



รายงานการวิจัยฉบับสมบูรณ์

การพัฒนาหุ่นยนต์อุตสาหกรรมด้วยซอฟต์แวร์โอเพนซอร์ส ROS
Development of the industry robot arm with an open-source Robot
Operating System (ROS)

นายคมกฤษ จักษุคำ

ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ 2560

วิทยาลัยนวัตกรรมการผลิตขั้นสูง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รายงานการวิจัยฉบับสมบูรณ์

การพัฒนาหุ่นยนต์อุตสาหกรรมด้วยซอฟต์แวร์โอเพนซอร์ส ROS
Development of the industry robot arm with an open-source
Robot Operating System (ROS)

นายคมกฤษ จักษุคำ

ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ 2560
วิทยาลัยนวัตกรรมการผลิตขั้นสูง
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ (ภาษาไทย)/ การพัฒนาหุ่นยนต์อุตสาหกรรมด้วยซอฟต์แวร์โอเพนซอร์ส ROS.....
แหล่งเงิน เงินรายได้.....
ประจำปีงบประมาณ 2560..... จำนวนเงินที่ได้รับการสนับสนุน 567,000..... บาท
ระยะเวลาทำการวิจัย 1..... ปี ตั้งแต่ 1 ต.ค. 59 ถึง 30 ก.ย. 60.....
ชื่อ-สกุล หัวหน้าโครงการ และผู้ร่วมโครงการวิจัย พร้อมระบุ หน่วยงานต้นสังกัด
นายคมกฤษ จักษ์คำ วิทยาลัยนวัตกรรมการผลิตขั้นสูง โทร. 02-329-8264 email: komgrit.ja@kmitl.ac.th.....
..... (หัวหน้าโครงการ).....

บทคัดย่อ

เนื่องด้วยปัจจุบัน ปัญหาการขาดแคลนแรงงานที่มีคุณภาพในภาคอุตสาหกรรมการผลิตเริ่มทวีความรุนแรงมากขึ้น การแก้ปัญหาดังกล่าวในเบื้องต้นภาคเอกชนหรือบริษัทขนาดใหญ่ที่มีงบประมาณสูงและพร้อมจะลงทุนด้านแขนกลอัตโนมัติ สามารถนำแขนกลอุตสาหกรรมมาช่วยเสริมแรงงานคนที่ขาดแคลนหรือแรงงานคนที่เหนื่อยล้าจากการทำงานติดต่อกันเป็นระยะเวลานานๆได้แต่สำหรับผู้ประกอบการธุรกิจขนาดกลางและขนาดย่อม (SME) ซึ่งมีจำนวนมากและจะเพิ่มจำนวนขึ้นในอนาคต ซึ่งมีงบประมาณค่อนข้างจำกัด นั้นถือว่าการลงทุนสูงและอาจมีระยะเวลาคู่มือทุนนานเนื่องจากแขนกลที่นำเข้าจากต่างประเทศซึ่งมีราคาค่อนข้างสูง เช่น แขนกลจากต่างประเทศที่ใช้กับโหลตไม่เกิน 3 กิโลกรัม นั้นราคามากกว่า 1 ล้านบาทเลยทีเดียว อีกทั้งค่าบำรุงรักษาสูงทั้งค่าแรงและค่าอะไหล่ และที่สำคัญที่สุดซอฟต์แวร์สำหรับแขนกลที่นำเข้าไม่สามารถแก้ไขได้ด้วยตัวผู้ซื้อเองในกรณีที่เกิดปัญหาหรือต้องการอัปเดตระบบดังนั้นการแก้ปัญหาของประเทศในภาพรวม คือผู้ผลิตในประเทศจะต้องมีความสามารถในการผลิตและออกแบบแขนกลอุตสาหกรรมขึ้นใช้เอง ซึ่งจะทำให้ราคาและอุปกรณ์ที่เกี่ยวข้องกับแขนกลทั้งหมดถูกลงมากเมื่อเทียบกับราคาที่สั่งซื้อจากต่างประเทศ เป็นการลดต้นทุนการผลิตของภาคอุตสาหกรรมต่างๆที่มีแผนจะใช้ระบบอัตโนมัติและเพิ่มความสามารถในการแข่งขันทางด้านธุรกิจกับต่างประเทศอีกทางหนึ่ง

โดยแขนกลอุตสาหกรรมที่ทางผู้วิจัยได้พัฒนาขึ้นด้วยเทคโนโลยีโอเพนซอร์สซึ่งจะเป็นระบบปฏิบัติการที่ควบคุมแขนกลอุตสาหกรรมให้สามารถเคลื่อนที่ได้ทั้ง 4 แกนสัมพันธ์กันและมีความคลาดเคลื่อนต่ำเหมาะกับการใช้งานในโรงงานอุตสาหกรรมภายในประเทศ โดยโปรแกรมที่ใช้ควบคุมการทำงานของแขนกลถูกเขียนขึ้นด้วยโปรแกรมภาษา C, C++ และ Python เป็นต้น ซึ่งทำงานบนระบบปฏิบัติการลินุกซ์ เชื่อมต่อระบบปฏิบัติการที่ควบคุมแขนกล (ROS) เข้ากับฮาร์ดแวร์ที่เกี่ยวข้องกับแขนกลต่างๆ

คำสำคัญ : แขนกลอุตสาหกรรม, เทคโนโลยีโอเพนซอร์ส, ระบบปฏิบัติการที่ควบคุมแขนกล

Research Title: Development of the industry robot arm with an open-source Robot Operating System (ROS)

Researcher: Komgrit Jaksukam

Faculty: College of Advance Manufacturing Innovation **Department:** -

ABSTRACT

The shortage of qualified workers in the manufacturing sector nowadays has intensified. The basic solution for private or large companies with high income is to invest in robotics and automation. Industrial robots can be used to help the labor or those who are tired of working for long periods of time. However, for small and medium-sized businesses (SMEs), which are growing in the future, the investment is quite limited and considered as a high investment and may have a long break-even period due to the relatively high cost of imported mechanical arms, such as foreign arms with a load of no more than 3 kilograms, cost more than 1 million baht ever. The high maintenance cost, both wage and spare parts and most importantly, the software for the imported mechanical arms, in case of problems or needed upgrade the system, cannot be modified by the users. To solve the problem of the country as a whole, domestic manufacturers must be able to produce and design their own industrial robots. This will make the price and equipment on the arm much lower compared to the price ordered from abroad. This will reduce the cost of production for the various industries that plan to use automation and increase the competitiveness of the international business.

The industrial robot arm was developed based on an open source technology, the operating system that controls the arm of the industrial arm to move the 4-axis and low tolerances, suitable for using in industrial abrasives for automotive industry. The program that controls the arm is written in C, C ++ and Python, which running on the Linux operating system and connecting the operating system that controls the arm (ROS) to the hardware associated with the arm.

Keywords : The industrial robot arm, open source technology, Robot Operating System (ROS)

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยดีจากความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่สามารถกล่าวได้ทั้งหมด ซึ่งผู้จัดทำใคร่ขอขอบคุณ รองศาสตราจารย์ ดร.ศิริเดช บุญแสง คณบดีวิทยาลัยนวัตกรรมการผลิตขั้นสูง ดร.สันหัต ฐวงคอินทร์ และนายธีรวัฒน์ ทองลอย ที่ได้ให้ความช่วยเหลือต่างๆ ในการทำโครงการนี้ และขอบคุณที่มิวิจัยทุกท่านของศูนย์หุ่นยนต์อุตสาหกรรมและระบบอัตโนมัติ (CiRA) ของวิทยาลัยนวัตกรรมการผลิตขั้นสูง ที่ได้ให้การช่วยเหลือและให้ความร่วมมือเป็นอย่างดีในการวิจัยครั้งนี้

การวิจัยครั้งนี้ได้รับทุนสนับสนุนการวิจัยจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จากแหล่งทุนงบประมาณรายได้ ประจำปีงบประมาณ พ.ศ. 2560



คมกฤษ จักษุคำ
วิทยาลัยนวัตกรรมการผลิตขั้นสูง
พ.ศ. 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ (ไทย).....	ก
บทคัดย่อ (อังกฤษ).....	ก
กิตติกรรมประกาศ.....	ก
สารบัญ	ข
สารบัญตาราง	ข
สารบัญภาพ	ข
บทที่ 1 : บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงการวิจัย	1
1.3 ขอบเขตของโครงการวิจัย	2
1.4 ระเบียบวิธีวิจัย	2
1.5 แนวทางการดำเนินการวิจัย	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 : แนวคิดทฤษฎี และงานวิจัยที่เกี่ยวข้อง.....	4
2.1 งานวิจัยที่เกี่ยวข้อง	4
2.2 ระบบปฏิบัติการหุ่นยนต์ (Robot Operating System)	13
2.3 อุปกรณ์ที่ใช้ในงานวิจัย	21
2.3.1 ชุต Servo drive.....	21
2.3.2 ชุตเอนโค้ดเดอร์ (Encoder).....	32
2.3.3 เอซีเซอร์โวมอเตอร์ (AC servo motor).....	33
บทที่ 3 : วิธีการวิจัย.....	37
3.1 แนวคิดของงานวิจัย	37
3.2 การออกแบบระบบปฏิบัติการแขนกลอุตสาหกรรม.....	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3	ระบบควบคุมหุ่นยนต์ที่ได้พัฒนาขึ้น.....	42
3.4	ขั้นตอนในการติดตั้ง ROS.....	44
บทที่ 4 : ผลการทดสอบ		47
4.1	การทดสอบควบคุมการทำงานของ SCARA robot ด้วย PID Control.....	47
4.2	การทดสอบการทำงานของหุ่นยนต์ SCARA robot	61
บทที่ 5 : สรุป		66
5.1	บทสรุป.....	66
5.2	ข้อเสนอแนะ.....	66
เอกสารอ้างอิง.....		67



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ค่าพารามิเตอร์ต่าง ๆ ของหุ่นยนต์	8
2.2 ค่าผิดพลาดสูงสุดของทุกมิติจะลดลงหลังจากการการหาค่าที่เหมาะสมที่สุด	13
2.3 แสดงพารามิเตอร์โครงสร้างของหุ่นยนต์ palletizing หลังจากการออกแบบที่ดีที่สุด	13
2.4 เอซีเซอร์โวมอเตอร์ของบริษัท Yaksakawa ที่ใช้ในการสร้างหุ่นยนต์.....	36



สารบัญตาราง

ภาพที่	หน้า
2.1 Grasp envelope ในการหยิบชิ้นงาน	4
2.2 Cluster Grasping	5
2.3 The forced diagram of the robot	7
2.4 The coordinates of a parallelogram	11
2.5 The mechanism schematic of palletizing robot.....	12
2.6 ตัวอย่างการใช้งานหยิบ-วางชิ้นงาน ในภาคอุตสาหกรรม.....	14
2.7 สมาชิกหลักของ ROS industrial consortium	15
2.8 การประยุกต์ใช้หลักๆ ของ I-ROS.....	16
2.9 เวอร์ชันของ ROS	17
2.10 รูปแบบ ROS communication.....	17
2.11 โลโก้ของ 5 ไลบรารีสำหรับการพัฒนาหุ่นยนต์	19
2.12 ซอฟต์แวร์จำลอง GAZEBO เป็นการจำลองการทำงานของหุ่นยนต์ (Robot simulation).....	19
2.13 Robot Geometry Library.....	20
2.14 Pose Estimation, Localization, and Navigation.....	20
2.15 Module MoveIt สำหรับการเชื่อมโยงกับหุ่นยนต์อุตสาหกรรม.....	20
2.16 ภาพรวมอุปกรณ์ที่ใช้สร้างหุ่นยนต์อุตสาหกรรม	21
2.17 ชุด Servo Drive.....	21
2.18 โหมดการควบคุมการเคลื่อนที่ของ AC Servo motor	22
2.19 โหมดซอฟต์แวร์ที่ใช้การควบคุมของชุด Servo Drive	23
2.20 ระบบควบคุมโดยทั่วไป	23
2.21 ลักษณะการควบคุมที่ใช้สัญญาณปฏิบัติการของระบบ	24
2.22 รูปแบบการควบคุมแบบ PID control ภายในชุด Servo Drive.....	24
2.23 สัญญาณการควบคุมแบบ PID control สำหรับควบคุมกำลังบิด (Torque control system)	25
2.24 การเชื่อมต่อสัญญาณระหว่างชุดควบคุมผ่านสาย CANBus.....	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.25 คำสั่งการทำงานแบบ S-curve และ trapeziodal profiles	26
2.26 การควบคุมการเคลื่อนที่ของมอเตอร์ที่สัมพันธ์กันกับเวลา.....	26
2.27 บล็อกไดอะแกรมของระบบการควบคุมการเคลื่อนที่	27
2.28 สัญญาณการเคลื่อนที่ด้วยกราฟแรงบิด (torque) และอัตราความเร็ว (speed)	27
2.29 ความสัมพันธ์ของความเร็ว ความเร่ง การลดความเร่ง	27
2.30 ตัวอย่างในการปรับค่าเกนสำหรับควบคุมการทำงานของมอเตอร์.....	28
2.31 ระบบการควบคุมอัตราความเร็วด้วยคำสั่งอัตราความเร็วอ้างอิง (Speed reference) เทียบกับอัตราความเร็วของมอเตอร์	28
2.32 รูปแบบการเคลื่อนที่ของหุ่นยนต์	29
2.33 รูปแบบการเคลื่อนที่ของหุ่นยนต์	30
2.34 ลักษณะคำสั่งควบคุมของของเกน Aff, Fff, Vff เมื่อเทียบกับคำสั่งควบคุมความเร็ว (velocity)..	32
2.35 สัญญาณเอนโค้ดเดอร์ (Encoder).....	33
2.36 ส่วนการลดสัญญาณรบกวน (Complementary signals).....	33
2.37 ส่วนประกอบของเอซีเซอร์โวมอเตอร์	34
2.38 โครงสร้างเอซีเซอร์โวมอเตอร์ (AC servo motor).....	34
2.39 การจ่ายกระแสให้กับขดลวดของเอซีเซอร์โวมอเตอร์ (AC servo motor).....	35
2.40 การควบคุมทิศทางเคลื่อนที่ของเซอร์โวมอเตอร์ของเอซีเซอร์โวมอเตอร์ (AC servo motor)..	35
3.1 แนวคิดในการพัฒนาหุ่นยนต์อุตสาหกรรมด้วยเทคโนโลยีโอเพ่นซอร์สสำหรับกระบวนการผลิต ในอุตสาหกรรม.....	37
3.2 หุ่นยนต์อุตสาหกรรม ABB IRB6640 ถูกใช้ทำงาน Palletizing ภายใต้ ROS-Industrial	38
3.3 ระบบปฏิบัติการที่ ROS และสามารถทำงานร่วมกับไลบรารีอื่นๆ	39
3.4 (a) สถาปัตยกรรมของ ROS industrial (b) ในแง่ของการสื่อสาร	40
3.5 การเชื่อม ROS industrial ผ่าน ROS Bridge กับ OpenCV	41
3.6 การจำลองการทำงานของหุ่นยนต์ (Robot simulation) ด้วย ROS.....	41
3.7 ไดอะแกรมการควบคุมของ Servo Motor ด้วยชุดขับเคลื่อนเซอร์โวมอเตอร์.....	42
3.8 ระบบควบคุมการทำงานของการทำงานของการควบคุมการทำงานของ Servo Motor r	42
3.9 ระบบควบคุมแบบปิดของ Servo Motor.....	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 ชุดควบคุมหุ่นยนต์ SCARA robot	43
3.11 การต่อวงจร Motor connection และ Incremental Encoder	44
4.1 โครงสร้างของการทำงานของ SCARA robot.....	47
4.2 โครงสร้างทางฮาร์ดแวร์ของหุ่นยนต์	48
4.3 ไอคอนโปรแกรม CME2	48
4.4 โปรแกรมชุดควบคุม Copley Motion.....	49
4.5 โปรแกรมชุดควบคุม Copley Motiont	49
4.6 หน้าต่างโปรแกรม CME2 ที่แสดงบล็อกไดอะแกรม PID control.....	50
4.7 สถานะการทำงานของชุดควบคุมมอเตอร์.....	50
4.8 แทนค่าพารามิเตอร์ของ Servo มอเตอร์ที่ได้.....	51
4.9 แทนค่าพารามิเตอร์ของมอเตอร์ที่ได้	52
4.10 หน้าต่างโปรแกรมของการคานวณหาค่า count.....	52
4.11 หน้าต่างโปรแกรมของการหาค่าพารามิเตอร์ Servo motor.....	53
4.12 หน้าต่างโปรแกรมของการหาค่า Gain current	53
4.13 ชุดควบคุมหาค่าที่เหมาะสมของผลตอบสนองทางระบบ.....	54
4.14 ผลตอบสนองระบบทางด้านแบนด์วิธของชุด Control.....	54
4.15 หน้าต่างโปรแกรมของเกรนระบบ	55
4.16 ผลตอบสนองสัญญาณที่ได้เทียบกับสัญญาณ Command current	56
4.17 หน้าต่างการตั้งค่า V Loop.....	57
4.18 หน้าต่างฟังก์ชัน Scope ของ Velocity Loop ด้วยการ Turning.....	58
4.19 หน้าต่างการตั้งค่า P Loop	59
4.20 หน้าต่างการตั้งปรับค่าเกรน PID Control	60
4.21 หน้าต่างฟังก์ชัน Scope ของ Position Loop ด้วยการ Turning.....	60
4.22 SCARA Robot ที่ผู้วิจัยได้พัฒนาขึ้นบนแพลตฟอร์ม ROS	61
4.23 การแสดงผลในแต่ละแกนของหุ่นยนต์ บนแพลตฟอร์ม ROS	62
4.24 การแสดงการจำลองการเคลื่อนที่ของหุ่นยนต์ บนแพลตฟอร์ม ROS	62
4.25 การแสดงการควบคุมการทำงานของหุ่นยนต์ผ่าน RViz	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.26 การติดตั้งหุ่นยนต์ที่เสร็จเรียบร้อยแล้ว ที่ศูนย์หุ่นยนต์อุตสาหกรรมและระบบอัตโนมัติ (CiRA).....	63
4.27 การติดตั้งหุ่นยนต์และชุดควบคุม Servo motor.....	64
4.28 การติดตั้งหุ่นยนต์ทั้งหมด ที่ศูนย์หุ่นยนต์อุตสาหกรรมและระบบอัตโนมัติ (CiRA)	65



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันการปฏิวัติอุตสาหกรรมครั้งที่ 4 ที่กำลังจะเกิดขึ้นนั้นได้ก่อให้เกิดการเปลี่ยนแปลงระบบทั้งระบบ ส่งผลต่อการพัฒนาเศรษฐกิจและอุตสาหกรรมภายในประเทศอย่างมาก โดยเฉพาะในกระบวนการผลิตของภาคส่วนอุตสาหกรรมที่ต้องอาศัยระบบการผลิตอัตโนมัติด้วยหุ่นยนต์ซึ่งถือว่าเป็นหัวใจหลักในการทำงานของระบบ เนื่องจากสามารถทำงานแทนมนุษย์ได้ อย่างไรก็ตามในการพัฒนาต้นแบบของแขนกลอุตสาหกรรมเพื่อใช้งาน อุตสาหกรรมการผลิตยังมีขั้นตอนที่ซับซ้อนและต้องอาศัยเวลามาก และการที่ซอฟต์แวร์ทำงานอยู่บนหน่วยประมวลผลเดี่ยว (Single processor) ทำให้เกิดข้อจำกัดเรื่องจำนวนของอุปกรณ์ที่จะติดตั้ง และการตอบสนอง การทำงานของอุปกรณ์เหล่านั้น ดังนั้นการที่ซอฟต์แวร์ที่กระจายการควบคุม (Distributed control architecture) โดยมีแพลตฟอร์มของหุ่นยนต์ที่หลากหลายมารองรับ จะทำให้การพัฒนาหุ่นยนต์มีประสิทธิภาพ และสะดวกรวดเร็ว ลดข้อจำกัดในการเชื่อมโยงและติดตั้งอุปกรณ์ต่าง ๆ บนหุ่นยนต์ได้ ดังนั้นงานวิจัยนี้จึงสนใจ ศึกษาและพัฒนาระบบควบคุมหุ่นยนต์โดยใช้ซอฟต์แวร์โอเพนซอร์สสำหรับแขนกลอุตสาหกรรม Robot Operating System (ROS) ซึ่งเป็นเครื่องมือหนึ่งที่มีความนิยมสำหรับการพัฒนาเทคโนโลยีแขนกล อุตสาหกรรม โดยมีเฟรมเวิร์กของซอฟต์แวร์ที่ทำงานหลากหลาย อาทิเช่น การทำงานของแขนกล ระบบการประมวลผลภาพ เซนเซอร์และตัวเข้ารหัสต่าง ๆ เป็นต้น เพื่อใช้เป็นต้นแบบในการพัฒนาหุ่นยนต์ในงาน อุตสาหกรรมการผลิตต่อไป

1.2 วัตถุประสงค์ของโครงการวิจัย

1. เพื่อศึกษาหลักการทำงานของอุปกรณ์ชุดควบคุม มอเตอร์ รวมถึงทฤษฎีการขับเคลื่อนขั้นพื้นฐานที่จำเป็น สำหรับการพัฒนาหุ่นยนต์
2. เพื่อพัฒนาปรับปรุงชุดควบคุมหุ่นยนต์ให้มีประสิทธิภาพดีขึ้น
3. เพื่อเป็นอุปกรณ์ประกอบการเรียนการสอนของนักศึกษาภายในวิทยาลัยฯ
4. เพื่อใช้เป็นต้นแบบในพัฒนาหุ่นยนต์ต้นทุนต่ำ
5. เพื่อเป็นการสนับสนุนการวิจัยและพัฒนาเทคโนโลยีการผลิตด้วยแขนกลอุตสาหกรรม

1.3 ขอบเขตของโครงการวิจัย

โครงการวิจัยนี้ได้ให้ความสำคัญในการปรับปรุงระบบขับเคลื่อนหุ่นยนต์ Epson ซึ่งประกอบไปด้วยมอเตอร์ไฟฟ้าที่ทำหน้าที่ในการเคลื่อนแขนหุ่นยนต์ไปยังตำแหน่งต่างๆ ทั้งการหมุน (Rotation) ในแนวแกน X , Y , Z และการเลื่อน (Displacement) และเซนเซอร์เอ็นโค้ดเดอร์ (Encoder) รวมทั้งระบบการขับเคลื่อน (Gear) ของหุ่นยนต์ชุดเดิมให้สามารถทำงานได้อย่างปกติ โดยใช้ซอฟต์แวร์โอเพนซอร์สสำหรับหุ่นยนต์ Robot Operating System (ROS) ซึ่งอาศัยระบบปฏิบัติการลินุกซ์มารองรับการทำงานร่วมกัน ข้อดีคือเป็นระบบปฏิบัติการโอเพนซอร์สสำหรับหุ่นยนต์ (ROS) ทำให้ง่ายต่อการวิจัยและพัฒนาต่อยอดต่อไป

1.4 ระเบียบวิธีวิจัย

1. ดำเนินการจัดซื้อวัสดุอุปกรณ์
2. ติดตั้งมอเตอร์ไฟฟ้าเข้ากับหุ่นยนต์ Epson
3. พัฒนาชุดควบคุมการทำงานด้วยระบบปฏิบัติการ ROS
4. ปรับปรุงแก้ไขและทดสอบประสิทธิภาพการทำงาน
5. สรุปรายและรายงานการวิจัย

1.5 แนวทางการดำเนินการวิจัย

งานวิจัยมีขั้นตอนดังต่อไปนี้

1. ดำเนินการจัดซื้อวัสดุอุปกรณ์
2. ติดตั้งมอเตอร์ไฟฟ้าเข้ากับหุ่นยนต์ Epson
3. พัฒนาชุดควบคุมการทำงานด้วยระบบปฏิบัติการ ROS
4. ปรับปรุงแก้ไขและทดสอบประสิทธิภาพการทำงาน
5. สรุปรายและรายงานการวิจัย

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ก่อให้เกิดองค์ความรู้ที่เกี่ยวข้องกับการพัฒนาระบบควบคุมหุ่นยนต์โดยใช้ซอฟต์แวร์โอเพนซอร์สสำหรับแขนกลอุตสาหกรรม
2. เป็นแขนกลอุตสาหกรรมต้นแบบในการเรียนการสอนและปฏิบัติการในศูนย์หุ่นยนต์อุตสาหกรรมและระบบอัตโนมัติของวิทยาลัยนวัตกรรมการผลิตขั้นสูง
3. เป็นการพัฒนาส่วนควบคุมหุ่นยนต์อุตสาหกรรมด้วยตำแหน่ง ทิศทาง การเคลื่อนที่ของหุ่นยนต์ได้
4. เป็นการพัฒนาหุ่นยนต์ด้วยเทคโนโลยีโอเพนซอร์สเป็นระบบปฏิบัติการที่ถูกพัฒนาขึ้นเพื่อต่อยอดการทำงานในอุตสาหกรรมการผลิตขั้นภายในประเทศ
5. เป็นแหล่งเผยแพร่และประชาสัมพันธ์ให้แก่ผู้ประกอบการที่เกี่ยวข้องได้เยี่ยมชม และเรียนรู้การพัฒนาหุ่นยนต์อุตสาหกรรมมาเพิ่มศักยภาพและประสิทธิภาพการผลิตแก่ผู้ที่สนใจ ซึ่งจะทำให้เป็นส่วนหนึ่งของการประชาสัมพันธ์ความสำเร็จในการปรับตัวของภาคอุตสาหกรรมไทยเข้าสู่ยุค Industry 4.0

บทที่ 2

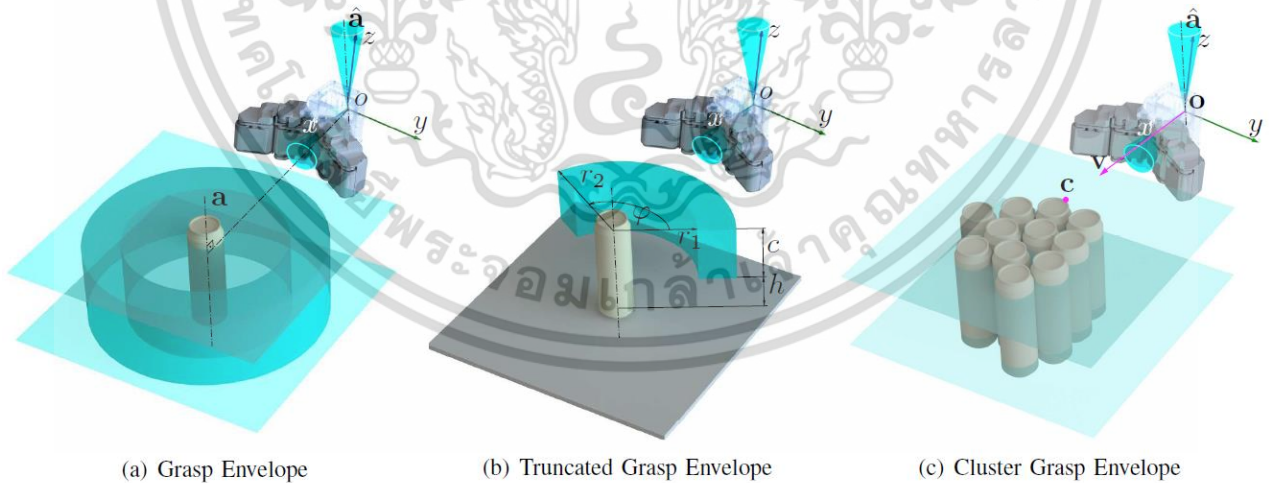
แนวคิดทฤษฎี และงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้กล่าวถึงรายละเอียดของการศึกษาค้นคว้าความรู้พื้นฐานสำหรับการพัฒนาแขนกลอุตสาหกรรม ระบบปฏิบัติการลินุกซ์ที่เชื่อมต่อระบบปฏิบัติการที่ควบคุมหุ่นยนต์ (ROS) และงานวิจัยที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

จากงานวิจัยของ Robert Krug และคณะ [1] เกี่ยวกับการทดสอบการใช้งานหุ่นยนต์ด้าน Picking & Palletizing ซึ่งยังมีประเด็นด้าน ความเร็ว ความปลอดภัย และความแม่นยำของหุ่นยนต์ในการทำงานร่วมกับคน อยู่ ในงานวิจัยดังกล่าวนี้จะโฟกัสไปที่การหยิบจับชิ้นงานในโรงงานที่ถูกติดตั้งขึ้นแบบง่ายๆ โดยเน้นที่หัวข้อสำคัญ ดังต่อไปนี้

1) เรื่องของการควบคุมโคออร์ดิเนตเพื่อไปหา pose ของชิ้นงานแบบ on-the-fly ซึ่งจากการทดสอบ ใช้เวลาโดยเฉลี่ยในการหยิบจับชิ้นงานที่ประมาณ 23.5 วินาที ที่ 94.7% สามารถหยิบได้ ระบบดังกล่าวสามารถทำงานตามคำสั่งง่ายๆในสภาวะที่ปลอดภัยกับคนที่ทำงานร่วมด้วยและจะเป็นจุดเริ่มต้นระบบอัตโนมัติด้านโลจิสติกส์เชิงพาณิชย์ในอนาคต



รูปที่ 2.1 Grasp envelope ในการหยิบชิ้นงาน

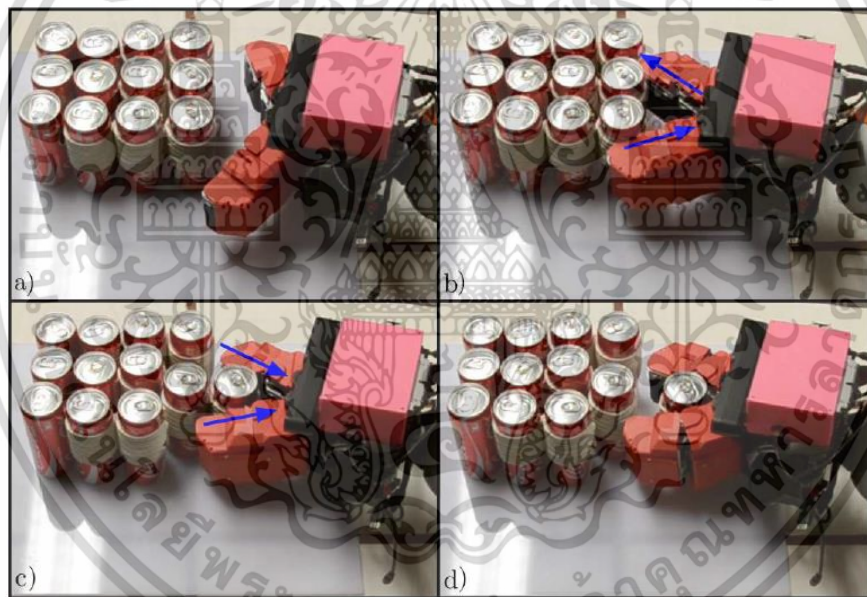
2) หัวข้อที่ถูกแสดงในรูปที่ 2.1 ซึ่งแสดงข้อจำกัดในการหยิบชิ้นงาน (Grasp Constraints) ในรูป a) บริเวณสีคล้ำแสดงด้านที่เป็น grasp envelope constraints สำหรับวัตถุทรงกระบอก หุ่นยนต์จะสามารถหยิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชิ้นงานได้ก็ต่อเมื่อตำแหน่ง Gripper ที่เฟรม o ต้องอยู่ภายในโครงของรูปทรงกระบอกที่ได้แนวกับแกนวัตถุ a ความสูงของทรงกระบอกถูกจำกัดด้วยระนาบที่ตั้งฉากกับวัตถุ a นอกจากนี้แนวแกนตั้งของ Gripper (z) ยังต้องอยู่ในรูปทรงกรวยโดยที่แกน a จะขนานไปกับแกนวัตถุ a . นอกจากนี้แกน x ของ Gripper ต้องอยู่ในรูปกรวยตรงกึ่งกลางที่เชื่อมต่อแกน a กับจุด o

ในรูป b) ในระหว่างการทำงานนั้น grasp envelope ที่แสดงในรูป 2.2 (a) จำเป็นต้องถูก truncated (โดยใช้พารามิเตอร์สำหรับ truncated ได้แก่ r_1, r_2, c, h และ (p)) เพื่อให้เหมาะสมกับกับมิติของวัตถุและอธิบายบางบริเวณของ grasp envelope ซึ่งอาจจะเป็นไปได้เนื่องจากการถูกกีดขวางโดยสภาพแวดล้อม

ในรูป c) Gripper ที่เฟรม o ต้องอยู่ระหว่างระนาบที่มีสีฟ้า แกน z ของ Gripper จะอยู่ในกรวยที่มีแกน a เป็นแนวตั้ง นอกจากนี้แกน x ของ Gripper ต้องอยู่ในรูปกรวยตรงกึ่งกลางของแนวแกน z และแกน z ต้องชี้ไปที่จุดที่ใกล้ที่สุด c บนคาล์สเตอร์ซึ่งถูกแสดงในรูปที่ 2.2



รูปที่ 2.2 Cluster Grasping

ทางคณะวิจัยได้อธิบายองค์ประกอบของข้อต่อหุ่นยนต์โดยใช้ค่าเวกเตอร์ q และอินพุทควบคุมที่สัมพันธ์กับความเร็วของข้อต่อ \dot{q} ฟังก์ชัน $e(q)$ คือฟังก์ชันที่ขึ้นอยู่กับ q ตัวอย่างเช่น งานที่มีวัตถุประสงค์ของการนำจุด end effector $p(q)$ บนระนาบที่ถูกอธิบายด้วยเวกเตอร์หน่วยที่ตั้งฉาก n และออฟเซต d โดยฟังก์ชัน $e(q)$ สามารถสร้างเป็นสมการได้ดังต่อไปนี้

$$e(q) = n^T p(q) - d \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นสูตร projection residual ที่เหลือระหว่างระนาบกับ $p(q)$ งานที่เกิดขึ้นจะถูกแสดงได้ดังสมการด้านล่าง

$$J\dot{q} = \dot{e} \quad (2.2)$$

โดยที่จาโคเบียน $J = \frac{\partial e}{\partial q}$ เป้าหมายคือการคำนวณความเร็วข้อต่อ จนกว่างานที่ได้เป็นไปตามที่ต้องการ \dot{e}^* (ในงานวิจัยนี้เลือกเป็น (exponential decay) $\dot{e}^* = -\lambda e$ โดยที่ $\lambda \in R_+$) สำหรับงานแบบ single equality จำเป็นต้องแก้ปัญหาโดยใช้ least squares Quadratic Program (QP)

$$\dot{q}^* \in \arg \min_{\dot{q}} \|J\dot{q} - \dot{e}^*\| \quad (2.3)$$

ทางคณะวิจัยเลือกใช้ขอบเขตบนของสมการงานใน (2) ดังต่อไปนี้

$$J\dot{q} \leq \dot{e}^* \quad (2.4)$$

ถ้าข้อจำกัด ใน (2.4) เป็นไปไม่ได้ least squares solution สำหรับ \dot{q}^* ตามที่ (2.3) สามารถหาได้โดยใช้ตัวแปร w ใน decision variables

$$\min_{\dot{q}, w} \|w\| \text{ ภายใต้เงื่อนไข } J\dot{q} \leq \dot{e}^* + w \quad (2.5)$$

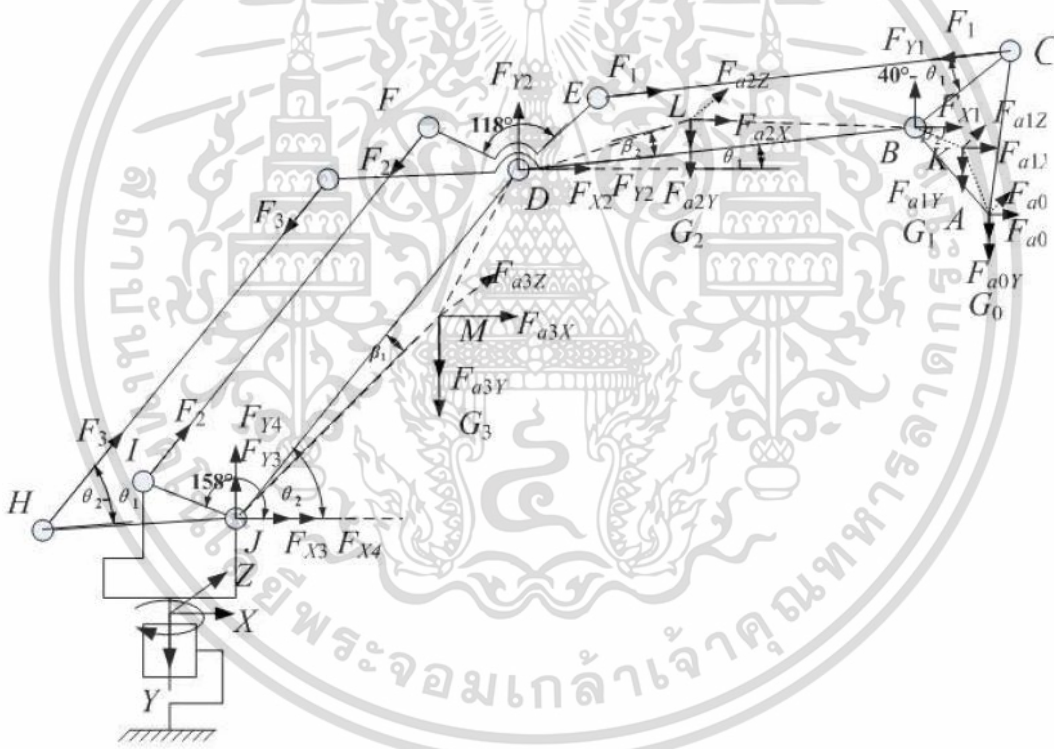
ทางคณะวิจัยเก็บ jacobians ทั้งหมดของงาน ใน (4) ด้วยการกำหนดเป็นเมทริกซ์ A_p และความเร็วอ้างอิงที่เกี่ยวข้องทั้งหมดในเวกเตอร์ b_p เพื่อสร้างสมการ $A_p \dot{q} \leq b_p$ สำหรับแต่ละชั้น จุดมุ่งหมายคือตอบสนองข้อจำกัดแต่ละชั้นที่กล่าวมาให้ได้ที่สุดใน least square ดังนั้น QP สำหรับตัวแปรก่อนหน้า w_i^* และให้ $p = 1, \dots, P$ คือ

$$\min_{\dot{q}, w_p} \|w_p\| \text{ ภายใต้เงื่อนไข } A_i \dot{q} \leq b_i + w_i^* \quad , i = 1, \dots, p - 1$$

$$A_p \dot{q} \leq b_p + w_p \quad (2.6)$$

ซึ่งเวกเตอร์ควบคุม \dot{q}^* จะหาได้จาก P^{th} solution ของ (2.6)

สำหรับงานวิจัยของ Lv Mei-yu และคณะ [2] เกี่ยวกับการวิเคราะห์พลวัตของหุ่นยนต์ โดยทางที่วิจัยสร้างแบบจำลอง dynamic statics ของหุ่นยนต์ palletizing แบบจำลองนี้แสดงความสัมพันธ์ระหว่างการเคลื่อนไหวและแรงที่สามารถนำมาใช้ในการวิเคราะห์สถานะแรงในหุ่นยนต์อื่น ๆ ที่คล้ายกัน ดังแสดงในรูปที่ 2.3 ทางที่วิจัยใช้ โปรแกรม MATLAB วิเคราะห์แรงของหุ่นยนต์ในการทำงาน และให้ข้อมูลที่สำคัญสำหรับการตรวจสอบโครงสร้าง การปรับปรุงการออกแบบและการควบคุมไดนามิกตามลำดับ



รูปที่ 2.3 The forced diagram of the robot

จากรูปที่ 2.3 จะสังเกตว่าหุ่นยนต์มี BCED, DFJ, DGHJ เป็นรูปสี่เหลี่ยมด้านขนาน, DE และ DF มีความยาวเท่ากัน ในรูปที่ 2.3 แสดงภาพการวิเคราะห์แรงของหุ่นยนต์ palletizing ซึ่ง θ_1 , θ_2 แสดงถึงการหมุนของท่อนแขนเทียบกับแกนนอน; G_0 , G_1 , G_2 , G_3 ตามลำดับสำหรับแรงโน้มถ่วงของโหลด คือปลายของตัวขับเคลื่อน,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท่อนแขน K, L, M หมายถึงศูนย์กลางของตำแหน่งมวลของปลายของตัวข้อรับรื้อ, ท่อนแขน สำหรับพารามิเตอร์ ในรูปที่ 2.3 แสดงในตารางที่ 2.1 ด้านล่าง

ตารางที่ 2.1 ค่าพารามิเตอร์ต่าง ๆ ของหุ่นยนต์

parameter	LBC	LCE	LDE	LDJ	LHJ	m0	m1	m2	m3
Value	450Mm	1350 mm	450mm	1280 mm	500 Mm	180Kg	71.7 Kg	284.9Kg	367.6Kg
parameter	$\angle EDF$	$\angle IJN$	β_1	β_2	β_3	LDL	LJM	θ_1	θ_2
Value	118°	158°	12.6°	36.0°	6.8°	280.7mm	589.0mm	(-120° ,20°)	(20°,132°)

CE, GH, FI คือ two-force link, F1, F2, F3 คือแรง ตามลำดับ

Faix, Faiy, Faiz : แรงเฉื่อยของโหนดที่ปลายของตัวข้อรับรื้อ ท่อนแขน X, Y, Z ซึ่ง $i=0, 1, 2, 3$

Fxi, Fyi : แรงที่จุด B, D, J ในทิศทาง X, Y ซึ่ง $i=1, 2, 3, 4$

Fx3, Fy3, Fx4, Fy4 : แรงของแขน reducer shaft.

M3 , M4 : โมเมนต์ของแรงที่เกิดจากมอเตอร์ของท่อนแขนและ reducer

โดยจะพิจารณาเพียงแรงเฉื่อยของโหนด ปลายของตัวข้อรับรื้อ ท่อนแขนโดยแรงเฉื่อยและโมเมนต์ที่เกิดจากอย่างอื่น ๆ จะไม่นำมาพิจารณาเนื่องจากมีค่าน้อยมาก

ตามแผนภาพการวิเคราะห์แรง แต่ละข้อต่อสามารถแสดงโดยสมการสมดุลของแรงและโมเมนต์ โดยใช้หลักการ D'Alembert ถ้าใช้ ΔABC แรงและสมการสมดุลโมเมนต์ บนแกน B: จะเป็นดังนี้

$$\begin{cases} F_1 \times L_{BC} \times \sin(40^\circ - \theta_1) + F_{a0x} \times L_4 + F_{a1x} \times L_{y1} = (F_{a0y} + G_0) \times L_3 + (F_{a1y} + G_1) \times L_{x1} \\ F_{x1} = F_1 \times \cos \theta_1 - F_{a0x} - F_{a1x} \\ F_{y1} = F_1 \times \sin \theta_1 + (F_{a0y} + G_0) + (F_{a1y} + G_1) \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าใช้ ΔDEF แรงและสมการสมดุลโมเมนต์ บนแกน D: จะเป็นดังนี้

$$\begin{cases} F_1 \times L_{BC} \times \sin(40^\circ - \theta_1) = F_2 \times L_{BC} \times \sin(158^\circ - \theta_2) \\ F_{x21} = F_2 \times \cos \theta_2 - F_1 \times \cos \theta_1 \\ F_{y21} = F_2 \times \sin \theta_2 - F_1 \times \sin \theta_1 \end{cases}$$

ถ้าใช้ BG แรงและสมการสมดุลโมเมนต์ บนแกน D: จะเป็นดังนี้

$$\begin{cases} F_{y1} \times L_{BD} \times \cos \theta_1 + F_3 \times L_{GD} \times \sin(\theta_2 - \theta_1) = F_{a2x} \times L_1 \times \sin(\theta_1 + \beta_2) \\ + (F_{a2y} + G_2) \times L_1 \times \cos(\theta_1 + \beta_2) + F_{x1} \times L_{BD} \times \sin \theta_1 \\ F_{x22} = F_3 \times \cos \theta_2 - F_{x1} - F_{a2x} \\ F_{y22} = F_3 \times \sin \theta_2 + (G_2 + F_{a2y}) - F_{y1} \end{cases}$$

ถ้าใช้ DJ แรงและสมการสมดุลโมเมนต์ บนแกน J: จะเป็นดังนี้

$$\begin{cases} M_3 + F_{y22} \times L_{JD} \times \cos \theta_2 = F_{x22} \times L_{JD} \times \sin \theta_2 + F_{a3x} \times L_2 \times \sin(\theta_2 - \beta_3) \\ + (F_{a3y} + G_3) \times L_2 \times \cos(\theta_2 - \beta_3) \\ F_{x3} = -F_{a3x} - F_{x22} \\ F_{y3} = (G_3 + F_{a3y}) - F_{y22} \end{cases}$$

ถ้าใช้ HJ แรงและสมการสมดุลโมเมนต์ บนแกน J: จะเป็นดังนี้

$$\begin{cases} M_4 = F_3 \times L_{GD} \times \sin(\theta_2 - \theta_1) \\ F_{x4} = -F_3 \times \cos \theta_2 \\ F_{y4} = -F_3 \times \sin \theta_2 \end{cases}$$

ใน ΔABC , ตามที่ ΔABC มีระนาบการเคลื่อนที่ การเคลื่อนที่ของแต่ละจุดและจุด B มีความเหมือนกัน ขึ้นอยู่กับ ทฤษฎีของความเร่ง โดยความเร่งของจุด B คือ:

$$\begin{cases} \vec{a}_B = \vec{a}_D + \vec{a}_{BD} + \vec{a}_{c1} \\ \vec{a}_{c1} = 2 \vec{\omega}_D \times \vec{v}_{BD} \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นความเร่งของปลายของตัวข้อรับรื้อ และโหนด a0, a1 คือ:

$$\begin{cases} a_{0x} = a_{1x} = \alpha_3 \times L_{JD} \times \sin \theta_2 + \omega_3^2 \times L_{JD} \times \cos \theta_2 + \alpha_4 \times L_{BD} \times \sin \theta_1 \\ \quad + (\omega_4^2 \times L_{BD} + 2 \times \omega_3 \times \omega_4 \times L_{BD}) \times \cos \theta_1 \\ a_{0y} = a_{1y} = \alpha_3 \times L_{JD} \times \cos \theta_2 - \omega_3^2 \times L_{JD} \times \sin \theta_2 + \alpha_4 \times L_{BD} \times \cos \theta_1 \\ \quad - (\omega_4^2 \times L_{BD} + 2 \times \omega_3 \times \omega_4 \times L_{BD}) \times \sin \theta_1 \end{cases}$$

ศูนย์กลางมวลของปลายแขน L หมุนรอบแกน D และ D หมุนไปรอบ ๆ แกน J ตามทฤษฎีบทขององค์ประกอบของความเร่ง:

$$\begin{cases} \vec{a}_2 = \vec{a}_D + \vec{a}_{LD} + \vec{a}_{c2} \\ \vec{a}_{c2} = 2 \vec{\omega}_3 \times \vec{v}_{LD} \end{cases}$$

จากนั้นทางที่มิวิจัหาความเร่งของศูนย์กลางของมวลของปลายแขน a2 คือ:

$$\begin{cases} a_{2x} = \alpha_3 \times L_{JD} \times \sin \theta_2 + \omega_3^2 \times L_{JD} \times \cos \theta_2 + \alpha_4 \times L_1 \times \sin(\theta_1 + \beta_2) \\ \quad + (\omega_4^2 \times L_1 + 2 \times \omega_3 \times \omega_4 \times L_1) \times \cos(\theta_1 + \beta_2) \\ a_{2y} = \alpha_3 \times L_{JD} \times \cos \theta_2 - \omega_3^2 \times L_{JD} \times \sin \theta_2 + \alpha_4 \times L_1 \times \cos(\theta_1 + \beta_2) \\ \quad - (\omega_4^2 \times L_1 + 2 \times \omega_3 \times \omega_4 \times L_1) \times \sin(\theta_1 + \beta_2) \end{cases}$$

ศูนย์กลางมวลของแขน M จะหมุนรอบแกน J เท่านั้น ดังนั้นความเร่งของศูนย์กลางของมวลของแขน a3 คือ:

$$\begin{cases} a_{3x} = \alpha_3 \times L_2 \times \sin(\theta_2 - \beta_3) + \omega_3^2 \times L_2 \times \cos(\theta_2 - \beta_3) \\ a_{3y} = \alpha_3 \times L_2 \times \cos(\theta_2 - \beta_3) - \omega_3^2 \times L_2 \times \sin(\theta_2 - \beta_3) \end{cases}$$

ซึ่งรู้ได้จากกฎข้อที่สองของนิวตัน:

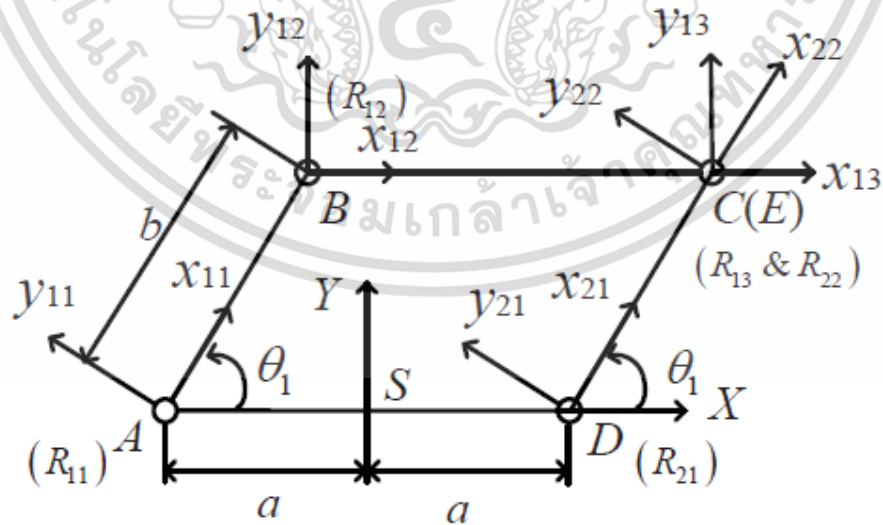
$$\begin{cases} F_{a0x} = m_0 \times a_{0x} & F_{a1x} = m_1 \times a_{1x} & F_{a2x} = m_2 \times a_{2x} & F_{a3x} = m_3 \times a_{3x} \\ F_{a0y} = m_0 \times a_{0y} & F_{a1y} = m_1 \times a_{1y} & F_{a2y} = m_2 \times a_{2y} & F_{a3y} = m_3 \times a_{3y} \end{cases}$$

สมการทั้งหมดด้านบน สร้างเป็นแบบจำลอง dynamics statics ทางคณิตศาสตร์ของหุ่นยนต์ palletizing

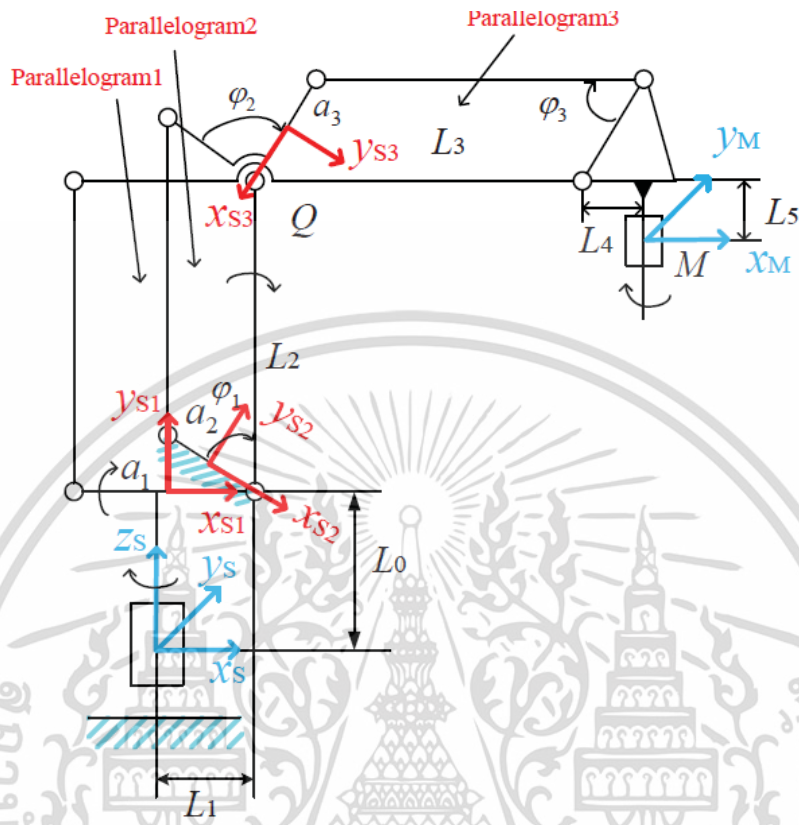
เราจะเห็นว่าสมการมีเพียง θ_1, θ_2 สองตัวแปร, ส่วนค่า $M_3, M_4, FJ_3, FJ_4, FD_1, FD_2, FB$ เป็นฟังก์ชันของ θ_1, θ_2 โดยการเขียนโปรแกรมใน MATLAB เพื่อหาค่าเหล่านี้ เราจะได้แรงของแต่ละแกนหลักของหุ่นยนต์ palletizing ในการทำงานใด ๆ

นอกจากงานวิจัยของ Wang และคณะ [3] เกี่ยวกับการวิเคราะห์ความถูกต้องและการออกแบบที่เหมาะสมของ Palletizing robot ด้วย Joint Clearance ในบทความนี้เสนอวิธีการคำนวณ *pose* (ตำแหน่งและทิศทาง (position and orientation)) error ของหุ่นยนต์ palletizing บนพื้นฐานของ Joint Clearance โดยการแบ่งกลไกของหุ่นยนต์เป็น 3 กลไกของ *special four-link* หรือ *parallelograms* ดังรูปที่ 2.4 ,มันเป็นเรื่องง่ายที่จะศึกษาการ *error transformation* ของทั้งสามข้อต่อที่ขนานกันด้วยจุดตัดของสามรูปสี่เหลี่ยมขนมเปียกปูน (parallelograms) ดังรูปที่ 2.5 ดังนั้น แบบจำลอง *pose error* สร้างได้โดยใช้วิธี *parallel manipulators accuracy analysis* จากโมเดลนี้ค่าสูงสุดของ *pose error* ของหุ่นยนต์ palletizing ในพื้นที่ทำงานใดๆ สามารถหาได้ มันเป็นพื้นฐานสำหรับการออกแบบโครงสร้างด้วยพารามิเตอร์ที่ดีที่สุด ในตัวอย่างทางการคำนวณเชิงตัวเลข ความถูกต้องของหุ่นยนต์ palletizing จะสูงขึ้นโดยการลด *error* ของทุกมิติหลังจากการหาค่าที่เหมาะสมที่สุด ดังนั้นผลของการออกแบบที่ดีที่สุดนี้สามารถปรับปรุงความแม่นยำของหุ่นยนต์ palletizing ซึ่งไม่เปลี่ยนแปลงสมบัติทางจลนศาสตร์ เช่น ความสามารถเข้าถึงได้ในพื้นที่ทำงาน, การใช้งาน (*manipulate*) และอื่น ๆ

ในการทำงานในอนาคต จะมีการพิจารณาดังนี้เพิ่มขึ้นในการออกแบบที่ดีที่สุดเพื่อปรับปรุงคุณสมบัติของความเร็วและความเร่งสำหรับหุ่นยนต์ palletizing ด้วยวิธีนี้ทางที่วิจัยสามารถหาโครงสร้างที่ดีที่สุดของหุ่นยนต์ palletizing เพื่อให้มั่นใจว่าสามารถประยุกต์ใช้กับงานด้านอื่นได้หลากหลายมากขึ้น



รูปที่ 2.4 The coordinates of a parallelogram



รูปที่ 2.5 The mechanism schematic of palletizing robot

โดยการวิเคราะห์ข้อมูลในตารางที่ 2.2 ค่าผิดพลาดสูงสุดของทุกมิติจะลดลงหลังจากการการหาค่าเหมาะที่สุด แสดงให้เห็นเหตุผลของการ normalize error ที่ทางที่มิวิจัยได้เสนอไว้ การลดข้อผิดพลาดในทุกมิติจะแสดงถึงอิทธิพลที่สำคัญของพารามิเตอร์โครงสร้างที่มีต่อความถูกต้องของหุ่นยนต์ ตารางที่ 2.3 แสดงพารามิเตอร์โครงสร้างของหุ่นยนต์ palletizing หลังจากการออกแบบที่ดีที่สุด ค่าของ ϕ_1 , ϕ_2 และ ϕ_3 มีการเปลี่ยนแปลงเล็กน้อย a_1 อยู่ใกล้กับค่าเริ่มต้น ค่าของ a_2 มีค่าเพิ่มขึ้น และค่าของ a_3 มีค่าเข้าใกล้ขอบเขตบน

ตารางที่ 2.2 ค่าผิดพลาดสูงสุดของทุกมิติจะลดลงหลังจากการหาค่าเหมาะที่สุด

Item	Δx (mm)	Δy (mm)	Δz (mm)	$\Delta \alpha$ (rad)	$\Delta \beta$ (rad)
Original	3.4465	1.5384	0.8160	0.0026	0.0026
Optimization	0.8011	0.8880	0.2222	0.0014	0.0014
Reducing	76.76%	42.28%	72.77%	47.32%	47.32%

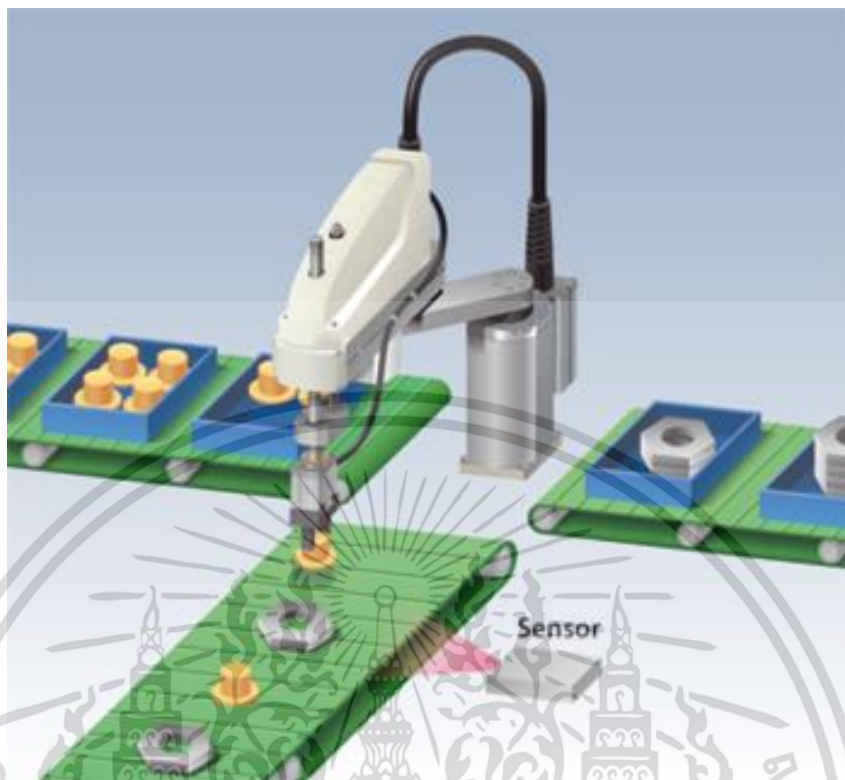
ตารางที่ 2.3 แสดงพารามิเตอร์โครงสร้างของหุ่นยนต์ palletizing หลังจากการออกแบบที่ดีที่สุด

Item	φ_1 (deg)	φ_2 (deg)	φ_3 (deg)	a_1 (m)	a_2 (m)	a_3 (m)
Original	62	112	40	0.135	0.1	0.1
Optimization	66.65	118.86	37.79	0.1395	0.1983	0.2440

สำหรับการพัฒนาในอนาคต จำเป็นต้องพิจารณาปัจจัยอื่นๆ เช่น ความเร็วและความเร่งที่ต้องการของ end effector เพื่อให้โครงสร้างหุ่นยนต์ดีขึ้น

2.2 ระบบปฏิบัติการหุ่นยนต์ (Robot Operating System)

จากศึกษาวิจัยที่เกี่ยวข้อง [1-5] การออกแบบและสร้างพัฒนาแขนกลอุตสาหกรรมเพื่อช่วยในกระบวนการผลิตขึ้น โดยผู้วิจัยได้มีแนวทางในการสร้างหุ่นยนต์ด้วยเทคโนโลยีโอเพ่นซอร์สซึ่งจะเป็นระบบปฏิบัติการที่ควบคุมหุ่นยนต์อุตสาหกรรมให้สามารถเคลื่อนที่ได้ทั้ง 4 แกนสัมพันธ์กันและมีความคลาดเคลื่อนต่ำเหมาะกับการใช้งานที่มีความหลากหลายสำหรับอุตสาหกรรมภายในประเทศ เช่น การหยิบ-วาง ชิ้นงานต่างๆ ของโรงงาน ดังแสดงในรูปที่ 2.6 เป็นต้น โดยโปรแกรมที่ใช้ควบคุมการทำงานของหุ่นยนต์ถูกเขียนขึ้นด้วยโปรแกรมภาษา C, C++ และ Python ซึ่งทำงานบนระบบปฏิบัติการลินุกซ์ เชื่อมต่อระบบปฏิบัติการที่ควบคุมหุ่นยนต์ (ROS) เข้ากับฮาร์ดแวร์ที่เกี่ยวข้องกับหุ่นยนต์ต่างๆ ได้



รูปที่ 2.6 ตัวอย่างการใช้งานหยิบ-วางชิ้นงาน ในภาคอุตสาหกรรม

ปัจจุบันหุ่นยนต์ถูกนำมาประยุกต์ใช้ในกระบวนการผลิตของอุตสาหกรรมภายในประเทศ อาทิเช่น งานประกอบชิ้นส่วนรถยนต์ งานเชื่อม งานผลิตอาหารในโรงงานอุตสาหกรรม และงานในบริเวณที่เสี่ยงต่ออันตรายที่มนุษย์ไม่สามารถปฏิบัติงานได้ เป็นต้น ซึ่งหุ่นยนต์อัตโนมัติในอุตสาหกรรมมีต้นทุนหรือราคาที่ค่อนข้างสูงส่งผลให้มีต้นทุนในการผลิตเพิ่มมากยิ่งขึ้น จากเหตุผลดังกล่าวได้พิจารณาหัวข้อในการศึกษาและพัฒนาเทคโนโลยีของการควบคุมหุ่นยนต์ที่ได้แก่ การพัฒนาระบบควบคุมหุ่นยนต์ด้วยโปรแกรมภาษา C, C++ และการพัฒนาแขนกลอุตสาหกรรมด้วยเทคโนโลยีโอเพนซอร์ส การออกแบบและควบคุมหุ่นยนต์ให้เคลื่อนที่ตามจุดอ้างอิงที่กำหนด (path planning and trajectory) ซึ่งในการพัฒนาต้นแบบแขนกลอุตสาหกรรมด้วยเทคโนโลยีโอเพนซอร์สจะเป็นระบบปฏิบัติการที่ควบคุมแขนกลอุตสาหกรรมให้สามารถเคลื่อนที่ได้ทั้ง 3 แกนสัมพันธ์กันและมีความคลาดเคลื่อนต่ำเหมาะกับการใช้งานในกระบวนการผลิตอุตสาหกรรม

ในส่วนของระบบปฏิบัติการหุ่นยนต์ (Robot Operating System) ซึ่งจากหัวข้อการศึกษาเบื้องต้นที่ได้กล่าวมาแล้วจะเห็นว่าข้อดีของ ROS ในการพัฒนาระบบหุ่นยนต์นั้นมีมากมาย แต่อย่างไรก็ตามในการประยุกต์ใช้กับงานอุตสาหกรรมนั้นอาจจะมีการพัฒนาโครงสร้างให้มีความเสถียรและน่าเชื่อถือขึ้นไปอีก โดยโครงการ ROS-Industrial Open Source ซึ่งเป็นจุดเริ่มต้นของความพยายามร่วมกันของ Yaskawa

Motoman Robotics, Southwest Research Institute และ Willow Garage เพื่อสนับสนุนการใช้ ROS สำหรับระบบการผลิตอัตโนมัติ พื้นที่เก็บข้อมูลซอฟต์แวร์ซึ่งเป็นเจ้าของร่วมใน GitHub ก่อตั้งขึ้นโดย Shaun Edwards (SwRI) ในเดือนมกราคม 2555 นำโดย SwRI ROS-Industrial Consortium America เปิดตัวในเดือนมีนาคม 2556 นอกจากนี้ยังมี ROS- I Consortium Europe ซึ่งนำโดย Fraunhofer IPA ใน Stuttgart ประเทศเยอรมนีและ ROS-Industrial Consortium Asia Pacific นำโดย ARTC และ NTU ในสิงคโปร์ Consortia มีอยู่เพื่อสนับสนุนนักพัฒนาซอฟต์แวร์สำหรับหุ่นยนต์ ROS-industrial โดยการให้การฝึกอบรม การสนับสนุนด้านเทคนิคและการกำหนดแผนงานสำหรับ ROS-I Consortia และยังคงดำเนินโครงการอุตสาหกรรมร่วมกันก่อนการแข่งขันเพื่อพัฒนาขีดความสามารถใหม่สำหรับ ROS-I รูปที่ 2.7 แสดงกลุ่มสมาชิกในอเมริกาเป็นต้น



รูปที่ 2.7 สมาชิกหลักของ ROS industrial consortium

ROS-Industrial เป็นโครงการโอเพ่นซอร์สที่ขยายขีดความสามารถขั้นสูงของ ROS ให้เป็นระบบอัตโนมัติและหุ่นยนต์การผลิต พื้นที่เก็บข้อมูล ROS-Industrial ประกอบด้วยการเชื่อมต่อสำหรับ manipulators อุตสาหกรรมทั่วไป grippers เซ็นเซอร์และเครือข่ายอุปกรณ์ นอกจากนี้ยังมีไลบรารีซอฟต์แวร์สำหรับการปรับเทียบเซนเซอร์ 2D / 3D อัตโนมัติการวางแผนเส้นทาง / การวางแผนการเคลื่อนที่ของแอปพลิเคชันเช่น Scan-N-Plan เครื่องมือสำหรับนักพัฒนาซอฟต์แวร์เช่น Qt Creator ROS Plugin และหลักสูตรการฝึกอบรมที่ตรงกับความต้องการของผู้ผลิต ROS-I ได้รับการสนับสนุนโดยสมาคมนานาชาติด้านอุตสาหกรรมและสมาชิกในการวิจัย

โดยสรุป ROS-Industrial มีความสามารถดังต่อไปนี้

1. เป็น platform ที่ครบวงจรสำหรับซอฟต์แวร์ ROS ที่เกี่ยวข้องกับการผลิต
2. ซอฟต์แวร์ มีการทดสอบความเสถียรและความน่าเชื่อถือที่ตอบสนองความต้องการของอุตสาหกรรม
3. เป็น platform ที่รวมจุดเด่นของ ROS และเทคโนโลยีที่มีอยู่รวมฟังก์ชันการทำงานระดับสูงของ ROS ด้วยความเชื่อถือได้และความปลอดภัยในระดับต่ำของตัวควบคุมหุ่นยนต์อุตสาหกรรม
4. ช่วยกระตุ้นการพัฒนาซอฟต์แวร์โดยใช้อินเทอร์เฟซมาตรฐาน
5. เป็นระบบที่ "ง่าย" ในการใช้งานวิจัยที่ทันสมัยเพื่อการใช้งานในภาคอุตสาหกรรมโดยใช้สถาปัตยกรรม ROS ทั่วไป จัดเตรียมอินเทอร์เฟซการเขียนโปรแกรมแอปพลิเคชันที่ใช้งานง่ายและง่ายต่อการใช้งาน ในรูปที่ 2.8 แสดงการประยุกต์ใช้หลักๆ ของ I-ROS

What is ROS-Industrial used for?

Prototyping, exploratory R&D **Actual products, deployments**

- **Example: Scan'n'plan**
 - Leverages 3D scanning to generate part geometry and trajectory on the fly, then perform blending/polishing
- **Example: ROS AGVs**
 - Leverage standard and custom ROS components to add intelligent navigation

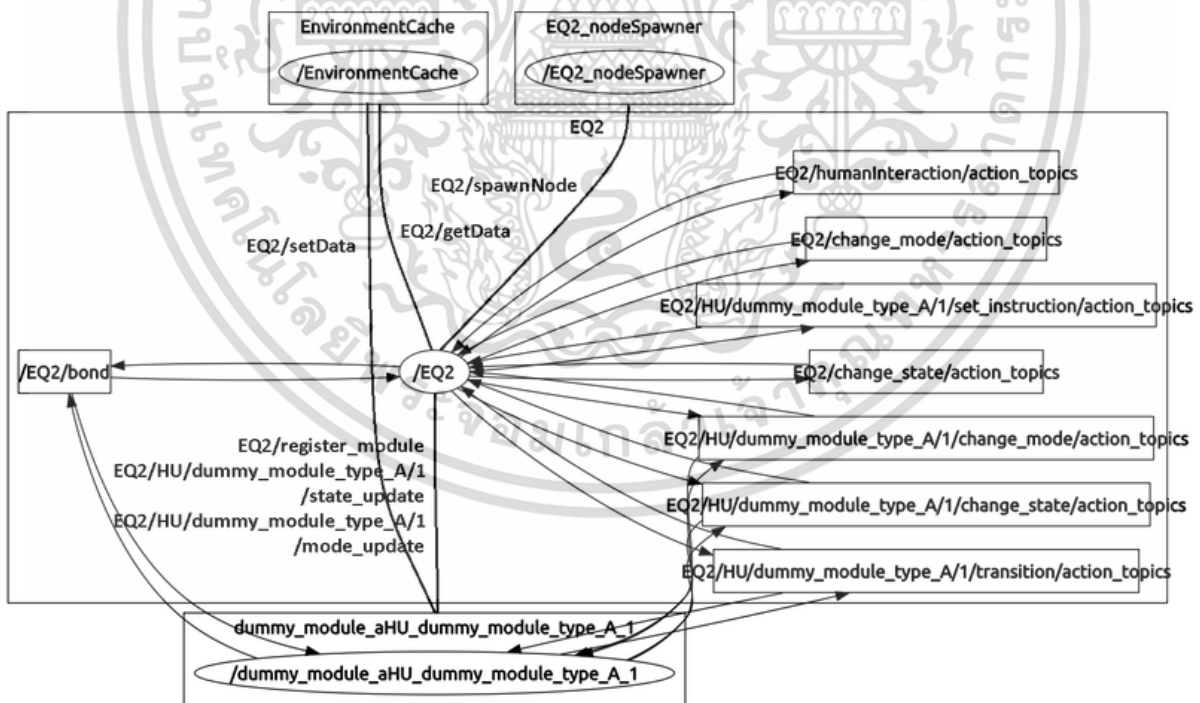
Credits: Southwest Research Institute, the Boeing Company Credits: Fraunhofer IPA, Bär Automation GmbH

รูปที่ 2.8 การประยุกต์ใช้หลักๆ ของ I-ROS

ROS (Robot Operating System) ได้ถูกพัฒนาตั้งแต่ปี 2007 โดยมีการพัฒนาอย่างต่อเนื่องมาถึงปัจจุบัน โดยแบ่งออกเป็นเวอร์ชันดังนี้ Box Turtle,C Turtle,Diamondback,Electric Emys,Fuerte Turtle,Groovy Galapagos,Hydro Medusa,Indigo Igloo,Jade Turtle,Kinetic Kame ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 เวอร์ชันของ ROS



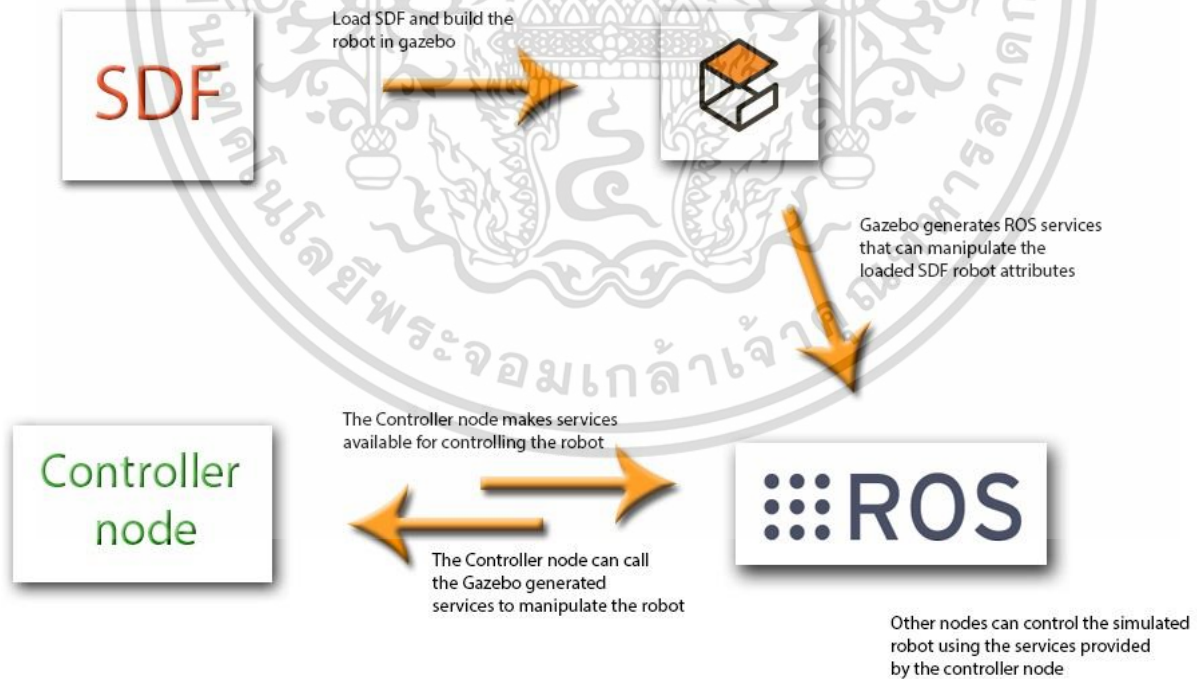
รูปที่ 2.10 รูปแบบ ROS communication

โดยรูปที่ 2.11 - 2.13 มีเครื่องมือและไลบรารีต่างๆ ที่ใช้ในการรวบรวมเข้ากับ ROS หลักๆ มี 5 ไลบรารี สำหรับการพัฒนาหุ่นยนต์ดัง ดังนี้

- Gazebo คือ สร้างซิมูเลชันแบบ 3 มิติ ได้ทั้งแบบ indoor และ outdoor อีกทั้งฟังก์ชันการ คำนวณแบบพลศาสตร์ (Dynamics) และ คิเนมาติกส์ (Kinematics) ฉบับสมบูรณ์
- OpenCV คือ ไลบรารีด้านการประมวลผลภาพ หรือ image processing ซึ่งเป็นไลบรารีที่ใช้การอย่างแพร่หลายทั่วโลก โดยจะมีอัลกอริทึมในการตรวจหาวัตถุ หรือ ติดตามวัตถุเป้าหมาย อีกทั้งยังสามารถนำไปประยุกต์ใช้ในการหาระยะห่าง ความถี่และลึกของวัตถุด้วยเทคนิค Stereo vision แล้วนำไปรวมเข้ากับหุ่นยนต์เพื่อให้สามารถหยิบและหาวัตถุอย่างมีประสิทธิภาพมากยิ่งขึ้น
- PCL (the Point Cloud Library) คือ ไลบรารีวิเคราะห์และจัดการภาพ 3 มิติ ซึ่ง PCL ได้นำอัลกอริทึม feature detection, registration, kd-trees, octrees, sample consensus และที่สำคัญถ้ามีการทำงานร่วมกับเซนเซอร์ 3 มิติ อย่าง Microsoft Kinect หรือ Scanning laser โดย PCL สามารถเชื่อมต่อ ROS ไว้เรียบร้อยแล้วเพื่อวิเคราะห์ข้อมูล 3 มิติ
- MoveIt คือ ไลบรารีวางแผนในการเคลื่อนที่ (motion planning library) ที่มีประสิทธิภาพและปรับปรุงจากการทดลองเป็นอย่างดี โดยอัลกอริทึมนี้ใช้กับหุ่นยนต์อย่างแพร่หลาย อีกทั้ง ROS รองรับการใช้ไลบรารีอย่างมีประสิทธิภาพ
- Rviz คือ ไลบรารีสำหรับการเชื่อมต่อกับหุ่นยนต์จริงพร้อมเห็นภาพการทำงานของหุ่นยนต์ (visualization) ซึ่งได้จากการเขียนโครงสร้างของหุ่นยนต์จริง โดยสามารถ Sketch ทั้งสองมิติและสามมิติ (2D Sketch, 3D Sketch) จากไฟล์ SDF สร้างเป็นภาพการทำงานของหุ่นยนต์ได้
- ROS-Industrial คือ เป็นโอเพ่นซอร์ส (open-source) ที่มีความสามารถขั้นสูงของ ROS ที่นำไปใช้ในกระบวนการผลิตอัตโนมัติ และหุ่นยนต์ในอุตสาหกรรม โดยไลบรารีนี้มีอินเตอร์เฟซ (Interfaces) ที่ใช้ร่วมกับกริปเปอร์ (Grippers), เซนเซอร์ (Sensor) และอุปกรณ์เน็ตเวิร์ค (Device networks) ซึ่งซอฟต์แวร์ได้พัฒนามาสำหรับระบบเซนเซอร์ 2 มิติ และ 3 มิติ สามารถทำการคำนวณหาทิศทาง การเคลื่อนที่ การประยุกต์ใช้ในงานแอปพลิเคชันนำเข้ากับ ROS ไปใช้งานในการคำนวณทิศทาง การเคลื่อนที่และซิมูเลชัน

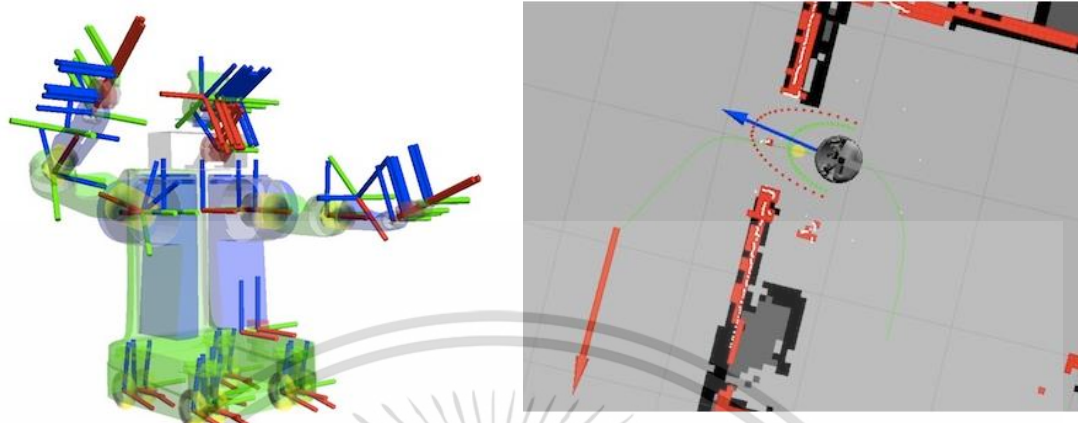


รูปที่ 2.11 โลโก้ของ 5 ไบรารีสำหรับการพัฒนาหุ่นยนต์



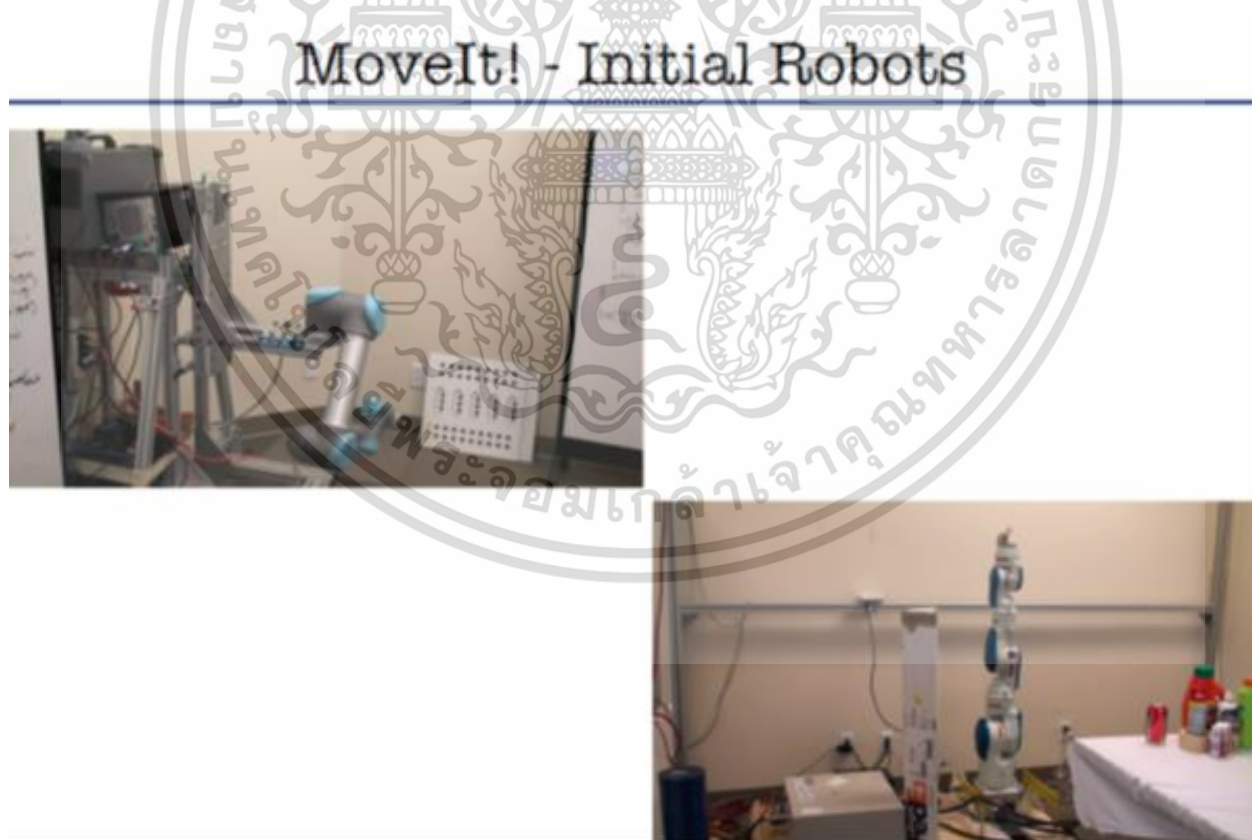
รูปที่ 2.12 ซอฟต์แวร์จำลอง GAZEBO เป็นการจำลองการทำงานของหุ่นยนต์ (Robot simulation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 Robot Geometry Library รูปที่ 2.14 Pose Estimation, Localization, and Navigation

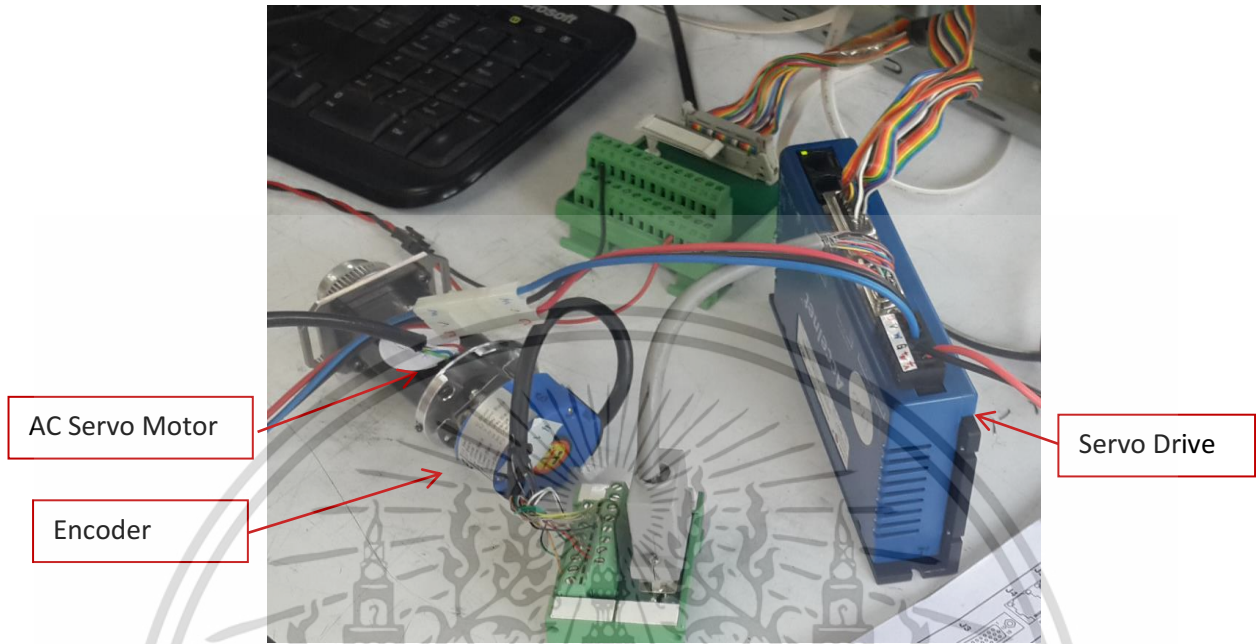
ดังนั้น ROS จึงเป็นแหล่งรวบรวมไลบรารีและเครื่องมือไม่ว่าจะเป็น hardware abstraction, device drivers, libraries, visualizers, message-passing, package management และอื่นๆ อีกมากมาย เพื่อช่วยให้นักพัฒนาซอฟต์แวร์ในการสร้างแอปพลิเคชันทางด้านหุ่นยนต์ดังในรูปที่ 2.15



รูปที่ 2.15 Module MoveIt สำหรับการเชื่อมโยงกับหุ่นยนต์อุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 อุปกรณ์ที่ใช้ในงานวิจัย



รูปที่ 2.16 ภาพรวมอุปกรณ์ที่ใช้สร้างหุ่นยนต์อุตสาหกรรม

หุ่นยนต์ที่จะสร้างขึ้นนี้เป็นแบบ Scara robot มีจำนวนแกน 4 แกน ในแต่ละแกนของหุ่นยนต์ประกอบด้วย

- Servo Drive
- AC Servo Motor
- Encoder

2.3.1 ชุด Servo Drive

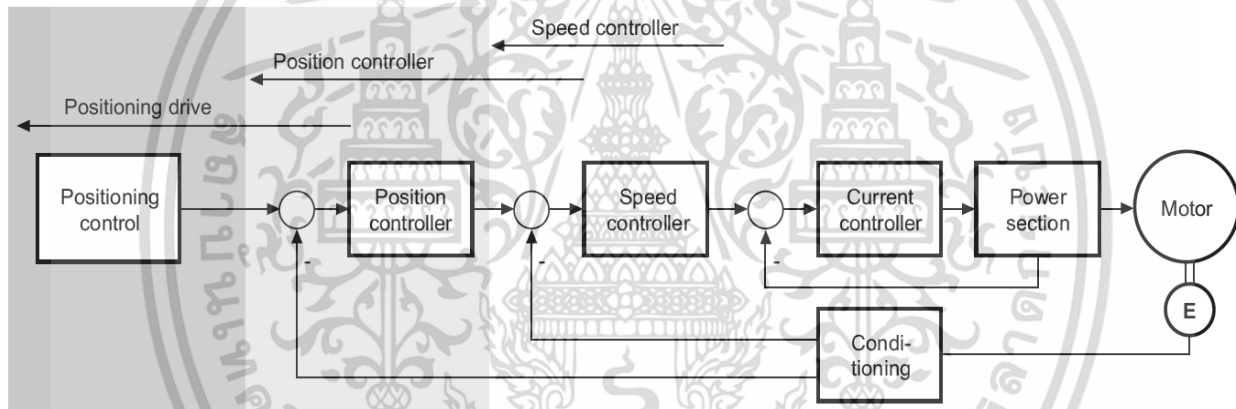


รูปที่ 2.17 ชุด Servo Drive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

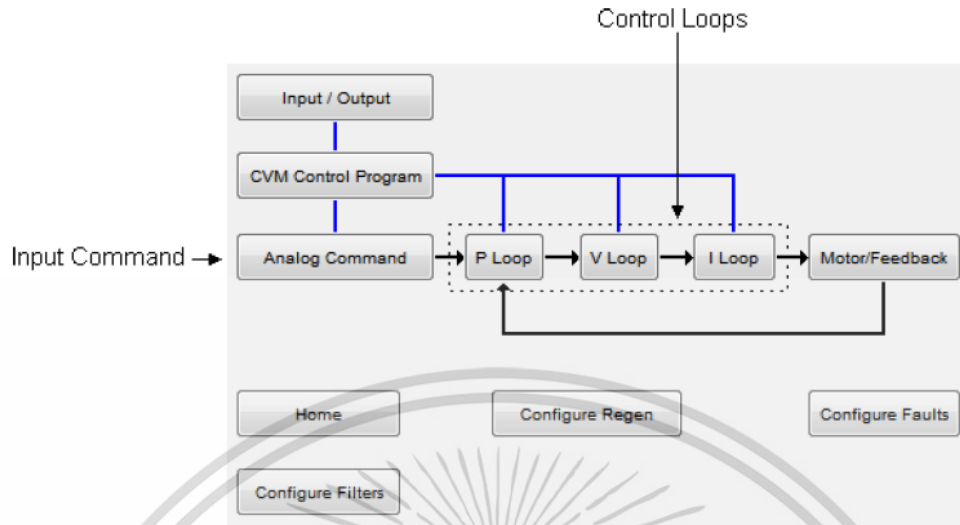
ชุด Servo drive ทำหน้าที่เป็นชุดควบคุมการทำงานของ AC Servo Motor ขนาด 50W และ 100W โดยสามารถควบคุม ความเร็ว, อัตราเร่ง และตำแหน่งการเคลื่อนที่ได้โดยใช้ Servo motor ผ่านชุดควบคุม ซึ่งจะส่งสัญญาณควบคุม (Signal command) ผ่านชุดควบคุมมอเตอร์มีรูปแบบของสัญญาณควบคุม (signal command) การทำงานได้ทั้งเป็นแบบคลื่นรูปไซน์ (Sine wave) และแบบคลื่นรูปสี่เหลี่ยม (Square wave) นอกจากนี้ยังสามารถควบคุมแบบระบบปิดป้อนกลับ (close loop) ของชุดมอเตอร์และรับคำสั่งได้ไม่น้อยกว่านี้ Stepper commands, Quadrature input, position/velocity/torque $\pm 10V$ และ PWM velocity/torque รวมถึงการตรวจจับค่าความผิดพลาด โดยการควบคุมการเคลื่อนที่ของ AC Servo motor ประกอบด้วย 3 โหมด คือ

- โหมดการควบคุมกระแสไฟที่จ่ายให้กับมอเตอร์ (Current control loop)
- โหมดการควบคุมอัตราเร่ง (Velocity Control Mode)
- โหมดควบคุมตำแหน่งการเคลื่อนที่ (Position Control mode)



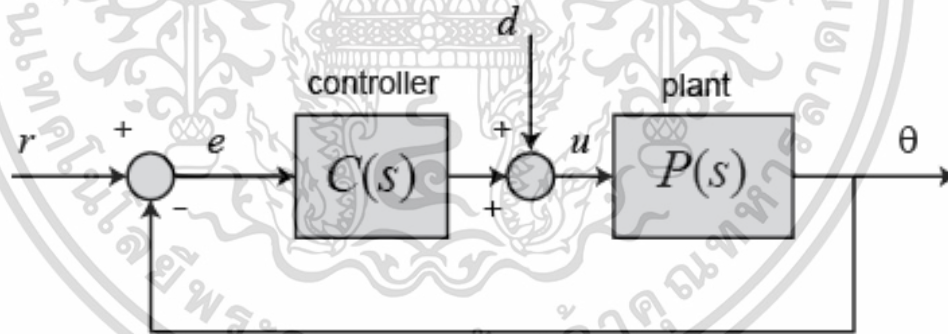
รูปที่ 2.18 โหมดการควบคุมการเคลื่อนที่ของ AC Servo motor

ทางด้านการใช้งานของชุด Servo drive จะมีซอฟต์แวร์ที่ใช้การควบคุมที่พร้อมสายเชื่อมต่อกับคอมพิวเตอร์ด้วย สาย RS-232 และสาย CANBus โดยสามารถเชื่อมต่อสัญญาณระหว่างชุดควบคุมผ่านสาย CANBus เมื่อปรับเกรนควบคุมทั้ง 3 โหมดแล้ว สามารถเก็บข้อมูลการตั้งค่าไว้ในหน่วยความจำสำหรับการควบคุมแบบปิด (Close Loop) ของมอเตอร์



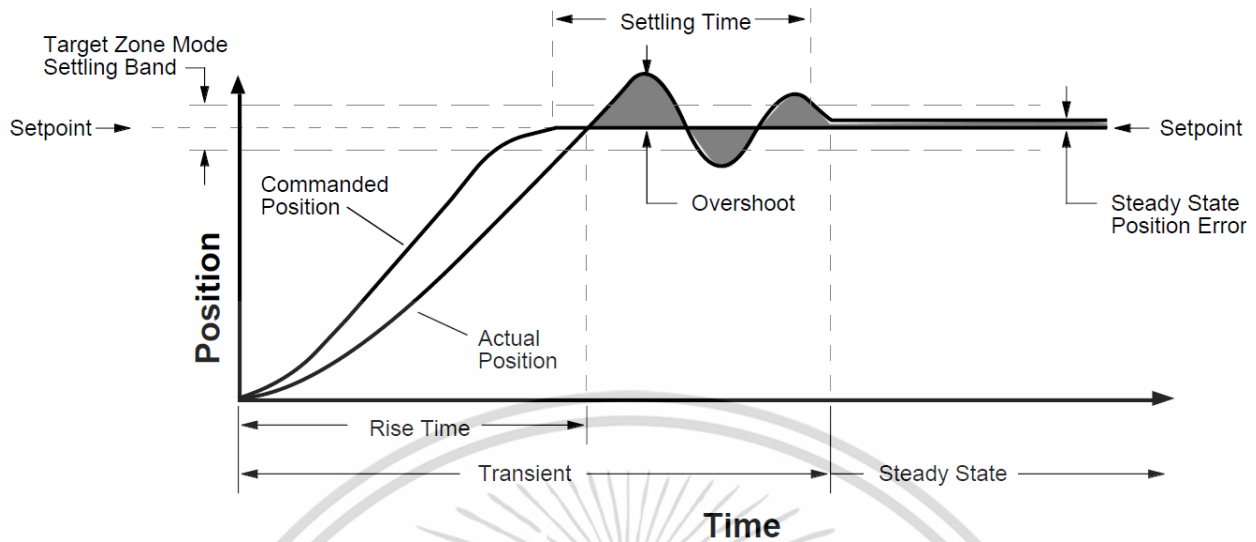
รูปที่ 2.19 โหมดซอฟต์แวร์ที่ใช้การควบคุมของชุด Servo Drive

โดยมีใช้การควบคุมแบบ PID control ที่อาศัยเวลาเป็นตัวแปรที่สำคัญการหาผลตอบสนองของระบบโดเมนเวลาซึ่งวิเคราะห์ในโดเมนเวลาเพื่อที่จะได้ระบบซึ่งเหมาะสมกับการนำไปใช้งานสิ่งที่จะถูกควบคุมเป็นกระบวนการที่ต้องการควบคุมหรือระบบ (Plant) ดังแสดงในรูปที่ 2.20



รูปที่ 2.20 ระบบควบคุมโดยทั่วไป

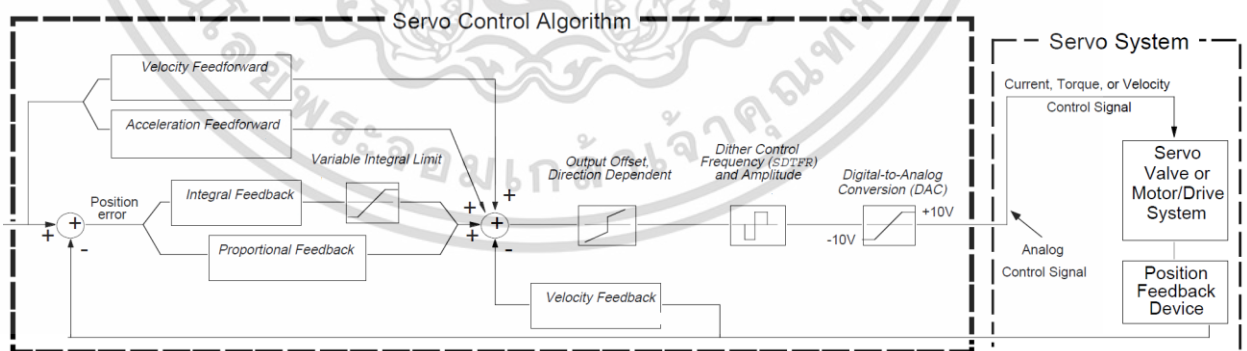
หลักการของระบบควบคุมแบบป้อนกลับ สัญญาณความผิดพลาดซึ่งได้จากการเปรียบเทียบระหว่างสัญญาณป้อนเข้าและสัญญาณออก (ตัวแปรควบคุม) จะถูกส่งไปยังตัวควบคุม (Controller) โดยระบบควบคุมแบบป้อนกลับจะเกิดการปฏิบัติการควบคุมขึ้น (Control action) คือลักษณะการควบคุมที่ใช้สัญญาณปฏิบัติการเพื่อให้เกิดการแก้ไขข้อผิดพลาดของระบบ ด้วยการปฏิบัติการควบคุมพื้นฐานในแบบต่างๆ ดังแสดงในรูปที่ 2.21



รูปที่ 2.21 ลักษณะการควบคุมที่ใช้สัญญาณปฏิบัติการของระบบ

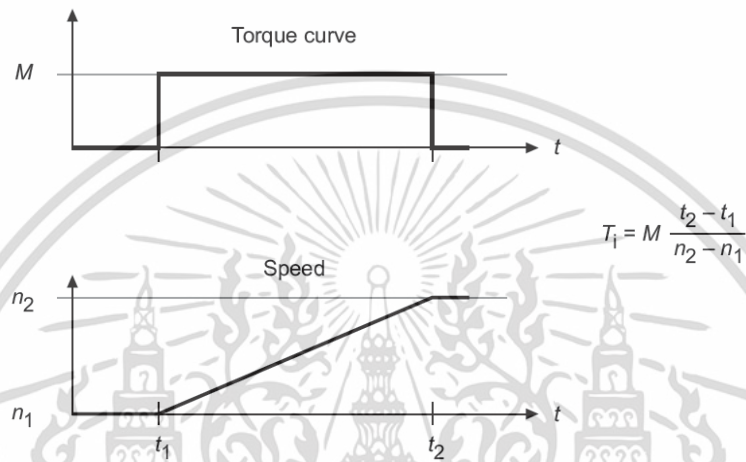
ทั้งนี้คุณภาพของระบบควบคุมที่ดีจะต้องมีความแม่นยำ หรือควรจะมี error น้อยๆ และค่าเกน K_p (Proportional gain) ที่เหมาะสมในการตอบสนองของระบบจะต้องรวดเร็ว ดังแสดงในรูปที่ 2.22 โดยมีสมการของระบบควบคุมแบบ PID control

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.7)$$

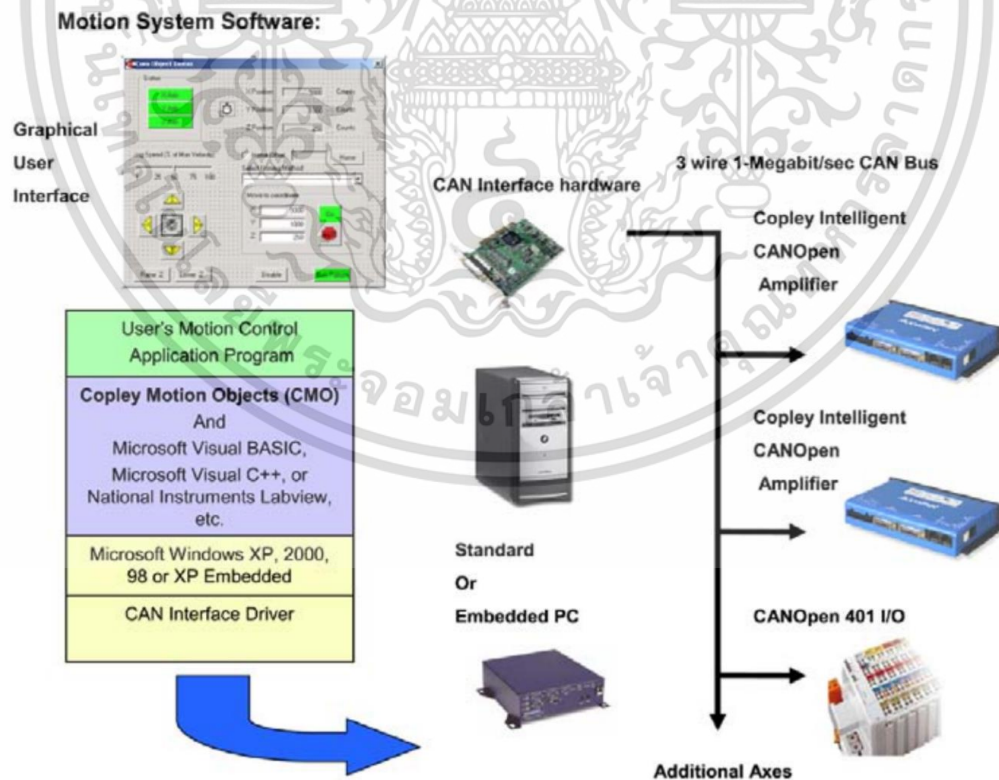


รูปที่ 2.22 รูปแบบการควบคุมแบบ PID control ภายในชุด Servo Drive

ในระบบควบคุมกำลังบิด (Torque control system) กำลังบิดของมอเตอร์เซอร์โวจะถูกรักษาให้มีค่าคงที่ เนื่องจากแรงบิดของมอเตอร์แปรตามกระแสของมอเตอร์ ดังนั้นกระแสที่ป้อนให้กับมอเตอร์ เพื่อรักษาค่าแรงบิดให้คงที่เอาไว้ วิธีนี้สามารถทำได้ด้วยวงจรที่ทำการเปรียบเทียบกระแสเอาต์พุทของมอเตอร์กับกระแสอินพุทของมอเตอร์ และ ขยายผลต่างเพื่อใช้เป็นวงจรควบคุมแรงบิดป้อนกลับ (Torque control feedback circuit) ดังกราฟในรูปที่ 2.23 ทั้งนี้สามารถเชื่อมต่อชุดควบคุมเข้าด้วยกัน ดังแสดงในรูปที่ 2.24

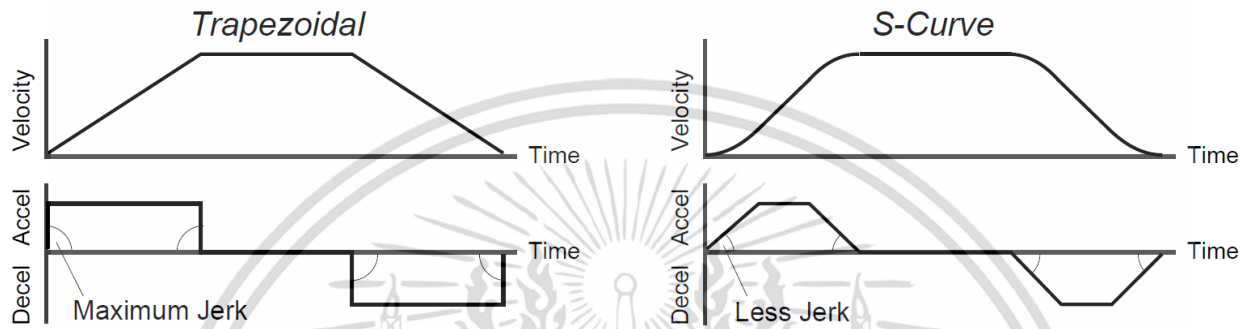


รูปที่ 2.23 สัญญาณการควบคุมแบบ PID control สำหรับควบคุมกำลังบิด (Torque control system)

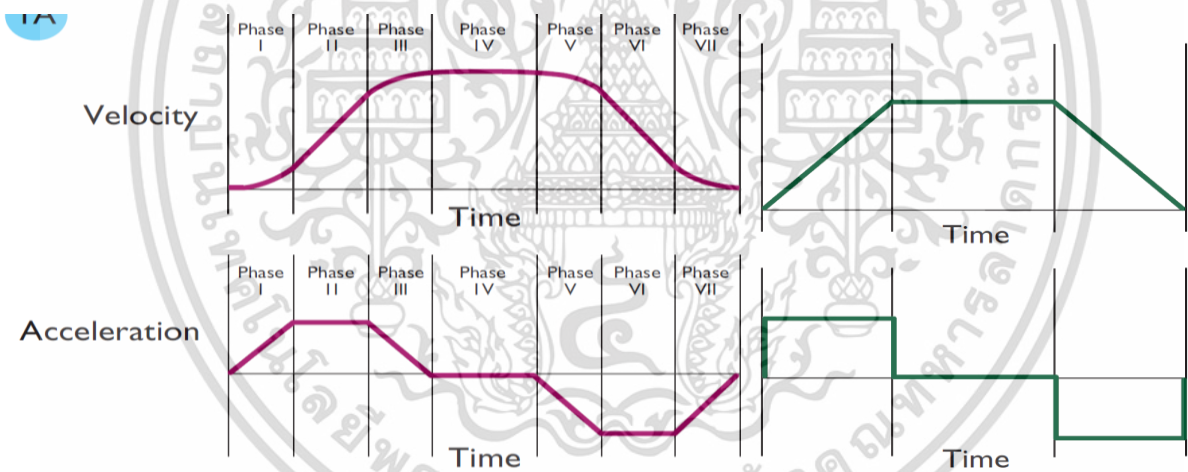


รูปที่ 2.24 การเชื่อมต่อสัญญาณระหว่างชุดควบคุมผ่านสาย CANBus

ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ด้วยคำสั่งการทำงานนั้นจะมีอยู่ด้วยกันสองแบบคือ S-curve และ trapezoidal profiles ดังแสดงในรูปที่ 2.25 โดยถ้าใช้คำสั่งควบคุมการเคลื่อนที่แบบ S-curve จะมีลักษณะการตอบสนองที่มีความราบเรียบ (smooth) มากกว่าทำให้กลไกทางกลจะสึกกร่อนน้อยเมื่อเทียบกับคำสั่งควบคุมการเคลื่อนที่แบบ trapezoidal profiles ที่จะมีความเร็วสูงสุดในการเคลื่อนที่และหยุดทันทีเมื่อถึงตำแหน่งต้องการ ทำให้การควบคุมแบบนี้กลไกทางกลจะมีโอกาสสึกมากกว่าแบบ S-curve ดังแสดงในรูปที่ 2.26

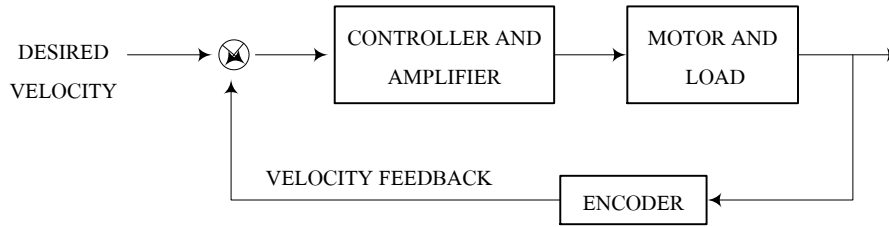


รูปที่ 2.25 คำสั่งการทำงานแบบ S-curve และ trapezoidal profiles

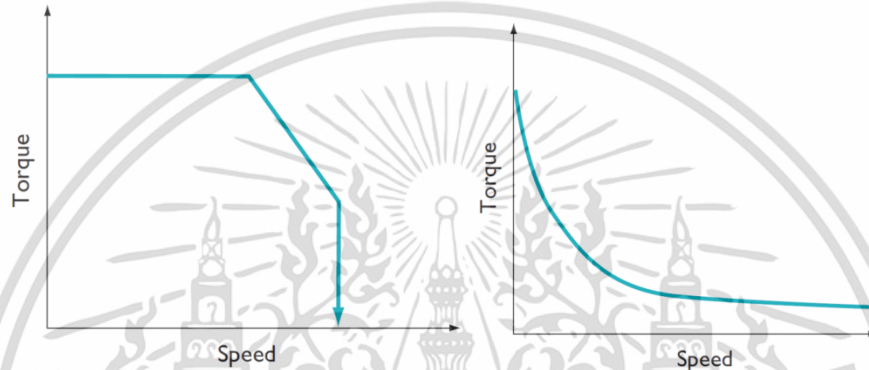


รูปที่ 2.26 การควบคุมการเคลื่อนที่ของมอเตอร์ที่สัมพันธ์กันกับเวลา

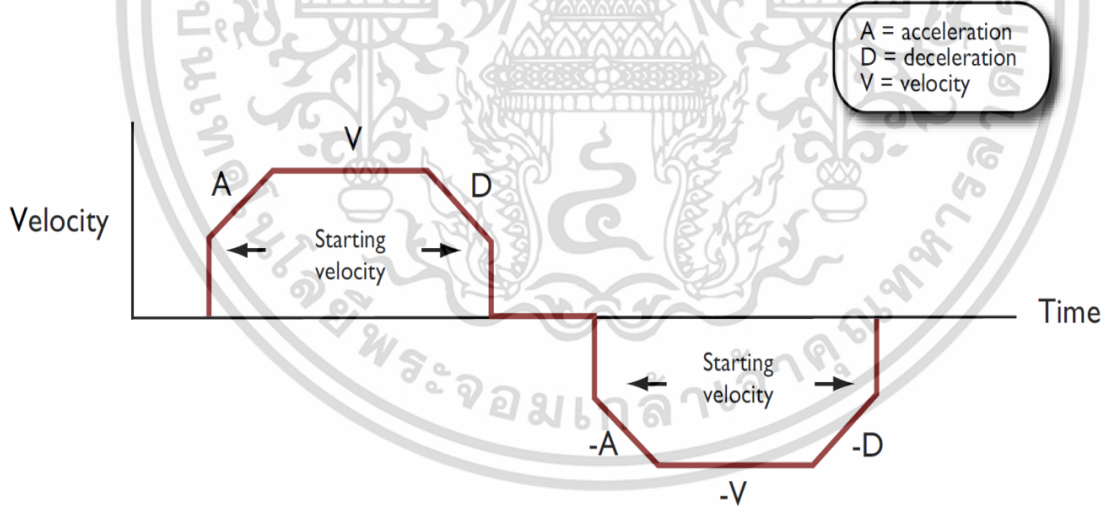
ตัวอย่างระบบการควบคุมการเคลื่อนที่ด้วยกราฟแรงบิด (torque) และอัตราความเร็ว (speed) สำหรับ เซอร์โวมอเตอร์ในรูปแบบ Non-zero starting velocity ดังแสดงในรูปที่ 2.27, 2.28 และมีความสัมพันธ์ของ ความเร็ว ความเร่ง การลดความเร่ง ดังแสดงในรูปที่ 2.29



รูปที่ 2.27 บล็อกไดอะแกรมของระบบการควบคุมการเคลื่อนที่

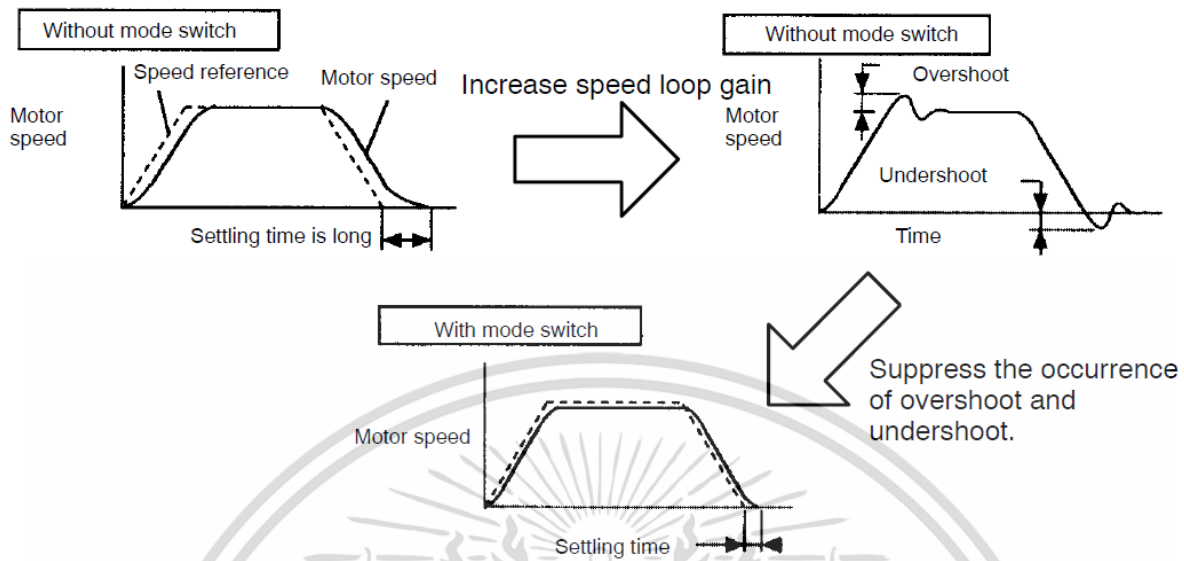


รูปที่ 2.28 สัญญาณการเคลื่อนที่ด้วยกราฟแรงบิด (torque) และอัตราความเร็ว (speed)



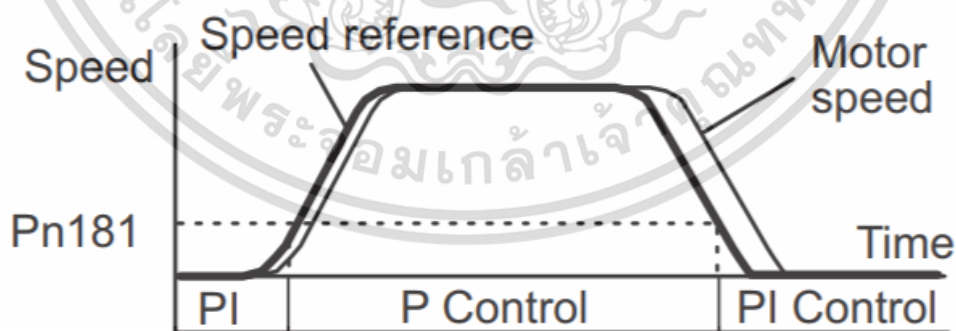
รูปที่ 2.29 ความสัมพันธ์ของความเร็ว ความเร่ง การลดความเร่ง

ตัวอย่างในการปรับค่าเกนสำหรับควบคุมอัตราความเร็วของมอเตอร์ (Motor speed) ภายใต้ผลตอบสนองของระบบควบคุมแบบ PID control เพื่อลดค่า overshoot ดังแสดงในรูปที่ 2.30



รูปที่ 2.30 ตัวอย่างในการปรับค่าเกนสำหรับควบคุมการทำงานของมอเตอร์

ทั้งนี้ในการควบคุมอัตราความเร็วด้วยคำสั่งอัตราความเร็วอ้างอิง (Speed reference) เทียบกับอัตราความเร็วของมอเตอร์ ซึ่งพบว่าในช่วงเวลาชั่วครู่จะต้องใช้เกน PI (Propotional Integral gain) ในการขับมอเตอร์หมุนที่แรงบิดเริ่มต้นเพื่อทำให้มอเตอร์เกิดการสั่นและผลกระทบทางกลไม่มากนัก หลังจากนั้นระบบควบคุมจะใช้เกน P (Propotional gain) เพื่อให้มอเตอร์หมุนจนกระทั่งได้อัตราความเร็วอ้างอิง เมื่อได้ตำแหน่งที่ต้องการ จะใช้เกนเกน PI (Propotional Integral gain) เพื่อให้มอเตอร์หยุดหมุนแบบราบเรียบ ดังแสดงในรูปที่ 2.31



รูปที่ 2.31 ระบบการควบคุมอัตราความเร็วด้วยคำสั่งอัตราความเร็วอ้างอิง (Speed reference) เทียบกับอัตราความเร็วของมอเตอร์

โดยหากพิจารณาระบบควบคุมการทำงานของมอเตอร์ดังกล่าวจะมีลักษณะที่สอดคล้องกับการทำงานของหุ่นยนต์ที่มีการเคลื่อนที่แบบ JOG และ JERK

- JERK เป็นการเคลื่อนที่แบบ Point-to-Point โดยใช้รูปแบบคำสั่งการเคลื่อนที่แบบ S-curve profiling ซึ่งการเคลื่อนที่แบบนี้จะถูกใช้ในการเคลื่อนที่แบบเส้นตรงที่ใช้หลักการเทียบค่า (Interpolated) จากความสัมพันธ์ของความเร็วต่อเวลา หรือระยะทางต่อเวลา ตามสมการข้างล่างนี้

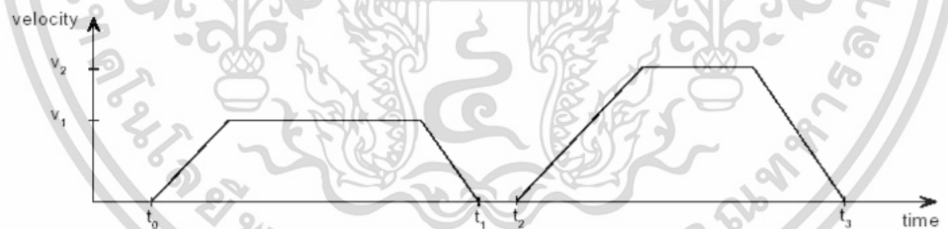
$$\text{Time} = \frac{\text{Velocity}}{A_{\text{avg}}} \quad \text{or} \quad \text{Time} = \sqrt{\frac{2 * \text{Distance}}{A_{\text{avg}}}} \quad (2.8)$$

- JOG เป็นการเคลื่อนที่แบบระบุตำแหน่ง ณ ปัจจุบันในการอ้างอิง ซึ่งเรียกว่า Relative Moves สามารถคำนวณหาค่าได้จาก

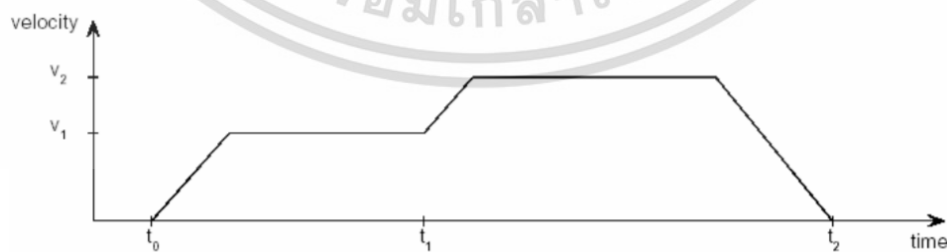
$$\text{Target position} = \text{current \& actual position} + \text{command position} \quad (2.9)$$

และมีการเคลื่อนที่แบบ JOG ที่อ้างอิงตำแหน่ง home เทียบกับตำแหน่งปัจจุบันซึ่งเรียกว่า Absolute Moves สามารถคำนวณหาค่าได้จาก

$$\text{Target position} = \text{command position}$$

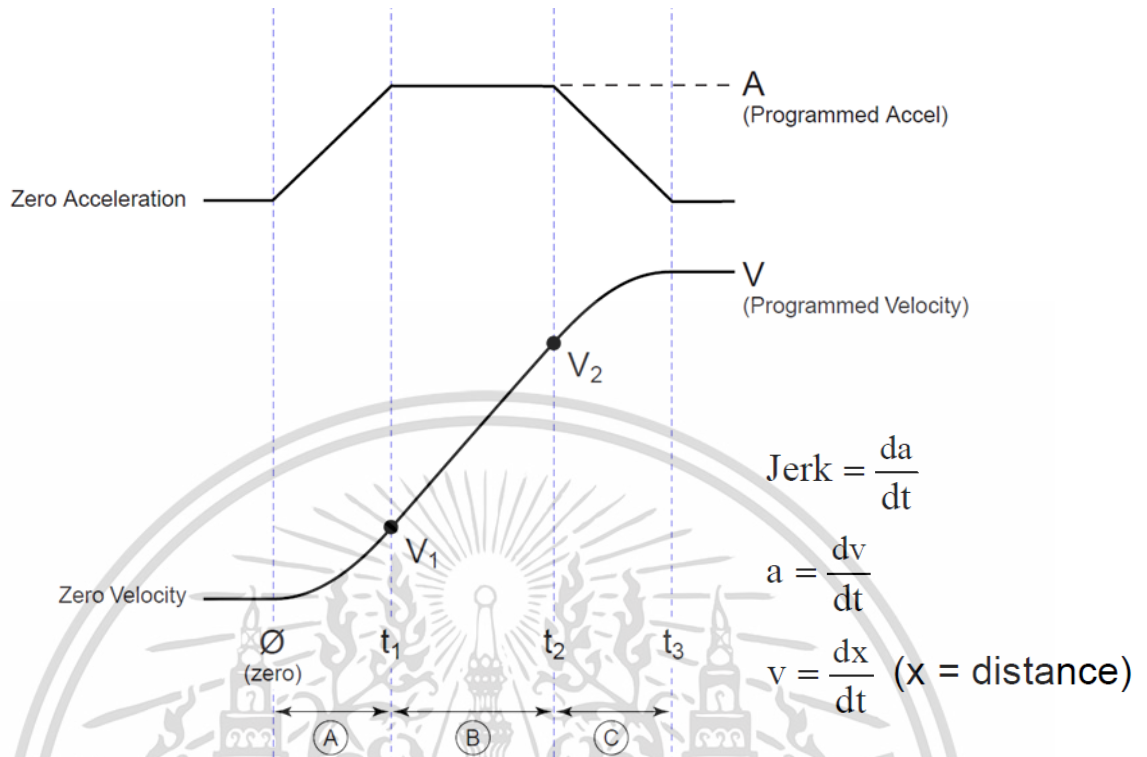


ก) Relative Moves



ข) Absolute Moves

รูปที่ 2.32 รูปแบบการเคลื่อนที่ของหุ่นยนต์



รูปที่ 2.33 รูปแบบการเคลื่อนที่ของหุ่นยนต์

จากรูปที่ 2.33 มีขั้นตอนในการคำนวณหาค่า Jerk ดังนี้

- Step 1: คำนวณหาค่าความเร่งและความเร็วเทียบกับเวลาที่ $t_1 \geq t \geq 0$

$$t_1 \geq t \geq 0 \quad a(t) = J_A * t$$

$$v(t) = \frac{J_A * t^2}{2}$$

$$d(t) = \frac{J_A * t^3}{6}$$

- Step 2: คำนวณหาค่าความเร่งและความเร็วเทียบกับเวลา $t_2 \geq t > t_1$

$$t_2 \geq t > t_1 \quad a(t) = A$$

$$v(t) = \frac{A^2}{2J_A} + A * (t - t_1)$$

$$d(t) = \frac{J_A * t_1^3}{6} + \left(\frac{A * (t - t_1)^2}{2} \right) + \left(V_1 * (t - t_1) \right)$$

- Step 3: คำนวณหาค่าความเร่งและความเร็วเทียบกับเวลา $t_3 \geq t > t_2$

$$t_3 \geq t > t_2 \quad a(t) = A - (J_A * (t - t_2))$$

$$v(t) = V - \left(\frac{J_A * (t_3 - t)^2}{2} \right)$$

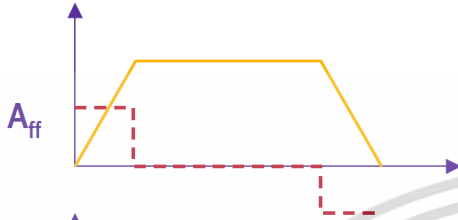
$$d(t) = \frac{V^2}{2AA} + \left(\frac{J_A (t_3 - t)^3}{6} \right) - \left(V * (t_3 - t) \right)$$

โหมตการทำงานเพิ่มเติมสำหรับชุดควบคุมการเคลื่อนที่หุ่นยนต์ มีดังนี้

- Acceleration feed-forward (Aff) ค่าเกรนความเร่งถูกใช้สำหรับการชดเชยแรงเฉื่อยของมอเตอร์ (inertia)
- Friction feed-forward (Fff) ค่าเกรนแรงเสียดทานของแกนหุ่นยนต์ถูกใช้สำหรับการชดเชยแรงเสียดทานจลน์ของหุ่นยนต์ (Kinetic friction) เป็นแรงเสียดทานที่เกิดขึ้นระหว่างผิวสัมผัสของแกนหุ่นยนต์ ในสถานะที่แกนหุ่นยนต์ได้รับแรงกระทำแล้วทำให้เกิดการเคลื่อนที่ของหุ่นยนต์ด้วยความเร็วคงที่
- Velocity feed-forward (Vff) ค่าเกรนความเร็วถูกใช้สำหรับชดเชยความฝืดหนืด (Viscous friction) ที่เกิดขึ้นจากความฝืดหรือความต้านทานต่อการเคลื่อนที่ระหว่างแกนหุ่นยนต์ในชั้นของสารหล่อลื่นจะเกาะติดกับผิวหน้าเกียร์และผิวหน้าแกนหุ่นยนต์ที่ล้อมรอบในขณะที่แกนหุ่นยนต์หมุน (friction that is proportional to velocity)



Commanded trapezoidal velocity profile



Acceleration feed-forward has additive effect on servo command during acceleration, no effect during slew, and subtractive effect during deceleration



Friction feed-forward has additive effect on servo command at all times during move.

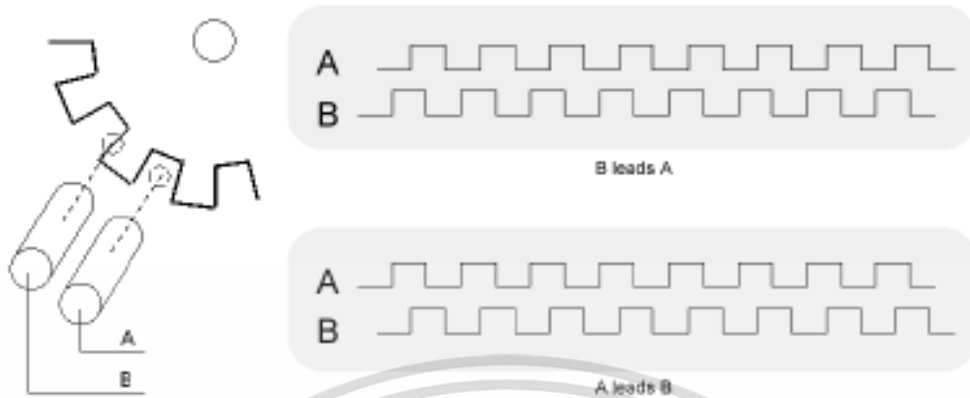


Velocity feed-forward has additive effect on servo command directly proportional to velocity profile.

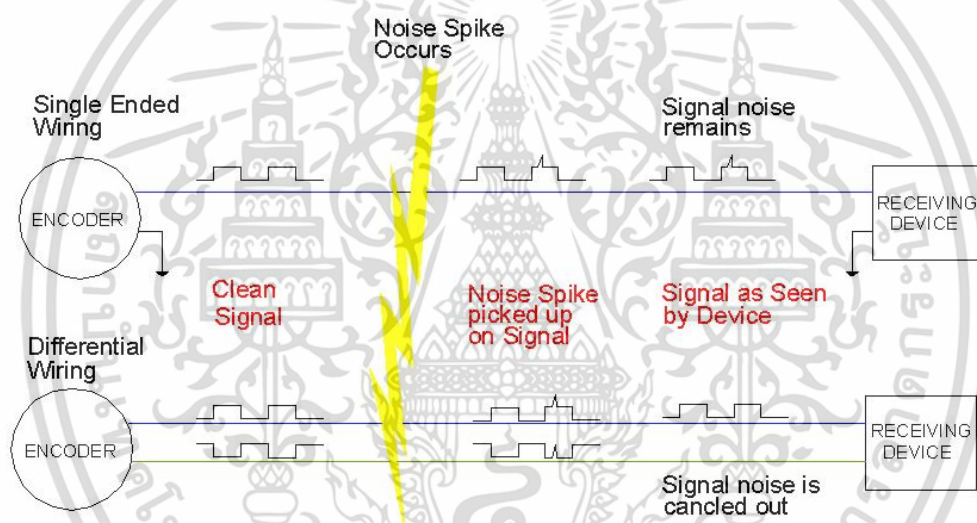
รูปที่ 2.34 ลักษณะคำสั่งควบคุมของของแอมพลิจูด A_{ff} , F_{ff} , V_{ff} เมื่อเทียบกับคำสั่งควบคุมความเร็ว (velocity)

2.3.2 ชุดเอนโค้ดเดอร์ (Encoder)

เอนโค้ดเดอร์ (Encoder) เป็นแบบเอนโค้ดเดอร์แบบเพิ่มค่า (Incremental Encoder) และมีความละเอียดจำนวนรอบที่ 2500 P/R จำนวน 8 โพล (pole) สามารถทำงานได้ทั้งแบบ sine และ cosine โดยที่สัญญาณ AB channels บอทิศทาง (Bidirectional) และความเร็ว (speed and rotation sense detection) ซึ่งถ้า B สัญญาณนำหน้า A จะหมุนตามเข็มนาฬิกา แต่ถ้า A สัญญาณนำหน้า B จะหมุนทวนเข็มนาฬิกา ทั้งนี้ ABZ บอทิศทางและตำแหน่งเริ่มต้น (Bidirectional / Zero) ส่วน Signal Z จะมี 1 Pulse/รอบ เพื่อแจ้งให้รู้ว่าครบรอบแล้ว ดังแสดงในรูปที่ 2.35 นอกจากนี้ยังมีสัญญาณกลับ (invert) /A, /B, /Z ถูกใช้สำหรับการลดสัญญาณรบกวน (Complementary signals) เนื่องจากสัญญาณรบกวนจะส่งผลกับสัญญาณ ABZ ด้วยทำให้สัญญาณรบกวนจะหายไป โดยที่ /A, /B, /Z จะเป็นสัญญาณที่ต่างจาก ABZ อยู่ 180 องศา ดังแสดงในรูปที่ 2.36



รูปที่ 2.35 สัญญาณเอนโค้ดเดอร์ (Encoder)



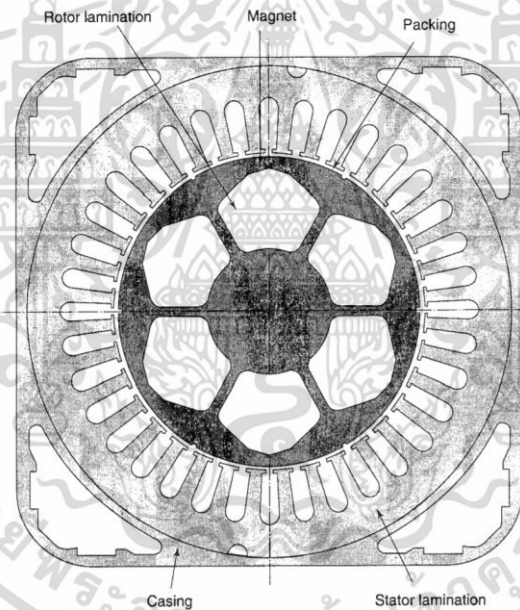
รูปที่ 2.36 ส่วนการลดสัญญาณรบกวน (Complementary signals)

2.3.3 เอซีเซอร์โวมอเตอร์ (AC servo motor)

เอซีเซอร์โวมอเตอร์ (AC servo motor) เป็นแบบไม่มีแปรงถ่านมีส่วนประกอบด้วยแกนแม่เหล็กถาวร ชองสเตเตอร์ แกนโรเตอร์ ดังแสดงในรูปที่ 2.37 โดยโรเตอร์ทำมาจากแม่เหล็กถาวรที่มีประสิทธิภาพสูง ซึ่งทำมาจากวัสดุหลายอย่าง อาทิเช่น เซรามิก (เฟอไรต์) หรือ ซามาเรียม โคบอลต์ เป็นต้น ดังแสดงในรูปที่ 2.38

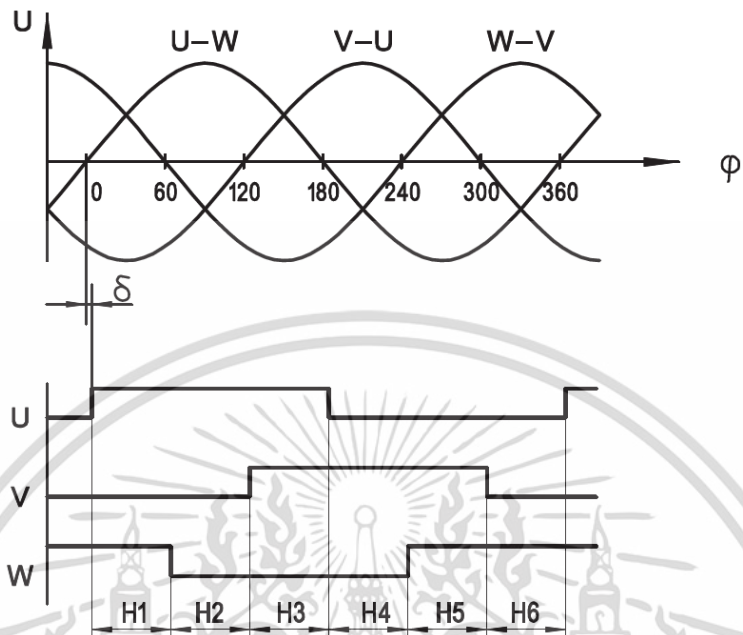


รูปที่ 2.37 ส่วนประกอบของเอซีเซอร์โวมอเตอร์



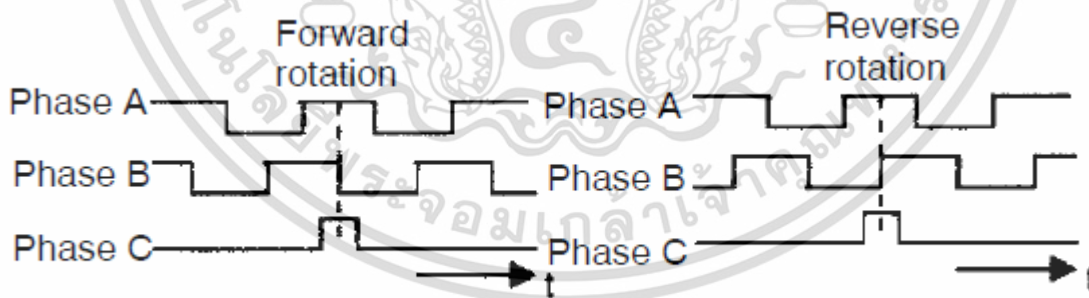
รูปที่ 2.38 โครงสร้างเอซีเซอร์โวมอเตอร์ (AC servo motor)

ส่วนสเตเตอร์จะเป็นแม่เหล็กไฟฟ้าที่มีขนาดเล็กแต่มีแรงบิดที่สูงจึงทำให้สามารถควบคุมการเคลื่อนที่และสลับทิศทางของเซอร์โวมอเตอร์ได้อย่างมีประสิทธิภาพ เพื่อรักษาแรงบิดและความเร็วให้คงที่ โดยเอซีเซอร์โวมอเตอร์จะมีสองแบบคือแบบซิงโครนัสเซอร์โวและอะซิงโครนัสเซอร์โวซึ่งจะทำงานคล้ายกับอินดักชันมอเตอร์ที่มีขดลวดอาร์เมเจอร์ เมื่อจ่ายกระแสเข้าไปยังขดลวดที่สเตเตอร์ด้วยมุมที่ต่างกัน 120 องศา ทำให้มอเตอร์หมุนด้วยความเร็วซิงโครนัส (Synchronous speed) ดังแสดงในรูปที่ 2.39



รูปที่ 2.39 การจ่ายกระแสให้กับขดลวดของเอซีเซอร์โวมอเตอร์ (AC servo motor)

ในการควบคุมทิศทางเคลื่อนที่ของเซอร์โวมอเตอร์อาศัยหลักการป้อนกระแสให้กับขดลวดสเตเตอร์ ในกรณีที่เฟส B นำหน้า เฟส A เซอร์โวมอเตอร์จะหมุนทวนเข็มนาฬิกา และถ้าเฟส A นำหน้า เฟส B เซอร์โวมอเตอร์จะหมุนตามเข็มนาฬิกา ดังแสดงในรูปที่ 2.40



รูปที่ 2.40 การควบคุมทิศทางเคลื่อนที่ของเซอร์โวมอเตอร์ของเอซีเซอร์โวมอเตอร์ (AC servo motor)

ทั้งนี้การออกแบบสร้างหุ่นยนต์ได้เลือกใช้เอซีเซอร์โวมอเตอร์ของบริษัท Yaksakawa รุ่น SGM A5A, 01A และ 02A โดยมีรายละเอียดของเอซีเซอร์โวมอเตอร์ดังตารางที่ 2.4

ตารางที่ 2.4 เอซีเซอร์โวมอเตอร์ของบริษัท Yaksakawa ที่ใช้ในการสร้างหุ่นยนต์

SGM Servomotor		A3A	A5A	01A	02A	04A	08A
Rated Output* ¹	W (HP)	30 (0.04)	50 (0.07)	100 (0.13)	200 (0.27)	400 (0.53)	750 (1.01)
Rated Torque* ¹ * ²	N.m	0.095	0.159	0.318	0.637	1.27	2.39
	(oz.in)	(13.5)	(22.6)	(45.1)	(90.1)	(181)	(338)
Instantaneous Peak Torque* ¹	N.m	0.29	0.48	0.96	1.91	3.82	7.1
	(oz.in)	(40.5)	(67.7)	(135)	(270)	(542)	(1010)
Rated Current* ¹	A (rms)	0.42	0.6	0.87	2.0	2.6	4.4
Instantaneous Max Current* ¹	A (rms)	1.3	1.9	2.8	6.0	8.0	13.9
Rated Speed* ¹	r/min	3000					
Instantaneous Max Speed* ¹	r/min	4500					
Torque Constant* ¹	N.m/A (rms)	0.255	0.286	0.408	0.355	0.533	0.590
	(oz.in/A) (rms)	(36.2)	(40.5)	(57.8)	(50.2)	(75.5)	(83.5)
Moment of Inertia [J _M]	kg.m ² × 10 ⁻⁴	0.021	0.026	0.040	0.123	0.191	0.671
	(oz.in.s ² × 10 ⁻³)	(0.288)	(0.368)	(0.576)	(1.74)	(2.70)	(9.52)
Rated Power Rate* ¹	kW/s	4.36	9.63	25.4	32.8	84.6	85.1
Rated Angular Acceleration* ¹	rad/s ²	45200	61200	79500	51800	666000	35600
Inertia Time Constant	ms	1.5	0.9	0.5	0.4	0.3	0.3
Inductive Time Constant	ms	1.5	1.8	1.9	5.4	6.4	13

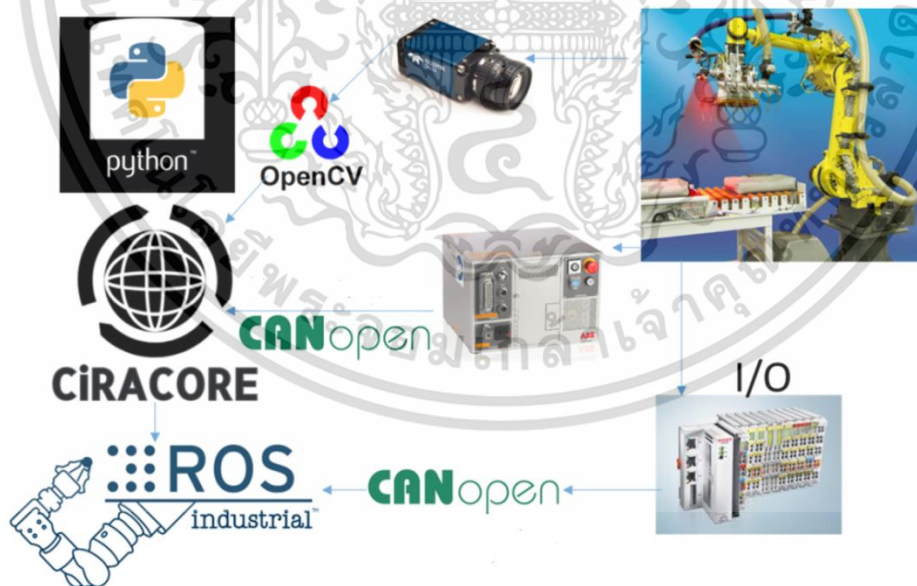
บทที่ 3

วิธีการวิจัย

เนื้อหาในบทนี้ได้อธิบายแนวคิดในการพัฒนาระบบปฏิบัติการแขนกลอุตสาหกรรม การออกแบบระบบปฏิบัติการแขนกลอุตสาหกรรม และการพัฒนาซอฟต์แวร์ที่ควบคุมหุ่นยนต์อุตสาหกรรมให้สามารถเคลื่อนที่ได้ทั้ง 4 แกนสัมพันธ์กันและมีความคลาดเคลื่อนต่ำเหมาะกับการใช้งานบนพื้นฐานระบบปฏิบัติการหุ่นยนต์ ROS (Robot Operating System) ได้

3.1 แนวคิดของงานวิจัย

จากแนวคิดดังกล่าวนั้นการเปลี่ยนเป็นระบบการผลิตที่ใช้เครื่องจักรมากขึ้นเพื่อลดการพึ่งพาแรงงานคนในขณะเดียวกันก็เพิ่มกำลังการผลิตและเพิ่มประสิทธิภาพในกระบวนการผลิตของภาคอุตสาหกรรมด้วยระบบปฏิบัติการที่ควบคุมหุ่นยนต์ (ROS) เข้ากับฮาร์ดแวร์ที่เกี่ยวข้องกับหุ่นยนต์ต่างๆ ซึ่งจากปัญหาที่กล่าวมาเหล่านี้ นำไปสู่โครงการวิจัยและพัฒนาในหัวข้อ “การพัฒนาหุ่นยนต์อุตสาหกรรมด้วยซอฟต์แวร์โอเพนซอร์ส ROS” ดังแสดงให้เห็นการพัฒนาโครงการโดยย่อดังรูปที่ 3.1 ด้านล่าง



รูปที่ 3.1 แนวคิดในการพัฒนาหุ่นยนต์อุตสาหกรรมด้วยเทคโนโลยีโอเพนซอร์สสำหรับกระบวนการผลิตในอุตสาหกรรม

3.2 การออกแบบระบบปฏิบัติการแขนกลอุตสาหกรรม

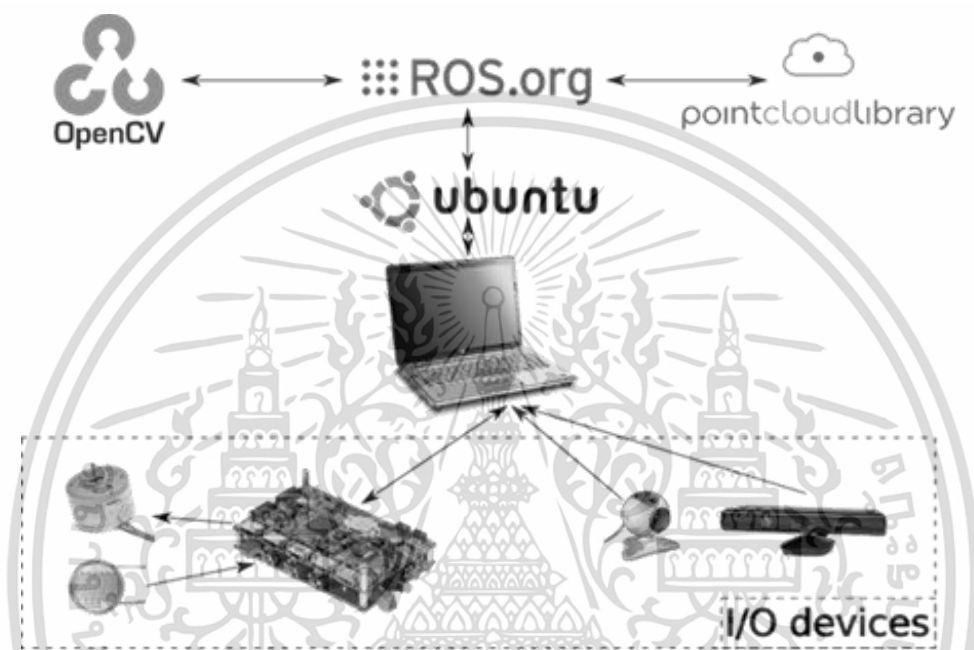
จากแนวคิดหุ่นยนต์อุตสาหกรรม ABB IRB6640 ถูกใช้สาธิตเพื่อแสดงให้เห็นว่าระบบที่ใช้ ROS-Industrial สามารถรับมือกับผลิตภัณฑ์ที่ไม่รู้จักและมีความไม่แน่นอนในสิ่งแวดล้อมได้ง่ายเพียงใช้เวลาในการติดตั้งฮาร์ดแวร์เพียงเล็กน้อยสำหรับฮาร์ดแวร์ที่ใช้งานจริง พาเลทสองชั้นวางอยู่ในพื้นที่ทำงานของหุ่นยนต์ หนึ่งในพาเลทเหล่านี้มีกล่องสามขนาดที่ไม่รู้จักวางแบบสุ่ม ขนาดตำแหน่งและทิศทางของกล่องถูกกำหนดโดยใช้กล้อง Kinect, Openni package และไลบรารี PCL จากข้อมูลนี้เส้นทางถูกคำนวณโดยใช้ไลบรารี MoveIt! และถูกส่งไปยังตัวควบคุมหุ่นยนต์โดยใช้แพคเกจ ABB ROS-Industrial ดังแสดงในรูปที่ 3.2 เพื่อทำการทดสอบออฟไลน์ของการเชื่อมต่อระหว่าง ROS-PC และตัวควบคุมการเคลื่อนไหวของ ABB IRB6640 ได้ทำการติดตั้ง ROS-PC ที่เชื่อมต่อกับเครื่องพีซีที่ใช้ซอฟต์แวร์ ABB Robot Studio ของ ABB อย่างเป็นทางการ ทำให้ฮาร์ดแวร์ลดเวลาดำเนินการลงเหลือน้อยกว่าหนึ่งวัน การใช้งานในปัจจุบันมีความแม่นยำในการจับประมาณ 1 เซนติเมตร ซึ่งแสดงถึงความถูกต้องของระบบวิชั่นสามารถปรับปรุงได้ ซึ่งสามารถทำได้โดยใช้เซ็นเซอร์ที่แม่นยำยิ่งขึ้น (เช่น laser triangulation, Time of Flight cameras) และอัลกอริทึมฟิลเตอร์ขั้นสูงอื่น ๆ สำหรับผู้ใช้งานนี้ต้องมีการปรับไดรเวอร์ ABB ที่มีอยู่เพื่อให้สามารถควบคุมการจับยึดได้โดยใช้อินเทอร์เฟซ I/O ของคอนโทรลเลอร์ IRC5 ซึ่งบ่งบอกถึงความจำเป็นในการใช้งาน I/O interface ใน ROS-Industrial อย่างง่ายดาย



รูปที่ 3.2 หุ่นยนต์อุตสาหกรรม ABB IRB6640 ถูกใช้ทำงาน Palletizing ภายใต้ ROS-Industrial

โดยโครงการวิจัยนี้ได้ใช้ ROS (Robot Operating System) เพื่อในการพัฒนาหุ่นยนต์ในต้นทุนต่ำ ซึ่งเฟรมเวิร์ก (Framework) นี้เป็นโอเพนซอร์ส (Open source) นักพัฒนาสามารถพัฒนาโดยไม่มีต้นทุนทางด้านซอฟต์แวร์ ช่วยลดเวลาในการพัฒนา อีกทั้งมีความยืดหยุ่นสำหรับการเขียนโปรแกรมทางด้านหุ่นยนต์

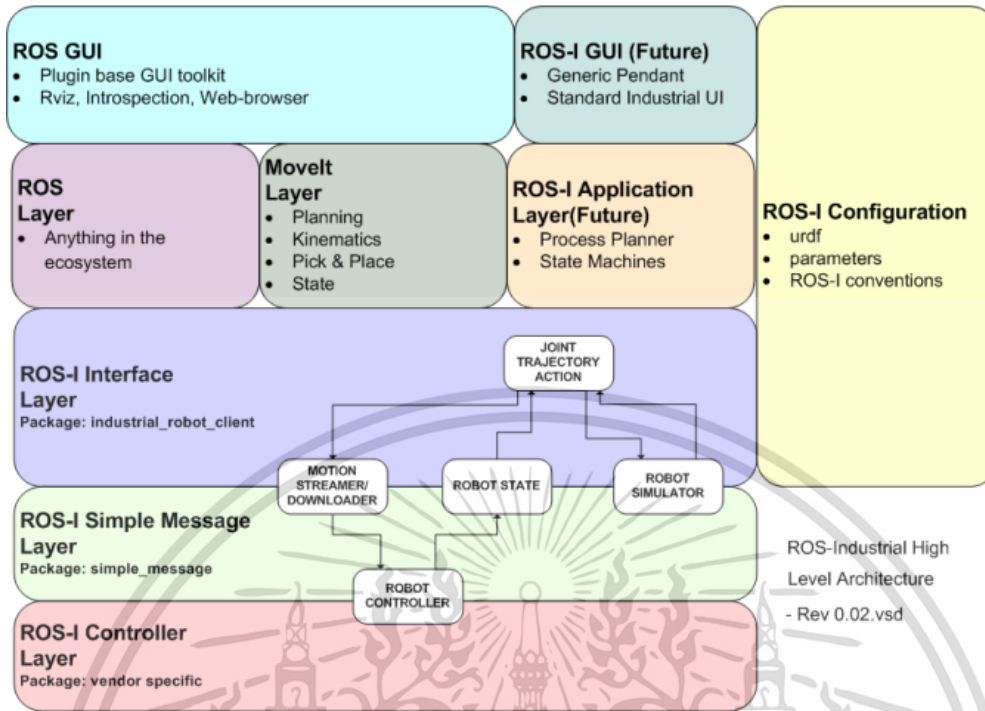
โดยได้รวบรวมเครื่องมือและไลบรารีไว้เป็นหมวดหมู่ ลดความซับซ้อนและเพิ่มประสิทธิภาพในการทำงานกับแพลตฟอร์มที่หลากหลายของหุ่นยนต์ ทั้งยังสามารถกระจายการทำงานแบบโหนดที่แตกต่างกันตามหน้าที่ของแต่ละ node โดยจะติดต่อสื่อสารกันผ่าน topic และ services ซึ่ง ROS นี้รองรับการทำงานร่วมกับระบบปฏิบัติการเดเบียนลินุกซ์และอูบุนตุลินุกซ์ (Debian and Ubuntu Trusty (14.04 LTS) ขึ้นไป โดยการเชื่อมต่อระบบปฏิบัติการที่ควบคุมแขนกล (ROS) เข้ากับฮาร์ดแวร์ที่เกี่ยวข้องกับแขนกลต่างๆ แสดงดังรูปที่ 3.3



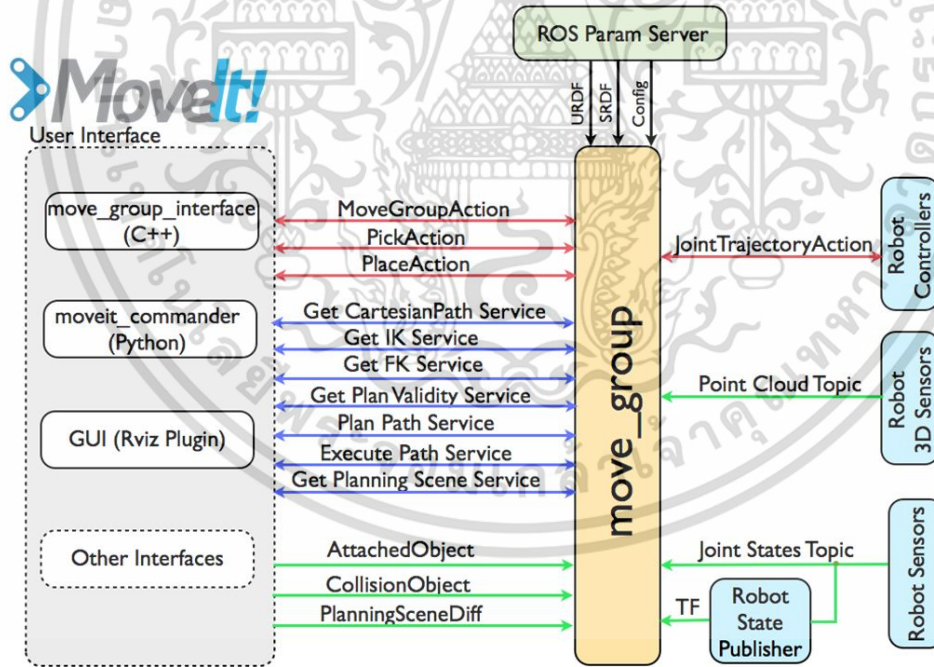
รูปที่ 3.3 ระบบปฏิบัติการที่ ROS และสามารถทำงานร่วมกับไลบรารีอื่นๆ

ดังนั้นการใช้ ROS รวบรวมไลบรารีและเครื่องมือไม่ว่าจะเป็น hardware abstraction, device drivers, libraries, visualizers, message-passing, package management และอื่นๆ อีกมากมาย เพื่อพัฒนาหุ่นยนต์อุตสาหกรรมสำหรับกระบวนการผลิตในอุตสาหกรรม

รูปที่ 3.4 แสดงสถาปัตยกรรม ของ ROS industrial โดยรูป 3.4(a) แสดงให้เห็นถึง Module ต่างๆ ที่ประกอบกันเพื่อเชื่อมโยงข้อมูลการควบคุม ตั้งแต่ระดับ controller ในแต่ละ joint ของแขนกล ไปจนถึง high-level GUI สำหรับผู้ใช้งานในการโปรแกรมหุ่นยนต์ให้ทำงานตามที่ต้องการ ส่วนในรูป 3.4(b) เป็นการแสดงให้เห็นถึงสถาปัตยกรรมของ ROS industrial ในแง่การสื่อสารข้อมูลในแต่ละ topic ของแต่ละโมดูล รูปที่ 3.5 แสดงการเชื่อมต่อของ ROS industrial กับหุ่นยนต์อุตสาหกรรม ผ่าน Module MoveIt ซึ่งสามารถเชื่อมต่อได้หลายยี่ห้อ หลายรุ่น

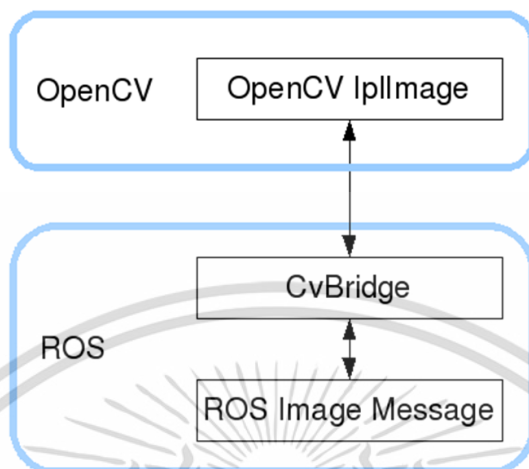


(a)

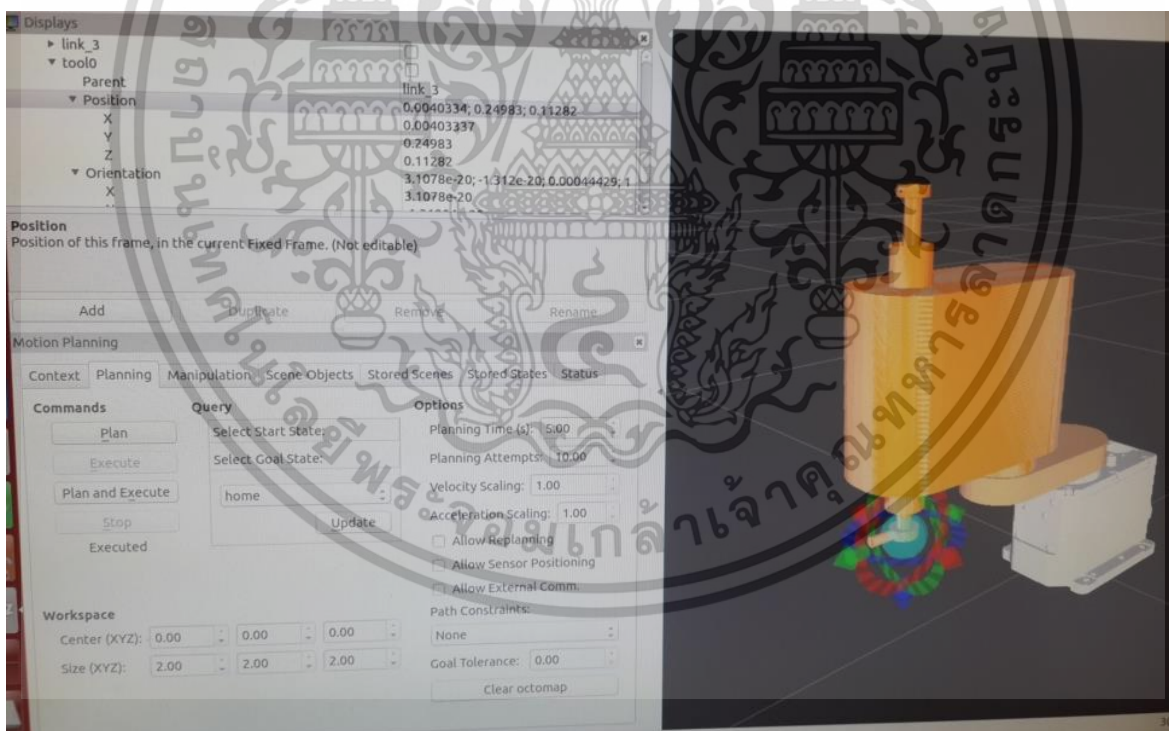


(b)

รูปที่ 3.4 (a) สถาปัตยกรรมของ ROS industrial (b) ในแง่ของการสื่อสาร



รูปที่ 3.5 การเชื่อม ROS industrial ผ่าน ROS Bridge กับ OpenCV

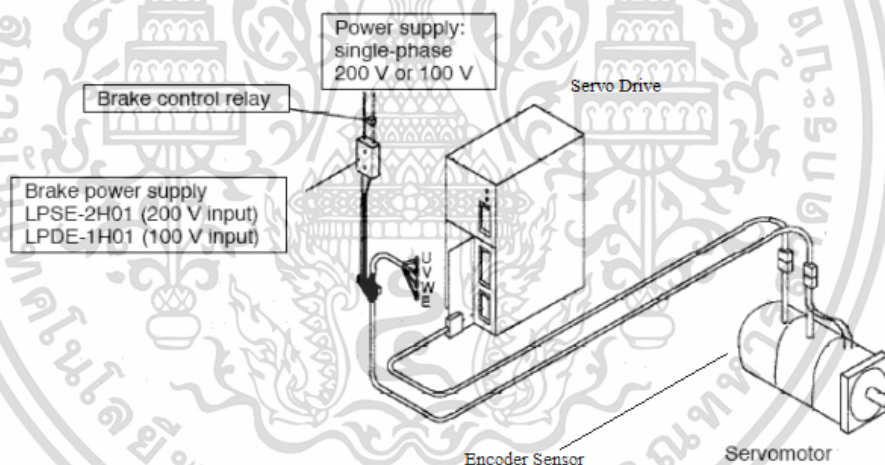


รูปที่ 3.6 การจำลองการทำงานของหุ่นยนต์ (Robot simulation) ด้วย ROS

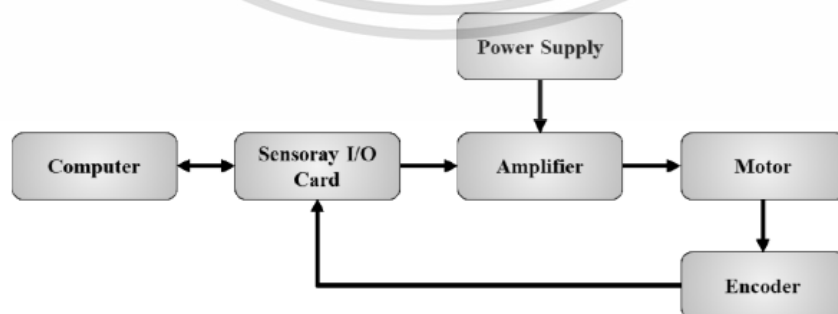
3.3 ระบบควบคุมหุ่นยนต์ที่ได้พัฒนาขึ้น

ระบบควบคุมหุ่นยนต์ที่สร้างขึ้นได้ถูกพัฒนาบนพื้นฐานการเชื่อมต่อผ่านโพรโทคอล CANopen ที่สามารถรับส่งข้อมูลแบบทันทีทันใดพร้อมกันได้ด้วยโปรแกรมภาษา C, C++ เพื่อการควบคุมการทำงานของ Servo Motor ของแต่ละข้อต่อ (Joint) แขนกล โดยมีรายละเอียดของชุดควบคุมการเคลื่อนที่หุ่นยนต์ทั้ง 4 แขนดังนี้

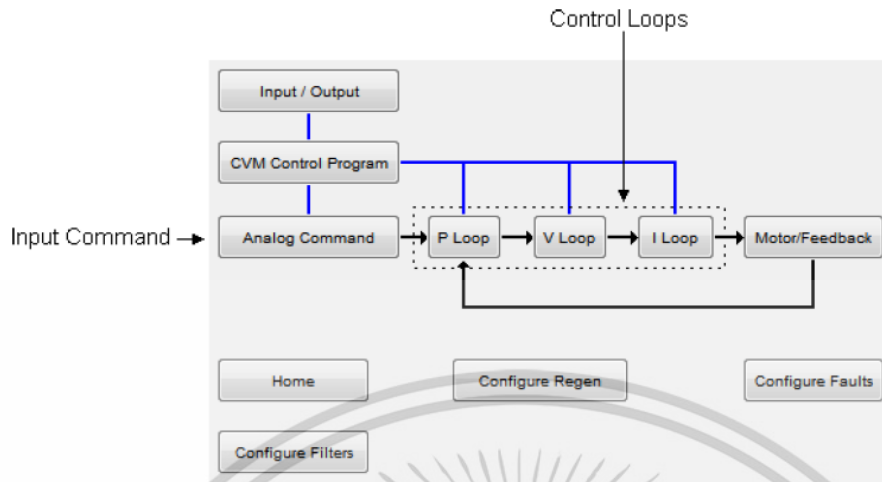
- ชุดขับเคลื่อนเซอร์โวมอเตอร์ (Servo Drive Motor) ที่สามารถควบคุม ความเร็ว, อัตราเร่ง และตำแหน่งการเคลื่อนที่ได้โดยใช้ Servo motor ผ่านชุดควบคุม ซึ่งจะส่งสัญญาณควบคุม (Signal command)
- มีโหมดในการควบคุมป้อนกลับแบบปิด (Closed loop control) โดยประกอบด้วยสามส่วนคือฟังก์ชันในการควบคุมกระแสไฟที่จ่ายให้กับมอเตอร์ (Current control loop) ฟังก์ชันในการควบคุมอัตราเร่ง (Velocity Control Mode) และฟังก์ชันในการควบคุมตำแหน่ง (Position Control Mode)
- มีเซนเซอร์เอนโค้ดเดอร์ (Encoder) เป็นชนิดโรตารีเอนโค้ดเดอร์ (Rotary Encoder) และมีความละเอียดจำนวนรอบที่ 2500 P/R จำนวน 8 โพล (pole) สามารถทำงานได้ทั้งแบบ sine และ cosine



รูปที่ 3.7 ไตอะแกรมการควบคุมของ Servo Motor ด้วยชุดขับเคลื่อนเซอร์โวมอเตอร์



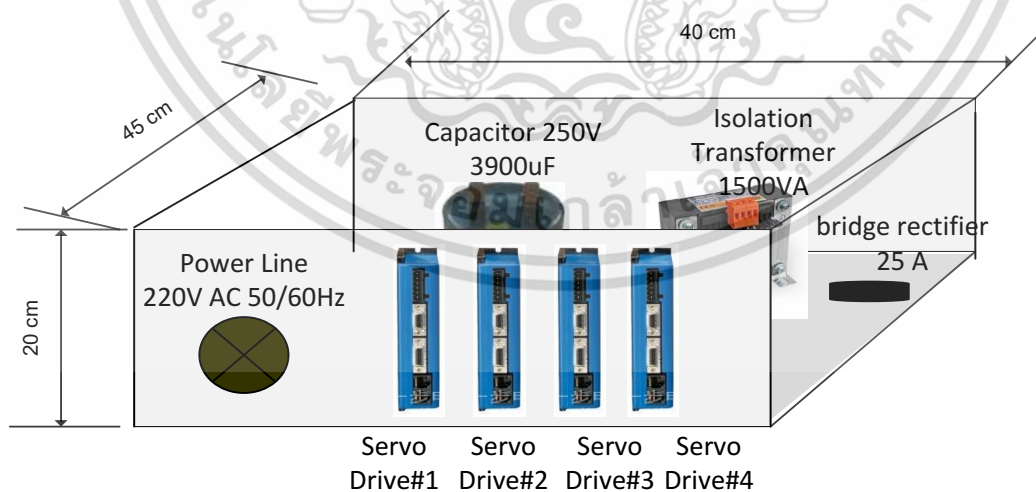
รูปที่ 3.8 ระบบควบคุมการทำงานของการทำงานของการควบคุมการทำงานของ Servo Motor



รูปที่ 3.9 ระบบควบคุมแบบปิดของ Servo Motor

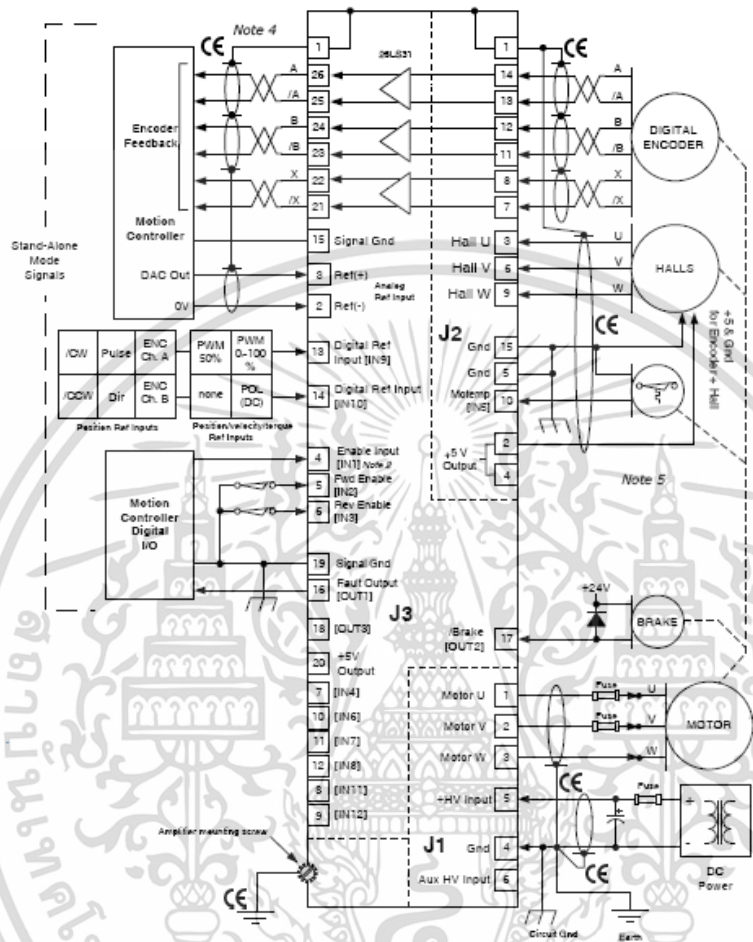
ทั้งนี้ชุดควบคุมการเคลื่อนที่หุ่นยนต์มีเครื่องมือ อุปกรณ์ และซอฟต์แวร์ที่ถูกตั้งค่าเกณฑ์ควบคุมไว้ในหน่วยความจำที่ถูกติดตั้งไว้ในชุดควบคุมเพื่อขับเคลื่อนเซอร์โวมอเตอร์จำนวน 4 แกน ดังแสดงในรูปที่ 3.10

- โปรแกรมชุดควบคุม CME 2 Version 7.0 และ Amplifier firmware version 1.90
- ชุด Servo drive :ADP-180-09
- ชุด Encoder ADP
- Servo motor YASKAWA 200AC 100W 4Pole



รูปที่ 3.10 ชุดควบคุมหุ่นยนต์ SCARA robot

โดยการต่อวงจร Servo Motor connection และวงจร Encoder กับชุดควบคุม Motion Controller ตามรูปที่ 3.11



รูปที่ 3.11 การต่อวงจร Motor connection และ Incremental Encoder

3.4 ขั้นตอนในการติดตั้ง ROS

ROS Kinematic เป็นสมการการคำนวณทางคณิตศาสตร์การเคลื่อนที่ โดยคำนวณตำแหน่งของหุ่นยนต์ว่าจะเคลื่อนที่ไปยังเป้าหมายที่จุด XYZ ด้วยการหมุนก๊องศา เคลื่อนที่ไปก๊องศา ถึงจะมาถึง ณ ตำแหน่งเป้าหมายได้ โดยแปลงค่าจาก joint เป็น XYZ ซึ่งอาศัยการ Forward /Inverse Kinematic มา Solver หาค่าพารามิเตอร์ในการควบคุมทิศทางเคลื่อนที่ในแต่ละแบบ

ขั้นตอนการติดตั้ง ROS โดยมีลำดับดังนี้

1. ติดตั้ง ROS Kinetic บน Linux Ubuntu 16.04

- a. ติดตั้งแพ็คเกจด้วยที่ไฟล์ sources.list ใน path: /etc/apt/sources.list.d/ros-latest.list' หลังจากนั้นพิมพ์คำสั่ง

```
>>sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main"
```

- b. ติดตั้ง key เฉพาะ ด้วยคำสั่ง

```
>>sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
```

- c. Update แพ็คเกจเพิ่มเติมด้วยคำสั่ง

```
>>sudo apt-get update
```

- d. ติดตั้ง ROS Desktop ที่ประกอบด้วยไลบรารี rqt, rviz, robot-generic libraries, การแสดงผลแบบ 2D/3D, ระบบนำทาง และกลไกการรับรู้ด้วย 2D/3D โดยใช้คำสั่ง

```
>>sudo apt-get install ros-kinetic-desktop-full
```

- e. ก่อนใช้ ROS ต้องติดตั้งค่าเริ่มต้นด้วย ROSdep ที่รวบรวมคอมโพเนนท์ต่าง ๆ เข้าด้วยกันด้วยคำสั่ง

```
>>sudo rosdep init
```

```
>>rosdep update ต
```

f. ติดตั้งค่าพารามิเตอร์ต่าง ๆ ของสภาพแวดล้อมของหุ่นยนต์ ด้วยคำสั่ง

```
>>echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
```

```
>>source ~/.bashrc
```

g. ติดตั้งแพ็คเกจที่ใช้ในการรันโปรแกรมด้วยคำสั่ง

```
>>sudo apt-get install python-rosinstall python-rosinstall-generator python-  
wstool build-essential
```

h. คำสั่งเรียก sourcec ของ ROS ด้วยคำสั่ง

```
>>source /opt/ros/kinetic/setup.bash
```

i. สร้าง workspace สำหรับ ROS โดยในโฟลเดอร์ ด้วยคำสั่ง

```
>>catkin source /opt/ros/kinetic/setup.bash
```

j. ติดตั้ง Moveit สำหรับแสดงการเคลื่อนที่หุ่นยนต์ ด้วยคำสั่ง

```
>>sudo apt-get install ros-kinetic-moveit
```

k. หลังจากนั้นทำการโหลดไฟล์ URDF (ที่ได้จากหุ่นยนต์) มาตั้งค่าใน Moveit Setup Assitance

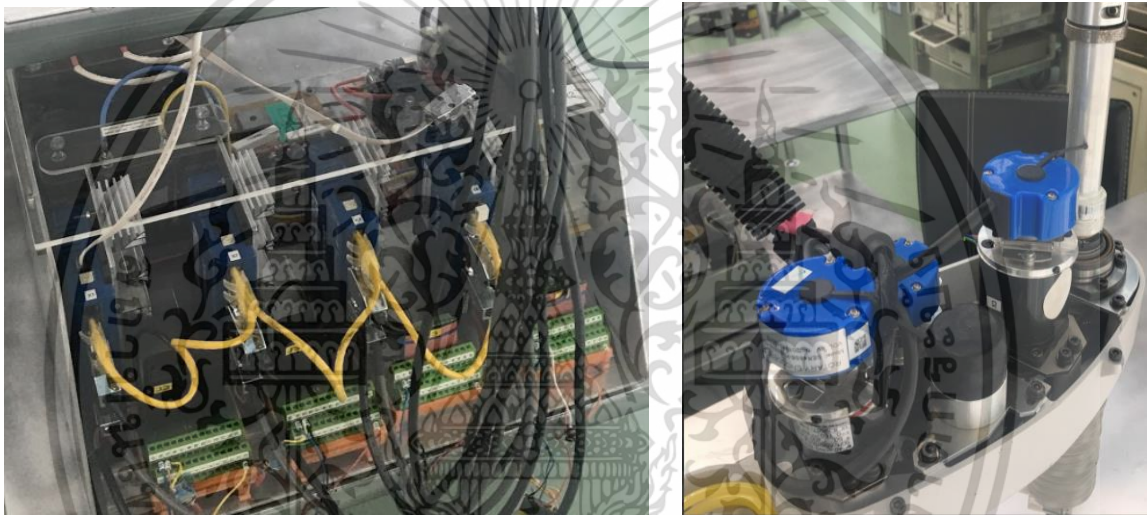
l. จำลองหุ่นยนต์โดยใช้ RViz ได้ผลจากการติดตั้ง ROS ดังแสดงในบทที่ 4

บทที่ 4

ผลการทดสอบ

4.1 การทดสอบควบคุมการทำงานของ SCARA robot ด้วย PID Control

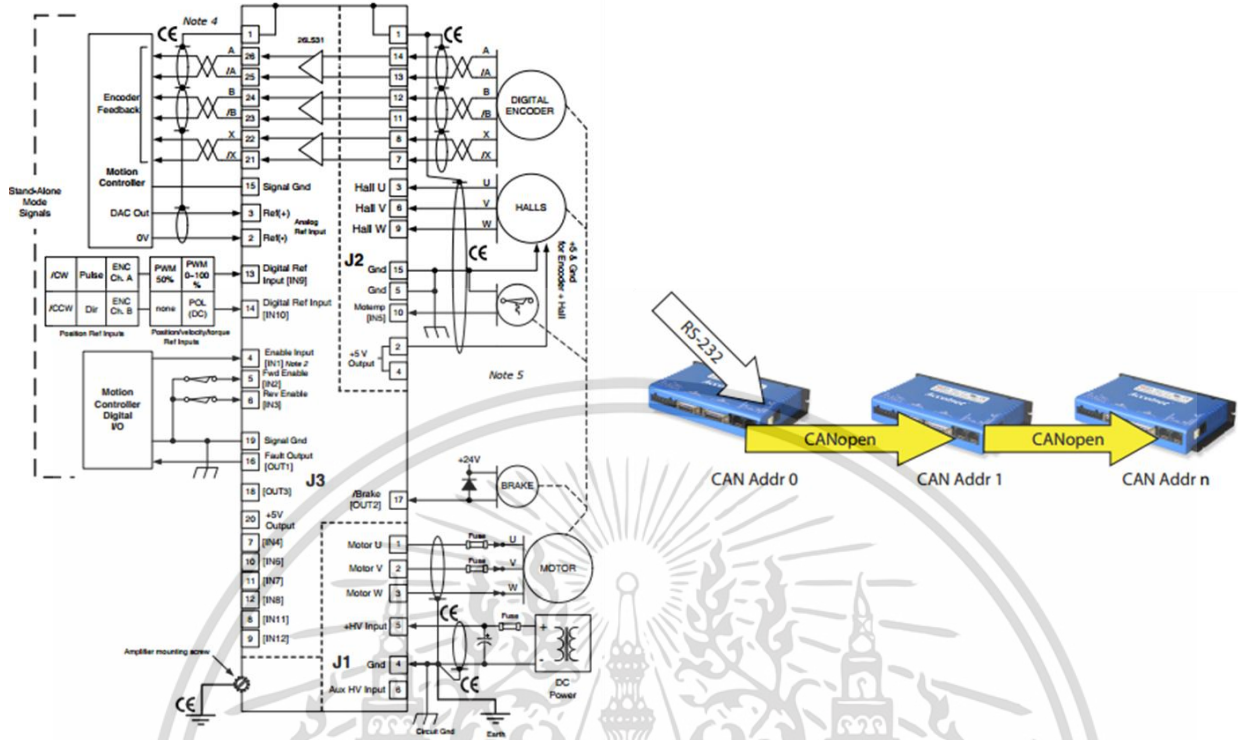
การควบคุมการทำงานของ AC Servo motor และการเขียนโปรแกรมควบคุม AC Servo motor สำหรับจำลองการหมุนของมอเตอร์ 360, 180, 90, 60 องศา โดยมีเครื่องมือและอุปกรณ์การจากการติดตั้งหุ่นยนต์ โดยใช้โปรแกรมชุดควบคุม CME 2 เพื่อขยับชุดควบคุมการเคลื่อนที่ของหุ่นยนต์



รูปที่ 4.1 โครงสร้างของการทำงานของ SCARA robot

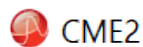
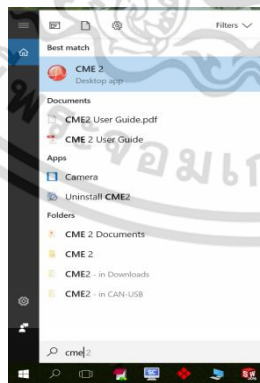
ทั้งนี้ มีขั้นตอนในการทดสอบการทำงานของหุ่นยนต์

1. ต่อวงจร Servo Motor connection และวงจร Encoder กับชุดควบคุม Motion Controller ตามรูป



รูปที่ 4.2 โครงสร้างทางฮาร์ดแวร์ของหุ่นยนต์

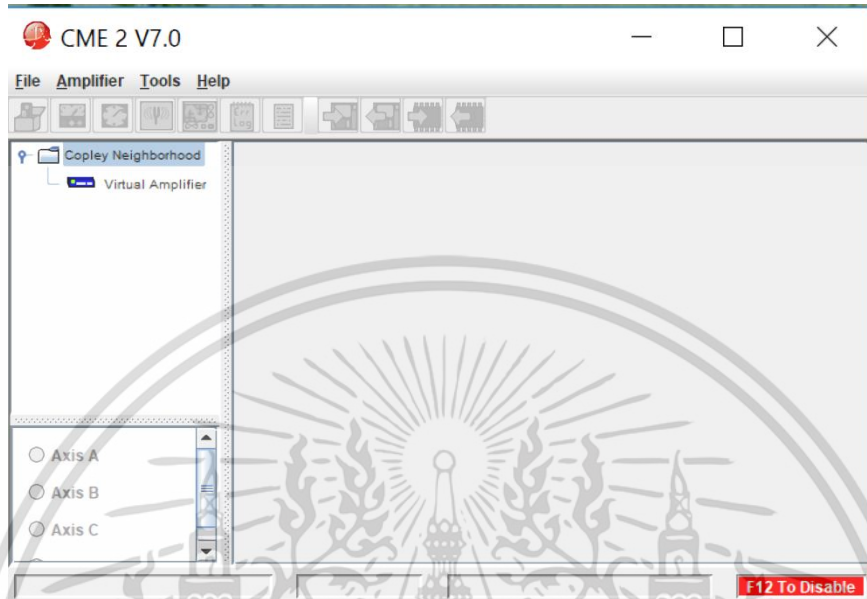
- เปิดระบบปฏิบัติการ Windows โดย SCARA robot มีการเชื่อมต่อผ่าน CAN bus ทั้งหมด 4 ID (CAN1, CAN2, CAN3, CAN4) ซึ่งจะควบคุมการทำงานทั้ง 4 แกน และการตั้งค่า PID Turning ในแต่ละแกนของหุ่นยนต์ โดยคลิกเลือก แกนที่จะทำการ Turning ผ่านโปรแกรม CME2 ตามรูป



รูปที่ 4.3 ไอคอนโปรแกรม CME2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รันโปรแกรม Copley Motion โดยใช้คีย์บอร์ดเพื่อเปิดหน้าต่างโปรแกรมการทำงาน
Start > Program > CopleyMotion > CME2 > CME2



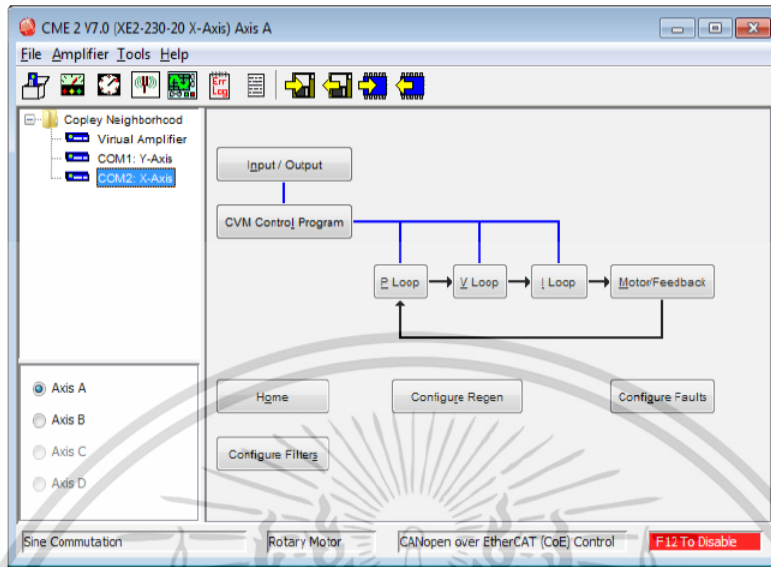
รูปที่ 4.4 โปรแกรมชุดควบคุม Copley Motion

- เมื่อเจอหน้าต่างโปรแกรม ให้กดเลือก Tool > Communication wizard



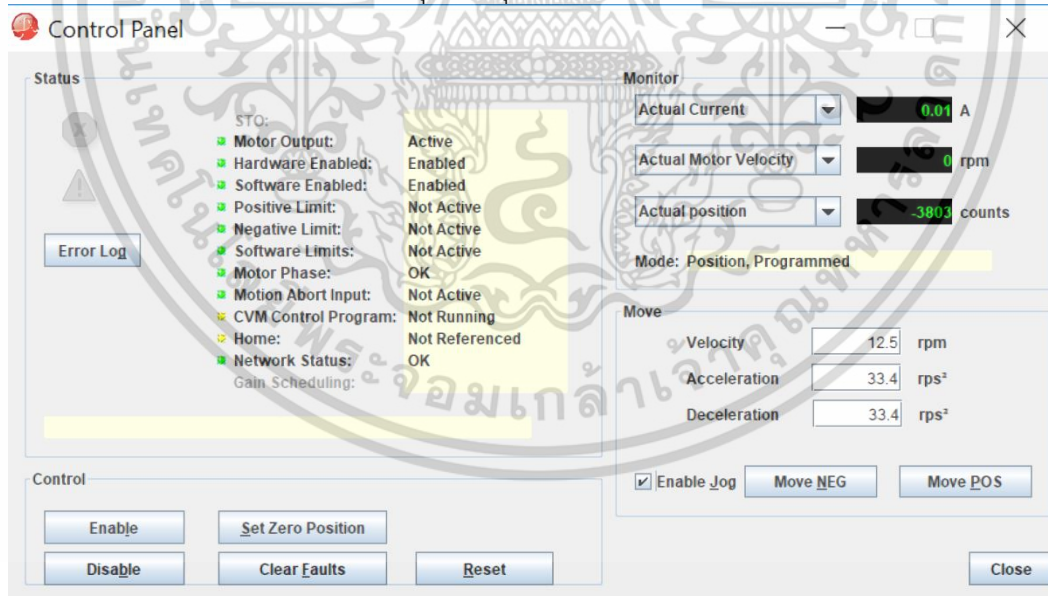
รูปที่ 4.5 โปรแกรมชุดควบคุม Copley Motion

5. จะแสดงหน้าต่างโปรแกรมทั้งหมดดังรูป



รูปที่ 4.6 หน้าต่างโปรแกรม CME2 ที่แสดงบล็อกไดอะแกรม PID control

6. ตรวจสอบสถานะการทำงานของชุดควบคุมมอเตอร์



รูปที่ 4.7 สถานะการทำงานของชุดควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. จากนั้นตั้งค่า Gain current ของ Stepping Motor แบบ 2 Phase
ขั้นตอนในการตั้งค่าและการหา Gain current

Step 1: แทนค่าพารามิเตอร์ของมอเตอร์ในการทำงานชุด Stepping motor

Step 2: กดปุ่ม I Loop หลังจากนั้นเลือกการตั้งค่าแบบ Maximize Speed

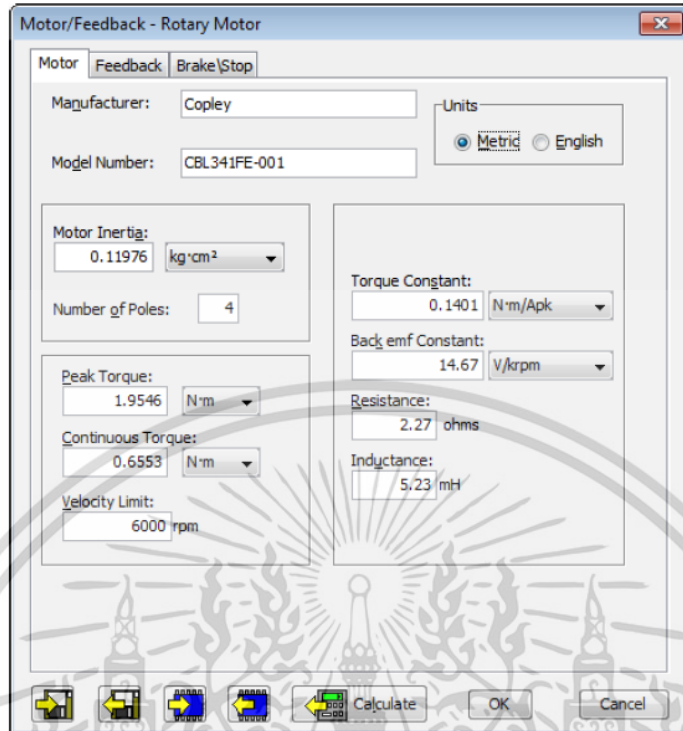
Step 3: กดปุ่ม Auto Tune โปรแกรมจะเริ่มกระบวนการในการหาค่าพารามิเตอร์ที่เหมาะสมของ
เกรนกระแสไฟฟ้า และแบนด์วิดของมอเตอร์จากการทดสอบ

Step 4: หลังจากนั้นกดปุ่ม Save Cp and Ci to flash และกดปุ่ม OK เลือกเกรนระดับเหมาะสม

แล้ว

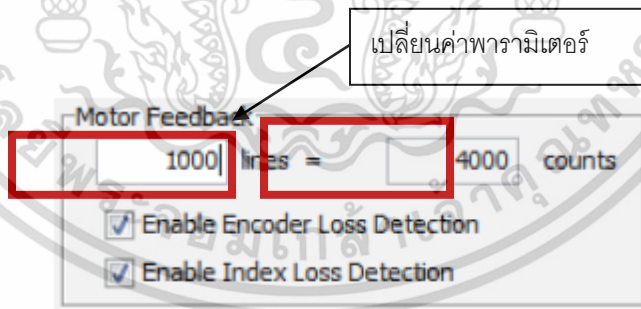
SGM Servomotor		A3A	A5A	01A	02A	04A	08A
Rated Output*1	W (HP)	30 (0.04)	50 (0.07)	100 (0.13)	200 (0.27)	400 (0.53)	750 (1.01)
Rated Torque*1 *2	N.m	0.095	0.159	0.318	0.637	1.27	2.39
	(oz-in)	(13.5)	(22.6)	(45.1)	(90.1)	(181)	(338)
Instantaneous Peak Torque*1	N.m	0.29	0.48	0.96	1.91	3.82	7.1
	(oz-in)	(40.5)	(67.7)	(135)	(270)	(542)	(1010)
Rated Current*1	A (rms)	0.42	0.6	0.87	2.0	2.6	4.4
Instantaneous Max Current*1	A (rms)	1.3	1.9	2.8	6.0	8.0	13.9
Rated Speed*1	r/min	3000					
Instantaneous Max Speed*1	r/min	4500					
Torque Constant*1	N.m/A (rms)	0.255	0.286	0.408	0.355	0.533	0.590
	(oz-in/A) (rms)	(36.2)	(40.5)	(57.8)	(50.2)	(75.5)	(83.5)
Moment of Inertia [J _M]	kg.m ² × 10 ⁻⁴	0.021	0.026	0.040	0.123	0.191	0.671
	(oz-in.s ² × 10 ⁻³)	(0.288)	(0.368)	(0.576)	(1.74)	(2.70)	(9.52)
Rated Power Rate*1	kW/s	4.36	9.63	25.4	32.8	84.6	85.1
Rated Angular Acceleration*1	rad/s ²	45200	61200	79500	51800	666000	35600
Inertia Time Constant	ms	1.5	0.9	0.5	0.4	0.3	0.3
Inductive Time Constant	ms	1.5	1.8	1.9	5.4	6.4	13

รูปที่ 4.8 แทนค่าพารามิเตอร์ของ Servo มอเตอร์ที่ได้



รูปที่ 4.9 แทนค่าพารามิเตอร์ของมอเตอร์ที่ได้

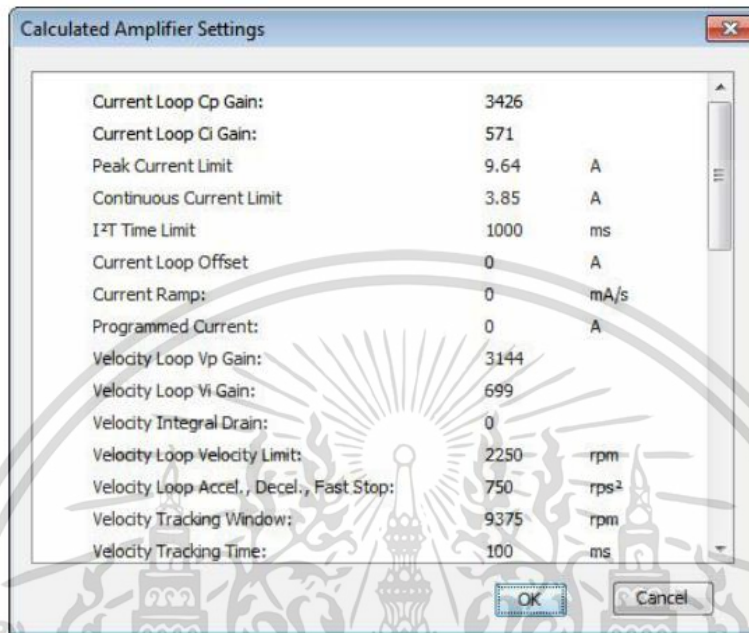
- กำหนดค่าพารามิเตอร์ Feedback Parameters, Rotary ของ Servo motor YASKAWA 200AC 100W 4Pole โดยมีค่าเท่ากับ 2500 lines =10000 count



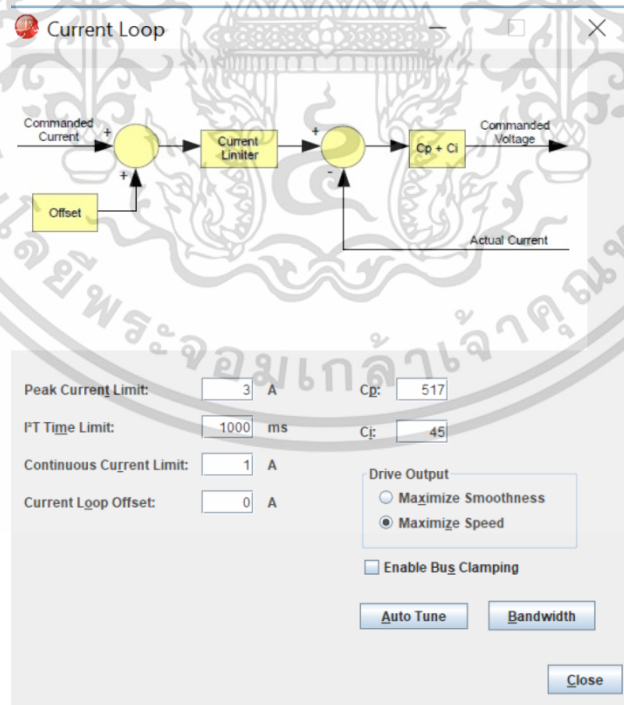
รูปที่ 4.10 หน้าต่างโปรแกรมของการคำนวณหาค่า count



9. การคำนวณค่าพารามิเตอร์โดยรวมด้วยฟังก์ชัน Calculate

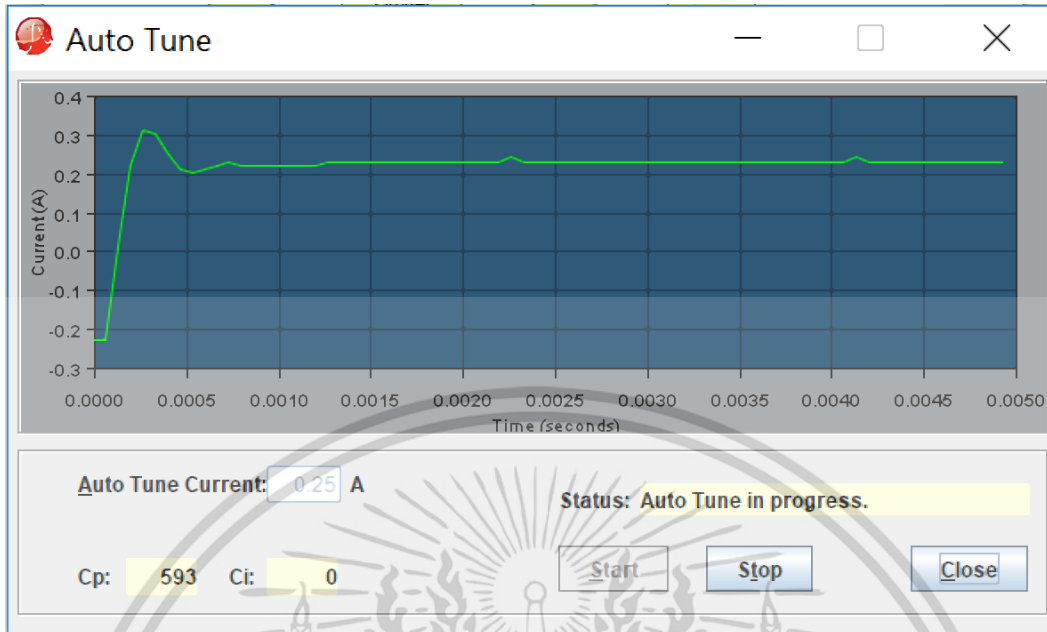


รูปที่ 4.11 หน้าต่างโปรแกรมของการหาค่าพารามิเตอร์ Servo motor

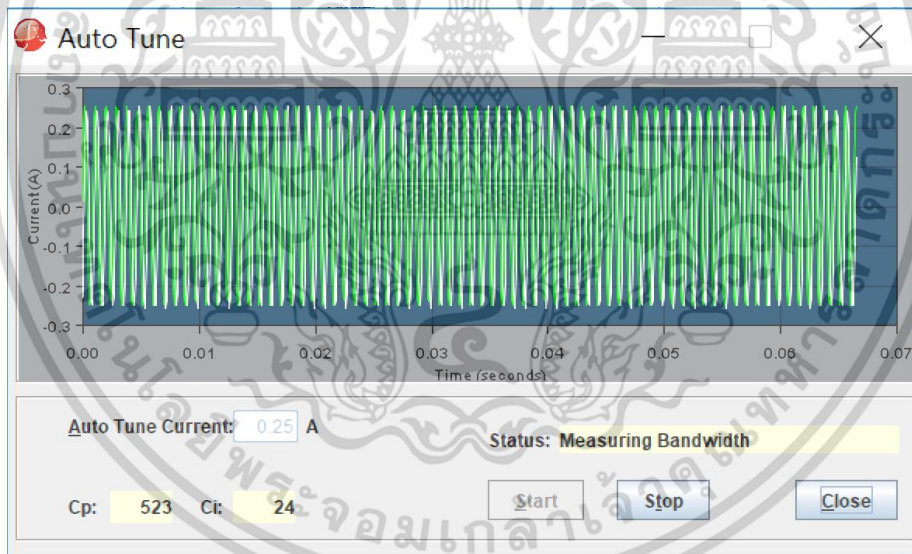


รูปที่ 4.12 หน้าต่างโปรแกรมของการหาค่า Gain current

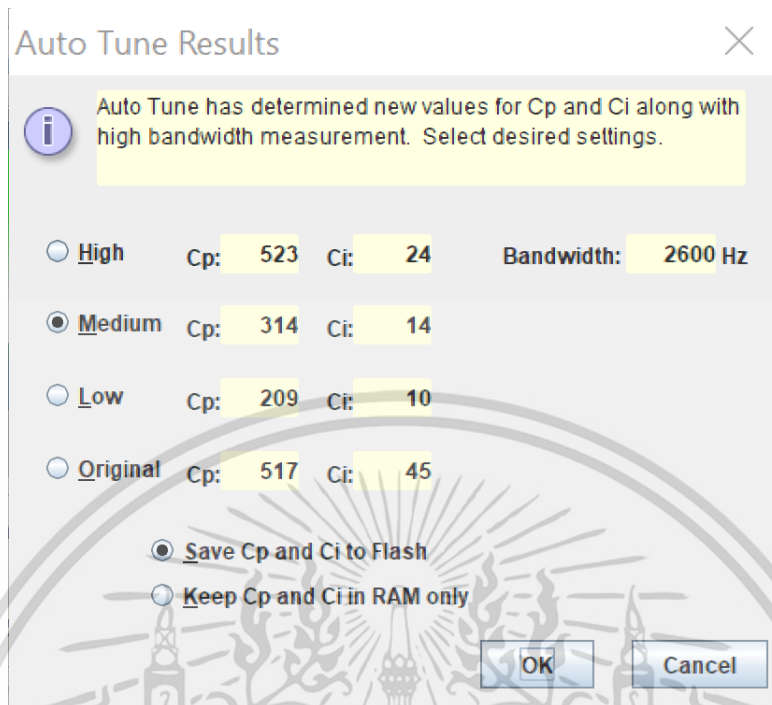
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ชุดควบคุมหาค่าที่เหมาะสมของผลตอบสนองทางระบบ



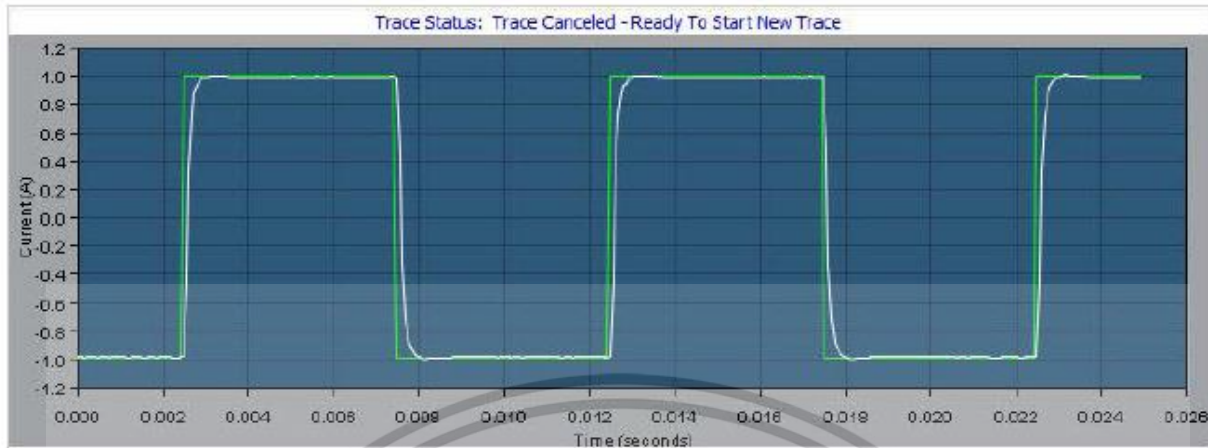
รูปที่ 4.14 ผลตอบสนองระบบทางด้านแบนด์วิธของชุด Control



รูปที่ 4.15 หน้าต่างโปรแกรมของเกรนระบบ

Step 5: เลือกหน้าต่างโปรแกรม Amplifier>Scope

- ปรับโมดการทำงานของฟังก์ชันกำเนิดสัญญาณ (Function Generator)
 - การเลือก Apply to : โมด Current
 - การเลือก Function: โมด Square Wave สำหรับการรูปแบบสัญญาณป้อนเข้ายังมอเตอร์
 - การเลือก Amplitude : 0.1 A สำหรับการทดสอบกระแสของมอเตอร์
 - การเลือก Frequency : 2 Hz สำหรับการดูช่วงความถี่แสดงผล
- เลือกการแสดงผลของการทดสอบ (Settings) ของโปรแกรม
 - CH1 เลือก Command current
 - CH2 เลือก Actual current motor
- กดปุ่ม Start เพื่อดูสัญญาณที่ได้เทียบกับสัญญาณ Command current



รูปที่ 4.16 ผลตอบสนองสัญญาณที่ได้เทียบกับสัญญาณ Command current

10. การตั้งค่า Gain Velocity Loop ด้วยการ Turning

ขั้นตอนในการตั้งค่า

Step 1: กดปุ่ม V Loop หลังจากนั้นเลือกการตั้งค่าแบบ Maximize Velocity = 2500 rpm และเลือกเกรนเริ่มต้น $V_p=8000$, $V_i=250$ (โดย V_p มีค่าโดยประมาณมากกว่า V_i 30 เท่า ซึ่งพิจารณาจากรูปแบบสัญญาณที่ได้เป็นหลักในการปรับค่า) จากนั้นกดปุ่ม Close

Step 2: เลือกหน้าต่างโปรแกรม Amplifier>Scope

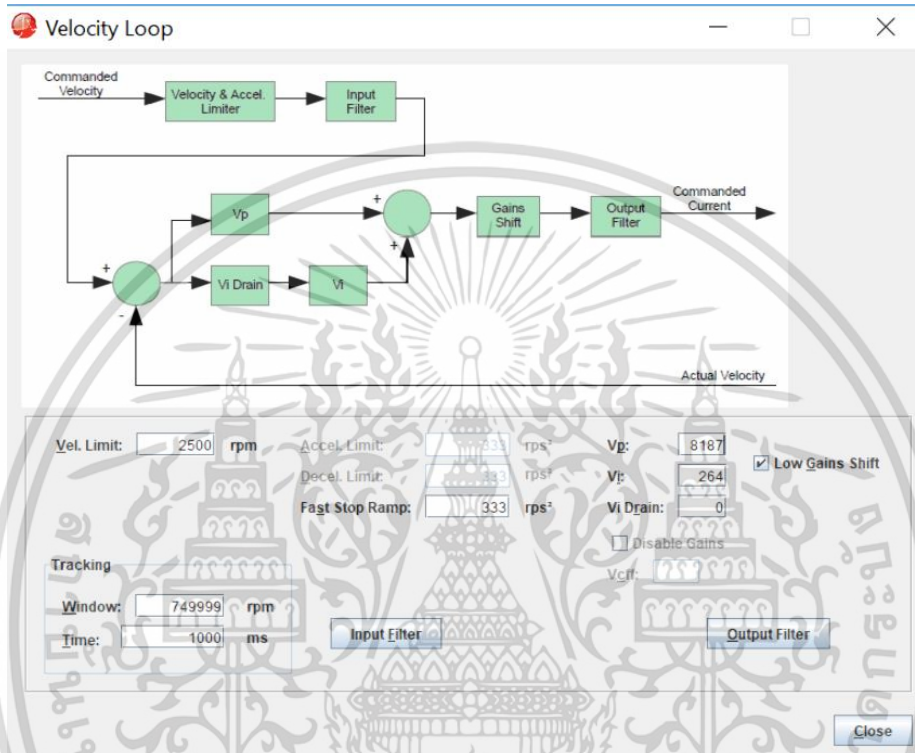
- ปรับโหมดการทำงานของฟังก์ชันกำเนิดสัญญาณ (Function Generator)
 - การเลือก Apply to : โหมด Velocity
 - การเลือก Function: โหมด Square Wave สำหรับการรูปแบบสัญญาณป้อนเข้ายังมอเตอร์
 - การเลือก Amplitude : 200 rpm สำหรับการทดสอบความเร็วของมอเตอร์
 - การเลือก Frequency : 2 Hz สำหรับการดูช่วงความถี่แสดงผล
- เลือกการแสดงผลของการทดสอบ (Settings) ของโปรแกรม
 - CH1 เลือก Command Velocity
 - CH2 เลือก Actual Velocity
- กดปุ่ม Start เพื่อดูสัญญาณที่ได้เทียบกับสัญญาณ Command Velocity

Step 3: หลังจากนั้นเลือกหน้าต่าง Gains ปรับค่าเกนของ Velocity loop ในระดับเหมาะสม

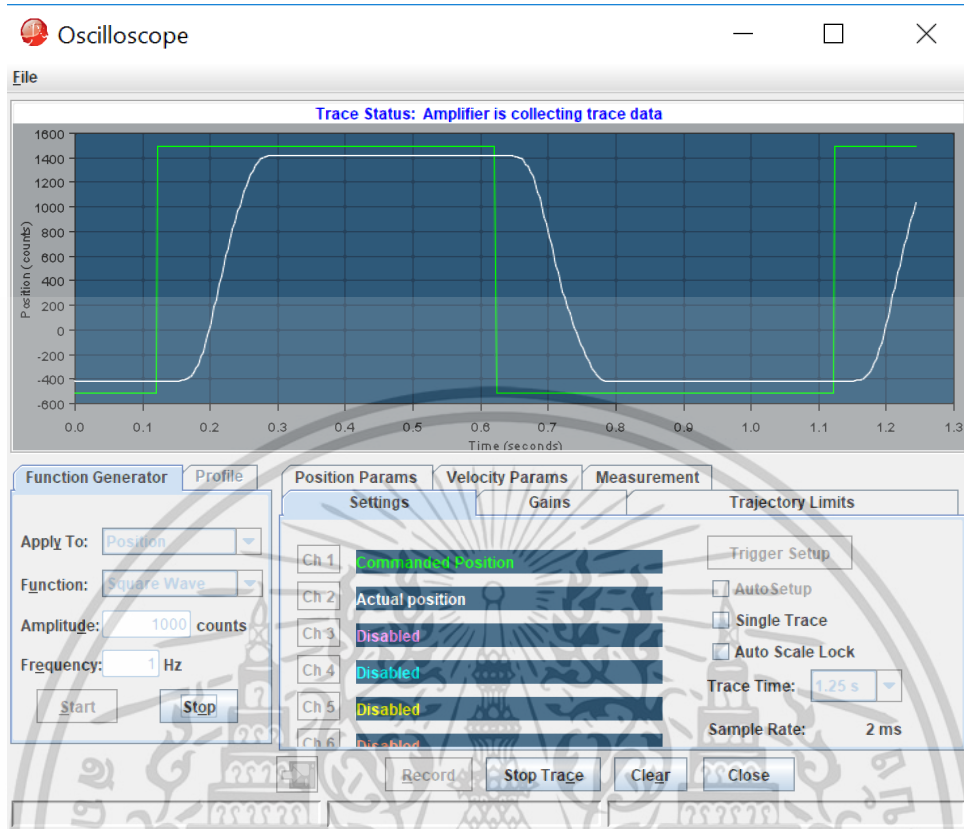
- เลือกเกน $V_p=8000$

- เลือกเกน $V_i=250$

และเมื่อสัญญาณที่ได้ใกล้เคียงกับสัญญาณ Command กดปุ่ม Close



รูปที่ 4.17 หน้าต่างการตั้งค่า V Loop



รูปที่ 4.18 หน้าต่างฟังก์ชัน Scope ของ Velocity Loop ด้วยการ Turning

11. การตั้งค่า Gain Position Loop ด้วยการ Turning

ขั้นตอนในการตั้งค่า

Step 1: กดปุ่ม P Loop หลังจากนั้นเลือกเกรนเริ่มต้น $P_p=6000$, $Aff=250$ และ

$V_{ff} = 16384$ จากนั้นกดปุ่ม Close

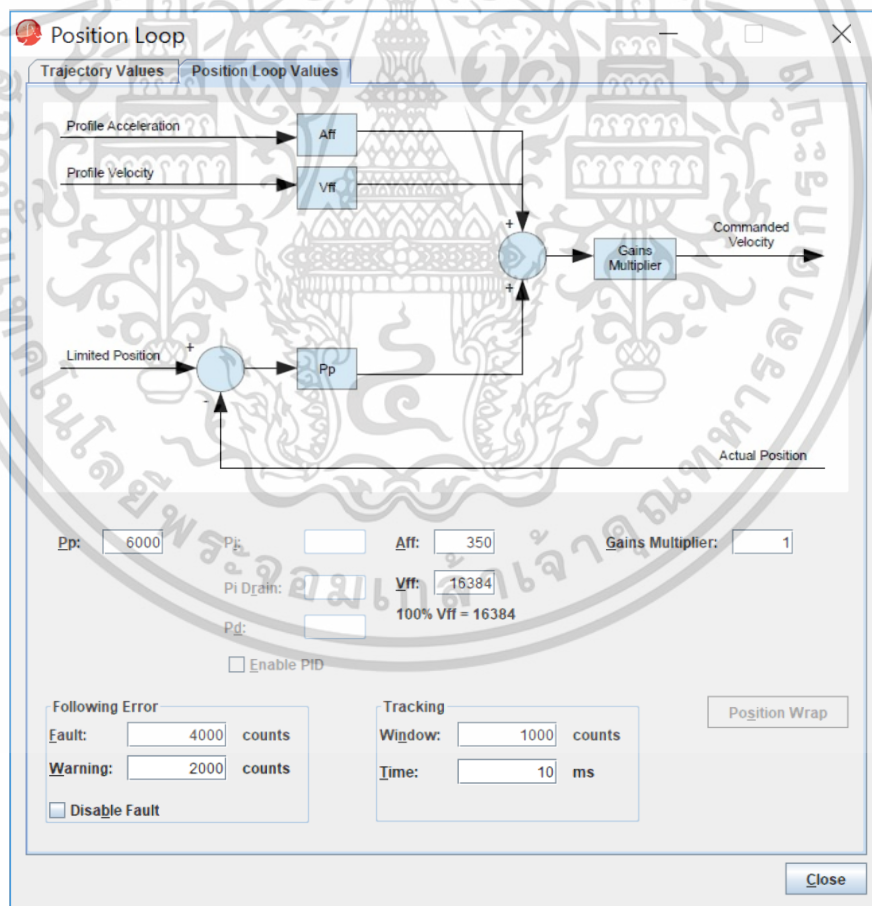
Step 2: เลือกหน้าต่างโปรแกรม Amplifier>Scope

- ปรับโหมดการทำงานของฟังก์ชันกำเนิดสัญญาณ (Function Generator)
 - การเลือก Apply to : โหมด Position
 - การเลือก Function: โหมด Square Wave สำหรับการรูปแบบสัญญาณป้อนเข้ายังมอเตอร์

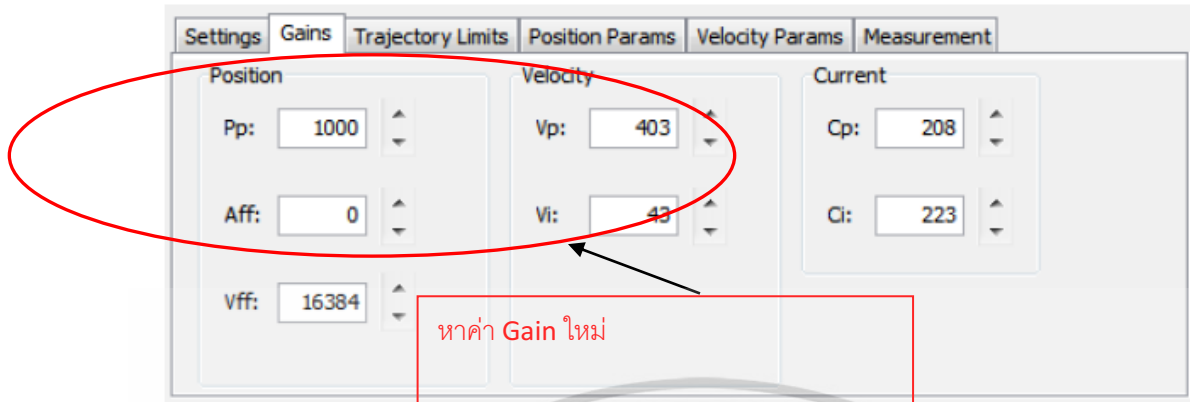
- การเลือก Amplitude : 1000 counts สำหรับการทดสอบตำแหน่งในการควบคุมร่วมกับชุด Encoder
- การเลือก Frequency : 1 Hz สำหรับการดูช่วงความถี่แสดงผล
- เลือกการแสดงผลของการทดสอบ (Settings) ของโปรแกรม
 - CH1 เลือก Command Position
 - CH2 เลือก Actual Position
- กดปุ่ม Start เพื่อดูสัญญาณที่ได้เทียบกับสัญญาณ Command Position

Step 3: หลังจากนั้นเลือกหน้าต่าง Gains ปรับค่าเกนของ Position loop ในระดับเหมาะสม

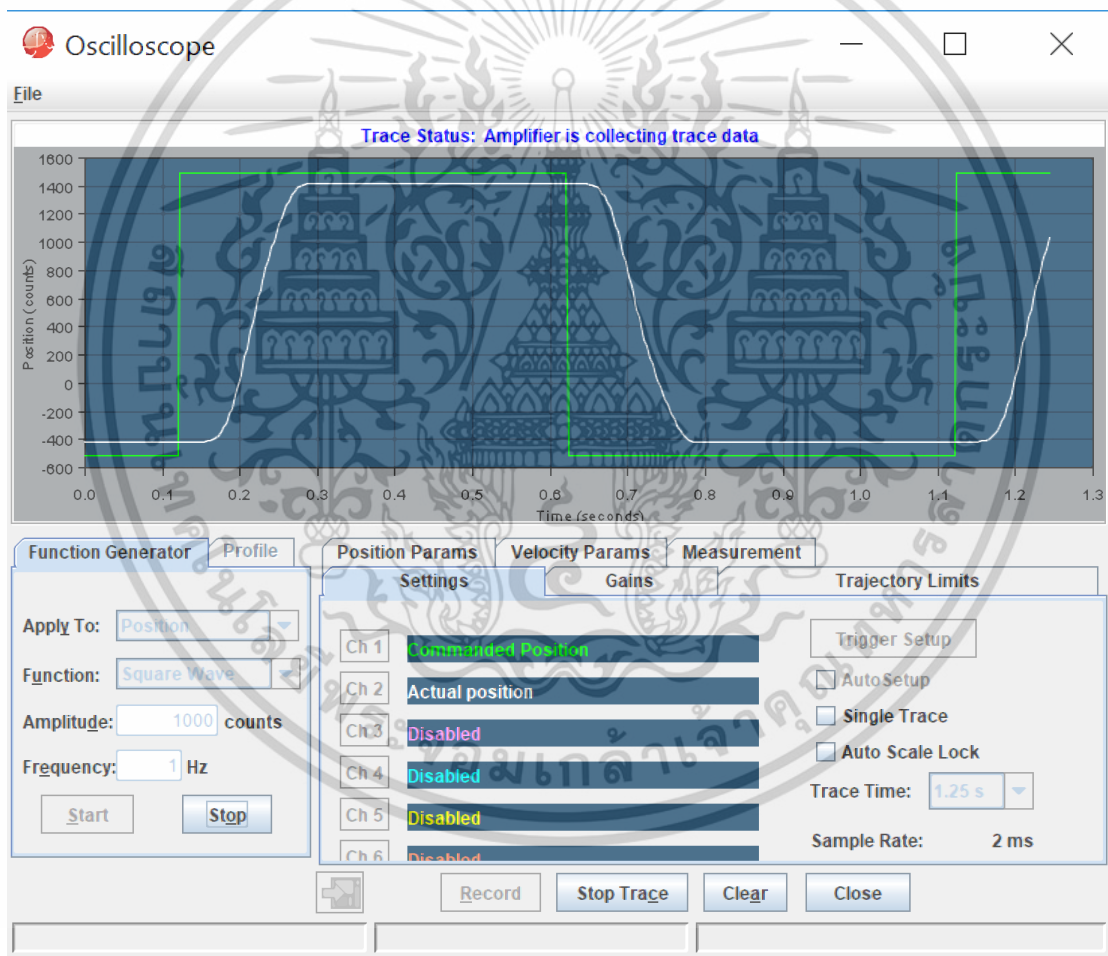
- เลือกเกน $P_p=6000$
 - เลือกเกน $A_{ff}=350$
 - เลือกเกน $V_{ff}=16384$
- และเมื่อสัญญาณที่ได้ใกล้เคียงกับสัญญาณ Command กดปุ่ม Close



รูปที่ 4.19 หน้าต่างการตั้งค่า P Loop



รูปที่ 4.20 หน้าต่างการตั้งปรับค่าเกรน PID Control



รูปที่ 4.21 หน้าต่างฟังก์ชัน Scope ของ Position Loop ด้วยการ Turning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

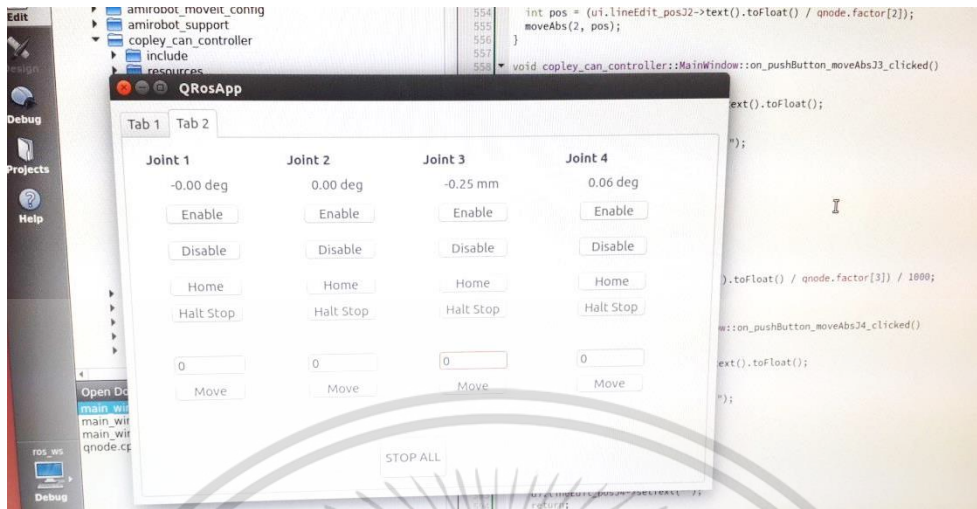
จากการออกแบบกลไกและรูปแบบของแขนกลอุตสาหกรรม และจำลองการทำงานของแขนกลภายในโรงงานอุตสาหกรรมรวมทั้งแขนกลอุตสาหกรรมต้นแบบที่ทำการออกแบบไว้ สร้างระบบควบคุมแขนกลและชุดซอฟต์แวร์ ได้นำไปสู่สร้างระบบควบคุมแขนกลและชุดซอฟต์แวร์ ทดลองระบบกลไกกับแขนกลต้นแบบแล้วนำระบบปฏิบัติการเชื่อมต่อกับแขนกล

4.2 การทดสอบการทำงานของหุ่นยนต์ SCARA robot

การพัฒนาหุ่นยนต์ด้วยเทคโนโลยีโอเพนซอร์สบนแพลตฟอร์ม ROS ซึ่งเปรียบเสมือนระบบปฏิบัติการสำหรับหุ่นยนต์ เป็นกรอบการทำงาน (framework) ที่ทำหน้าที่ควบคุมและติดต่อกับฮาร์ดแวร์ต่างๆ ของหุ่นยนต์ ให้ทำงานอย่างมีประสิทธิภาพและลดความยุ่งยากซับซ้อนในการพัฒนาหุ่นยนต์ การดำเนินการสร้างแพลตฟอร์มซอฟต์แวร์และฮาร์ดแวร์สำหรับเชื่อมต่อกับหุ่นยนต์และระบบอัตโนมัติที่ใช้งานในภาคอุตสาหกรรม โดยหุ่นยนต์ Plug-in Device for ROS bridge



รูปที่ 4.22 SCARA Robot ที่ผู้วิจัยได้พัฒนาขึ้นบนแพลตฟอร์ม ROS



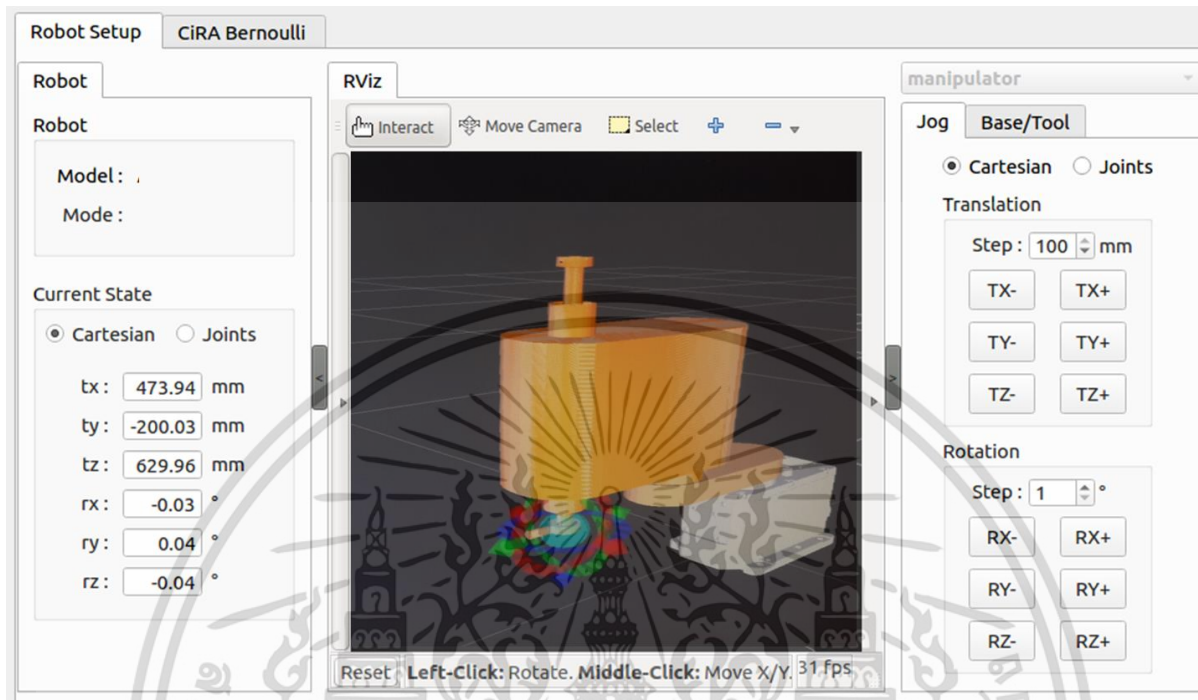
รูปที่ 4.23 การแสดงผลในแต่ละแกนของหุ่นยนต์ บนแพลตฟอร์ม ROS



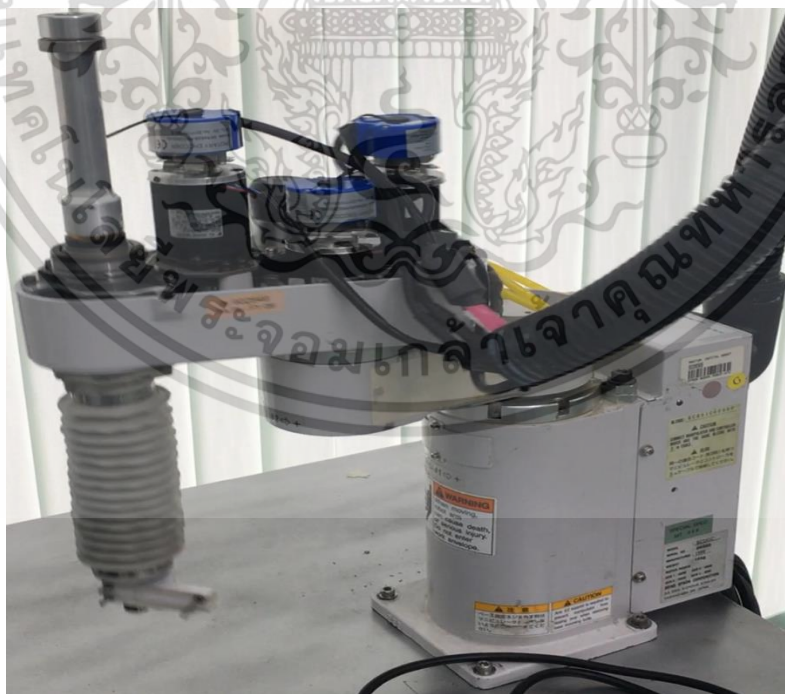
รูปที่ 4.24 การแสดงการจำลองการเคลื่อนที่ของหุ่นยนต์ บนแพลตฟอร์ม ROS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถควบคุมการทำงานของหุ่นยนต์ผ่าน RViz เพื่อสังเกตการณ์เคลื่อนที่ ดังแสดงในรูปที่ 4.25

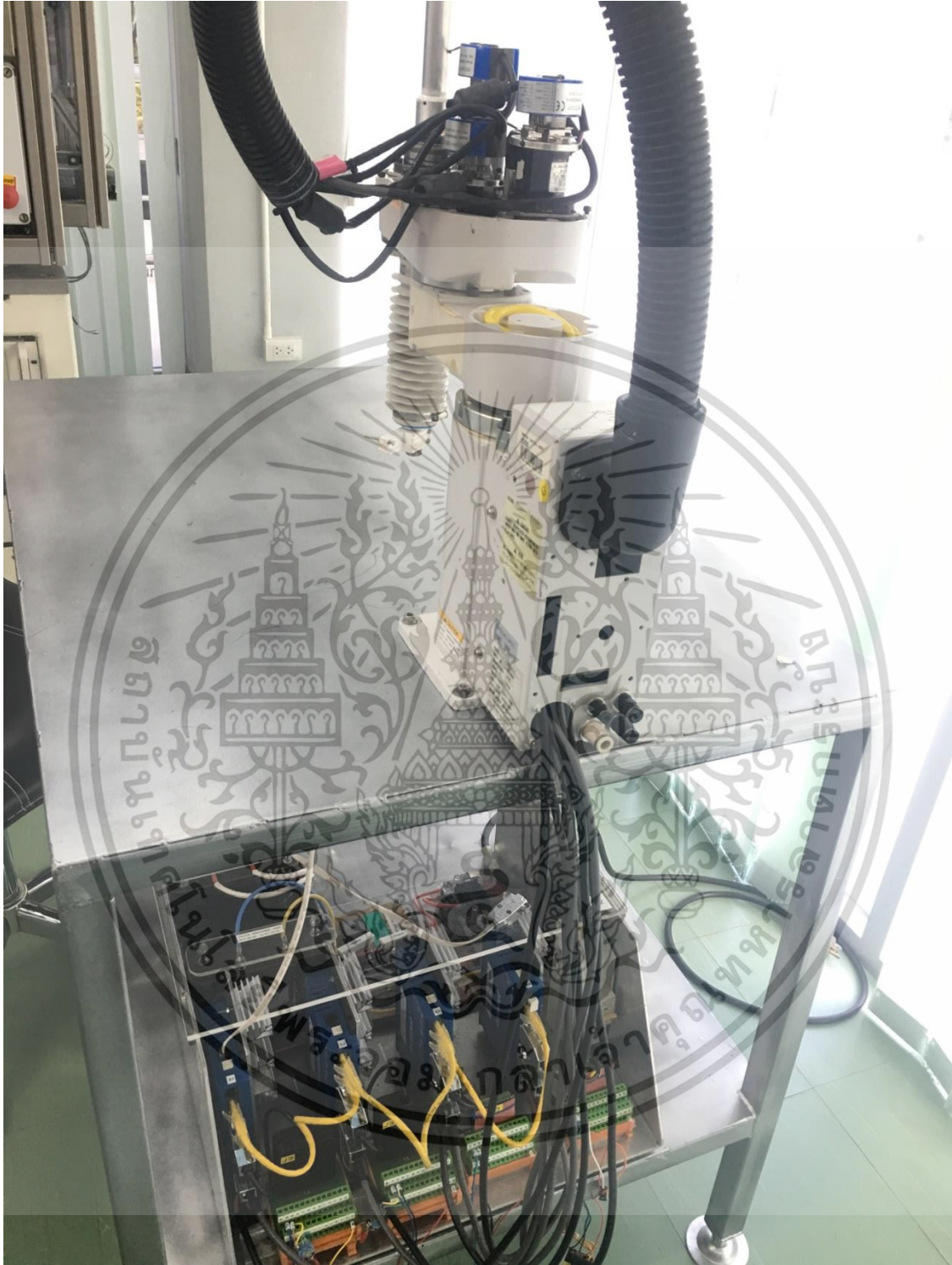


รูปที่ 4.25 การแสดงการควบคุมการทำงานของหุ่นยนต์ผ่าน RViz



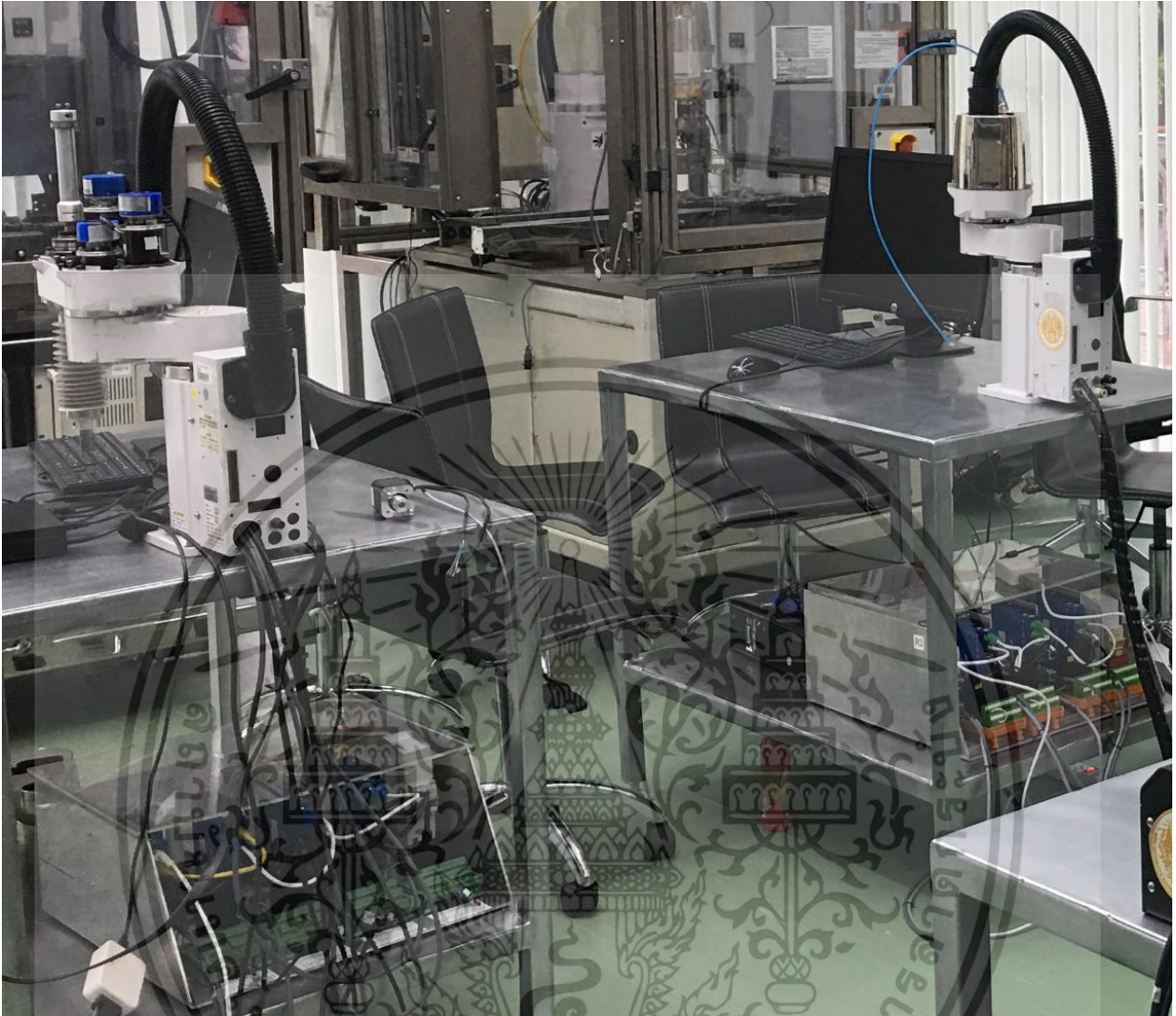
รูปที่ 4.26 การติดตั้งหุ่นยนต์ที่เสร็จเรียบร้อยแล้ว ที่ศูนย์หุ่นยนต์อุตสาหกรรมและระบบอัตโนมัติ (CiRA)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 การติดตั้งหุ่นยนต์และชุดควบคุม Servo motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 การติดตั้งหุ่นยนต์ทั้งหมด ที่ศูนย์หุ่นยนต์อุตสาหกรรมและระบบอัตโนมัติ (CiRA)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุป

5.1 บทสรุป

จากการพัฒนาแขนกลอุตสาหกรรมที่ได้สร้างขึ้นนี้ ก่อให้เกิดองค์ความรู้ที่เกี่ยวข้องกับการพัฒนาระบบควบคุมหุ่นยนต์โดยใช้ซอฟต์แวร์โอเพนซอร์สสำหรับแขนกลอุตสาหกรรม Robot Operating System (ROS) นอกจากนี้ยังถูกนำไปใช้เป็นแขนกลอุตสาหกรรมต้นแบบในการเรียนการสอนและปฏิบัติการในศูนย์หุ่นยนต์อุตสาหกรรมและระบบอัตโนมัติของวิทยาลัยนวัตกรรมการผลิตขั้นสูง ทั้งนี้ยังสามารถต่อยอดในการพัฒนาแขนกลอุตสาหกรรมที่นำไปใช้ในกระบวนการผลิตทางภาคอุตสาหกรรมได้และมีต้นทุนในการสร้างที่ถูกกว่าการจ้างบริษัทเอกชนเข้ามาดำเนินการ เนื่องจากระบบปฏิบัติการถูกพัฒนาขึ้น การอัปเดตซอฟต์แวร์ไม่มีค่าใช้จ่าย สามารถปรับเปลี่ยนรูปแบบลักษณะที่ต้องการให้หุ่นยนต์ทำงานได้อย่างง่ายดาย อีกทั้งยังสามารถเชื่อมต่อกับระบบการประมวลผลภาพ เช่น เซอร์และตัวเข้าเราต์ต่างๆ เพื่อสร้างระบบหุ่นยนต์ที่มีความสามารถควบคุม ตำแหน่ง ทิศทาง แรง รวมถึงการแยกแยะชิ้นงานด้วยระบบประมวลผลภาพ ที่ผ่านการออกแบบ สร้าง และทดสอบประสิทธิภาพระบบของหุ่นยนต์ โดยผู้วิจัยได้คำนึงถึงความสำคัญของระบบที่มีผลต่อการนำไปใช้งานในกระบวนการผลิตในภาคอุตสาหกรรม ให้มีความสามารถใช้งานได้อย่างมีความแม่นยำและจากการพัฒนาไปใช้งานอุปกรณ์รวมทั้งหมดหรือระบบอัตโนมัติที่ถูกสร้างโดย System Integrator (SI) ด้วยการใช้หุ่นยนต์อุตสาหกรรมที่ถูกพัฒนาขึ้น

5.2 ข้อเสนอแนะ

ระบบปฏิบัติการแขนกลอุตสาหกรรมที่มีในปัจจุบันยังต้องการการประยุกต์ใช้งานที่หลากหลาย โดยอาศัยเทคโนโลยีแขนกลเพื่อเพิ่มขีดความสามารถในการแข่งขันทางธุรกิจและเกิดประสิทธิภาพการคุมต้นทุนการผลิตที่ดีซึ่งระบบจำเป็นต้องมีความเที่ยงตรง แม่นยำ และรวดเร็วมากขึ้น จึงต้องมีการพัฒนาระบบควบคุมแขนกลให้มีความสามารถที่หลากหลายในการทำงาน ความยืดหยุ่น และการรวมระบบอื่นๆ เข้ามาด้วย เพื่อเพิ่มความสามารถในกระบวนการผลิต ยกตัวอย่างเช่น ระบบการแยกแยะชิ้นงาน (Classifications) ด้วยระบบประมวลผลภาพ เพื่อให้แขนกลทำงานได้อย่างถูกต้อง แม่นยำ ทั้งตำแหน่ง ทิศทาง แรง และมีความยืดหยุ่นในการทำงานใกล้เคียงกับมนุษย์ได้ (Collaborative robotics arms) ซึ่งระบบที่พัฒนาขึ้นนี้จะเกี่ยวข้องกับเทคโนโลยีการควบคุมการเคลื่อนที่ของแขนกลผ่านการตัดสินใจด้วยภาพ ด้วยระบบปัญญาประดิษฐ์ (Artificial Intelligence) ต่อไป

เอกสารอ้างอิง

1. Robert Krug et al., "The Next Step in Robot Commissioning: Autonomous Picking & Palletizing", 2016, IEEE ROBOTICS AND AUTOMATION LETTERS.
2. Lv Mei-yu et al., "The Dynamics Analysis on a Type Palletizing Robot", 2012, Applied Mechanics and Materials Vols. 157-158 (2012) pp 982-986.
3. Robert Krug et al., "Accuracy Analysis and Optimal Design of Palletizing Robot with Joint Clearance", 2017, Proceedings of the 2017 IEEE International Conference on Real-time Computing and Robotics, July 14-18,2017, Japan.
4. Dieter Faude, "Palletizing device and method for loading and/or unloading pallets." E.P. Patent 2,217,515 B1, issued Oct. 26, 2011.
5. Yoshiharu Nagatsuka, Masahiro Oyamada, "Robot programming device for palletizing operation by robot.", U.S. Patent 8,644,984 B2, issued Feb. 4, 2014.
6. Zmaj Petrovic, "Palletizer, palletizing method, and transfer of a layer of objects by palletizer from a conveyor to a layer depositing tool." U.S. Patent 9,908,721 B2, issued Mar. 6, 2018.
7. Liu, B.; Zhang, F.; Qu, X. A., *Method for Improving the Pose Accuracy of a Robot Manipulator Based on Multi-Sensor Combined Measurement and Data Fusion*, Sensors 2015, 15, 7933-7952.
8. Carroll, J., *3D vision system assists in robotic bin picking*, Vis. Syst. Des. 2014, 19.
9. The Pick-it robot vision for the ABB YuMi robot.,<https://www.pickit3d.com/>
10. Abbeloos, Wim, and Toon Goedemé., *Point Pair Feature based Object Detection for Random Bin Picking*, Computer and Robot Vision (CRV), 2016 13th Conference on. IEEE, 2016.
11. Yardley, J.V., et al., *Downward compatible AGV system and methods*, 1997, Google Patents.
12. Park, T.E., *AGV control system and method*, 2007, Google Patents.
13. Barry, D.T., Z. Fan, and A.L. Hardie, *Gesture enabled telepresence robot and system*, 2017, Google Patents.
14. Davey, N.S., et al., *Pharmacy automation using autonomous robot*, 2013, Google Patents.
15. McCabe, P.P. and T.E. Donahue, *Integration of an autonomous industrial vehicle into an asset management system*, 2017, Google Patents.

17. Steger, *Image gradient based shape prior for the segmentation of not that spherical structures*, 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI), 2012
18. Chris McCool, Vinod Chandran, Sridha Sridharan, *2D-3D Hybrid Face Recognition Based on PCA and Feature Modelling*, IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9), 2003
19. <http://www.pensamientoscomputables.com>
20. <http://wiki.ros.org/kinetic/Installation/Ubuntu>
21. <http://wiki.ros.org/ROS/Tutorials>



ข้อมูลประวัติผู้วิจัย

ประวัติส่วนตัว

ชื่อ-สกุลค.มกฤษ จักษุคำ.....

ตำแหน่งปัจจุบันอาจารย์.....

ประวัติการศึกษา

ชื่อย่อปริญญา	สาขา	สถาบันที่จบ	ปีที่จบ
อุตสาหกรรมศาสตรบัณฑิต (อ.ส.บ.)	เทคโนโลยีไฟฟ้าอุตสาหกรรม	มหาวิทยาลัยเทคโนโลยีพระจอม เกล้าพระนครเหนือ	2545
วิศวกรรมศาสตร มหาบัณฑิต (วศ.ม.)	วิศวกรรมไฟฟ้า	มหาวิทยาลัยธรรมศาสตร์	2550
ปรัชญาดุษฎีบัณฑิต (ปร.ด.)	วิศวกรรมศาสตร์/ วิศวกรรมไฟฟ้า	มหาวิทยาลัยธรรมศาสตร์	2558

สาขาวิจัยที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา)..... Wireless sensor networks, Multi-hop
networks, Embedded system, Ultrasound measurement application, Robotics and Automation
System.....

ผลงานวิจัย/งานสร้างสรรค์

ผลงานวิจัย/งานสร้างสรรค์ที่ตีพิมพ์เผยแพร่ (ระดับชาติและนานาชาติ).....

ผลงานที่ตีพิมพ์ (วารสารวิชาการ)

[1] Komgrit Jaksukam and Supachai Vorapojpisu, "A Study on Data Collection Behaviors for a Class of Multi-hop Wireless Sensor Networks Based on Probability Model" International Review on Computers and Software (I.RE.CO.S.), Vol. 10 No.6, 2015.

[2] Komgrit Jaksukam and Sumet Umchid, "A PRIMARY LEVEL ULTRASONIC POWER MEASUREMENT SYSTEM DEVELOPED AT NATIONAL INSTITUTE OF METROLOGY" with THAILAND INTERNATIONAL JOURNAL OF APPLIED BIOMEDICAL ENGINEERING VOL.4, NO.1 2011

[3] ค.มกฤษ จักษุคำ, อริญชยา พันธุ์พำนัก, วิรัตน์ ปลั่งแสงมาศ, "การตรวจสอบคุณสมบัติสนามคลื่นของอัลตราซาวด์ทรานสดิวเซอร์ในย่านความถี่ 1MHz -15MHz" ในวารสารวิชาการ Metrology Info ฉบับประจำเดือน ตุลาคม - ธันวาคม 2555

งานประชุมทางวิชาการในประเทศและต่างประเทศ

- [1] T. Tongloy ; S. Chuwongin ; K. Jaksukam ; C Chousangsunton ; S. Boonsang, “Asynchronous deep reinforcement learning for the mobile robot navigation with supervised auxiliary tasks”, 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Shanghai, China, 29-31 Dec. 2017.
- [2] **Komgrit Jaksukam** and Supachai Vorapojpisut “A Probability Model of a Class of Multi-hop WSNs” International Conference on Embedded Systems and Intelligent Technology (ICESIT), Gwangju, KOREA, September 18th – 20th, 2014.
- [3] **คมกฤษ จักขุคำ**, อริญา พันธ์พานัก , พัชริญา เพชรผ่อง, “Investigation of Field Characteristics for High Intensity Focused Ultrasound (HIFU) with Fibre-Optic Hydrophone System” การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 37 (EECON-37) 19 – 21 พฤศจิกายน 2557 มหาวิทยาลัยขอนแก่น.
- [4] **Komgrit Jaksukam** and Sumet Umchid, “Development of Ultrasonic Power Measurement Standards in Thailand” The 10th International Conference on Electronic Measurement & Instruments (IEEE ICEMI’2011) ณ เมือง ฉิงตู ประเทศจีน ในระหว่างวันที่ 16-18 สิงหาคม 2554
- [5] **คมกฤษ จักขุคำ**, อริญา พันธ์พานัก, วิรัตน์ ปลั่งแสงมาศ, สุเมธ อ่ำชาติ, “การวัดคุณสมบัติของสนามคลื่นอัลตราซาวด์จากอัลตราซาวด์ทรานสดิวเซอร์” ในงานประชุมวิชาการวิศวกรรมชีวการแพทย์ ครั้งที่ 3 ณ โรงแรมเอวันรอยัลครุยส์ พัทยา ชลบุรี ในระหว่างวันที่ 4-5 สิงหาคม 2554
- [6] **Komgrit Jaksukam**, Virat Plangsangmas and Sumet Umchid, “Ultrasound DEVELOPMENT OF A PRIMARY LEVEL ULTRASONIC POWER MEASUREMENT SYSTEM AT NIMT” The 3th Biomedical Engineering International Conference (BMEiCon2010) August 27-28, 2010, Kyoto, Japan, 2010.
- [7] ขนิษฐา สวัสดิ์, อรวรรณ หริสดี, **คมกฤษ จักขุคำ**, วิรัตน์ ปลั่งแสงมาศ และ สุเมธ อ่ำชาติ, “การสอบเทียบเครื่องมือวัดกำลังอัลตราซาวด์ที่ใช้ในทางการแพทย์” การประชุมวิชาการวิศวกรรมชีวการแพทย์ไทย ครั้งที่ 2 ในวันที่ 5-6 สิงหาคม 2553 ณ โรงแรมเดอะทวินทาวเวอร์ กรุงเทพมหานคร
- [8] Sumet Umchid and **Komgrit Jaksukam**, “Development of the Primary Level Ultrasound Power Measurement System in Thailand” The 4th International Symposium on Biomedical Engineering (IEEE ISBME 2009), Bangkok, Thailand 2009.
- [9] Sumet Umchid, **Komgrit Jaksukam**, Kornpatsitt Promasa, Virat Plangsangmas , “Ultrasound Power Measurement System Using Radiation Force Balance Technique” The 2th Biomedical Engineering International Conference(BMEiCon2009) August 13-14, 2009, Phuket, Thailand, 2009.
- [10] W.Punurai and **K.Jaksukam**, “The development of sensor network for structural health monitoring using CANopen protocol” East Asia Pacific Conference on Structural Engineering and Construction (EASEC-11) , National Taiwan University Taipei , Taiwan 2008.
- [11] **k. jaksukam** and s. vorapojpisut., “The Development of sensor network for Structural Testing” The 12th National Convention on Civil Engineering, Thailand, Phitsanulok, May 2 – 4, 2007
- [12] **k. jaksukam** and s. vorapojpisut , “A Study on Real-Time Performance of CANopen – Structural testing system” National Computer Science and Engineering Conference NCSEC2006, Khon Kaen, Thailand, October 25-27, 2006