

การศึกษาระบบควบคุมแบบพีไอดี โดยอาศัยชุดทดลอง Roller on Beam
ผ่านโปรแกรม LabVIEW
Study of PID Control system applied to Roller on Beam apparatus
via LabVIEW

นายศิริกร พิธิฐวรกุล
นายชนสิทธิ์ สินวรพิพัฒน์
นายธีรวุฒิ ลิงห์แก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

การศึกษาระบบควบคุมแบบพีไอดี โดยอาศัยชุดทดลอง Roller on Beam

ผ่านโปรแกรม LabVIEW

Study of PID Control system applied to Roller on Beam apparatus
via LabVIEW

นายศิริกร พิสิฐวรกุล

นายชนสิทธิ์ สินวรพิพัฒน์

นายธีรภูมิ สิงห์แก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

Study of PID Control system applied to
Roller on Beam apparatus via LabVIEW

KIRAKORN PISITWORAKUL
CHANASIT SINWORAPIPAT
THEERAWUT SINGKAEW

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHANICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2017

ปริญญาโทปีการศึกษา 2560

สาขาวิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาระบบควบคุมแบบพีไอดี โดยอาศัยชุดทดลอง Roller on Beam ผ่านโปรแกรม LabVIEW

Study of PID Control system applied to Roller on Beam apparatus via LabVIEW

ผู้จัดทำ

1. นายศิริกร พิสิฐวรกุล 57010142
2. นายชนสิทธิ์ สีนวรพัฒน์ 57010239
3. นายธีรวุฒิ สิงห์แก้ว 57010638



อาจารย์ที่ปรึกษา

(ดร.ณัฐวุฒิ เรืองตระกูล)

การศึกษาระบบควบคุมแบบพีไอดี โดยอาศัยชุดทดลอง Roller on Beam ผ่านโปรแกรม LabVIEW

นายศิริกร พิสิฐวรกุล 57010142

นายชนสิทธิ์ สินวรพิพัฒน์ 57010239

นายธีรวุฒิ สิงห์แก้ว 57010638

ดร.ณัฐวุฒิ เรืองตระกูล อาจารย์ที่ปรึกษา

ปีการศึกษา 2560

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาระบบควบคุมแบบพีไอดี โดยอาศัยชุดทดลอง Roller on Beam apparatus ซึ่งเป็นชุดทดลองควบคุมแบบอัตโนมัติที่ควบคุมโดยโปรแกรม LabVIEW ซึ่งเป็นโปรแกรมสำเร็จรูป โดยชุดทดลองนี้ถูกสร้างขึ้นเพื่อศึกษาควบคุมลูกกิ้งให้อยู่ตามตำแหน่งบนคานที่เรากำหนด โดยอาศัยอุปกรณ์ที่สำคัญคือ ultrasonic sensor และ Servo motor โดย ultrasonic sensor จะทำการรับค่าสัญญาณตำแหน่งของลูกกิ้ง และ Servo motor จะทำการปรับเปลี่ยนตำแหน่งของลูกกิ้งที่ไหลบนคาน และสิ่งสำคัญในการควบคุมลูกกิ้งที่ไหลบนคาน เราจะต้องทำการออกแบบตัวควบคุมพีไอดี (PID ; Proportional Integral Derivative) เพื่อควบคุมตำแหน่งลูกกิ้ง โดยการควบคุมตำแหน่งของลูกกิ้งนั้นจะมีการปรับค่าพารามิเตอร์ที่สำคัญที่เกี่ยวข้องได้แก่ K_p , T_I และ T_D เพื่อให้ลูกกิ้งไหลบนคานไหลไปยังตำแหน่งที่ต้องการ ซึ่งจากการทดลอง พบว่าผลลัพธ์ที่ดีที่สุด(ค่าความผิดพลาดที่สถานะคงตัว และเวลาในการเข้าสู่สถานะคงตัว มีค่าน้อยที่สุด) เมื่อกำหนดค่า $K_p = 20$, $T_I = 0.286$ และ $T_D = 0.003$ และการใช้โปรแกรม LabVIEW นั้น จะเป็นตัวช่วยในการออกแบบระบบที่ใช้ควบคุมชุดทดลอง Roller on Beam

Study of PID Control system applied to Roller on Beam apparatus via LabVIEW

KIRAKORN	PISITWORAKUL	57010142
CHANASIT	SINWORAPIPAT	57010239
THEERAWUT	SINGKAEW	57010638
DR.NATTHAWUT	RUANGTRAKOON	ADVISOR
YEAR 2017		

ABSTRACT

This thesis is to study of PID control applied to Roller on Beam apparatus which is the automation control by using package software, LabVIEW. The Roller on beam apparatus which can control the Roller position was constructed. In this study, we have to control the Roller at position that we set. Important equipment are ultrasonic sensor and servo motor. Ultrasonic sensor is used to receive the signal which is related to Roller position. Servo motor is used to change the Roller position on beam. Moreover, in order to control the Roller position, we have to design PID (Proportional Integral Derivative) control system to control the Roller position at designed point by adjusting the important gain parameters which include K_P , T_I and T_D . From experiment, we found that the best results (minimum in steady state error and setting time) were obtained when $K_P = 20$, $T_I = 0.286$ and $T_D = 0.003$. Moreover, LabVIEW software is the helpful tool to design the control system for Roller on beam apparatus.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีจากการร่วมมือและความช่วยเหลือจากหลายฝ่ายโดยเฉพาะอย่างยิ่ง อาจารย์ณัฐวุฒิ เรืองตระกูล อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ซึ่งท่านได้ให้ความใส่ใจ ความเมตตา และความรักแก่คณะผู้จัดทำเป็นอย่างมาก รวมทั้งให้คำปรึกษา คำแนะนำที่เปี่ยมไปด้วยความรู้และประสบการณ์ ซึ่งทำให้พวกเราคณะผู้จัดทำซาบซึ้งมาก จึงขอขอบพระคุณมา ณ ที่นี้

ขอขอบคุณภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับความรู้ แรงบันดาลใจ และเอื้อเฟื้อสถานที่ในการศึกษา รวมไปถึงเพื่อนๆจากภาควิชาวิศวกรรมเครื่องกลสำหรับกำลังใจและคำปรึกษาต่างๆมากมายที่นำมาใช้ประยุกต์ในการสร้างปริญญาานิพนธ์ฉบับนี้ จึงขอขอบพระคุณมา ณ ที่นี้

ท้ายที่สุดนี้ขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดาและมารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนทั้งหมดมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสูงสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายคิรากร พิสิฐวรกุล

นายชนสิทธิ์ สินวรพิพัฒน์

นายธีรวุฒิ สิงห์แก้ว

สารบัญ

	หน้า
บทคัดย่อ	i
กิตติกรรมประกาศ	iii
สารบัญ.....	iv
สารบัญรูปภาพ	vi
สารบัญตาราง.....	viii
บทที่ 1.....	1
บทนำ	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของงาน	2
1.3 ขอบเขตงานวิจัย.....	2
1.4 สมมติฐานของการศึกษา.....	2
1.5 วิธีดำเนินงานวิจัย	2
1.6 ประโยชน์ที่ได้รับจากโครงการ.....	2
บทที่ 2.....	4
งานวิจัยและทฤษฎีที่เกี่ยวข้อง.....	4
2.1 หัวข้อที่ศึกษา.....	4
2.1.1 งานวิจัยที่เกี่ยวข้อง	4
2.2 ตัวควบคุม.....	5
2.3 ระบบควบคุมพีไอดี	5
2.3.1 ค่าตัวแปรที่ใช้ของระบบควบคุมพีไอดี	6
2.3.2 การทดสอบหาค่าเกณฑ์ที่เหมาะสมของตัวควบคุมแบบ PID.....	11
2.4 ความรู้เบื้องต้นเกี่ยวกับโปรแกรม LabVIEW	12
2.4.1 พื้นฐานและการใช้งานโปรแกรม LabVIEW	13
2.4.2 การสร้างไฟล์ออกแบบวงจร Front Panel และ Block Diagram	14
บทที่ 3.....	17
วิธีการดำเนินงานและการออกแบบชุดทดลอง	17
3.1 หลักการและแนวคิด.....	17

3.2 หลักการทำงานของชุดทดลอง	17
3.3 การออกแบบชุดทดลอง	18
3.4 ความสัมพันธ์ของแต่ละอุปกรณ์	19
3.5 การนำอุปกรณ์มาใช้ในชุดทดลอง	20
3.5.1 เครื่องมือวัดระยะทาง (Displacement sensors)	20
3.5.2 Servo motor	22
3.5.3 Arduino board.....	23
3.6 ขั้นตอนการสร้างชุดทดลอง	25
บทที่ 4.....	27
วิธีการทดลองและผลลัพธ์.....	27
4.1 การสอบเทียบ ultrasonic sensor	27
4.2 สร้างบล็อกไดอะแกรมผ่านโปรแกรม LabVIEW	29
4.3 การทดลอง.....	30
บทที่ 5.....	39
สรุปผลการทดลองและข้อเสนอแนะ	39
5.1 สรุปผลการทดลอง	39
5.2 ปัญหาจากการศึกษา.....	39
5.3 ข้อเสนอแนะ	40
บรรณานุกรม	41
ภาคผนวก ก	42
ภาคผนวก ข	48

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 Free body diagram ของชุดทดลอง.....	4
รูปที่ 2.2 ภาพแสดงนิยามของระบบควบคุม.....	5
รูปที่ 2.3 ตัวควบคุมแบบสัดส่วน.....	6
รูปที่ 2.4 ตัวควบคุมแบบปริพันธ์.....	7
รูปที่ 2.5 ตัวควบคุมแบบอนุพันธ์.....	8
รูปที่ 2.6 ตัวควบคุมแบบ PI.....	9
รูปที่ 2.7 ตัวควบคุมแบบ PD.....	10
รูปที่ 2.8 ตัวควบคุมแบบ PID.....	10
รูปที่ 2.9 ผลตอบสนองจากการนำตัวควบคุมแบบ PID แบบต่าง ๆ ไปใช้กับระบบควบคุม.....	10
รูปที่ 2.10 ลักษณะของ Front Panel.....	12
รูปที่ 2.11 ลักษณะของ Block Diagram.....	13
รูปที่ 2.12 ลักษณะของ Icon และ Connector ในโปรแกรม LabVIEW.....	13
รูปที่ 2.13 หน้าต่างสร้างไฟล์งานของโปรแกรม LabVIEW 2014.....	14
รูปที่ 2.14 หน้าต่าง Front Panel ของโปรแกรม LabVIEW 2014.....	15
รูปที่ 2.15 หน้าต่าง Block Diagram ของโปรแกรม LabVIEW 2017.....	16
รูปที่ 3.1 Block diagram ของระบบ.....	17
รูปที่ 3.2 Block diagram ในโปรแกรม LabVIEW.....	18
รูปที่ 3.3 หลักการตรวจจับและวัดระยะห่างระหว่างวัตถุด้วยคลื่นเสียง.....	20
รูปที่ 3.4 การต่อใช้งานโมดูลultrasonic sensor HC-SR04.....	21
รูปที่ 3.5 สัญญาณที่ขา Trig และขา Echo ของโมดูลเซนเซอร์ultrasonic sensor HC-SR04.....	21
รูปที่ 3.6 Servo motor แบบ DC ที่ใช้ในการทำโครงการ.....	22
รูปที่ 3.7 บอร์ด Arduino.....	23
รูปที่ 3.8 การเชื่อมต่อระหว่างบอร์ด Arduino.....	24
รูปที่ 3.9 ชื่อในแต่ละส่วนของบอร์ด Arduino.....	24
รูปที่ 3.10 ส่วนประกอบของชุดทดลอง.....	26
รูปที่ 4.1 แสดงหน้าต่าง Block diagram เพื่อเตรียมการสอบเทียบ.....	27
รูปที่ 4.2 หน้าต่าง Front Panel และ Block diagram.....	27
รูปที่ 4.3 แสดงกราฟความสัมพันธ์ระหว่างสัญญาณพัลส์กับระยะจริง.....	28
รูปที่ 4.4 Block diagram ในการทดลอง.....	29
รูปที่ 4.5 หน้าต่าง Front Panel เพื่อแสดงผลการทดลอง.....	30
รูปที่ 4.6 กราฟการทำงานที่ $K_C = 20$, $T_I = 0.25$ และ $T_D = 0.003$	31
รูปที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่างค่า K_P กับค่า Maximum overshoot.....	32
รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่างค่า K_P กับค่า Setting time.....	33

รูปที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างค่า K_p กับค่า Steady state error.....	33
รูปที่ 4.10 กราฟแสดงความสัมพันธ์ระหว่างค่า K_I กับค่า Maximum overshoot.....	34
รูปที่ 4.11 กราฟแสดงความสัมพันธ์ระหว่างค่า K_I กับค่า Setting time.....	35
รูปที่ 4.12 กราฟแสดงความสัมพันธ์ระหว่างค่า K_I กับค่า Steady state error.....	35
รูปที่ 4.13 กราฟการทำงานที่ $K_p = 20$, $K_I = 70$ และ $K_D = 0.06$	36
รูปที่ 4.14 กราฟแสดงความสัมพันธ์ระหว่างค่า K_D กับค่า Maximum overshoot.....	37
รูปที่ 4.15 กราฟแสดงความสัมพันธ์ระหว่างค่า K_D กับค่า Setting time.....	37
รูปที่ 4.16 กราฟแสดงความสัมพันธ์ระหว่างค่า K_D กับค่า Steady state error.....	37

สารบัญตาราง

หน้า

ตารางที่ 1.1	ขั้นตอนการดำเนินงานวิจัย.....	3
ตารางที่ 2.1	ค่าเกนของชุดทดลอง ball and beam.....	5
ตารางที่ 2.2	ผลตอบสนองของระบบควบคุมPID.....	5
ตารางที่ 2.3	ผลของการเพิ่มค่าพารามิเตอร์ของตัวควบคุมแบบ PID.....	11
ตารางที่ 2.4	คำศัพท์ที่ใช้เรียกใช้โปรแกรม LabVIEW.....	12
ตารางที่ 2.5	Icon สำหรับหน้าต่าง Front Panel.....	15
ตารางที่ 4.1	ตารางแสดงสมการการสอบเทียบในแต่ละจุด.....	28
ตารางที่ 4.2	แสดงค่าผลลัพธ์ในค่าเกนแต่ละตัว.....	31
ตารางที่ 4.3	ผลลัพธ์ของแต่ละค่าเมื่อเปลี่ยนค่า K_p	32
ตารางที่ 4.4	ผลลัพธ์ของแต่ละค่าเมื่อเปลี่ยนค่า K_I	34
ตารางที่ 4.5	แสดงผลลัพธ์ของแต่ละค่าเมื่อเปลี่ยนค่า K_D	36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในระบบควบคุมเป็นระบบที่สำคัญในขบวนการผลิตต่างๆเป็นอย่างสูง เนื่องจากการผลิตที่มีขนาดใหญ่ล้วนมีขบวนการผลิตที่ซับซ้อนค่อนข้างมาก การควบคุมขบวนการผลิตจึงเป็นปัจจัยหนึ่งที่จะทำให้ขบวนการผลิตมีประสิทธิภาพมากที่สุด ในปัจจุบันเทคโนโลยีระบบควบคุมได้รับการพัฒนาอย่างรวดเร็ว อุปกรณ์ใหม่ๆจึงเกิดขึ้นมาอย่างมากมายเพื่อให้บรรลุเป้าหมายตามที่ต้องการ

ในปัจจุบันนี้ตามโรงงานอุตสาหกรรมการผลิตทั่วไปส่วนใหญ่จะมีขั้นตอนในการผลิตแต่ละขั้นตอนจะใช้ระบบควบคุมแบบอัตโนมัติเพื่อทดแทนการใช้แรงงานคนในการทำ ทั้งนี้ยังช่วยลดต้นทุนในระยะยาว และยังสามารถผลิตสินค้าได้จำนวนมากกว่าเมื่อเทียบกับการใช้แรงงานคนในช่วงระยะเวลาที่เดียวกัน เมื่อพูดถึงส่วนของตัวควบคุมที่ใช้ในโรงงานอุตสาหกรรมส่วนใหญ่จะใช้ระบบควบคุมแบบพีโอดีเพื่อเพิ่มประสิทธิภาพในการผลิต

ซึ่งในการใช้ตัวควบคุมแบบพีโอดีที่ใช้กันอย่างแพร่หลายในหลายๆงาน ทั้งในด้านอุตสาหกรรมการผลิต การควบคุมในยานพาหนะ และอื่นๆอีกมากมาย อย่างเช่น ระบบแบบจำแนกสินค้า โดยที่เซ็นเซอร์จะตรวจจับแล้วจำแนกตามขนาดของวัตถุก่อนที่จะส่งแยกวัตถุต่อไป ระบบ Cruise Control ที่ใช้บ่อยในรถยนต์เพื่อควบคุมความเร็วของรถยนต์ให้คงที่ตามที่ต้องการ ขณะที่มียอดอยู่ข้างหน้า เซ็นเซอร์จะตรวจจับความเร็วของรถคันข้างหน้าเพื่อปรับระดับความเร็วให้มีความสัมพันธ์กับรถคันข้างหน้า เป็นต้น

จากที่ทราบกันว่าตัวควบคุมพีโอดีที่กล่าวมานั้นจะช่วยในการลดค่าความผิดพลาด สามารถลดระยะเวลาในการควบคุมการผลิตได้ ซึ่งชุดทดลอง Roller on Beam จะเป็นตัวที่สามารถแสดงให้เห็นถึงการควบคุมของตัวพีโอดีซึ่งจะเป็นแนวทางในการเพิ่มประสิทธิภาพในการผลิตต่างๆได้ ในการศึกษาในระดับอุดมศึกษาเรื่องระบบควบคุมนี้เป็นส่วนหนึ่งของวิชา system dynamics ซึ่งเป็นศาสตร์ที่ยากต่อความเข้าใจ และได้รับในส่วนแต่ละของทฤษฎี แต่จะไม่สามารถเห็นภาพได้ ชุดทดลองนี้จะสามารถแสดงให้เห็นถึงผลของตัวควบคุมได้ชัดเจนมากยิ่งขึ้นอีกทั้งยังสามารถแสดงผลเป็นรูปแบบกราฟควบคุมในทุกๆขณะที่ระบบเริ่มทำงานจนกระทั่งระบบเข้าสู่สภาวะคงตัวได้

ซึ่งจะใช้โปรแกรม LabVIEW เป็นตัวควบคุมอุปกรณ์ Arduino board โปรแกรมนี้สามารถเขียน Block diagram เพื่อไปสั่งให้ Arduino board ทำงานตามที่เราต้องการได้ ในส่วนของอุปกรณ์ของชุดทดลองนี้จะมี ultrasonic sensor จะเป็นตัวนำเข้าสู่สัญญาณที่วัดระยะทางของวัตถุไปยัง Arduino board จากนั้นจะเข้าสู่กระบวนการควบคุมแบบพีโอดีเพื่อคำนวณค่าที่จะต้องส่งให้ servo motor ไปควบคุมคานของชุดทดลอง ซึ่งจะเห็นได้ว่าการทำงานของชุดทดลองนี้จะมีการทำงานอย่างเป็นระบบ ซึ่งการสร้างชุดทดลองนี้สามารถช่วยลดต้นทุนในการศึกษาระบบควบคุมที่ใช้ในการผลิตจริง อีกทั้งยังสามารถนำไปประยุกต์ใช้เป็นชุดทดลองให้ศึกษาในหลักสูตรวิศวกรรมเครื่องกลต่อไป

1.2 วัตถุประสงค์ของงาน

1. ศึกษาการเปลี่ยนแปลงของระบบเมื่อปรับค่าเกน (K_p , K_i และ K_d) โดยชุดทดลอง Roller on beam
2. ลดค่าความผิดพลาดของระบบที่สภาวะคงตัวให้เหลือไม่เกิน 5% โดยชุดทดลอง Roller on beam

1.3 ขอบเขตงานวิจัย

1. เลือกใช้แบบชุดทดลอง Roller on Beam และใช้ ultrasonic sensor, servo motor, Arduino board เป็นอุปกรณ์ในชุดทดลอง
2. เลือกใช้ระบบควบคุมแบบพีไอดี
3. กำหนดให้ที่ตำแหน่ง Overshoot ไม่เกิน 5% มีค่า Setting time ไม่เกิน 6 วินาที

1.4 สมมติฐานของการศึกษา

1. ค่าความผิดพลาดที่สภาวะคงตัวสามารถลดลงเหลือ 2-5% ด้วยระบบควบคุมแบบพีไอดี
2. ค่าในระบบไม่เกินขอบเขตที่กำหนด

1.5 วิธีดำเนินงานวิจัย

1. ศึกษาระบบควบคุมแบบพีไอดี
2. ศึกษาการใช้โปรแกรม LabVIEW
3. ออกแบบชุดทดลอง Roller on beam
4. สร้างและทำการประกอบชุดทดลอง Roller on beam
5. ออกแบบระบบควบคุมแบบพีไอดีให้แก่ชุดทดลอง Roller on beam
6. ติดตั้งและทดสอบ ultrasonic sensor, servo motor และทำการส่งข้อมูลด้วยโปรแกรม LabVIEW ผ่านบอร์ด Arduino
7. ทำการปรับค่าเกน (K_p , K_i และ K_d) และเก็บค่า Maximum overshoot, Setting time และ Steady state error ในระบบของชุดการทดลอง Roller on beam
8. ทำการบันทึกผลการทดลองและวิเคราะห์
9. สรุปและจัดทำรายงานวิจัย

1.6 ประโยชน์ที่ได้รับจากโครงการ

1. ช่วยเพิ่มความรู้ความเข้าใจเกี่ยวกับระบบควบคุมแบบพีไอดี
2. ให้ความรู้แก่ผู้ที่มาศึกษาโครงการให้เข้าใจเกี่ยวกับระบบควบคุมแบบพีไอดีและโปรแกรม LabVIEW
3. สามารถนำมาเป็นชุดการทดลองสำหรับการเรียนรู้ในหลักสูตรวิศวกรรมเครื่องกล

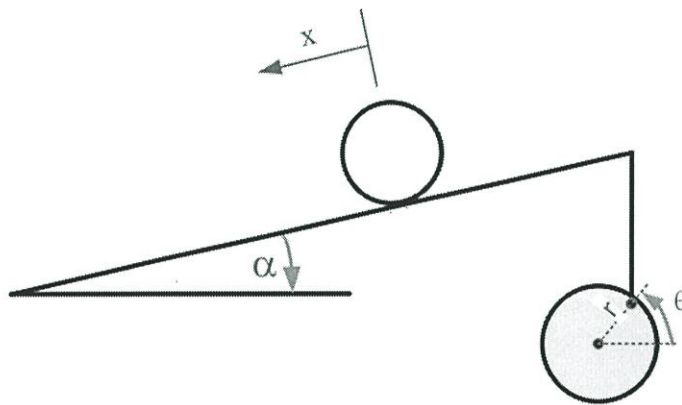
บทที่ 2

งานวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 หัวข้อที่ศึกษา

ในการศึกษาเรื่องระบบควบคุมแบบพีไอดี จำเป็นต้องมีความรู้เบื้องต้นเกี่ยวกับระบบควบคุมแบบพีไอดี และการเขียนแผนภาพ Block diagram ซึ่งจะเขียนแผนภาพผ่านโปรแกรม LabVIEW ดังนั้นจะต้องมีความรู้เกี่ยวกับวิธีการใช้โปรแกรม LabVIEW ที่จะกล่าวในหัวข้อต่อไป

2.1.1 งานวิจัยที่เกี่ยวข้อง



รูปที่ 2.1 Free body diagram ของชุดทดลอง[1]

1. จากงานวิจัย [1] A Ball-on-Beam Project Kit ของ Evencio A. Rosales, 2004 กล่าวว่าไว้ว่าชุดทดลอง Ball and Beam ระบบนี้ตั้งรูปที่ 2.1 พบว่าจะมี input ของระบบคือมุมของมอเตอร์ (θ) และ output ของระบบคือตำแหน่งของบอล (x) โดย ตัวควบคุมจะทำการควบคุมลูกบอลให้อยู่บนตำแหน่งที่ต้องการบนคาน โดยให้เวลาในการเข้าสู่สภาวะคงตัว (Settling time) และแอมพลิจูด (overshoot) ให้มีค่าน้อยๆ และสามารถรักษาเสถียรภาพไว้ได้ เมื่อมีการรบกวนระบบ โดยในงานวิจัยจะนำหลักการนี้มาใช้ในการประยุกต์สร้างชุดทดลอง Roller on beam ในงานวิจัยนี้
2. จากงานวิจัย [2] Modelling and control of Ball and beam system using model base and non-model base control approaches ของ Wen-Fang Xie, 2012 พบว่าในการปรับจูนค่าเกนแต่ละค่าเกนของระบบควบคุม PID ที่ใช้ควบคุมในชุดทดลอง ball and Beam ค่าเกนที่เหมาะสมกับชุดทดลองนั้น ควรจะปรับจูนโดยใช้วิธีการ Trial-and-Error ซึ่งค่าเกนของชุดทดลองที่ใช้ในระบบควบคุมและทำให้ระบบคงที่ที่ถูกแสดงในตารางที่ 2.1 และเมื่อนำค่าดังกล่าวมาใช้กับระบบจะได้ผลตอบสนองดังตารางที่ 2.2

ตารางที่ 2.1 ค่าเกณฑ์ของชุดทดลอง Ball and Beam[2]

K_p	K_i	K_d
0.7	0.15	0.0375

ตารางที่ 2.2 ผลตอบสนองของระบบควบคุม PID[2]

Controller	Setting time 5%	Steady-state error
PID	6.053 วินาที	11.9%

2.2 ตัวควบคุม

ความหมายและนิยามของระบบควบคุมคือ



รูปที่ 2.2 ภาพแสดงนิยามของระบบควบคุม

- ระบบ(System) หมายถึง ส่วนหรือหน่วยที่รวบรวมสิ่งต่างๆเข้าไว้ด้วยกัน
- ควบคุม(Control) หมายถึงการบังคับหรือสั่งการ
- ระบบควบคุม(Control Systems) หมายถึง ส่วนหรือหน่วยที่รวบรวมสิ่งต่างๆเข้าไว้ด้วยกัน เพื่อใช้บังคับหรือสั่งการเพื่อให้สิ่งใดสิ่งหนึ่งเป็นไปตามความต้องการ

โดยพื้นฐานของระบบควบคุมแบบดั้งเดิม คือการนำแบบจำลองทางคณิตศาสตร์ที่ไม่ซับซ้อนมากนักมาใช้ควบคุมระบบที่เป็นเชิงเส้น(Linear Systems) และระบบที่ไม่แปรผันตามเวลา(Time-invariant Systems) โดยแบบจำลองทางคณิตศาสตร์ที่ใช้ส่วนใหญ่จะอยู่ในรูปแบบของฟังก์ชันถ่ายโอน(Transfer Function) โดยตัวอย่างของระบบควบคุมแบบนี้ได้แก่ ระบบควบคุมพีไอดี(PID Controllers) เส้นทางการเดินราก (Root Locus) แผนภาพโบด (Bode Plot) และแผนภาพไนควิสต์ (Nyquist Plot) เป็นต้น

2.3 ระบบควบคุมพีไอดี

ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ (PID controller) เป็นระบบควบคุมแบบป้อนกลับที่นิยมใช้กันอย่างกว้างขวาง ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรใน

กระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของ PID ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ

2.3.1 ค่าตัวแปรที่ใช้ของระบบควบคุมพีไอดี

กลุ่มของตัวควบคุมที่มีสัญญาณเอาต์พุตเป็นอนาล็อก

1. ตัวควบคุมแบบสัดส่วน (Proportional Control)

เทอมของสัดส่วน (บางครั้งเรียก อัตราขยาย) จะเปลี่ยนแปลงเป็นสัดส่วนของค่าความผิดพลาด การตอบสนองของสัดส่วนสามารถทำได้โดยการคูณค่าความผิดพลาดด้วยค่าคงที่ K_p หรือที่เรียกว่า อัตราขยายสัดส่วน

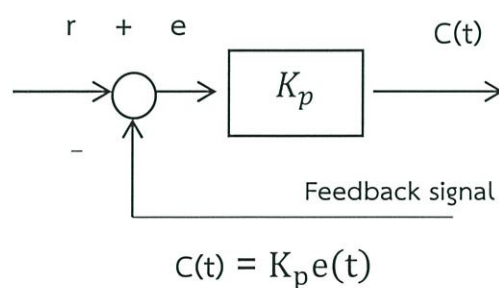
เทอมของสัดส่วนจะเป็นไปตามสมการ:

$$P_{out} = K_p e(t) \quad (1)$$

เมื่อ P_{out} , $C(t)$ สัญญาณขาออกของเทอมสัดส่วน

K_p	อัตราขยายสัดส่วน, ตัวแปรปรับค่าได้
e	ค่าความผิดพลาด
t	เวลา
r	สัญญาณนำเข้า

ผลอัตราขยายสัดส่วนที่สูงค่าความผิดพลาดก็จะเปลี่ยนแปลงมากเช่นกัน แต่ถ้าสูงเกินไประบบจะไม่เสถียรได้ ในทางตรงกันข้าม ผลอัตราขยายสัดส่วนที่ต่ำ ระบบควบคุมจะมีผลตอบสนองต่อกระบวนการน้อยตามไปด้วย บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.3



รูปที่ 2.3 ตัวควบคุมแบบสัดส่วน

จากรูปที่ 2.3 K_p คือ ตัวแปรที่ปรับค่าได้ ซึ่งจะรับค่าความผิดพลาดที่เกิดจากผลต่างระหว่างสัญญาณนำเข้ากับสัญญาณป้อนกลับมาปรับปรุงค่าความผิดพลาด โดยจะออกมาในรูปสัญญาณขาออก

จุดเด่นของตัวควบคุมแบบ P เมื่อนำไปใช้งานคือ การปรับค่าเกณฑ์ให้สูงขึ้นจะมีผลทำให้ระบบมีผลตอบสนองที่เร็วขึ้น ปัญหาที่อาจจะเกิดขึ้นในการนำไปใช้งานคือ ถ้านำไปใช้กับระบบชนิด 0 (System Type 0)

ตัวควบคุมแบบนี้จะไม่สามารถจัดค่าความผิดพลาดในสภาวะคงตัวได้ แต่สามารถทำให้ค่าความผิดพลาดดังกล่าวมีค่าน้อยลงได้ด้วยการปรับค่าเกนให้สูง ซึ่งในทางปฏิบัติ แล้วการปรับค่าเกนให้สูงมากขนาดไหนเอาต์พุตที่ออกจริงๆ จากตัวควบคุมมักมีค่าจำกัด และการปรับค่าเกนให้มีค่าสูงสำหรับระบบที่มีอันดับสูง อาจจะทำให้เกิดผลตอบสนองที่ไม่พึงประสงค์ เช่น การปรับค่าเกนให้สูงขึ้นในระบบอันดับสอง ผลที่ตามมาคือค่า Overshoot ก็สูงขึ้นตามด้วย ซึ่งอาจก่อให้เกิดอันตรายต่อระบบได้

2. ตัวควบคุมแบบปริพันธ์ (Integral Control)

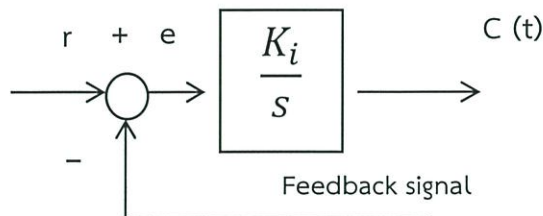
ผลจากเทอมปริพันธ์ (บางครั้งเรียก reset) เป็นสัดส่วนของขนาดความผิดพลาดและระยะเวลาของความผิดพลาด ผลรวมของความผิดพลาดในทุกช่วงเวลา (ปริพันธ์ของความผิดพลาด) จะให้ออฟเซตสะสมที่ควรจะเป็นในก่อนหน้า ความผิดพลาดสะสมจะถูกคูณโดยอัตราขยายปริพันธ์ ขนาดของผลของเทอมปริพันธ์จะกำหนดโดยอัตราขยายปริพันธ์

เทอมปริพันธ์จะเป็นไปตามสมการ:

$$I_{out} = K_I \int_0^t e(t) dt \quad (2)$$

เมื่อ	I_{out} , $C(t)$	สัญญาณขาออกของเทอมปริพันธ์
	K_I	อัตราขยายปริพันธ์ , ตัวแปรปรับค่าได้
	e	ความผิดพลาด
	t	เวลา
	r	สัญญาณนำเข้า

เทอมปริพันธ์เป็นการตอบสนองต่อความผิดพลาดสะสมในอดีต จึงสามารถทำให้เกิด overshoot ได้ (ข้ามจุดที่ต้องการและเกิดการหันเหไปทางทิศทางอื่น) บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.4



$$C(t) = K_I \int_0^t e(t) dt$$

รูปที่ 2.4 ตัวควบคุมแบบปริพันธ์

จากรูปที่ 2.4 K_I คือ ตัวแปรที่ปรับค่าได้ ซึ่งจะรับค่าความผิดพลาดที่เกิดจากผลต่างระหว่างสัญญาณนำเข้ากับสัญญาณป้อนกลับมาปรับปรุงค่าความผิดพลาด โดยจะออกมาในรูปสัญญาณขาออก

จุดเด่นของตัวควบคุมแบบ I เมื่อนำไปใช้งานคือ ถ้านำไปใช้กับระบบชนิด 0 ตัวควบคุมแบบนี้จะสามารถขจัดค่าความผิดพลาดในสภาวะคงตัวได้ ข้อด้อยที่อาจเกิดขึ้นในการนำไปใช้งานคือ ตัวควบคุมแบบนี้ไม่สามารถลดผลการ Overshoot ของผลตอบสนองได้และการปรับค่าเกินให้มีค่าสูงอาจจะทำให้ได้ผลตอบสนองที่ไม่พึงประสงค์ เช่น การปรับค่าเกินให้สูงขึ้นอาจจะทำให้ผลตอบสนองของระบบเกิดการแกว่งได้

3. ตัวควบคุมแบบอนุพันธ์ (Derivative Control)

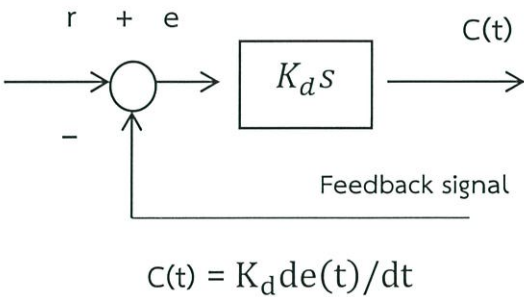
ลักษณะการทำงานคือ การส่งสัญญาณเอาต์พุตออกมาจากการอนุพันธ์สัญญาณค่าความผิดพลาด อัตราการเปลี่ยนแปลงของความผิดพลาดจากกระบวนการนั้นคำนวณหาจากความชันของความผิดพลาดทุกๆเวลา (นั่นคือเป็นอนุพันธ์อันดับหนึ่งสัมพันธ์กับเวลา) และคูณด้วยอัตราขยายอนุพันธ์ K_d ขนาดของผลของเทอมอนุพันธ์ (บางครั้งเรียก อัตรา) ขึ้นกับ อัตราขยายอนุพันธ์

เทอมอนุพันธ์จะเป็นไปตามสมการ:

$$D_{out} = K_d \frac{d}{dt} e(t) \tag{3}$$

เมื่อ $D_{out}, C(t)$	สัญญาณขาออกของเทอมอนุพันธ์
K_d	อัตราขยายอนุพันธ์, ตัวแปรปรับค่าได้
e	ค่าความผิดพลาด
t	เวลา

บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.5



รูปที่ 2.5 ตัวควบคุมแบบอนุพันธ์

จากรูปที่ 2.5 K_D คือ ตัวแปรที่ปรับค่าได้ ซึ่งจะรับค่าความผิดพลาดที่เกิดจากผลต่างระหว่างสัญญาณนำเข้ากับสัญญาณป้อนกลับมาปรับปรุงค่าความผิดพลาด โดยจะออกมาในรูปสัญญาณขาออก

จุดเด่นของตัวควบคุมแบบ D เมื่อนำไปใช้งานคือ ลดผลการ Overshoot ของผลตอบสนองได้ลดผลตอบสนองที่มีการเปลี่ยนแปลงไปมาได้ แต่ต้องปรับค่าเกินให้เหมาะสมด้วย ไม่เช่นนั้นอาจจะทำให้ได้ผล

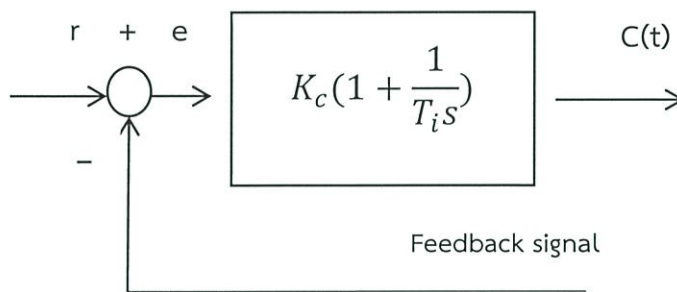
ตอบสนองที่ไม่พึงประสงค์ ปัญหาที่อาจเกิดขึ้นในการนำเอาตัวควบคุมแบบ D ไปใช้งานคือ ตัวควบคุมแบบ D จะไม่สามารถจัดค่าความผิดพลาดในสภาวะคงตัวได้ และการใช้ตัวควบคุมนี้อาจทำให้ได้ผลตอบสนองที่ช้าลง

4. ตัวควบคุมแบบสัดส่วนร่วมกับปริพันธ์ หรือตัวควบคุมแบบ PI (Proportional–Integral Control: PI Control) แบบใช้ค่าเกนร่วม

เทอมปริพันธ์ เมื่อรวมกับเทอมสัดส่วน จะเร่งกระบวนการให้เข้าสู่จุดที่ต้องการและจัดค่าความผิดพลาดที่เหลืออยู่ที่เกิดจากการใช้เพียงเทอมสัดส่วน เนื่องจากในเทอมปริพันธ์จะช่วยให้การจัดค่าความผิดพลาดที่สภาวะคงตัว ลักษณะการทำงานคือ การทำงานร่วมกันระหว่างตัวควบคุมแบบสัดส่วนและตัวควบคุมแบบปริพันธ์ บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.6

$$\text{โดย } K_c = K_p \quad (4)$$

$$\frac{K_c}{T_i} = K_i \quad (5)$$

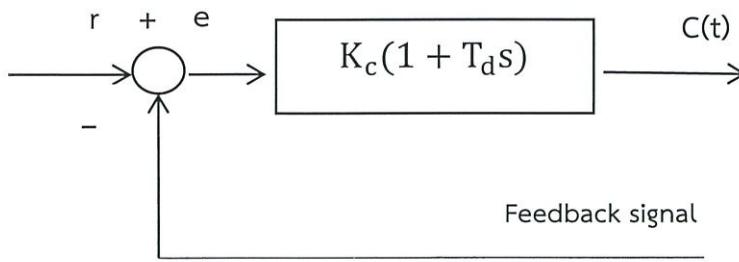


รูปที่ 2.6 ตัวควบคุมแบบ PI

5. ตัวควบคุมแบบสัดส่วนร่วมกับอนุพันธ์ หรือตัวควบคุมแบบ PD (Proportional–Derivative Control: PD Control) แบบใช้ค่าเกนร่วม

เทอมอนุพันธ์ เมื่อรวมกับเทอมสัดส่วน จะช่วยลดค่า overshoot และทำให้ระบบเข้าสู่สภาวะคงตัวได้ เนื่องจากในเทอมอนุพันธ์ จะช่วยลด overshoot ของผลตอบสนอง ลักษณะการทำงานคือ การทำงานร่วมกันระหว่างตัวควบคุมแบบสัดส่วนและตัวควบคุมแบบอนุพันธ์ บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.7

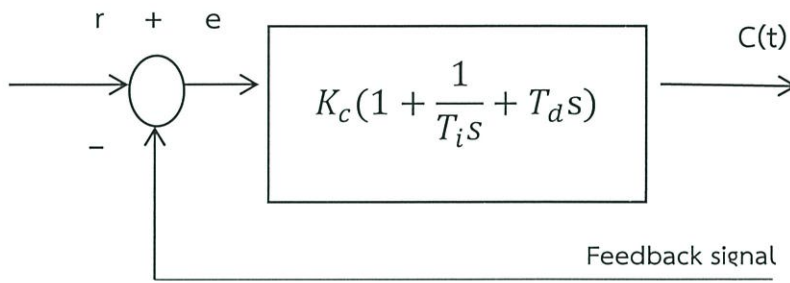
$$\text{โดย } K_c * T_d = K_d \quad (6)$$



รูปที่ 2.7 ตัวควบคุมแบบ PD

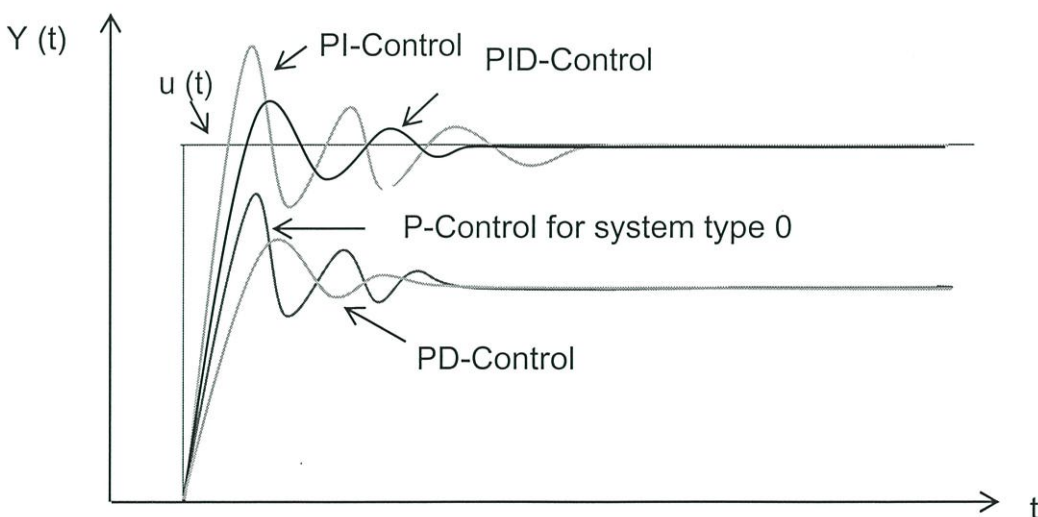
6. ตัวควบคุมแบบสัดส่วนร่วมกับปริพันธ์และอนุพันธ์ หรือตัวควบคุมแบบ PID (Proportional-Integral-Derivative Control: PID Control) แบบใช้ค่าเกินร่วม

เมื่อรวมกันทั้งสามเทอม แต่ละเทอมก็จะทำหน้าที่ที่แตกต่างกันไป ซึ่งข้อดีของแต่ละเทอมก็จะช่วยขจัดข้อด้อยของเทอมอื่นๆ จนทำให้ได้ค่าที่เหมาะสม ลักษณะการทำงานคือ การทำงานร่วมกันของตัวควบคุมแบบสัดส่วน ตัวควบคุมแบบปริพันธ์และตัวควบคุมแบบอนุพันธ์ บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.8



รูปที่ 2.8 ตัวควบคุมแบบ PID

ตัวอย่างผลตอบสนองต่ออินพุตแบบขั้นบันไดของระบบ ด้วยการใช้งานระบบควบคุม PID ในลักษณะต่างกัน แสดงดังรูปที่ 2.9



รูปที่ 2.9 ผลตอบสนองจากการนำตัวควบคุมแบบ PID แบบต่าง ๆ ไปใช้กับระบบควบคุม[4]

จากรูปที่ 2.9 จะเห็นว่าหากนำตัวควบคุมแบบ P ไปใช้กับระบบที่เป็นอันดับหนึ่ง ผลตอบสนองที่ได้จะมีค่าความผิดพลาดในสภาวะคงตัว ซึ่งสามารถลดผลกระทบได้ด้วยการเพิ่มค่าเกนของตัวควบคุมให้สูงขึ้น สำหรับการนำไปใช้กับระบบที่มีอันดับสูงกว่าและเป็นระบบชนิด 0 นั้นค่าความผิดพลาดในสภาวะคงตัวจะยังคงมีค่าอยู่ และการลดผลกระทบด้วยการปรับค่าเกนของตัวควบคุม P ให้สูงขึ้น อาจส่งผลให้ Overshoot มีค่าสูงขึ้นได้ ถ้าหากใช้ตัวควบคุมร่วมกันระหว่าง P กับ I หรือตัวควบคุมแบบ PI กับระบบนี้แล้ว ตัวควบคุมแบบ I จะช่วยขจัดค่าความผิดพลาดในสภาวะคงตัว แต่ผลตอบสนองที่ได้จะยังมีค่า Overshoot เหมือนเดิม ถ้าหากใช้ตัวควบคุมร่วมกันระหว่าง P กับ D หรือตัวควบคุมแบบ PD กับระบบนี้แล้ว การ Overshoot ของผลตอบสนองจะลดลง แต่ค่าความผิดพลาดในสภาวะคงตัวก็ยังคงอยู่ดังนั้นถ้าใช้ตัวควบคุมแบบ PID ด้วยการปรับค่าเกนให้เหมาะสมกับระบบ ก็จะได้ผลตอบสนองแบบหน่วงต่ำกว่าวิกฤตที่มีค่า Overshoot เหมาะสมกับระบบนั้น ๆ

2.3.2 การทดสอบหาค่าเกนที่เหมาะสมของตัวควบคุมแบบ PID

จากรูปที่ 2.9 จะเห็นว่าการทำงานที่จะได้ผลตอบสนองที่เหมาะสมของตัวควบคุมแต่ละแบบนั้น ขึ้นอยู่กับอันดับและชนิดของระบบเป็นสำคัญ และระบบแต่ละระบบมีค่าเกนที่เหมาะสมต่าง ๆ กันออกไป ทั้งนี้ยังต้องรวมไปถึงข้อกำหนดคุณลักษณะในการออกแบบด้วย เพื่อให้ได้ผลตอบสนองที่ผู้ใช้และออกแบบพึงพอใจการให้ได้มาซึ่งค่าเกนที่เหมาะสมสำหรับระบบทั่ว ๆ ไปที่อาจจะมีอันดับสูง เป็นเรื่องที่ยาก ในที่นี้จะนำเสนอวิธีการทดสอบระบบเพื่อหาค่าเกนที่เหมาะสมของตัวควบคุมแบบ PID คือ วิธีการปรับจูนเพื่อหาค่าที่เหมาะสม (Fine tuning)

เริ่มโดยการเซตค่า K_I และ K_d เป็นศูนย์ และค่อย ๆ ทำการเพิ่มค่า K_p ขึ้นเรื่อย ๆ จนกระทั่งระบบเริ่มเกิดการแกว่ง (Oscillate) จากนั้นให้เริ่มเพิ่มค่า K_I จนได้เวลาการตอบสนองของระบบที่ต้องการ เพราะถ้า K_I มีค่ามากเกินไปจะทำให้ระบบไม่เสถียร สุดท้ายหากมีความจำเป็น เพิ่มค่า K_d จนกระทั่งการควบคุมเร็วหรืออยู่ในระดับที่ยอมรับได้โดยเทียบกับระบบเมื่อมี Disturbance อ้างอิง ถ้าค่า K_d มากเกินไปจะทำให้การตอบสนองช้าเกินไปและเกิด Overshoot เกินยอมรับได้ปกติการปรับจูน PID ถ้าเกิด Overshoot เล็กน้อยจะช่วยให้เข้าสู่จุดที่ต้องการเร็วขึ้น แต่สำหรับบางระบบไม่สามารถยอมให้เกิด Overshoot ได้ ซึ่งสามารถแสดงผลของค่าพารามิเตอร์ได้ดังตารางที่ 2.3

ตารางที่ 2.3 ผลของการเพิ่มค่าพารามิเตอร์ของตัวควบคุมแบบ PID[4]

ผลของการเพิ่มค่าพารามิเตอร์ของตัวควบคุมแบบ PID				
พารามิเตอร์	Rise time	Overshoot	Setting time	Steady-state error
K_p	ลดลง	เพิ่มขึ้น	เปลี่ยนแปลงเล็กน้อย	ลดลง
K_i	ลดลง	เพิ่มขึ้น	เพิ่มขึ้น	ลดลงอย่างมีนัยสำคัญ
K_d	ลดลงหรือเพิ่มขึ้นเล็กน้อย	ลดลงเล็กน้อย	ลดลง	ไม่มีผล

2.4 ความรู้เบื้องต้นเกี่ยวกับโปรแกรม LabVIEW

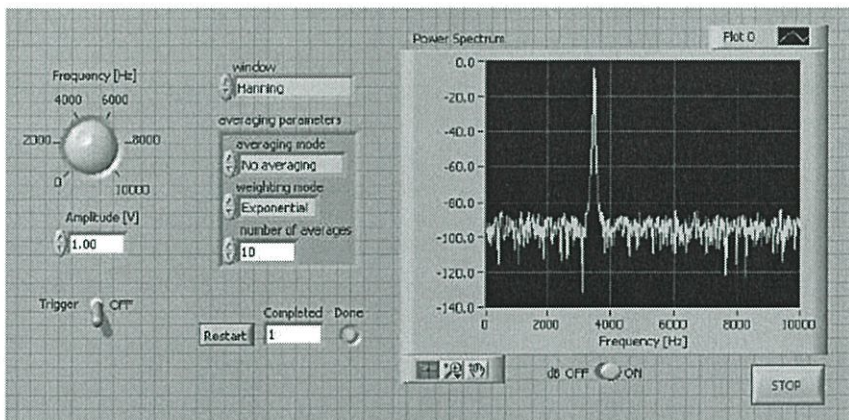
คำศัพท์ที่ใช้เรียกในโปรแกรม LabVIEW นั้น จะมีความแตกต่างจากภาษาอื่น ๆ ที่เรารู้จักกันอยู่พอสมควร กล่าวคือจะมีชื่อเรียกไม่เหมือนกันแต่มีหน้าที่การใช้งานเหมือนกัน ดังนั้นเพื่อให้เข้าใจได้ง่ายขึ้นควรทำความเข้าใจกับคำศัพท์ที่ใช้เรียกในโปรแกรม LabVIEW กับโปรแกรมพื้นฐานกันก่อนดังตารางที่ 2.4

ตารางที่ 2.4 คำศัพท์ที่ใช้เรียกใช้โปรแกรม LabVIEW[3]

LabVIEW	โปรแกรมพื้นฐาน	ความหมาย
Function	Function	ฟังก์ชันที่ถูกสร้างขึ้นมาด้วยตัวโปรแกรมเอง
VI	Program	ตัวโปรแกรมหลัก
Sub VI	Subroutine	โปรแกรมย่อย
Block Diagram	Program code	การเขียนโปรแกรมเพื่อให้วงจรทำงานตามที่กำหนด
Front Panel	User Interface	ส่วนควบคุมจากผู้ใช้งาน

โปรแกรมที่ถูกสร้างโดยโปรแกรม LabVIEW จะเรียกว่า Virtual Instrument (VI) เพราะมีลักษณะเหมือนกับเครื่องมือจริง ๆ ที่แสดงออกมาเป็นจอภาพ ซึ่งจะเป็นการทำงานของฟังก์ชันโปรแกรมย่อยและโปรแกรมหลัก เหมือนกับภาษาทั่ว ๆ ไป สำหรับ Virtual Instrument จะประกอบด้วยส่วนประกอบที่สำคัญสามส่วนคือ

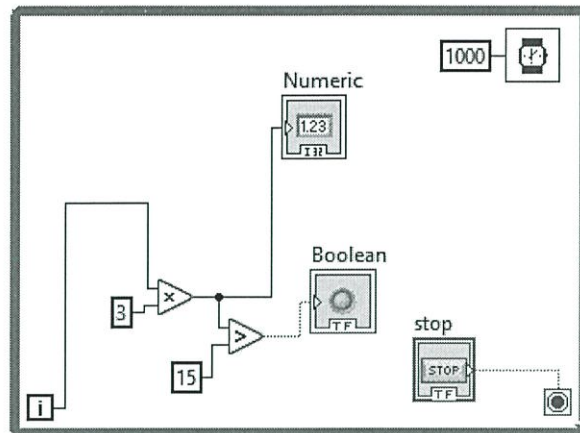
1. Front Panel หรือแผงหน้า จะเป็นส่วนที่ใช้เชื่อมโยงระหว่างโปรแกรมกับตัวผู้ใช้หรือที่นิยมเรียก User Interface ลักษณะทั่วไปจะเหมือนกับแผงควบคุมของเครื่องมือวัดต่าง ๆ ดังรูปที่ 2.10 โดยมี ปุ่มหมุนเปิด-ปิดสวิตช์โยก สวิตช์กด และจอภาพแสดงผล เป็นต้น หรือจะกำหนดการใช้งานเองก็ได้ ซึ่ง Front Panel ก็เหมือน GUI (Graphical user interface) ของโปรแกรมหลักนั่นเอง



รูปที่ 2.10 ลักษณะของ Front Panel[3]

2. Block Diagram ก็เปรียบได้กับ Source code ของโปรแกรมอื่น ๆ ซึ่งโปรแกรม LabVIEW จะอยู่ในรูปของภาษารูปภาพ (graphic language) ซึ่งข้อดีของ Block Diagram นี้จะมีการตรวจสอบความ

ผิดพลาดอยู่ตลอดเวลาทำให้ไม่สามารถทำงานได้ถ้าหากมีข้อผิดพลาดเกิดขึ้นถ้าเกิดมีข้อผิดพลาดขึ้นเราก็สามารถดูรายละเอียดข้อผิดพลาดนั้นได้ทันที ทำให้สะดวกในการเขียนโปรแกรมมากขึ้นสำหรับ Block Diagram จะมีลักษณะเป็นฟังก์ชันที่มีค่าเป็น Block ใด Block หนึ่งดังรูปที่ 2.11 ซึ่งจะถูกเชื่อมโยงโดยสายสัญญาณเพื่อให้เกิดการส่งข้อมูลแล้วไปแสดงผลในFront Panel ต่อไป



รูปที่ 2.11 ลักษณะของ Block Diagram[3]

3. Icon และ Connector คือโปรแกรมย่อยที่ใช้ในการส่งข้อมูลเข้า – ออกผ่านทาง Connector ซึ่งโปรแกรม LabVIEW นี้จะเรียกว่า SubVI สำหรับลักษณะทั่วไปของ Icon และ Connector จะแสดงในรูป 2.12



รูปที่ 2.12 ลักษณะของ Icon และ Connector ในโปรแกรม LabVIEW[3]

2.4.1 พื้นฐานและการใช้งานโปรแกรม LabVIEW

LabVIEW คือโปรแกรมคอมพิวเตอร์ที่สร้างขึ้นมาเพื่อนำไปใช้ในด้านการวัดและเครื่องมือวัดสำหรับงานทางวิศวกรรม LabVIEW ย่อมาจาก Laboratory Virtual Instrument Engineering Workbench ซึ่งเป็นโปรแกรมที่สร้างเครื่องมือวัดที่เสมือนจริง ในที่นี้อ้างอิงจากโปรแกรม LabVIEW 2017 การใช้งานโปรแกรม LabVIEW นั้นเป็นการทำงานภายใต้สถานะที่เรียกว่า GUI (Graphic User Interface) ซึ่งเป็นการเขียนโปรแกรมแบบลากวางแล้วโยงเส้นสัญญาณเชื่อมเข้าหากันในแต่ละบล็อกให้ข้อมูลที่เรต้องการนั้นไหล แต่ด้วย LabVIEW ซึ่งทำงานแบบ GUI Programming ทำให้การทำงานของเราง่ายยิ่งขึ้น เพียงแค่ลากวางแล้วทำการเขียนโปรแกรมอีกนิดหน่อย เท่านั้นก็สามารถที่จะนำข้อมูลออกมาได้แล้ว ในส่วนเรื่องการติดต่อกับอุปกรณ์ฮาร์ดแวร์ภายนอกนั้นทางบริษัท NI ซึ่งผลิตโปรแกรม LabVIEW นี้ได้กล่าวว่า โปรแกรม LabVIEW สามารถรองรับอุปกรณ์ภายนอกได้

หลายชนิดและยังมีไลบรารีรับรองอีกมากมาย แต่ยังมีข้อเสียคือ ไม่สามารถทำงานเป็น Stand alone ได้ ซึ่งหมายความว่า ถ้าต้องการทำอุปกรณ์ขึ้นมาใหม่สักชิ้นโดยใช้เพียงแต่ตัวไมโครคอนโทรลเลอร์จะไม่สามารถทำได้ เพราะการใช้งานไม่เหมาะกับงานขนาดเล็ก เพราะฉะนั้นจะเห็นได้ว่า LabVIEW นั้นถูกใช้ในโรงงานอุตสาหกรรมหรือไม่ก็ตามห้องทดลองที่เกี่ยวข้องกับด้านสัญญาณเป็นส่วนใหญ่

2.4.2 การสร้างไฟล์ออกแบบวงจร Front Panel และ Block Diagram

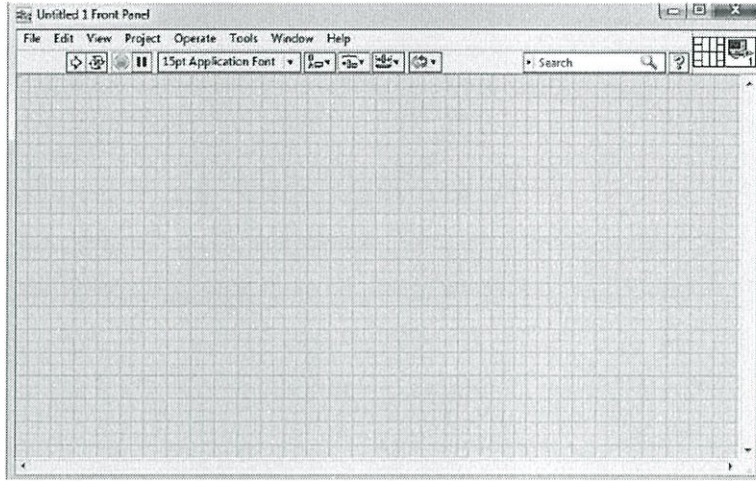
ไฟล์ออกแบบวงจร Front Panel คือไฟล์ที่ใช้สำหรับสร้างหรือออกแบบวงจรโดยการนำอุปกรณ์ที่ต้องการใช้มาวางลงบนพื้นที่ทำงานในตำแหน่งใดก็ได้ เช่น ต้องการสร้างวงจรปิด-เปิดไฟก็นำอุปกรณ์สวิตช์กับหลอดไฟมาวางลงบน Front Panel ซึ่งในตอนนี้องจรจะยังไม่ทำงานเราต้องสร้างหน้าต่าง Block Diagram ขึ้นมาด้วย ซึ่งเป็นหน้าต่างที่ใช้เชื่อมต่อกับอุปกรณ์ให้สามารถทำงานได้

1. เปิดโปรแกรม LabVIEW แล้วคลิกที่ Blank VI ดังรูปที่ 2.13 เพื่อสร้างไฟล์เอกสารใหม่



รูปที่ 2.13 หน้าต่างสร้างไฟล์งานของโปรแกรม LabVIEW 2014[3]

2. จะปรากฏหน้าต่าง Front Panel ดังรูปที่ 2.14 ที่ใช้สำหรับวางอุปกรณ์ขึ้นมา



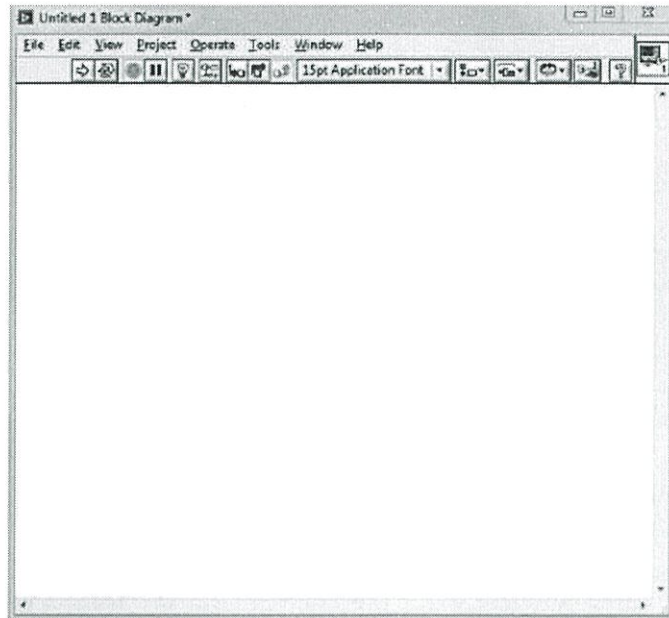
รูปที่ 2.14 หน้าต่าง Front Panel ของโปรแกรม LabVIEW 2014[3]

Icon ต่าง ๆ ที่ปรากฏในหน้าต่าง Front Panel สามารถตรวจสอบได้จากตารางที่ 2.5

ตารางที่ 2.5 Icon สำหรับหน้าต่าง Front Panel[3]

Icon	ชื่อ Icon	หน้าที่ของ Icon
	RUN	ใช้สั่งงานให้โปรแกรมเริ่มทำงาน
	Running	โปรแกรมกำลังทำงานอยู่
	Run continuously	สั่งงานให้โปรแกรมทำงานอย่างต่อเนื่อง
	Running continuously	โปรแกรมกำลังทำงานอย่างต่อเนื่อง
	Abort Execution	สั่งให้โปรแกรมหยุดทำงาน
	Pause	สั่งให้โปรแกรมหยุดทำงานชั่วคราว
15pt Application Font	Text Settings	กำหนดรูปแบบตัวอักษรได้ตามต้องการ
	Align Objects	การจัดการรูปแบบของวัตถุให้อยู่ในระนาบเดียวกัน (ลากเมาส์คลุมอุปกรณ์)
	Distribute Objects	การจัดระยะห่างของวัตถุในรูปแบบต่าง ๆ (ลากเมาส์คลุมอุปกรณ์)
	Resize Objects	ปรับขนาดวัตถุของอุปกรณ์ (ลากเมาส์คลุมอุปกรณ์)
	Reorder	จัดลำดับอุปกรณ์ที่วางซ้อนกัน
	Help	คู่มือการใช้งานโปรแกรม

3. จะปรากฏหน้าต่าง Block Diagram ดังรูปที่ 2.15 ที่ใช้สำหรับเชื่อมต่อวงจรขึ้นมา



รูปที่ 2.15 หน้าต่าง Block Diagram ของโปรแกรม LabVIEW 2017[3]

บทที่ 3

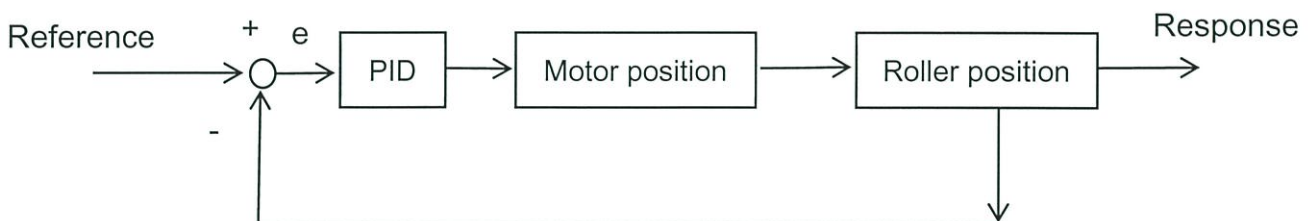
วิธีการดำเนินงานและการออกแบบชุดทดลอง

3.1 หลักการและแนวคิด

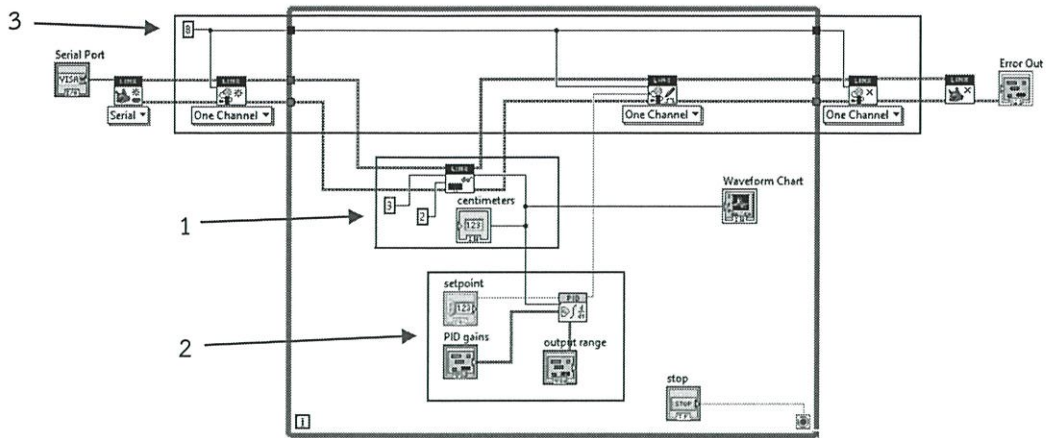
ในโครงการนี้ได้แนวคิดมาจากในโรงอุตสาหกรรม ซึ่งในกระบวนการผลิตส่วนใหญ่แล้วได้นำเทคโนโลยีสมัยใหม่ๆเข้ามาช่วยในการผลิตแทนการใช้แรงงานคน เราจึงสนใจเทคโนโลยีในส่วนของระบบการลำเลียงสินค้าและระบบคัดแยกสินค้า เช่น เครื่องแยกสินค้าที่มีขนาดกล่องต่างกัน ซึ่งจะต้องมีทั้งอุปกรณ์วัดและอุปกรณ์ควบคุมหรือจะสามารถเรียกได้ว่าระบบควบคุม ซึ่งในโรงงานอุตสาหกรรมทุกที่ล้วนต้องการให้มีกระบวนการผลิตที่มีประสิทธิภาพและประสิทธิผลมากที่สุด ซึ่งในส่วนของระบบควบคุมการผลิตนับว่าเป็นปัจจัยสำคัญที่ช่วยเพิ่มประสิทธิภาพในการผลิตให้มีสินค้าที่คุณภาพดี ช่วยลดต้นทุนและประหยัดพลังงาน ดังนั้นจึงจะต้องออกแบบระบบควบคุมให้มีความแม่นยำและเที่ยงตรงที่สุด

3.2 หลักการทำงานของชุดทดลอง

ชุดทดลองนี้เป็นชุดทดลองที่ทำการควบคุมลูกกลิ้งให้เคลื่อนที่ไปยังตำแหน่งที่กำหนดบนคาน โดยในเริ่มต้นเราจะป้อนค่าระยะที่เราต้องการให้ลูกกลิ้งหยุดนิ่ง (reference) เข้าไปในระบบ จากนั้น ultrasonic sensor จะทำการวัดระยะที่ลูกกลิ้งอยู่ห่างจากตัว sensor แล้วส่งค่าที่ได้ให้กับระบบควบคุม PID เพื่อทำการเปรียบเทียบค่าที่วัดได้กับค่าที่ป้อนให้แล้วจึงแก้ไขให้ค่าเท่ากัน จากนั้นจึงส่งต่อไปกับมอเตอร์เพื่อหมุนคานให้ไปยังตำแหน่งที่สามารถพาลูกกลิ้งไปถึงจุดที่เราต้องการ แต่ถ้าลูกกลิ้งยังไม่เข้าใกล้ระยะที่เราต้องการ ultrasonic sensor จะวัดระยะ ณ ตำแหน่งนั้นๆ และส่งกลับมาเข้าสู่ระบบควบคุม PID เพื่อทำการแก้ไขต่อไป สามารถแสดงได้ดังรูป 3.1



รูปที่ 3.1 Block diagram ของระบบ



รูปที่ 3.2 Block diagram ในโปรแกรม LabVIEW

จากรูป 3.2 เห็นว่าส่วนที่ 1. คือ ultrasonic block ประกอบไปด้วยช่องขาเข้าและช่องขาออกและช่องสำหรับแสดงผลจากการวัดระยะทาง ส่วนที่ 2. PID Block ประกอบไปด้วย 3 ส่วน ส่วนแรกคือ Setpoint สำหรับกำหนดตำแหน่งของวัตถุที่ต้องการให้เคลื่อนที่ไป ณ จุดนั้น ส่วนที่สองคือ PID gains คือค่าสำหรับปรับจูนเพื่อให้วัตถุเข้าสู่จุดที่ต้องการซึ่งประกอบไปด้วย 3 ตัว คือ K_p , K_I และ K_d ต่อมาในส่วนที่สามคือ Output range คือช่วงของ Output ใน PID Block ที่จะกำหนดช่วงก่อนส่งเข้าสู่ Servo motor block และส่วนที่ 3. Servo motor block ประกอบด้วยช่องขาเข้าและช่องสำหรับรับค่าจาก PID block เพื่อทำการปรับตำแหน่งของวัตถุ

3.3 การออกแบบชุดทดลอง

การออกแบบชุดทดลองจะเน้นในส่วนของโปรแกรมเป็นตัวควบคุมการทำงาน ซึ่งผู้ใช้งานสามารถควบคุมการทำงานของชุดทดลองผ่านโปรแกรม LabVIEW โดยจะมีส่วนสำคัญหลักๆได้แก่ ส่วนประสานกับผู้ใช้งาน และ ส่วนติดต่อสื่อสารระหว่างอุปกรณ์ควบคุมกับโปรแกรม LabVIEW ในส่วนของการออกแบบชุดทดลองจะมีองค์ประกอบที่สำคัญด้วยกัน 6 องค์ประกอบคือ

1. ผู้ใช้งาน
2. คอมพิวเตอร์ที่ติดตั้งโปรแกรม LabVIEW
3. อุปกรณ์ Arduino
4. เซนเซอร์วัดระยะทาง
5. servomotor
6. อุปกรณ์ภายในของระบบควบคุม

3.4 ความสัมพันธ์ของแต่ละอุปกรณ์

โดยการเชื่อมต่อระหว่างแต่ละองค์ประกอบเข้าด้วยกัน สามารถแบ่งได้เป็น 5 ส่วนหลักๆ ดังนี้

1. User กับ Computer

โดยคอมพิวเตอร์นี้จะติดตั้งโปรแกรม LabVIEW พร้อมเขียนโปรแกรมเพื่อสื่อสารข้อมูลกับผู้ใช้งานผ่านทางหน้าจอกราฟิก ซึ่งโปรแกรม LabVIEW จะรับคำสั่งจากผู้ใช้งานเพื่อไปส่งคำสั่งไปยังส่วนอื่นๆต่อไป

2. Computer กับบอร์ด Arduino

ส่วนนี้คอมพิวเตอร์เชื่อมต่อกับบอร์ด Arduino ผ่านทางพอร์ต USB ซึ่งบอร์ด Arduino จะทำงานได้ก็ต่อเมื่อได้รับไฟเลี้ยงจากภายนอก ซึ่งในส่วนนี้ไฟเลี้ยงที่บอร์ด Arduino ได้รับก็คือไฟเลี้ยงจากคอมพิวเตอร์ส่งผ่านทางพอร์ต USB เมื่อ Arduino ทำงานได้ บอร์ดนี้จะเป็นตัวรับคำสั่งจากส่วนของคอมพิวเตอร์ที่ติดตั้งโปรแกรม LabVIEW มาควบคุมระบบต่อไป นอกจากนี้บอร์ด Arduino ยังสามารถรับสัญญาณจากอุปกรณ์วัดมาแสดงผลบนคอมพิวเตอร์ได้อีกด้วย โดยผ่านทางพอร์ต Input ของบอร์ด Arduino

3. Arduino กับ ultrasonic sensor

ในระหว่างการทำงานของชุดทดลอง ในส่วนของบอร์ด Arduino จะจ่ายแรงดันไฟขนาด 5 V ให้กับเซนเซอร์วัดระยะทางหรือ ultrasonic sensor จากนั้นบอร์ด Arduino จะรับสัญญาณ Output โดยจะแปลงค่าระยะทางให้เป็นสัญญาณไฟฟ้าเพื่อมาแสดงผลบนโปรแกรม LabVIEW และนำค่าระยะทางมาประมวลผลในการควบคุมมอเตอร์ผ่าน Digital channel ในบอร์ด Arduino ต่อไป

4. Arduino กับ Servomotor

ในระหว่างการทำงานของชุดทดลอง ในส่วนของบอร์ด Arduino จะจ่ายแรงดันไฟขนาด 5 V ให้กับ servomotor ซึ่งในส่วนของ servomotor จะรับสัญญาณจากบอร์ด Arduino ที่ผ่านการประมวลผลมาแล้ว มาแปลงสัญญาณให้เป็นค่าความเร็วรอบมอเตอร์แล้วทำงาน servomotor ต่อไปโดยสัญญาณที่ servomotor ได้รับเป็นสัญญาณ Input

5. Arduino กับ อุปกรณ์ภายในควบคุม

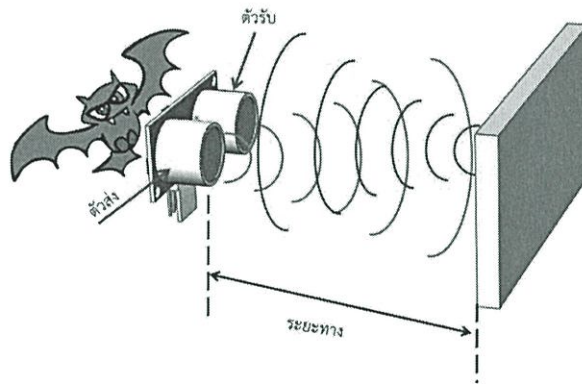
เมื่อบอร์ด Arduino ได้รับคำสั่งจากผู้ใช้งานและข้อมูลระยะทางผ่านโปรแกรม LabVIEW แล้วจากนั้นบอร์ด Arduino จะทำการประมวลผลข้อมูลแล้วส่งออกเป็นสัญญาณผ่านทาง Digital channels ไปยังอุปกรณ์ต่างๆภายในระบบควบคุม ซึ่งโครงการในที่นี่จะส่งข้อมูลไปยังไมโครคอนโทรลเลอร์และ servo motor

3.5 การนำอุปกรณ์มาใช้ในชุดทดลอง

3.5.1 เครื่องมือวัดระยะทาง (Displacement sensors)

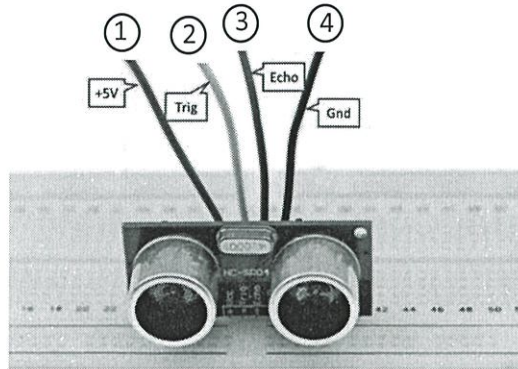
ในชุดทดลองนี้จะใช้ ultrasonic sensor รุ่น HC-SR04 ในการวัดระยะทาง โดยหลักการทำงานคือ ultrasonic sensor เป็นเซนเซอร์โมดูลสำหรับตรวจจับวัตถุและวัดระยะทางแบบไม่สัมผัส โดยใช้คลื่น ultrasonic sensor ซึ่งเป็นคลื่นเสียงความถี่สูงเกินกว่าการได้ยินของมนุษย์ วัดระยะได้ตั้งแต่ 2 – 400 เซนติเมตร หรือ 1 – 156 นิ้ว สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้ง่าย ใช้พลังงานต่ำ เหมาะกับการนำไปประยุกต์ใช้งานด้านระบบควบคุมอัตโนมัติ หรืองานด้านหุ่นยนต์ หลักการทำงาน จะเหมือนกันกับการตรวจจับวัตถุด้วยเสียงของค้างคาว โดยจะประกอบไปด้วยตัว รับ-ส่ง ultrasonic sensor ตัวส่งจะส่งคลื่นความถี่ 40 kHz ออกไปในอากาศด้วยความเร็วประมาณ 346 เมตรต่อวินาที และตัวรับจะคอยรับสัญญาณที่สะท้อนกลับจากวัตถุ เมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น, เวลาที่ใช้ในการเดินทางไป-กลับ (t) ก็จะสามารถคำนวณหาระยะห่างของวัตถุ (S) ได้จากความสัมพันธ์ดังสมการนี้

$$S = 346 \times 0.5t \quad (7)$$



รูปที่ 3.3 หลักการตรวจจับและวัดระยะห่างระหว่างวัตถุด้วยคลื่นเสียง[5]

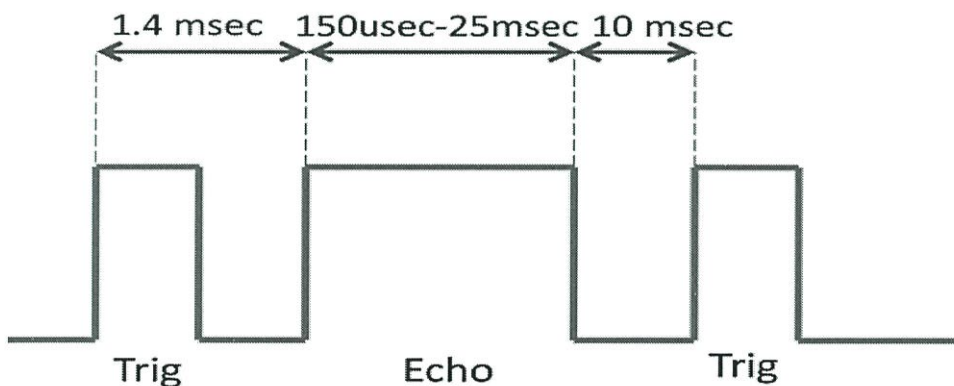
เพื่อให้การคำนวณหาระยะเป็นไปด้วยความง่าย โมดูลเซนเซอร์นี้จึงได้ประมวลผลให้เรียบร้อยแล้ว และส่งผลลัพธ์ของการคำนวณเป็นสัญญาณพัลส์ที่มีความกว้างสัมพันธ์กับระยะทางที่วัดได้ของโมดูลนี้มีจุดต่อใช้งานทั้งหมด 4 จุด โดยคุณสมบัติแต่ละขาเป็นดังรูปที่ 3.4



รูปที่ 3.4 การต่อใช้งานโมดูลultrasonic sensor HC-SR04[5]

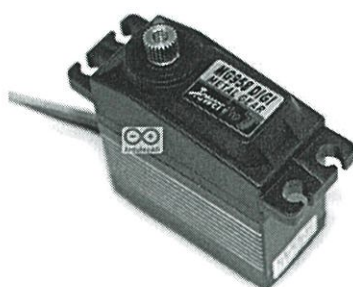
1. ขา VCC สำหรับต่อแรงดันไฟเลี้ยงไม่เกิน 5V
2. ขา Trig เป็นขาอินพุตรับสัญญาณพัลส์ความกว้าง 10 ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นultrasonic sensorความถี่ 40KHz ออกสู่อากาศจากตัวส่ง
3. ขา Echo เป็นขาเอาต์พุตสำหรับส่งสัญญาณพัลส์ออกจากโมดูลไปยังไมโครคอนโทรลเลอร์ เพื่อตรวจจับความกว้างของสัญญาณพัลส์และคำนวณเป็นระยะทาง
4. ขา GND สำหรับต่อจุดกราวด์ร่วมแรงดันและสัญญาณ

ตามคุณลักษณะของเซนเซอร์ จะต้องสร้างสัญญาณพัลส์ความกว้างไม่น้อยกว่า 10 msec ป้อนเข้าที่ขา Trig หลังจากนั้นอีกประมาณ 1.4 msec จึงจะเริ่มมีสัญญาณพัลส์เกิดขึ้นที่ขา Echo มีความกว้างของสัญญาณตั้งแต่ 150 usec – 25 msec ซึ่งถ้าหากกว้างกว่านี้จะถือว่าตรวจไม่พบวัตถุ หลังจากนั้นควรหน่วงเวลาออกไปอีก 10 msec จึงจะส่งสัญญาณ Trig ออกไปอีกรอบ ดังรูปที่3.5



รูปที่ 3.5 สัญญาณที่ขา Trig และขา Echo ของโมดูลเซนเซอร์ultrasonic sensor HC-SR04[5]

3.5.2 Servo motor



รูปที่ 3.6 Servo motor แบบ DC ที่ใช้ในการทำโครงงาน[6]

Servo ในทางวิศวกรรมเป็นการควบคุมการขับเคลื่อน (Servomechanism) คือ อุปกรณ์อิเล็กทรอนิกส์สำหรับควบคุมมอเตอร์หรือระบบขับเคลื่อนต่าง ๆ ด้วยการตรวจสอบสัญญาณจากระบบขับเคลื่อนและปรับค่าให้ถูกต้องตามที่กำหนดไว้อย่างต่อเนื่อง โดยปกติระบบ Servo จะตรวจสอบข้อผิดพลาดจากสัญญาณตอบกลับเช่น ตำแหน่งทางกล ความเร็วหรือค่าผิดพลาดอื่น ๆ เช่น ข้อผิดพลาดจากระบบหน้าต่างของรถยนต์อาจไม่ได้รวมอยู่ในระบบขับเคลื่อน (none servomechanism) หรือระบบควบคุมเรือซึ่งใช้การควบคุมแบบปิด (close-loop feedback) ถือว่าอยู่ในระบบขับเคลื่อนได้

Servo motor เป็นมอเตอร์ไฟฟ้ากระแสตรง DC Motor ที่ถูกประกอบรวมด้วยชุดเกียร์และส่วนควบคุมต่างๆ ไว้ในโมดูลเดียวกัน โดยจะมีสัญญาณใช้งาน 1 เส้น และอีก 2 เส้น เป็น VCC และ GND ซึ่งสามารถควบคุมให้ตัว Servo motor หมุนซ้าย หรือ ขวาได้ +90 องศา - 90 องศา (180 องศา) โดยสามารถสั่งงานในการหมุนให้หมุนไปได้ ตามองศาต่างๆ ที่ต้องการ ได้ด้วยตัวของ Servo motor เอง เช่น ต้องการหมุน 1 องศา หรือ 15 องศา ก็ไม่ต้องมีส่วนควบคุม หรือ SENSOR ใดๆ กลับมาตรวจสอบอีกทำให้ใช้งานได้ง่าย และสะดวกในการนำไปประยุกต์ใช้งานต่างๆ ได้จริง

อุปกรณ์ขับเคลื่อนที่ใช้ในชุดทดลองนี้จะใช้ MG996R High Torque Servo Tower Pro MG996R เพียงโลหะดังรูปที่ 3.6

- ขนาด : 40 x 19 x 43 มิลลิเมตร
- น้ำหนัก : 55 กรัม
- ความเร็วเชิงมุม : 6.16 รอบ(เรเดียน)ต่อวินาที (4.8V no load)
- ความเร็วเชิงมุม : 8.06 รอบ(เรเดียน)ต่อวินาที (6.0V no load)
- ค่าทอร์กสูงสุดที่รอบต่ำสุด : 13 กิโลกรัม-เซนติเมตร ที่ 4.8V
- ค่าทอร์กสูงสุดที่รอบต่ำสุด : 15 กิโลกรัม-เซนติเมตร ที่ 6V
- ช่วงของค่าความต่างศักย์ไฟฟ้า : 4.8 - 7.2Volts
- ชนิดเฟือง : เฟืองโลหะ

สำหรับการนำ servo motor ไปใช้งานจะประกอบไปด้วย 2 ประเภท ดังต่อไปนี้

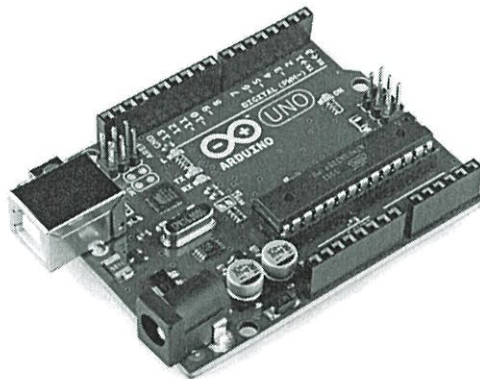
1.ควบคุมตำแหน่ง (Position control)

Servo ทั่วไปใช้สำหรับการควบคุมไฟฟ้า ไฮดรอลิกส์ หรือนิวเมติก โดยทำงานอาศัยพื้นฐานการป้อนกลับแบบลบ คือตำแหน่งอินพุตเทียบกับตำแหน่งที่แท้จริงของระบบเครื่องกล ที่วัดได้โดยตัวแปลงสัญญาณต่าง ๆ แล้วหาความแตกต่างระหว่างค่าจริงและค่าที่ต้องการ (เรียกว่าสัญญาณผิดพลาด Error signal) เมื่อได้ค่านี้แล้วก็ทำการขยายหรือแปลงเพื่อใช้ขับเคลื่อนระบบในตำแหน่งที่ต้องการ เพื่อลดหรือขจัดข้อผิดพลาด ขั้นตอนนี้เป็นทฤษฎีการควบคุมที่ใช้กันอย่างแพร่หลาย servo ทั่วไปจะให้ผลลัพธ์เชิงมุม (angular) หรือเชิงเส้น (linear output)

2.ควบคุมความเร็ว (Speed control)

การควบคุมความเร็วเป็นอีกรูปแบบหนึ่งของระบบ servo เครื่องจักรไอน้ำใช้ระบบนี้ ในสมัยก่อนคือการควบคุมกังหันน้ำ ก่อนสงครามโลกครั้งที่สอง ใบพัดความเร็วคงที่ได้รับการพัฒนาเพื่อควบคุมความเร็วของเครื่องบิน การควบคุมน้ำมันเชื้อเพลิงสำหรับเครื่องยนต์กังหันแก๊ส ในระบบไฮดรอลิกส์ และระบบอิเล็กทรอนิกส์

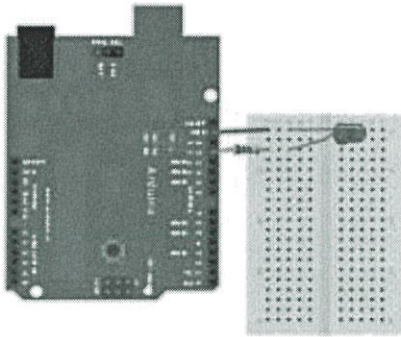
3.5.3 Arduino board



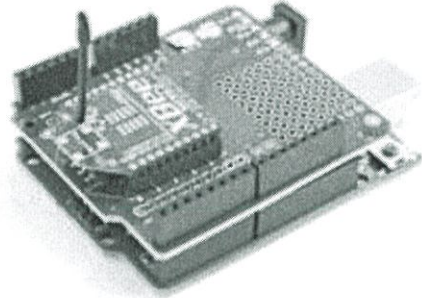
รูปที่ 3.7 บอร์ด Arduino Uno R3[6]

Arduino อ่านว่า (อา-ดู-อิ-โน้ หรือ อาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด (ดูตัวอย่างรูปที่ 1 แสดงดังรูปที่ 3.8) หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ (ดูตัวอย่างรูปที่ 2 แสดงดังรูปที่ 3.8) เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้ และส่วนประกอบของ บอร์ด Arduino แสดงดังรูปที่ 3.9

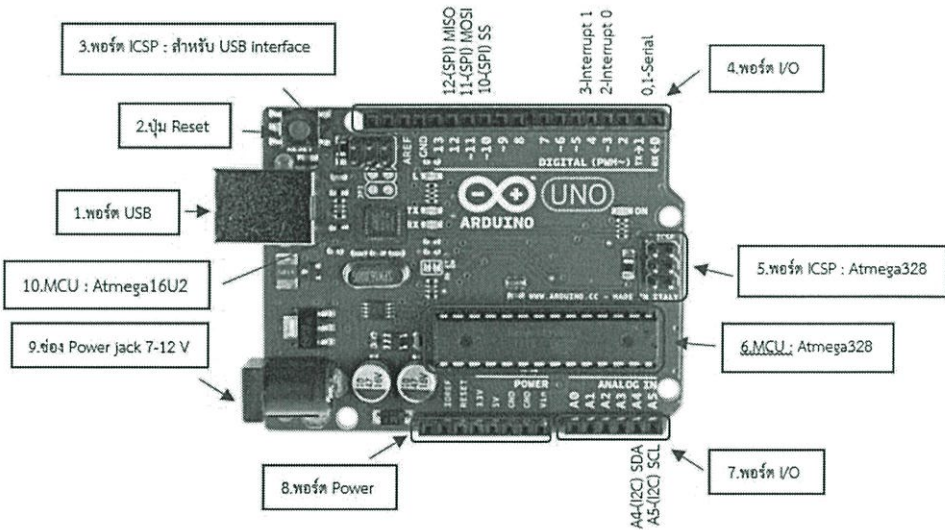


รูปที่ 1 บอร์ด Arduino ต่อกับ LED



รูปที่ 2 บอร์ด Arduino ต่อกับบอร์ด XBee Shield

รูปที่ 3.8 การเชื่อมต่อระหว่างบอร์ด Arduino[6]



รูปที่ 3.9 ชื่อในแต่ละส่วนของบอร์ด Arduino[6]

ชื่อในแต่ละส่วนของบอร์ด Arduino

1. USBPort ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

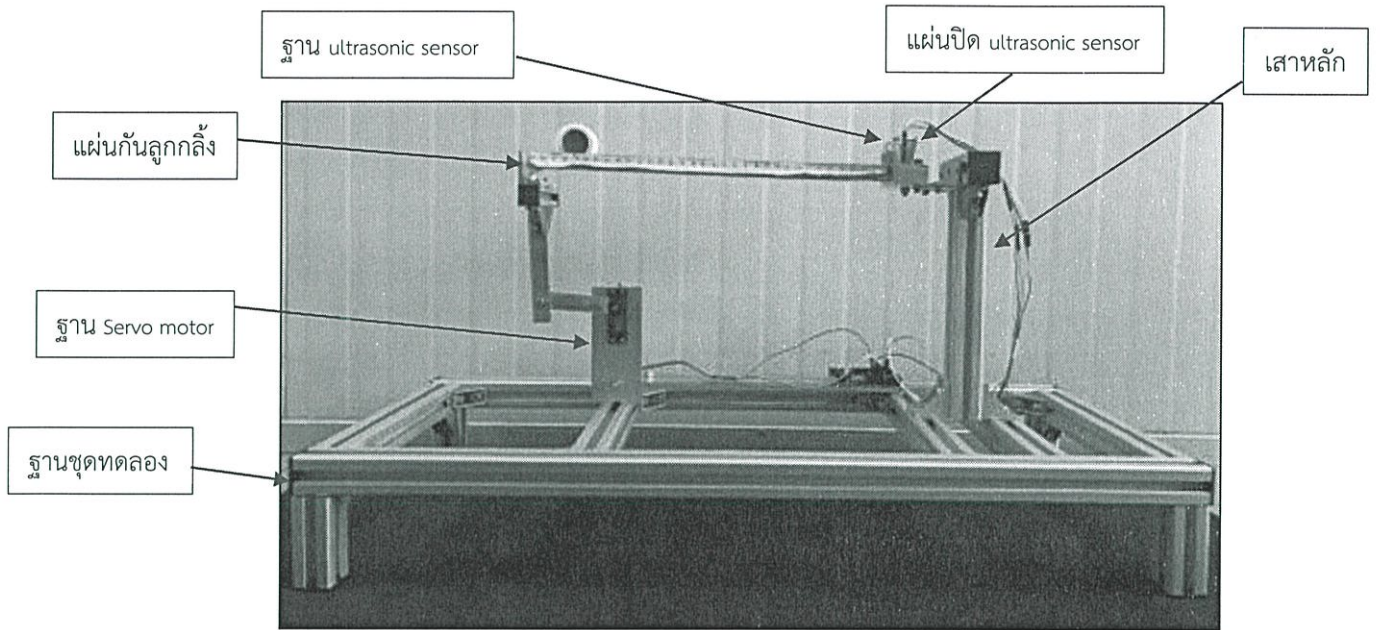
- | | |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 3. ICSP Port | ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2 |
| 4. I/O Port | Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM |
| 5. ICSP Port | ของ Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader |
| 6. MCU | Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino |
| 7. I/O Port | นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5 |
| 8. Power Port | ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V_{in} |
| 9. Power Jack | รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V |
| 10. MCU | ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2 |

3.6 ขั้นตอนการสร้างชุดทดลอง

สำหรับอุปกรณ์ที่ใช้ในการสร้างชุดทดลองมีดังนี้

1. แผ่นอะคริลิกหนา 5 mm
2. สเตนเลสขนาดเส้นผ่านศูนย์กลาง 5 mm
3. ลูกกลิ้ง
4. อลูมิเนียมโปรไฟล์ขนาด 30*30 mm
5. ฉากเข้ามอลูมิเนียมโปรไฟล์
6. อลูมิเนียมแผ่นหนา 1 mm, 2 mm, 6 mm และ 13 mm
7. อลูมิเนียมฉากหนา 3 mm
8. บานพับสเตนเลส
9. น็อตตัวผู้, น็อตตัวเมีย และแหวนรองน็อต

ซึ่งขั้นตอนการสร้างชุดทดลองจะเริ่มต้นด้วยการออกแบบแต่ละชิ้นส่วนในชุดทดลองผ่านโปรแกรม AutoCAD ซึ่งประกอบด้วย ฐานของชุดทดลอง, เสาหลักของชุดทดลอง, ฐานสำหรับใส่ ultrasonic sensor, แผ่นสำหรับปิดที่ตำแหน่งข้างหลัง ultrasonic sensor แผ่นสำหรับกันลูกกลิ้งที่ปลายราง, ฐานสำหรับใส่ Servo motor จากนั้นทำการสร้างชิ้นส่วนตามทีออกแบบไว้ เมื่อสร้างครบทุกชิ้นส่วนแล้วจึงประกอบทุกชิ้นส่วนเข้าด้วยกัน จากนั้นทำการปรับตำแหน่งของชิ้นส่วนให้เหมาะสม จึงได้ชุดทดลองที่เสร็จเรียบร้อยดังรูปที่ 3.10



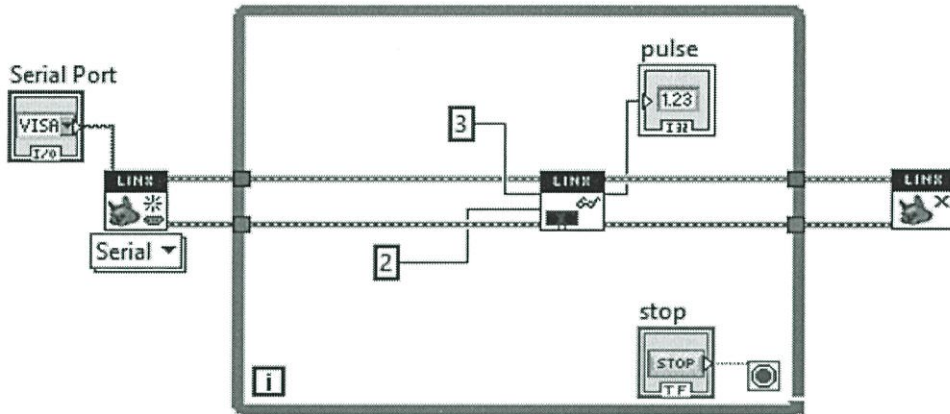
รูปที่ 3.10 ชุดทดลอง Roller on Beam

บทที่ 4

วิธีการทดลองและผลลัพธ์

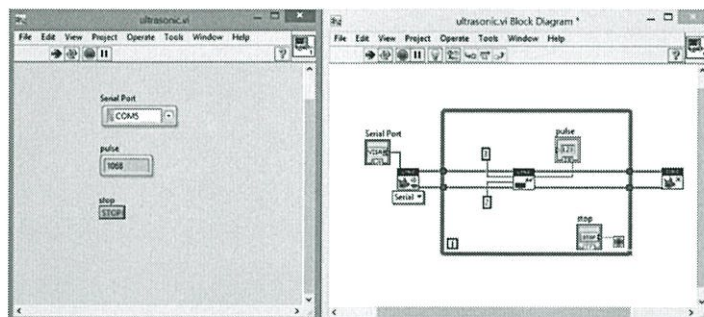
4.1 การสอบเทียบ ultrasonic sensor

การเปรียบเทียบค่าระยะทางของultrasonic sensorของโรงงาน กับค่าระยะทางจริงในชุดทดลอง เพื่อทดสอบอุปกรณ์วัดว่ามีคุณสมบัติที่มีคุณภาพและมีมาตรฐานที่สามารถนำไปใช้งานจริงได้ ซึ่งการสอบเทียบนี้จะทำการสอบเทียบผ่านโปรแกรม LabVIEW ซึ่งสามารถแสดงค่าสัญญาณพัลส์ผ่านทางหน้าจอแล้วนำมาเทียบกับระยะจริงที่วัดได้ สามารถเขียนบล็อกไดอะแกรมเพื่อทำการสอบเทียบได้ดังรูปต่อไปนี้



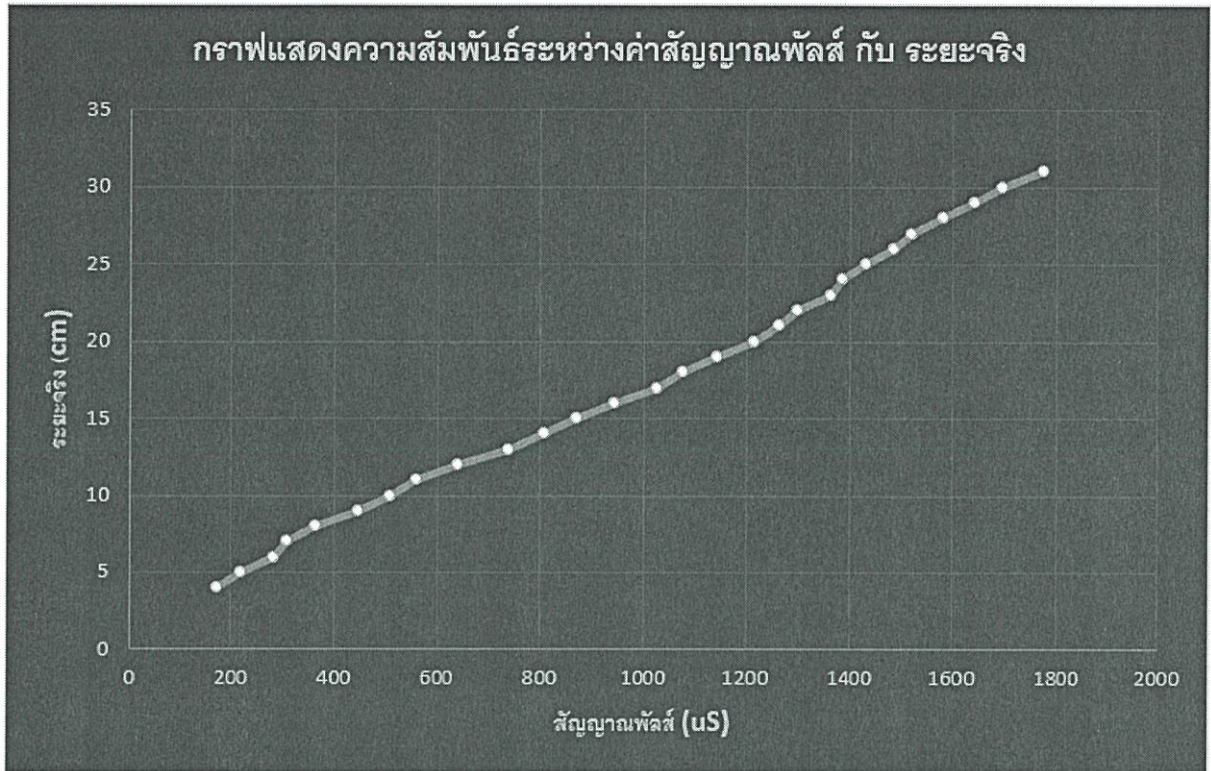
รูปที่ 4.1 แสดงหน้าต่าง Block diagram เพื่อเตรียมการสอบเทียบ

เมื่อเริ่มการทำงานของโปรแกรม ค่าที่วัดได้จะแสดงผลผ่านทางหน้าต่าง Front Panel ดังรูปต่อไปนี้



รูปที่ 4.2 หน้าต่าง Front Panel และ Block diagram

เมื่อทำการสอบเทียบ ultrasonic sensor แล้ว จากนั้นมาวิเคราะห์ผลลัพธ์ที่ได้แล้วแสดงผลผ่านกราฟความสัมพันธ์ระหว่างค่าสัญญาณพัลส์และค่าระยะจริงดังรูป



รูปที่ 4.3 แสดงกราฟความสัมพันธ์ระหว่างสัญญาณพัลส์กับระยะจริง

จากรูปจะพบว่า เมื่อทำการสอบเทียบค่าระยะทางทั้งหมด 28 จุด จะได้สมการเส้นตรงทั้งหมด 27 สมการ ตารางต่อไปนี้

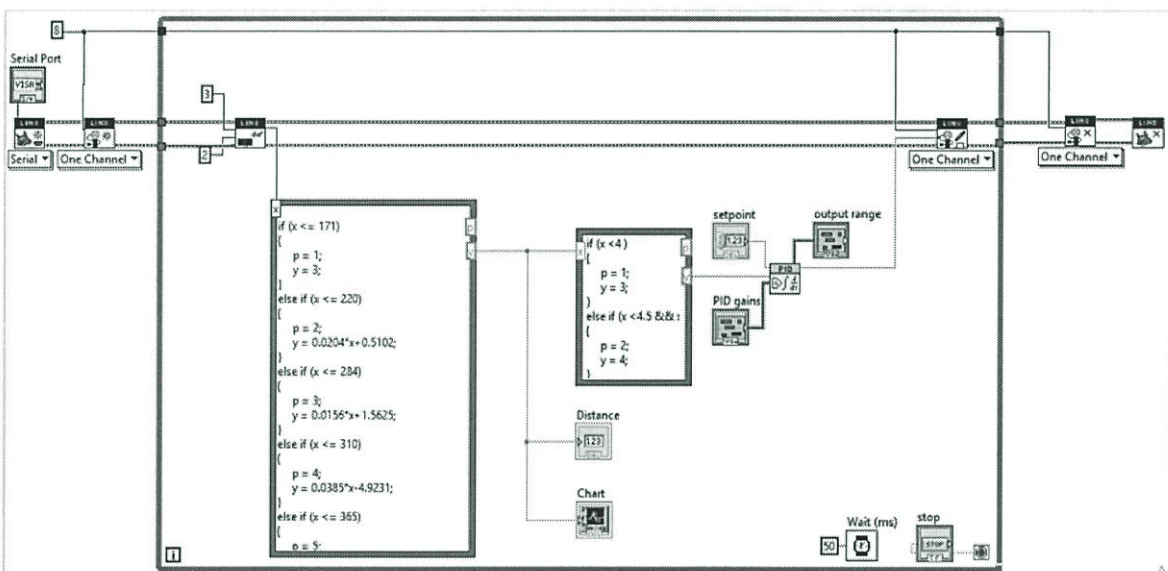
ตารางที่ 4.1 ตารางแสดงสมการการสอบเทียบในแต่ละจุด

ระยะ (cm)	สมการการสอบเทียบ	ระยะ (cm)	สมการการสอบเทียบ
4-5	$y = 0.0204x + 0.5102$	18-19	$y = 0.0148x + 2.0074$
5-6	$y = 0.0156x + 1.5625$	19-20	$y = 0.0143x + 2.6143$
6-7	$y = 0.0385x - 4.9231$	20-21	$y = 0.0219x - 6.6496$
7-8	$y = 0.0182x + 1.3636$	21-22	$y = 0.0287x - 15.249$
8-9	$y = 0.0123x + 3.4938$	22-23	$y = 0.0156x + 1.7792$
9-10	$y = 0.0164x + 1.6885$	23-24	$y = 0.0448x - 37.9700$
10-11	$y = 0.0192x + 0.25$	24-25	$y = 0.0214x - 5.6571$

11-12	$y = 0.0121x + 4.2379$	25-26	$y = 0.018x - 0.7006$
12-13	$y = 0.0101x + 5.5403$	26-27	$y = 0.0291x - 17.291$
13-14	$y = 0.015x + 1.8571$	27-28	$y = 0.0162x + 2.3405$
14-15	$y = 0.0165x + 0.6529$	28-29	$y = 0.0172x + 0.7184$
15-16	$y = 0.0133x + 3.4267$	29-30	$y = 0.0186x - 1.5652$
16-17	$y = 0.0119x + 4.7738$	30-31	$y = 0.0127x + 8.557$
17-18	$y = 0.019x - 2.5619$		

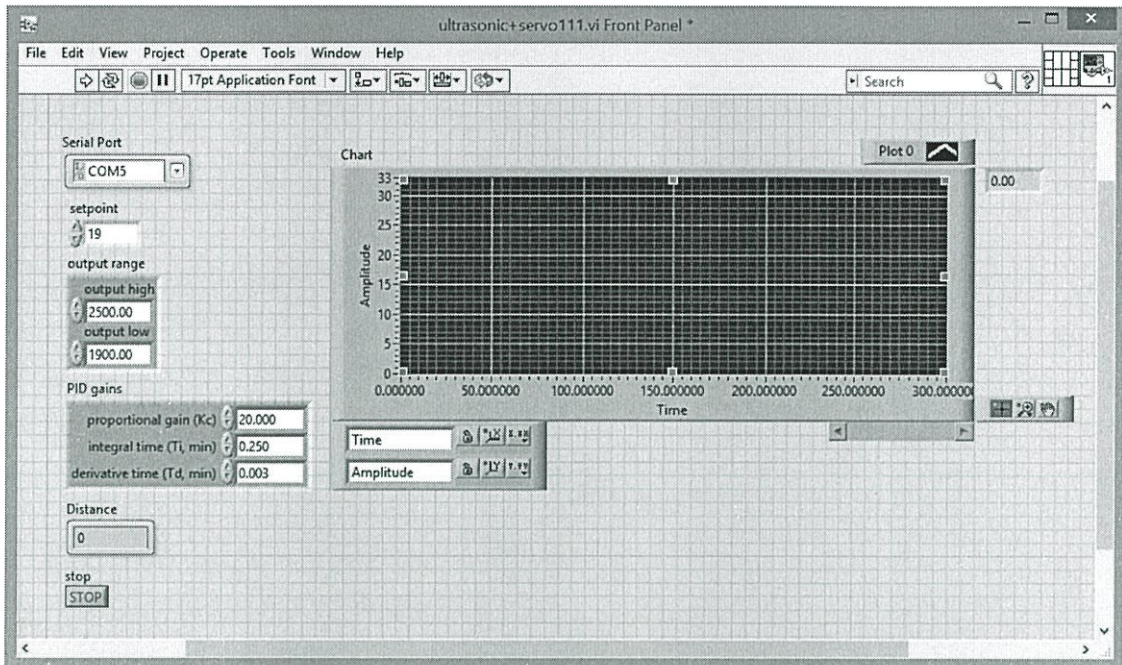
4.2 สร้างบล็อกไดอะแกรมผ่านโปรแกรม LabVIEW

จากนั้นนำสมการการสอบเทียบที่ได้ทั้งหมดนำมาใส่ในหน้าต่าง Block diagram ในโปรแกรม LabVIEW และทำการสร้าง Block diagram เพื่อทำการสั่งการทำงานของชุดทดลองในขั้นตอนต่อไป หน้าต่าง Block diagram ที่ได้เป็นไปตามดังรูปต่อไปนี้



รูปที่ 4.4 Block diagram ในการทดลอง

และหน้าต่างแสดงผลของการทำงานผ่านทางหน้าต่าง Front Panel ดังต่อไปนี้



รูปที่ 4.5 หน้าต่าง Front Panel เพื่อแสดงผลการทดลอง

4.3 การทดลอง

การทดลองที่ 1: ทำการปรับค่า K_p , T_I และ T_D เพื่อหาแนวโน้มพฤติกรรมของค่าผลลัพธ์ในแต่ละตัว และเลือกค่าที่ปรับที่เหมาะสมที่สุด ในขั้นตอนแรกเราจะทำการสุ่มค่าขึ้นมาค่าหนึ่ง จากนั้นทำการปรับค่าตัวหนึ่ง โดยที่อีกสองค่ามีค่าคงที่ พอได้ค่าที่ดีที่สุด จะทำการปรับค่าตัวที่สอง โดยที่อีกสองตัวที่เหลือคงที่ จากนั้นเลือกค่าที่ดีที่สุดมา แล้วทำการปรับตัวที่สาม โดยอีกสองตัวที่เหลือคงที่ จนได้ค่าที่ดีที่สุดมาค่าหนึ่ง

เนื่องจากในโปรแกรม LabVIEW ในหน้าต่าง Front Panel ที่ได้แสดงรูปให้เห็นในหัวข้อที่ผ่านนั้น จะเห็นได้ว่า ค่าเกณฑ์ไว้สำหรับทำการปรับจะเป็นค่า K_C , T_I และ T_D ค่าตัวแปรเหล่านี้จะมีความสัมพันธ์กับค่า K_p , K_I และ K_D ดังสมการที่ (4), (5) และ(6) จากบทที่2

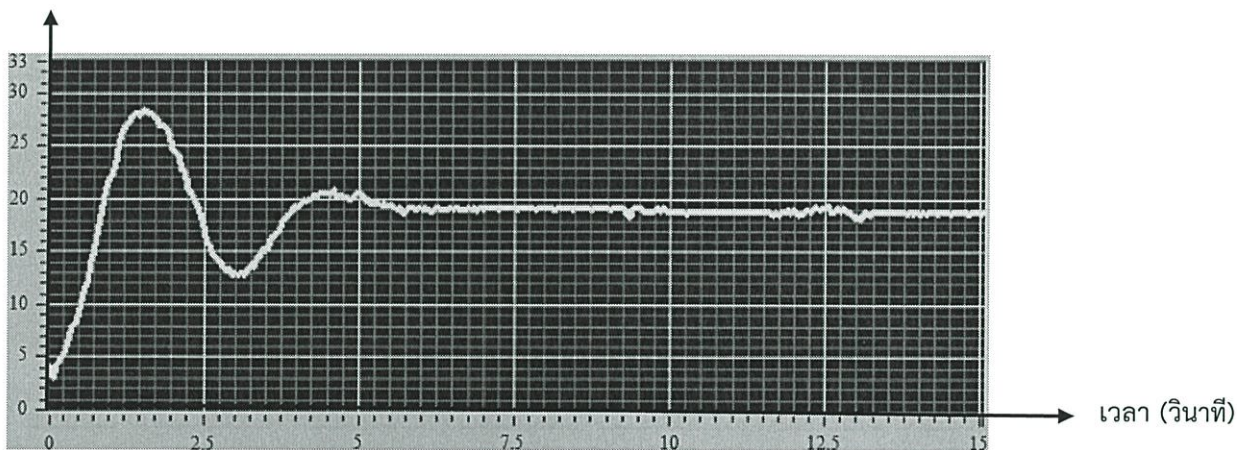
เมื่อเราทำการเลือกค่า K_p , K_I และ K_D มาค่าหนึ่ง จากนั้นก็จะทำการเปลี่ยนค่าตัวแปรเหล่านี้ให้เป็น K_C , T_I และ T_D แล้วจึงนำมาใส่ในโปรแกรม โดยทำการตั้งค่า Set point ที่ตำแหน่ง 19 เซนติเมตร

ตารางที่ 4.2 แสดงค่าผลลัพธ์ในค่าเกณฑ์แต่ละตัว

K_P	K_I	K_D	M_p (cm)	T_s (sec)	Output (cm)	Error (%)
5	50	0.005	More than 12	More than 10	-	-
20	200	0.02	11	10	19.4123	2.1700
50	500	0.05	More than 12	More than 10	-	-
20	20	0.02	6.75	6	19.6728	3.3640
20	80	0.02	11.25	6.5	19.1347	0.6735
20	80	0.06	9.15	4.5014	19.0982	0.5168
20	80	0.10	7.5	5.25	18.4286	2.8570

จากผลลัพธ์ที่แสดงในตารางที่ 4.2 จะพบว่า จะเริ่มที่ค่า $K_c = 5$ ($K_p=5$) , $T_I = 0.1$ ($K_I=50$) และ $T_D = 0.001$ ($K_D=0.005$) เมื่อปรับค่า K_c จะทำให้ค่า Maximum overshoot, Setting time และ error แตกต่างกันไป ในการเลือกค่าที่เหมาะสมจะทำการพิจารณาจากค่า Steady state error เป็นอันดับแรก และค่า Setting time เป็นอันดับต่อมา ดังนั้นจึงเลือกที่ค่า $K_c = 20$ ($K_p=20$) เป็นค่าที่ทำให้ผลลัพธ์ออกมาดีที่สุด จากนั้นปรับค่า T_I จนได้ค่าที่ดีที่สุดค่าหนึ่ง คือ $T_I = 0.25$ ($K_I=80$) จากนั้นปรับค่า T_D ต่อไป จะเห็นว่าที่ค่า $T_D = 0.003$ ($K_D=0.06$) จะทำให้ผลลัพธ์ออกมาดีที่สุด ดังนั้นจากการปรับจูนค่าในขั้นตอนนี้ จะได้ค่าที่ดีที่สุดคือ $K_p = 20$, $T_I = 0.25$ ($K_I=80$) และ $T_D = 0.003$ ($K_D=0.06$) สามารถแสดงกราฟตั้งแต่เริ่มทำงานจนถึงตำแหน่งที่สภาวะคงตัวได้ดังรูป

ระยะทาง (ซม.)

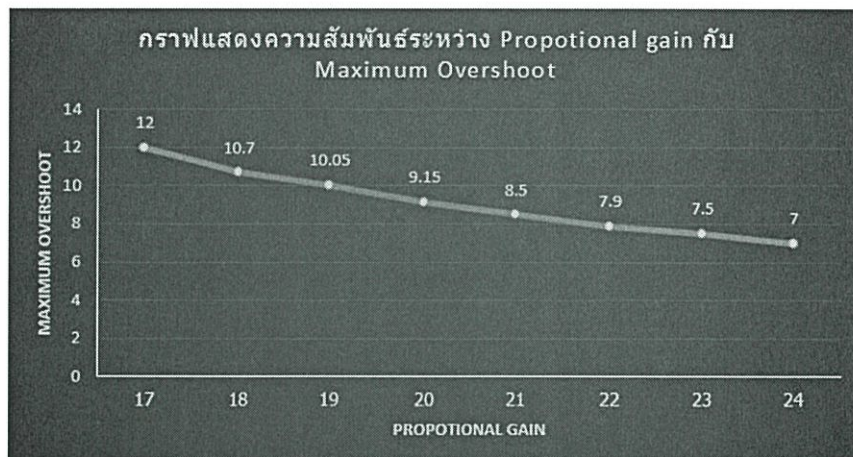
รูปที่ 4.6 กราฟการทำงานที่ $K_c = 20$, $T_I = 0.25$ และ $T_D = 0.003$

การทดลองที่ 2: ทำการปรับค่า K_p โดยให้ค่า K_I และ K_D คงที่ เพื่อหาความสัมพันธ์ของตัวแปร K_p ที่มีผลต่อระบบ โดยใช้วิธีการเพิ่มค่า K_p โดยกำหนดค่า Setpoint ที่ 19 เซนติเมตร

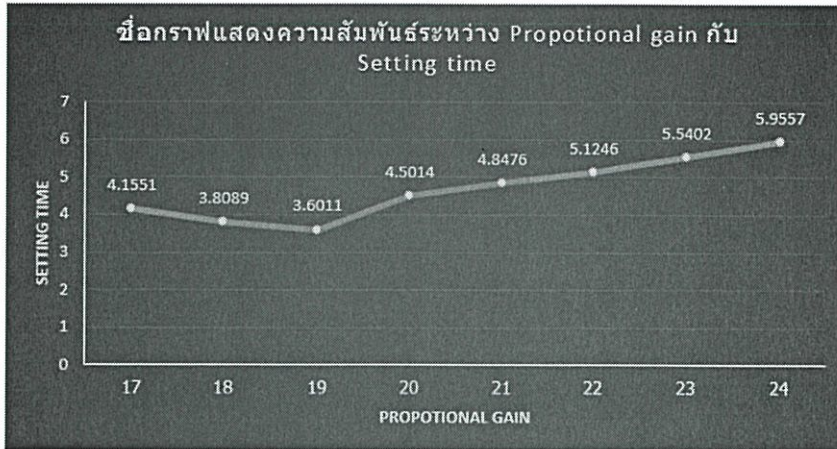
ตารางที่ 4.3 ผลลัพธ์ของแต่ละค่าเมื่อเปลี่ยนค่า K_p

K_p	K_I	K_D	$M_p(\text{cm})$	$T_s(\text{sec})$	Output (cm)	Error (%)
17	80	0.06	12.00	4.1551	21.4521	12.9058
18	80	0.06	10.70	3.8089	19.5981	3.1479
19	80	0.06	10.05	3.6011	19.4123	2.1700
20	80	0.06	9.15	4.5014	19.0982	0.5168
21	80	0.06	8.50	4.8476	18.5688	2.2695
22	80	0.06	7.90	5.1246	18.3700	3.3158
23	80	0.06	7.50	5.5402	17.9651	5.4468
24	80	0.06	7.00	5.9557	17.2500	9.2105

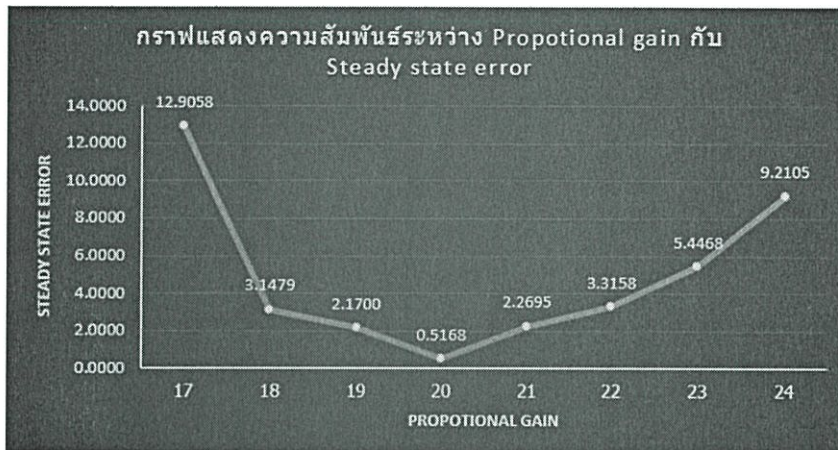
จากตารางที่ 4.3 จะเห็นว่าเมื่อปรับค่า K_p มากขึ้นค่า Maximum overshoot มีค่าลดลง ส่วนค่า Setting time มีค่าลดลงจนถึงค่าหนึ่งแล้วจากนั้นจะมากขึ้น และค่า Output ที่อ่านได้มีค่าลดลง ซึ่งสามารถแสดงผลได้ในรูปแบบกราฟความสัมพันธ์ได้ดังรูปต่อไปนี้



รูปที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่างค่า K_p กับค่า Maximum overshoot



รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่างค่า K_p กับค่า Setting time



รูปที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างค่า K_p กับค่า Steady state error

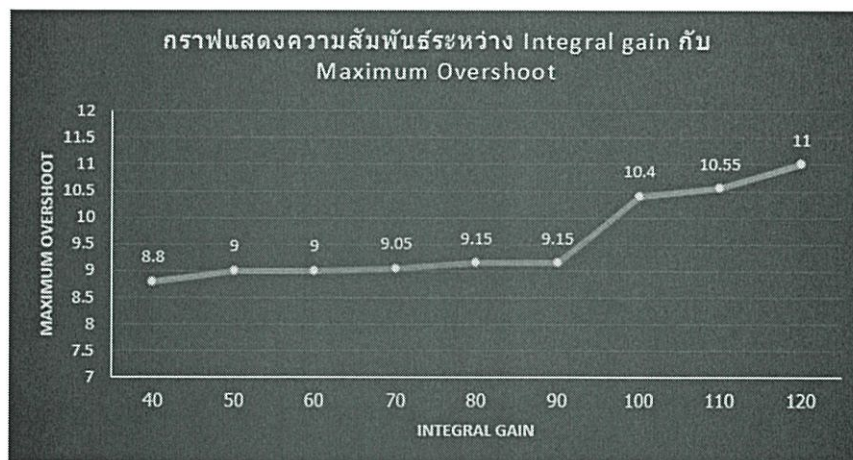
จะผลการทดลองดังกล่าวจะเห็นได้ว่า เมื่อ K_p มากขึ้นค่า Maximum overshoot มีแนวโน้มลดลง ส่วนค่า Setting time มีค่าลดลงจนถึงที่ตำแหน่ง $K_p = 19$ จากนั้นถึงเพิ่มมากขึ้น ส่วนค่า Steady state error มีลักษณะคล้ายกับ Setting time เมื่อพิจารณาที่ตั้งแต่ $K_p = 17$ ถึง 19 จะเห็นได้ว่าค่า Setting time มีค่าน้อยกว่าที่ $K_p = 20$ เล็กน้อย แต่ในส่วนของค่า Steady state error มีค่าในปริมาณที่มาก อีกทั้งเราต้องการผลการทดลองที่ค่า Steady state error ที่น้อยที่สุด ดังนั้นเราจึงเลือกค่า $K_p = 20$, $K_I = 80$ และ $K_D = 0.06$ เพื่อมาทำการทดลองต่อไป ซึ่งกราฟการทำงานนี้สามารถแสดงให้เห็นได้ในรูปที่ 4.6 ดังที่ได้กล่าวมาในการทดลองที่ 1 แล้ว

การทดลองที่ 3: ทำการปรับค่า K_I โดยให้ค่า K_P และ K_D คงที่ เพื่อหาความสัมพันธ์ของตัวแปร K_I ที่มีผลต่อระบบ โดยใช้วิธีการเพิ่มค่า K_I โดยกำหนดค่า Setpoint ที่ 19 เซนติเมตร

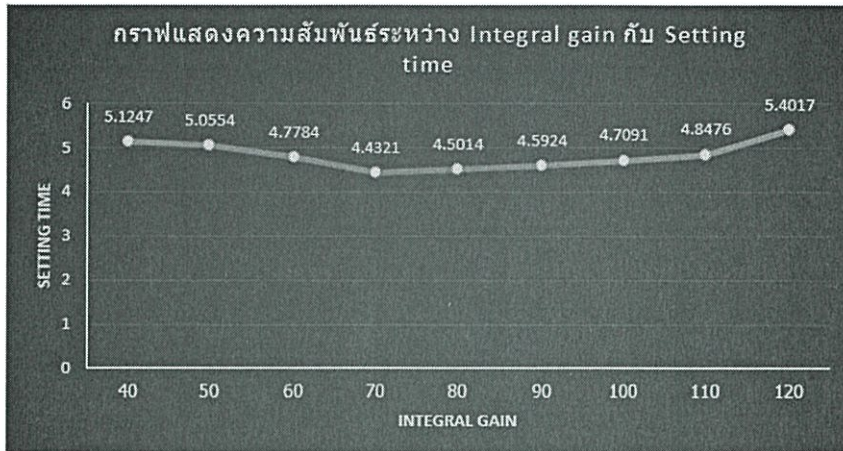
ตารางที่ 4.4 ผลลัพธ์ของแต่ละค่าเมื่อเปลี่ยนค่า K_I

K_P	K_I	K_D	M_P (cm)	T_S (sec)	Output (cm)	Error (%)
20	40	0.06	8.80	5.1247	19.1546	0.8137
20	50	0.06	9.00	5.0554	18.8644	0.7137
20	60	0.06	9.00	4.7784	19.1129	0.5942
20	70	0.06	9.05	4.4321	18.9187	0.4279
20	80	0.06	9.15	4.5014	19.0982	0.5168
20	90	0.06	9.15	4.5924	18.9001	0.5258
20	100	0.06	10.40	4.7091	19.1059	0.5574
20	110	0.06	10.55	4.8476	18.8910	0.5737
20	120	0.06	11.00	5.4017	19.1241	0.6532

จากตารางที่ 4.4 จะเห็นว่าเมื่อปรับค่า K_I มากขึ้น ค่า Maximum overshoot มีค่ามากขึ้น ส่วนค่า Setting time มีค่าลดลงจนถึงค่าหนึ่งแล้วจากนั้นจะมากขึ้น และค่า Output ที่อ่านได้จะมีค่า Error ที่น้อย ซึ่งสามารถแสดงผลได้ในรูปแบบกราฟความสัมพันธ์ได้ดังรูปต่อไปนี้



รูปที่ 4.10 กราฟแสดงความสัมพันธ์ระหว่างค่า K_I กับค่า Maximum overshoot



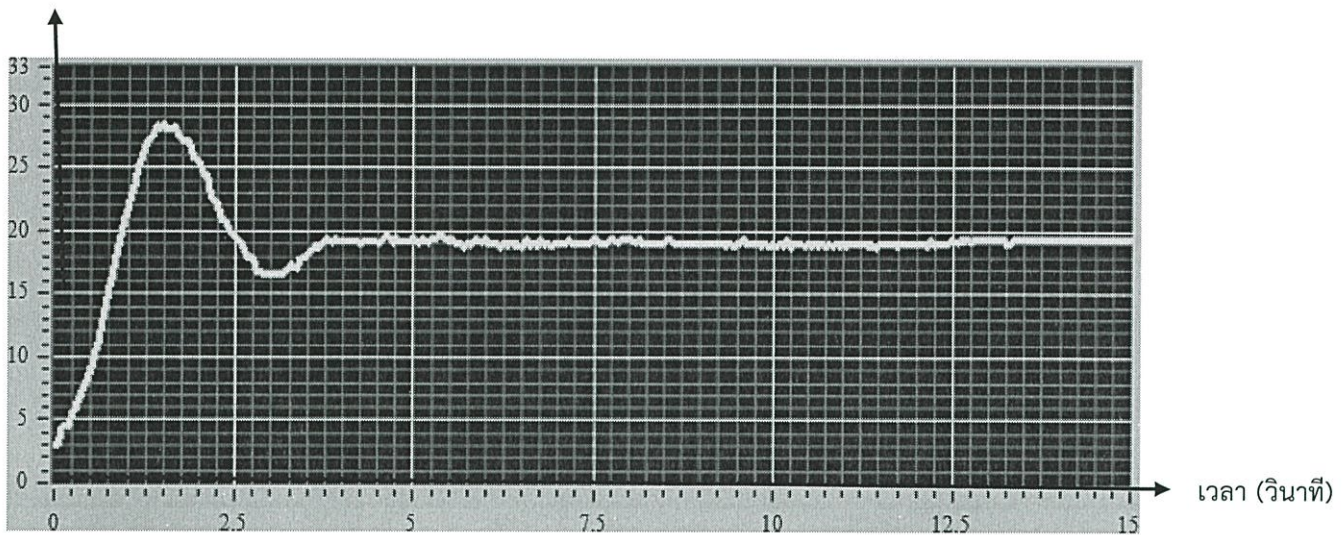
รูปที่ 4.11 กราฟแสดงความสัมพันธ์ระหว่างค่า K_I กับค่า Setting time



รูปที่ 4.12 กราฟแสดงความสัมพันธ์ระหว่างค่า K_I กับค่า Steady state error

จากผลการทดลองข้างต้นจะเห็นได้ว่า เมื่อ K_I มากขึ้นค่า Maximum overshoot มีแนวโน้มเพิ่มขึ้น ส่วนค่า Setting time มีค่าลดลงจนถึงที่ตำแหน่ง $K_I = 70$ จากนั้นถึงเพิ่มมากขึ้น ส่วนค่า Steady state error มีค่าลักษณะคล้ายกับ Setting time ซึ่งมีค่าเปลี่ยนแปลงเพียงเล็กน้อย พบว่าที่ $K_I = 70$ มีค่า Steady state error และค่า Setting time น้อยที่สุด ดังนั้นเราต้องการผลการทดลองที่ให้ค่า Steady state error ที่น้อยที่สุด เราจึงเลือกค่า $K_p = 20$, $K_I = 70$ และ $K_D = 0.06$ เพื่อมาทำการทดลองต่อไป ซึ่งกราฟการทำงานนี้สามารถแสดงให้เห็นได้ดังรูปต่อไปนี้

ระยะทาง (ซม.)

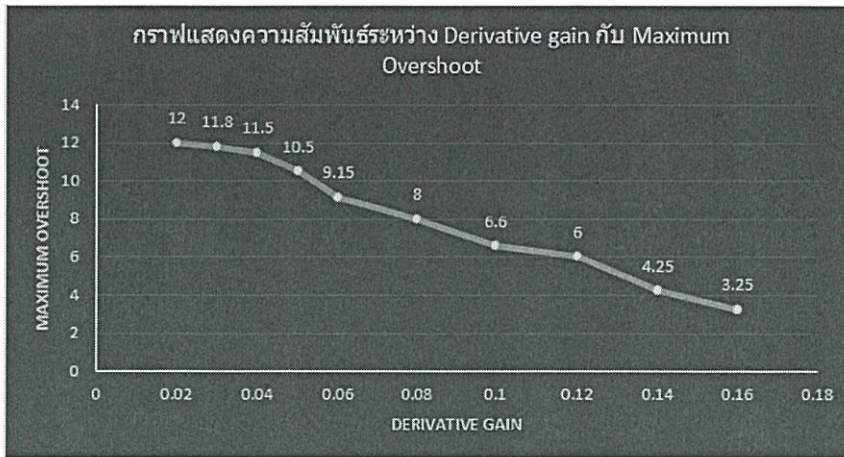
รูปที่ 4.13 กราฟการทำงานที่ $K_p = 20$, $K_I = 70$ และ $K_D = 0.06$

การทดลองที่ 4: ทำการปรับค่า K_D โดยให้ค่า K_p และ K_I คงที่ เพื่อหาความสัมพันธ์ของตัวแปร K_D ที่มีผลต่อระบบ โดยใช้วิธีการเพิ่มค่า K_D โดยกำหนดค่า Setpoint ที่ 19 เซนติเมตร

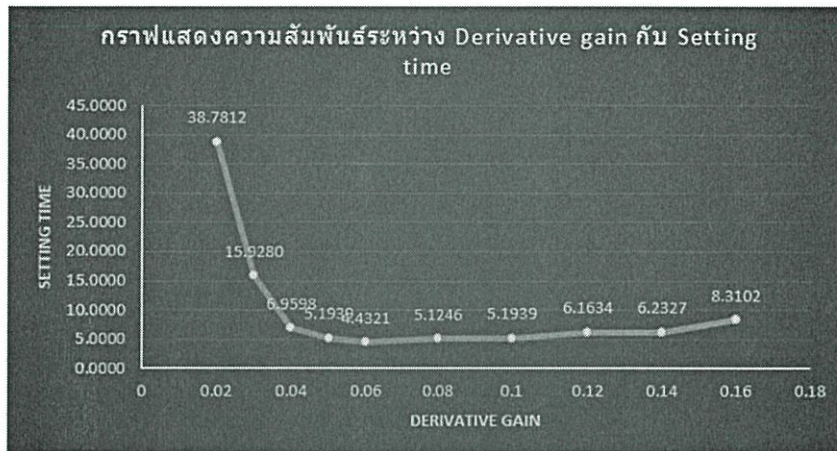
ตารางที่ 4.5 แสดงผลลัพธ์ของแต่ละค่าเมื่อเปลี่ยนค่า K_D

K_p	K_I	K_D	M_p (cm)	T_s (sec)	Output (cm)	Error (%)
20	70	0.02	12.00	38.7812	18.7906	1.1021
20	70	0.03	11.80	15.9280	18.6459	1.8637
20	70	0.04	11.50	6.9598	18.6722	1.7253
20	70	0.05	10.50	5.1939	19.1420	0.7474
20	70	0.06	9.15	4.4321	18.9187	0.4279
20	70	0.08	8.00	5.1246	19.1864	0.9811
20	70	0.10	6.60	5.1939	19.2576	1.3558
20	70	0.12	6.00	6.1634	19.4840	2.5474
20	70	0.14	4.25	6.2327	19.5129	2.6995
20	70	0.16	3.25	8.3102	19.2166	1.1400

จากตารางที่ 4.5 จะเห็นว่าเมื่อปรับค่า K_D มากขึ้นค่า Maximum overshoot มีค่าลดลง ส่วนค่า Setting time มีค่าลดลงจนถึงค่าหนึ่งแล้วจากนั้นจะมากขึ้น และค่า Output ที่อ่านได้มีค่าค่อนข้างใกล้เคียงกัน ซึ่งสามารถแสดงผลได้ในรูปแบบกราฟความสัมพันธ์ได้ดังรูปต่อไปนี้



รูปที่ 4.14 กราฟแสดงความสัมพันธ์ระหว่างค่า K_D กับค่า Maximum overshoot



รูปที่ 4.15 กราฟแสดงความสัมพันธ์ระหว่างค่า K_D กับค่า Setting time



รูปที่ 4.16 กราฟแสดงความสัมพันธ์ระหว่างค่า K_D กับค่า Steady state error

จากผลการทดลองข้างต้นจะเห็นได้ว่า เมื่อ K_D มากขึ้นค่า Maximum overshoot มีแนวโน้มลดลง ส่วนค่า Setting time มีค่าลดลงจนถึงที่ตำแหน่ง $K_D = 0.06$ จากนั้นถึงเพิ่มมากขึ้น ส่วนค่า Steady state error ยังมีค่าความผิดพลาดในหลายตำแหน่ง เพราะเมื่อเพิ่มค่า K_D ทำให้ระบบมีการตอบสนองที่ไวขึ้น และเนื่องจากมีสิ่งแวดล้อมภายนอกอย่างเช่น อุณหภูมิ, ลม เป็นต้นมากระทำกับระบบ จึงทำให้เกิดค่าความผิดพลาดมากขึ้นอีก และจากผลการทดลองพบว่าที่ $K_D = 0.06$ ลงไปมีค่า Steady state error ที่ใกล้เคียงกัน แต่เนื่องจากค่า Setting time ที่ $K_D = 0.05$ ลงไปจะมีค่าที่มากกว่าที่ $K_D = 0.06$ อย่างเห็นได้ชัด ดังนั้นจากผลการทดลองจึงเลือกค่า $K_p = 20$, $K_I = 70$ และ $K_D = 0.06$ ที่มีค่า Steady state error และค่า Setting time น้อยที่สุด ซึ่งกราฟการทำงานนี้สามารถแสดงให้เห็นจากที่ได้กล่าวมาแล้วในการทดลองที่ 3

จากผลลัพธ์ที่ได้จากการทดลองทั้งหมด 4 การทดลอง พบว่า ในการทดลองที่ 1 จะได้ค่า $K_p = 20$, $K_I = 80$ และ $K_D = 0.06$ ซึ่งเป็นการปรับค่าในช่วงระยะกว้างเพื่อหาค่าที่เหมาะสม จากนั้นในการทดลองที่ 2-4 จะทำการปรับแต่ละค่าโดยละเอียดเพื่อหาผลลัพธ์ที่ดีที่สุด ซึ่งจากผลการทดลองทั้งสามจะเห็นได้ว่าค่าที่ดีที่สุดในการทดลองนี้คือ $K_p = 20$, $K_I = 70$ และ $K_D = 0.06$ ซึ่งจะได้ผลลัพธ์ดังต่อไปนี้

Maximum overshoot = 9.15 เซนติเมตร

Setting time = 4.4321 วินาที

Steady state error = 0.4279%

ซึ่งเป็นค่าที่ดีที่สุดในการทดลองนี้ เมื่อนำผลลัพธ์มาเปรียบเทียบกับวัตถุประสงค์ที่นำมาแสดงให้เห็นในบทที่ 1 ของเล่มนี้ว่าลดค่า Steady state error ในปริมาณที่ไม่เกิน 5%

ดังนั้นค่า Steady state error ที่ได้จากการทดลองนี้มีค่าเท่ากับ 0.4279% ซึ่งมีค่าไม่เกิน 5% ซึ่งตรงตามวัตถุประสงค์

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองพบว่าค่าพารามิเตอร์ของตัวควบคุมPIDที่ดีที่สุดคือ $K_P = 20$, $K_I = 70$ และ $K_D = 0.06$ จะส่งผลให้การตอบสนองของระบบมีค่าความผิดพลาดในสภาวะคงตัว(Steady-state error) เท่ากับ 0.4065% และเวลาตั้งแต่เริ่มต้นจนกระทั่งเริ่มเข้าสู่สภาวะคงตัว(Setting time) คือ 4.4321 วินาที ซึ่งตรงตามวัตถุประสงค์ที่ตั้งไว้ โดยผลของการปรับค่าเกณฑ์แต่ละค่าโดยกำหนดให้อีก 2 ค่าคงที่

1. P-gain (K_P) : เมื่อทำการปรับค่า K_P เพิ่มขึ้น โดยที่ $K_I = 80$ และ $K_D = 0.06$ ส่งผลให้Maximum overshoot มีค่าลดลง เวลาตั้งแต่เริ่มจนกระทั่งเริ่มเข้าสู่สภาวะคงตัวมีแนวโน้มเป็นแบบ V-Shape และค่าความผิดพลาดในสภาวะคงตัวมีแนวโน้มเป็นแบบ V-Shape และจากการวิเคราะห์ผลการทดลองสรุปได้ว่าค่า $K_P=20$ เป็นค่าที่ดีที่สุด
2. I-gain (K_I) : เมื่อทำการปรับค่า K_D เพิ่มขึ้น โดยที่ $K_P= 20$ และ $K_D = 0.06$ ส่งผลให้Maximum overshoot มีค่าเพิ่มขึ้น เวลาตั้งแต่เริ่มจนกระทั่งเริ่มเข้าสู่สภาวะคงตัวมีแนวโน้มในลักษณะที่คล้ายกับแบบ V-Shape และค่าความผิดพลาดในสภาวะคงตัวอยู่ในช่วง 0-1 % และจากการวิเคราะห์ผลการทดลองสรุปได้ว่าค่า $K_I=70$ เป็นค่าที่ดีที่สุด
3. D-gain (K_D) : เมื่อทำการปรับค่า K_D เพิ่มขึ้น โดยที่ $K_P= 20$ และ $K_I = 70$ ส่งผลให้ Maximum overshoot มีค่าลดลง เวลาตั้งแต่เริ่มจนกระทั่งเริ่มเข้าสู่สภาวะคงตัวมีแนวโน้มลดลงตั้งแต่ช่วง $K_D= 0.02 - 0.06$ และมีแนวโน้มเพิ่มขึ้นตั้งแต่ $K_D= 0.06$ เป็นต้นไปและค่าความผิดพลาดในสภาวะคงตัวอยู่ในช่วง 0-3 % และจากการวิเคราะห์ผลการทดลองสรุปได้ว่าค่า $K_D=0.06$ เป็นค่าที่ดีที่สุด

5.2 ปัญหาจากการศึกษา

1. เนื่องจาก ultrasonic sensor มีความไวต่อการตอบสนองสูง รวมไปถึงมีปัจจัยที่ส่งผลต่อประสิทธิภาพในการใช้งาน เช่น อุณหภูมิ การสะท้อนของคลื่น เป็นต้น สำหรับการทดลองนี้เมื่ออุณหภูมิในการทดลองเปลี่ยนแปลงจะทำให้ความกว้างของสัญญาณแกว่งมากขึ้นเป็นผลให้มีความละเอียดในการวัดน้อย
2. ในการทดลองนี้ มีบางช่วงของการไหลไม่สม่ำเสมอ เนื่องจากมีแรงเสียดทานระหว่างผิวของวัสดุที่ใช้กึ่งกับราง
3. เนื่องจากในระหว่างการทำงานของชุดทดลองจะเกิดการสั่นของโครงสร้างขึ้นเล็กน้อย ทำให้มีแรงภายนอกมากกระทำกับวัตถุ จึงทำให้มีค่าความผิดพลาดเกิดขึ้นเล็กน้อย

5.3 ข้อเสนอแนะ

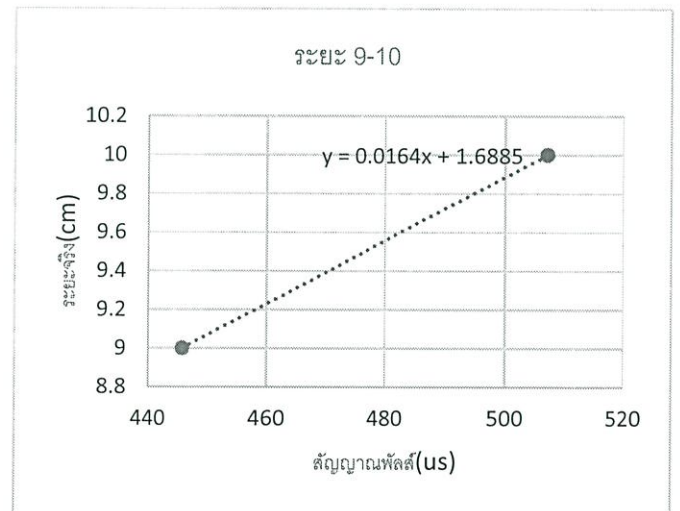
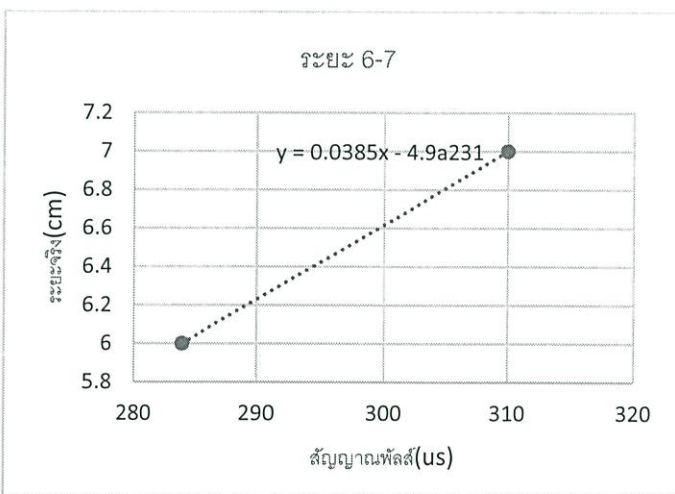
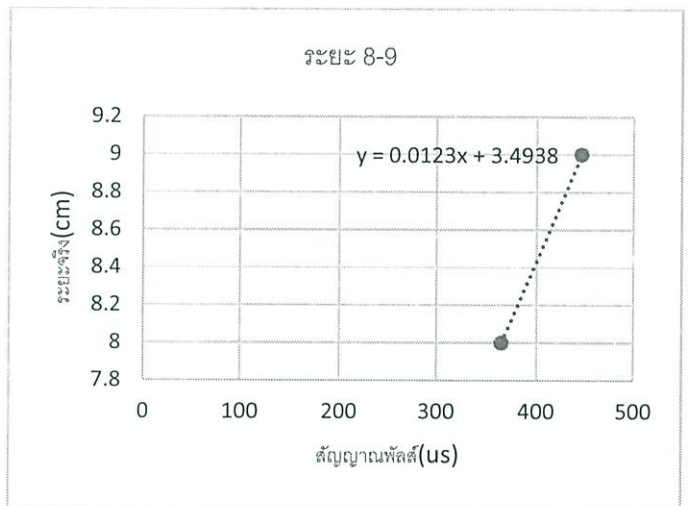
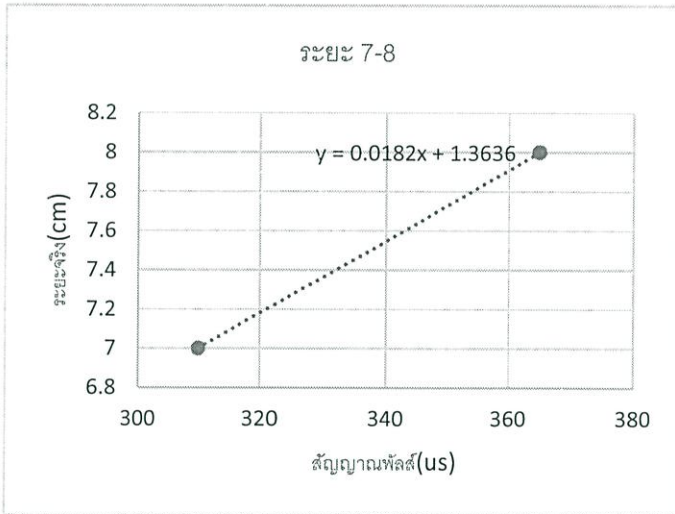
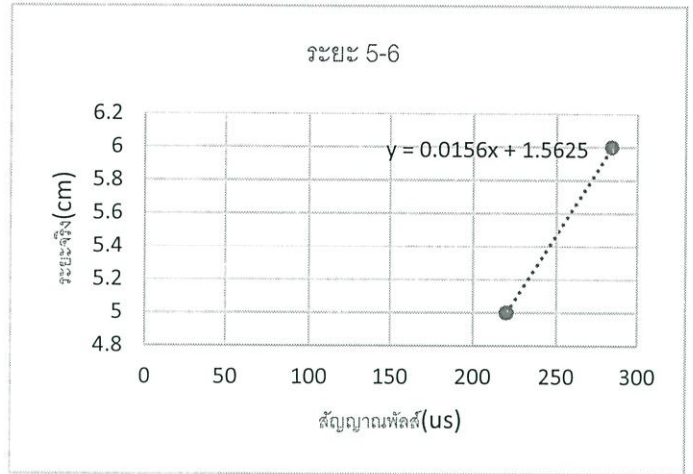
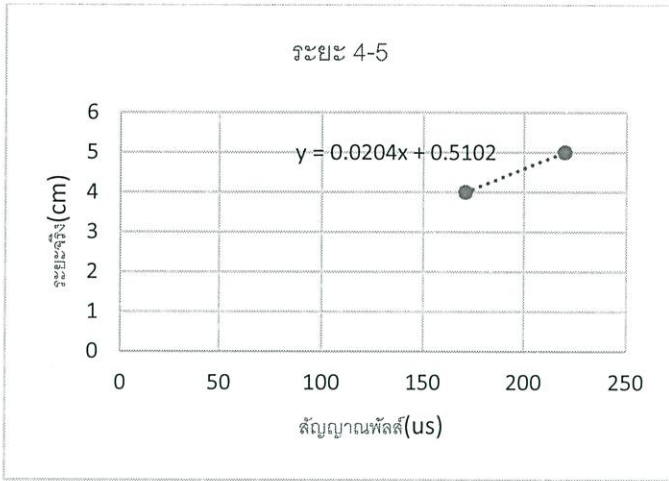
1. วัตถุที่ใช้ตรวจจับจาก ultrasonic sensor ควรมีผิวเรียบและตั้งฉากกับคลื่น เพื่อลดการกระจายจากการสะท้อนของคลื่นและง่ายต่อการทำงานของตัวรับสัญญาณ
2. ในการติดตั้ง ultrasonic sensor ควรติดตั้งให้มุมในการวัดอยู่ในช่วงไม่เกิน 15 องศา

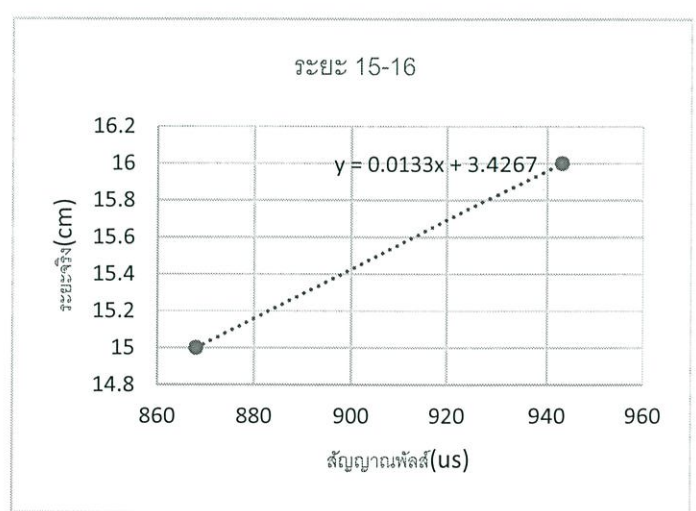
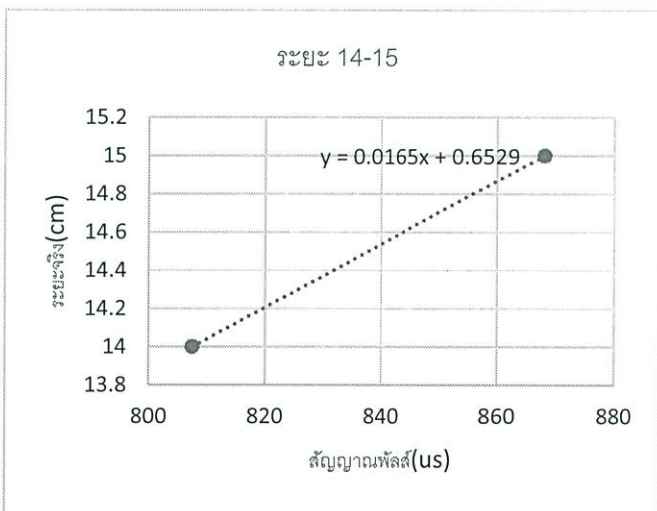
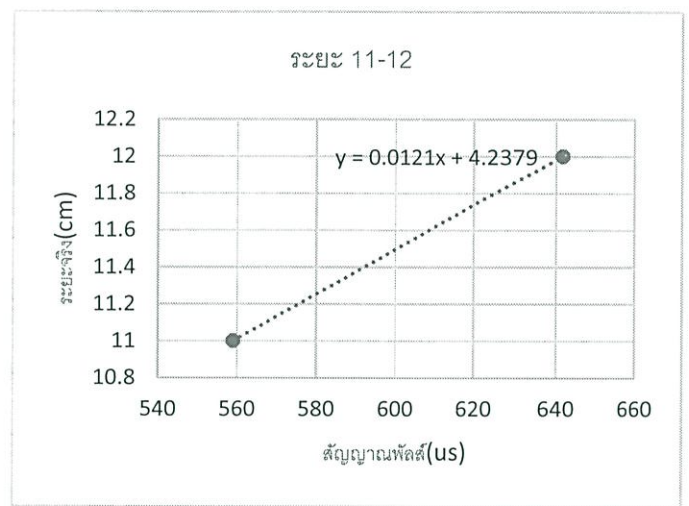
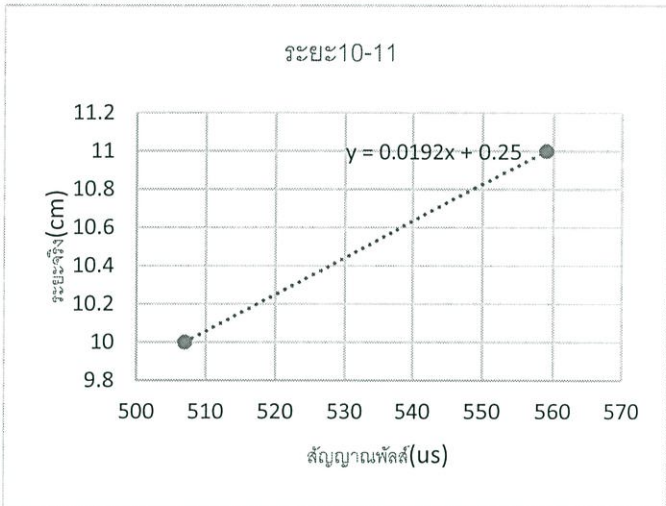
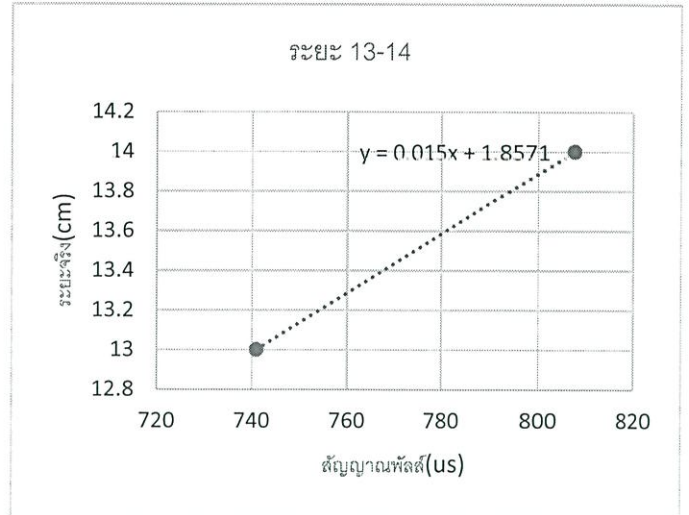
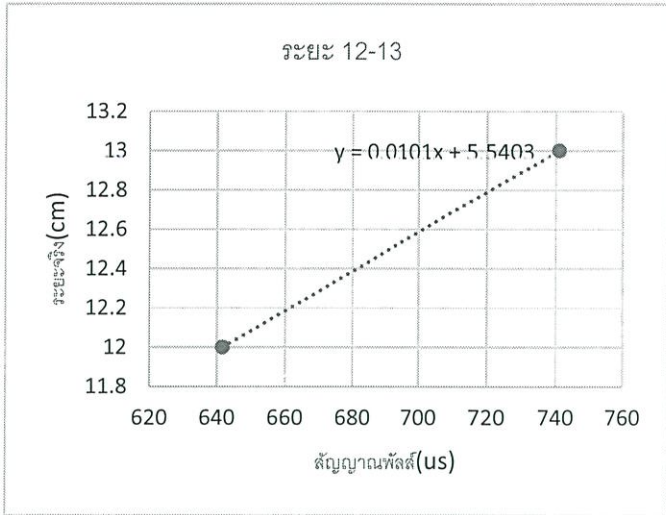
บรรณานุกรม

- [1] Evencio A. Rosales, A Ball-on-Beam Project Kit, Massachusetts Institute of Technology (2004)
- [2] Wen-Fang Xie, Modelling and control of ball and beam system using model based and non-model based control approaches , Mechanical and Industrial Engineering, Concordia University (2012)
- [3] กิจไพบุลย์ ชิวพันธุ์ศรี, การออกแบบแอฟพลีเคชันในระบบกราฟิกด้วย LabVIEW, กรุงเทพฯ, สำนักพิมพ์ ซี เอ็ดดูเคชัน, 2554
- [4] William Palm III, System Dynamics(Third edition), McGraw-Hill Education(2014)
- [5] <https://www.arduitronics.com/product/20/ultrasonic-sensor-module-hc-sr04>
- [6] <https://www.thaieasyelec.com/article-wiki/basic-electronics>

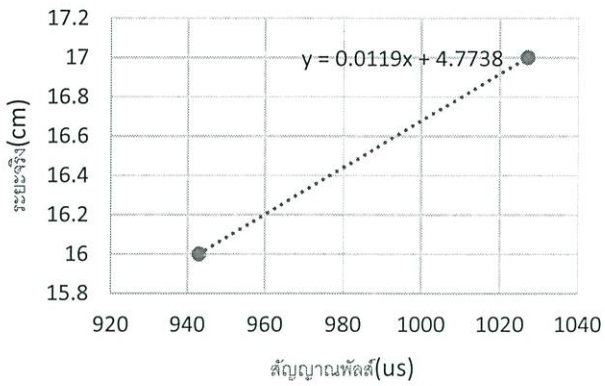
ภาคผนวก ก

กราฟการสอบเทียบ ultrasonic sensor

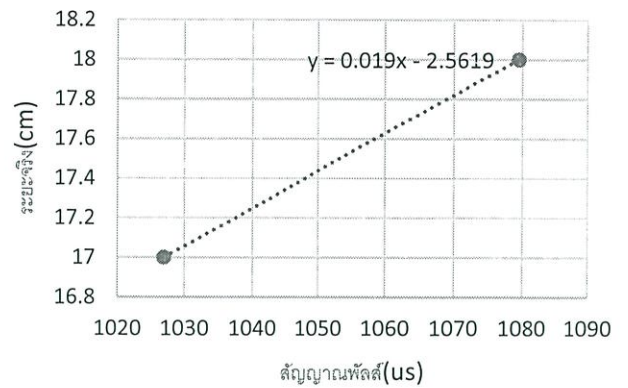




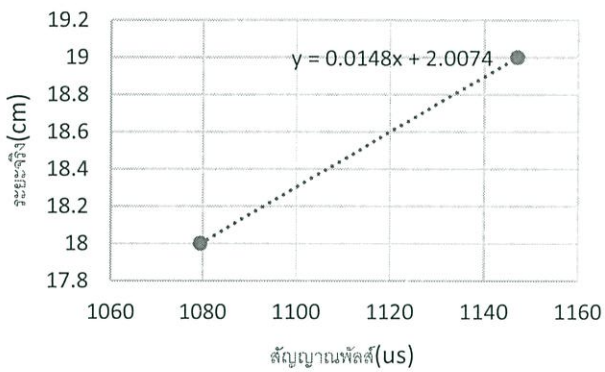
ระยะ 16-17



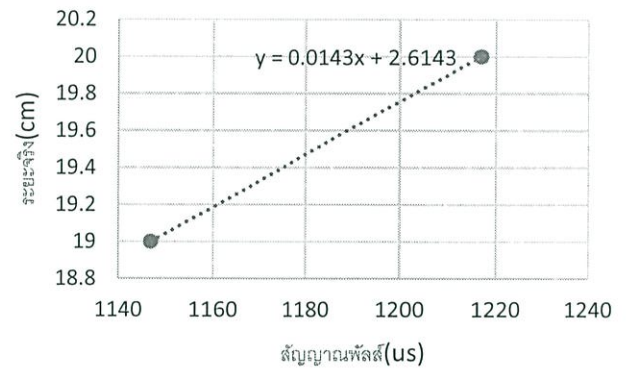
ระยะ 17-18



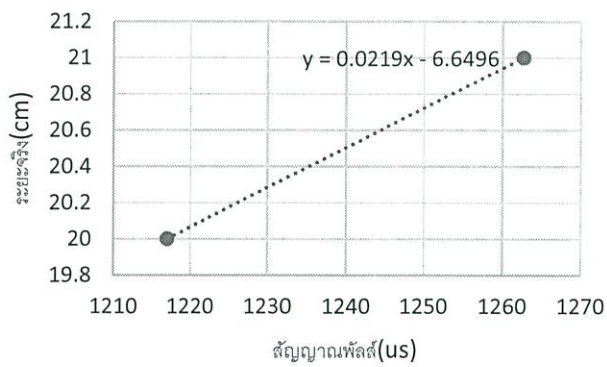
ระยะ 18-19



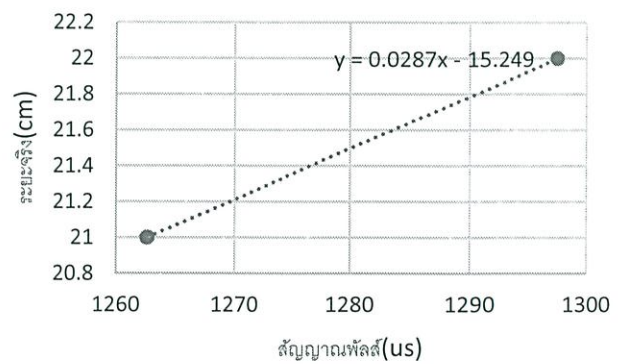
ระยะ 19-20

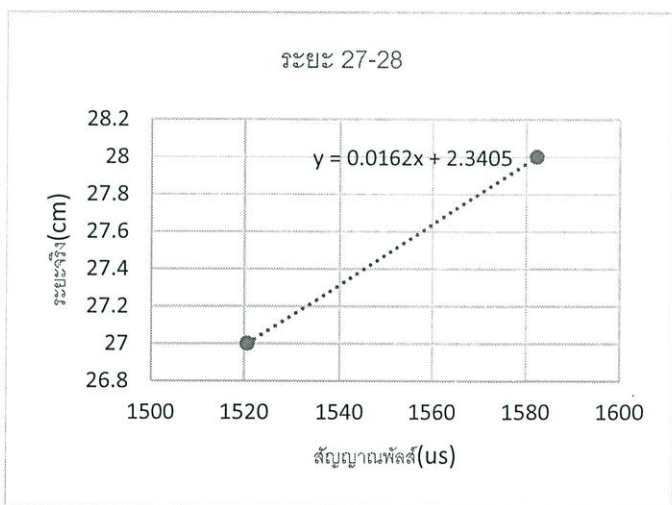
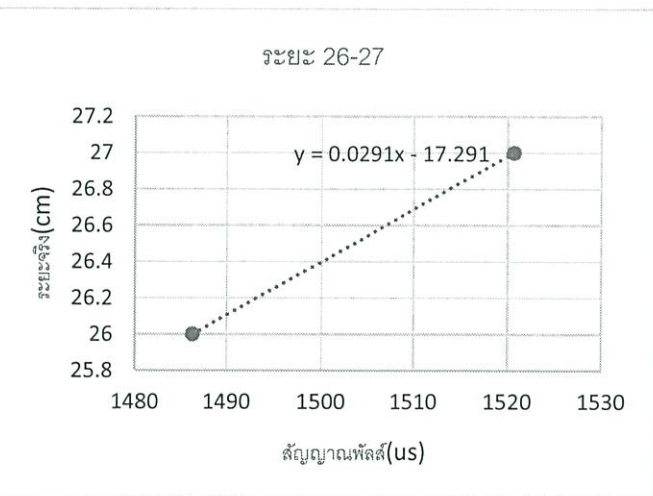
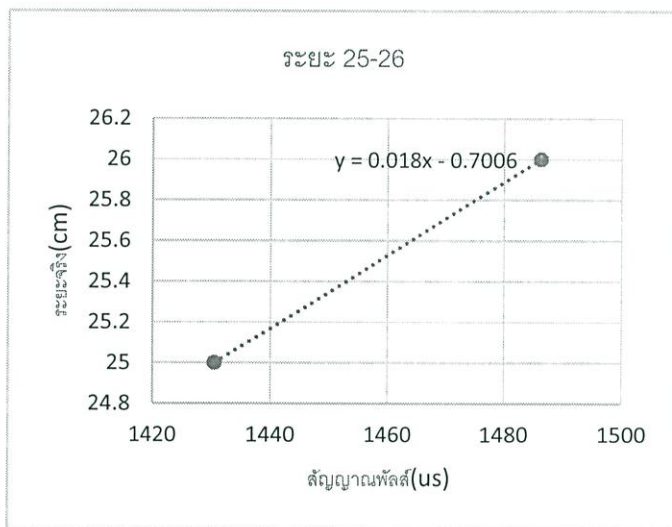
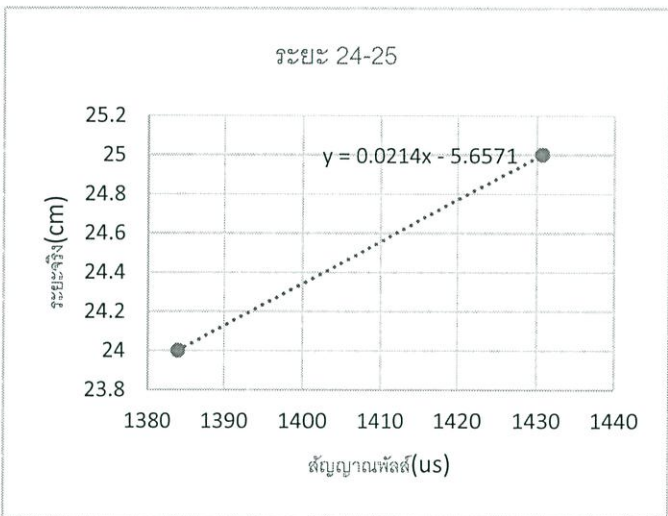
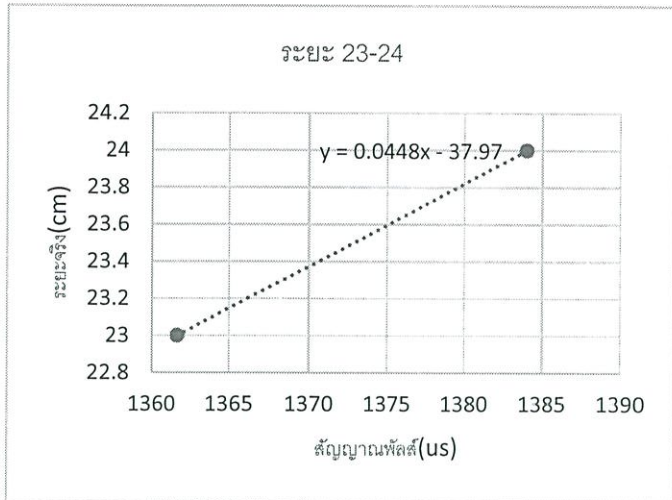
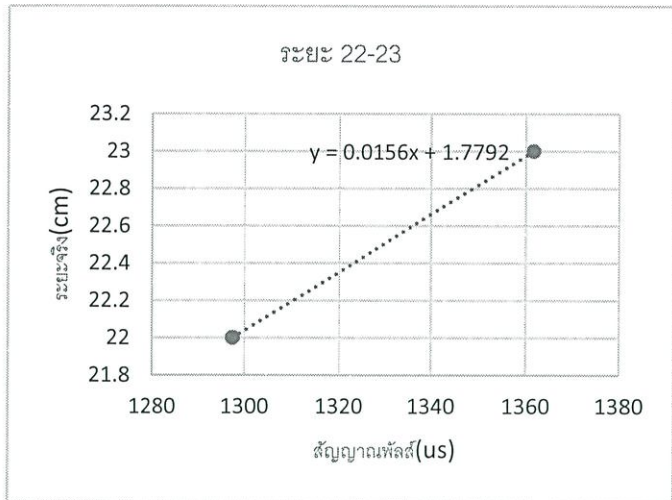


ระยะ 20-21

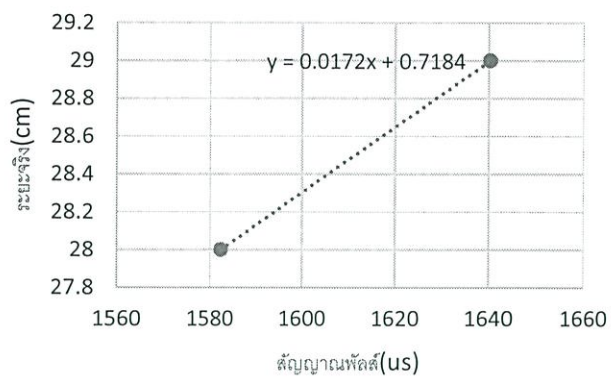


ระยะ 21-22

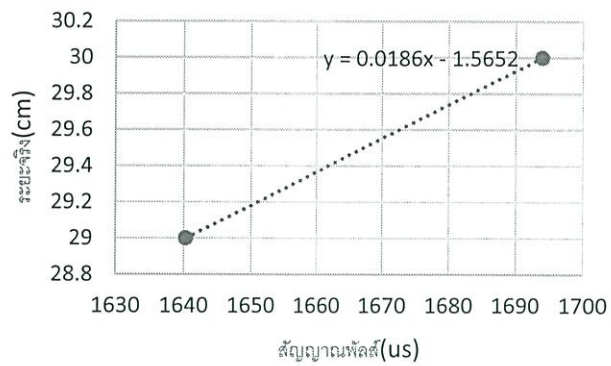




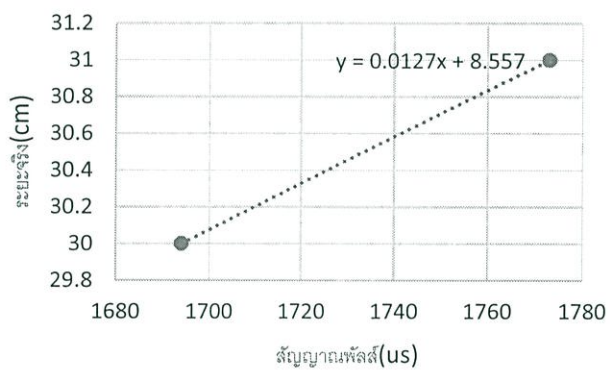
ระยะ 28-29



ระยะ 29-30

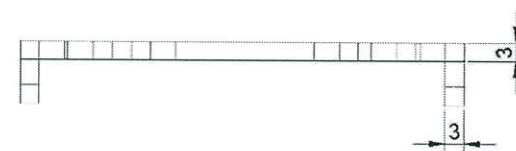
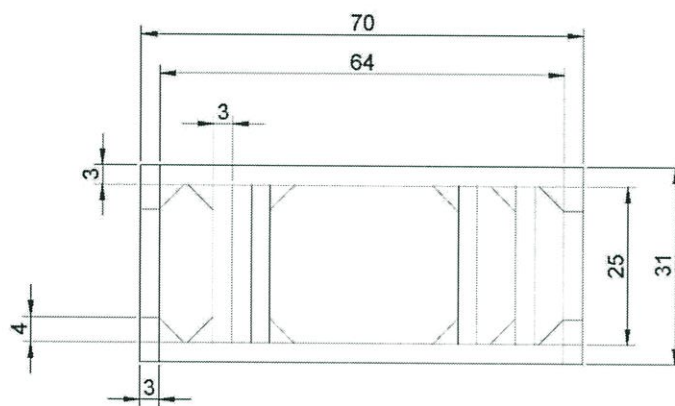
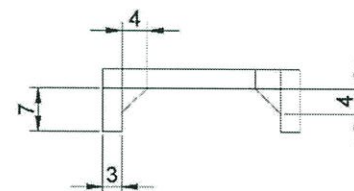
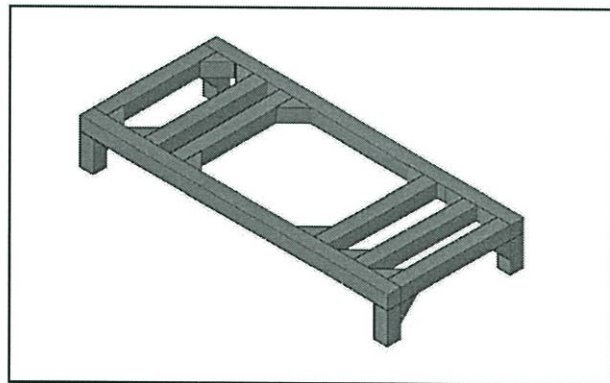


ระยะ 30-31

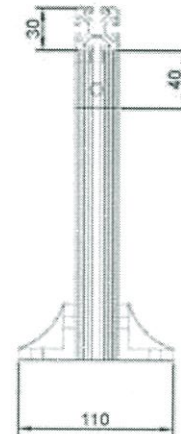
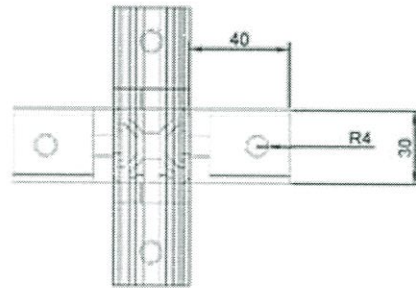
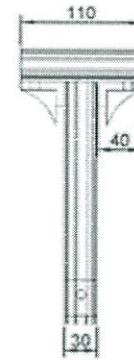
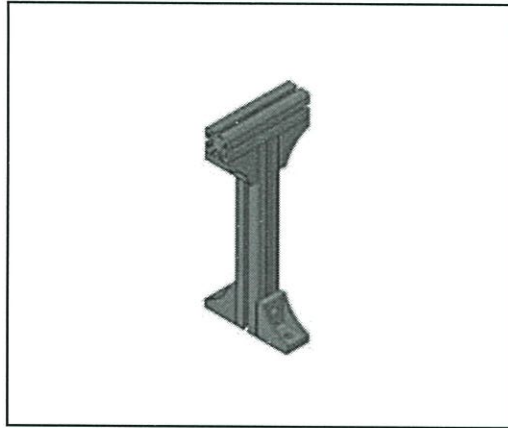


ภาคผนวก ข

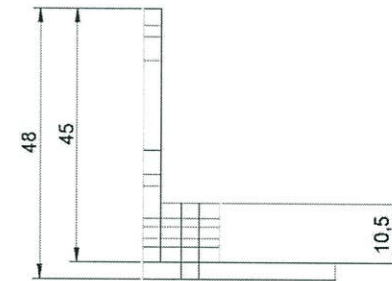
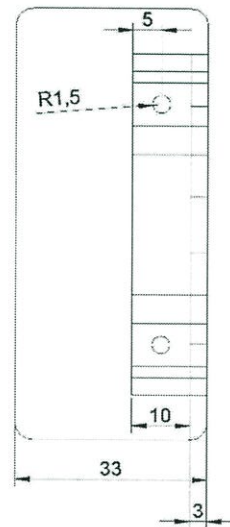
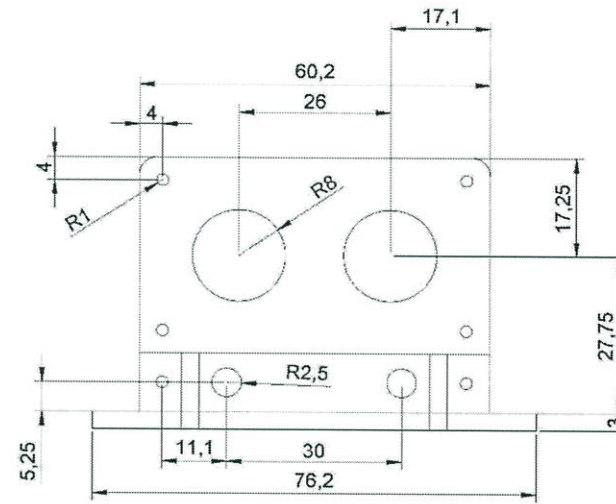
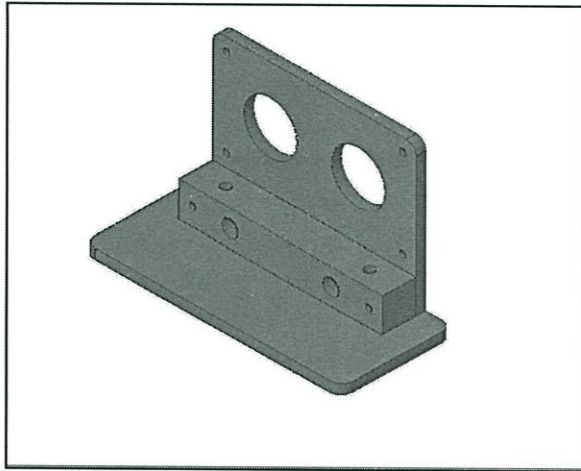
แบบโครงสร้างของชุดทดลอง



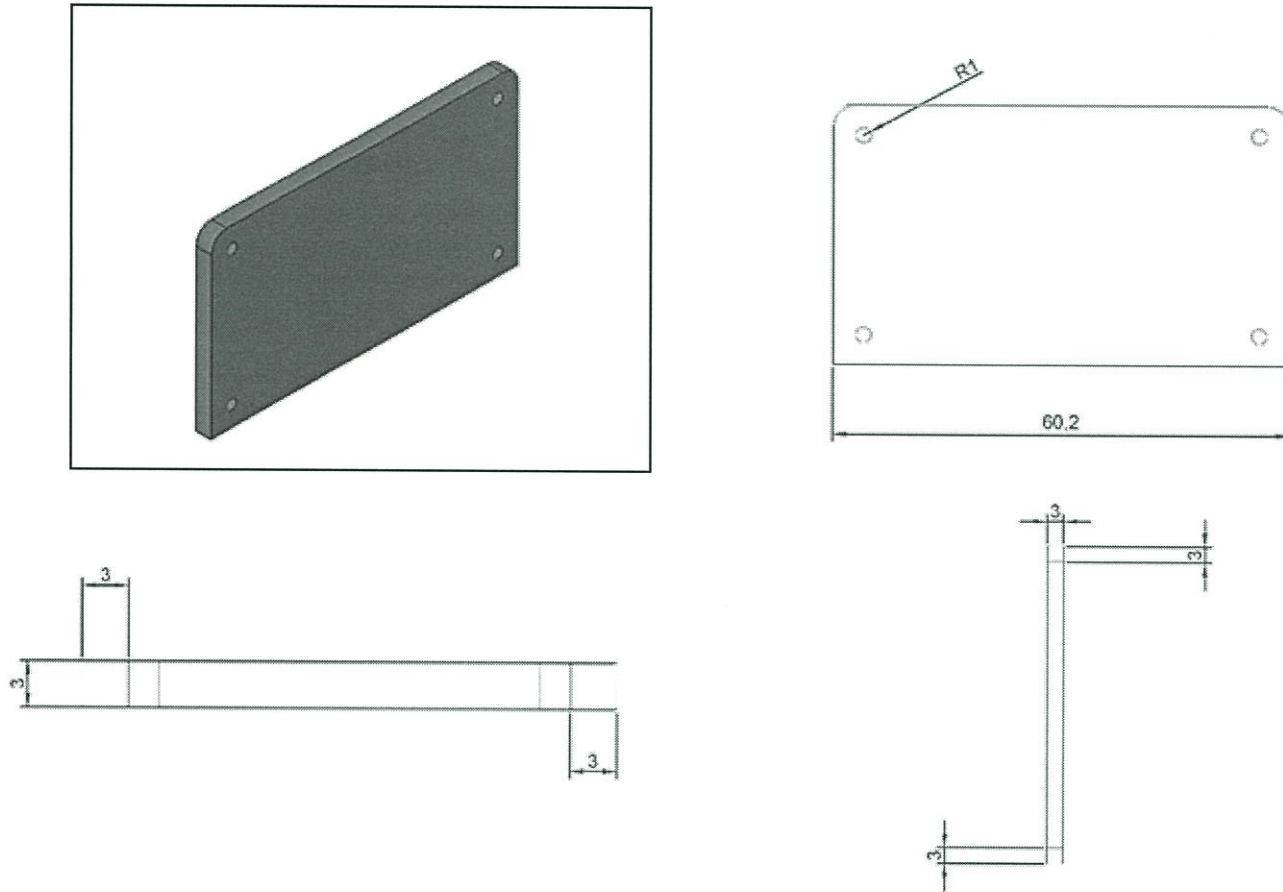
รูป ก-1 แบบฐานของชุดทดลอง (หน่วยมิลลิเมตร)



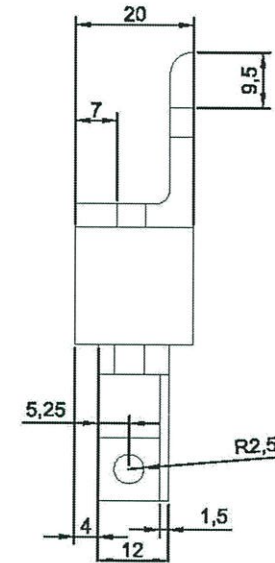
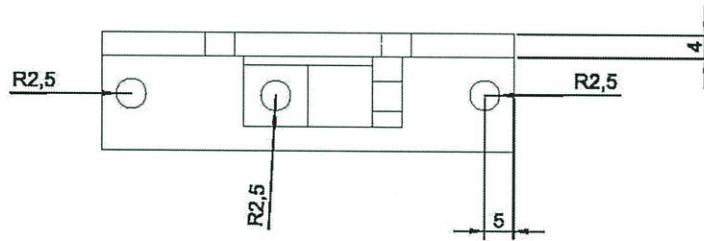
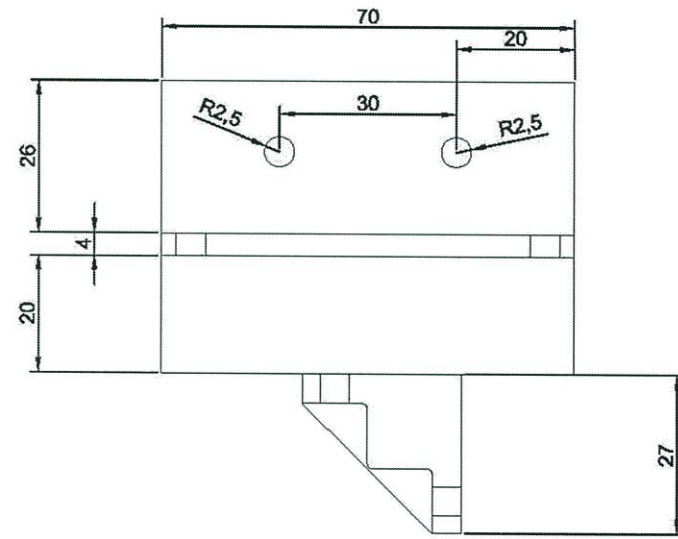
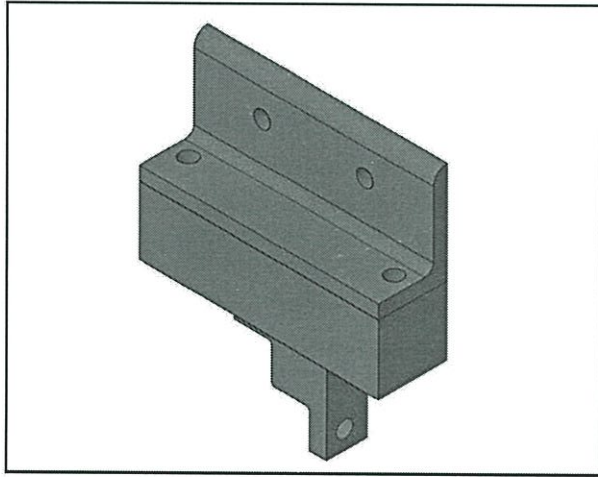
รูป ก-2 แบบเสาหลักของชุดทดลอง (หน่วยมิลลิเมตร)



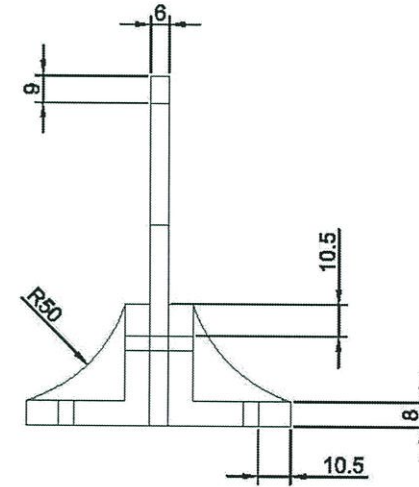
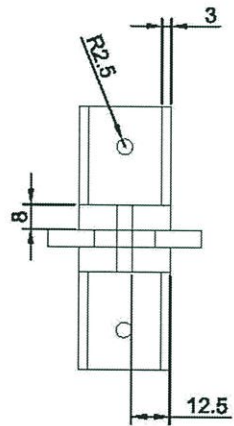
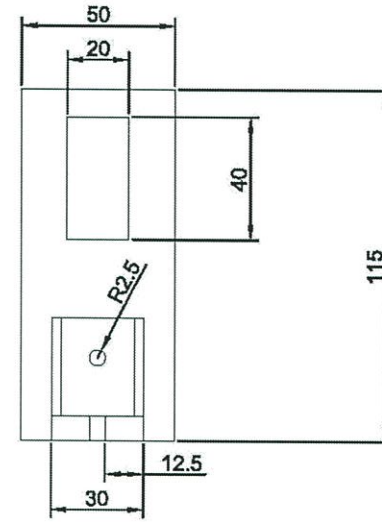
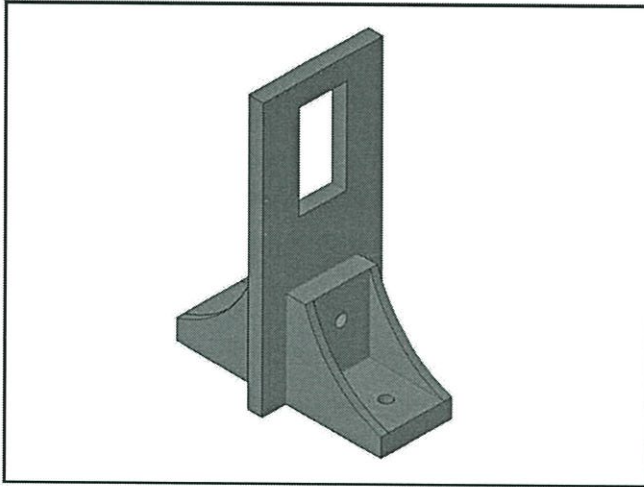
รูป ก-3 แบบฐานสำหรับใส่ ultrasonic sensor (หน่วยมิลลิเมตร)



รูป ก-4 แบบแผ่นสำหรับปิดที่ตำแหน่งข้างหลังของ ultrasonic sensor (หน่วยมิลลิเมตร)



รูป ก-5 แบบส่วนปลายของรางสำหรับกันลูกกลิ้ง (หน่วยมิลลิเมตร)



รูป ก-6 แบบฐาน servo motor (หน่วยมิลลิเมตร)