดเพื่อทำเป็นระบบ SCADA ในการควบคุมแขนกล เพื่อนำไปใช้งานสำหรับงานที่มีการเคลื่อนที่ไปในหลาย ๆ ตำแหน่ง เช่น การเขียนภาพอักษร 2 มิติ เป็นต้น

บรรณานุกรม

- [1] วโรดม ตู๋จินดา, “การวิเคราะห์และควบคุมหุ่นยนต์อุตสาหกรรม”, กรุงเทพฯ:ศูนย์หนังสือแห่งจุฬาลงกรณ์มหาวิทยาลัย, 2559.
- [2] Ahmad Albalasie, Irfan Hussain, Mamoun Horoub, Sikandar Khan, Sajid Ali and Dongming Gan, “Design, Prototype, and Control Design Based on Computed Torque Control of Selective Compliance Assembly Robot Arm”, IEEE International Conference, Vol.9, pp.70-75, 2019.
- [3] พิษณุ ตันทกิจ, พัฒนศักดิ์ ศรีสกุล และอชวิน ถาวรวงษ์, “การค้นหาวัดจุดด้วยกล้อง Smart Camera ในระนาบแบบ 2 มิติบนเครื่อง CNC และจำแนกไปยังพื้นที่จัดเก็บแบบอัตโนมัติด้วยเครื่องควบคุม PLC และรายงานตำแหน่งวัดจุดด้วยจอ HMI”, ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2564.
- [4] จิรายุ เพชรแทน, “การควบคุมวีลแชร์ด้วยตัวควบคุมที่โปรแกรมได้ด้วยการบูรณาการเข้ากับอินเตอร์เน็ตของสรรพสิ่ง”, ปริญญานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมการวัดคุม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2562.
- [5] “การควบคุมสเต็ปมอเตอร์”, 2561. [ระบบออนไลน์] แหล่งที่มา: <http://suwitkiravittaya.eng.chula.ac.th/B2i2019BookWeb/stepmotor.html> (12 ตุลาคม 2565).
- [6] ศูนย์ฝึกอบรมไฟฟ้าและอิเล็กทรอนิกส์ แอดวานด์ อิเล็กทรอนิกส์ เทรนนิ่ง เซ็นเตอร์, “สเต็ปมอเตอร์”, 2560. [ระบบออนไลน์] แหล่งที่มา: <http://www.advance-electronic.com/about-us> (24 ตุลาคม 2565).
- [7] “TB6600 Stepper Motor Driver User Guide”, 2560. [ระบบออนไลน์] แหล่งที่มา: <https://www.makerguides.com/wp-content/uploads/2019/10/TB6600-Manual.pdf> (24 ตุลาคม 2565).
- [8] MISUMI, “วงจรรีบ (Driver IC)”, 2560. [ระบบออนไลน์] แหล่งที่มา: https://th.misumiec.com/th/vona2/el_control/E2400000000/E2409000000/ (9 พฤศจิกายน 2565).
- [9] บริษัท แฟคโตมาร์ท จำกัด, “Photoelectric Sensor โฟโตอิเล็กทริก เซ็นเซอร์”, 2563. [ระบบออนไลน์] แหล่งที่มา: <https://mall.factomart.com/what-is-photoelectric-sensor/> (12 พฤศจิกายน 2565).
- [10] SUNX Panasonic, “Small U-shaped Micro Photoelectric Sensor Amplifier Built-in PM-44 SERIES PM-54 SERIES”, 2560. [ระบบออนไลน์] แหล่งที่มา: https://www.sentronic/data/product_datasheets/PAN-DAT__PM54_PM44_PM24_PM64-EN.pdf (9 พฤศจิกายน 2565).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



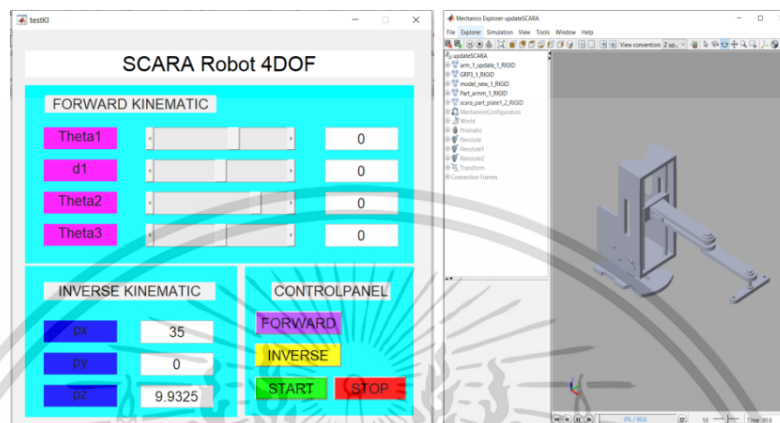
ภาคผนวก ก

แสดงคู่มือการใช้งานแบบจำลองทางคณิตศาสตร์และ HMI

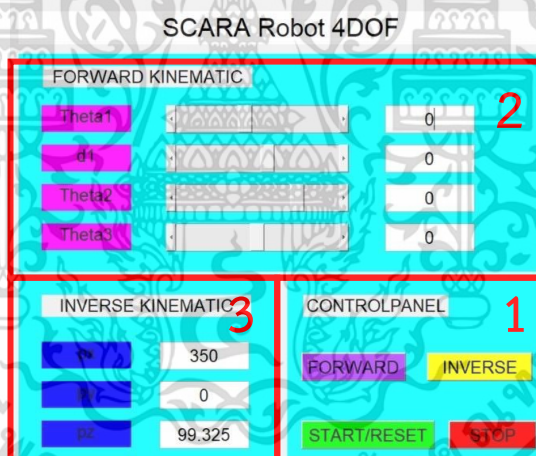
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.1 คู่มือการใช้งานแบบจำลองทางคณิตศาสตร์

แบบจำลองทางคณิตศาสตร์มีส่วนประกอบแบ่งออกเป็น 2 ส่วน ซึ่งประกอบไปด้วยส่วนที่ใช้สำหรับแสดงผลในการจำลองการเคลื่อนที่ของโมเดลแขนกลจำลอง และส่วนที่เป็น Graphical user interface หรือ GUI สำหรับให้ผู้ใช้ป้อนค่าและแสดงผลดังรูปที่ ก-1



รูปที่ ก-1 แบบจำลองทางคณิตศาสตร์



รูปที่ ก-2 หน้าต่าง GUI สำหรับผู้ใช้งาน

โดยในส่วนของ GUI สำหรับผู้ใช้งานจะประกอบไปด้วย 3 ส่วน ซึ่งได้แก่

- 1) CONTROL PANEL ซึ่งเป็นส่วนสำหรับปุ่มควบคุมต่าง ๆ ทั้งหมดสำหรับแบบจำลองทางคณิตศาสตร์ โดยมีปุ่ม START/RESET มีไว้ใช้สำหรับการกดเพื่อเริ่มใช้งาน และรีเซ็ตแบบจำลองให้กลับสู่สภาพเริ่มต้น, ปุ่ม STOP มีไว้สำหรับปิดโปรแกรม และปุ่มสำหรับการคำนวณค่าจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ
- 2) FORWARD KINEMATIC เป็นส่วนสำหรับจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า ซึ่งจะมีชื่อของข้อต่อต่าง ๆ มีแถบเลื่อน ซึ่งใช้แทน Encoder ในการควบคุมด้วยระบบ Manual ในการปรับการเคลื่อนที่ของข้อต่อแต่ละแกน และช่องสำหรับกำหนดค่า

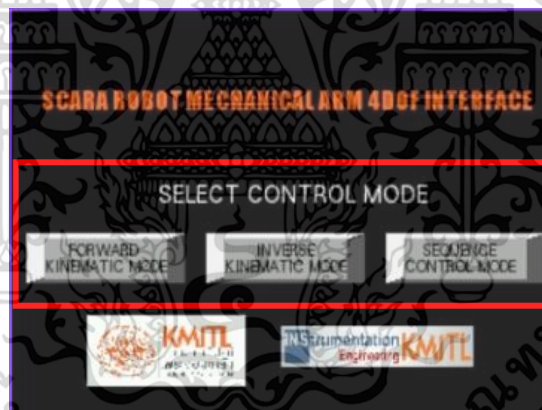
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของมุมหรือระยะเลื่อนของข้อต่อที่ให้เคลื่อนที่ไป ซึ่งช่องนี้จะใช้สำหรับการแสดงค่าที่คำนวณได้จากจลนศาสตร์การเคลื่อนที่แบบย้อนกลับเช่นกัน

- 3) INVERSE KINEMATIC เป็นส่วนสำหรับจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ ซึ่งจะแสดงค่าพิกัดตำแหน่งของส่วนปลายแขนกล และช่องสำหรับใส่ค่าพิกัดที่จะกำหนดให้แขนกลมีการเคลื่อนที่ไป

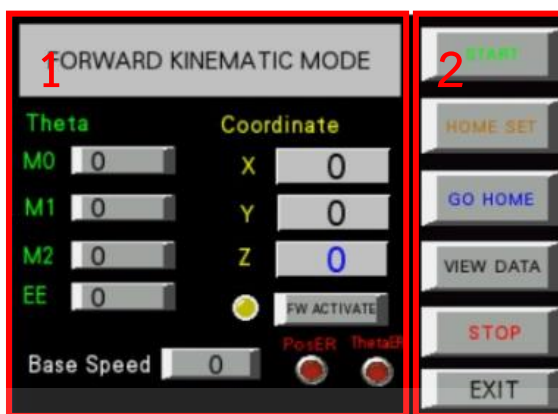
ก.2 คู่มือการใช้งานส่วนเชื่อมต่อกับผู้ใช้งาน (Human machine interface หรือ HMI)

ในหน้าแรกของส่วนเชื่อมต่อกับผู้ใช้งานจะแสดงโหมดในการควบคุมสำหรับให้ผู้ใช้งานได้เลือกในการควบคุมแขนกล SCARA Robot 4DOF ให้มีการเคลื่อนที่ตามที่กำหนด ซึ่งจะประกอบด้วย 3 ตัวเลือก ได้แก่ FORWARD KINEMATIC MODE หรือการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า, INVERSE KINEMATIC MODE หรือการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ และ SEQUENCE CONTROL MODE เป็นการควบคุมแขนกลให้มีการเคลื่อนที่ไปยังตำแหน่งใด ๆ โดยอัตโนมัติ



รูปที่ ก-3 หน้าแรกของส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) และตัวเลือกในการควบคุม

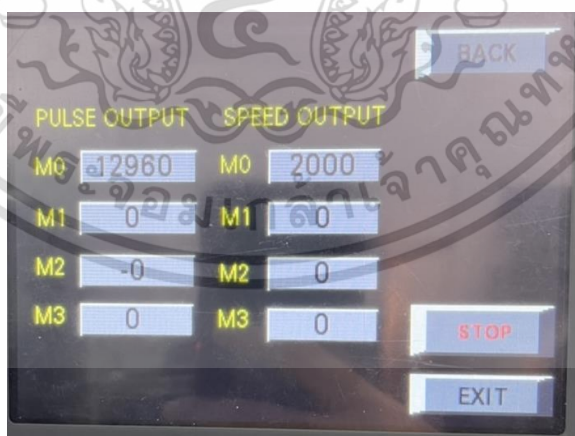
เมื่อทำการเลือกตัวเลือกของโหมดในการควบคุมแขนกลโหมดใดโหมดหนึ่งแล้วจะเข้าสู่หน้าต่างของการควบคุม ซึ่งจะแบ่งออกเป็น 2 ส่วน โดยส่วนแรกจะอยู่ทางด้านซ้ายเป็นส่วนสำหรับการป้อนค่าตำแหน่งเป้าหมาย และแสดงผลค่าที่ได้จากการคำนวณตามจลนศาสตร์การเคลื่อนที่ของแขนกล ส่วนที่สองจะอยู่ทางด้านขวาเป็นปุ่มสำหรับควบคุมการทำงาน



รูปที่ ก-4 ตัวอย่างของหน้าต่างควบคุมการเคลื่อนที่ของแขนกลโดย 1 คือส่วนป้อนค่าและแสดงผล 2 คือส่วนสำหรับปุ่มควบคุม

โดยส่วนสำหรับปุ่มควบคุมจะประกอบไปด้วยปุ่มทั้งหมด 6 ปุ่มซึ่งได้แก่

- 1) START เป็นปุ่มสำหรับกดเพื่อให้โปรแกรมเริ่มทำงาน และให้แขนกลเคลื่อนที่ไปตามการควบคุมที่ป้อน
- 2) HOME SET เป็นปุ่มสำหรับตั้งจุดกำเนิด หรือจุดเริ่มต้นของแขนกล
- 3) GO HOME เป็นปุ่มที่ใช้สำหรับกดให้แขนกลเคลื่อนที่จากตำแหน่งเป้าหมาย กลับมายังจุดเริ่มต้น
- 4) VIEW DATA เป็นปุ่มที่ใช้สำหรับการดูสัญญาณเอาต์พุตที่ออกจาก PLC ซึ่งเป็นสัญญาณพัลส์ที่ได้จากการคำนวณที่ส่งออกไปยังวงจรขับเคลื่อนมอเตอร์เพื่อให้อุปกรณ์เกิดการเคลื่อนที่ และเคลื่อนข้อต่อของแขนกลไปยังตำแหน่งที่กำหนด

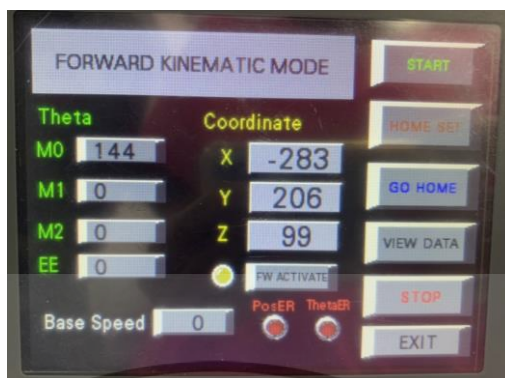


รูปที่ ก-5 หน้าต่าง VIEW DATA

- 5) STOP เป็นปุ่มสำหรับการกดเพื่อยุติการทำงานของโปรแกรมทั้งหมด
- 6) EXIT เป็นปุ่มที่ใช้กดเพื่อกลับไปยังหน้าแรกของ HMI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2.1 การใช้งานการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า

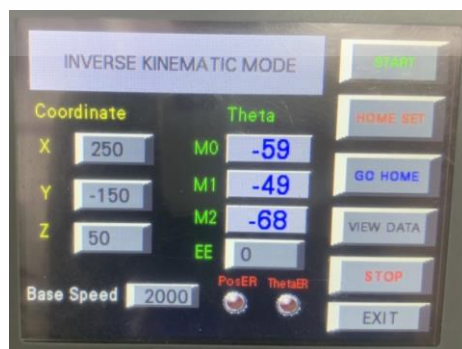


รูปที่ ก-6 การใช้งาน FORWARD KINEMATIC MODE

ในส่วนนี้จะแบ่งออกเป็น 4 ส่วนซึ่งทางด้านซ้ายจะเป็นส่วนที่มีไว้สำหรับให้ผู้ใช้ในการป้อนค่ามุมหรือระยะเลื่อนของข้อต่อแต่ละแกน และทางด้านขวาจะเป็นส่วนแสดงผลค่าตำแหน่งที่ได้จากการคำนวณ ส่วนด้านล่างจะเป็นส่วนสำหรับการป้อนค่าความเร็วเริ่มต้นที่จะให้มอเตอร์เคลื่อนที่ไป และส่วนสุดท้ายคือส่วนที่ใช้แสดงสถานะ Error เมื่อตำแหน่งเป้าหมายมีค่าไม่อยู่ในพื้นที่การทำงานของแขนกล โดยสามารถอธิบายการใช้งานเบื้องต้นได้ดังนี้

- 1) ทำกดปุ่ม FW ACTIVATE เพื่อเปิดใช้งานโหมดการเคลื่อนที่ด้วยจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า
- 2) ป้อนค่ามุมหรือระยะเลื่อนของข้อต่อที่ต้องการให้มีการเคลื่อนที่ไป
- 3) ป้อนค่าความเร็วเริ่มต้น
- 4) หากไม่มีสถานะ Error ขึ้นแสดง ให้กดปุ่ม START เพื่อให้แขนกลเคลื่อนที่ไปยังตำแหน่งเป้าหมาย และค่าพิกัดตำแหน่งของส่วนปลายแขนกลจะแสดงขึ้นที่ด้านขวาของหน้าจอ

ก.2.2 การใช้งานการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ



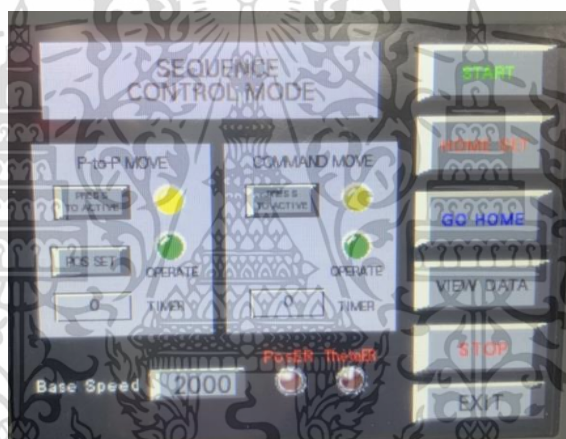
รูปที่ ก-7 การใช้งาน INVERSE KINEMATIC MODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการควบคุมด้วยจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับนั้นจะคล้ายคลึงกับแบบไปข้างหน้า แต่แตกต่างกันตรงที่ข้อมูลที่ป้อนให้ทางด้านซ้ายจะเป็นค่าของพิกัดส่วนปลายของแขนกลที่ต้องการให้เคลื่อนที่ไป และค่าทางด้านขวาจะแสดงค่ามุมหรือระยะเลื่อนของข้อต่อที่คำนวณออกมาได้โดยสามารถอธิบายการใช้งานเบื้องต้นได้ดังนี้

- 1) ป้อนค่าพิกัดตำแหน่งของส่วนปลายของแขนกลที่ต้องการให้เคลื่อนที่ไป
- 2) ป้อนค่าความเร็วเริ่มต้น
- 3) หากไม่มีสถานะ Error ขึ้นแสดง ให้กดปุ่ม START เพื่อให้แขนกลเคลื่อนที่ไปยังตำแหน่งเป้าหมาย และค่ามุมหรือระยะเลื่อนของข้อต่อที่เคลื่อนที่เพื่อไปยังพิกัดตำแหน่งเป้าหมายจะแสดงขึ้นที่ด้านขวาของหน้าจอ

ก.2.3 การใช้งานการควบคุมแบบอัตโนมัติ



รูปที่ ก-8 การใช้งาน SEQUENCE CONTROL MODE

การควบคุมการเคลื่อนที่แบบอัตโนมัติจะแบ่งลักษณะการทำงานออกเป็น 2 ส่วนคือการเคลื่อนที่แบบจุดต่อจุด หรือ Point to Point และการเคลื่อนที่แบบ Command Move หรือการเคลื่อนที่โดยใช้ชุดข้อมูลที่เป็นพิกัดตำแหน่ง x, y, z จำนวนหลายตำแหน่ง ซึ่งในการใช้งานจะต้องมีการกด PRESS TO ACTIVATE เพื่อเลือกลักษณะการทำงานก่อน

สำหรับการทำงานในลักษณะของการเคลื่อนที่แบบจุดต่อจุดนั้นจะต้องมีการป้อนค่าตำแหน่งของจุดที่จะให้แขนกลมีการเคลื่อนที่ไปในหน้าต่าง POS SETTING



รูปที่ ก-9 การป้อนค่าพิกัดตำแหน่งเป้าหมายใน POS SETTING

และสำหรับการควบคุมในลักษณะของ Command Move จะต้องมีการใส่ข้อมูลลงในไฟล์ Microsoft Excel ที่ได้จัดทำขึ้นและเชื่อมต่อเข้ากับ PLC เพื่อให้สามารถดึงค่าข้อมูลมาใช้งานได้

	A	B	C
1	X	Y	Z
2			
3	-50	-275	30
4	-65	250	60
5	125	-250	-100
6	-150	250	120
7	0	0	0
8	0	0	0

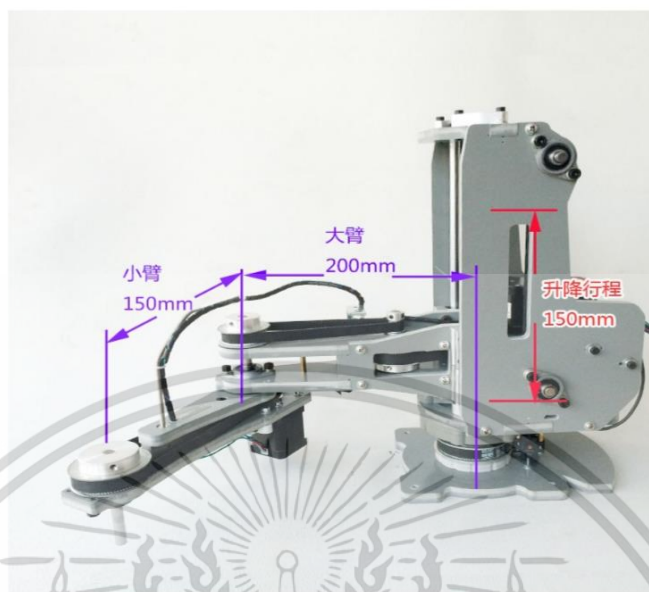
รูปที่ ก-10 การป้อนค่าพิกัดตำแหน่งเป้าหมายใน Microsoft Excel

ซึ่งเมื่อทำการป้อนค่าพิกัดตำแหน่งเป้าหมาย และเลือกลักษณะการทำงานแล้ว จากนั้นกดปุ่ม START ก็จะเป็นเริ่มต้นการทำงานแบบอัตโนมัติ โดยที่จะมีไฟสถานะติดเมื่อเริ่มทำงาน และไฟสถานะจะดับลงเมื่อแขนกลเคลื่อนที่กลับจุดเริ่มต้นจากตำแหน่งสุดท้าย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

四轴 scara 机械臂参数说明



臂长和升降行程 (图 1-2)

参数:

项目	参数	备注
动力	42 步进电机	四轴均使用 42 步进电机, 详细参数见表 1-1
负重	机械臂末端负重 2kg	含执行器的重量
旋转范围	大臂旋转 360° 小臂旋转 0°~290° 腕部旋转 360°+	见图 1-1
臂长	大臂: 200mm 小臂: 150mm Z 轴升降 0~150mm	见图 1-2
材质	主要板材为 PVC 工程塑料	
重量	全套机械臂 7.5kg	不含包装重量
外尺寸	长 500mm*宽 180mm*高 350mm	

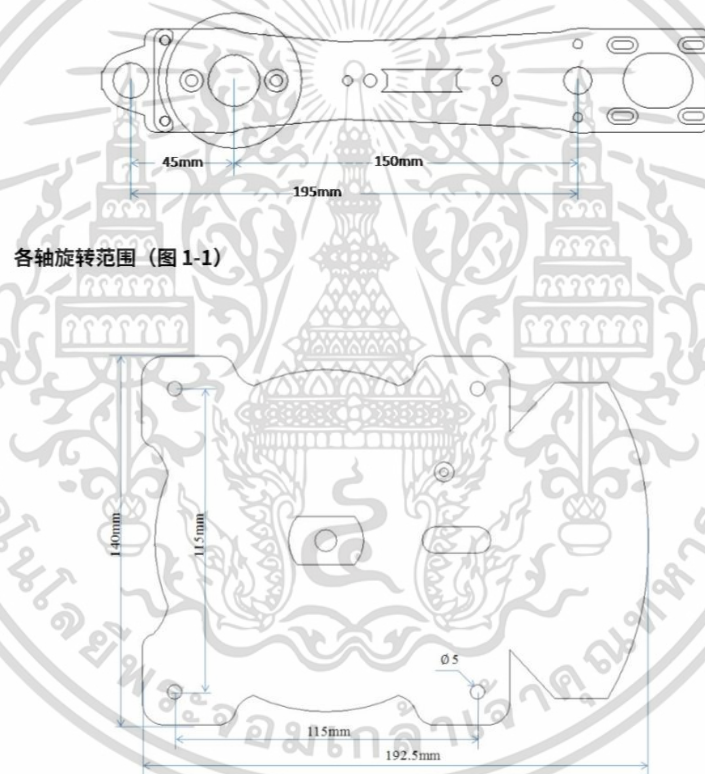
รูปที่ ข-1 แสดงคุณสมบัติทั่วไปทางเทคนิคของแขนกล SCARA Robot 4DOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

步进电机参数：（表 1-1）

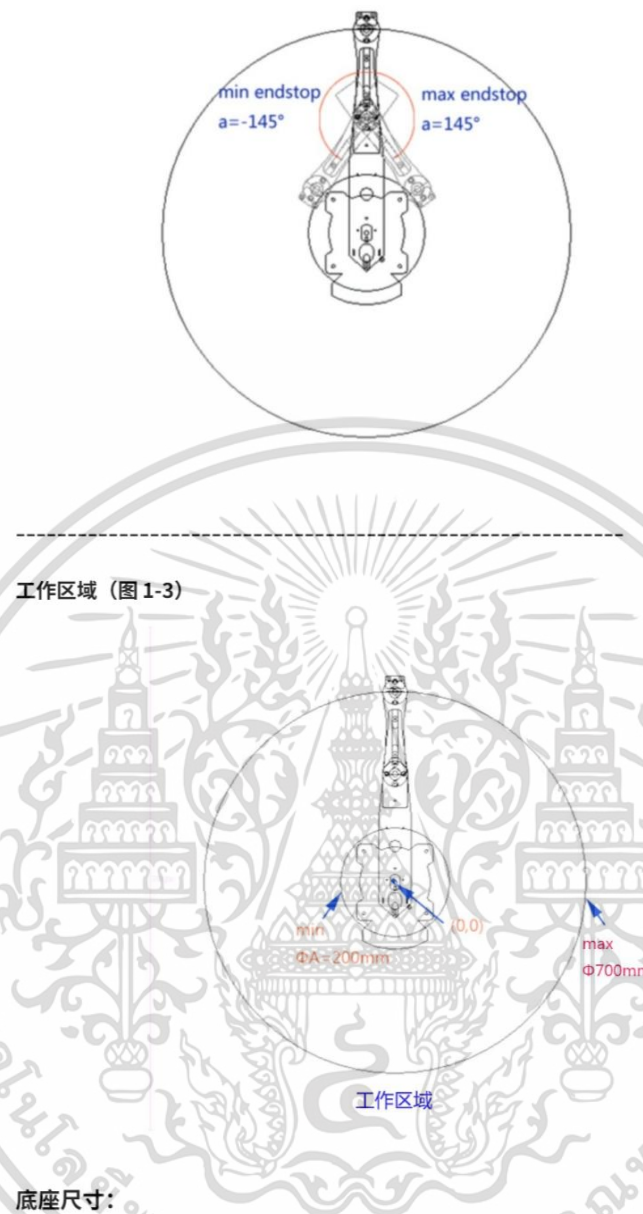
42 步进电机参数：
 型号：42BYGH47-401A 两项混合式步进电机
 最大驱动电流：1.5A
 驱动电压：24V
 引线数：4
 步距角：1.8 度
 表面温度：80°C MAX (额定电流下)
 最大静力矩：0.55N.M 相电阻：1.6Ω
 红、蓝为一组，黑、绿为一组

小臂尺寸：



รูปที่ ข-2 แสดงคุณสมบัติทั่วไปทางเทคนิคของแขนกล SCARA Robot 4DOF (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-3 แสดงคุณสมบัติทั่วไปทางเทคนิคของแขนกล SCARA Robot 4DOF (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP1H

General Specifications

Item	Type Model	AC power supply models	DC power supply models
		CP1H-□□□-A	CP1H-□□□-D
Power supply		100 to 240 VAC 50/60 Hz	24 VDC
Operating voltage range		85 264 VAC	20.4 to 26.4 VDC (with 4 or more Expansion Units and Expansion I/O Units: 21.6 to 26.4 VDC)
Power consumption		100 VA max. (CP1H-□□□-A)(page 28)	50 W max. (CP1H-□□□-D)(page 28)
Inrush current (See note.)		100 to 120 VAC inputs: 20 A max. (for cold start at room temperature) 8 ms max. 200 to 240 VAC inputs: 40 A max. (for cold start at room temperature), 8 ms max.	30 A max. (for cold start at room temperature) 20 ms max.
External power supply		300 mA at 24 VDC	None
Insulation resistance		20 MΩ min. (at 500 VDC) between the external AC terminals and GR terminals	No insulation between primary and secondary for DC power supply
Dielectric strength		2,300 VAC at 50/60 Hz for 1 min between the external AC and GR terminals, leakage current: 5 mA max.	No insulation between primary and secondary for DC power supply
Noise immunity		Conforms to IEC 61000-4-4, 2 kV (power supply line)	
Vibration resistance		Conforms to JIS C60068-2-6, 10 to 57 Hz, 0.075-mm amplitude, 57 to 150 Hz, acceleration: 9.8 m/s ² in X, Y, and Z directions for 80 minutes each. Sweep time: 8 minutes × 10 sweeps = total time of 80 minutes	
Shock resistance		Conforms to JIS C60068-2-27, 147 m/s ² three times each in X, Y, and Z directions	
Ambient operating temperature		0 to 55°C	
Ambient humidity		10% to 90% (with no condensation)	
Ambient operating environment		No corrosive gas	
Ambient storage temperature		-20 to 75°C (Excluding battery.)	
Power holding time		10 ms min.	2 ms min.

Note: The above values are for a cold start at room temperature for an AC power supply, and for a cold start for a DC power supply.

- A thermistor (with low-temperature current suppression characteristics) is used in the inrush current control circuitry for the AC power supply. The thermistor will not be sufficiently cooled if the ambient temperature is high or if a hot start is performed when the power supply has been OFF for only a short time. In those cases the inrush current values may be higher (as much as two times higher) than those shown above. Always allow for this when selecting fuses and breakers for external circuits.
- A capacitor charge-type delay circuit is used in the inrush current control circuitry for the DC power supply. The capacitor will not be charged if a hot start is performed when the power supply has been OFF for only a short time, so in those cases the inrush current values may be higher (as much as two times higher) than those shown above.

รูปที่ ข-4 แสดงคุณสมบัติทั่วไปทางเทคนิคของ PLC CP1H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP1H

Performance Specifications

Item	Type Models	CP1H-XA CPU Units	CP1H-X CPU Units	CP1H-Y CPU Units
		CP1H-XA-□□□□	CP1H-X-□□□□	CP1H-Y-□□□□
Control method		Stored program method		
I/O control method		Cyclic scan with immediate refreshing		
Program language		Ladder diagram		
Function blocks		Maximum number of function block definitions: 128 Maximum number of instances: 256 Languages usable in function block definitions: Ladder diagrams, structured text (ST)		
Instruction length		1 to 7 steps per instruction		
Instructions		Approx. 500 (function codes: 3 digits)		
Instruction execution time		Basic instructions: 0.10 μs min. Special instructions: 0.15 μs min.		
Common processing time		0.7 ms		
Program capacity		20K steps		
Number of tasks		288 (32 cyclic tasks and 256 interrupt tasks)		
	Scheduled interrupt tasks	1 (interrupt task No. 2, fixed)		
	Input interrupt tasks	8 (interrupt task No. 140 to 147, fixed)		6 (interrupt task No. 140 to 145, fixed)
		(Interrupt tasks can also be specified and executed for high-speed counter interrupts.)		
Maximum subroutine number		256		
Maximum jump number		256		
I/O areas (See note.)	Input bits	272bits (17 words) : CIO 0.00 to 16.15		
	Output bits	272bits (17 words) : CIO 100.00 to 116.16		
	Built-in Analog Inputs	CIO 200 to CIO 203		---
	Built-in Analog Outputs	CIO 210 to CIO 211		---
	Serial PLC Link Area	1,440 bits (90 words): CIO 3100.00 to CIO 3189.15 (CIO 3100 to CIO 3189)		
Work bits		8,192 bits (512 words): W0.00 to W511.15 (W0 to W511) CIO Area: 37,504 bits (2,344 words): CIO 3800.00 to CIO 6143.15 (CIO 3800 to CIO 6143)		
TR Area		16 bits: TR0 to TR15		
Holding Area		8,192 bits (512 words): H0.00 to H511.15 (H0 to H511)		
AR Area		Read-only (Write-prohibited): 7168 bits (448 words): A0.00 to A447.15 (A0 to A447) Read/Write: 8192 bits (512 words): A448.00 to A959.15 (A448 to A959)		
Timers		4,096 bits: T0 to T4095		
Counters		4,096 bits: C0 to C4095		
DM Area		32 Kwords: D0 to D32767		
Data Register Area		16 registers (16 bits): DR0 to DR15		
Index Register Area		16 registers (32 bits): IR0 to IR15		
Task Flag Area		32 flags (32 bits): TK0000 to TK0031		
Trace Memory		4,000 words (500 samples for the trace data maximum of 31 bits and 6 words.)		
Memory Cassette		A special Memory Cassette (CP1W-ME05M) can be mounted. Note: Can be used for program backups and auto-backing.		
Clock function		Supported. Accuracy (monthly deviation): -4.5 min to -0.5 min (ambient temperature: 55°C), -2.0 min to +2.0 min (ambient temperature: 25°C), -2.5 min to +1.5 min (ambient temperature: 0°C)		
Communications functions		One built-in peripheral port (USB 1.1): For connecting Support Software only. A maximum of two Serial Communications Option Boards can be mounted. A maximum of two Ethernet Option Boards can be mounted. When using CP1W-CIF41 Ver.1.0, one Ethernet Option Board can be mounted.		
Memory backup		Flash memory: User programs, parameters (such as the PLC Setup), comment data, and the entire DM Area can be saved to flash memory as initial values. Battery backup: The Holding Area, DM Area, and counter values (flags, PV) are backed up by a battery.		
Battery service life		5 years at 25°C. (Use the replacement battery within two years of manufacture.)		
Built-in input terminals		40 (24 inputs, 16 outputs)		20 (12 inputs, 8 outputs) Line-driver inputs: Two axes for phases A, B, and Z Line-driver outputs: Two axes for CW and CCW
Number of connectable Expansion (I/O) Units		CP Expansion I/O Units: 7 max.; CJ-series Special I/O Units or CPU Bus Units: 2 max.		
Max. number of I/O points		320 (40 built in + 40 per Expansion (I/O) Unit × 7 Units)		300 (20 built in + 40 per Expansion (I/O) Unit × 7 Units)
Interrupt inputs		8 inputs (Shared by the external interrupt inputs (counter mode) and the quick-response inputs.)		6 inputs (Shared by the external interrupt inputs (counter mode) and the quick-response inputs.)
Interrupt input counter mode		8 inputs (Response frequency: 5 kHz max. for all interrupt inputs), 16 bits Up or down counters		6 inputs (Response frequency: 5 kHz max. for all interrupt inputs), 16 bits Up or down counters
Quick-response inputs		8 points (Min. input pulse width: 50 μs max.)		6 points (Min. input pulse width: 50 μs max.)
Scheduled interrupts		1		

รูปที่ ข-5 แสดงคุณสมบัติทางเทคนิคของ PLC CP1H (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP1H

Type Models		CP1H-XA CPU Units	CP1H-X CPU Units	CP1H-Y CPU Units
Item		CP1H-XA□□□□	CP1H-X□□□□	CP1H-Y□□□□
High-speed counters		4 inputs: Differential phases (4x), 50 kHz or Single-phase (pulse plus direction, up/down, increment), 100 kHz Value range: 32 bits, Linear mode or ring mode Interrupts: Target value comparison or range comparison		2 inputs: Differential phases (4x), 500 kHz or Single-phase, 1 MHz and 2 inputs: Differential phases (4x), 50 kHz or Single-phase (pulse plus direction, up/down, increment), 100 kHz Value range: 32 bits, Linear mode or ring mode Interrupts: Target value comparison or range comparison
Pulse outputs (models with transistor outputs only)	Pulse outputs	Trapezoidal or S-curve acceleration and deceleration (Duty ratio: 50% fixed) 4 outputs, 1 Hz to 100 kHz (CCW/CW or pulse plus direction)		Trapezoidal or S-curve acceleration and deceleration (Duty ratio: 50% fixed) 2 outputs, 1 Hz to 1 MHz (CCW/CW or pulse plus direction) 2 outputs, 1 Hz to 100 kHz (CCW/CW or pulse plus direction)
	PWM outputs	Duty ratio: 0.0% to 100.0% (Unit: 0.1%) 2 outputs, 0.1 to 6553.5 Hz (Accuracy: ±5% at 1 kHz)		
Built-in analog I/O terminals		4 analog inputs and 2 analog outputs	None	
Analog control		1 (Setting range: 0 to 255)		
External analog input		1 input (Resolution: 1/256, Input range: 0 to 10 V), not isolated		

Note: The memory areas for CJ-series Special I/O Units and CPU Bus Units are allocated at the same as for the CJ-series. For details, refer to the CJ Series catalog (Cat. No. P052).

รูปที่ ข-6 แสดงคุณสมบัติทางเทคนิคของ PLC CP1H (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NB series

Specifications

HMI

Specifications	NB3Q	NB5Q	NB7W	NB10W
	TW00B	TW01B	TW00B	TW01B
Display type	3.5" TFT LCD	5.6" TFT LCD	7" TFT LCD	10.1" TFT LCD
Display resolution (H × V)	320 × 240	320 × 234	800 × 480	800 × 480
Number of colours	65,536			
Backlight	LED			
Backlight lifetime	50,000 hours of operating time at the normal temperature (25°C) ^{†1}			
Touch panel	Analogue resistive membrane, resolution 1024 × 1024, life: 1 million touch operations			
Dimensions in mm (H × W × D)	103.8 × 129.8 × 52.8	142 × 184 × 46	148 × 202 × 46	210.8 × 268.8 × 54.0
Weight	310 g max.	620 g max.	710 g max.	1,545 g max.

^{†1} This is the estimated time when the luminous intensity is decreased by 50% per LED at room temperature and humidity. It is a typical value.

Functionality

Specifications	NB3Q	NB5Q	NB7W	NB10W
	TW00B	TW00B	TW00B	TW01B
Internal memory	128MB (including system area)			
Memory interface	—	USB Memory	—	USB Memory
Serial (COM1)	RS-232C/422A/485 (not isolated), Transmission distance: 15m Max. (RS-232C), 500m Max. (RS-422A/485), Connector: D-Sub 9-pin	RS-232C, Transmission distance: 15 m Max., Connector: D-Sub 9-pin		
Serial (COM2)	—	RS-232C/422A/485 (not isolated), Transmission distance: 15m Max. (RS-232C), 500m Max. (RS-422A/485), Connector: D-Sub 9-pin		
USB Host	Equivalent to USB 2.0 full speed, type A. Output power 5V, 150mA			
USB Slave	Equivalent to USB 2.0 full speed, type B. Transmission distance: 5m			
Printer connection	PictBridge support			
Ethernet	10/100 base-T	10/100 base-T	10/100 base-T	10/100 base-T

General

Specifications	NB3Q	NB5Q	NB7W	NB10W
	TW00B	TW00B	TW00B	TW01B
Line voltage	20.4 to 27.6 VDC (24 VDC—15 to 15V)			
Power consumption	5 W	6 W	7 W	14 W
Battery lifetime	5 years (at 25°C)			
Enclosure rating (front side)	Front operation part: IP65 (Dust proof and drip proof only from the front of the panel)			
Obtained standards	IEC Directives, KC, cUL505			
Operating environment	No corrosive gases			
Noise immunity	Compliant with IEC61000-4-4, 2kV (Power cable)			
Ambient operating temperature	0 to 50°C			
Ambient operating humidity	10% to 90% RH (without condensation)			

Applicable Controllers

Brand	Series	Brand	Series
OMRON	Omron C Series Host Link	Schneider	Schneider Modicon Uni-Terway
	Omron CJ/CS Series Host Link		Schneider Twido Modbus RTU
	Omron CP Series		Delta DVP
Mitsubishi	Mitsubishi Q-QnA (Link Port)	Delta (LS)	LS Master-K Cnet
	Mitsubishi FX-485ADP/485QJ/422BD (Modbus station)		LS Master-K CPU Direct
	Mitsubishi FX0N/1N/2N/3Q		LS Master-K Modbus RTU ^{†1}
	Mitsubishi FX1S		LS XGT CPU Direct
	Mitsubishi FX2N-10GM/20GM		LS XGT Cnet
	Mitsubishi FX3U	GE Fanuc Automation ^{†1}	GE Fanuc Series SNP
	Mitsubishi Q-series (CPU Port)		GE SNP-4 V
	Mitsubishi Q00J (CPU Port)	Modbus	Modbus ASCII
Panasonic	FP series		Modbus RTU
	Siemens S7-200		Modbus RTU Slave
Siemens	Siemens S7-300/400 (PC Adapter Direct)		Modbus RTU Extend
			Modbus TCP
Allen-Bradley ^{†1} (Rockwell)	AB DF1		
	AB CompactLogix/ControlLogix		

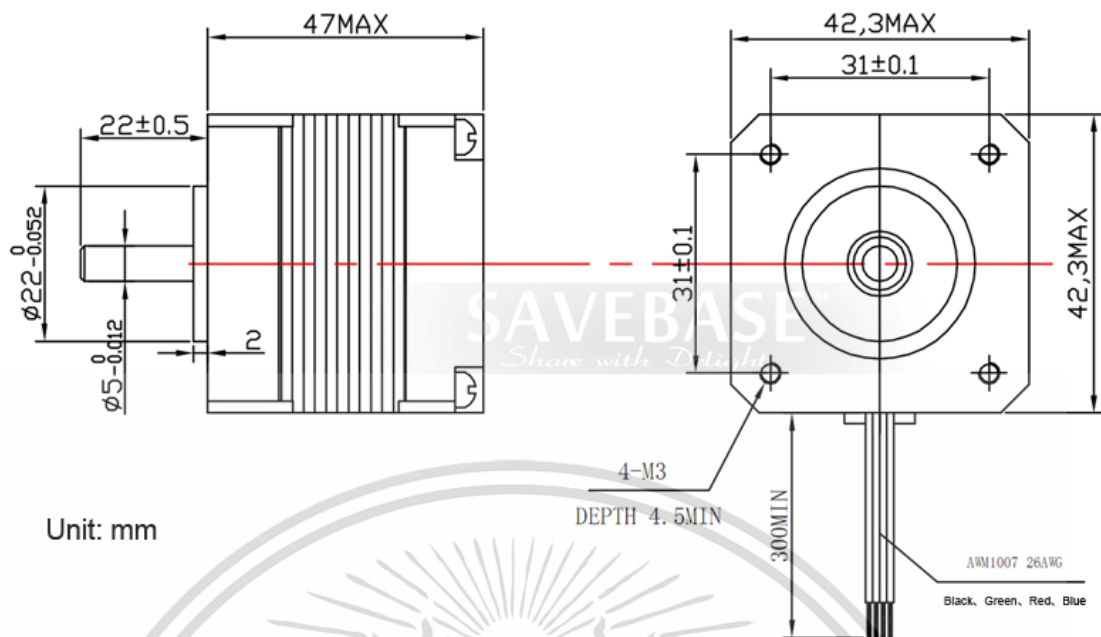
^{†1} AB and GE will be supported by NB-Designer version 1.20 or higher.

Note: For details, refer to NB Series Host Connection Manual (Cat.No V108).

OMRON

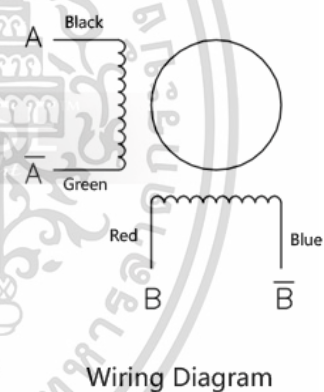
รูปที่ ข-7 แสดงคุณสมบัติทางเทคนิคของ HMI Omron NB5QTW00B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Unit: mm

Motor specifications	
Model	42BYGH47-401A
Rated Voltage	2.8VDC
Rated Current	1.68A/Phase
Step Angle	$1.8^\circ \pm 5\%$
Phase	2
Phase resistance	$1.65 \times (1 + 10\%) \Omega / \text{Phase}$
Phase inductance	$2.8 \times (1 + 20\%) \text{mH} / \text{Phase}$
Holding torque	0.54Nm (approx. 76 Oz-in)
Positioning torque	150gf.cm
Insulation resistance	$\geq 100 \text{M}\Omega$ (500VDC)
Insulation class	CLASS: B
Weight	0.35kg



รูปที่ ข-8 แสดงคุณสมบัติทางเทคนิคของมอเตอร์แบบสเต็ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

www.sorotec.de

TB6600
Stepper Motor Driver

**Analog Driver
Model TB6600**

Analog Technology, max. 40 VDC / 4.0 A (PEAK)

Product Description:
The TB6600 single axis drive is a low cost microstepping drive. It is suitable for driving 2-phase and 4-phase hybrid stepper motors. Not for professional applications.

Features:

- Cost-effective
- Supply voltage up to +40 VDC, Output current up to 4.0 A (PEAK)
- Output current selectable in 8 steps via DIP-switch
- Automatic idle-current reduction (in standstill mode) to reduce motor heating
- Pulse input frequency up to 20 kHz
- Input suitable for 5 V signals
- Inputs are optically isolated
- 6 selectable microstep resolutions, up to 6400 steps/rev with standard 1.8° motors
- Suitable for 2-phase and 4-phase motors
- Supports PUL/DIR mode
- Over current and overheat protection

Electrical Specifications:

Parameters	Min	Typ.	Max	Unit
Output current	0.7	-	4.0 (3.5 RMS)	A
Supply voltage	+9	+36	+40	VDC
Logic signal current	8	10	15	mA
Puls input frequency	0	-	20 when duty cycle is 25 high / 75 low 13 when duty cycle is 50 / 50	kHz
Insulation resistance	500			MΩ

Further Specifications:

Microsteps / 1,8°	200		6400
PUL / DIR		yes	
NEMA sizes	17		24
Motor type Mechatron	42BYGH-XXXX		60BYGH-XXX



22.01.18

Right of techn. modifications is reserved
www.sorotec.de
Technische Änderungen vorbehalten

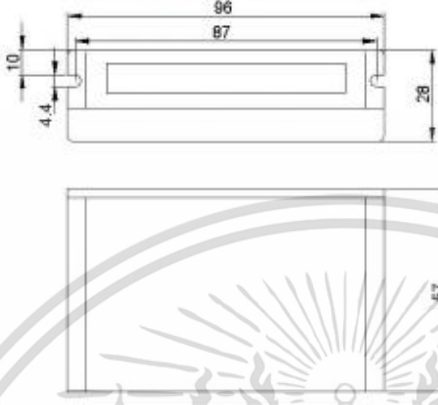
รูปที่ ข-9 แสดงคุณสมบัติทางเทคนิคของวงจรขับเคลื่อนมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TB6600
Stepper motor driver

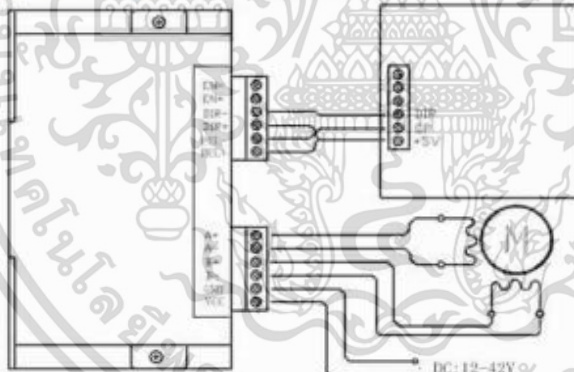
Analog Driver
Model TB6600

Mechanical Specifications: (Unit: mm)



Applications:
 Suitable for a wide range of stepping motors of NEMA sizes 17, 23 and 24 (42x42 mm to 60x60 mm). It can be used in various kinds of machines, such as X-Y tables, engraving machines, labeling machines, laser cutters, pick-place devices, and so on. Particularly well suited for applications where low noise levels, less heat development, high speed and high precision are desired.

Typical Connection Schematic:
 A typical system consists of stepper motor, stepper motor driver, power supply and controller. The following image shows a typical connection schematic:



Logic control signals which have 5 V can be connected directly;
 R 1k Ω must be connected in line when control signal is 12V;
 R 2k Ω must be connected in line when control signal is 24V to ensure control signal current is 8mA to 15mA.

22.01.18

Right of techn. modifications is reserved
www.sorotec.de
Technische Änderungen vorbehalten

รูปที่ ข-10 แสดงคุณสมบัติทางเทคนิคของวงจรขับเคลื่อนมอเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPECIFICATIONS

Item	Model No.	Type	Small	
			With cable	With connector
		NPN output	PM-Q44	PM-Q54
		PNP output	PM-Q44P	PM-Q54P
Sensing range		5 mm 0.197 in (fixed)		
Minimum sensing object		0.8 × 1.8 mm 0.031 × 0.071 in opaque object		
Hysteresis		0.05 mm 0.002 in or less		
Repeatability		0.03 mm 0.001 in or less		
Supply voltage		5 to 24 V DC ±10 % Ripple P-P 10 % or less		
Current consumption		15 mA or less		
Output		<NPN output type> NPN open-collector transistor - Maximum sink current: 50 mA - Applied voltage: 30 V DC or less (between output and 0 V) - Residual voltage: 0.7 V or less (at 50 mA sink current) 0.4 V or less (at 16 mA sink current) <PNP output type> PNP open-collector transistor - Maximum source current: 50 mA - Applied voltage: 30 V DC or less (between output and +V) - Residual voltage: 0.7 V or less (at 50 mA source current) 0.4 V or less (at 16 mA source current)		
Utilization category		DC-12 or DC-13		
Output operation		Incorporated with 2 outputs: Light-ON / Dark-ON		
Response time		Under light received condition: 20 μs or less; Under light interrupted condition: 100 μs or less (Response frequency: 1 kHz or more) (Note 2)		
Operation indicator		Vermilion LED (lights up under light received condition)		
Pollution degree		3 (Industrial environment)		
Ambient temperature		-25 to +55 °C -13 to +131 °F (No dew condensation or icing allowed). Storage: -30 to +80 °C -22 to +176 °F		
Ambient humidity		35 to 85 % RH. Storage: 35 to 85 % RH		
Ambient illuminance		Fluorescent light: 1,000 lx at the light-receiving face		
EMC		EN 60947-5-2		
Voltage withstandability		1,000 V AC for one min. between all supply terminals connected together and enclosure		
Insulation resistance		50 MΩ, or more, with 250 V DC megger between all supply terminals connected together and enclosure		
Vibration resistance		10 to 2,000 Hz frequency, 1.5 mm 0.059 in amplitude in X, Y and Z directions for two hours each		
Shock resistance		15,000 m/s ² acceleration (1,500 G approx.) in X, Y and Z directions for three times each		
Emitting element		Infrared LED (Peak emission wavelength: 940 nm 0.037 mil, non-modulated)		
Material		Enclosure: PBT, Silt cover: Polycarbonate, Terminal part (PM-Q54(P) only): Solder plated		
Cable		0.09 mm ² 4-core cabtyne cable, 1 m 3.251 ft long		
Cable extension		Extension up to total 100 m 328/344 ft is possible with 0.3 mm ² , or more, cable.		
Weight		Net weight: 15 g approx.		Net weight: 3 g approx.

Notes: 1) Where measurement conditions have not been specified precisely, the conditions used were an ambient temperature of +23 °C +73.4 °F.
2) The response frequency is the value when the disc, given in the figure below, is rotated.



รูปที่ ข-11 แสดงคุณสมบัติทางเทคนิคของไฟโตอิเล็กทริกเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค

โปรแกรมแบบจำลองทางคณิตศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 1 of 14

```
function varargout = testKI(varargin)
% TESTKI MATLAB code for testKI.fig
%     TESTKI, by itself, creates a new TESTKI or raises the existing
%     singleton*.
%
%     H = TESTKI returns the handle to a new TESTKI or the handle to
%     the existing singleton*.
%
%     TESTKI('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in TESTKI.M with the given input arguments.
%
%     TESTKI('Property','Value',...) creates a new TESTKI or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before testKI_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to testKI_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help testKI
%
% Last Modified by GUIDE v2.5 18-Dec-2022 04:02:15
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @testKI_OpeningFcn, ...
                  'gui_OutputFcn',  @testKI_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before testKI is made visible.
function testKI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```

รูปที่ ค-1 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 2 of 14

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to testKI (see VARARGIN)

% Choose default command line output for testKI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes testKI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = testKI_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
modelName = 'updateSCARA';
global var;
t1 = get(handles.slider1, 'value');
set(handles.edit1, 'string', num2str(t1));
d1 = get(handles.slider2, 'value');
set(handles.edit2, 'string', num2str(d1));
t2 = get(handles.slider3, 'value');
t2 = t2*(-1);
set(handles.edit3, 'string', num2str(t2));
t3 = get(handles.slider4, 'value');
set(handles.edit4, 'string', num2str(t3));
set_param([modelName ' /Slider Gain'], 'Gain', num2str(t1))
set_param([modelName ' /Slider Gain1'], 'Gain', num2str(d1))
set_param([modelName ' /Slider Gain2'], 'Gain', num2str(t2))
set_param([modelName ' /Slider Gain3'], 'Gain', num2str(t3))
%%Kinematics Forward Model
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
H01 = [cosd(t1) -sind(t1) 0 0; sind(t1) cosd(t1) 0 0; 0 0 1 d0; 0 0 0 1];
H12 = [1 0 0 0; 0 1 0 0; 0 0 1 d1/1000; 0 0 0 1];
H23 = [cosd(t2) -sind(t2) 0 L2; sind(t2) cosd(t2) 0 0; 0 0 1 0; 0 0 0 1];
```

รูปที่ ค-2 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 3 of 14

```
H34 = [ cosd(t3) -sind(t3) 0 L3; -sind(t3) -cosd(t3) 0 0; 0 0 -1 -d4; 0 0 0 1];
H4E = [1 0 0 0; 0 1 0 0; 0 0 1 -d5; 0 0 0 1];
H0E = H01*H12*H23*H34*H4E;
px = H0E(1,4)*1000;
py = H0E(2,4)*1000;
pz = H0E(3,4)*1000;
set(handles.edit5, 'string', num2str(px));
set(handles.edit6, 'string', num2str(py));
set(handles.edit7, 'string', num2str(pz));
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, ~, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
```

รูปที่ ค-3 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 4 of 14

```

set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)
ModelName = 'updateSCARA';
global var;
t1 = get(handles.slider1,'value');
set(handles.edit1,'string',num2str(t1));
d1 = get(handles.slider2,'value');
set(handles.edit2,'string',num2str(d1));
t2 = get(handles.slider3,'value');
t2 = t2*(-1);
set(handles.edit3,'string',num2str(t2));
t3 = get(handles.slider4,'value');
set(handles.edit4,'string',num2str(t3));
set_param([ModelName '/Slider Gain'], 'Gain', num2str(t1))
set_param([ModelName '/Slider Gain1'], 'Gain', num2str(d1))
set_param([ModelName '/Slider Gain2'], 'Gain', num2str(t2))
set_param([ModelName '/Slider Gain3'], 'Gain', num2str(t3))
%%Kinematics Forward Model
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
H01 = [cosd(t1) -sind(t1) 0 0; sind(t1) cosd(t1) 0 0; 0 0 1 d0; 0 0 0 1];
H12 = [1 0 0 0; 0 1 0 0; 0 0 1 d1/1000; 0 0 0 1];
H23 = [cosd(t2) -sind(t2) 0 L2; sind(t2) cosd(t2) 0 0; 0 0 1 0; 0 0 0 1];
H34 = [ cosd(t3) -sind(t3) 0 L3; -sind(t3) -cosd(t3) 0 0; 0 0 -1 -d4; 0 0 0 1];
H4E = [1 0 0 0; 0 1 0 0; 0 0 1 -d5; 0 0 0 1];
H0E = H01*H12*H23*H34*H4E;
px = H0E(1,4)*1000;
py = H0E(2,4)*1000;
pz = H0E(3,4)*1000;
set(handles.edit5,'string',num2str(px));
set(handles.edit6,'string',num2str(py));
set(handles.edit7,'string',num2str(pz));
% hObject handle to slider2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
% hObject handle to slider2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

รูปที่ ค-4 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 5 of 14

```
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider3_Callback(hObject, eventdata, handles)
ModelName = 'updateSCARA';
global var;
t1 = get(handles.slider1,'value');
set(handles.edit1,'string',num2str(t1));
d1 = get(handles.slider2,'value');
set(handles.edit2,'string',num2str(d1));
t2 = get(handles.slider3,'value');
set(handles.edit3,'string',num2str(t2));
t2 = t2*(-1);
t3 = get(handles.slider4,'value');
set(handles.edit4,'string',num2str(t3));
set_param([ModelName '/Slider Gain'],'Gain',num2str(t1))
set_param([ModelName '/Slider Gain1'],'Gain',num2str(d1))
set_param([ModelName '/Slider Gain2'],'Gain',num2str(t2))
set_param([ModelName '/Slider Gain3'],'Gain',num2str(t3))
%Kinematics Forward Model
```

รูปที่ ค-5 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 6 of 14

```
d0 = 0.12; d4 = -0.039325; d5= -0.06;
L2 = 0.20; L3 = 0.15;
H01 = [cosd(t1) -sind(t1) 0 0; sind(t1) cosd(t1) 0 0; 0 0 1 d0; 0 0 0 1];
H12 = [1 0 0 0; 0 1 0 0; 0 0 1 d1/1000; 0 0 0 1];
H23 = [cosd(t2) -sind(t2) 0 L2; sind(t2) cosd(t2) 0 0; 0 0 1 0; 0 0 0 1];
H34 = [ cosd(t3) -sind(t3) 0 L3; -sind(t3) -cosd(t3) 0 0; 0 0 -1 -d4; 0 0 0 1];
H4E = [1 0 0 0; 0 1 0 0; 0 0 1 -d5; 0 0 0 1];
H0E = H01*H12*H23*H34*H4E;
px = H0E(1,4)*1000;
py = H0E(2,4)*1000;
pz = H0E(3,4)*1000;
set(handles.edit5, 'string', num2str(px));
set(handles.edit6, 'string', num2str(py));
set(handles.edit7, 'string', num2str(pz));
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%        get(hObject, 'Min') and get(hObject, 'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit3 as text
%        str2double(get(hObject, 'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called
```

รูปที่ ค-6 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 7 of 14

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider4_Callback(hObject, eventdata, handles)
ModelName = 'updateSCARA';
global var;
t1 = get(handles.slider1,'value');
set(handles.edit1,'string',num2str(t1));
d1 = get(handles.slider2,'value');
set(handles.edit2,'string',num2str(d1));
t2 = get(handles.slider3,'value');
set(handles.edit3,'string',num2str(t2));
t2 = t2*(-1);
t3 = get(handles.slider4,'value');
set(handles.edit4,'string',num2str(t3));
set_param([ModelName '/Slider Gain'], 'Gain', num2str(t1))
set_param([ModelName '/Slider Gain1'], 'Gain', num2str(d1))
set_param([ModelName '/Slider Gain2'], 'Gain', num2str(t2))
set_param([ModelName '/Slider Gain3'], 'Gain', num2str(t3))
%%Kinematics Forward Model
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
H01 = [cosd(t1) -sind(t1) 0 0; sind(t1) cosd(t1) 0 0; 0 0 1 d0; 0 0 0 1];
H12 = [1 0 0 0; 0 1 0 0; 0 0 1 d1/1000; 0 0 0 1];
H23 = [cosd(t2) -sind(t2) 0 L2; sind(t2) cosd(t2) 0 0; 0 0 1 0; 0 0 0 1];
H34 = [ cosd(t3) -sind(t3) 0 L3; -sind(t3) -cosd(t3) 0 0; 0 0 -1 -d4; 0 0 0 1];
H4E = [1 0 0 0; 0 1 0 0; 0 0 1 -d5; 0 0 0 1];
H0E = H01*H12*H23*H34*H4E;
px = H0E(1,4)*1000;
py = H0E(2,4)*1000;
pz = H0E(3,4)*1000;
set(handles.edit5,'string',num2str(px));
set(handles.edit6,'string',num2str(py));
set(handles.edit7,'string',num2str(pz));
% hObject handle to slider4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
% get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

รูปที่ ค-7 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้า ที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 8 of 14

```
% --- Executes during object creation, after setting all properties.
function slider4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider5_Callback(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
```

รูปที่ ค-8 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 9 of 14

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit5_Callback(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
% str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider6_Callback(hObject, eventdata, handles)
% hObject handle to slider6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider6_CreateFcn(hObject, eventdata, handles)
% hObject handle to slider6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

รูปที่ ค-9 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 10 of 14

```
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider7_Callback(hObject, eventdata, handles)
% hObject    handle to slider7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
```

รูปที่ ค-10 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 11 of 14

```
end

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
modelName = 'updateSCARA';
global var;
t1 = 0;
set(handles.edit1,'string',num2str(t1));
d1 = 0;
set(handles.edit2,'string',num2str(d1));
t2 = 0;
set(handles.edit3,'string',num2str(t2));
t3 = 0;
set(handles.edit4,'string',num2str(t3));
set_param([modelName '/Slider Gain'], 'Gain', num2str(t1))
set_param([modelName '/Slider Gain1'], 'Gain', num2str(d1))
set_param([modelName '/Slider Gain2'], 'Gain', num2str(t2))
set_param([modelName '/Slider Gain3'], 'Gain', num2str(t3))
%%Kinematics Forward Model
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
H01 = [cosd(t1) -sind(t1) 0 0; sind(t1) cosd(t1) 0 0; 0 0 1 d0; 0 0 0 1];
H12 = [1 0 0 0; 0 1 0 0; 0 0 1 d1/1000; 0 0 0 1];
H23 = [cosd(t2) -sind(t2) 0 L2; sind(t2) cosd(t2) 0 0; 0 0 1 0; 0 0 0 1];
```

รูปที่ ค-11 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 12 of 14

```
H34 = [ cosd(t3) -sind(t3) 0 L3; -sind(t3) -cosd(t3) 0 0; 0 0 -1 -d4; 0 0 0 1];
H4E = [1 0 0 0; 0 1 0 0; 0 0 1 -d5; 0 0 0 1];
H0E = H01*H12*H23*H34*H4E;
px = H0E(1,4)*1000;
py = H0E(2,4)*1000;
pz = H0E(3,4)*1000;
set(handles.edit5, 'string', num2str(px));
set(handles.edit6, 'string', num2str(py));
set(handles.edit7, 'string', num2str(pz));
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
modelName = 'updatesCARA';
global var;
px=get(handles.edit5, 'string');
px = str2num(px);
if ((px <= -350)&(px >=350))
    msgbox('Please Select PX Value In Range -350-->350');
end
py=get(handles.edit6, 'string');
py = str2num(py);
if ((py <= -350)&(py >=350))
    msgbox('Please Select PY Value In Range -350-->350');
end
pz=get(handles.edit7, 'string');
pz = str2num(pz);
if ((pz <= 20)&(pz >= 180))
    msgbox('Please Select PZ Value In Range 20-->180');
end
pxi = px/1000;
pyi = py/1000;
pzi = pz/1000;
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
C2 = ((pxi)^2 + (pyi)^2 - (L2)^2 - (L3)^2) / (2*L2*L3);
C2f = round(C2*1000)/1000;
S2 = 1*sqrt(1-(C2f)^2);
if (S2 <= -1)
    msgbox('Robot Crash Close Program to Restart');at
end
t2 = atan2d(S2, C2f)*(-1);
tln = t2*(-1)
t1 = atan2d((pyi), (pxi))-atan2d(L3*S2, L2+(L3*C2f));
d1 = (pzi + d4 - d5 - d0)*1000 ;
t3 = 0;
```

รูปที่ ค-12 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12/18/22 11:15 PM C:\Users\User\OneDrive\D...\testKI.m 13 of 14

```

guidata(hObject, handles);
set(handles.edit1, 'string', t1);
set_param([modelName 'Slider Gain'], 'Gain', num2str(t1))
set(handles.edit2, 'string', d1);
set_param([modelName 'Slider Gain1'], 'Gain', num2str(d1))
set(handles.edit3, 'string', t2);
set_param([modelName 'Slider Gain2'], 'Gain', num2str(t2))
set(handles.edit4, 'string', t3);
set_param([modelName 'Slider Gain3'], 'Gain', num2str(t3))
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
close;
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
modelName = 'updatesCARA';
global var;
t1=get(handles.edit1, 'string');
t1 = str2num(t1);
if ((t1 <= -180)&(t1 >=180))
    msgbox('Please Select t1 Value In Range -180-->180');
end
d1=get(handles.edit2, 'string');
d1 = str2num(d1);
if ((d1 <= 20)&(d1 >=180))
    msgbox('Please Select d1 Value In Range 20-->180');
end
t2=get(handles.edit3, 'string');
t2 = str2num(t2);
t2 = t2*(-1);
if ((t2 <= -145)&(t2 >=145))
    msgbox('Please Select t2 Value In Range -145-->145');
end
t3=get(handles.edit4, 'string');
t3 = str2num(t3);
if ((t3 <= -180)&(t3 >=180))
    msgbox('Please Select t3 Value In Range -180-->180');
end
set_param([modelName 'Slider Gain'], 'Gain', num2str(t1))
set_param([modelName 'Slider Gain1'], 'Gain', num2str(d1))

set_param([modelName 'Slider Gain2'], 'Gain', num2str(t2))
set_param([modelName 'Slider Gain3'], 'Gain', num2str(t3))
set(handles.edit4, 'string', num2str(t3));
%%Kinematics Forward Model
d0 = 0.12; d4 = -0.039325; d5 = -0.06;
L2 = 0.20; L3 = 0.15;
H01 = [cosd(t1) -sind(t1) 0 0; sind(t1) cosd(t1) 0 0; 0 0 1 d0; 0 0 0 1];
H12 = [1 0 0 0; 0 1 0 0; 0 0 1 d1/1000; 0 0 0 1];
H23 = [cosd(t2) -sind(t2) 0 L2; sind(t2) cosd(t2) 0 0; 0 0 1 0; 0 0 0 1];
H34 = [ cosd(t3) -sind(t3) 0 L3; -sind(t3) -cosd(t3) 0 0; 0 0 -1 -d4; 0 0 0 1];
H4E = [1 0 0 0; 0 1 0 0; 0 0 1 -d5; 0 0 0 1];
H0E = H01*H12*H23*H34*H4E;
px = H0E(1,4)*1000;
py = H0E(2,4)*1000;
pz = H0E(3,4)*1000;
set(handles.edit5, 'string', num2str(px));
set(handles.edit6, 'string', num2str(py));
set(handles.edit7, 'string', num2str(pz));
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

```

รูปที่ ค-13 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
โปรแกรม PLC ควบคุมแขนกล SCARA Robot 4DOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000000	[Program Name : NewProgram1]		
	[Section Name : Sensor_Protection]		
	I: 1.00	10.00	Start Sw Buffer
	SW1 Start		
000001	I: 1.01	10.01	Stop Sw Buffer
	SW2 Stop		
000002	I: 1.02	10.02	Teach Sw Buffer
	Teach Sw	Tech	
000003	I: 1.03	10.03	Home Sw Buffer
	Home Sw		
000004	W1.04	10.09	Auto Comma...
	Comma...		
000005	W1.05	10.10	Auto Command B
	Comma...		
000006	W1.06	10.11	FW Buffer
	Forwa...		
000007	I: 0.00	10.04	Photo Light "ON"
	M1 Photo		
000008	I: 0.10	10.05	LS1 M2 UP Buffer
	LS1 M2 UP		
000009	I: 0.11	10.06	LS2 M2 DO...
	LS2 ...		
000010	I: 0.09	10.07	LS3 M3 RIG...
	LS3 ...		
000011	I: 1.08	10.08	LS3 M3 LEFT Buffer
	LS4 ...		
000012	10.04	A540.08	M1 Base Photect
	Photo Lig...		
000013	10.05	A541.09	M2 Up Protect
	LS1 M2 ...		
000014	10.06	A541.08	M2 Down Protect
	LS2 ...		
000015	10.07	A542.08	M3 Right protect
	LS3 ...		
000016	10.08	A542.09	M3 Left Protect
	LS3 ...		

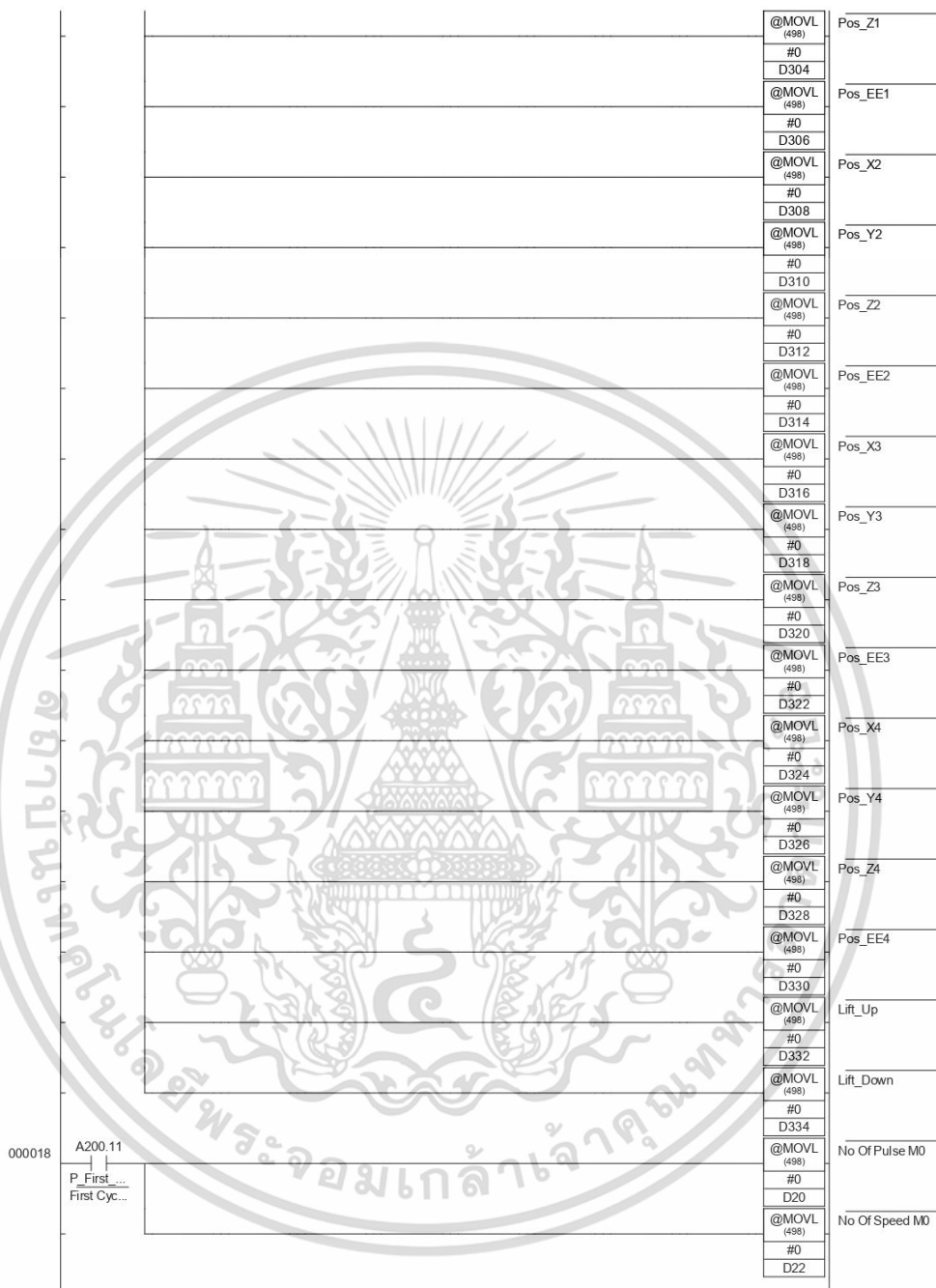
รูปที่ ง-1 โปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000017	A200.11		@MOVL (498)	Home Theta M0
	P_First...		#0	
	First Cyc...		D0	
			@MOVL (498)	Home Lift M1
			#0	
			D2	
			@MOVL (498)	Home Theta M2
			#0	
			D4	
			@MOVL (498)	Home Theta M3
			#0	
			D6	
			@MOVL (498)	BaseTheta
			#0	
			D50	
			@MOVL (498)	LiftTheta
			#0	
			D52	
			@MOVL (498)	ShoulderTheta
			#0	
			D54	
			@MOVL (498)	EETheta
			#0	
			D56	
			@MOVL (498)	X Value in mm.
			#0	
			D8	
			@MOVL (498)	Y Value in mm.
			#0	
			D10	
			@MOVL (498)	Z Value in mm.
			#0	
			D12	
			@MOVL (498)	Theta Of EE
			#0	
			D14	
			@MOVL (498)	M0 Actual Pulse
			#0	
			D100	
			@MOVL (498)	M1 Actual Pulse
			#0	
			D102	
			@MOVL (498)	M2 Actual Pulse
			#0	
			D104	
			@MOVL (498)	M3 Actual Pulse
			#0	
			D106	
			@MOVL (498)	Pos_X1
			#0	
			D300	
			@MOVL (498)	Pos_Y1
			#0	
			D302	

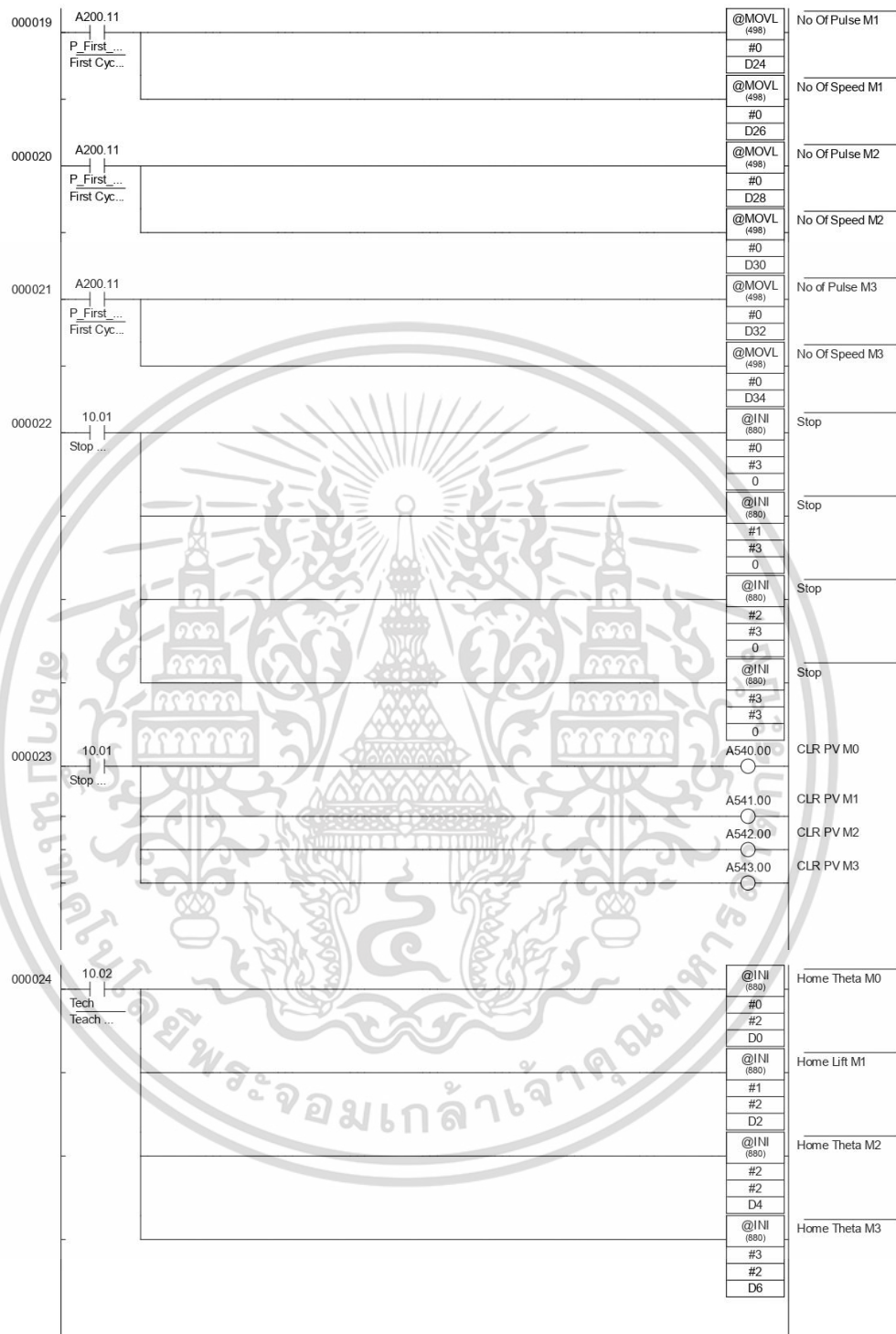
รูปที่ ง-2 โปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



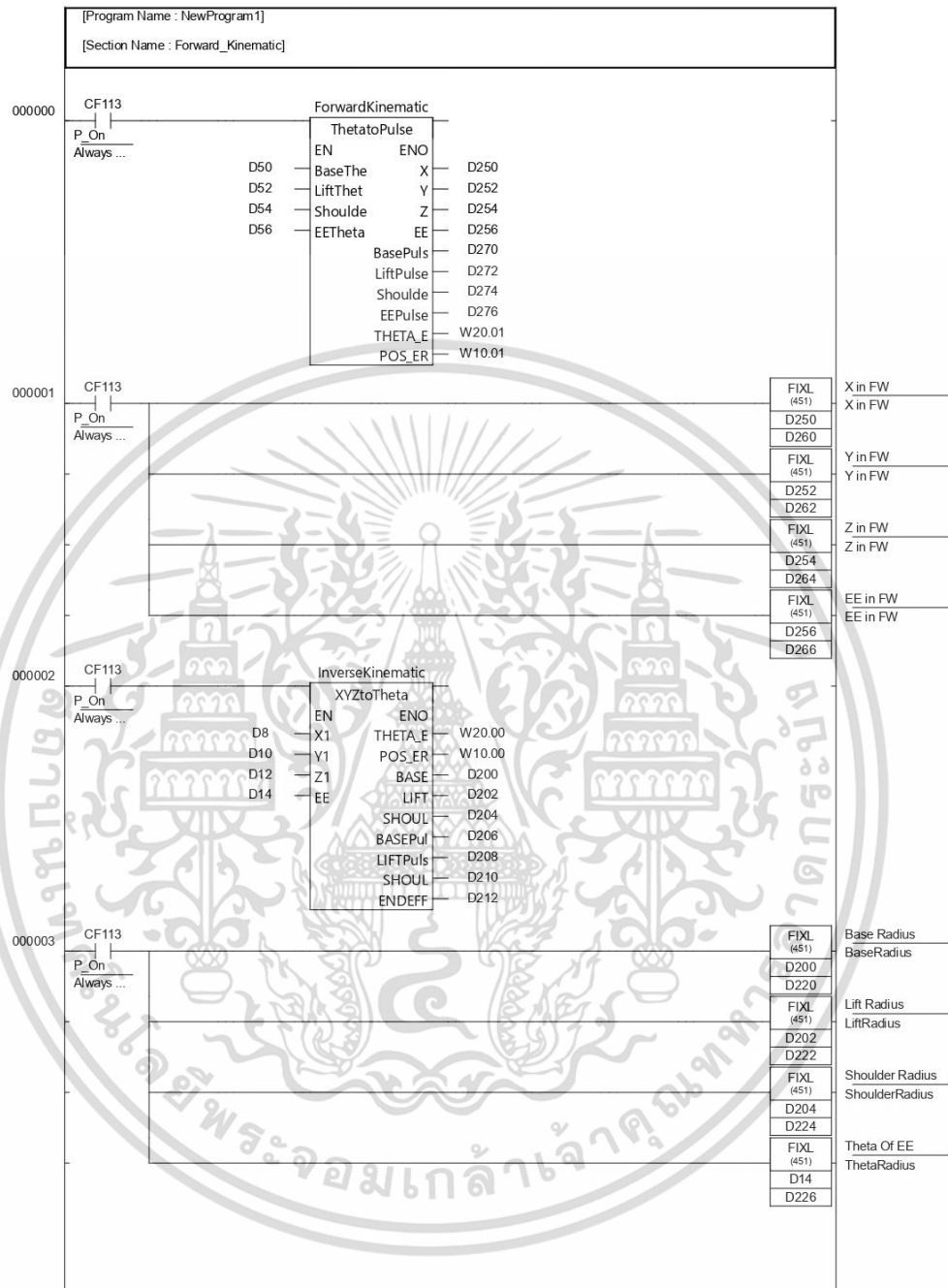
รูปที่ ง-3 โปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



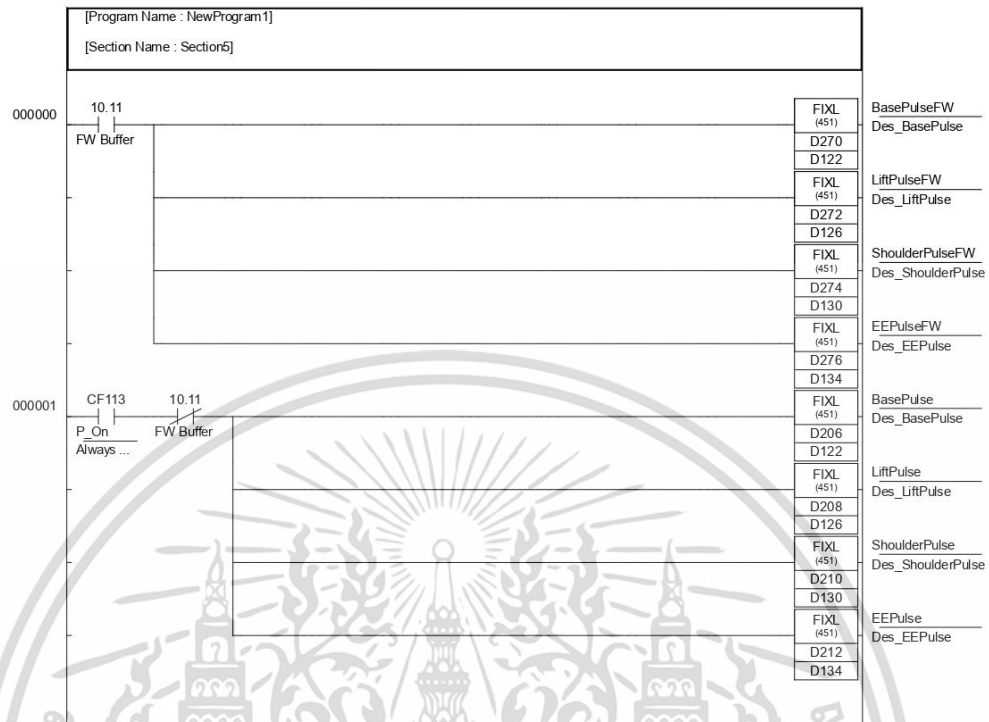
รูปที่ ง-4 โปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

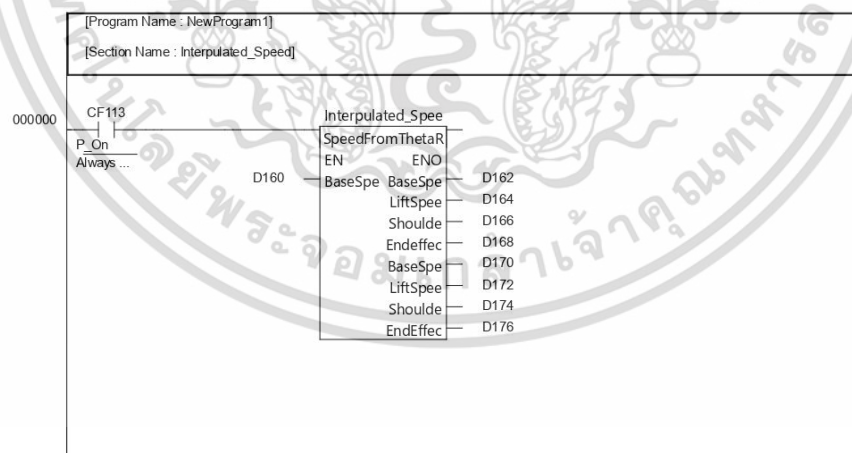


รูปที่ ง-5 โปรแกรม PLC การตั้งค่าพารามิเตอร์เริ่มต้น (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

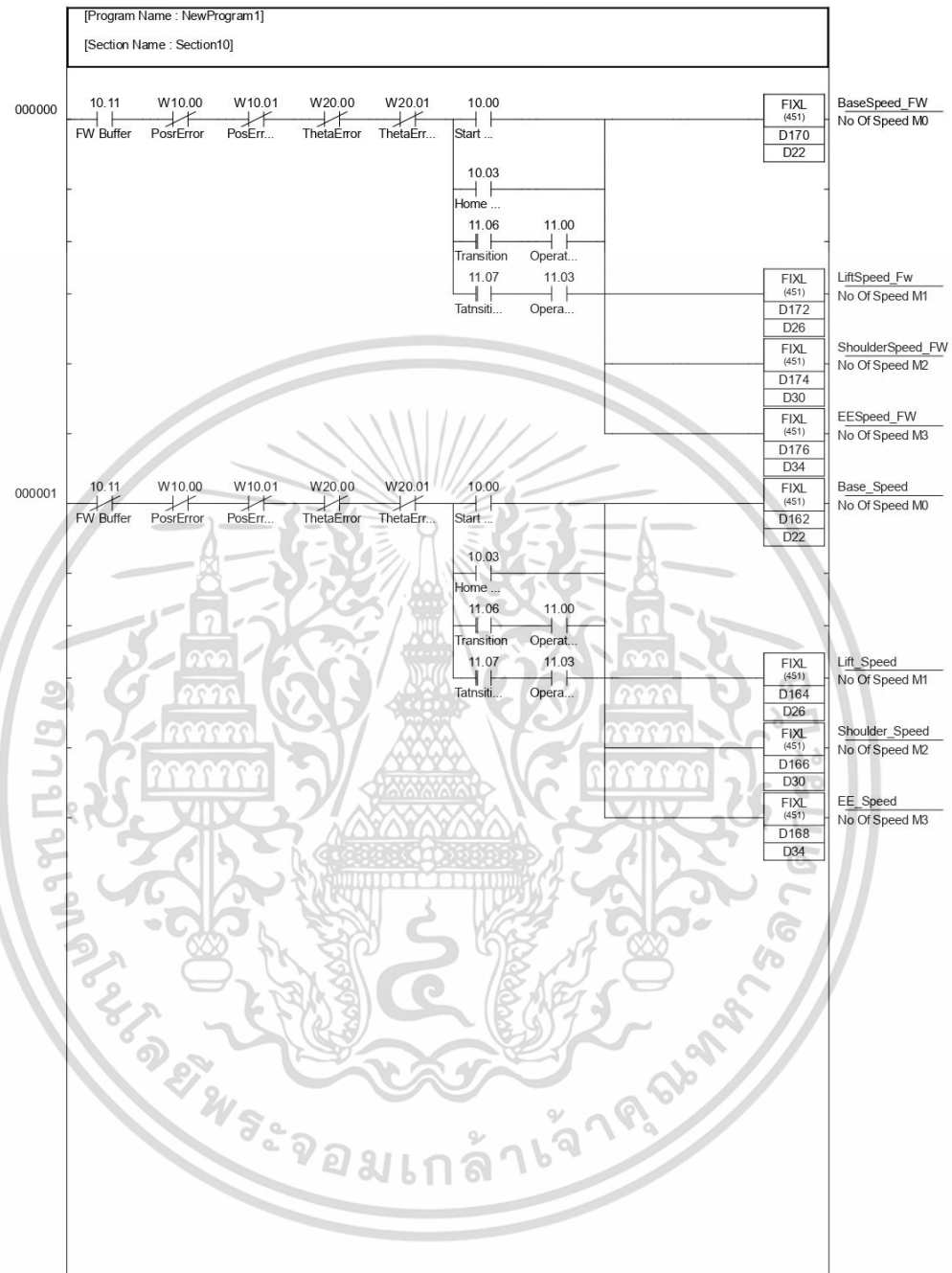


รูปที่ ง-6 แปลงค่าพัลส์จากFloating to DWord



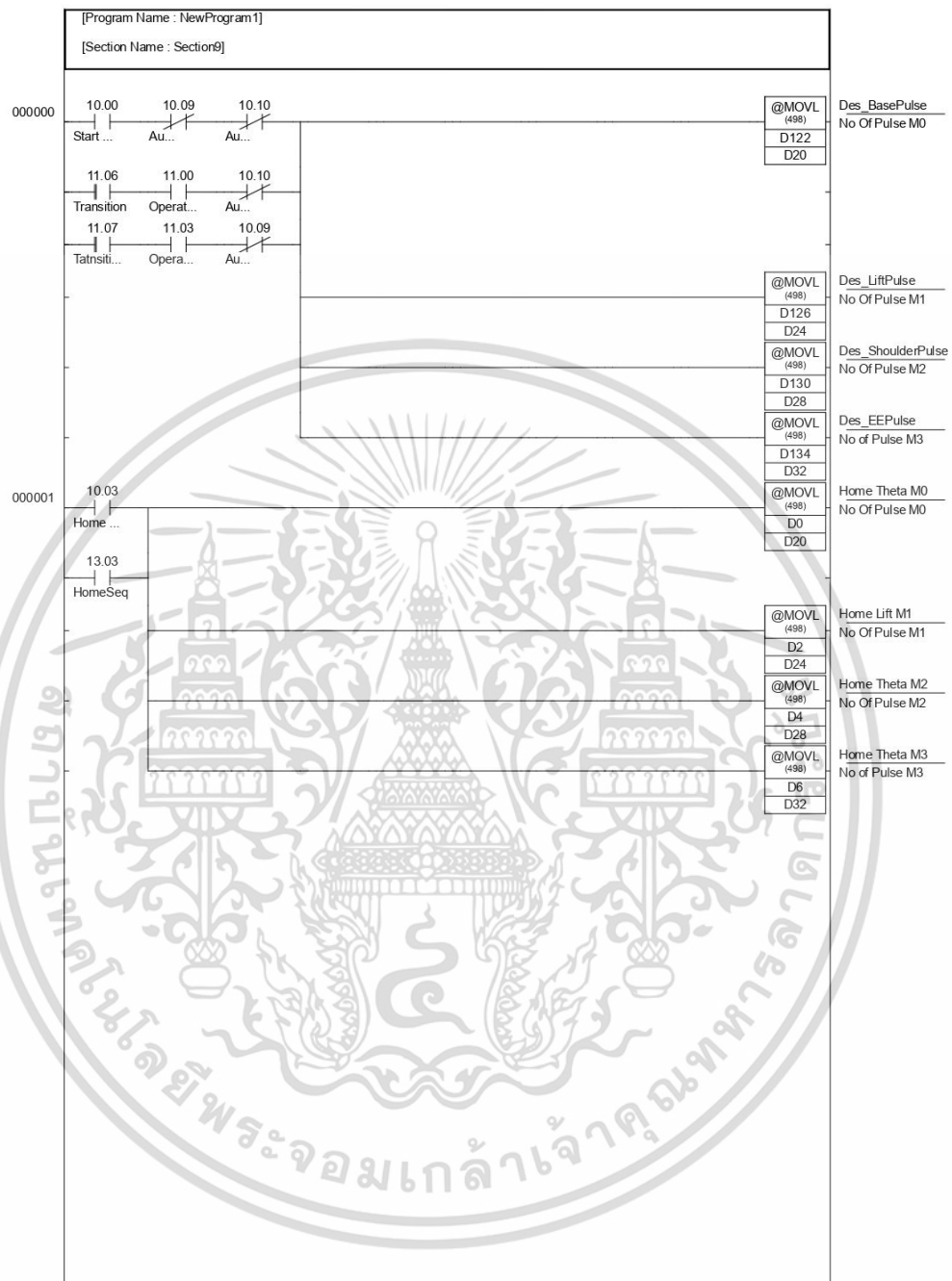
รูปที่ ง-7 คำนวณค่าความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



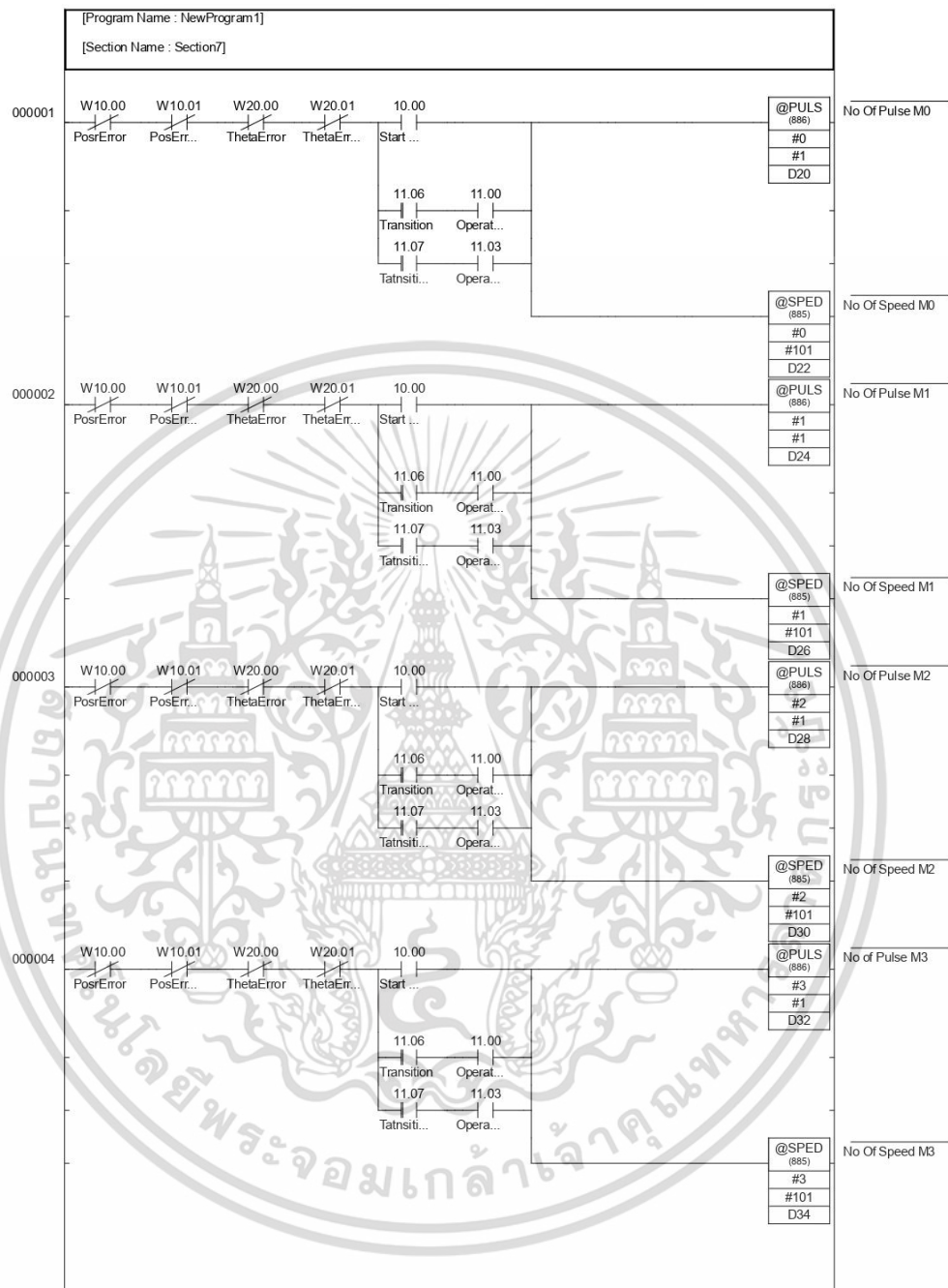
รูปที่ ง-8 แปลงค่าความเร็วจาก Float to DWord

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



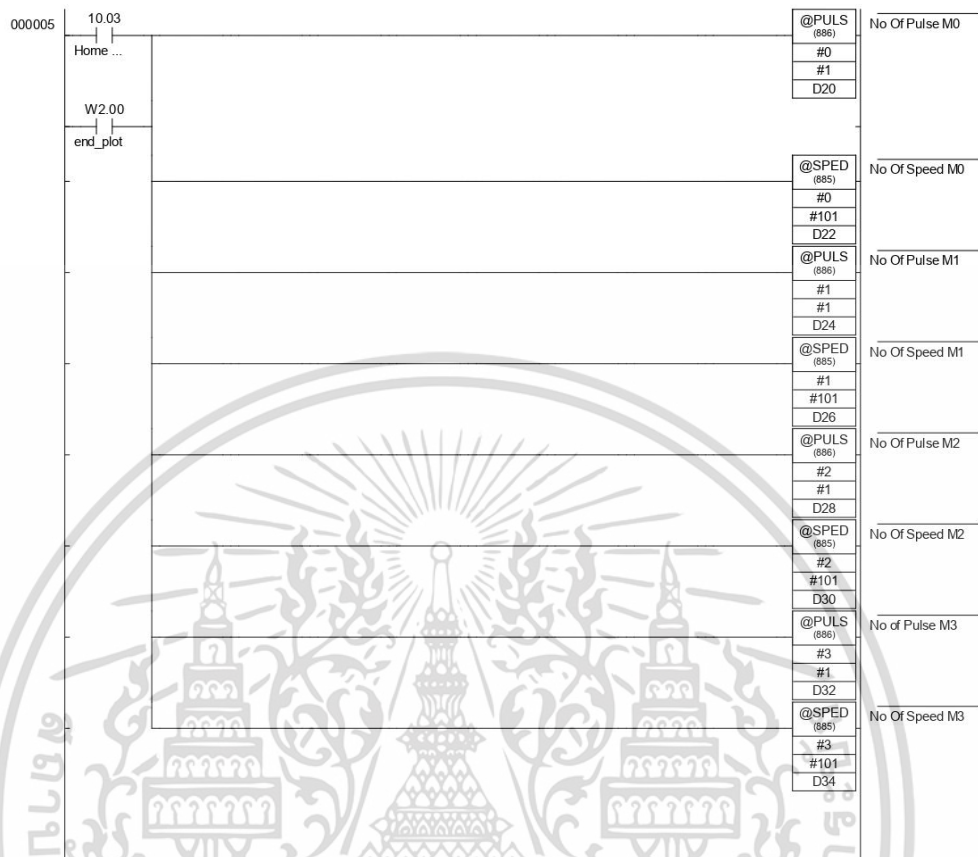
รูปที่ ง-9 คำสั่งในการกดสวิทช์เพื่อย้ายข้อมูลที่เป็นพัลส์ไปยังบิตที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

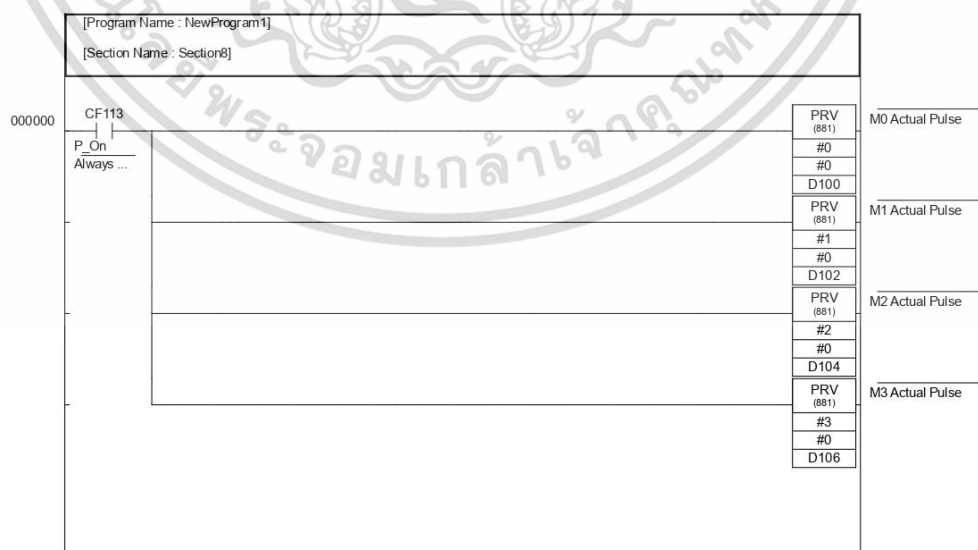


รูปที่ ง-10 ส่งค่าพัลส์และความเร็วไปขับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

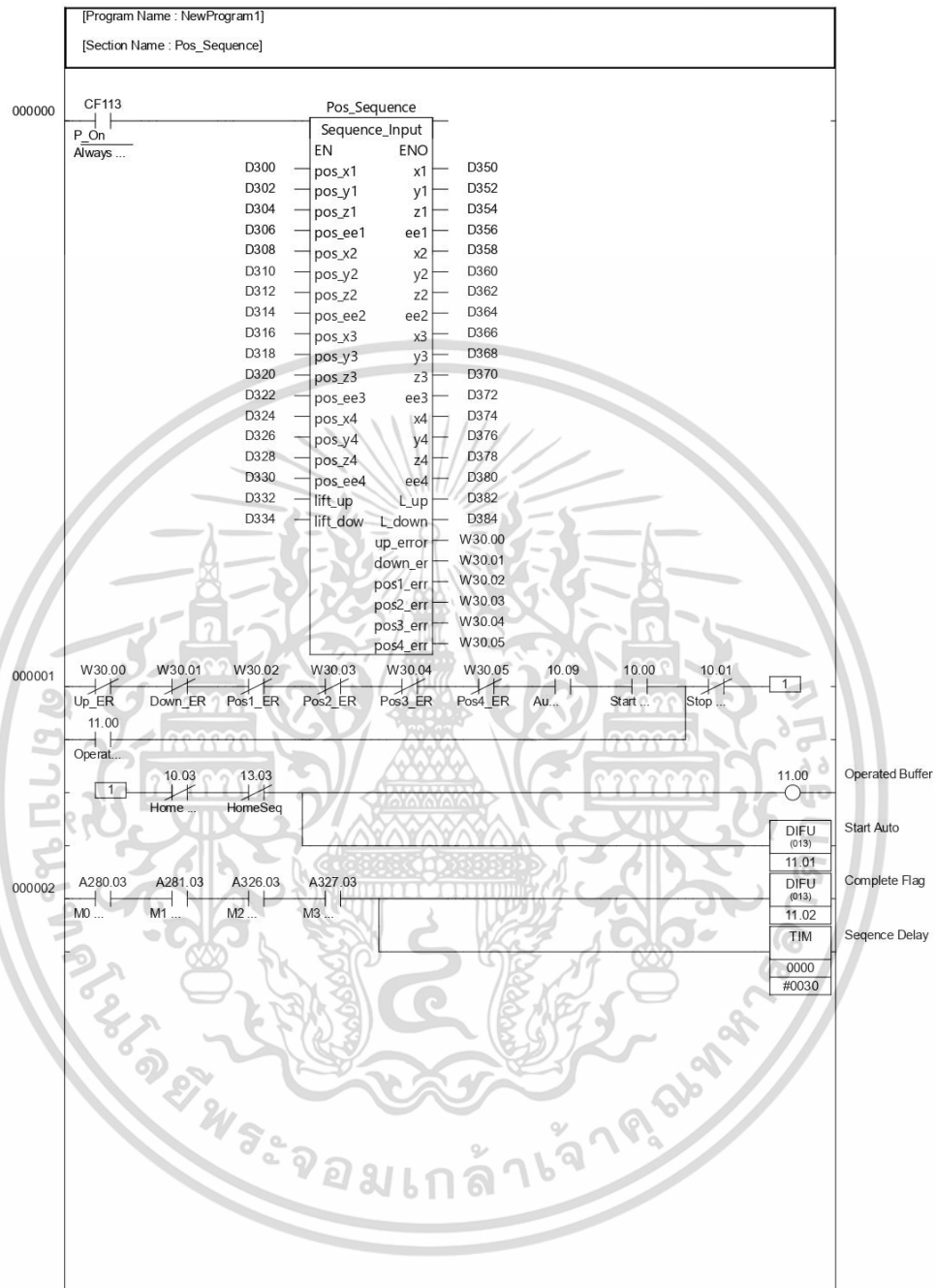


รูปที่ ง-11 ส่งค่าพัลส์และความเร็วไปขับเคลื่อนมอเตอร์ (ต่อ)



รูปที่ ง-12 วัดค่าการเคลื่อนที่ของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



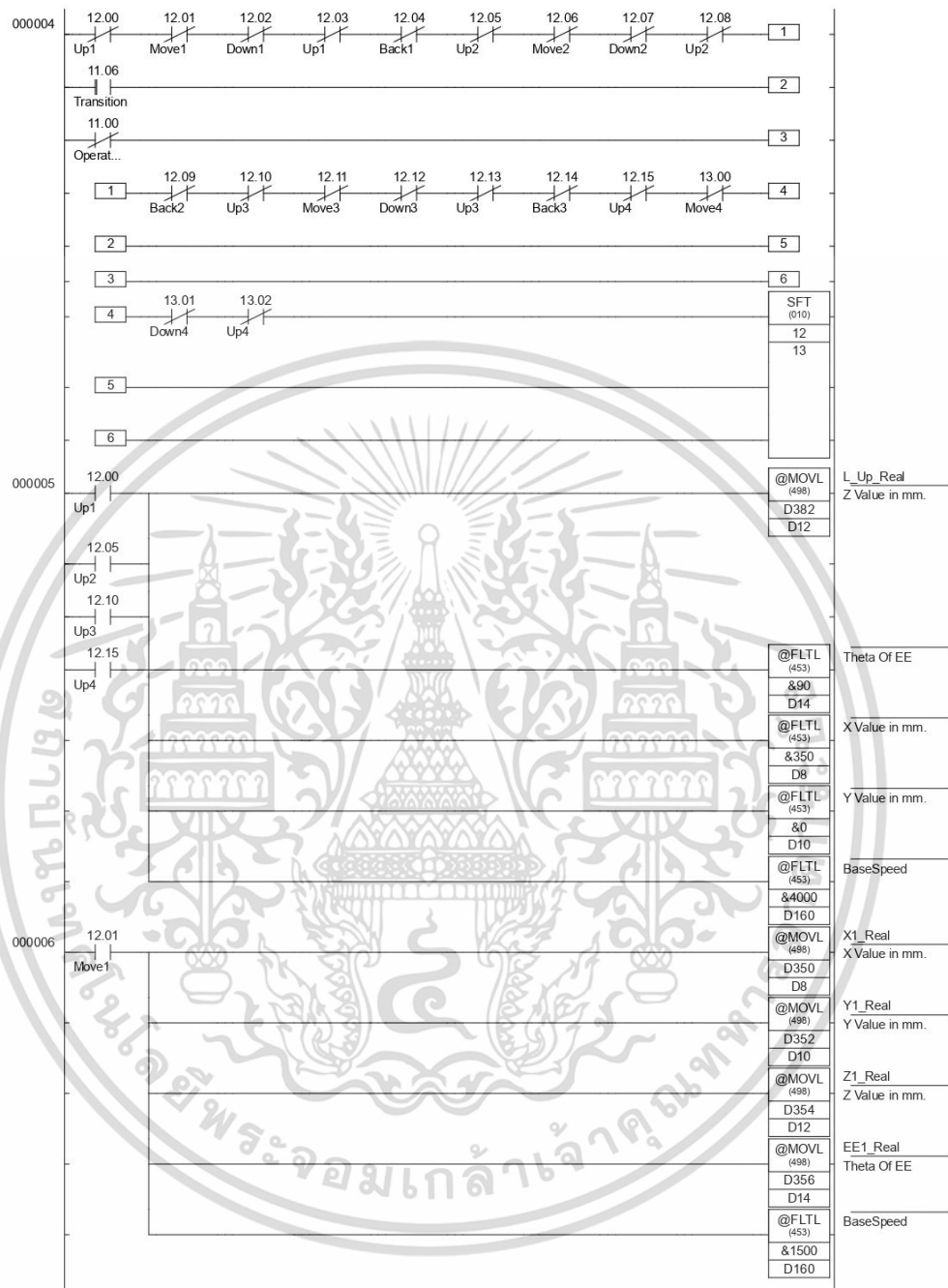
รูปที่ ง-13 คำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



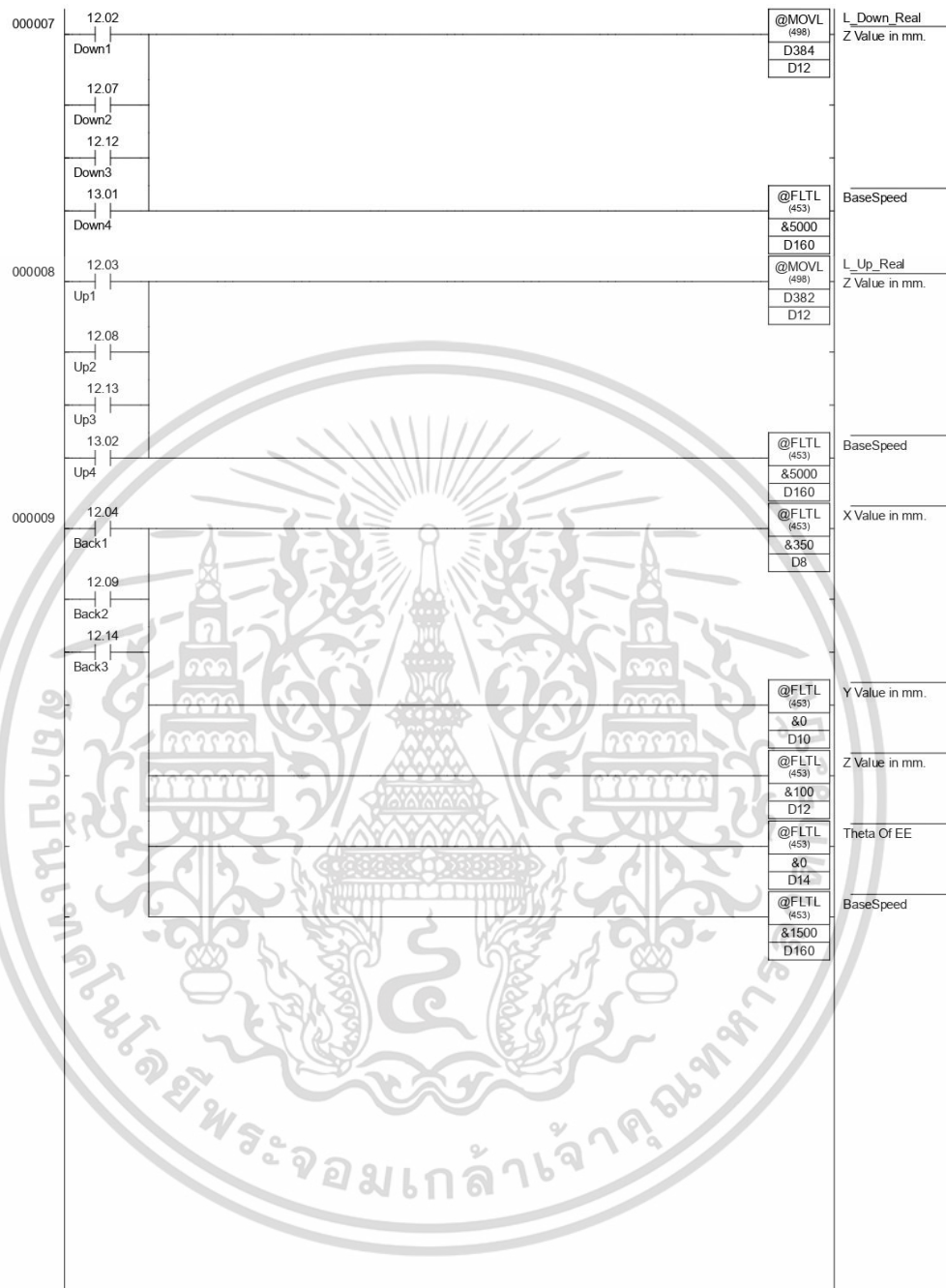
รูปที่ ง-14 คำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง-15 คำสั่งการควบคุมการทำงานของแกนกลแบบอัตโนมัติ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง-16 คำสั่งการควบคุมการทำงานของแกนกลแบบอัตโนมัติ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000010	12.06	Move2	@MOVL (498)	X2_Real X Value in mm.
			D358	
			D8	
			@MOVL (498)	Y2_Real Y Value in mm.
			D360	
			D10	
			@MOVL (498)	Z2_Real Z Value in mm.
			D362	
			D12	
			@MOVL (498)	EE2_Real Theta Of EE
			D364	
			D14	
			@FTLL (453)	BaseSpeed
			&1500	
			D160	
000011	12.11	Move3	@MOVL (498)	X3_Real X Value in mm.
			D366	
			D8	
			@MOVL (498)	Y3_Real Y Value in mm.
			D368	
			D10	
			@MOVL (498)	Z3_Real Z Value in mm.
			D370	
			D12	
			@MOVL (498)	EE3_Real Theta Of EE
			D372	
			D14	
			@FTLL (453)	BaseSpeed
			&1500	
			D160	
000012	13.00	Move4	@MOVL (498)	X4_Real X Value in mm.
			D374	
			D8	
			@MOVL (498)	Y4_Real Y Value in mm.
			D376	
			D10	
			@MOVL (498)	Z4_Real Z Value in mm.
			D378	
			D12	
			@MOVL (498)	EE4_Real Theta Of EE
			D380	
			D14	
			@FTLL (453)	BaseSpeed
			&1500	
			D160	

รูปที่ ง-17 คำสั่งการควบคุมการทำงานของแขนกลแบบอัตโนมัติ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้