



ใบรับรองวิทยานิพนธ์
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

ปริญญา

วิศวกรรมไฟฟ้า

วิศวกรรมไฟฟ้า

สาขา

ภาควิชา

เรื่อง การชดเชยแรงเสียดทานป้อนไปหน้าเพื่อเพิ่มประสิทธิภาพการทำงานของเครื่องซีเอ็นซี

Friction Feedforward Compensation For Efficiency Improvement In CNC Machines

นามผู้วิจัย นายพีพงศ์ รัตนวรหิรัญกุล

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์พีระยศ แสนโกชณ์, D.Sc.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(อาจารย์สมบัติ รุจามุรักษ์, Ph.D.)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(อาจารย์ยศวีร์ วีระกำแหง, Ph.D.)

หัวหน้าภาควิชา

(รองศาสตราจารย์มงคล รักษาพัชรวงค์, Ph.D.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

(รองศาสตราจารย์กัญจนา ชีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ เดือน พ.ศ.

สิงสิงห์ มทววิทยาลัยเกษตรศาสตร์

วิทยานิพนธ์

เรื่อง

การชดเชยแรงเสียดทานป้อนไปหน้าเพื่อเพิ่มประสิทธิภาพการทำงานของเครื่องซีเอ็นซี

Friction Feedforward Compensation For Efficiency Improvement In CNC Machines

โดย

นายรพีพงศ์ รัตนวารีญกุล

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

พ.ศ. 2553

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

รพีพงศ์ รัตนวรหิรัญกุล 2553: การชดเชยแรงเสียดทานป้อนไปหน้าเพื่อเพิ่ม
ประสิทธิภาพการทำงานของเครื่องซีเอ็นซี ปรินญาวิศวกรรมศาสตรมหาบัณฑิต
(วิศวกรรมไฟฟ้า) สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า อาจารย์ที่ปรึกษา
วิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์พีระยศ แสนโกชณ์, D.Sc. 71 หน้า

ในการเคลื่อนที่ของเครื่องจักรซีเอ็นซีพบว่า มีแรงเสียดทานเกิดขึ้นในระหว่างการ
เคลื่อนที่เนื่องจาก Ball Screw ราง และที่ Bearing ทำให้การเคลื่อนที่ของเครื่องจักรซีเอ็นซีมี
ความคลาดเคลื่อนจากตำแหน่งที่ต้องการ ในงานวิจัยนี้จึงมีจุดประสงค์หลักคือ การชดเชยแรง
เสียดทานที่เกิดขึ้นกับระบบ เพื่อให้เครื่องจักรซีเอ็นซีสามารถเคลื่อนที่ได้อย่างแม่นยำและมี
ประสิทธิภาพมากขึ้น โดยงานวิจัยนี้ได้ประยุกต์การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัว
ควบคุมพีไอดี เปรียบเทียบกับการควบคุมแบบพีไอดีทั่วไป

ในงานวิจัยนี้ได้แบ่งการทดสอบออกเป็นสองส่วนคือ การจำลองการเคลื่อนที่โดย
โปรแกรม Simulink และการทดสอบการเคลื่อนที่ของเครื่องซีเอ็นซีจริงโดยโปรแกรม LabVIEW
ผ่าน Field Point จากผลการทดสอบทั้งสองส่วนพบว่า การประยุกต์ใช้การชดเชยแรงเสียดทาน
ป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี สามารถเคลื่อนที่ได้แม่นยำและมีประสิทธิภาพมากกว่าการ
ควบคุมการเคลื่อนที่แบบพีไอดีทั่วไป

ลายมือชื่อนิสิต

ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Rapeepong Rattanawaorahirunkul 2010: Friction Feedforward Compensation For Efficiency Improvement In CNC Machines. Master of Engineering (Electrical Engineering), Major Field: Electrical Engineering, Department of Electrical Engineering. Thesis Advisor: Assistant Professor Peerayot Sanposh, D.Sc. 71 pages.

In the motion of the CNC machines, the friction happens from ball screws, rails, and bearing. This friction effect causes the position error on the CNC milling machines. In this research, the main purpose is to compensate the friction in the system for increasing accuracy and efficiency. This research applies Friction Feedforward technique with PID controller to the system.

The simulations are performed on Simulink and the experiments are performed on Field-Point controller with Laviw. These results indicate that the Friction Feedforward technique with PID controller has more accuracy and more efficiency than the general PID controller.

Student's signature

Thesis Advisor's signature

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.พีระยศ แสนโกชณ์ ประธานกรรมการที่ปรึกษา ที่ได้ให้คำปรึกษาแนะนำและตรวจแก้ไขข้อบกพร่องต่าง ๆ ที่เป็นประโยชน์ที่สามารถนำไปประยุกต์ใช้ในงานวิจัยและนำไปแก้ปัญหาที่ประสบขณะทำการวิจัย พร้อมทั้งให้ความสนับสนุนช่วยเหลือในด้านอุปกรณ์และเครื่องมือสำหรับการทำวิจัยในครั้งนี้ให้สำเร็จลุล่วงไปด้วยดี

ขอกราบขอบพระคุณ อ.ดร.ยศวีร์ วีระกำแหง และ อ.ดร.มิตี รุจานุรักษ์ กรรมการที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้คำปรึกษาแนะนำและตรวจแก้ไขข้อบกพร่องต่าง ๆ ที่เป็นประโยชน์ และสามารถนำไปประยุกต์ใช้ในงานวิจัยเพื่อให้งานวิจัยสามารถดำเนินการไปได้ด้วยดี

ขอกราบขอบพระคุณ ดร.กนกเวทย์ ตั้งพิมพ์รัตน์ ที่ได้ให้คำปรึกษาและสนับสนุนช่วยเหลือในการหาทุนในการทำวิจัย พร้อมทั้งแนะนำและช่วยแก้ไขข้อบกพร่องต่าง ๆ ในงานวิจัยที่เป็นประโยชน์อย่างยิ่ง

ขอขอบพระคุณ สถาบันบัณฑิตวิทยาศาสตร์และเทคโนโลยีไทย (TGIST) สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ที่ให้ทุนสนับสนุนงานการศึกษาและการทำวิจัยนี้ โดยได้รับทุนสนับสนุนจากสัญญารับทุนเลขที่ TGIST 01-51-074

ขอกราบขอบพระคุณ ครูอาจารย์ทุกท่านที่ให้การอบรมสั่งสอนและถ่ายทอดวิชาความรู้ตลอดมา จนประสบความสำเร็จในชีวิตและการเรียนในวันนี้

ขอขอบคุณพี่ต๋อง ฐิติวัชร เทอดบ้านเกิด และเพื่อนๆ พี่ๆ ในห้องปฏิบัติการควบคุมทุกคนที่ช่วยในการแก้ปัญหาในระหว่างการทำวิจัยหลายครั้ง

ท้ายที่สุด ขอกราบขอบพระคุณ บิดา มารดา อย่างสูง ที่คอยให้การสนับสนุนและเป็นที่กำลังใจขณะที่กำลังศึกษาในระดับปริญญาโทจนประสบผลสำเร็จในการเล่าเรียน

รพีพงศ์ รัตนวรหิรัญกุล

เมษายน 2553

สารบัญ

หน้า

สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(3)
คำอธิบายสัญลักษณ์และคำย่อ	(6)
คำนำ	1
วัตถุประสงค์	2
การตรวจเอกสาร	3
อุปกรณ์และวิธีการ	18
อุปกรณ์	18
วิธีการ	18
ผลและวิจารณ์	35
ผล	35
วิจารณ์	61
สรุปและข้อเสนอแนะ	63
สรุป	63
ข้อเสนอแนะ	64
เอกสารและสิ่งอ้างอิง	65
ภาคผนวก	68
ประวัติการศึกษา และการทำงาน	71

สารบัญตาราง

ตารางที่		หน้า
1	ฟังก์ชัน G-Code พื้นฐาน	14
2	อุปกรณ์ในการสร้างวงจรแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์	20
3	การเลือกค่า Parameter ใน Drive	22
4	การเลือกค่า Parameter ใน Drive เพื่อหาแรงเสียดทาน	30
5	ค่า G-Code ที่ป้อนให้ GUI สำหรับรูปประยุกต์อื่นๆ	45
6	ค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error) ของเครื่องซีเอ็นซี 3 แกนด้วย Simulink	49
7	ค่าความผิดพลาดสัมบูรณ์สูงสุด (Maximum Absolute Error)	49
8	ค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error) ของการทดสอบโดยโปรแกรม LabVIEW ผ่าน Fieldpoint	60
9	ค่าความผิดพลาดสัมบูรณ์สูงสุดของการทดสอบโดยโปรแกรม LabVIEW ผ่าน Fieldpoint	61

สารบัญภาพ

ภาพที่		หน้า
1	การส่งค่า Velocity Profile ในแต่ละช่วง	6
2	การเคลื่อนที่ในแนวเส้นโค้ง	7
3	แบบจำลองแรงเสียดทาน	9
4	การเคลื่อนที่ของมอเตอร์	11
5	แรงเสียดทานคูลอมบ์และแรงเสียดทานหนืด	12
6	การเคลื่อนที่ตามแบบจำลองของ Armstrong	12
7	การควบคุมมอเตอร์แบบวงปิด	13
8	Block Diagram การควบคุมทั้งระบบ	14
9	การชดเชยแรงเสียดทานป้อนไปหน้า	14
10	การทำงานโดยรวม	18
11	วงจรสำหรับการแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์	19
12	วงจรสำหรับการแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์	20
13	วงจรแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์	20
14	AC Servo Motor และ AC Servo Motor Drive	21
15	แสดงการหาแบบจำลองโดยรวม	22
16	Simulink แสดงการหาโมเดลของระบบ	22
17	การเก็บข้อมูลเพื่อหาแบบจำลองของระบบ	23
18	แสดงสัญญาณ Input U	24
19	แสดงสัญญาณ Output Y	24
20	แสดงหน้าต่างการนำข้อมูลมาหาโมเดลของระบบ	25
21	แสดงการ Import Data	25
22	แสดงการประมาณค่าเพื่อหาโมเดลของระบบ	26
23	แสดงการเลือก Order ของ ARX Model	26
24	แสดง ARX Model ที่ได้หลังจากเลือก Estimate	27
25	สัญญาณ Output ของระบบจริง	27

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
26	สัญญาณ Output ของโมเดล ARX111 เทียบกับสัญญาณจริงแสดงการหาแสดง	28
27	Output Response ของโมเดล ARX111	28
28	แสดง Bode Diagram ของโมเดล ARX111 แสดง Velocity Profile การเคลื่อนที่	28
29	แสดงการเลือกโมเดลที่สนใจมาที่ตำแหน่ง To Workspace	29
30	แสดงการใช้ Simulink เก็บข้อมูลเพื่อหาแบบจำลองของระบบ	31
31	กราฟความสัมพันธ์ระหว่างความเร็วกับเวลาของมอเตอร์และแกน	32
32	แสดงการหาแบบจำลองทางคณิตศาสตร์จากข้อมูลจริง	33
33	GUI รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	36
34	Velocity Profile การเคลื่อนที่รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	36
35	โปรแกรม Simulink การควบคุมมอเตอร์ในแต่ละวิธี	37
36	Velocity Command เทียบกับสัญญาณ Output มอเตอร์แกน X	38
37	เปรียบเทียบระยะการเคลื่อนที่ในแนวแกน X	38
38	Velocity Command กับสัญญาณ Output มอเตอร์แกน Y	38
39	เปรียบเทียบระยะการเคลื่อนที่ในแนวแกน Y	39
40	GUI รูปเส้นตรงแกน X -5 ซม. แกน Y -10 ซม.	39
41	Velocity Profile การเคลื่อนที่แกน X -5 ซม. แกน Y -10 ซม.	40
42	Velocity Command กับสัญญาณ Output มอเตอร์แกน X	40
43	เปรียบเทียบระยะการเคลื่อนที่ในแนวแกน X	41
44	Velocity Command กับสัญญาณ Output มอเตอร์แกน Y	41
45	เปรียบเทียบระยะการเคลื่อนที่ในแนวแกน Y	42
46	GUI รูปวงกลมรัศมี 5.6569 ซม.	42
47	Velocity Profile การเคลื่อนที่รูปวงกลมรัศมี 5.6569 ซม.	43
48	เปรียบเทียบ Velocity Command กับสัญญาณ Output มอเตอร์แกน X	43
49	Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน	44

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
50	เปรียบเทียบเส้นทางการเคลื่อนที่เป็นวงกลมของมอเตอร์ทั้งสองแกน	44
51	GUI แสดงรูปประยุคต์อื่นๆ	46
52	Velocity Profile การเคลื่อนที่รูปประยุคต์อื่นๆ	46
53	Velocity Command กับสัญญาณ Output มอเตอร์แกน X	47
54	Velocity Command กับสัญญาณ Output มอเตอร์แกน Y	47
55	เส้นทางการเคลื่อนที่รูปประยุคต์ของมอเตอร์ทั้งสองแกน	48
56	Field Point ขณะทำการทดสอบ	50
57	การต่อวงจรแปลงเป็น +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์	50
58	Power Supply และ Drive ขณะทำการทดสอบ	51
59	เครื่องซีเอ็นซี 3 แกนที่ใช้ในการทดสอบ	51
60	GUI รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	52
61	Velocity Profile การเคลื่อนที่รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	52
62	Block Diagram นำข้อมูลเข้ามาทำการควบคุมแบบพีไอดีแกน X	53
63	Block Diagram การควบคุมแบบพีไอดีแกน X	53
64	Front Panel การควบคุมมอเตอร์แกน X แบบพีไอดี	54
65	การนำข้อมูลเข้ามาควบคุมในแนวแกน X	54
66	Block diagram การชดเชยแรงเสียดทานป้อนไปหน้ากับตัวควบคุมพีไอดีแกน X	55
67	Front Panel การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีแกน X	55
68	Block Diagram นำข้อมูลเข้ามาทำการควบคุมแบบพีไอดีแกน Y	56
69	Block Diagram การควบคุมมอเตอร์แกน Y แบบพีไอดี	56
70	Front Panel การควบคุมมอเตอร์แกน Y แบบพีไอดี	57
71	การนำข้อมูลมาควบคุมในแนวแกน Y	58
72	Block Diagram การชดเชยแรงเสียดทานป้อนไปหน้ากับตัวควบคุมพีไอดีแกน Y	58
73	Front Panel การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีแกน Y	59
74	แสดงตัวอย่างไฟล์ .EXE	69

คำอธิบายสัญลักษณ์และคำย่อ

F_c	=	แรงเสียดทานคูลอมบ์ (N)
μ	=	สัมประสิทธิ์แรงเสียดทาน
F_N	=	แรงปฏิกิริยาที่กระทำต่อวัตถุ (N)
F_v	=	แรงเสียดทานหนืด (N)
v	=	ความเร็ว (m/s)
F_e	=	แรงภายนอก (N)
F_s	=	ค่าสูงสุดของแรงเสียดทานสถิตย์ (N)
GUI	=	Graphical User Interfaces
$V_a(s)$	=	ความเร็วเอาต์พุต
$V_s(s)$	=	ความเร็วอินพุต
$C_v(s)$	=	Controller
$G_m(s)$	=	ฟังก์ชันถ่ายโอนของมอเตอร์
$G_{plant}(s)$	=	ฟังก์ชันถ่ายโอนของระบบ
F_p	=	Friction Plant
K_p	=	อัตราขยายพี
K_i	=	อัตราขยายไอ
K_d	=	อัตราขยายดี
MAE	=	ค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error)
MaxAE	=	ค่าความผิดพลาดสัมบูรณ์สูงสุด (Maximum Absolute Error)
$f(k)$	=	อัตราป้อนที่ k ของมอเตอร์ (mm/sec)
$T(k)$	=	ช่วงเวลาคาบที่ k
f_0	=	อัตราป้อนเริ่มต้นของมอเตอร์
f_{max}	=	อัตราป้อนสูงสุดของมอเตอร์

การชดเชยแรงเสียดทานป้อนไปหน้าเพื่อเพิ่มประสิทธิภาพการทำงานของเครื่องซีเอ็นซี

Friction Feedforward Compensation For Efficiency Improvement In CNC Machines

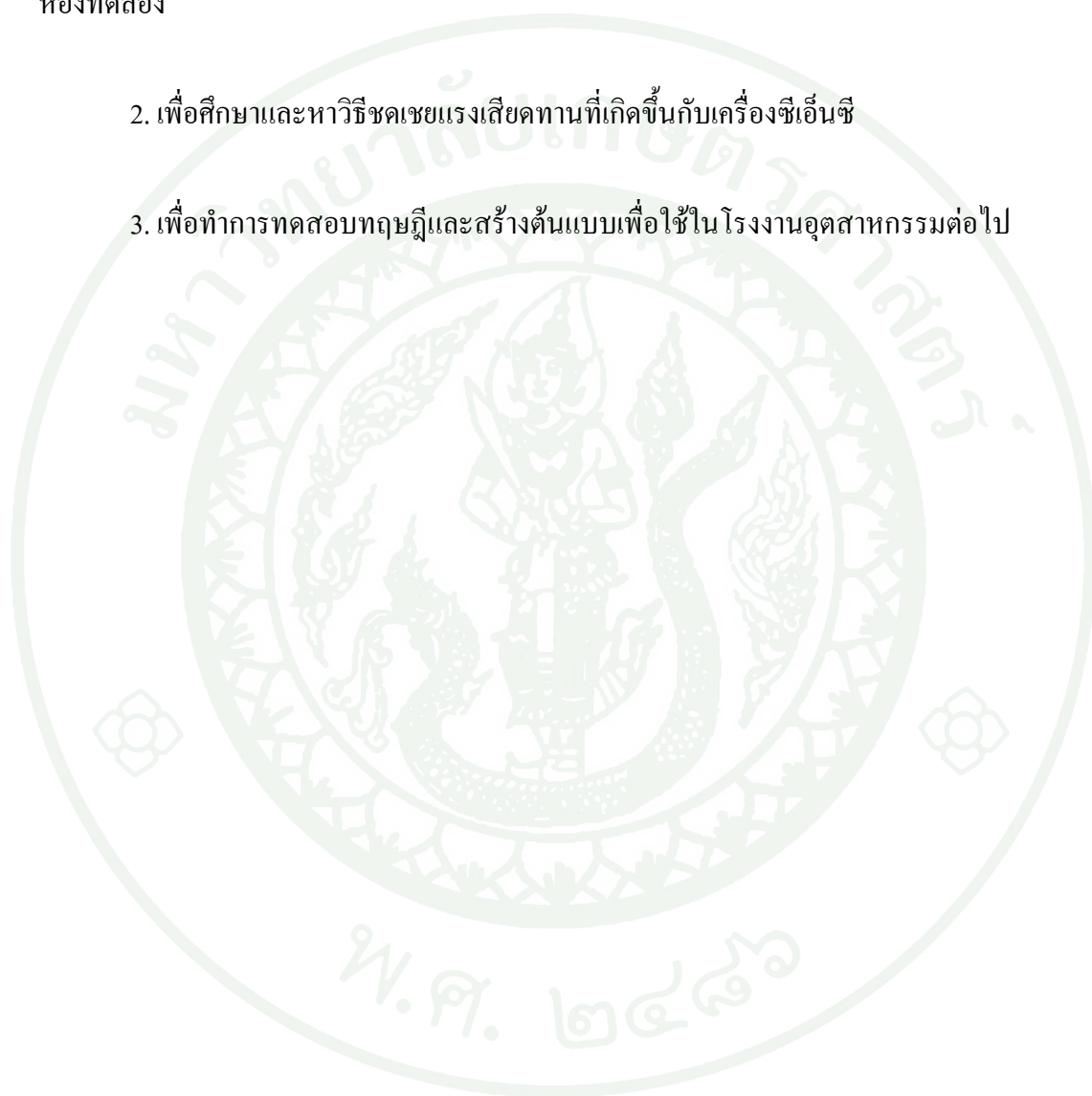
คำนำ

ในปัจจุบันนี้ชุดควบคุมแบบดิจิทัลได้ถูกนำไปใช้ในเครื่องจักรอุตสาหกรรมสมัยใหม่อย่างแพร่หลาย อาทิเช่น แขนหุ่นยนต์ประเภทเครื่องจักรซีเอ็นซี ซึ่งเป็นอุปกรณ์ที่มีการพัฒนาไปอย่างรวดเร็วเนื่องจากความก้าวหน้าของนวัตกรรมทางคอมพิวเตอร์และอิเล็กทรอนิกส์ จึงทำให้ชุดควบคุมที่นำมาใช้ในเครื่องจักรวันนี้อาจมีความล้าสมัยเมื่อเวลาผ่านไปไม่นาน ดังนั้นจึงส่งผลให้ต้องลงทุนเปลี่ยนชุดควบคุมเครื่องจักรหรืออาจต้องซื้อเครื่องจักรชุดใหม่ จากความต้องการชุดควบคุมสมัยใหม่ที่มีคุณภาพและราคาถูก เป็นผลให้มีการพัฒนาชุดควบคุมขึ้นมา แต่อย่างไรก็ตามชุดควบคุมที่ได้พัฒนานั้นก็ยังไม่มีประสิทธิภาพและถูกออกแบบให้มีการใช้งานที่จำกัดและใช้ได้เฉพาะงานมากจนเกินไป ดังนั้นเครื่องจักรในปัจจุบันจึงต้องการชุดควบคุมที่สามารถสั่งงานได้โดยง่าย และมีความยืดหยุ่นสามารถพัฒนาต่อไปได้โดยไม่ยาก วิทยานิพนธ์นี้จึงได้ทำการควบคุมเครื่องซีเอ็นซี โดยรับคำสั่งจากผู้ใช้งานทางคอมพิวเตอร์ส่วนบุคคล

ในการทำงานจริงของเครื่องซีเอ็นซี ขั้นตอนการออกแบบเส้นทางเดินถือเป็นส่วนสำคัญในการส่งค่าต่างๆในช่วงเวลาที่เหมาะสม เพื่อให้เครื่องจักรทำงานได้อย่างถูกต้องและมีประสิทธิภาพ นอกจากนั้นแรงเสียดทานที่เกิดขึ้นกับระบบมีความสำคัญมากต่อการปรับปรุงประสิทธิภาพการทำงานของเครื่องจักรซึ่งอาจทำให้ระบบขาดเสถียรภาพได้ จากปัญหาข้างต้นงานวิจัยชิ้นนี้จึงได้ทำการออกแบบเส้นทางเดิน และศึกษาผลกระทบของแรงเสียดทานที่เกิดขึ้นโดยนำวิธีการชดเชยป้อนไปหน้ามาประยุกต์ใช้ในการแก้ปัญหาแรงเสียดทานที่เกิดขึ้น ในการทดสอบจะทำการรับรหัสจี (G-Code) ผ่านทาง GUI (Graphical User Interfaces) จากนั้นโปรแกรมจะทำการคำนวณและส่งค่าให้กับ Simulink และ LabVIEW เพื่อทำการควบคุมมอเตอร์ต่อไป

วัตถุประสงค์

1. เพื่อศึกษาทฤษฎีและหาแบบจำลองแรงเสียดทานที่เกิดขึ้นกับเครื่องจักรซีเอ็นซีที่มีอยู่ในห้องทดลอง
2. เพื่อศึกษาและหาวิธีลดแรงเสียดทานที่เกิดขึ้นกับเครื่องซีเอ็นซี
3. เพื่อทำการทดสอบทฤษฎีและสร้างต้นแบบเพื่อใช้ในโรงงานอุตสาหกรรมต่อไป



การตรวจเอกสาร

จากการค้นคว้าข้อมูลการควบคุมมอเตอร์แบบ 2 แกนและ 3 แกนด้วยวิธีต่างๆ พบว่า (Simal *et al.*, 1955) ได้เขียนหนังสือเกี่ยวกับทฤษฎีควบคุมแกนหุ่นยนต์ โดยกล่าวถึงการหาสมการแกนหุ่นยนต์เบื้องต้น และการควบคุมแกนหุ่นยนต์ด้วยวิธี Close Loop Control อีกทั้งยังกล่าวถึงข้อควรระวังในการใช้กับแกนหุ่นยนต์จริง ส่วน (Lewis *et al.*, 1990) ได้พิสูจน์ Trajectory Generation สำหรับแกนหุ่นยนต์ 2 แกน โดยได้แสดงวิธีการหา Jacobians ของแกนหุ่นยนต์ 2 แกน ด้วยวิธีการหา Minimum Norm

จากนั้น (Iwazawa *et al.*, 1994) ได้เสนอการควบคุมมอเตอร์ด้วยวิธี Multi-Axis Coordinated Control (MACC) เทียบกับวิธีประมาณค่าทั่วไป ซึ่งจะเห็นว่าผลที่ได้มีค่าความผิดพลาดน้อยกว่าและความเที่ยงตรงมากกว่า ต่อมา (Yamazaki *et al.*, 1998) ได้เสนอวิธีใหม่ซึ่งควบคุมมอเตอร์โดยใช้ Dynamic Gain Control วิธีนี้จะคำนวณความผิดพลาดของการเคลื่อนที่แต่ละแกน แล้วนำมาปรับค่า Dynamic Gain Control เพื่อให้เหมาะสมกับการควบคุมมอเตอร์ ตามรายงานบอกว่าให้ผลดีกว่าวิธีที่มีอยู่ก่อนหน้านี้ ต่อมา (Gon *et al.*, 1999) ได้เสนอวิธีการใช้ Neural Network ในการปรับค่า PID Gain สำหรับการควบคุมมอเตอร์ 2 แกน

ต่อมา (Canudas *et al.*, 1995) ได้นำเสนอวิธีการหา Dynamic Model โดยพิจารณาการหา Stribeck Effect ร่วมกับแรงเสียดทานในช่วงอื่นๆ โดยขยายแบบจำลองแรงเสียดทานจาก Dahl Model ต่อมา (Jason *et al.*, 1996) ได้นำเสนอวิธีการใช้ Novel Fuzzy มาชดเชยแรงเสียดทานที่เกิดขึ้นในการควบคุมมอเตอร์กระแสดตรง จากนั้น (Rafael *et al.*, 1999) ได้เสนอการหาแบบจำลองแรงเสียดทานที่เกิดขึ้นกับมอเตอร์ โดยการป้อน Torque Ramp ให้กับมอเตอร์ แล้วนำค่าความเร็วมารวมเทียบกับเวลา เพื่อหาค่าพารามิเตอร์ต่างๆ ในสมการแรงเสียดทาน จากนั้น (Shuxun *et al.*, 1999) ได้เสนอการชดเชยแรงเสียดทานที่เกิดขึ้นกับระบบโดยวิธีการสร้างแบบจำลองและควบคุมโดย Reference Adaptive Control ต่อมา (Erkorkmaz *et al.*, 2001) ได้กล่าวถึงการออกแบบเส้นทางเดินแบบ Jerk Limitation และ Quintic Spline Interpolation อีกทั้งยังได้แสดงการหาแบบจำลองของระบบ และแสดงการออกแบบเส้นทางเดินมาใช้กับระบบจริง ซึ่งให้ผลใกล้เคียงกับตำแหน่งที่ต้องการมาก จากนั้น (Kiong *et al.*, 2001) ได้นำเสนอวิธีการหาแบบจำลองแรงเสียดทานและการชดเชยแรงเสียดทานด้วยวิธี Relay Feedback

ต่อมา (Shiuh Yeh *et al.*, 2003) ได้เสนอการควบคุมมอเตอร์ 2 แกนด้วยการปรับปรุงเส้นทางการเดินของรอกให้ดีขึ้น และได้ออกแบบ Contouring Error Transfer Function (CETF) ในชุดควบคุม ต่อมา Frew and Rock ได้ใช้กลไกการติดตามในการติดตามการเคลื่อนที่ของมอเตอร์ 2 แกน โดยใช้วิธี Pyramid, Breadth-First Search Algorithm ในการสร้างเส้นทางแบบ Real Time เพื่อหา Minimum Uncertainty Bound ในช่วงเวลาที่จำกัด

1. การสร้างเส้นทางเดิน

การกำหนดคำสั่งการเคลื่อนที่ไปยังจุดที่กำหนด (Erkorkmaz *et al.*, 2001) โดยการแปลงคำสั่งจากผู้ใช้งาน คำสั่งรหัสจึงเป็นสัญญาณคำสั่งที่ทำการสั่งให้ตัวควบคุมซึ่งมีลักษณะเป็นแถวของจุดข้อมูลตำแหน่งที่เรียกว่า จุดตั้งค่า (Set Point) โดยโปรแกรมที่ทำการกำหนดจุดตั้งค่าเรียกว่าส่วนกำหนดคำสั่ง (Command Generator) โดยปกติแล้วส่วนใหญ่ของรูปทรงต่างๆจะประกอบไปด้วยส่วนของเส้นตรงและเส้นโค้ง ดังนั้นจะกล่าวถึงวิธีการส่งค่าความเร็วสำหรับเส้นตรง 2 มิติโดยวิธีแบ่งระยะคงที่ และสร้างจุดตั้งค่าเป็นเส้นโค้ง

1.1 การส่งค่าความเร็วสำหรับเส้นตรง 2 มิติโดยวิธีแบ่งระยะคงที่

ในการส่งค่าแต่ละช่วงสามารถคำนวณได้ดังนี้ กำหนดให้ T_{min} คือช่วงเวลาที่สั้นที่สุดที่สามารถควบคุมให้มอเตอร์ทำงานได้ และกำหนดให้ f_{max} คือค่าความเร็วสูงสุดของมอเตอร์ ดังนั้นระยะการเคลื่อนที่สั้นที่สุดที่มอเตอร์เคลื่อนที่ได้คือ

$$\Delta U_{min} = T_{min} \cdot f_{max} \quad (1)$$

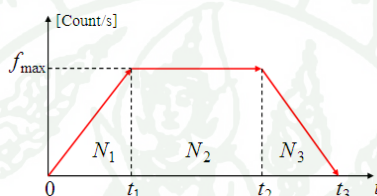
ในการส่งค่าความเร็วแบบระยะการเคลื่อนที่คงที่ค่า ΔU จะมีค่าคงที่ กำหนดให้ระยะทางการเคลื่อนที่ทั้งหมดเท่ากับ L และ N คือจำนวนช่วงทั้งหมดของการเคลื่อนที่ ซึ่งสามารถหาค่า N ได้จาก

$$N = \frac{L}{\Delta U} \quad (2)$$

สังเกตว่ามีความเป็นไปได้ที่ค่า N จะเป็นทศนิยม ดังนั้นต้องทำให้ N เป็นจำนวนเต็ม โดยการปัดค่า N ลงเพื่อไม่ให้ค่า $\Delta U < \Delta U_{min}$ จากนั้นคำนวณหาค่า ΔU ใหม่ซึ่งหาค่าได้จาก

$$\Delta U = \frac{L}{N} \quad (3)$$

เมื่อพิจารณาโปรไฟล์ความเร็วที่ป้อนให้กับมอเตอร์มีลักษณะเป็นรูปสี่เหลี่ยมคางหมู ซึ่งสามารถแบ่งโปรไฟล์ความเร็วได้เป็น 3 ช่วงใหญ่ๆ โดยส่วนแรกเป็นส่วนที่ความเร็วมีค่าเพิ่มขึ้น ในส่วนนี้จะแบ่งการเคลื่อนที่ออกเป็น N_1 ช่วง ในส่วนที่สองเป็นส่วนที่ความเร็วมีค่าคงที่ ในส่วนนี้จะถูกแบ่งจำนวนช่วงออกเป็น N_2 ช่วง และส่วนที่สามเป็นส่วนที่ความเร็วลดลงซึ่งมีจำนวน N_3 ช่วง เมื่อ $N = N_1 + N_2 + N_3$ และ $N_1 = N_3$ โดยกำหนดให้ช่วง N_1 มีความเร่งคือ A และช่วง N_2 มีความหน่วงคือ D ดังแสดงในภาพที่ 1



ภาพที่ 1 การส่งค่าความเร็วในแต่ละช่วง

กำหนดให้การเคลื่อนที่ที่จะเริ่มจากหยุดนิ่งแล้วเคลื่อนที่ไปเรื่อยๆ จนหยุดนิ่งอีกทีเมื่อถึงตำแหน่งที่กำหนด พิจารณาในช่วงที่กราฟมีความเร็วเพิ่มขึ้น ($0 < t < t_1$) ระยะทางที่เคลื่อนที่ได้ l_1 หาได้จาก

$$l_1 = \int_0^{t_1} A t dt = \frac{A t_1^2}{2} \quad (4)$$

เมื่อ $t_1 = \frac{f}{A}$ ดังนั้นสามารถหา N_1 ได้ดังนี้

$$N_1 = \frac{l_1}{\Delta U} = \frac{f^2}{2 \cdot A \cdot \Delta U} \quad (5)$$

โดยทั่วไปจะกำหนดให้ $A = D$ ดังนั้น $N_1 = N_3$ และหา N_2 จาก

$$N_2 = N - N_1 - N_3 \quad (6)$$

ดังนั้นจะสามารถหาความเร็วและระยะห่างในแต่ละช่วงได้ดังนี้

$$\begin{aligned} \text{For } k &= 1 : N_1 \\ f(k) &= \sqrt{f_0^2 + 2 \cdot k \cdot A \cdot \Delta U} \\ T(k) &= \frac{2 \cdot \Delta U}{f(k) + f(k-1)} \end{aligned} \quad (7)$$

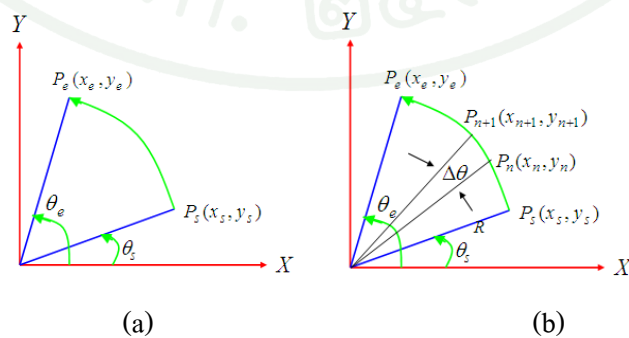
$$\begin{aligned} \text{For } k &= N_1 : N_2 \\ f(k) &= f_{max} \\ T(k) &= \frac{\Delta U}{f_{max}} \end{aligned} \quad (8)$$

$$\begin{aligned} \text{For } k &= N_2 : N_3 \\ f(k) &= \sqrt{f^2 - 2 \cdot k \cdot D \cdot \Delta U} \\ T(k) &= \frac{2 \cdot \Delta U}{f(k) - f(k-1)} \end{aligned} \quad (9)$$

1.2 การส่งค่าความเร็วสำหรับวงกลมสองมิติโดยวิธีแบ่งระยะคงที่

กำหนดให้การเคลื่อนที่ในแนวเส้นโค้งตามแนวเส้นรอบวงของวงกลมมีลักษณะดังภาพที่ 2(a) โดยจุด P_s เป็นจุดเริ่มต้นและ P_e เป็นจุดสิ้นสุด ระยะการเคลื่อนที่ตามแนวเส้นโค้งของวงกลมหาได้จาก

$$L = R(\theta_e - \theta_s) = R\theta_t \quad (10)$$



ภาพที่ 2 การเคลื่อนที่ในแนวเส้นโค้ง

จากนั้นระยะทางการเคลื่อนที่ตามแนวเส้นรอบวงจะถูกแบ่งออกเป็นช่วงสั้นๆ จำนวน N ช่วง แต่ละช่วงจะมีความยาวตามแนวเส้นรอบวงเท่ากับ ΔU และถูกรองรับด้วยมุม $\Delta\theta$ เท่ากัน ทุกช่วงดังภาพที่ 5(b) ซึ่งในการออกแบบต้องให้

$$\text{chord error} = R(1 - \cos \frac{\Delta\theta}{2}) < 1 \text{ (mm)} \quad (11)$$

ดังนั้นเราสามารถหาค่า $\Delta\theta$ ได้จาก

$$\Delta\theta = 2 \cdot \cos^{-1} \left(\frac{R-1}{R} \right) \quad (12)$$

และสามารถหาระยะห่างตามแนวเส้นรอบวงของแต่ละช่วงเท่ากับ

$$\Delta U = R \cdot \Delta\theta \quad (13)$$

การเคลื่อนที่แต่ละช่วงจะมีระยะห่างของมุมเท่ากับ $\Delta\theta$ และความเร็วเชิงมุม ω หาได้จาก

$$\omega(k) = \frac{f(k)}{R} \quad (14)$$

และมุมที่ตำแหน่ง k หาได้จาก

$$\theta(k) = \omega k = \frac{f(k)}{R} k \quad (15)$$

ดังนั้นจึงสามารถหาค่าตำแหน่งการเคลื่อนที่ของแต่ละช่วงในแต่ละแกนได้จาก

$$x(k) = R \cdot \cos \theta(k) = R \cdot \cos \left(\frac{f(k)}{R} k \right) \quad (16)$$

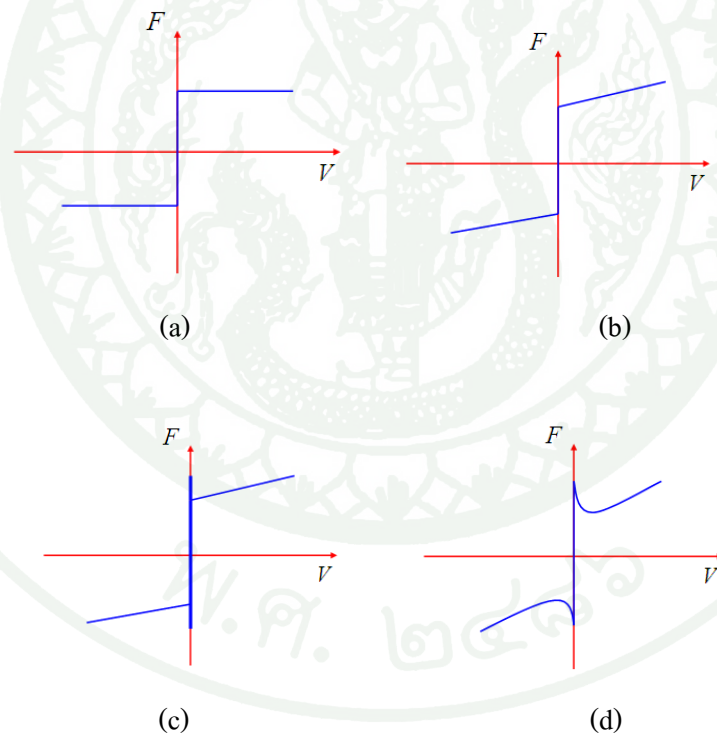
$$y(k) = R \cdot \sin \theta(k) = R \cdot \sin \left(\frac{f(k)}{R} k \right)$$

และสามารถหาความเร็วของการเคลื่อนที่ทั้งในแกน X และ Y ได้จากการหาอนุพันธ์ของระยะการเคลื่อนที่ในแนวแกน X และแนวแกน Y ตามลำดับ ดังนั้นจะได้ความเร็วของการเคลื่อนที่ทั้งในแกน X และ Y คือ

$$f_x(k) = -\frac{f(k)}{R} \cdot R \cdot \sin\left(\frac{f(k)}{R} \cdot k\right) = -\frac{f(k)}{R} \cdot y(k) \quad (17)$$

$$f_y(k) = \frac{f(k)}{R} \cdot R \cdot \cos\left(\frac{f(k)}{R} \cdot k\right) = \frac{f(k)}{R} \cdot x(k) \quad (18)$$

2. แบบจำลองแรงเสียดทาน



ภาพที่ 3 แบบจำลองแรงเสียดทาน

แรงเสียดทาน (Olsson *et al.*, 1998) คือแรงในแนวสัมผัสระหว่างผิวสัมผัสของวัตถุ 2 ชิ้นที่ติดกัน และมีแนวโน้มที่จะเกิดการเลื่อนระหว่างผิวสัมผัสคู่นี้ แรงเสียดทานที่กระทำบนวัตถุมีทิศ

ทางตรงข้ามกับแนวโน้มนการเคลื่อนที่ของวัตถุนั้น โดยทั่วไปสามารถแบ่งชนิดของแรงเสียดทานได้ดังนี้

2.1 ส่วนประกอบของแรงเสียดทาน

1) แรงเสียดทานคูลอมบ์ (Coulomb Friction) คือ แรงเสียดทานระหว่างผิวสัมผัส 2 ผิวที่สัมผัสกันโดยไม่มีการหล่อลื่น ดังภาพที่ 3(a) ซึ่งมีขนาดดังสมการ (19)

$$F_c = \mu F_N \quad (19)$$

2) แรงเสียดทานหนืด (Viscous Friction) เป็นแรงที่มีสัดส่วนกับความเร็วดังสมการที่ (20)

$$F = F_v v \quad (20)$$

3) แรงเสียดทานสถิตย์ (Static Friction) คือแรงเสียดทานที่เกิดขึ้นในสถานะที่วัตถุได้รับแรงจากหยุดหนึ่งจนถึงวัตถุเตรียมที่จะเคลื่อนที่ซึ่งมีขนาดดังสมการ (21)

$$F = \begin{cases} F_e & ; |F_e| < F_s \\ F_s \operatorname{sgn}(F_e) & ; |F_e| \geq F_s \end{cases} \quad (21)$$

4) ผลกระทบสไตรเบ็ค (Stribeck Effect) เป็นปรากฏการณ์ในช่วงของการเคลื่อนที่ซึ่งแรงเสียดทานเริ่มเปลี่ยนจากแรงเสียดทานสถิตย์เป็นแรงเสียดทานหนืด เมื่อรวมแรงเสียดทานทุกส่วนเข้าด้วยกันสามารถแสดงได้ดังนี้

$$F = \begin{cases} F(v) & ; v \neq 0 \\ F_e & ; v = 0 \end{cases} \quad (22)$$

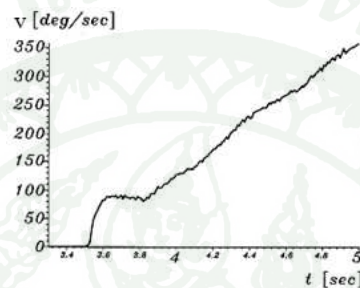
ซึ่งค่า $F(v)$ โดยทั่วไปมีลักษณะดังสมการ (23)

$$F(v) = \begin{cases} F_c^+ + F_v^+ v & ; v > a \\ F_c^- + F_v^- v & ; v < -a \end{cases} \quad (23)$$

เมื่อ a เป็นจำนวนจริงบวกใดๆ

2.2 การหาแบบจำลองแรงเสียดทานตามวิธีของ K. Rafael

โดยทั่วไปการเคลื่อนที่ของมอเตอร์ (Rafael *et al.*, 1999) สามารถแสดงได้ดังภาพที่ 4



ภาพที่ 4 การเคลื่อนที่ของมอเตอร์ (Rafael *et al.*, 1999)

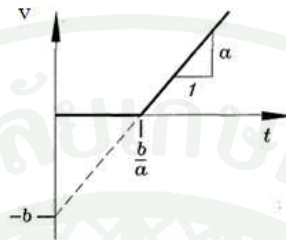
จากภาพที่ 4 จะเห็นได้ว่าการเคลื่อนที่ของมอเตอร์สามารถแบ่งได้เป็นสามช่วงคือ

1. ช่วงที่มอเตอร์หยุดนิ่งในช่วงเวลาหนึ่ง จากนั้นมอเตอร์เริ่มมีการเคลื่อนที่อย่างรวดเร็ว
2. หลังจากที่มีมอเตอร์เริ่มเคลื่อนที่แล้ว มอเตอร์จะหมุนด้วยความเร็วคงที่อยู่ช่วงเวลาหนึ่ง หรืออาจจะลดความเร็วลงในช่วงเวลาสั้นๆ
3. ในช่วงเวลาต่อมา เป็นช่วงเวลาที่มอเตอร์มีการเคลื่อนที่เป็นแบบเชิงเส้น

ในการหาแบบจำลองแรงเสียดทาน ถ้าพิจารณาเฉพาะผลจากแรงเสียดทานคูลอมบ์ (Coulomb Friction) และแรงเสียดทานหนืด (Viscous Friction) จะได้สมการแรงเสียดทานคือ

$$F(v) = F_v v + F_c \operatorname{sgn}(v) \quad (24)$$

จากการป้อนค่า Torque Ramp และกำหนดให้ความเร็วเริ่มต้นเป็นศูนย์ จะสามารถแสดงลักษณะของแรงเสียดทานคูลอมบ์ (Coulomb Friction) และแรงเสียดทานหนืด (Viscous Friction) ได้ดังภาพที่ 5



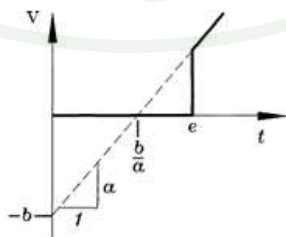
ภาพที่ 5 แรงเสียดทานคูลอมบ์และแรงเสียดทานหนืด (Rafael *et al.*, 1999)

เมื่อ a คือ ความชันของกราฟ
-b คือ จุดตัดแกน v

ถ้าคิดปรากฏการสไตรเบค (Stribeck Effect) ที่เกิดขึ้น เพื่อให้แบบจำลองแรงเสียดทานมีลักษณะใกล้เคียงกับแรงเสียดทานที่เกิดขึ้นจริงจะได้สมการตามแบบจำลองของ Armstrong (Armstrong Model) สามารถแสดงได้ดังนี้

$$F(v) = F_v v + F_c \operatorname{sgn}(v) + [F_s - F_c] \operatorname{sgn}(v) e^{(v/v_s)^2} \quad (25)$$

และสามารถแสดงรูปแบบการเคลื่อนที่ตามแบบจำลองของ Armstrong (Armstrong Model) ได้ดังภาพที่ 6



ภาพที่ 6 การเคลื่อนที่ตามแบบจำลองของ Armstrong (Rafael *et al.*, 1999)

- เมื่อ a คือ ความชันของกราฟ
 -b คือ จุดตัดแกน v
 c คือ จุดที่มีการเปลี่ยนแปลงความเร็วอย่างรวดเร็ว
 F_s คือ แรงเสียดทานสถิต (Static Friction)

จากกราฟเราสามารถคำนวณหาค่าแรงเสียดทานคูลอมบ์ (Coulomb Friction) และแรงเสียดทานหนืด (Viscous Friction) ได้ดังนี้

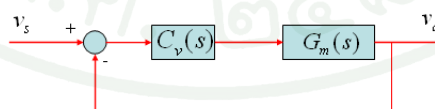
$$F_v = \frac{m}{a} \quad (26)$$

$$F_c = \frac{b}{a}m \quad (27)$$

และค่าแรงเสียดทานสถิต (Static Friction) สามารถหาได้โดยการป้อน Torque Ramp Input แล้ววัดค่าแรงเสียดทานในจุดที่มอเตอร์เริ่มเคลื่อนที่

3. การชดเชยแรงเสียดทานป้อนไปหน้า

ในการควบคุมมอเตอร์โดยทั่วไปสามารถทำการควบคุมด้วยวิธีการควบคุมแบบวงปิด ดังภาพที่ 7

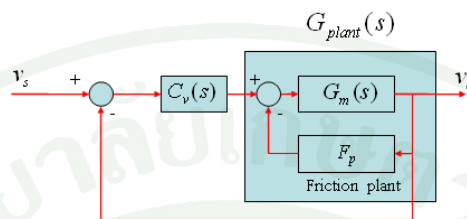


ภาพที่ 7 การควบคุมมอเตอร์แบบวงปิด

จากภาพที่ 7 จะได้ฟังก์ชันถ่ายโอนของระบบคือ

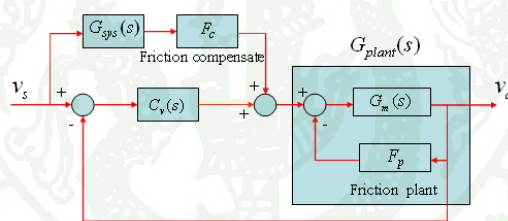
$$\frac{V_a(s)}{V_s(s)} = \frac{C_v(s)G_m(s)}{1 + C_v(s)G_m(s)} \quad (28)$$

เมื่อทำการต่อมอเตอร์เข้ากับรางจะเกิดแรงเสียดทานขึ้นที่ Ball Screw ราง และที่ Bearing ทำให้ Block Diagram การควบคุมของทั้งระบบเป็นดังภาพที่ 8



ภาพที่ 8 Block Diagram การควบคุมทั้งระบบ

ต่อไปเป็นการออกแบบแรงเสียดทานป้อนไปหน้า ซึ่งทำการออกแบบเพื่อกำจัดแรงเสียดทานที่เกิดขึ้นระหว่างรางกับมอเตอร์ โดยทำการออกแบบแรงเสียดทานป้อนไปหน้าดังภาพที่ 9



ภาพที่ 9 การชดเชยแรงเสียดทานป้อนไปหน้า

จากภาพที่ 9 เราต้องการทำให้แรงเสียดทานถูกหักล้าง ดังนั้น

$$F_p(v_a) = F_c(L^{-1}\{G_{sys}(s)V_s(s)\}) \quad (29)$$

โดยที่ $L^{-1}\{G_{sys}(s)V_s(s)\}$ คือการแปลงลาปลาซผกผัน เงื่อนไขที่ทำให้แรงเสียดทานถูกหักล้างมี 2 เงื่อนไขดังนี้

เงื่อนไขแรกคือแรงเสียดทานที่มีอยู่ใน Plant ต้องมีค่าเท่ากับแรงเสียดทานที่ต้องการนำมาชดเชย กล่าวคือ

$$F_p = F_c \quad (30)$$

และอีกเงื่อนไขคือ

$$v_a = L^{-1}\{G_{sys}(s)V_s(s)\} \quad (31)$$

โดยเงื่อนไขที่สองจะเป็นจริงได้เมื่อทำการออกแบบ $G_{sys}(s)$ ให้เป็น Close Loop Transfer Function กล่าวคือ

$$G_{sys}(s) = \frac{C_v(s)G_m(s)}{1 + C_v(s)G_m(s)} \quad (32)$$

4. ภาษาโปรแกรมสำหรับระบบควบคุมซีเอ็นซี

ทรงชัย (2549) ได้สรุปการใช้คำสั่งสำหรับระบบควบคุมซีเอ็นซีคือรหัสจี (G-Code) ในการติดต่อกับผู้ใช้เพื่อสั่งการทำงานของแท่น XYZ โดยมีรายละเอียดของแต่ละคำสั่งดังตารางที่ 1

ตารางที่ 1 ฟังก์ชัน G-Code พื้นฐาน

คำสั่งรหัส	ความหมาย/การทำงาน
G00	การเคลื่อนที่เร็ว
G01	การเคลื่อนที่แนวเส้นตรงโดยสามารถควบคุมอัตราการป้อน
G02	การเคลื่อนที่แนวส่วนโค้งตามเข็มนาฬิกา
G03	การเคลื่อนที่แนวส่วนโค้งทวนเข็มนาฬิกา
G04	เวลาหยุด (Dwell)
G13-G16	การเลือกแกน
G17-G19	การเลือกระนาบ
G40	ยกเลิกการชดเชยขนาดรัศมีมีดกัด

ตารางที่ 1 (ต่อ)

คำสั่งรหัส	ความหมาย/การทำงาน
G41	เรียกใช้การชดเชยขนาดรัศมีมีดกัดด้านซ้ายของเส้นขอบรูป
G42	เรียกใช้การชดเชยขนาดรัศมีมีดกัดด้านขวาของเส้นขอบรูป
G43	เรียกใช้การชดเชยขนาดรัศมีมีดกัดมุมใน
G44	เรียกใช้การชดเชยขนาดรัศมีมีดกัดมุมนอก
G90	การกำหนดขนาดแบบสัมบูรณ์ (Absolute)
G91	การกำหนดขนาดเชิงส่วนเพิ่ม (Incremental)
G94	อัตราป้อนเป็น มม.ต่อนาที
G95	อัตราป้อนเป็น มม. ต่อรอบ
M00	หยุดโปรแกรม
M03	สปินเดิลหมุนตามเข็มนาฬิกา
M04	สปินเดิลหมุนทวนเข็มนาฬิกา
M05	หยุดสปินเดิล
M06	เปลี่ยนเครื่องมือ
M07	เปิดปั๊มสารหล่อเย็นหมายเลข 2
M08	เปิดปั๊มสารหล่อเย็นหมายเลข 1
M09	ปิดปั๊มสารหล่อเย็น

ส่วนของคำสั่งสำคัญๆ ที่ใช้เป็นมาตรฐานในภาษาโปรแกรมสำหรับระบบควบคุมซีเอ็นซีได้แก่

- G00 การเคลื่อนที่เร็ว
- G01 การเคลื่อนที่แนวเส้นตรงตามค่าอัตราป้อน
- G02 การเคลื่อนที่แนวส่วนโค้งตามเข็มนาฬิกา
- G03 การเคลื่อนที่แนวส่วนโค้งทวนเข็มนาฬิกา

4.1 คำสั่งการเคลื่อนที่เร็ว (G00)

คำสั่งที่ใช้ในการทำงานแบบเคลื่อนที่เร็วจะกำหนดในโปรแกรมด้วยคำสั่ง G00 คำสั่งนี้จะใช้ในการเคลื่อนที่ของเครื่องมือ เช่น มีดกัด มีดกลึง เป็นต้น ไปยังจุดเป้าหมายอัตราเคลื่อนที่เร็วของเครื่องมือใช้คำสั่งนี้จะต้องมีเงื่อนไขเสริมคือ ค่าพิกัดแกน XYZ ของจุดเป้าหมายที่ต้องการเคลื่อนเครื่องมือไป ตัวอย่างการใช้คำสั่ง G00 คือ

G00 ตำแหน่งค่าแกน X ตำแหน่งค่าแกน Y ตำแหน่งค่าแกน Z

ตัวอย่าง เช่น G00 X5 Y-5 Z3 หมายความว่า วิ่งไปที่แกน X เป็นระยะ 5 ซม. แกน Y เป็นระยะ -5 ซม. และแกน Z เป็นระยะ 3 ซม. ด้วยอัตราเร็วสูงสุด

4.2 การเคลื่อนที่แนวเส้นตรงตามค่าอัตราป้อน (G01)

เป็นคำสั่งที่ใช้สำหรับการเคลื่อนที่แนวเส้นตรงตามค่าอัตราป้อนที่ใช้ซึ่งจะมีเงื่อนไขในการทำงานคือ ค่าพิกัดแกน XYZ ของจุดเป้าหมายและอัตราป้อน ตัวอย่างการใช้คำสั่ง G01 คือ

G01 ตำแหน่งจุดเป้าหมาย อัตราป้อน

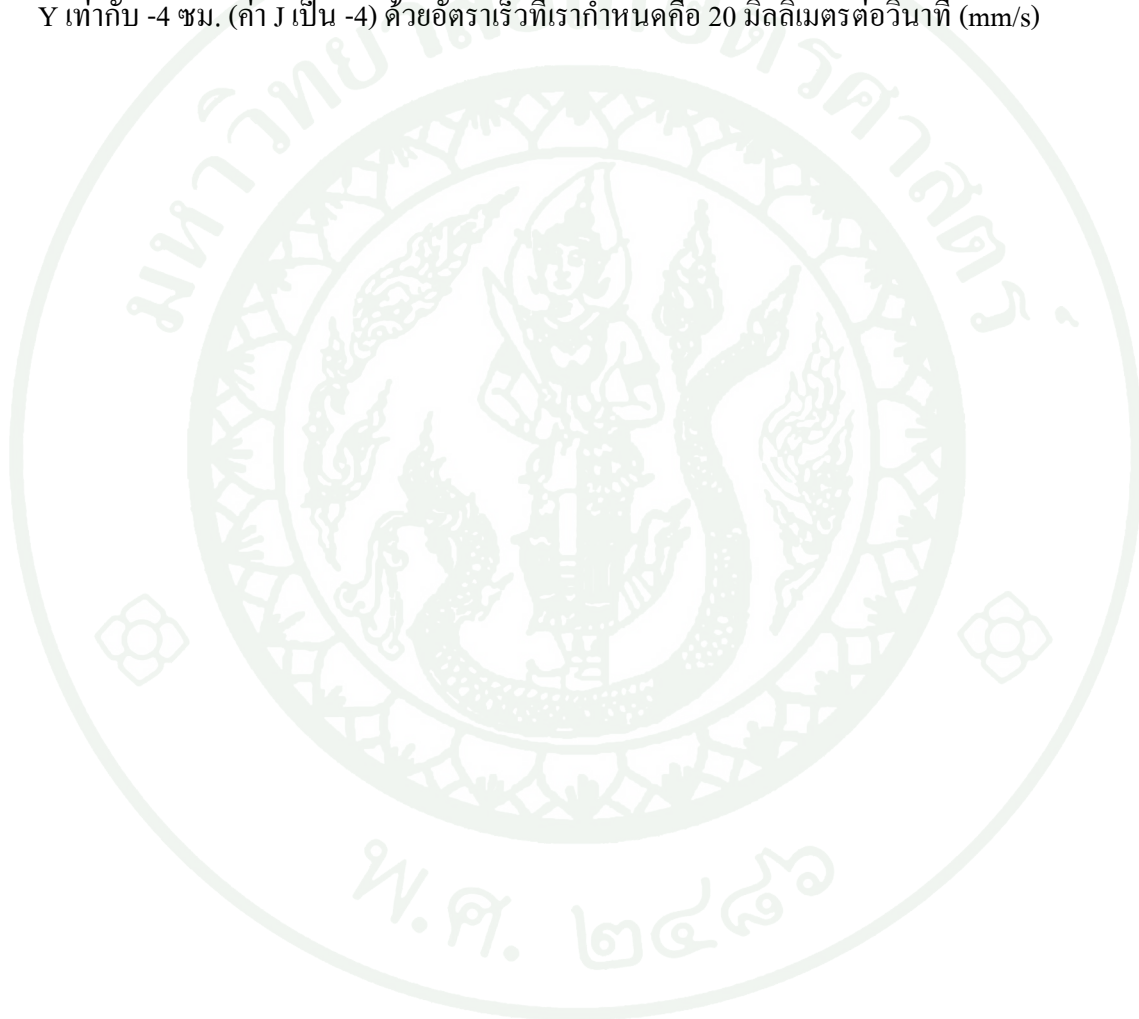
ตัวอย่าง เช่น G01 X10 Y-5 Z3 F10 หมายความว่า วิ่งไปที่ค่าแกน X เป็นระยะ 10 ซม. ค่าแกน Y เป็นระยะ -5 ซม. และค่าแกน Z เป็นระยะ 3 ซม. ด้วยอัตราเร็วที่เรากำหนดคือ 10 โดยหน่วยเป็นมิลลิเมตรต่อวินาที (mm/s)

4.3 การเคลื่อนที่แนวส่วนโค้งตามเข็มนาฬิกาและทวนเข็มนาฬิกา (G02/G03)

คำสั่งสำหรับการเคลื่อนที่แนวเส้นโค้งที่กำหนดเป็นมาตรฐานจะมีลักษณะการเคลื่อนที่แตกต่างกันระหว่างคำสั่ง G02 และ G03 ซึ่งขึ้นอยู่กับทิศทางการหมุน การใช้คำสั่งเคลื่อนที่แนวส่วนโค้งตามเข็มนาฬิกา (G02) และการใช้คำสั่งเคลื่อนที่แนวส่วนโค้งทวนเข็มนาฬิกา (G03) จะมีเงื่อนไขในการทำงานคือ ค่าพิกัดแกน XYZ ของจุดเป้าหมาย ข้อมูลของขนาดรัศมีหรือจุดศูนย์กลางส่วนโค้งและอัตราป้อน ตัวอย่างการใช้คำสั่ง G02/G03 คือ

G02/G03 ตำแหน่งจุดเป้าหมาย ข้อมูลรหัสมีส่วนโค้ง อัตราป้อน

ตัวอย่าง เช่น G02 X0 Y0 Z1 I-4 J-4 F20 หมายความว่า วิ่งเป็นวงกลมในทิศทางตามเข็มนาฬิกาไปที่ค่าแกน X เป็นระยะ 0 ซม. ค่าแกน Y เป็นระยะ 0 ซม. และค่าแกน Z เป็นระยะ 1 ซม. โดยจุดเริ่มต้นห่างจากจุดศูนย์กลางในทิศทางแกน X เท่ากับ -4 ซม. (ค่า I เป็น -4) และทิศทางแกน Y เท่ากับ -4 ซม. (ค่า J เป็น -4) ด้วยอัตราเร็วที่เรากำหนดคือ 20 มิลลิเมตรต่อวินาที (mm/s)



อุปกรณ์และวิธีการ

อุปกรณ์

1. คอมพิวเตอร์ 1 ชุด
2. โปรแกรม Matlab Version 7
3. โปรแกรม LabVIEW Version 8.2
4. เครื่อง CNC 3 แกน 1 ชุด
5. AC Servo Motor ของ Panasonic รุ่น MSMA082A1C พร้อมชุดขับ 3 ชุด
6. ชุดวงจรสำหรับการแปลงสัญญาณ +/- 10 Volt เป็น 4-20 mAmp
7. Power Supply
8. NI PCI-6251 M SERIES DAQ
9. Compaq Fieldpoint รุ่น 2100
10. Compaq Fieldpoint รุ่น cFP-AIO-600
11. Compaq Fieldpoint รุ่น cFP-QUAD-510

วิธีการ

1. การทำงานของระบบโดยรวม

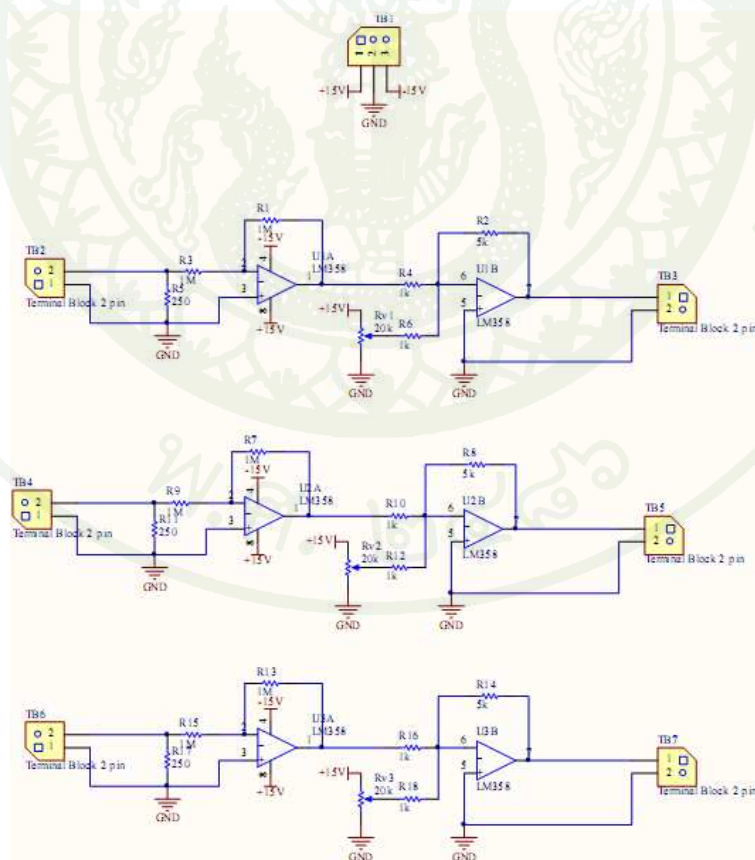
ในการทดลองจะทำการสร้างเส้นทางการเคลื่อนที่ โดยใช้ผ่านทาง GUI ของโปรแกรม Matlab โดยผู้ใช้งานจะทำการป้อนรหัสจี (G-Code) ผ่านทาง GUI เมื่อผู้ใช้งานป้อนรหัสจีเรียบร้อยแล้ว Matlab จะส่งข้อมูลเข้าไปที่ส่วนของ Trajectory Generation เพื่อคำนวณค่าตำแหน่งและเวลาซึ่งใช้สำหรับส่งเป็นค่า Set Point เมื่อกระบวนการในการคำนวณเรียบร้อยแล้ว จะนำค่าตำแหน่งและเวลาทั้งหมดมาทำการควบคุมด้วยวิธีต่างๆต่อไป การทำงานโดยรวมสามารถแสดงดังภาพที่ 10



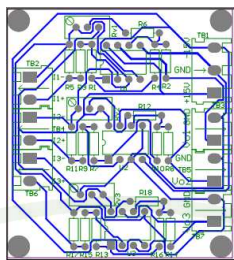
ภาพที่ 10 การทำงานโดยรวม

2. การออกแบบวงจรแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์

เนื่องจากสัญญาณที่ออกมาจากบอร์ดมีค่าแรงดันระหว่าง -10 ถึง +10 โวลต์ แต่สัญญาณ Input ที่ต้องส่งเข้า Field point ต้องส่งอยู่ในช่วง 4 ถึง 20 มิลลิแอมป์ ดังนั้นจึงต้องสร้างวงจรเพื่อทำการแปลงสัญญาณให้อยู่ในช่วงดังกล่าว ซึ่งวงจรสำหรับการแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์ สามารถแสดงได้ดังภาพที่ 11 และภาพที่ 12



ภาพที่ 11 วงจรสำหรับการแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์



ภาพที่ 12 วงจรสำหรับการแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์

วงจรแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์ สามารถแสดงได้ดังภาพที่ 13



ภาพที่ 13 วงจรแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์

รายการอุปกรณ์ที่ใช้ในการสร้างวงจรสำหรับการแปลงสัญญาณ +/- 10 โวลต์ เป็น 4-20 มิลลิแอมป์ สามารถแสดงได้ดังตารางที่ 2

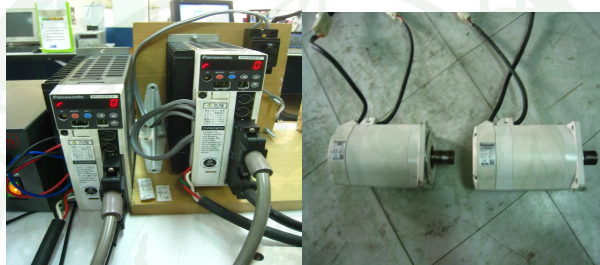
ตารางที่ 2 อุปกรณ์ในการสร้างวงจรแปลงสัญญาณ +/- 10 โวลต์เป็น 4-20 มิลลิแอมป์

สัญลักษณ์	ประเภท	รายละเอียด
R1, R3, R7, R9, R13, R15	Resister	1 M Ohms 1% 0.25 Watts
R2, R8, R14	Resister	5 K Ohms 1% 0.25 Watts
R4, R6, R10, R12, R16, R18	Resister	1 K Ohms 1% 0.25 Watts
R5, R11, R17	Resister	250 Ohms 1% 0.25 Watts
Rv1, Rv2, Rv3	Potentiometer	20K Ohms
TB1	Terminal Block	Terminal Block 3 Pins
TB2, TB3, TB4, TB5, TB6, TB7	Terminal Block	Terminal Block 2 Pins
U1, U2, U3	Op-Amp	Dual Op-Amp LM385

3. ชุดขับเคลื่อน AC Servo Motor และ Motor

ในส่วนนี้จะกล่าวถึงชุดขับเคลื่อน AC Servo Motor โดยชุดขับเคลื่อน AC Servo Motor เป็นชุดขับเคลื่อนสำเร็จรูปที่มาพร้อมกับ AC Servo Motor ของบริษัท Panasonic

ชุดขับเคลื่อน AC Servo Motor จะทำงานโดยรับค่าแรงดันระหว่าง -10 ถึง +10 Volt เข้ามาในชุดขับเคลื่อนที่มาพร้อมกับ AC Servo Motor ของบริษัท Panasonic เพื่อมาแปลงเป็นแรงดันที่จะไปขับ AC Servo Motor โดย AC Servo Motor และ AC Servo Motor Drive ที่ทำการทดลองแสดงดังภาพที่ 14



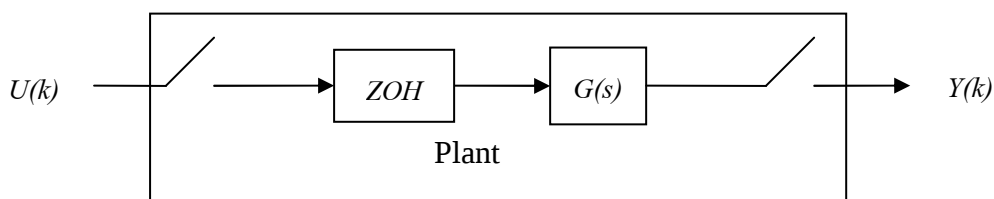
ภาพที่ 14 AC Servo Motor และ AC Servo Motor Drive

4. การหาแบบจำลองทางคณิตศาสตร์ในระบบจริงของ AC Motor

4.1 แบบจำลองเบื้องต้น

ในเบื้องต้นเราจะสมมุติว่าระบบจริงของเรามีโมเดลเป็น First Order โดยมีดังภาพที่ 15

$$G_{ZOH}(z)$$

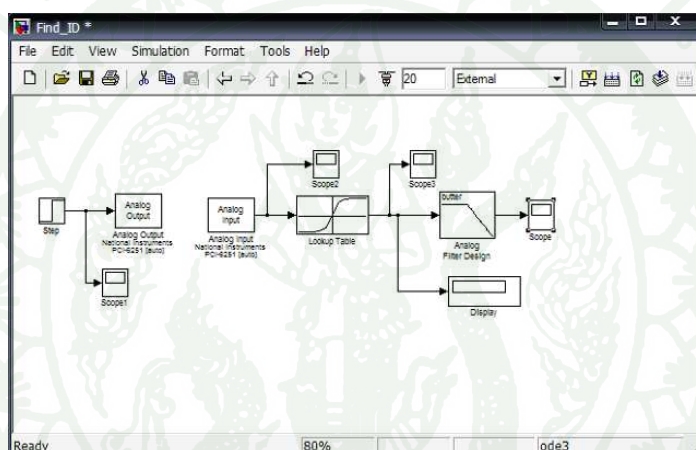


ภาพที่ 15 แสดงการหาแบบจำลองโดยรวม

โดยที่ $G(s) = \frac{K}{s+a}$ เมื่อ K และ a เป็นค่าคงที่

4.2 การทดลองและเก็บข้อมูล

ในการเก็บข้อมูลเพื่อหาโมเดลของระบบจะใช้ Simulink ที่อยู่ในโปรแกรม Matlab โดยทำการป้อนสัญญาณ Step ให้กับระบบโดยผ่านทาง NI Card และใช้ Sampling time ที่ 0.001 Second โดยมีรูปแบบการหาโมเดลของระบบโดยใช้ Real Time Windows Target ดังภาพที่ 16



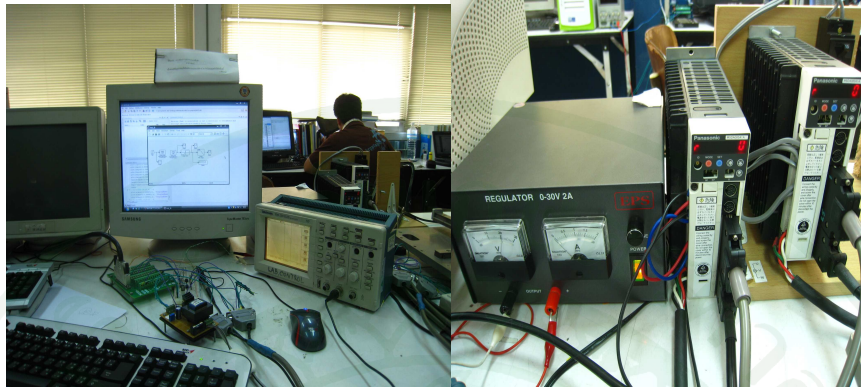
ภาพที่ 16 Simulink แสดงการหาโมเดลของระบบ

ในการทดลองนี้ได้ทำการเลือกค่า Parameter ใน Drive ตามตารางที่ 3

ตารางที่ 3 การเลือกค่า Parameter ใน Drive

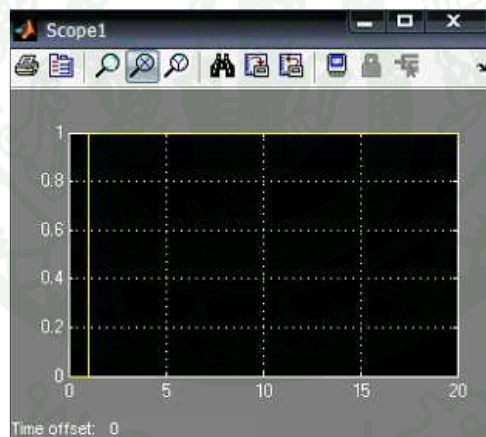
Parameter	รายละเอียด	ค่าที่เลือก
Pr 02	Control Mode Setup	1
Pr 07	Speed Monitor Selection	3
Pr 50	Velocity Command Input Gain	500
Pr 51	Velocity Command Input Logic Inversion	1
Pr 52	Velocity Command Offset	0

สามารถแสดงการเก็บข้อมูลดังภาพที่ 17



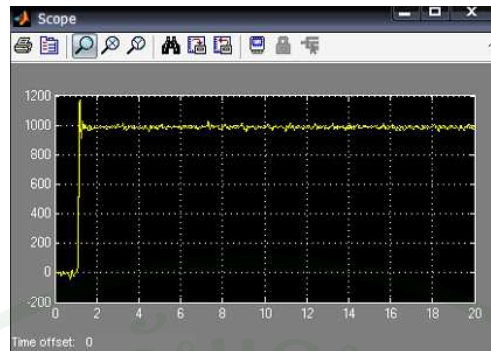
ภาพที่ 17 การเก็บข้อมูลเพื่อหาแบบจำลองของระบบ

หลังจากที่ Run โปรแกรมแล้วจะได้สัญญาณ Input U ซึ่งป้อนเป็น Step Input ดังภาพที่ 18



ภาพที่ 18 แสดงสัญญาณ Input U

และได้สัญญาณ Output Y ที่มอเตอร์เคลื่อนที่ได้ดังภาพที่ 19

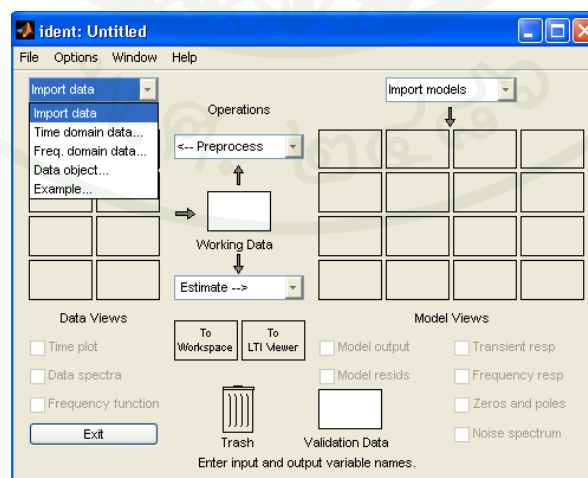


ภาพที่ 19 แสดงสัญญาณ Output Y

4.3 การหา Systems ID

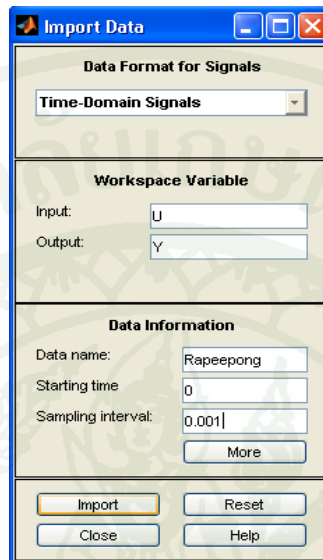
หลังจากที่เก็บข้อมูลเสร็จตามข้อ 8.2 แล้ว จะได้ข้อมูลอยู่ 2 ชุดคือ Input U และ Output Y ซึ่งข้อมูลที่ได้มานี้จะมี 2 คอลัมน์ คือส่วนของเวลาและความเร็ว แต่เมื่อเรานำข้อมูลมาทำการหาโมเดลของระบบ เราต้องการเพียงแค่ส่วนของความเร็ว โดยสามารถใช้คำสั่ง $U = \text{InputU}(:, \text{end});$ และ $Y = \text{OutputY}(:, \text{end});$ จากนั้นจะได้ข้อมูลที่จะนำมาหาโมเดลของระบบนั้นคือ U และ Y ต่อมาพิมพ์คำสั่ง `Ident` ในหน้าต่าง Command Windows ของโปรแกรม Matlab เพื่อนำข้อมูลมาหาโมเดลของระบบ โดยมีขั้นตอนดังนี้

ในช่อง Import data ให้เลือกที่ Time domain data ดังภาพที่ 20



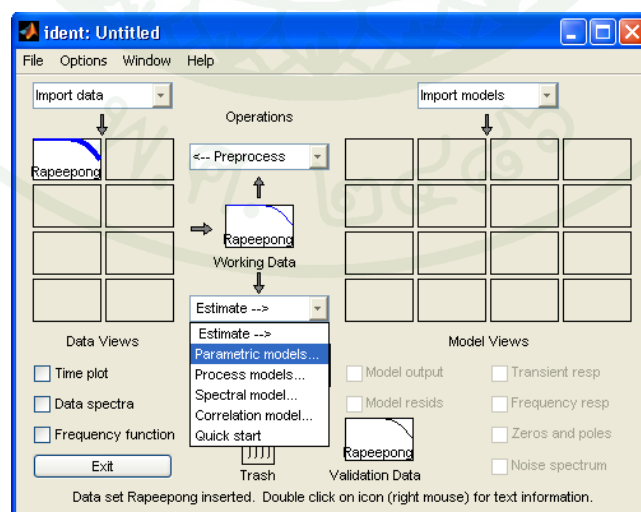
ภาพที่ 20 แสดงหน้าต่างการนำข้อมูลมาหาโมเดลของระบบ

หลังจากที่เลือก Import data จะพบหน้าต่างให้ใส่ข้อมูลที่ต้องการเพื่อหาโมเดลของระบบ เมื่อใส่ข้อมูลครบทุกตำแหน่งแล้วให้เลือกที่ Import ดังแสดงในภาพที่ 21



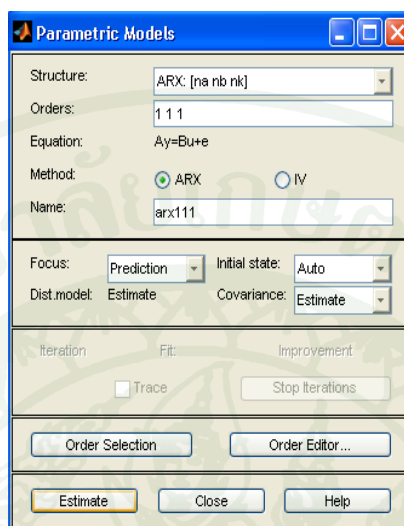
ภาพที่ 21 แสดงการ Import Data

หลังจากที่ Import Data เสร็จเรียบร้อยให้คลิกที่ตำแหน่ง Estimate และเลือก Parametric Models ดังแสดงในภาพที่ 22



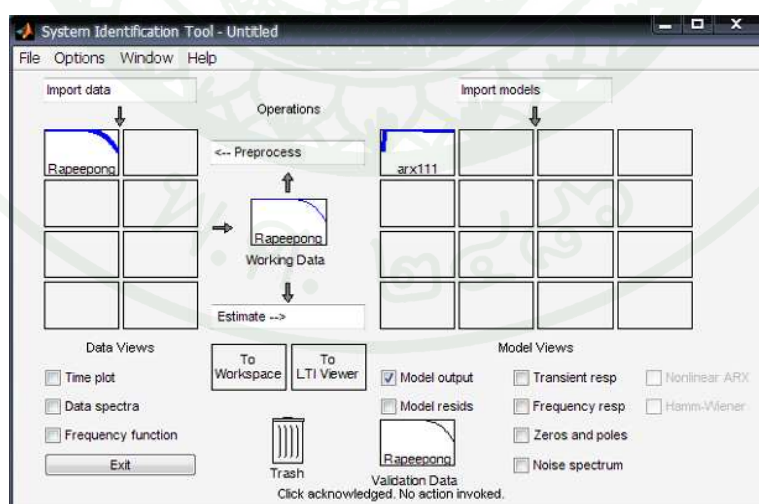
ภาพที่ 22 แสดงการประมาณค่าเพื่อหาโมเดลของระบบ

หลังจากที่เลือก Parametric Models แล้ว จะพบหน้าต่างดังภาพที่ 23 โดยสามารถเลือก โมเดลใดๆที่ต้องการทดสอบได้โดยการกำหนดค่าที่ตำแหน่ง Orders จากนั้นเลือก Estimate



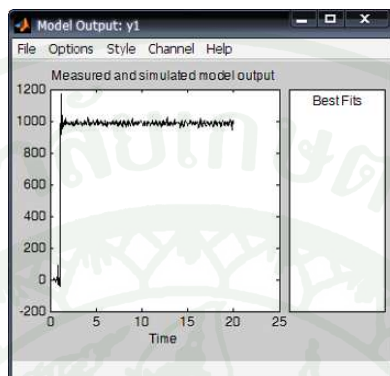
ภาพที่ 23 แสดงการเลือก Order ของ ARX Model

หลังจากที่เลือก Estimate จะพบว่าข้อมูลเข้ามาในตำแหน่งด้านล่างของ Import Models ดังภาพที่ 24

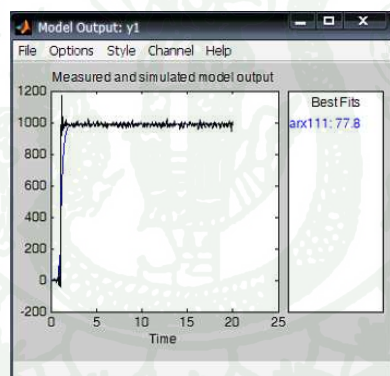


ภาพที่ 24 แสดง ARX Model ที่ได้หลังจากเลือก Estimate

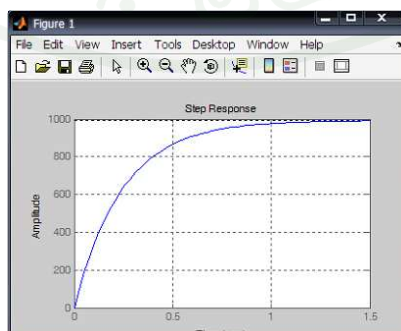
จากตรงนี้สามารถดู Output ของระบบจริงหรือดู Output ของโมเดลที่หาได้ หรือสามารถดู Output ของทั้งสองระบบพร้อมกันได้ นอกจากนี้ยังสามารถดูผลตอบสนองของระบบหรือสัญญาณอื่นๆตามที่ต้องการได้ ดังแสดงในภาพที่ 25 ถึงภาพที่ 28



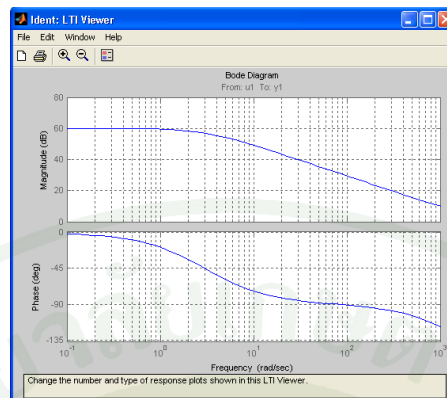
ภาพที่ 25 สัญญาณ Output ของระบบจริง



ภาพที่ 26 สัญญาณ Output ของโมเดล ARX111 เทียบกับสัญญาณจริง

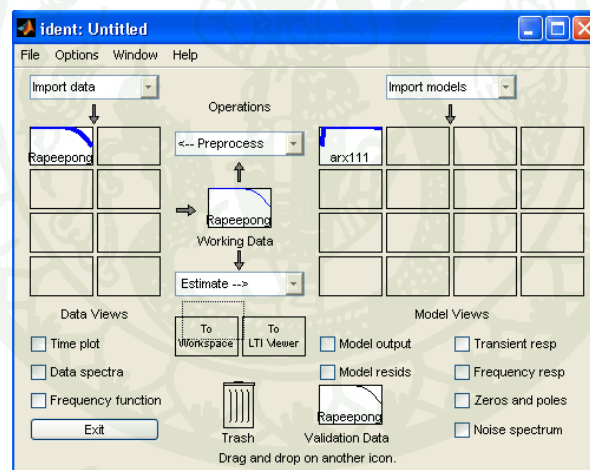


ภาพที่ 27 แสดง Output Response ของโมเดล ARX111



ภาพที่ 28 แสดง Bode Diagram ของโมเดล ARX11

ในลำดับต่อมาเลือกโมเดลที่สนใจเข้ามาที่ตำแหน่ง To Workspace ดังภาพที่ 29



ภาพที่ 29 แสดงการเลือกโมเดลที่สนใจมาที่ตำแหน่ง To Workspace

จากนั้นกลับมาที่หน้าหลักของ Matlab และใช้คำสั่งในการหาโมเดลที่ได้มา โดยมีขั้นตอนการเขียนโปรแกรมดังนี้

```
arx111
```

```
[n,d] = d2cm(arx111.B,arx111.A,0.001)
```

```
MySysID = tf(n,d)
```

จะได้ ARX Model ในแต่ละค่าดังนี้

ARX111:

Transfer Function: $G(z) = \frac{3.012}{z - 0.997}$

(33)

$$G(s) = \frac{3017}{s + 3.005} \quad (34)$$

Loss Functions = 0.0431905

Final Prediction Error = 0.0418927

ARX120:

Transfer Function: $G(z) = \frac{-0.05542z + 3.068}{z - 0.997}$ (35)

$$G(s) = \frac{-0.05542s + 3017}{s + 3.005} \quad (36)$$

Loss Functions = 0.418843

Final Prediction Error = 0.418969

ARX1110:

Transfer Function: $G(z) = \frac{0.3192}{z - 0.997}$ (37)

$$G(s) = \frac{319.3}{s + 0.3182} \quad (38)$$

Loss Functions = 0.431705

Final Prediction Error = 0.430969

4.4 สรุปการเลือกแบบจำลอง

จากการหาโมเดลของระบบโดยใช้ ARX Model ที่ได้กล่าวมาแล้วในข้างต้น จะเห็นว่า ARX111 มีค่า Loss Functions น้อยที่สุดและให้ค่า Output Response ที่ใกล้เคียงกับระบบจริงมาก

ที่สุด ส่วน ARX120 จะมี Zero อยู่ในตำแหน่งที่ไกลมากดังนั้นจึงสามารถประมาณค่าเป็นศูนย์ จะเห็นว่าเมื่อประมาณค่า Zero ของ ARX120 ให้เป็นศูนย์แล้วจะได้ค่า Transfer Function ของ ARX120 เท่ากับของ ARX111 ในกรณีของ ARX1110 จะให้ค่า Output Response ที่มี Delay ถึง 10 หน่วย และมีลักษณะของ Output Response ที่แตกต่างจากระบบจริงค่อนข้างมาก

จากเหตุผลทั้งหมดที่กล่าวมาสามารถสรุปได้ว่า โมเดลที่มีความใกล้เคียงกับระบบจริงมากที่สุดคือ ARX111 ดังนั้นโมเดลทางคณิตศาสตร์ในระบบจริงของ AC Motor จึงเป็นดังนี้

$$G(z) = \frac{3.012}{z - 0.997} \quad (39)$$

$$G(s) = \frac{3017}{s + 3.005} \quad (40)$$

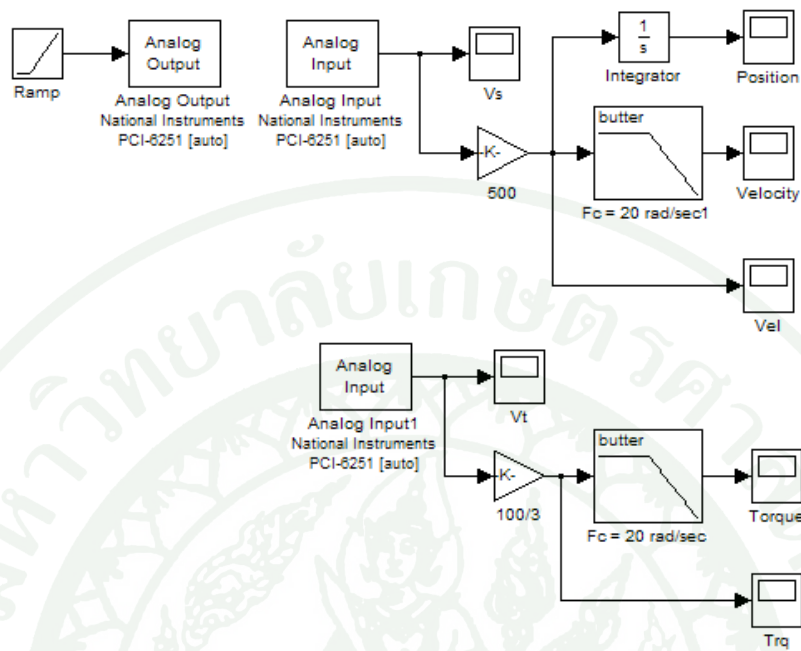
5. การหาแบบจำลองแรงเสียดทานที่เกิดขึ้นในระบบ

ในการหาแบบจำลองแรงเสียดทานที่เกิดขึ้นในระบบ สามารถทำได้โดยการป้อน Torque Ramp ให้กับระบบโดยผ่าน Field Point เริ่มจากการปรับ Drive ให้อยู่ใน Torque Mode และทำการเลือกค่า Parameter ใน Drive ตามตารางที่ 4

ตารางที่ 4 การเลือกค่า Parameter ใน Drive เพื่อหาแรงเสียดทาน

Parameter	รายละเอียด	ค่าที่เลือก
Pr 02	Control Mode Setup	2
Pr 08	Torque Monitor Selection	0
Pr 5C	Torque Command Input Gain	30
Pr 5D	Torque Command Input Inversion	0
Pr 5E	Torque Limit Setup	300

จากนั้นทำการป้อนสัญญาณ Torque Ramp Input ให้กับระบบผ่านโปรแกรม Simulink โดยเลือกความชันของสัญญาณ Torque Ramp Input (m) = 0.5 เพื่อทำการเก็บสัญญาณดังภาพที่ 30



ภาพที่ 30 แสดงการใช้ Simulink เก็บข้อมูลเพื่อหาแบบจำลองของระบบ

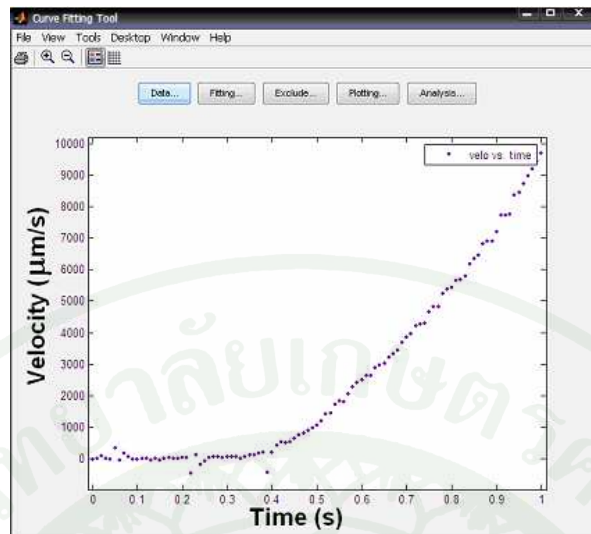
5.1 การวัดค่าแรงเสียดทานสถิต (Static Friction)

จากการวัดค่าแรงเสียดทานสถิต คือวัดค่าแรงเสียดทานก่อนที่มอเตอร์จะเริ่มเคลื่อนที่ หลังจากที่เราตั้งค่าต่างๆ ใน Drive แล้วทำการวัดค่าแรงเสียดทานก่อนที่มอเตอร์เริ่มเคลื่อนที่จะได้ว่า

$$F_s = 0.3145 \text{ [Nm]}$$

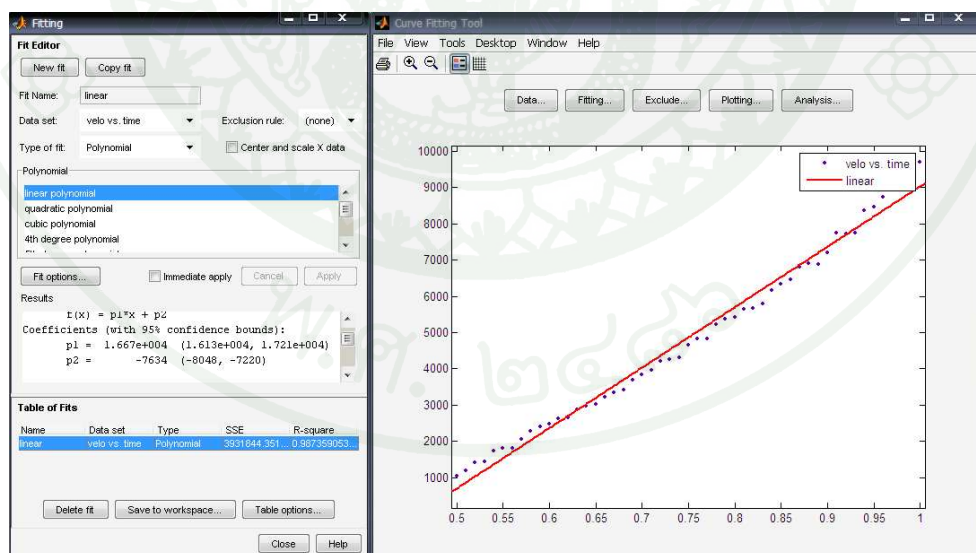
5.2 การวัดค่าแรงเสียดทานคูลอมบ์และแรงเสียดทานหนืด

หลังจากที่ทำการตั้งค่าใน Drive และเขียนโปรแกรม Simulink ตามภาพที่ 31 แล้ว จากนั้นนำสัญญาณที่วัดได้มาทำการพล็อตกราฟ โดยแนวนอนเป็นเวลาในหน่วยวินาทีและแนวตั้งเป็นความเร็วในหน่วยไมโครเมตรต่อวินาทีดังภาพที่ 31



ภาพที่ 31 กราฟความสัมพันธ์ระหว่างความเร็วกับเวลาของมอเตอร์และแกน

จากนั้นนำข้อมูลมาทำการประมาณหาพารามิเตอร์ต่างๆตามวิธีการของ K. Rafael ด้วยโปรแกรม MatLAB โดยในช่วงเวลา 0 - 0.5 วินาทีที่เราถือว่าเป็นช่วงที่เกิดขึ้นเร็วมาก จึงนำเฉพาะส่วนที่เหลือมาทำการพล็อตหาสมการเส้นตรง ดังภาพที่ 32



ภาพที่ 32 แสดงการหาแบบจำลองทางคณิตศาสตร์จากข้อมูลจริง

จากวิธีการหาแบบจำลองแรงเสียดทานตามแบบของ K. Rafael, and L. JeSus. (1999) ที่ได้กล่าวมาแล้วข้างต้น เมื่อทำการเปรียบเทียบกับสัญญาณจริงที่วัดได้จากกราฟจะให้ความชันของกราฟซึ่งแทนด้วยตัวแปร $a = 16,670$ และจุดตัดแกน y ซึ่งแทนด้วยตัวแปร $-b = -763.4$ ดังนั้นจะได้ $b = 763.4$ และจุดที่กราฟมีการเปลี่ยนแปลงความชันอย่างรวดเร็วซึ่งแทนด้วยตัวแปร $c = 0.5$ ดังนั้นเราสามารถแทนค่าหาแรงเสียดทานหนืด (Viscous Friction) ได้จากสมการ

$$\begin{aligned} F_v &= \frac{m}{a} \\ &= \frac{0.5}{16670} \\ &= 2.994 \times 10^{-5} [(Nm)(sec)/deg] \\ &= 0.0017 [(Nm)(sec)/rad] \end{aligned} \quad (41)$$

และสามารถแทนค่าหาแรงเสียดทานคูลอมบ์ (Coulomb Friction) ได้จากสมการ

$$\begin{aligned} F_c &= \frac{b}{a}m \\ &= \frac{763.4}{16670}(0.5) \\ &= 0.0229 [Nm] \end{aligned} \quad (42)$$

และจากสมการแรงเสียดทาน

$$F(v) = F_v v + F_c \operatorname{sgn}(v) + [F_s - F_c] \operatorname{sgn}(v) e^{(v/v_s)^2} \quad (43)$$

เมื่อแทนค่าสัมประสิทธิ์ทุกตัวที่หาได้ลงในสมการที่ (43) จะได้แบบจำลองแรงเสียดทานของระบบดังนี้

$$F(v) = 0.0017v + 0.0229 \operatorname{sgn}(v) + 0.0855 \operatorname{sgn}(v) 0.5^{(v/v_s)^2} \quad (44)$$

สถานที่ทำการวิจัย

ห้อง 2308 ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ และห้อง 6 ชั้น 8 ตึก
วิศวกรรมศาสตร์ 60 ปี คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ระยะเวลาทำการวิจัย

การวิจัยเริ่มตั้งแต่ เดือนตุลาคม 2551 สิ้นสุดเดือนมีนาคม 2553



ผลและวิจารณ์

ผล

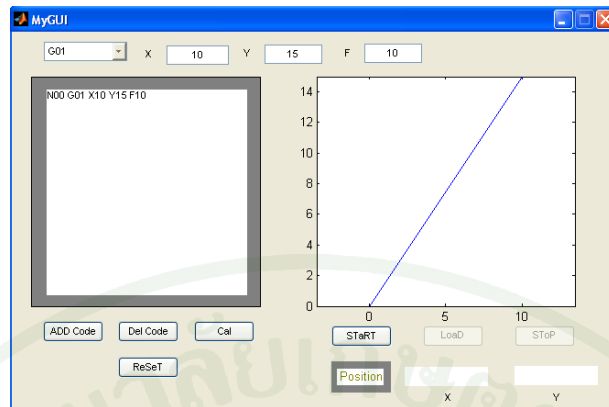
ในการทดสอบในงานวิจัยนี้ได้แบ่งการทดสอบออกเป็นสองส่วนหลักๆคือ การจำลองโดยโปรแกรม Simulink และการทดสอบในการเคลื่อนที่จริงโดยโปรแกรม LabVIEW ผ่าน Fieldpoint โดยในการทดสอบทั้งสองส่วนจะทำการเปรียบเทียบระหว่างการใช้ตัวควบคุมแบบพีไอดี และแบบที่มีการชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี ในการทดสอบนี้เราทำการวัดค่าความผิดพลาดในการเคลื่อนที่อยู่ 2 แบบคือ ค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error) และค่าความผิดพลาดสัมบูรณ์สูงสุด (Maximum Absolute Error)

1. การจำลองโดยโปรแกรม Simulink

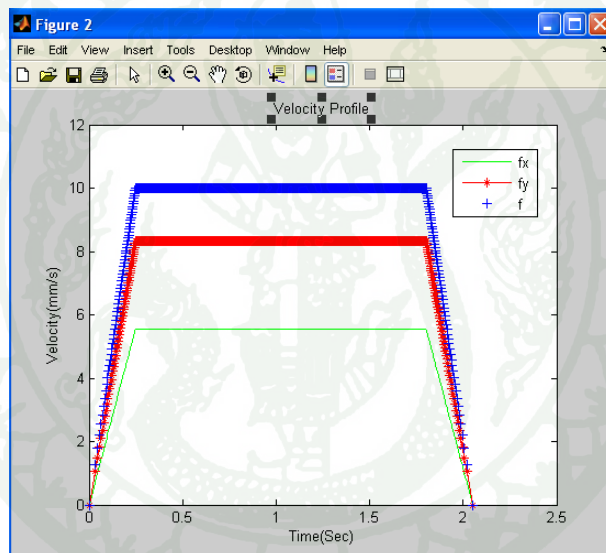
ในการจำลองโดยโปรแกรม Simulink เราจะทำการป้อนค่าการเคลื่อนที่เป็นรูปแบบต่างๆผ่านหน้าต่าง GUI (Graphic User Interface) แล้วส่งค่าความเร็วให้กับโปรแกรม Simulink เพื่อทำการควบคุมด้วยการใช้ตัวควบคุมแบบพีไอดี และแบบที่มีการชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี เพื่อทำการเปรียบเทียบผลต่อไป โดยการจำลองการเคลื่อนที่แบ่งเป็นการเคลื่อนที่รูปต่างๆดังนี้

1.1 รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.

เริ่มต้นจากการป้อนรหัสใน GUI ของโปรแกรม Matlab คือ G01 X10 Y15 F10 เป็นการเคลื่อนที่เป็นเส้นตรงไปที่จุด (10, 15) ด้วยอัตราป้อน 10 mm/s สามารถแสดงได้ดังภาพที่ 33 และสามารถแสดง Velocity Profile ของมอเตอร์แต่ละแกนดังภาพที่ 34 ตามลำดับ

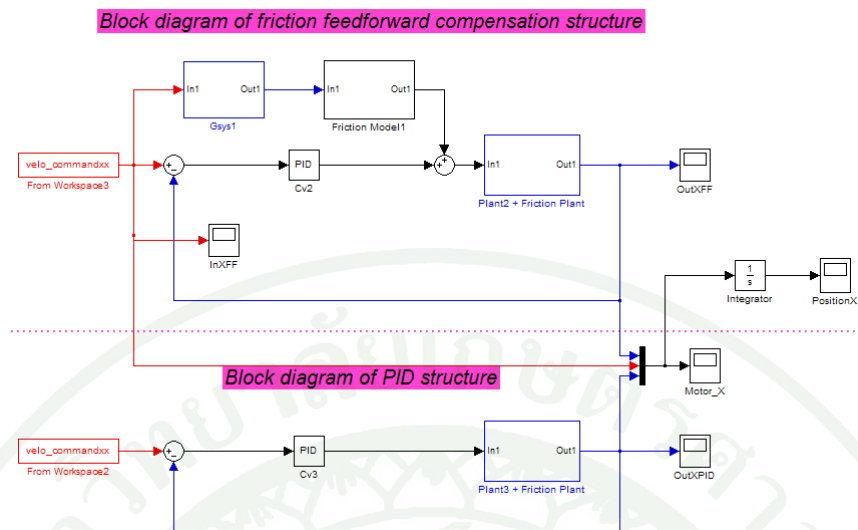


ภาพที่ 33 GUI รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.



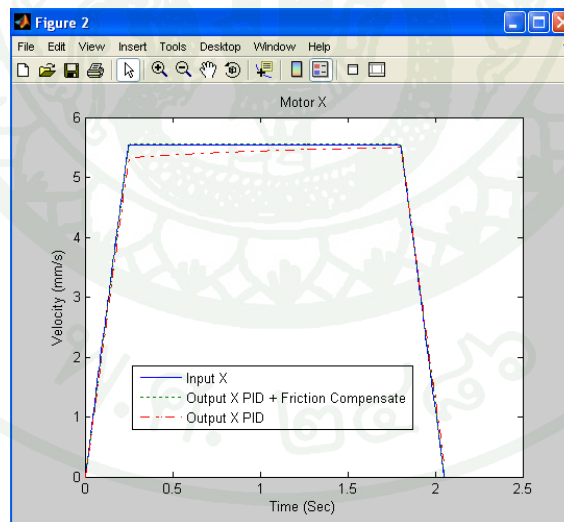
ภาพที่ 34 Velocity Profile การเคลื่อนที่รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.

จากนั้นทำการส่งค่า Velocity Profile ให้กับ Simulink เพื่อทำการควบคุมมอเตอร์ในแต่ละแกน โดยในแต่ละแกนจะทำการควบคุมทั้งสองแบบตามที่ได้กล่าวมาแล้วข้างต้น คือแบบพีไอดีธรรมดา และแบบพีไอดีร่วมกับการชดเชยแรงเสียดทานป้อนไปหน้า ดังภาพที่ 35



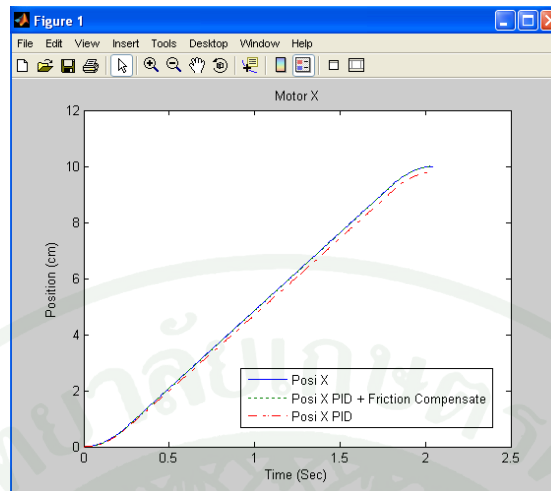
ภาพที่ 35 โปรแกรม Simulink การควบคุมมอเตอร์ในแต่ละวิธี

จากนั้นทำการเปรียบเทียบ Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน X ในแต่ละวิธีการควบคุมได้ดังภาพที่ 36



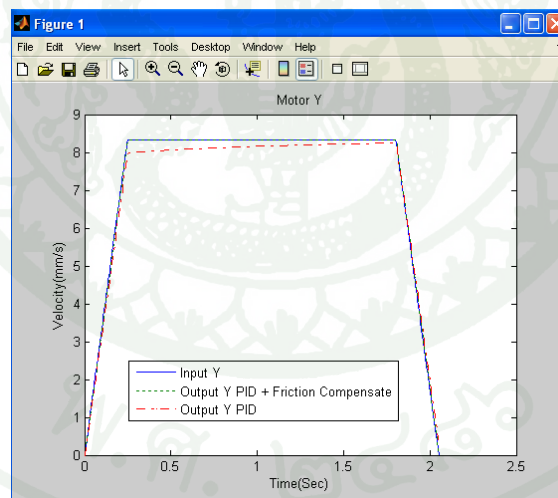
ภาพที่ 36 Velocity Command เทียบกับสัญญาณ Output มอเตอร์แกน X

และสามารถเปรียบเทียบระยะการเคลื่อนที่ในแนวแกน X ได้ดังภาพที่ 37



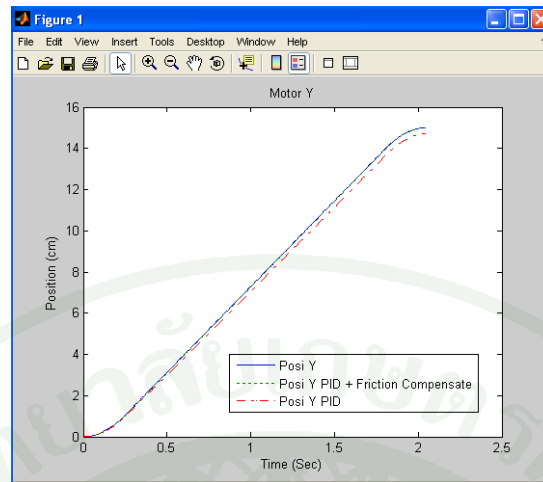
ภาพที่ 37 เปรียบเทียบระยะการเคลื่อนที่ในแนวแกน X

ในลักษณะเดียวกันทำการเปรียบเทียบ Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน Y ในแต่ละวิธีการควบคุมได้ดังภาพที่ 38



ภาพที่ 38 Velocity Command กับสัญญาณ Output มอเตอร์แกน Y

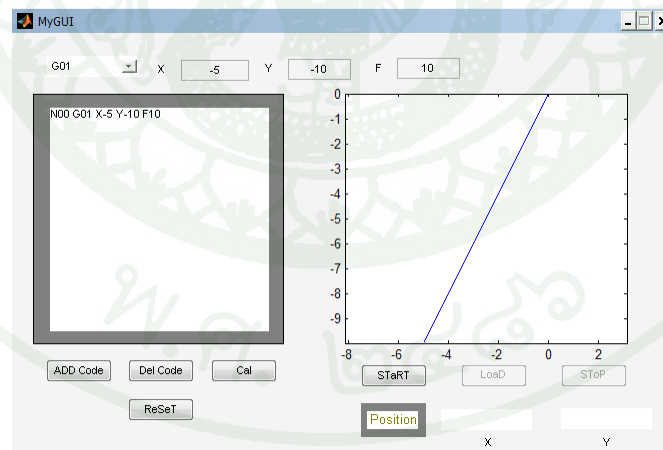
และสามารถเปรียบเทียบระยะการเคลื่อนที่ในแนวแกน Y ได้ดังภาพที่ 39



ภาพที่ 39 เปรียบเทียบระยะการเคลื่อนที่ในแนวแกน Y

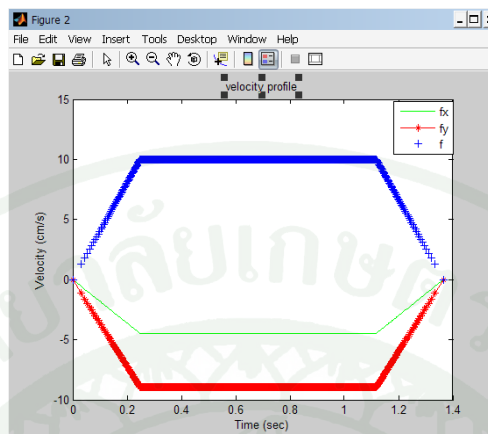
1.2 การเคลื่อนที่รูปเส้นตรงแกน X -5 ซม. แกน Y -10 ซม.

ทำการป้อนรหัสใน GUI ของ Matlab คือ G01 X-5 Y-10 F10 เป็นการเคลื่อนที่เป็นเส้นตรงไปที่จุด (-5, -10) ด้วยอัตราป้อน 10 mm/s สามารถแสดงได้ดังภาพที่ 40



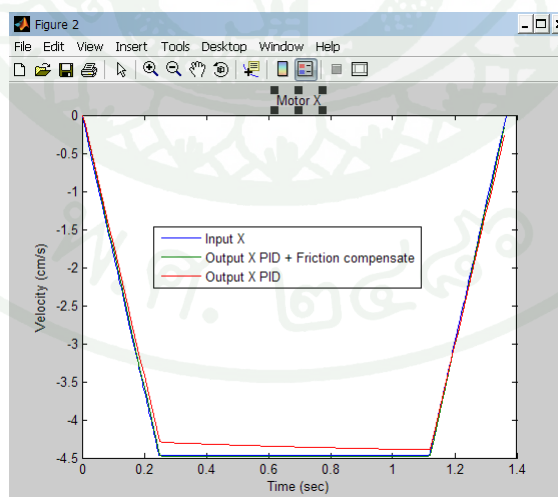
ภาพที่ 40 GUI รูปเส้นตรงแกน X -5 ซม. แกน Y -10 ซม.

และสามารถแสดง Velocity Profile ดังภาพที่ 41



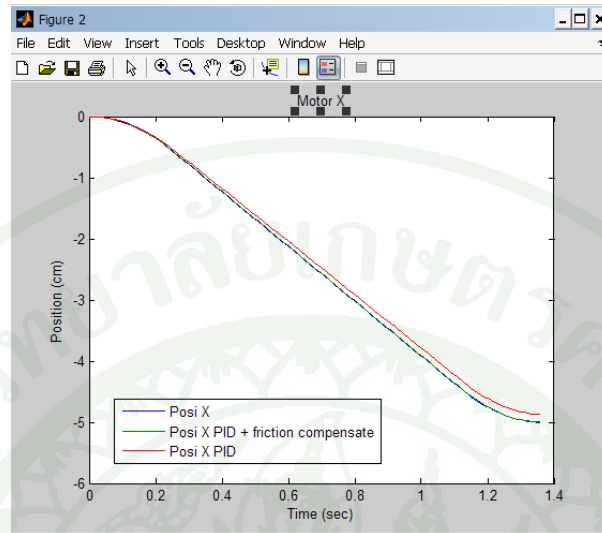
ภาพที่ 41 Velocity Profile การเคลื่อนที่แกน X -5 ซม. แกน Y -10 ซม.

จากนั้นทำการส่งค่า Velocity Profile ให้กับ Simulink เพื่อทำการควบคุมมอเตอร์ในแต่ละแกน โดยในแต่ละแกนจะทำการควบคุมทั้งสองแบบตามที่ได้กล่าวมาแล้วข้างต้น แล้วทำการเปรียบเทียบ Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน X ในแต่ละวิธีการควบคุมได้ดังภาพที่ 42



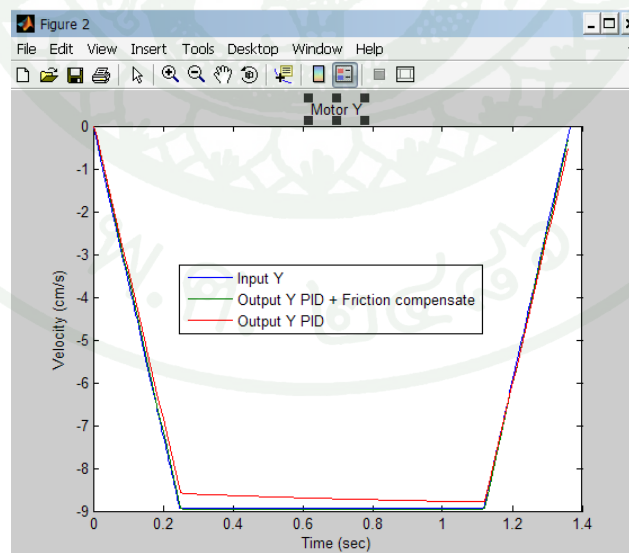
ภาพที่ 42 Velocity Command กับสัญญาณ Output มอเตอร์แกน X

และสามารถเปรียบเทียบระยะการเคลื่อนที่ในแนວแกน X ได้ดังภาพที่ 43



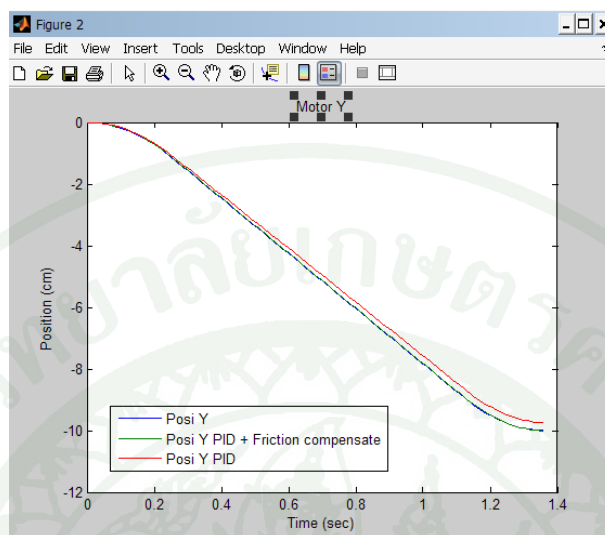
ภาพที่ 43 เปรียบเทียบระยะการเคลื่อนที่ในแนວแกน X

ในลักษณะเดียวกันทำการเปรียบเทียบ Velocity Command กับสัญญาณ Output มอเตอร์ แกน Y ในแต่ละวิธีการควบคุมได้ดังภาพที่ 44



ภาพที่ 44 Velocity Command กับสัญญาณ Output มอเตอร์แกน Y

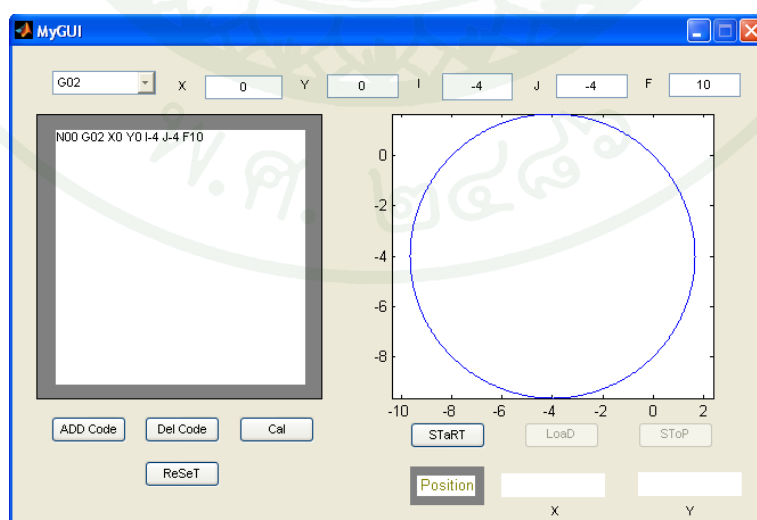
และสามารถเปรียบเทียบระยะการเคลื่อนที่ในแนวแกน Y ได้ดังภาพที่ 45



ภาพที่ 45 เปรียบเทียบระยะการเคลื่อนที่ในแนวแกน Y

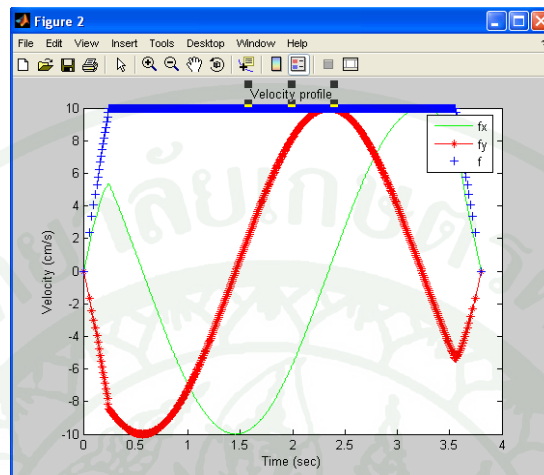
1.3 การเคลื่อนที่รูปวงกลมรัศมี 5.6569 ซม.

ทำการป้อนรหัสจีใน GUI ของ Matlab คือ G02 X0 Y0 I-4 J-4 F10 เป็นการเคลื่อนที่เป็นวงกลมรัศมี 5.6569 ซม. ด้วยอัตราป้อน 10 mm/s สามารถแสดงได้ดังภาพที่ 46



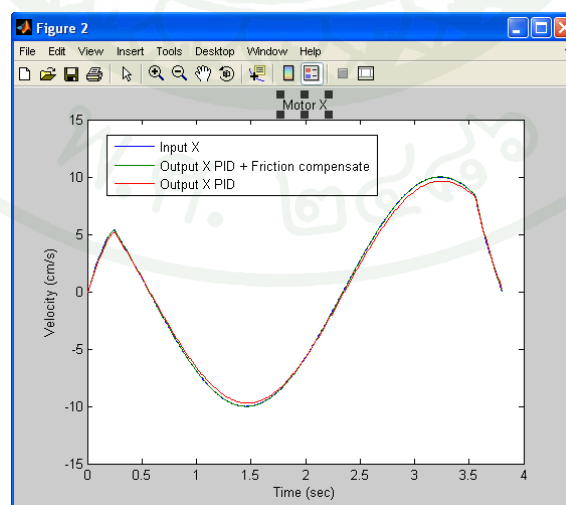
ภาพที่ 46 GUI รูปวงกลมรัศมี 5.6569 ซม.

และสามารถแสดง Velocity Profile ดังภาพที่ 47



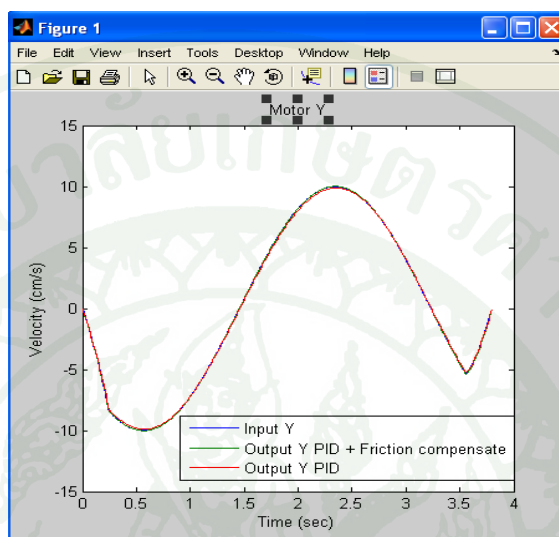
ภาพที่ 47 Velocity Profile การเคลื่อนรูปวงกลมรัศมี 5.6569 ซม.

จากนั้นทำการส่งค่า Velocity Profile ให้กับ Simulink เพื่อทำการควบคุมมอเตอร์ในแต่ละแกน โดยในแต่ละแกนจะทำการควบคุมทั้งสองแบบตามที่ได้กล่าวมาแล้วข้างต้น แล้วทำการเปรียบเทียบ Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน X ในแต่ละวิธีการควบคุมได้ดังภาพที่ 48



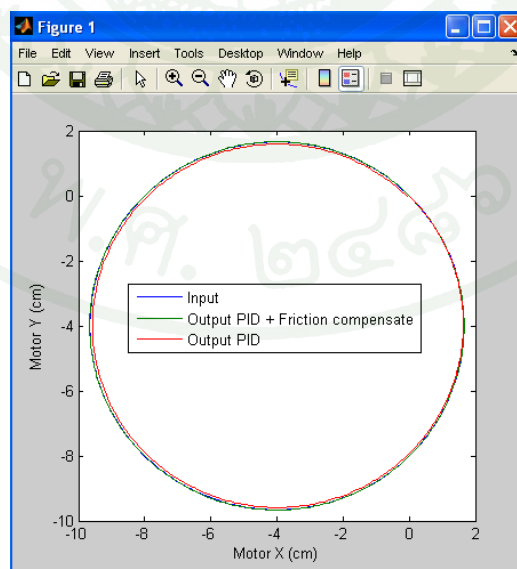
ภาพที่ 48 เปรียบเทียบ Velocity Command กับสัญญาณ Output มอเตอร์แกน X

ในลักษณะเดียวกันเราทำการเปรียบเทียบสัญญาณ Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน Y ในแต่ละวิธีการควบคุมได้ดังภาพที่ 49



ภาพที่ 49 Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน Y

สามารถเปรียบเทียบเส้นทางการเคลื่อนที่เป็นวงกลมของมอเตอร์ทั้งสองแกนได้ดังภาพที่ 50



ภาพที่ 50 เปรียบเทียบเส้นทางการเคลื่อนที่เป็นวงกลมของมอเตอร์ทั้งสองแกน

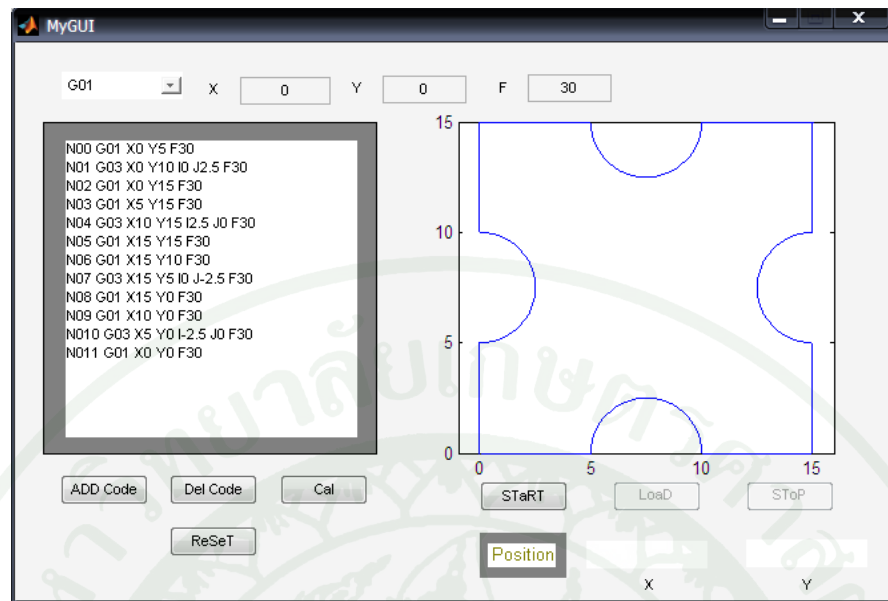
1.4 รูปประยุกต์อื่นๆ

ทำการป้อนรหัสจีใน GUI ของ Matlab โดยป้อนรหัสจิดังตารางที่ 5

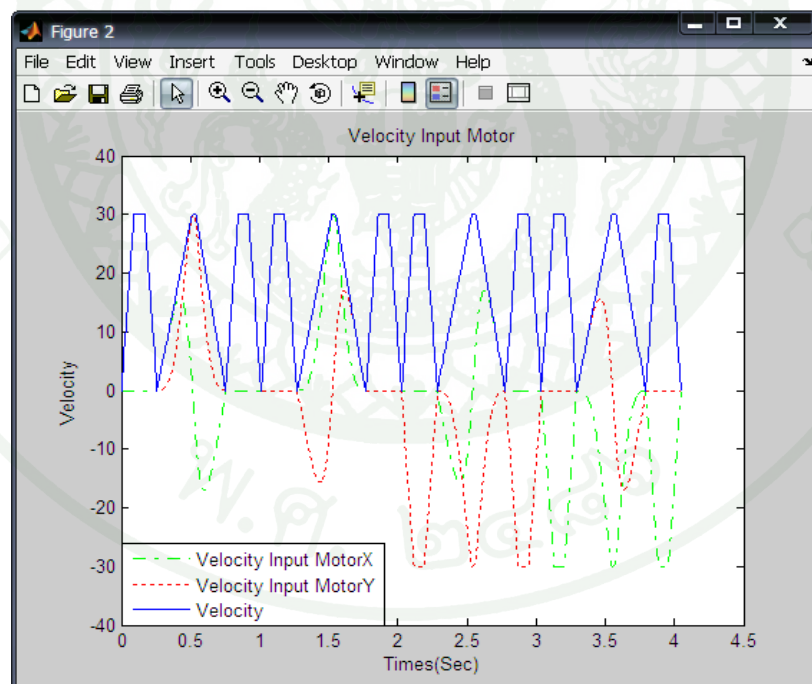
ตารางที่ 5 ค่า G-Code ที่ป้อนให้ GUI สำหรับรูปประยุกต์อื่นๆ

G-Code	X-Axis (cm)	Y-Axis (cm)	I	J	Feed Rate (mm/s)
G01	0	5	-	-	30
G03	0	10	0	2.5	30
G01	0	15	-	-	30
G01	5	15	-	-	30
G03	10	15	2.5	0	30
G01	15	15	-	-	30
G01	15	10	-	-	30
G03	15	5	0	-2.5	30
G01	15	0	-	-	30
G01	10	0	-	-	30
G03	5	0	-2.5	0	30
G01	0	0	-	-	30

เมื่อได้รับค่า G-Code แล้ว โปรแกรมจะแสดงรหัส G-Code ที่ได้รับและแสดงผลการทดสอบเส้นทางการเคลื่อนที่ผ่านทาง GUI ดังภาพที่ 51 และแสดง Velocity Profile ดังภาพที่ 52

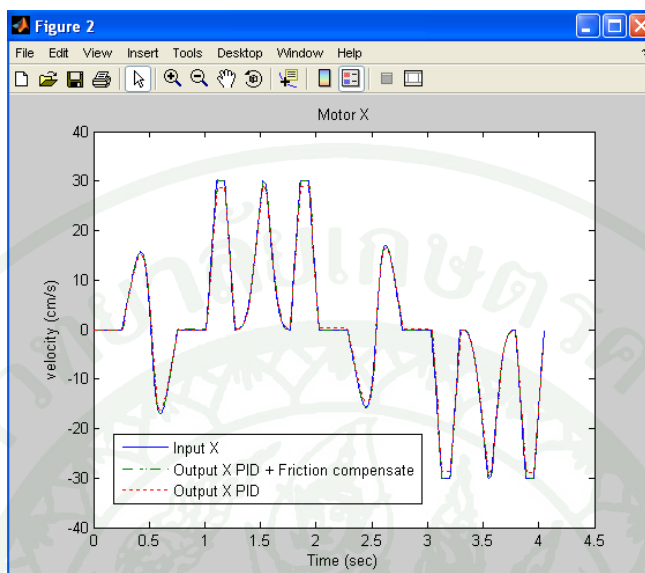


ภาพที่ 51 GUI แสดงรูปประกอบอื่นๆ



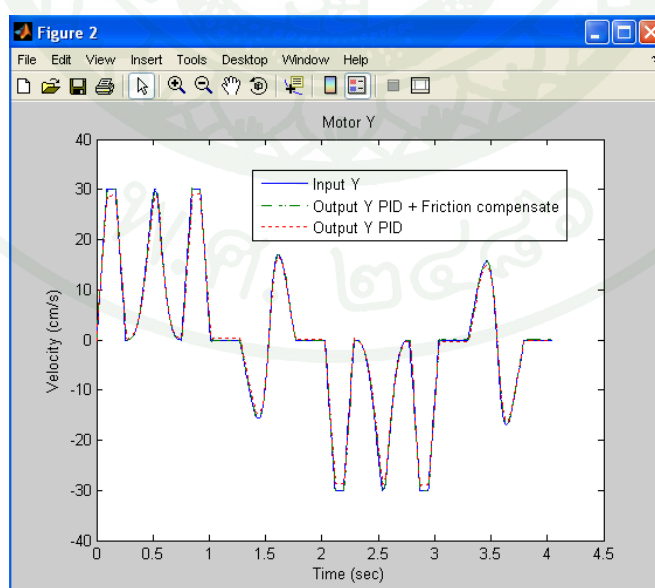
ภาพที่ 52 Velocity Profile การเคลื่อนที่รูปประกอบอื่นๆ

และสามารถเปรียบเทียบระยะการเคลื่อนที่ในแนวแกน X ได้ดังภาพที่ 53



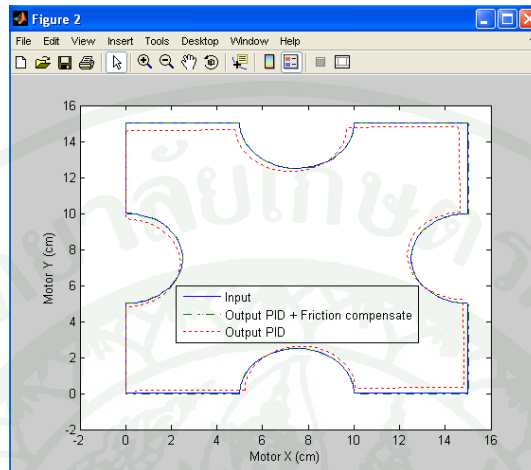
ภาพที่ 53 Velocity Command กับสัญญาณ Output มอเตอร์แกน X

ในลักษณะเดียวกันเราทำการเปรียบเทียบสัญญาณ Velocity Command เทียบกับสัญญาณ Output ของมอเตอร์แกน Y ในแต่ละวิธีการควบคุมได้ดังภาพที่ 54



ภาพที่ 54 Velocity Command กับสัญญาณ Output มอเตอร์แกน Y

สามารถเปรียบเทียบเส้นทางการเคลื่อนที่รูปประยุกต์ของมอเตอร์ทั้งสองแกนได้ดังภาพที่ 55



ภาพที่ 55 เส้นทางการเคลื่อนที่รูปประยุกต์ของมอเตอร์ทั้งสองแกน

2. ค่าความผิดพลาดทางตัวเลขของการทดลอง

เราสามารถสรุปค่าความผิดพลาดในการติดตามการเคลื่อนที่โดยแสดงค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error) ด้วยสมการ

$$MAE = \frac{\sum_{k=1}^N |P_a(t(k)) - P_s(t(k))|}{N} \quad (45)$$

สามารถแสดงค่าความผิดพลาดสัมบูรณ์สูงสุด (Maximum Absolute Error) ด้วยสมการ

$$MaxAE = \max |P_a(t(k)) - P_s(t(k))| \quad (46)$$

จากสมการที่ (45) เราสามารถแสดงค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error) ของเครื่องซีเอ็นซี 3 แกนด้วย Simulink ดังตารางที่ 6

ตารางที่ 6 ค่าเฉลี่ยความผิดพลาดสัมบูรณ์ของเครื่องซีเอ็นซี 3 แกนด้วย Simulink

การทดลองที่	รูปแบบการเคลื่อนที่	ค่าความผิดพลาดของแกน X (cm)		ค่าความผิดพลาดของแกน Y (cm)	
		PID	PID + Friction	PID	PID + Friction
			Feedforward		Feedforward
2.1	รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	0.0325	0.0019	0.0487	0.0028
2.2	รูปเส้นตรงแกน X -5 ซม. แกน Y -10 ซม.	0.0209	0.0017	0.0419	0.0034
2.3	รูปวงกลมรัศมี 5.6569 ซม.	0.0959	0.0094	0.1113	0.0096
2.4	รูปประยุกต์อื่นๆ	0.0562	0.0073	0.0715	0.0076

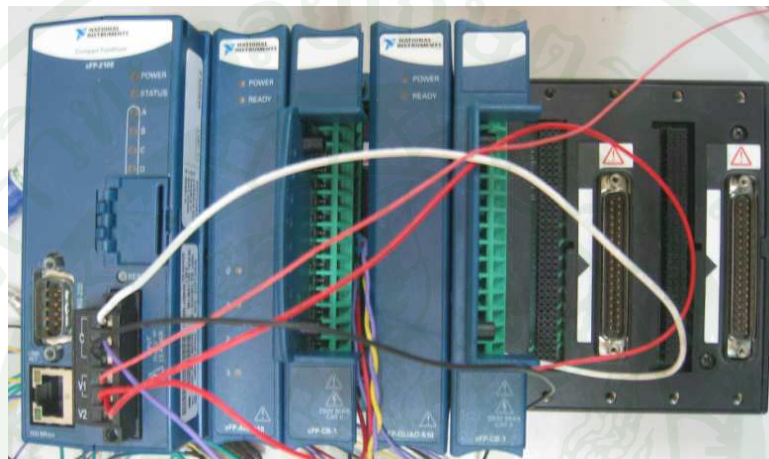
จากสมการที่ (45) สามารถแสดงค่าความผิดพลาดสัมบูรณ์สูงสุด (Maximum Absolute Error) ของเครื่องซีเอ็นซี 3 แกนด้วย Simulink ดังตารางที่ 7

ตารางที่ 7 ค่าความผิดพลาดสัมบูรณ์สูงสุด (Maximum Absolute Error)

การทดลองที่	รูปแบบการเคลื่อนที่	ค่าความผิดพลาดของแกน X (cm)		ค่าความผิดพลาดของแกน Y (cm)	
		PID	PID + Friction	PID	PID + Friction
			Feedforward		Feedforward
2.1	รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	0.2199	0.0130	0.3298	0.0195
2.2	รูปเส้นตรงแกน X -5 ซม. แกน Y -10 ซม.	0.1391	0.0105	0.2781	0.0210
2.3	รูปวงกลมรัศมี 5.6569 ซม.	0.2701	0.0227	0.2586	0.0226
2.4	รูปประยุกต์อื่นๆ	0.4729	0.0792	0.4786	0.0796

3. การทดสอบโดยโปรแกรม LabVIEW ผ่าน Fieldpoint

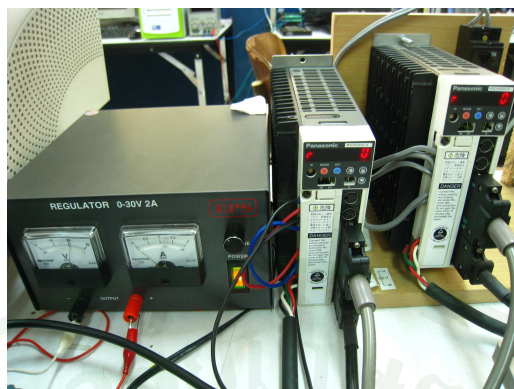
หลังจากที่ทำการจำลองโดยโปรแกรม Simulink แล้วเราจะทำการทดสอบจริงโดยโปรแกรม LabVIEW ผ่าน Fieldpoint โดยทำการทดสอบเหมือนกับการจำลองโดยโปรแกรม Simulink สามารถแสดงภาพการทดสอบจริงได้ดังภาพที่ 56 ถึงภาพที่ 59



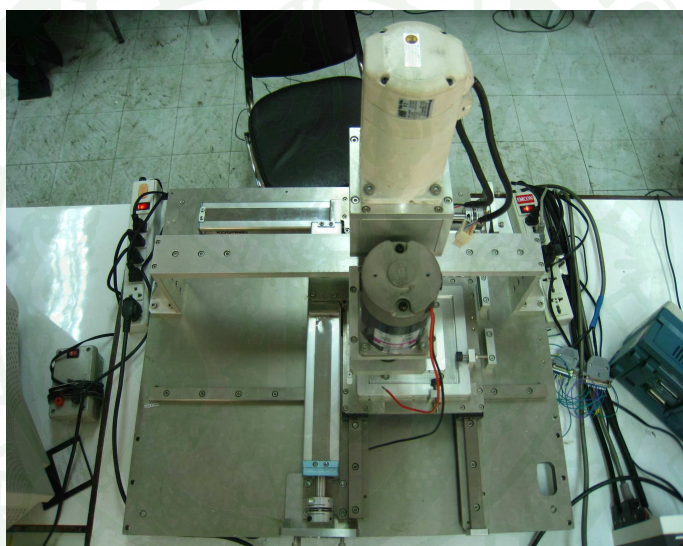
ภาพที่ 56 Field Point ขณะทำการทดสอบ



ภาพที่ 57 การต่อวงจรแปลงเป็น ± 10 โวลต์เป็น 4-20 มิลลิแอมป์



ภาพที่ 58 Power Supply และ Drive ขณะทำการทดสอบ

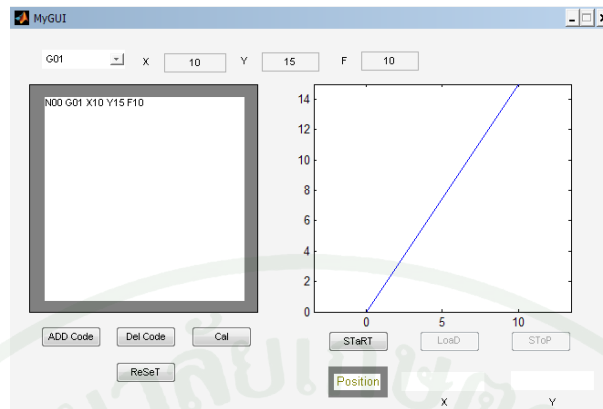


ภาพที่ 59 เครื่องซีเอ็นซี 3 แกนที่ใช้ในการทดสอบ

โดยผลการทดสอบการเคลื่อนที่จริงของการเคลื่อนที่รูปแบบต่างๆ ได้ผลดังนี้

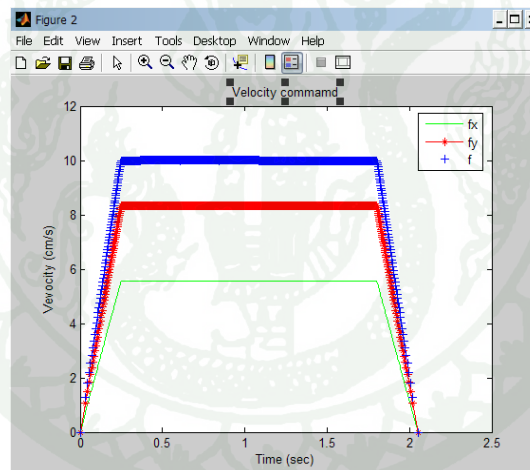
3.1 รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.

ทำการป้อนรหัสใน GUI ของ Matlab คือ G01 X10 Y15 F10 เป็นการเคลื่อนที่เป็นเส้นตรงไปที่จุด (10, 15) ด้วยอัตราป้อน 10 mm/s สามารถแสดงได้ดังภาพที่ 60



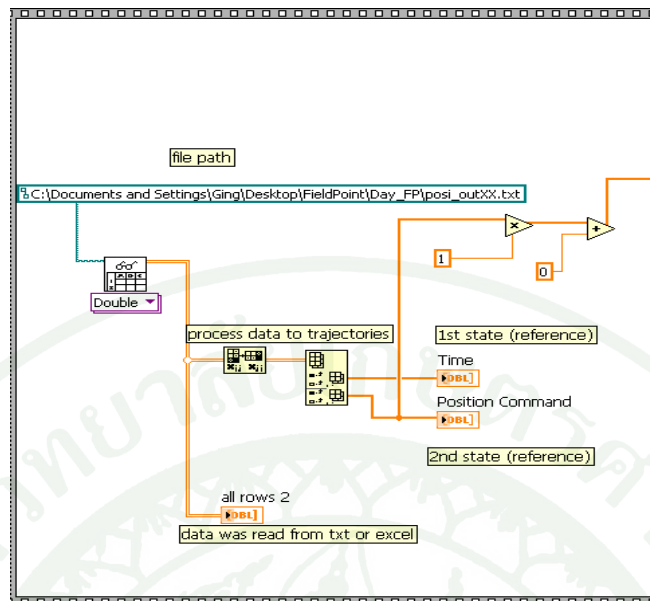
ภาพที่ 60 GUI รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.

และสามารถแสดง Velocity Profile ดังภาพที่ 61

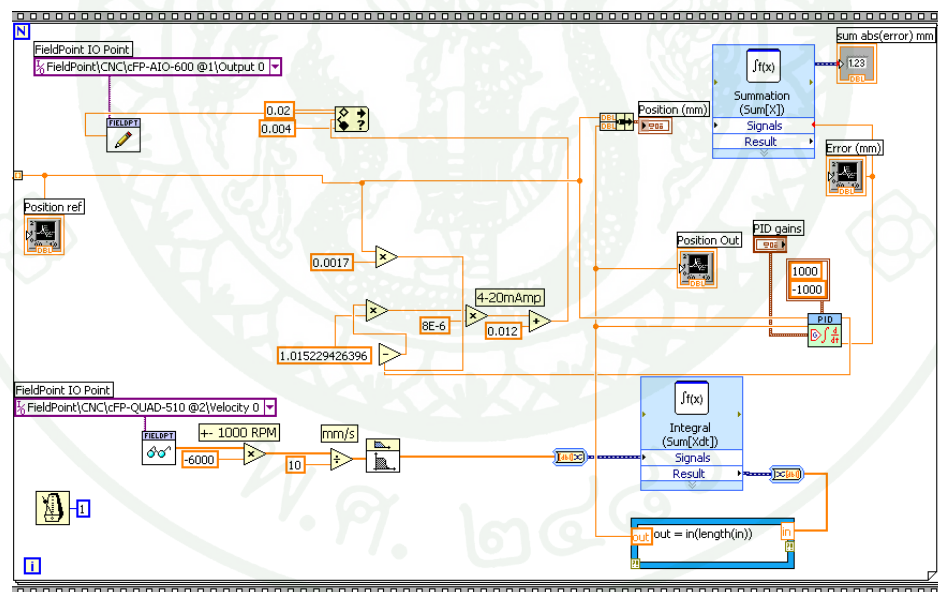


ภาพที่ 61 Velocity Profile การเคลื่อนที่รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.

จากนั้นนำ Velocity Profile ที่ได้เข้ามาในส่วนของโปรแกรม LabVIEW ซึ่งแสดง Block Diagram ดังภาพที่ 62 และ ภาพที่ 63 เป็นการควบคุมแบบพีไอดีของมอเตอร์แกน X โดยเลือกค่า $K_p = 18$ $K_i = 2$ $K_d = 0$

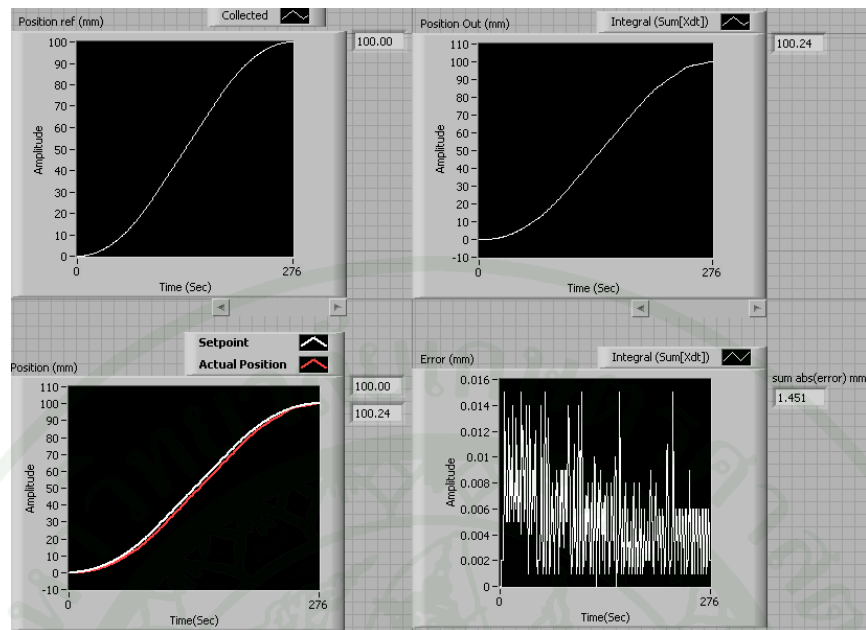


ภาพที่ 62 Block Diagram นำข้อมูลเข้ามาทำการควบคุมแบบพีไอดีแกน X



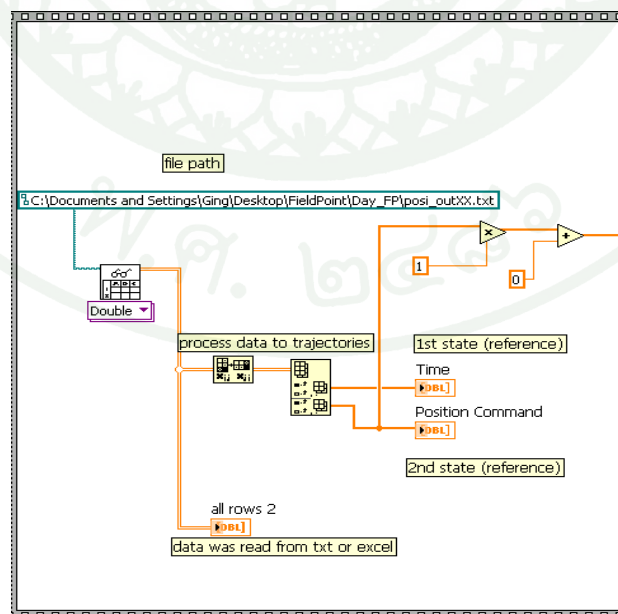
ภาพที่ 63 Block Diagram การควบคุมแบบพีไอดีแกน X

และ Front Panel มอเตอร์แกน X แบบพีไอดีสามารถแสดงได้ดังภาพที่ 64

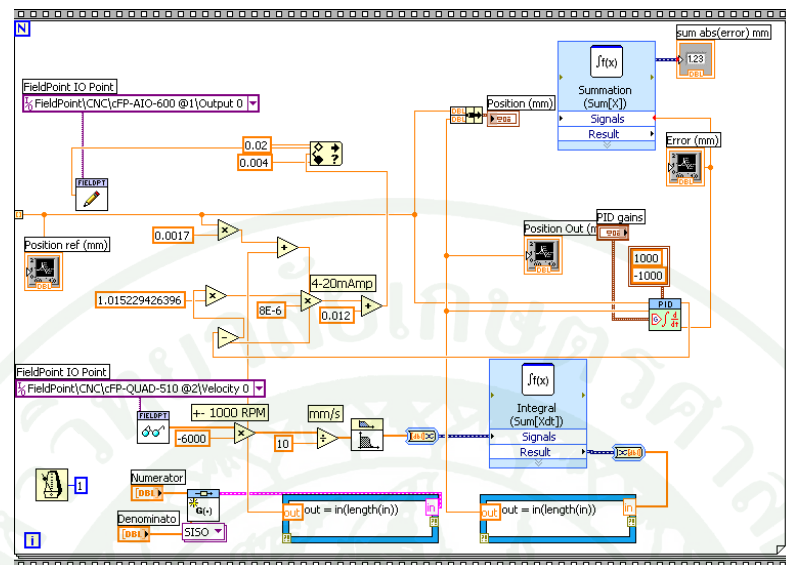


ภาพที่ 64 Front Panel การควบคุมมอเตอร์แกน X แบบพีไอดี

ต่อมาเป็นการควบคุมมอเตอร์ในแนวแกน X โดยประยุกต์การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีโดยเลือกค่า $K_p = 18$ $K_i = 2$ $K_d = 0$ ซึ่ง Block Diagram การควบคุมมอเตอร์แกน X สามารถแสดงได้ดังภาพที่ 65 และภาพที่ 66

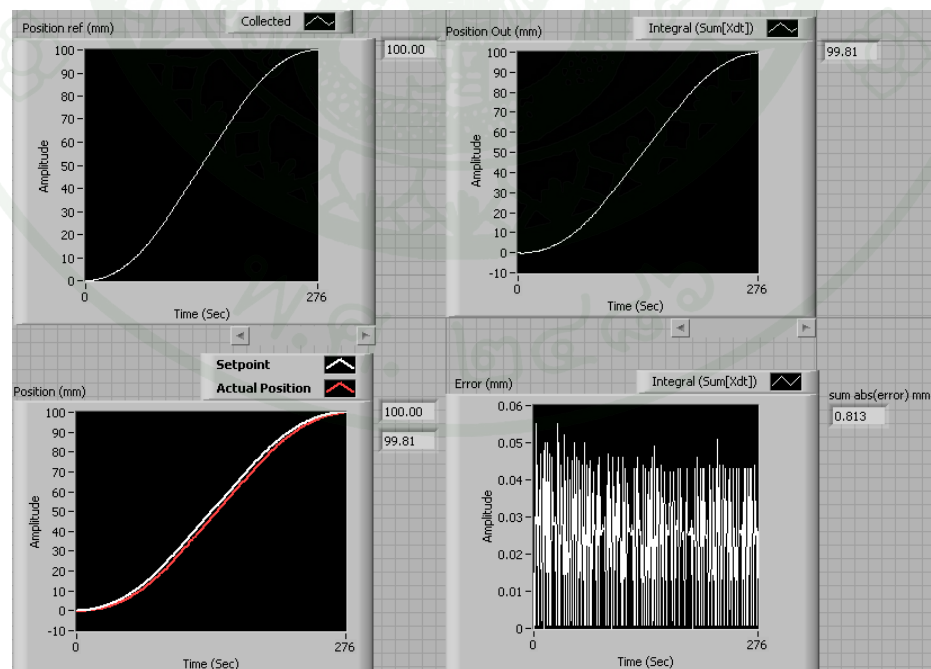


ภาพที่ 65 การนำข้อมูลเข้ามาควบคุมในแนวแกน X



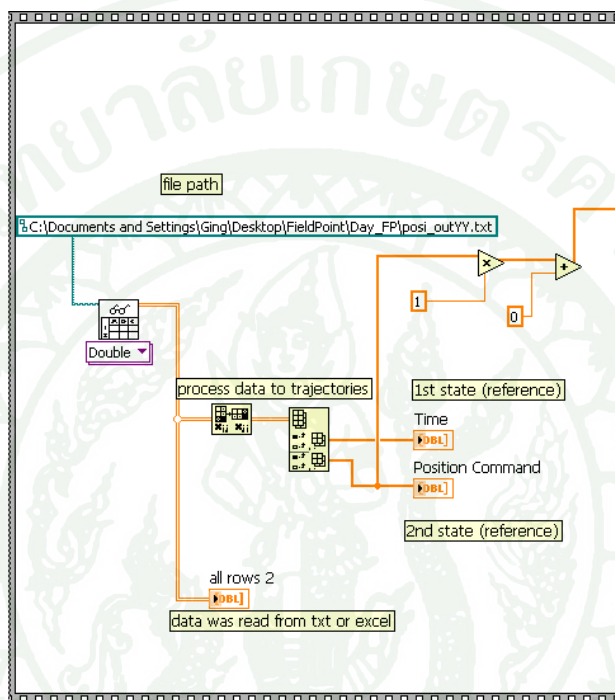
ภาพที่ 66 Block Diagram การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีแกน X

ในส่วนของ Front Panel ที่ทำการควบคุมมอเตอร์แกน X โดยใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีสามารถแสดงได้ดังภาพที่ 67

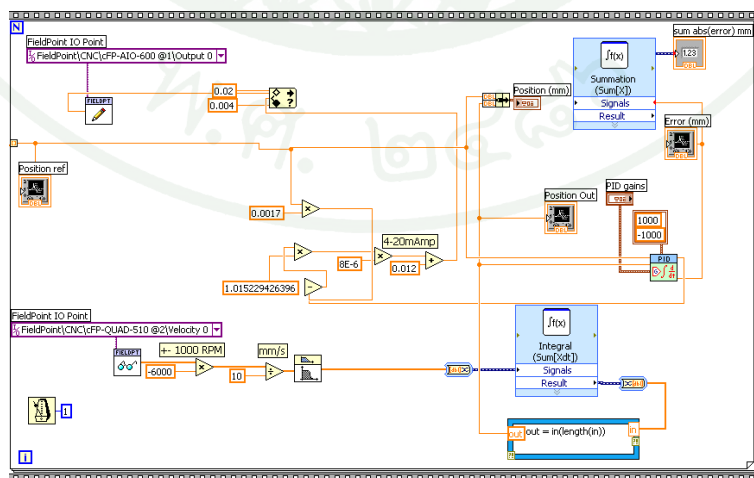


ภาพที่ 67 Front Panel การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีแกน X

ต่อมาเป็นการควบคุมมอเตอร์ในแกน Y โดยนำ Velocity Profile ที่ได้เข้ามาในส่วน
ของโปรแกรม LabVIEW ที่แสดงในส่วนของ Block Diagram ซึ่งในภาพที่ 68 และภาพที่ 69
เป็นภาพแสดง Block Diagram การควบคุมแบบพีไอดีของมอเตอร์แกน Y โดยเลือกค่า $K_p = 18$
 $K_i = 2$ $K_d = 0$

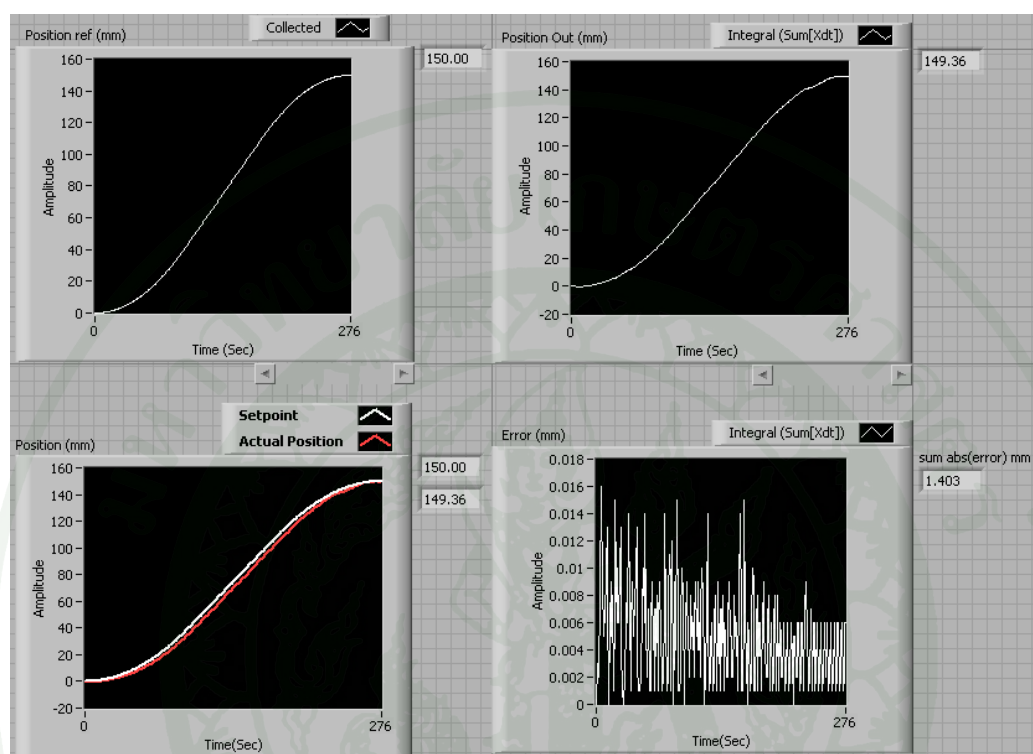


ภาพที่ 68 Block Diagram นำข้อมูลเข้ามาทำการควบคุมแบบพีไอดีแกน Y



ภาพที่ 69 Block Diagram การควบคุมมอเตอร์แกน Y แบบพีไอดี

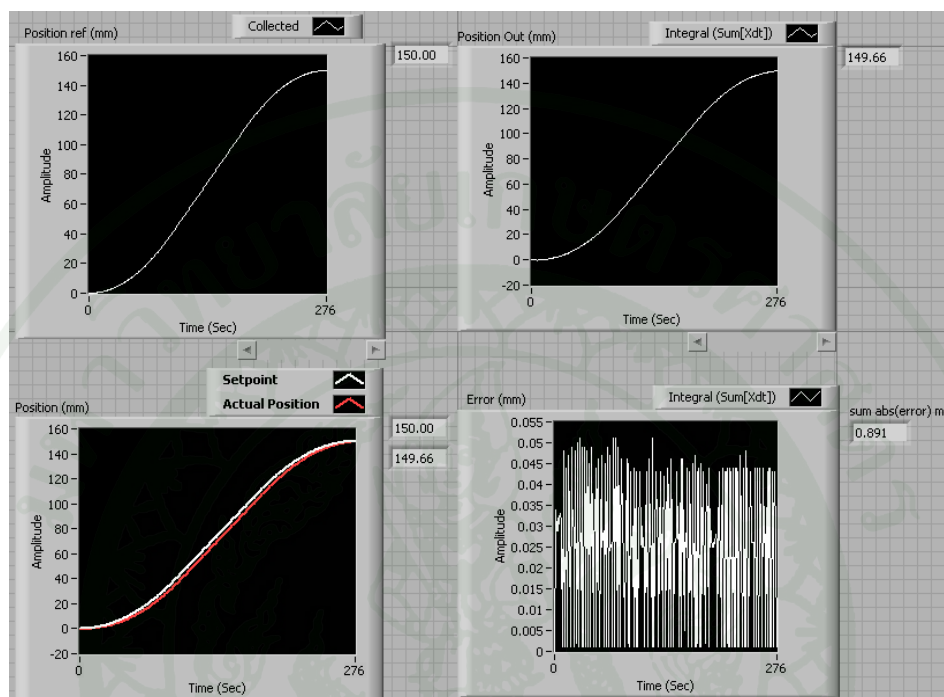
ในส่วนของ Front Panel ที่เราทำการควบคุมมอเตอร์แกน Y แบบพีไอดี สามารถแสดง
ได้ดังภาพที่ 70



ภาพที่ 70 Front Panel การควบคุมมอเตอร์แกน Y แบบพีไอดี

ต่อมาเป็นการควบคุมมอเตอร์ในแนวแกน Y โดยประยุกต์ใช้การชดเชยแรงเสียดทานป้อน
ไปหน้าร่วมกับตัวควบคุมพีไอดี โดยเลือกค่า $K_p = 18$ $K_i = 2$ $K_d = 0$ ซึ่งในส่วนของ Block
Diagram การควบคุมมอเตอร์แกน Y สามารถแสดงได้ดังภาพที่ 71 และภาพที่ 72

ในส่วนของ Front Panel ที่เราทำการควบคุมมอเตอร์แกน Y โดยใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีสามารถแสดงได้ดังภาพที่ 73



ภาพที่ 73 Front Panel การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีแกน Y

3.2 รูปเส้นตรงแกน X-5 ซม. แกน Y -10 ซม.

สำหรับการทดลองที่ 3.2 รูปเส้นตรงแกน X-5 ซม. แกน Y -10 ซม. ทำการป้อนค่าให้กับ GUI ค่าเดียวกับการทดลองที่ 1.2 จากนั้นนำ Velocity Profile ที่ได้เข้ามาทำการทดสอบเหมือนการทดลองที่ 3.1

3.3 รูปวงกลมรัศมี 5.6569 ซม.

สำหรับการทดลองที่ 3.3 รูปวงกลมรัศมี 5.6569 ซม. ทำการป้อนค่าให้กับ GUI ค่าเดียวกับการทดลองที่ 1.3 จากนั้นนำ Velocity Profile ที่ได้เข้ามาทำการทดสอบเหมือนการทดลองที่ 3.1

3.4 รูปประยุกต์อื่นๆ

และการทดลองที่ 3.4 รูปประยุกต์อื่นๆ ทำการป้อนค่าให้กับ GUI ค่าเดียวกับการทดลองที่ 1.4 จากนั้นนำ Velocity Profile ที่ได้เข้ามาทำการทดสอบเหมือนการทดลองที่ 3.1

จากการทดลองที่ 3.1 ถึง 3.4 และจากสมการที่ (45) เราสามารถแสดงค่าเฉลี่ยความผิดพลาดสัมบูรณ์ (Mean Absolute Error) ของการทดสอบโดยโปรแกรม LabVIEW ผ่าน Fieldpoint ดังตารางที่ 8

ตารางที่ 8 ค่าเฉลี่ยความผิดพลาดสัมบูรณ์ของการทดสอบโปรแกรม LabVIEW ผ่าน Fieldpoint

การทดลองที่	รูปแบบการเคลื่อนที่	ค่าความผิดพลาดของแกน X (cm)		ค่าความผิดพลาดของแกน Y (cm)	
		PID	PID + Friction Feedforward	PID	PID + Friction Feedforward
3.1	รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	0.1021	0.0791	0.1287	0.1028
3.2	รูปเส้นตรงแกน X -5 ซม. แกน Y -10 ซม.	0.0509	0.0172	0.2419	0.1034
3.3	รูปวงกลมรัศมี 5.6569 ซม.	0.1959	0.1094	0.2113	0.1296
3.4	รูปประยุกต์อื่นๆ	0.2562	0.2073	0.1715	0.1072

จากสมการที่ (46) สามารถแสดงค่าความผิดพลาดสัมบูรณ์สูงสุด (Maximum Absolute Error) ของการทดสอบโดยโปรแกรม LabVIEW ผ่าน Fieldpoint ดังตารางที่ 9

ตารางที่ 9 ค่าความผิดพลาดสัมบูรณ์สูงสุดของการทดสอบโปรแกรม LabVIEW ผ่าน Fieldpoint

การทดลอง ที่	รูปแบบการเคลื่อนที่	ค่าความผิดพลาดของแกน X (cm)		ค่าความผิดพลาดของแกน Y (cm)	
		PID	PID + Friction	PID	PID + Friction
			Feedforward		Feedforward
3.1	รูปเส้นตรงแกน X 10 ซม. แกน Y 15 ซม.	0.2014	0.1790	0.3870	0.3028
3.2	รูปเส้นตรงแกน X -5 ซม. แกน Y -10 ซม.	0.2509	0.1172	0.2940	0.2043
3.3	รูปวงกลมรัศมี 5.6569 ซม.	0.3929	0.2094	0.3211	0.2964
3.4	รูปประยุกต์อื่นๆ	0.5562	0.3073	0.3717	0.2078

วิจารณ์

จากการแสดงค่าความผิดพลาดในการจำลองการเคลื่อนที่ตามตารางที่ 6 และ 7 และการแสดงค่าความผิดพลาดในการเคลื่อนที่จริงตามตารางที่ 8 และ 9 จากการทดสอบทั้งสองแบบพบว่าการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี ให้ความแม่นยำในการเคลื่อนที่มากกว่าการควบคุมแบบพีไอดีเพียงอย่างเดียว ตัวอย่างเช่น ในรูปประยุกต์อื่นๆ จากค่าความผิดพลาดในการจำลองการเคลื่อนที่ตามตารางที่ 6 ให้ค่าความผิดพลาดสัมบูรณ์สูงสุดจากการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีเฉลี่ยทั้งสองแกนเท่ากับ 0.00754 ซม. ให้ค่าความผิดพลาดสัมบูรณ์สูงสุดจากการควบคุมแบบพีไอดีเพียงอย่างเดียวเฉลี่ยทั้งสองแกนเท่ากับ 0.06385 ซม. จากค่าความผิดพลาดในการเคลื่อนที่จริงตามตารางที่ 8 ให้ค่าความผิดพลาดสัมบูรณ์สูงสุดจากการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดีเฉลี่ยทั้งสองแกนเท่ากับ 0.15725 ซม. ให้ค่าความผิดพลาดสัมบูรณ์สูงสุดจากการควบคุมแบบพีไอดีเพียงอย่างเดียวเฉลี่ยทั้งสองแกนเท่ากับ 0.21385 ซม. ซึ่งเห็นว่าค่าความผิดพลาดจากการ

จำลองการเคลื่อนที่และจากการทดสอบการเคลื่อนที่จริงมีลักษณะใกล้เคียงกัน และค่าความ
ผิดพลาดต่างกันเล็กน้อยเนื่องจากแรงเสียดทานที่เกิดขึ้นกับระบบในแต่ละช่วงมีค่าไม่สม่ำเสมอ



สรุปและข้อเสนอแนะ

สรุป

จากการศึกษาการศึกษาผลกระทบจากแรงเสียดทานที่เกิดขึ้นกับเครื่อง CNC 3 แกน และการหาแบบจำลองแรงเสียดทานที่เกิดขึ้น พร้อมทั้งการทดสอบการควบคุมการเคลื่อนที่ของเครื่อง CNC 3 แกน โดยการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี เพื่อเปรียบเทียบกับการควบคุมแบบพีไอดีพบว่า

จากการจำลองการเคลื่อนที่โดยใช้โปรแกรม Simulink ให้ค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี โดยเฉลี่ยในทุกการทดสอบของแกน X มีค่า 0.005075 ซม. ค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการใช้ตัวควบคุมพีไอดีโดยเฉลี่ยในทุกการทดสอบของแกน X มีค่า 0.051375 ซม. ค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี โดยเฉลี่ยในทุกการทดสอบของแกน Y มีค่า 0.00585 ซม. ค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการใช้ตัวควบคุมพีไอดีโดยเฉลี่ยในทุกการทดสอบของแกน Y มีค่า 0.06835 ซม.

จากการทดลองจริงโดยโปรแกรม LabVIEW ผ่าน Fieldpoint พบว่าค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี โดยเฉลี่ยในทุกการทดสอบของแกน X มีค่า 0.10325 ซม. ค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการใช้ตัวควบคุมพีไอดี โดยเฉลี่ยในทุกการทดสอบของแกน X มีค่า 0.151275 ซม. ค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการประยุกต์ใช้การชดเชยแรงเสียดทานป้อนไปหน้าร่วมกับตัวควบคุมพีไอดี โดยเฉลี่ยในทุกการทดสอบของแกน Y มีค่า 0.11075 ซม. ค่าเฉลี่ยความผิดพลาดสัมบูรณ์จากการใช้ตัวควบคุมพีไอดี โดยเฉลี่ยในทุกการทดสอบของแกน Y มีค่า 0.18835 ซม.

จากผลการจำลองการเคลื่อนที่โดยใช้โปรแกรม Simulink และจากผลการทดลองจริงโดยโปรแกรม LabVIEW ผ่าน Fieldpoint ทั้งหมดพบว่าพบว่าการประยุกต์ใช้การควบคุมแบบพีไอดี ร่วมกับการชดเชยแรงเสียดทานป้อนไปหน้า สามารถเคลื่อนที่ได้แม่นยำและมีประสิทธิภาพว่าการควบคุมแบบพีไอดีทั่วไป จากการทดลองพบว่าค่าความผิดพลาดจากการจำลองการเคลื่อนที่

และจากการทดสอบการเคลื่อนที่จริงมีลักษณะใกล้เคียงกัน และค่าความผิดพลาดต่างกันบ้าง เนื่องจากแรงเสียดทานที่เกิดขึ้นกับระบบในแต่ละช่วงมีค่าไม่สม่ำเสมอ

ข้อเสนอแนะ

จากผลการทดลองที่ได้ให้ผลที่น่าพอใจระดับหนึ่ง แต่ก็ยังพบปัญหาในส่วนของ Software และ Hardware ดังนี้

1. ในช่วงที่มีการเคลื่อนที่ด้วยความเร็วต่ำจะมีค่าความผิดพลาดมากขึ้น เนื่องจากที่ความเร็วต่ำมอเตอร์จะเกิดการสั่น ส่งผลทำให้มีความผิดพลาดเกิดขึ้นในการเคลื่อนที่มากขึ้น
2. ค่า K_p , K_i , K_d ในสมการ PID Controller นั้นเราประมาณค่ามาจากสมการเอกลักษณ์ทั่วไปของมอเตอร์ซึ่งอาจจะมีผลทำให้มีค่าความผิดพลาดในการเคลื่อนที่ ถ้าเราใช้หาเอกลักษณ์ของระบบมาช่วยในการหาสมการเอกลักษณ์ที่แท้จริงของมอเตอร์ร่วมกับชุดขับที่มีความไม่เป็นเชิงเส้น คาดว่าจะส่งผลทำให้ค่าความผิดพลาดน้อยลงได้

เอกสารและสิ่งอ้างอิง

ชาติ ตระการกุล. 2548. เทคโนโลยีซีเอ็นซี. ครั้งที่ 12. ศ.ศ.ท., กรุงเทพฯ.

ทรงชัย แซ่ตั้ง. 2549. การออกแบบชุดควบคุม CNC 3 แกน. วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยเกษตรศาสตร์.

วโรดม ตู้จินดา. 2547. การออกแบบสร้างและรีโทรฟิตชุดควบคุมเครื่องซีเอ็นซี. ครั้งที่ 1. ศ.ศ.ท., กรุงเทพฯ.

Acosta Calderon, Rosales Pena Alfaro, E.M. Gan and J.Q. Hu 2003. **Trajectory Generation and Tracking of A 5-DOF Robotic Arm.**

Aoki, K., N. Iwazawa, T.Tsujisawa, Y. Sakaguchi and K. Nakawatase, 1994. **High Accuracy Path Tracking by Multi-axis coordination in CNC machines.**

Canudas de Wit C., H. Olsson, K. J. Htrom and P. Lischinsky. 1995. **A New Model for Control of Systems with Friction.**

Craig Simal and J. Inc John. 1955. **Introduction to Robotics, Mechanic And Control.** Addison-Wesley.

Erkorkmaz K. and Y. Altintas. 2001. High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. **International Journal of Machine Tools and Manufacture: 1323-1345.**

Erkorkmaz K. and Y. Altintas. 2001. High speed CNC system design. Part II: Modeling and Identification of feed drives. **International Journal of Machine Tools and Manufacture: 1487-1509.**

Frew Eric W. and Rock Stephen. 2003. **Trajectory Generation for Constant Velocity Target Motion Estimation Using Monocular Vision.**

Gon Jen Wang and Tzong Jing Lee. 1999. Neural-network cross-coupled control system with application on circular tracking of linear motor X-Y table. **IJCNN '99**: 2194 – 2199.

Jason T. Teeter, MO-yuen Chow and J. James Brickley. 1996. A Novel Fuzzy Friction Compensation Approach to Improve the Performance of a DC Motor Control System. **IEEE Transactions On Industrial Electronics.**

Jiancheng Liu Yamazaki, K. and Y. Yokoyama. 1998. Dynamic gain motion control with multi-axis trajectory monitoring for machine tool systems, pp. 316 – 321. **AMC '98-Coimbra.**

Kok Kiong Tan, T. H. Lee, Sunan N. Huang and Xi Jiang. 2001. Friction Modeling and Adaptive Compensation Using a Relay Feedback Approach. **IEEE Transactions On Industrial Electronics.**

Kotaro Nakaoka and Tomonori Sato. 2006. Feedforward Controller for Continuous Path Control of CNC Machine Tools. **International Journal of simulation**: 39-46.

Lewis, Christopher L. and Maciejewski, Anthony A. 1990Trajectory generation for cooperating robots, pp. 300 – 303. **IEEE International Conference.**

Li Shuxun, Yao Yu, and Wang Zicai. 1999. Friction Compensation Based on Improved Performance Model Reference Adaptive Control. **International Conference on Advanced Intelligent Mechatronics.**

Olsson H., K. L. Åström, C. Canudas de Wit, M. Gäfvert and P. Lischinsky. 1998. Friction Models and Friction Compensation. **European Journal of Control**: 176-195.

Pierre Rouchon. 2001. Motion planing, equivalence, infinite dimensional systems. **International Journals Applied Math Computer Science**: 165-188.

Rafael K. and L. JeŚus. 1999. Determination of Viscous and Coulomb Friction by Using Velocity Response to Torque Ramp Inputs, pp. 1740-1745. **IEEE International Conference on Robotics & Automation**.

Syh-Shiuh, Yeh Pau-Lo Hsu. 2000. A new approach to bi-axial cross-coupled control. **IEEE Proceedings of Control Theory Application**: 168 - 173.

Tsai M.-C., I.-F. Chiu and M.-Y. Cheng. 2004. Design and Implement of Command and Friction Feedforward Control for CNC motion controllers. **IEEE Proceedings.-Control Theory Appl**: 13-20.



การแปลงไฟล์ MatLAB เป็นไฟล์ .EXE

ในการแปลงไฟล์ MatLAB ทุกไฟล์ที่มีอยู่ในโปรแกรม MatLAB เป็นไฟล์ .EXE นั้น มีวัตถุประสงค์เพื่อให้โปรแกรมที่เราเขียนจาก MatLAB สามารถที่จะรันได้กับเครื่องคอมพิวเตอร์ทุกเครื่อง แม้ว่าเครื่องคอมพิวเตอร์เครื่องนั้นจะไม่มีโปรแกรม MatLAB ก็ตามซึ่งถือว่ามีประโยชน์มาก ในการแปลง m-file หรือไฟล์รูปแบบอื่นๆที่มีในโปรแกรม MatLAB เป็นไฟล์ .EXE สามารถทำได้ดังนี้

1. เปิดโปรแกรม MatLAB
2. เปิดไฟล์ที่เราต้องการแปลงให้อยู่ใน Current directory ของโปรแกรม MatLAB ยกตัวอย่างเช่น hopGraph.m
3. ในหน้า Command windows ให้พิมพ์คำสั่ง >>mcc -m ตามด้วยชื่อไฟล์ที่เราต้องการแปลงเช่น >> mcc -m hopGraph.m
4. ถ้าโปรแกรมฟ้องเป็นตัวหนังสือสีแดงไม่ว่าจะเป็นข้อความใดก็ตาม ให้กด Enter เพื่อให้โปรแกรมสามารถทำงานต่อไปได้

เท่านี้ก็สามารถแปลงไฟล์ในโปรแกรม MatLAB เป็นไฟล์ .EXE ได้แล้ว หลังจากที่เราแปลงไฟล์เสร็จแล้ว โปรแกรมจะทำการบันทึกไฟล์ .EXE ไว้ที่โฟลเดอร์เดียวกับไฟล์เดิมที่เป็นโปรแกรม MatLAB ดังภาพที่ 74



ภาพที่ 74 แสดงตัวอย่างไฟล์ .EXE ที่ได้

ถ้ารันในเครื่องที่ไม่มี MatLAB ต้อง Setup ไฟล์ MCRInstaller.exe เนื่องจากบางเครื่องยังไม่ได้ติดตั้งซึ่งต้อง Copy จาก C:\MATLAB7\toolbox\compiler\deploy\win32\ MCRInstaller.exe

รายละเอียด AC Servo motor และ AC Servo Driver

Panasonic AC Servo motor

Input	3 phase AC	116 V
	4.3 V	
Rate Output		0.75kW
Rate freq.		200 Hz
Rate Rev.		3000 r/min

Panasonic AC Servo Driver

	Input	Output
Voltage	200-230 V	116V
Phase	3 phase	3 phase
F.L.C.	4.0 A	4.3 A
Freq.	50/60 Hz	0 – 300 Hz
Power		750 W

ประวัติการศึกษาและการทำงาน

ชื่อ	นายรพีพงศ์ รัตนวรหิรัญกุล
เกิดวันที่	28 พฤศจิกายน 2527
สถานที่เกิด	อำเภอเขาวง จังหวัดกาฬสินธุ์
ประวัติการศึกษา	วศ.บ. (ไฟฟ้า) มหาวิทยาลัยเกษตรศาสตร์
ตำแหน่งปัจจุบัน	นิสิต
สถานที่ทำงานปัจจุบัน	คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
ผลงานดีเด่นและ/หรือรางวัลทางวิชาการ	งานวิจัยเรื่องการชดเชยแรงเสียดทานป้อนไปหน้าเพื่อเพิ่มประสิทธิภาพการทำงานของเครื่องซีเอ็นซี
ทุนการศึกษาที่ได้รับ	งานวิจัยนี้ได้รับทุนสนับสนุนจากสถาบันบัณฑิตวิทยาศาสตร์และเทคโนโลยีไทย (TGIST) สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ สัญญารับทุนเลขที่ TGIST 01-51-074 และทุนสถาบันวิจัยและพัฒนาแห่งมหาวิทยาลัยเกษตรศาสตร์ (KURDI) รหัสทุนวิจัย ว-ท(ค) 52.51 และทุนผู้ช่วยสอนจากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์