



โครงการการออกแบบและการสร้างตุ้มผกผัน ควบคุมโดยระบบพีไอดี

Inverted Pendulum Design Using a PID Control

นายพงษ์พัฒน์ วันทองสังข์ รหัส 52362779
นายสุรศักดิ์ เพชรสุภาพร รหัส 52363097

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 20 ก.ค. 2555
เลขทะเบียน..... 16861192
เลขเรียกหนังสือ..... 915
มหาวิทยาลัยนเรศวร W161

2055



ใบรับรองปริญญาโท

หัวข้อโครงการ การออกแบบและการสร้างลูกตุ้มผกผัน ความคุม โดยระบบพีไอดี
ผู้ดำเนินโครงการ นายพงษ์พัฒน์ วันทองสังข์ รหัส 52362779
นายสุรศักดิ์ เพชรสุภาพร รหัส 52363097
อาจารย์ที่ปรึกษา ดร. พันธ์ นัฏฤทธิ
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2555

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงการ
(ดร. พันธ์ นัฏฤทธิ)

.....กรรมการ
(ดร. สุวิทย์ กิระวิทยา)

.....กรรมการ
(อาจารย์เศรษฐา ตั้งคำวานิช)

.....กรรมการ
(ดร. พรพิศุทธิ์ วรจิรันคน)

หัวข้อโครงการ การออกแบบและการสร้างระบบลูกตุ้มผกผัน ควบคุมโดยระบบพีไอดี

ผู้ดำเนินโครงการ นายพงษ์พัฒน์ วันทองสังข์ รหัส 52362779

นายสุรศักดิ์ เพชรสุภาพร รหัส 52363097

อาจารย์ที่ปรึกษา คร.พนัส นัดฤทธิ

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2555

บทคัดย่อ

ระบบลูกตุ้มผกผัน มีความสำคัญด้านการศึกษา ระบบควบคุมและเป็นระบบที่ไม่มีเสถียรภาพ จำเป็นต้องอาศัยตัวควบคุม โดยการทำงานของระบบควบคุมลูกตุ้มผกผันประกอบด้วยลูกตุ้มที่หมุนได้อย่างอิสระบนรถที่เคลื่อนที่ได้ในแนวระดับที่ถูกควบคุมด้วยมอเตอร์ไฟฟ้า ซึ่งในทางปฏิบัตินั้นเป็นเรื่องยากที่จะสามารถควบคุมให้ลูกตุ้มทรงตัวในแนวตั้งได้ ในโครงการนี้จึงได้มีความสนใจที่จะนำเอาทฤษฎีควบคุมแบบพีไอดีมาช่วยให้การทำงานของระบบลูกตุ้มผกผันมีเสถียรภาพในการทรงตัวมากขึ้น

Project title Inverted Pendulum Design Using a PID Control

Name Mr. Phongphat Wanthongsang ID. 52362779

 Mr. Surasak Phetsupaporn ID. 52363097

Project advisor DR. Panus Nattharith

Major Computer Engineering

Department Electrical and Computer Engineering

Academic year 2012

Abstract

An inverted pendulum system is important in control engineering study. It is an unstable system that needs a controller to achieve its stability. An inverted pendulum system consists of a pendulum and a cast that is actuated from a DC motor. In practice it is difficult to control a pendulum stabilized in vertical. In this project will present theory of PID controller that can stabilize the system.

กิตติกรรมประกาศ

โครงการฉบับนี้เป็นการศึกษาเกี่ยวกับเรื่องการออกแบบและสร้างระบบลูกคัมผกผัน
ควบคุมโดยระบบพีไอดี ซึ่งได้รับความช่วยเหลือจากบุคคลดังต่อไปนี้

ขอขอบพระคุณ ดร. พันธ์ นัถฤทธิ์ อาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
มหาวิทยาลัยนเรศวร ผู้เป็นอาจารย์ที่ปรึกษาโครงการที่ได้ให้ความรู้ ให้คำแนะนำและให้ความ
ช่วยเหลือแก่คณะผู้จัดทำเป็นอย่างดีตลอดมา

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวรทุก
ท่าน ที่ได้ให้ความรู้และให้คำสั่งสอนจนคณะผู้จัดทำมีความรู้ที่สามารถนำมาประยุกต์ใช้ในการ
จัดทำ โครงการในครั้งนี้

และที่สำคัญที่สุดขอขอบพระคุณบิดา มารดา ที่ได้เลี้ยงดูและอบรมสั่งสอนแก่คณะผู้จัดทำ
จนทำให้คณะผู้จัดทำทุกคนมีวันนี้ได้ ซึ่งเป็นพระคุณอันหาที่เปรียบไม่ได้

ท้ายนี้คณะผู้จัดทำใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องกับทุกท่านที่ไม่ได้กล่าวนามมา ณ
ที่นี้ ที่มีส่วนร่วมในการให้ข้อมูล เป็นที่ปรึกษาในการทำปริญญาานิพนธ์ฉบับนี้จนเสร็จสมบูรณ์
คณะผู้จัดทำ จึงขอขอบพระคุณไว้ ณ ที่นี้

คณะผู้จัดทำโครงการ

นายพงษ์พัฒน์ วันทองสังข์ รหัส 52362779

นายสุรศักดิ์ เพชรสุภาพร รหัส 52363097

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของ โครงการงาน.....	1
1.2 วัตถุประสงค์โครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 แผนการดำเนินงาน.....	3
1.6 ผลที่คาดว่าจะได้รับ.....	4
1.7 รายละเอียดงบประมาณของโครงการ.....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	1
2.1 หลักการของลูกคัมพกผัน.....	5
2.2 ระบบควบคุมที่ไอดี.....	5
2.3 บอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE AVR EASY88.....	10
2.4 MEMSIC2125 โมดูลวัดความเร่งแบบ 2 แกน.....	11
2.5 มอเตอร์กระแสตรง.....	12
บทที่ 3 วิธีการดำเนินการ.....	14
3.1 ภาพรวมของระบบ.....	14

สารบัญ (ต่อ)

หน้า

3.2 การสร้างระบบลูกตุ้มผกผันเพื่อใช้ในการทดลอง โครงงาน.....	15
3.3 การเขียนโปรแกรมบนบอร์ด ไมโครโปรเซสเซอร์เพื่อใช้ในการอ่านค่าจากเซนเซอร์ที่ ใช้สำหรับวัดมุมเอียงในแนวแกน X (Roll).....	17
3.4 การปรับค่า K_p , K_i และ K_d ด้วยมือ.....	22
3.5 บทสรุป.....	23
บทที่ 4 การทดลองและผลการทดลอง.....	24
4.1 ระบบที่ออกแบบและสร้างขึ้น.....	24
4.2 ผลการทดลองการวัดมุมเอียงของเซนเซอร์ MEMSIC2125.....	25
4.3 ผลการทดลองปรับค่า K_p , K_i และ K_d ด้วยวิธีการลองผิดลองถูก.....	27
4.4 บทสรุป.....	42
บทที่ 5 สรุปผลการทดลอง.....	43
5.1 สรุปผลการทดลอง.....	43
5.2 วิจารณ์ผลการทดลอง.....	44
5.3 การพัฒนาโครงงานต่อไปในอนาคต.....	45
เอกสารอ้างอิง.....	46
ภาคผนวก.....	47
ภาคผนวก ก.....	48
ภาคผนวก ข.....	56
ภาคผนวก ค.....	62
ภาคผนวก ง.....	71
ประวัติผู้เขียน โครงงาน.....	73

สารบัญตาราง

ตารางที่	หน้า
2.1 ผลของการเพิ่มค่าตัวแปรอย่างอิสระ.....	10
4.1 แสดงผลการทดลอง ค่าเฉลี่ย และค่าความคลาดเคลื่อนของการทดลอง ของการวัดมุมเอียงใน แนวแกน X.....	26
1 แสดงค่า g ที่เอาต์พุต X_{out} หรือ Y_{out} ของโมดูล MXD2125 ที่มุมเอียงค่าต่างๆ.....	60
2 แสดงค่าความผิดพลาดที่เกิดขึ้นจากการตรวจจับความลาดเอียงของ โมดูล MXD2125.....	61



สารบัญรูป

รูปที่	หน้า
2.1 ภาพรวมการทำงานของระบบ.....	5
2.2 ระบบลูกตุ้มผกผัน.....	6
2.3 ตำแหน่งที่ติดตั้งโมดูลวัดความเร่งแบบ 2 แกน.....	6
2.4 ตัวควบคุม PID ที่ต่อเข้ากับระบบแบบอนุกรม.....	7
2.5 ตัวควบคุม PID.....	7
2.6 ภาพรวมของระบบ.....	9
2.7 บอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE AVR EASY88.....	10
2.8 MEMSIC2125 โมดูลวัดความเร่งแบบ 2 แกน.....	11
2.9 ไอซีควบคุมมอเตอร์กระแสตรง L293D.....	12
3.1 ภาพรวมของระบบ.....	15
3.2 รางเลื่อนที่ทำมาจากแผ่นอะคริลิก.....	16
3.3 รางเลื่อนที่ทำมาจากแผ่นรางเครื่องพิมพ์ Canon.....	16
3.4 รางเคลื่อนพิมพ์แสดงมุมมองด้านหน้า.....	17
3.5 รางเคลื่อนพิมพ์แสดงมุมมองด้านบน.....	17
3.6 วงจรควบคุมมอเตอร์.....	18
3.7 แสดงการทำงานของตัวค้ำทานสำหรับปรับค่า K_p , K_i และ K_d	19
3.8 รูประบบควบคุม PID.....	20
4.1 ระบบที่ออกแบบและสร้างขึ้น.....	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.2 วิธีการทดลองวัดมุมเอียงในแนวแกน X.....	25
4.3 แสดงความสัมพันธ์ของค่ามุมเอียงในแนวแกน X ณ วินาทีที่ 1-10.....	27
4.4 ตัวคำนวณปรับค่าได้ที่ใช้ในการปรับค่า K_p , K_i และ K_d	28
4.5 แสดงการเคลื่อนที่ของรถตามทิศทางมุมเอียง.....	28
4.6 กราฟแสดงค่า $K_p = 13$, $K_i = 0$, $K_d = 0$	29
4.7 กราฟแสดงค่า $K_p = 23$, $K_i = 0$, $K_d = 0$	29
4.8 กราฟแสดงค่า $K_p = 33$, $K_i = 0$, $K_d = 0$	30
4.9 กราฟแสดงค่า $K_p = 43$, $K_i = 0$, $K_d = 0$	30
4.10 กราฟแสดงค่า $K_p = 53$, $K_i = 0$, $K_d = 0$	31
4.11 กราฟแสดงค่า $K_p = 63$, $K_i = 0$, $K_d = 0$	31
4.12 กราฟแสดงค่า $K_p = 126$ $K_i = 0$, $K_d = 0$	32
4.13 กราฟแสดงค่า $K_p = 178$ $K_i = 0$, $K_d = 0$	32
4.14 กราฟแสดงค่า $K_p = 252$ $K_i = 0$, $K_d = 0$	33
4.15 กราฟแสดงค่า $K_p = 311$ $K_i = 0$, $K_d = 0$	33
4.16 กราฟแสดงค่า $K_p = 367$ $K_i = 0$, $K_d = 0$	34
4.17 กราฟแสดงค่า $K_p = 456$ $K_i = 0$, $K_d = 0$	34
4.18 กราฟแสดงค่า $K_p = 456$ $K_i = 0$, $K_d = 0$	35
4.19 กราฟแสดงค่า $K_p = 456$ $K_i = 1$, $K_d = 0$	35

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.20 กราฟแสดงค่า $K_p = 456$ $K_i = 2$, $K_d = 0$	36
4.21 กราฟแสดงค่า $K_p = 456$ $K_i = 1$, $K_d = 30$	37
4.22 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 61$	37
4.23 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 116$	38
4.24 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 150$	38
4.25 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 194$	39
4.26 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 321$	39
4.27 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 405$	40
4.28 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 553$	40
5.1 รวงเลื่อนเชิงเส้น (Linear slide).....	45
5.2 ภาพรวมการทำงานที่เกิด Delay ของระบบขึ้น.....	45
1 รูปร่างและการจัดขาของ MXD2125 โมดูลวัดความเร่ง.....	57
2 แสดงไดอะแกรมการทำงานภายในโมดูล MXD2125.....	57
3 แสดงทิศทางการเร่งในแนวแกน X และ Y ที่ตัวตรวจจับความเร่งวัดได้ เมื่อมองจากด้านบนของตัวตรวจจับ.....	58
1 แสดง โครงสร้างของบอร์ด ET-BASE AVR EASY88.....	65
1 IC L293D.....	72
2 ตารางแสดงการทำงานของ IC L293D.....	72

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ระบบลูกตุ้มผกผัน เป็นเครื่องมือที่ได้รับความนิยมในการศึกษาระบบควบคุม ในงานทางด้านวิศวกรรม เนื่องจากระบบลูกตุ้มผกผันเป็นระบบที่มีข้อศึกษาอยู่หลายประการ เช่น เป็นระบบที่ไม่เป็นเชิงเส้น เป็นระบบที่ไม่มีความเสถียรภาพ เป็นต้น ดังนั้นระบบดังกล่าวจึงเหมาะแก่การนำมาศึกษาเพื่อออกแบบตัวควบคุม สำหรับการทำงานของระบบควบคุมนั้นจะต้องทำการรักษาสมดุลของลูกตุ้ม เพื่อให้ลูกตุ้มสามารถตั้งอยู่ได้ในแนวตั้งตลอดเวลา โดยระบบจะต้องส่งแรงให้ฐานของลูกตุ้มมีการเคลื่อนที่ด้วยแรง, ความเร็วและเวลาที่เหมาะสม หากลูกตุ้มจะล้มไปทางด้านซ้าย ฐานของลูกตุ้มก็จะเคลื่อนตัวไปทางด้านซ้ายเพื่อเลี้ยงให้ลูกตุ้มตั้งอยู่ในแนวตั้งเดิม ในทำนองเดียวกันเมื่อลูกตุ้มจะล้มไปทางด้านขวาฐานของลูกตุ้มก็จะเคลื่อนตัวไปทางด้านขวาเพื่อเลี้ยงให้ลูกตุ้มตั้งอยู่ในแนวตั้งเดิม เมื่อฐานของลูกตุ้มมีการเคลื่อนที่กลับไปกลับมา ก็จะเกิดแรงเหวี่ยงของตัวลูกตุ้ม แรงเหวี่ยงนี้จะช่วยให้ลูกตุ้มสามารถตั้งอยู่ในแนวตั้งได้

อย่างไรก็ตาม ในทางปฏิบัตินั้นเป็นเรื่องยากที่จะสามารถควบคุมให้ลูกตุ้มทรงตัวอยู่ในแนวตั้ง ดังนั้นในโครงการนี้ผู้ดำเนินโครงการจึงมีความสนใจที่จะนำเอาทฤษฎีควบคุมมาประยุกต์ใช้ในการออกแบบตัวควบคุมเพื่อช่วยให้การทำงานของระบบลูกตุ้มผกผันมีเสถียรภาพในการทรงตัวมากขึ้น ซึ่งตัวควบคุมที่ผู้ดำเนินโครงการเลือกใช้ก็คือ ตัวควบคุมแบบพีไอดี

ตัวควบคุมแบบพีไอดี (PID controller) เป็นระบบควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการ และค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของ PID ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ วิธีคำนวณของ PID ขึ้นอยู่กับสามตัวแปรคือค่าสัดส่วน, ปริพันธ์ และ อนุพันธ์ ค่าสัดส่วนกำหนดจากผลของความผิดพลาดในปัจจุบัน, ค่าปริพันธ์กำหนดจากผลบนพื้นฐานของผลรวมความผิดพลาดที่ซึ่งพุ่งผ่านไป, และค่าอนุพันธ์กำหนดจากผลบนพื้นฐานของอัตราการเปลี่ยนแปลงของค่าความผิดพลาด น้ำหนักที่เกิดจาก

การรวมกันของทั้งสามนี้จะใช้ในการปรับกระบวนการ โดยการปรับค่าคงที่ใน PID ตัวควบคุมสามารถปรับรูปแบบการควบคุมให้เหมาะกับที่กระบวนการต้องการได้ การตอบสนองของตัวควบคุมจะอยู่ในรูปของการไหวตัวของตัวควบคุมจนถึงค่าความผิดพลาด ค่าโอเวอร์ชูต (overshoots) และ ค่าแกว่งของระบบ (oscillation) การประยุกต์ใช้งานบางครั้งอาจใช้เพียงหนึ่งถึงสองรูปแบบ ขึ้นอยู่กับกระบวนการเป็นสำคัญ พีไอดีบางครั้งจะถูกเรียกว่าการควบคุมแบบ PI, PD, P หรือ I ขึ้นอยู่กับว่าใช้รูปแบบใดบ้าง

1.2 วัตถุประสงค์โครงการ

1. เพื่อการศึกษาการทำงานของระบบลูกตุ้มผกผัน
2. เพื่อนำความรู้ทางด้าน วงจรอิเล็กทรอนิกส์, ไมโครคอนโทรลเลอร์และการเชื่อมต่อกับอุปกรณ์ ภายนอกมาใช้ในการออกแบบและสร้างระบบลูกตุ้มผกผัน
3. เพื่อนำความรู้ทางด้านระบบควบคุม มาใช้ในการควบคุมระบบลูกตุ้มผกผัน

1.3 ขอบเขตของโครงการ

1. ออกแบบและสร้างระบบลูกตุ้มผกผัน
2. ออกแบบตัวควบคุมแบบพีไอดี สำหรับควบคุมลูกตุ้มผกผัน
3. ระบบลูกตุ้มผกผันที่พัฒนาขึ้นสามารถทรงตัวอยู่ในแนวตั้งได้

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาการทำงานของลูกตุ้มผกผัน
2. ออกแบบและสร้างระบบลูกตุ้มผกผัน
3. ศึกษาการทำงานของบอร์ดไมโครคอนโทรลเลอร์ตระกูล PIC
4. สร้างวงจรขับเคลื่อนมอเตอร์
5. ศึกษาและออกแบบ PID Control ที่ใช้ในการควบคุมระบบลูกตุ้มผกผัน
6. ทดสอบการทำงานของลูกตุ้มผกผัน
7. จัดทำเอกสารและคู่มือการใช้งาน

1.6 ผลที่คาดว่าจะได้รับ

1. ได้ระบบต้นแบบของลูกตุ้มหกผ่นซึ่งสามารถทรงตัวอยู่ในแนวตั้งได้
2. ได้เรียนรู้การออกแบบและสร้างกลไกทางด้าน Mechatronics รวมถึงการควบคุมกลไกดังกล่าวโดยการออกแบบทางด้านซอฟต์แวร์
3. ได้เรียนรู้การออกแบบตัวควบคุมแบบพีไอดี

1.7 รายละเอียดงบประมาณของโครงการ

1. ค่าส่วนประกอบทางด้าน Mechatronics และชุดอุปกรณ์อิเล็กทรอนิกส์ 1,000 บาท
2. ค่าถ่ายเอกสารและจัดทำรูปเล่มปริชญานิพนธ์ 1,000 บาท

รวมเป็นเงินทั้งสิ้น 2,000 บาท (สองพันบาทถ้วน)

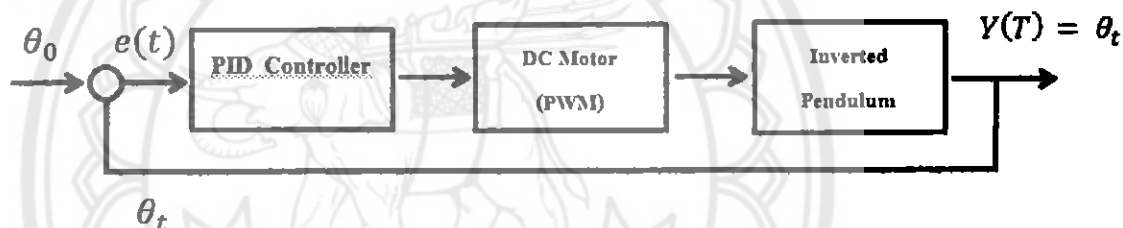
หมายเหตุ ถัวเฉลี่ยทุกรายการ



บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ระบบลูกตุ้มผกผัน เป็นเครื่องมือที่ได้รับความนิยมในการศึกษาระบบควบคุม ในงานทางด้านวิศวกรรม เนื่องจากระบบลูกตุ้มผกผันเป็นระบบที่มีข้อศึกษาอยู่หลายประการ เช่น เป็นระบบที่ไม่เป็นเชิงเส้น เป็นระบบที่ไม่มีความเสถียรภาพ เป็นต้น ดังนั้นระบบดังกล่าวจึงเหมาะแก่การนำมาศึกษาเพื่อออกแบบตัวควบคุม ซึ่งสิ่งที่คาดหวังจากโครงการนี้คือ สามารถออกแบบและสร้างระบบลูกตุ้มผกผันสามารถออกแบบตัวควบคุมแบบพีไอดี สำหรับควบคุมลูกตุ้มผกผัน และระบบลูกตุ้มผกผันที่พัฒนาขึ้นสามารถทรงตัวอยู่ในแนวตั้งได้

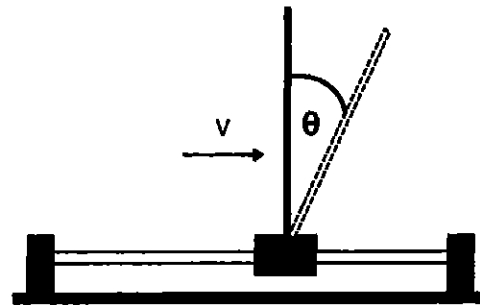


รูปที่ 2.1 ภาพรวมการทำงานของระบบ

จากรูปสามารถแบ่งระบบการทำงานได้ออกเป็น 3 ส่วน คือ

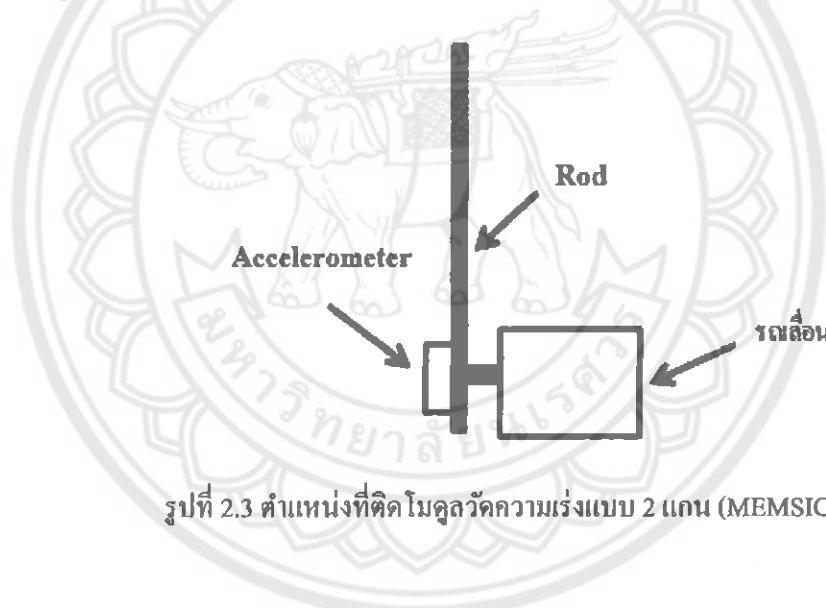
2.1 หลักการของลูกตุ้มผกผัน (Inverted Pendulum)

การทำงานของระบบควบคุมนั้นจะต้องทำการรักษาสมดุลของลูกตุ้ม เพื่อให้ลูกตุ้มสามารถตั้งอยู่ได้ในแนวตั้งตลอดเวลา โดยระบบจะต้องส่งแรงให้ฐานของลูกตุ้มมีการเคลื่อนที่ด้วยแรงความเร็วและเวลาที่เหมาะสม หากลูกตุ้มจะล้มไปทางด้านซ้าย ฐานของลูกตุ้มก็จะเคลื่อนตัวไปทางด้านซ้ายเพื่อเลี้ยงให้ลูกตุ้มตั้งอยู่ในแนวตั้งดั้งเดิม ในทำนองเดียวกันเมื่อลูกตุ้มจะล้มไปทางด้านขวาฐานของลูกตุ้มก็จะเคลื่อนตัวไปทางด้านขวาเพื่อเลี้ยงให้ลูกตุ้มตั้งอยู่ในแนวตั้งดั้งเดิม เมื่อฐานของลูกตุ้มมีการเคลื่อนที่กลับไปกลับมา ก็จะเกิดแรงเหวี่ยงของตัวลูกตุ้ม แรงเหวี่ยงนี้จะช่วยให้ลูกตุ้มสามารถตั้งอยู่ในแนวตั้งได้ โดยฐานของลูกตุ้มจะใช้มอเตอร์กระแสตรงเป็นตัวควบคุม ซึ่งความเร็วของมอเตอร์ จะขึ้นอยู่กับ มุมที่ได้จากการวัดของ โมดูลวัดความเร็วแบบสองแกน ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ระบบลูกตุ้มผกผัน

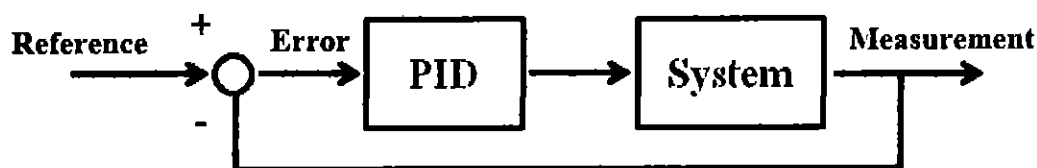
โดยมุม θ จะได้จากการเอียงท่ามุมของ โมดูลวัดความเร่งแบบ 2 แกน (Accelerometer) โดยจะติด โมดูลวัดความเร่งแบบ 2 แกน (MEMSIC2125) ไว้ที่ฐานของแท่งด่างน้ำหนัก ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตำแหน่งที่ติด โมดูลวัดความเร่งแบบ 2 แกน (MEMSIC2125)

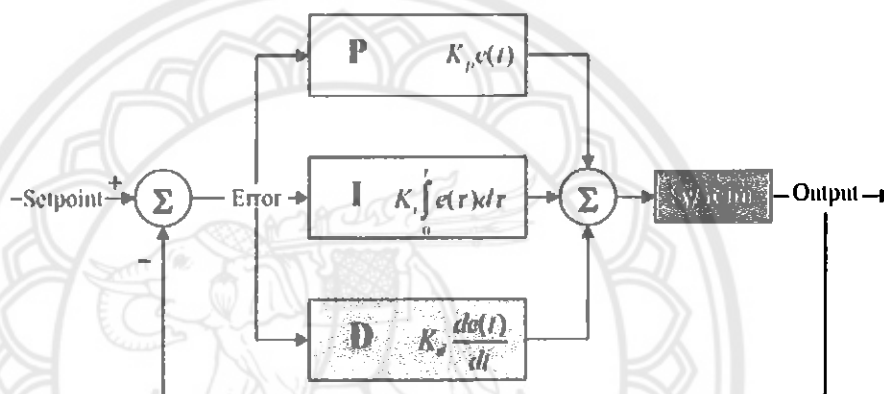
2.2 ตัวควบคุมพีไอดี (PID Controller)[2]

ตัวควบคุมแบบพีไอดี จะนำมาใช้ในการออกแบบตัวควบคุมเพื่อช่วยในการลดข้อผิดพลาดของการทำงานจากระบบลูกตุ้มผกผันให้มีเสถียรภาพในการทรงตัวมากขึ้น ตัวควบคุมส่วนใหญ่ที่ใช้ในการควบคุมกระบวนการเป็นแบบพีไอดี โดยต่อกับระบบที่จะต้องการควบคุมซึ่งจะช่วยให้ระบบเข้าสู่จุดที่ต้องการเร็วขึ้น ซึ่งสามารถแบ่งการควบคุมเป็นส่วนๆ[2] ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 ตัวควบคุม PID ที่ต่อเข้ากับระบบแบบอนุกรม

โดยปกติแล้วการควบคุมแบบป้อนกลับสามารถพิจารณาโดยแบ่งการควบคุมออกเป็น 3 ส่วนดังรูป



รูปที่ 2.5 ระบบควบคุม PID[2]

จากรูปจะเห็นว่าเราสามารถแบ่งการควบคุมแบบป้อนกลับออกเป็นสามส่วนดังสมการ ซึ่งค่า Setpoint เท่ากับ 0 และ $U(t)$ คือผลรวมของสมการ และนำค่าที่ได้มาแปลงเป็นค่า PWM ให้กับมอเตอร์ และจะไม่มีการใช้สมการในส่วนของ System ในโครงการนี้

$$u(t) = P_{out} + I_{out} + D_{out} \quad (2.1)$$

Proportation Control Action (P Action) : เทอมของสัดส่วน (บางครั้งเรียก อัตราขยาย) จะเปลี่ยนแปลงเป็นสัดส่วนของค่าความผิดพลาด การตอบสนองของสัดส่วนสามารถทำได้โดยการคูณค่าความผิดพลาดด้วยค่าคงที่ K_p หรือที่เรียกว่าอัตราขยายสัดส่วน เทอมของ P สามารถเขียนสมการ [2] ได้ดังนี้

$$P_{out} = K_p e(t) \quad (2.2)$$

เมื่อ P_{out} = สัญญาณขาออกของเทอมสัดส่วน

K_p = อัตราขยายสัดส่วน หรือตัวแปรปรับค่าได้

e = ค่าความผิดพลาด

t = เวลา

Integral Control Action (I Action) : ผลจากเทอมปริพันธ์ (บางครั้งเรียก reset) เป็นสัดส่วนของขนาดความผิดพลาดและระยะเวลาของความผิดพลาด ผลรวมของความผิดพลาดในทุกช่วงเวลา (ปริพันธ์ของความผิดพลาด) จะให้ออฟเซตสะสมที่ควรจะเป็นในก่อนหน้า ความผิดพลาดสะสมจะถูกคูณโดยอัตราขยายปริพันธ์ ขนาดของผลของเทอมปริพันธ์จะกำหนดโดยอัตราขยายปริพันธ์ (K_i) เทอมปริพันธ์จะเป็นไปตามสมการ[2] ได้ดังนี้

$$I_{out} = K_i \int_0^t e(t) dt \quad (2.3)$$

เมื่อ I_{out} = สัญญาณขาออกของเทอมปริพันธ์

K_i = อัตราขยายปริพันธ์, ตัวแปรปรับค่าได้

e = ความผิดพลาด

t = เวลา

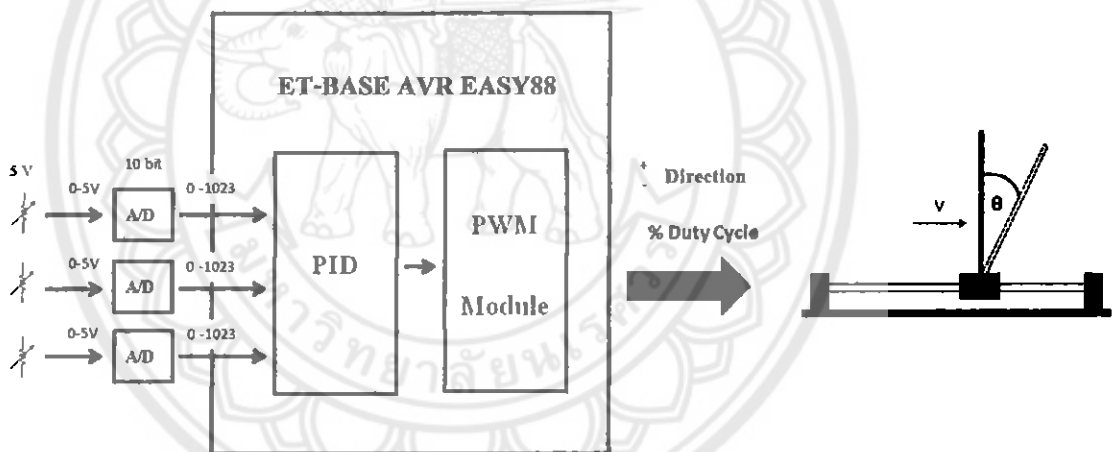
เทอมปริพันธ์ (เมื่อรวมกับเทอมสัดส่วน) จะเร่งกระบวนการให้เข้าสู่จุดที่ต้องการและขจัดความผิดพลาดที่เหลืออยู่ที่เกิดจากการใช้เพียงเทอมสัดส่วน แต่อย่างไรก็ตาม เทอมปริพันธ์เป็นการตอบสนองต่อความผิดพลาดสะสมในอดีต จึงสามารถทำให้เกิดโอเวอร์ชูตได้ (ข้ามจุดที่ต้องการและเกิดการหันเหไปทางทิศทางอื่น)

Derivative Control Action (D Action) : อัตราการเปลี่ยนแปลงของความผิดพลาดจากกระบวนการนั้นคำนวณหาจากความชันของความผิดพลาดทุกๆเวลา (นั่นคือ เป็นอนุพันธ์อันดับหนึ่งสัมพันธ์กับเวลา) และคูณด้วยอัตราขยายอนุพันธ์ K_d ขนาดของผลของเทอมอนุพันธ์ (บางครั้งเรียกอัตรา) ขึ้นกับ อัตราขยายอนุพันธ์ K_d เทอมอนุพันธ์เป็นไปตามสมการ[2] ดังนี้

$$D_{out} = K_d \frac{d}{dt} e(t) \quad (2.4)$$

เมื่อ D_{out} = สัญญาณขาออกของเทอมอนุพันธ์
 K_d = อัตราขยายอนุพันธ์, ตัวแปรปรับค่าได้
 e = ความผิดพลาด
 t = เวลา

เทอมอนุพันธ์จะชะลออัตราการเปลี่ยนแปลงของสัญญาณขาออกของระบบควบคุมและด้วยผลนี้จะช่วยให้ระบบควบคุมเข้าสู่จุดที่ต้องการ ดังนั้นเทอมอนุพันธ์จะใช้ในการลดขนาดของโอเวอร์ชูตที่เกิดจากเทอมปริพันธ์และทำให้เสถียรภาพของการรวมกันของระบบควบคุมดีขึ้น แต่อย่างไรก็ตามอนุพันธ์ของสัญญาณรบกวนที่ถูกขยายในระบบควบคุมจะไวมากต่อการรบกวนในเทอมของความผิดพลาดและสามารถทำให้กระบวนการไม่เสถียรได้ถ้าสัญญาณรบกวนและอัตราขยายอนุพันธ์มีขนาดใหญ่เพียงพอ[2]



รูปที่ 2.6 ภาพรวมของระบบ (System Overview)

การปรับค่า K_p , K_i และ K_d ด้วยมือ (Manully Tuned)

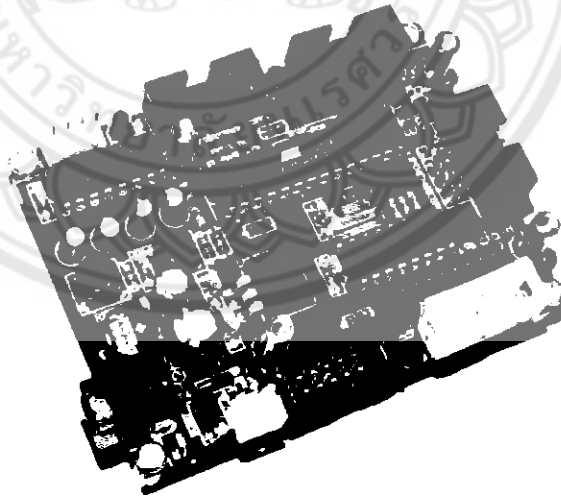
ถ้าระบบยังคงทำงาน ชั้นแรกให้ตั้งค่า K_i และ K_d เป็นศูนย์ เพิ่มค่า K_p จนกระทั่งสัญญาณขาออกเกิดการแกว่ง (oscillate) แล้วเพิ่ม K_i จนกระทั่งออฟเซตถูกต้องในเวลาที่ต้องการของกระบวนการ แต่ถ้า K_i มากไปจะทำให้ไม่เสถียร สุดท้ายถ้าต้องการ ให้เพิ่มค่า K_d จนกระทั่งรูปอยู่ในระดับที่ยอมรับได้ แต่ถ้า K_d มากเกินไปจะเป็นเหตุให้การตอบสนองและโอเวอร์ชูตเกินยอมรับได้

ปกติการปรับจูน PID ถ้าเกิดโอเวอร์ชูตเล็กน้อยจะช่วยให้เข้าสู่จุดที่ต้องการเร็วขึ้น แต่ในบางระบบไม่สามารถยอมให้เกิดโอเวอร์ชูตได้ และถ้าค่า K_p น้อยเกินไปก็จะทำให้เกิดการแกว่ง[2]

ตัวแปร	ช่วงเวลายกขึ้น (Rise time)	โอเวอร์ชูต (Overshoot)	เวลาสู่สมดุลย์ (Settling time)	ความผิดพลาดสถานะ คงตัว (Steady-state error)	เสถียรภาพ (Stable)
K_p	ลด	เพิ่ม	เปลี่ยนแปลงเล็กน้อย	ลด	ลด
K_i	ลด	เพิ่ม	เพิ่ม	ลดลงอย่างมีนัยสำคัญ	ลด
K_d	ลดลงเล็กน้อย	ลดลงเล็กน้อย	ลดลงเล็กน้อย	ตามทฤษฎีไม่มีผล	ดีขึ้นถ้า K_d ลดลง

ตารางที่ 2.1 ผลของการเพิ่มค่าตัวแปรอย่างอิสระ

2.3 บอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE AVR EASY88[8]



รูปที่ 2.7 บอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE AVR EASY88

ET-BASE AVR EASY88 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR โดยบอร์ดเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA88 ของ ATMEL เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ในตัวถังแบบ 28 Pin DIP ซึ่ง MCU สามารถทำงานได้ด้วยความถี่สูงสุด 20MHz ที่ 1 Clock / Machine Cycle นอกจากนี้แล้วยังมีความเพียบพร้อมด้วยอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งาน ไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลแบบ EEPROM ขนาด 512 Byte และหน่วยความจำใช้งานแบบ SRAM อีก 1 K Byte ส่วนในด้านของอุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วนเหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดีโดยจะมีทั้งระบบฮาร์ดแวร์ของ SPI, UART, I2C, Watchdog, Timer/Counter, PWM และ ADC ฯลฯ รายละเอียดต่างๆสามารถศึกษาได้ในภาคผนวก ก

2.4 MEMSIC2125 โมดูลวัดความเร่งแบบ 2 แกน[6]

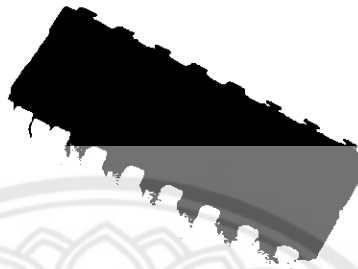


รูปที่ 2.8 MEMSIC2125 โมดูลวัดความเร่งแบบ 2 แกน

MEMSIC2125 เป็น โมดูลวัดความเร่ง (accelerometer sensor) ภายในบรรจุตัวตรวจจับความเร่งแบบ 2 แกน สามารถวัดได้ทั้งความเร่งในเชิงพลศาสตร์ (dynamic acceleration) หรือความสั่นสะเทือน (vibration) และเชิงสถิตยศาสตร์ (static acceleration) ซึ่งก็คือความเร่งจากแรงโน้มถ่วงโลก (gravity) รายละเอียดต่างๆสามารถศึกษาได้ในภาคผนวก ข

2.5 มอเตอร์กระแสตรง (DC Motor)[3]

ไอซีควบคุมมอเตอร์กระแสตรง L293D[9]



รูปที่ 2.9 ไอซีควบคุมมอเตอร์กระแสตรง L293D

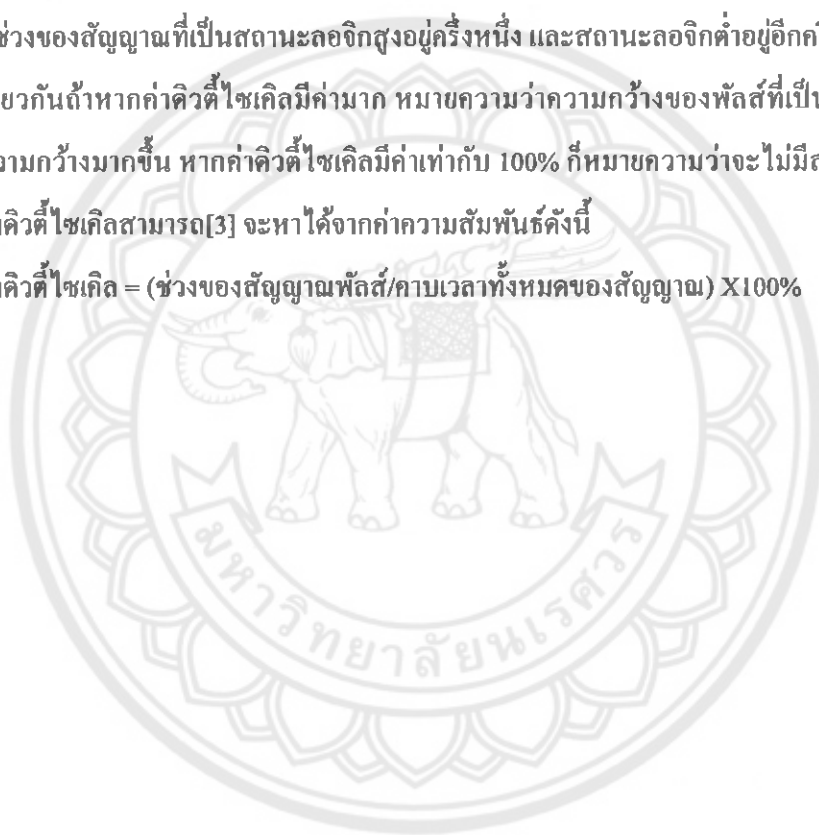
ไอซี L293D เป็นไอซีที่ใช้ในการควบคุมมอเตอร์กระแสตรง ด้วยระบบดิจิทัลที่รับข้อมูลจากบอร์ดไมโครคอนโทรลเลอร์ ให้เดินหน้า ถอยหลัง และหยุด รายละเอียดการทำงานของไอซี L293D สามารถศึกษาได้ในภาคผนวก ง

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไป เช่น การควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่า โดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการควบคุม โดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าว ถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นผู้จัดทำจึงเลือกใช้วิธีการควบคุม โดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation)[3]

วิธีการมอดูเลชันทางความกว้างของพัลส์

การมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation) จะเป็นการปรับเปลี่ยนที่สัดส่วน และความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มีการเปลี่ยนแปลง หรือเป็นการเปลี่ยนแปลงที่ค่าของดีวตี้ไซเคิล (duty cycle) นั้นเอง ซึ่งค่าของดีวตี้ไซเคิลคือช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าดีวตี้ไซเคิลมีค่าเท่ากับเท่ากับ 50% ก็หมายถึงใน 1 รูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง และในทำนองเดียวกันถ้าหากค่าดีวตี้ไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าดีวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่า จะไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าดีวตี้ไซเคิลสามารถ[3] จะหาได้จากค่าความสัมพันธ์ดังนี้

$$\text{ค่าดีวตี้ไซเคิล} = (\text{ช่วงของสัญญาณพัลส์} / \text{คาบเวลาทั้งหมดของสัญญาณ}) \times 100\% \quad (2.5)$$



บทที่ 3

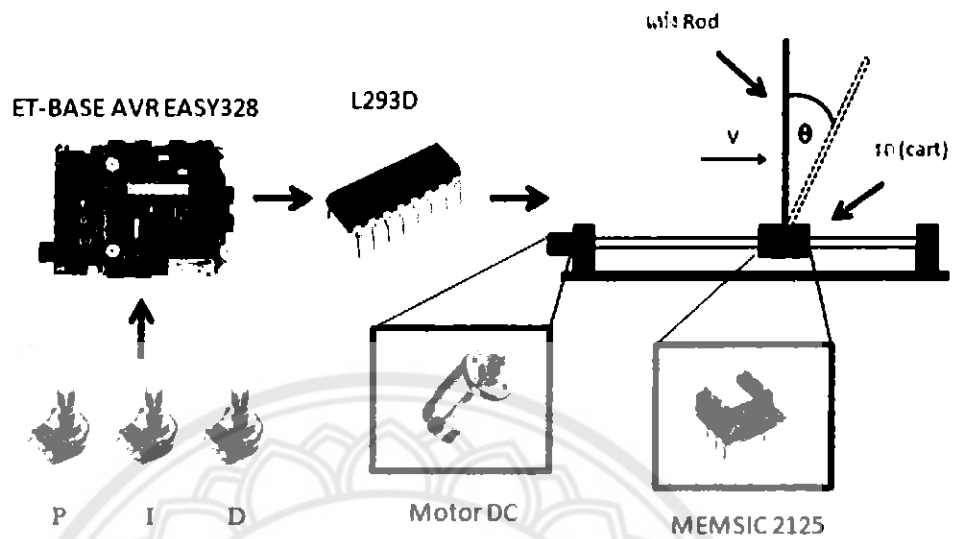
วิธีการดำเนินการ

ในบทนี้ผู้ดำเนินโครงการจะกล่าวถึงรายละเอียดการดำเนินโครงการ โดยแบ่งเป็น 4 ส่วน ดังนี้

1. ภาพรวมของระบบ ซึ่งจะบอกถึงภาพรวมของระบบว่าประกอบด้วยอะไรบ้าง ใช้โมดูลอะไรในการสร้างระบบขึ้นมา แต่ละส่วนสัมพันธ์กันอย่างไร และระบบมีการทำงานอย่างไร
2. การสร้างระบบลูกตุ้มผกผันเพื่อใช้ในการทดลอง โครงการงาน
3. การเขียน โปรแกรมลงบนบอร์ด ไมโคร โปรเซสเซอร์เพื่อใช้ในการควบคุมระบบลูกตุ้มผกผัน
4. การปรับค่า K_p K_i และ K_d ด้วยมือ (Manually Tuned)

3.1 ภาพรวมของระบบ (System Overview)

ในการวัดมุมเอียงของแท่ง Rod จะใช้เซนเซอร์ MEMSIC 2125 ซึ่งเป็นโมดูลวัดความเร่งแบบ 2 แกน ใช้วัดสำหรับมุมเอียงในแนวแกน X (Roll) และ Y (Pitch) ซึ่งระบบลูกตุ้มผกผันนี้ใช้มุมเอียงแค่ในแนวแกน X (Roll) หลังจากนั้นจะส่งสัญญาณที่ได้ไปที่ ET-BASE AVR EASY328 ซึ่งเป็นไมโครโปรเซสเซอร์ที่ทำหน้าที่ในการประมวลผลของสัญญาณที่ได้มาจากเซนเซอร์ จะได้ผลลัพธ์ออกมาเป็นมุมเอียงของแกน X แล้วส่งผลที่ได้นั้น ไปเข้าสมการ พีไอดีเพื่อใช้ในการควบคุมมอเตอร์ โดยมอเตอร์ที่ใช้ในโครงการเป็นมอเตอร์กระแสตรงขนาด 12 V ควบคุมผ่านไอซีเบอร์ L293D ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของระบบ (System Overview)

3.2 การสร้างระบบลูกตุ้มผกผันเพื่อใช้ในการทดลองโครงงาน

3.2.1 สร้างรางเลื่อนของระบบลูกตุ้มผกผันโดยใช้รางเลื่อนจากเครื่องพิมพ์ Canon[5]

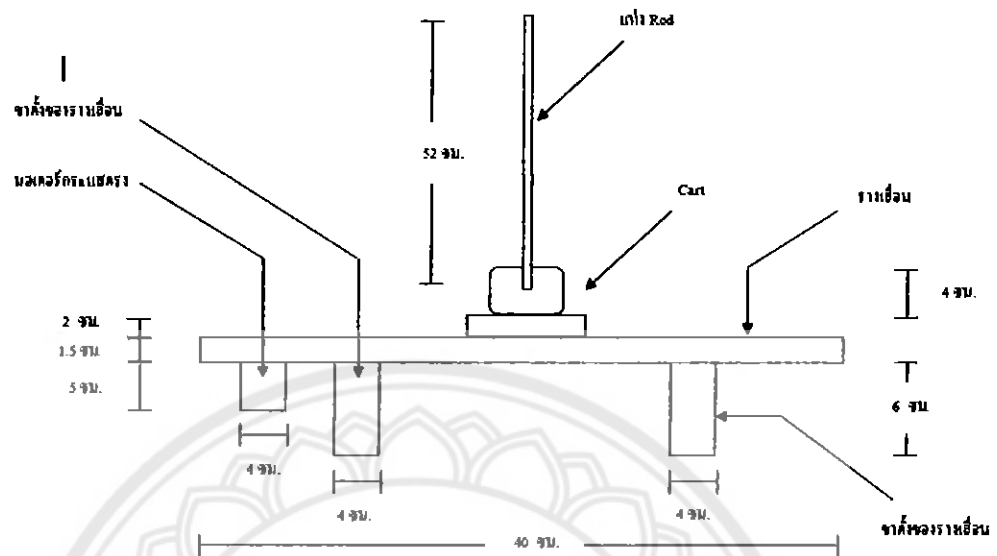
การสร้างรางเลื่อนของระบบลูกตุ้มผกผันนั้น ทางผู้ดำเนินโครงการได้เลือกใช้รางเลื่อนที่อยู่ในเครื่องพิมพ์ Canon เนื่องจากว่าทางผู้ดำเนินโครงการนั้นได้ทำการทดลองแล้วเห็นว่า มีความแข็งแรง และการขับเคลื่อนของมอเตอร์ดีกว่า โมเดลรางเลื่อนชิ้นแรกที่ได้จัดทำขึ้น โดยโมเดลชิ้นแรกนั้น ทางผู้ดำเนินโครงการได้ทำจากแผ่นอะคริลิก ซึ่งมีความยาว 100 เซนติเมตร ตัวรถทำจากอะคริลิก ดังรูปที่ 3.2 ซึ่งข้อดีของโมเดลชิ้นแรกคือ มีความขวมมาก ส่วนข้อเสียคือ ล้อรถเมื่อนำมาวิ่งบนแผ่นอะคริลิก เกิดการสั่นไถลของล้อรถ ทางผู้จัดทำจึงเปลี่ยนมาใช้รางเลื่อนของเครื่องพิมพ์ Canon เป็นโมเดลในการสร้างรางเลื่อนของระบบลูกตุ้มผกผัน ดังรูปที่ 3.3 ข้อดีของรางเลื่อนนี้ คือ การขับเคลื่อนของมอเตอร์ดีขึ้นเนื่องจากใช้สายพานในการขับเคลื่อน ส่วนข้อเสียคือ มีความยาวเพียง 40 เซนติเมตร



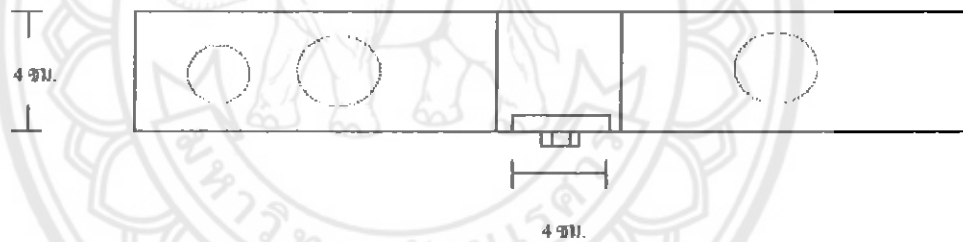
รูปที่ 3.2 รางเลื่อนที่ทำมาจากแผ่นอะคริลิก



รูปที่ 3.3 รางเลื่อนที่ทำมาจากแผ่นรางเครื่องพิมพ์ Canon[5]



รูปที่ 3.4 รางเครื่องพิมพ์แสดงมุมมองด้านหน้า



รูปที่ 3.5 รางเครื่องพิมพ์แสดงมุมมองด้านบน

3.3 การเขียนโปรแกรมบนบอร์ดไมโครโปรเซสเซอร์เพื่อใช้ในการอ่านค่าจากเซนเซอร์ที่ใช้สำหรับวัดมุมเอียงในแนวแกน X (Roll)

3.3.1 หลักการประมวลผลสัญญาณที่ได้จากโมดูลวัดความเร่ง 2 แกน MEMSIC 2125

เอาต์พุตของเซนเซอร์ MEMSIC 2125 จะแสดงออกมาในรูปสัญญาณพัลส์ ทางผู้ดำเนินโครงการ จึงใช้คำสั่ง pulseIn ในการอ่านค่าสัญญาณพัลส์ เพื่อทำการอ่านค่าของสัญญาณพัลส์ที่มี logic เป็น “ 1 ”

$$Xroll = pulseIn(xPin, HIGH)$$

สมการที่ใช้สำหรับควมค่าหาความเร่งในแนวแกน X (accelerationX) สำหรับ โมดูลวัด
ความเร่ง 2 แกน คือ

$$\text{accelerationX} = ((X_{\text{roll}} / 10) - 500) * 8 \quad (3.1)$$

โดยที่ค่า Xroll ต้องอยู่ในหน่วยไมโครวินาที

ดังนั้นจะสามารถหามุมเอียงตามแนวแกน X ได้จาก

$$X_{\text{angle}} (\text{radian}) = \text{asin} (\text{accelerationX} / 1000.0) \quad (3.2)$$

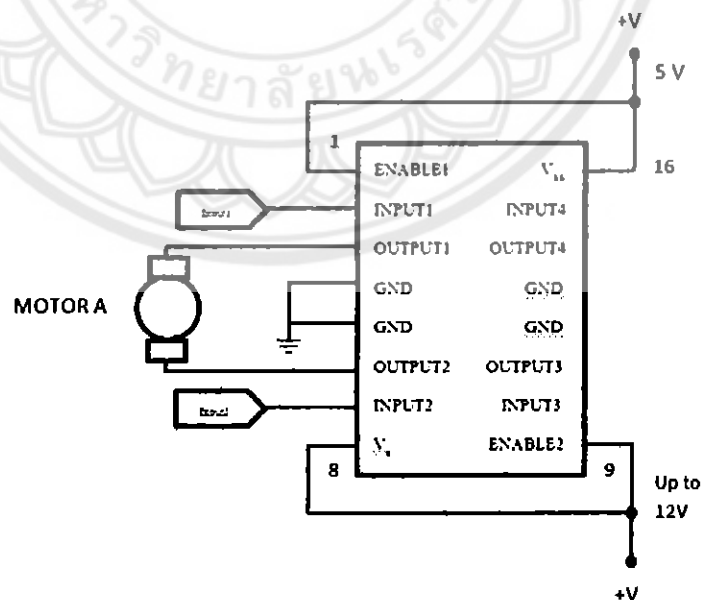
สำหรับค่าที่ได้ออกมานั้นจะได้มุมในหน่วยขององศาเรเดียน (radian) หากต้องการให้ค่ามุมอยู่ใน
หน่วยองศาดีกรี (degree) สามารถทำได้โดย

$$X_{\text{angle}} = X_{\text{angle}} * (360 / (2 * M_PI)) \quad (3.3)$$

โดยค่า M_PI คือ 3.141592653589

3.3.2 หลักการใช้ IC เบอร์ L293D เพื่อควบคุมมอเตอร์

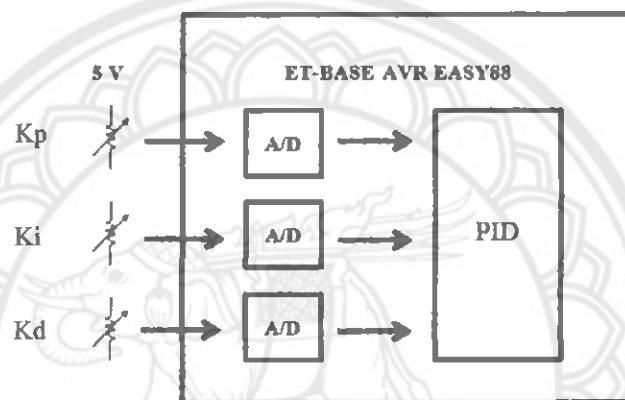
โดยแรงดันที่จ่ายให้กับ IC เบอร์ L293D นั้น ทางผู้ดำเนินโครงการ ได้ใช้กระแสไฟฟ้าขนาด 8
โวลต์ ต่อที่ขาที่ 8 เพราะ ถ้าหากมากกว่านี้ จะทำให้มอเตอร์หมุนเร็วเกินไป และทำการต่อวงจรตามรูป
ที่ 3.4



รูปที่ 3.6 วงจรควบคุมมอเตอร์

3.3.3 หลักการประมวลผลสัญญาณจากตัวต้านทานสำหรับใช้ปรับค่า Kp, Ki และ Kd

การสร้างวงจรปรับค่าได้ โดยการใช้ตัวต้านทานปรับค่าได้ขนาด 10 k Ω ซึ่งใช้แรงดันขนาด 5 V จำนวน 3 ตัวต่อเข้ากับขา Arduino Analog[0...5] ขาไหนก็ได้จำนวน 3 ขา สำหรับวิธีการประมวลของตัวต้านทานปรับค่าได้ นั้นบอร์ด Arduino จะใช้ฟังก์ชัน analogRead ในการอ่านค่าสัญญาณอนาล็อก แล้วนำสัญญาณที่อ่านนั้นมาแปลงค่าเป็นค่าที่อยู่ในช่วง 0 ถึง 1023 เพื่อใช้สำหรับการคำนวณหาค่า PID ในลำดับต่อไป ดังแสดงในรูป 3.7



รูปที่ 3.7 แสดงการทำงานของตัวต้านทานสำหรับปรับค่า Kp, Ki และ Kd

3.3.4 Pseudo code ของการควบคุมไมโครคอนโทรลเลอร์

ในส่วนของหัวข้อนี้จะขอแยกกล่าวถึงเป็น 4 ส่วนคือ

3.3.4.1 Pseudo code ในการอ่านค่าสัญญาณพัลส์ที่มี logic = 1 จากโมดูล MEMSIC 2125

รับค่าแกน X เข้า Port Analog I2

แสดง Pseudo code ในการอ่านค่าคาบสัญญาณพัลส์ที่มี logic = 1 ของแกน X

$X_{raw} = \text{pulseIn}(xPin, HIGH)$ // อ่านค่าสัญญาณพัลส์ที่เป็น logic 1

$xGForce = ((X_{raw} / 10) - 500) * 8$ // คำนวณค่า xGforce ในหน่วยมิลิวินาที

$Xangle = \text{asin}(xGForce / 1000.0)$ // คำนวณค่ามุมเอียงในหน่วยเรเดียน

$Xangle = Xangle * (360 / (2 * M_PI))$ // คำนวณค่ามุมเอียงในหน่วยดีกรี

3.3.4.2 Pseudo code ในการอ่านค่าจากตัวต้านทานปรับค่าได้

BEGIN

readPotenKP = analogRead(potentiometerKP) // อ่านค่าจากขา A/D ที่อยู่ในบอร์ด

ไมโคร

kp = (double)readPotenKP

int readPotenKI = analogRead(potentiometerKI) // อ่านค่าจากขา A/D ที่อยู่ในบอร์ด

ไมโคร

ki = (int)readPotenKI

int readPotenKD = analogRead(potentiometerKD) // อ่านค่าจากขา A/D ที่อยู่ใน

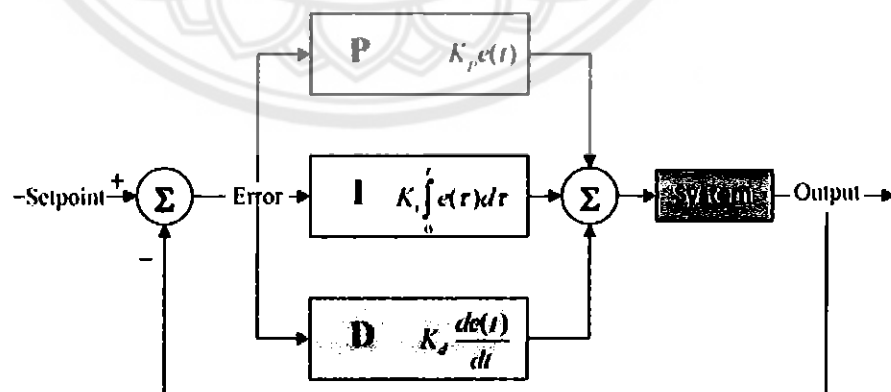
บอร์ดไมโคร

kd = (int)readPotenKD

END

3.3.4.3 Pseudo Code ของสมการพีไอดี

ระบบควบคุมแบบพีไอดี จะนำมาใช้ในการออกแบบตัวควบคุมเพื่อช่วยในการลดข้อผิดพลาดของการทำงานของระบบลูกค้ำผกผันให้มีเสถียรภาพในการทรงตัวมากขึ้น ซึ่งระบบการควบคุมพีไอดี จะเป็นการควบคุมแบบป้อนกลับ สามารถพิจารณาโดยแบ่งการควบคุมออกเป็น 3 ส่วนดังรูป



รูปที่ 3.8 รูประบบควบคุม PID[2]

จากรูปที่ 3.8 สามารถเขียนเป็น Pseudo Code ของระบบฟีดแบ็คได้ดังนี้

BEGIN

now = millis // เก็บค่าหน่วยมิลลิวินาที ในตัวแปร now

timeChange = (now - lastTime) // ค่า sampling ที่ใช้ในการทำงานของโปรแกรมใน

แต่ละรอบ

error = Setpoint - Input // เก็บค่า error

errSum += (error * timeChange) // ค่าผลรวมของ error

dErr = (error - lastErr) / timeChange

p = kp * error

i = ki * errSum

d = kd * dErr

Output = p + i + d // ผลรวมของสมการฟีดแบ็ค

lastErr = error // ค่า error ครั้งที่ผ่านมา

lastTime = now // ค่า sampling ครั้งที่ผ่านมา

// กำหนดค่าสูงสุดของฟีดแบ็ค

if(Output > 20000)

Output = 20000

if(Output < -20000)

Output = -20000

END

ต้องมีการ reset ค่า errSum เพราะค่านี้เกิดจากการสะสมของค่ามุมเอียง เมื่อมีการสะสมค่านี้เป็นเวลานานๆ ในทิศทางเดิม จะทำให้ค่า I มีค่ามาก จนทำให้ค่าโอเวอร์ชูตเกินขอบเขตของระบบที่จะรับได้ มีผลทำให้ระบบไม่เสถียรภาพไปในทันที

BEGIN

if(fall == 4 || fall == -4) // ถ้าค่า fall มีการสะสมครบ 4 ค่าให้ทำการ reset ค่า errSum และ fall

fall = 0;


```

errSum = 0;

end

if(Input > 4) fall = fall + 1; //ค่าม้วนวัดได้เกิน 4 ค่า fall+1
if(Input < -4) fall = fall - 1; //ค่าม้วนวัดได้เกิน -4 ค่า fall-1

```

3.3.4.4 Pseudo code ของการใช้ค่าที่ได้จากสมการพีไอดีเพื่อควบคุมมอเตอร์

```

BEGIN // เรียกใช้ฟังก์ชัน Compute ที่ทำการคำนวณพีไอดี
SEND Xangle, Output, Kp, Ki, Kd, timeChange TO PC RS232 // ส่งค่าทั้ง 6 ค่าไปยัง PC
ผ่าน RS232

if (Output > 0) // ถ้าค่า Output มากกว่า 0 ให้หมุนไปทางซ้าย
    Order motor to Left
    PWM = (int)map(Output, 1, 20000, 135, 255) // กำหนดความเร็วของมอเตอร์ ค่าสั่ง
map ทำการเทียบแบบบัญญัติไตรยางศ์ค่า output เป็นค่า PWM
else if (Output < 0) // ถ้าค่า Output มากกว่า 0 ให้หมุนไปทางขวา
    Order motor to Right
    Output = abs(Output)
    PWM = (int)map(Output, 1, 20000, 135, 255) // กำหนดความเร็วของมอเตอร์ ค่าสั่ง
map ทำการเทียบแบบบัญญัติไตรยางศ์ค่า output เป็นค่า PWM
else if (Output == 0) // ถ้าค่า Output เท่ากับ 0 สั่งมอเตอร์หยุด
    Stop motor

END

```

การเทียบแบบบัญญัติไตรยางศ์ค่า Output เป็นค่า PWM เช่น ค่า output 20,000 เทียบกับค่า PWM เท่ากับ 255 ถ้าค่า Output 15,000 ค่า PWM เท่ากับ $(15,000 \times 255) / 20,000$ เท่ากับ 191.25 ค่านี้คือความเร็วที่มอเตอร์เคลื่อนที่

3.4 การปรับค่า K_p , K_i และ K_d ด้วยมือ (Manually Tuned)

เมื่อระบบทำงาน ชั้นแรกให้ตั้งค่า K_i และ K_d เป็นศูนย์ เพิ่มค่า K_p จนกระทั่งสัญญาณขาออกเกิดการแกว่ง (oscillate) แล้วเพิ่ม K_i จนกระทั่งออฟเซตถูกต้อง ในเวลาที่พอสเพียงของกระบวนการ ค่าออฟเซตในที่นี้ คือค่าจากสมการปริพันธ์ (integral) ขดเซตค่าจากสมการสัดส่วน (Proportional) ให้ระบบเข้าสู่จุดสมดุลในเวลาเร็วขึ้น ซึ่งถ้า K_i มากไปจะทำให้ไม่เสถียร สุดท้ายระบบยังไม่สามารถเข้าสู่จุดสมดุลตามที่ต้องการ ให้เพิ่มค่า K_d จนกระทั่งระบบเข้าสู่จุดสมดุล แต่ถ้า K_d มากเกินไปจะเป็นเหตุให้การตอบสนองและช่วงของโอเวอร์ชูตเกินที่จะทำให้ระบบสมดุล (โอเวอร์ชูต คือ ค่าการตอบสนองที่วัดได้จากสถานะอยู่ตัวค่าสุดท้าย) ปกติการปรับจูน PID ถ้าเกิดโอเวอร์ชูตเล็กน้อยจะช่วยให้เข้าสู่จุดที่ต้องการเร็วขึ้น แต่ในบางระบบไม่สามารถยอมให้เกิดโอเวอร์ชูตได้ และถ้าค่า K_p น้อยเกินไปก็จะทำให้เกิดการแกว่ง

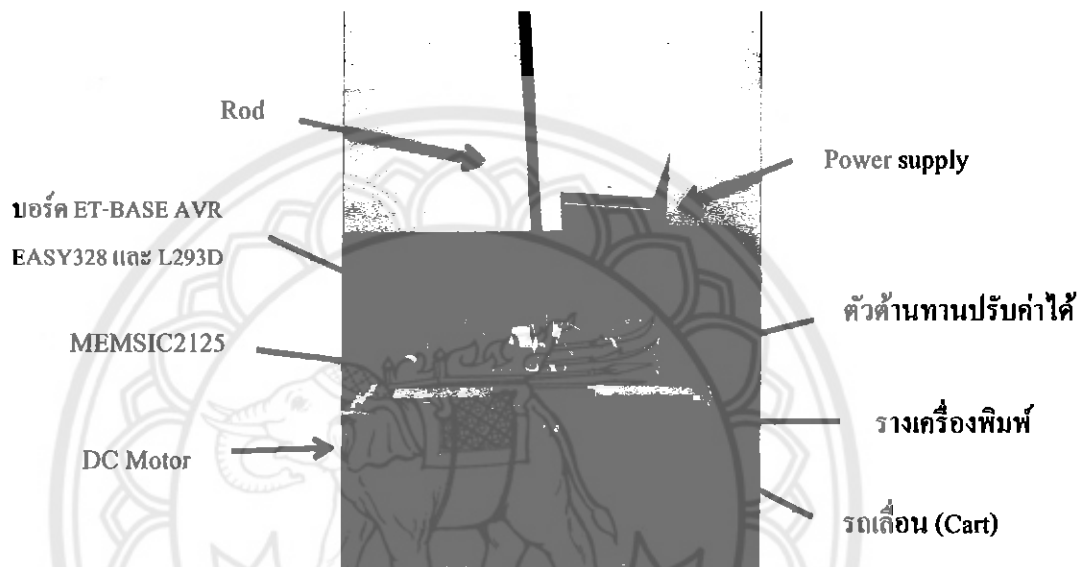
3.5 บทสรุป

ในบทที่ 3 นี้จะกล่าวถึงการดำเนินงานทั้งหมด ซึ่งอันดับแรกจะกล่าวถึงภาพรวมของระบบ (System Overview) ว่าระบบนั้นมีองค์ประกอบอะไรและมีหลักการทำงานอย่างไร จากนั้นจะกล่าวถึงการเขียนโปรแกรมลงบนบอร์ดไมโครคอนโทรลเลอร์สำหรับอ่านค่าเซนเซอร์เพื่อนำมาประมวลผลและคำนวณโดยสมการพีไอดี พร้อมทั้ง Pseudo code ในการใช้ค่าที่ได้จากสมการพีไอดีในการขับเคลื่อนมอเตอร์

บทที่ 4

การทดลองและผลการทดลอง

4.1 ระบบที่ออกแบบและสร้างขึ้น



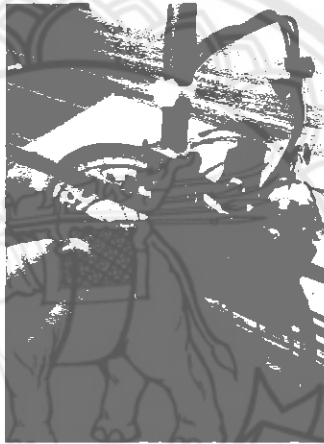
รูปที่ 4.1 ระบบที่ออกแบบและสร้างขึ้น

ระบบที่ออกแบบและสร้างขึ้น (รูปที่ 4.1) คือ รางเลื่อนเชิงเส้น ซึ่งใช้มอเตอร์กระแสตรงทำหน้าที่ในการขับเคลื่อนให้ตัวรถบนสายพานเคลื่อนที่ได้ บนตัวรถจะมีแกนหมุนที่ติดกับ Rod (แท่งอะลูมิเนียมที่ใช้ในการถ่วงน้ำหนัก) ซึ่ง Rod จะมีโมดูลวัดมุมเยี่ยง MEMSIC 2125 ติดอยู่ จากนั้นจะส่งมุมเยี่ยงที่วัดได้ไปยังบอร์ดไมโครคอนโทรลเลอร์ ซึ่งบอร์ดไมโครคอนโทรลเลอร์ประมวลผลโดยการนำมุมเยี่ยงที่วัดได้ไปคำนวณ โดยสมการพีไอดี ค่าที่ได้จากสมการจะนำมาแปลงเป็นค่า PWM (Pulse Width Modulation) เพื่อใช้ในการขับมอเตอร์กระแสตรงให้เคลื่อนที่ ซึ่งแสดงการทำงานดังในรูป 2.1

4.2 ผลการทดลองการวัดมุมเอียงของเซนเซอร์ MEMSIC 2125

4.2.1 วิธีการทดลอง

ระบบลูกตุ้มผกผันใช้แกน X แกนเดียวในการหามุมเอียงของแท่ง Rod ซึ่งการทดลองนั้นจะใช้เครื่องวงกลมสำหรับวัดมุมมาใช้วัดมุมเอียงของ MEMSIC 2125 ในแนวแกน X เพื่อใช้เป็นจุดเปรียบเทียบกับเซนเซอร์ที่วัดได้ แสดงดังรูปที่ 4.2 สำหรับการวัดค่ามุมเอียงการวัดมุมเอียงนั้น ระบบจะใช้ค่าตั้งแต่ -45 องศา ถึง 45 องศา โดยจะเพิ่มขึ้นทุกๆ 5 องศา โดยในแต่ละค่าที่วัดจะใช้ทั้งหมด 10 ค่า โดยแต่ละค่าที่วัดนั้นจะมีเวลาห่างกัน 1 วินาที



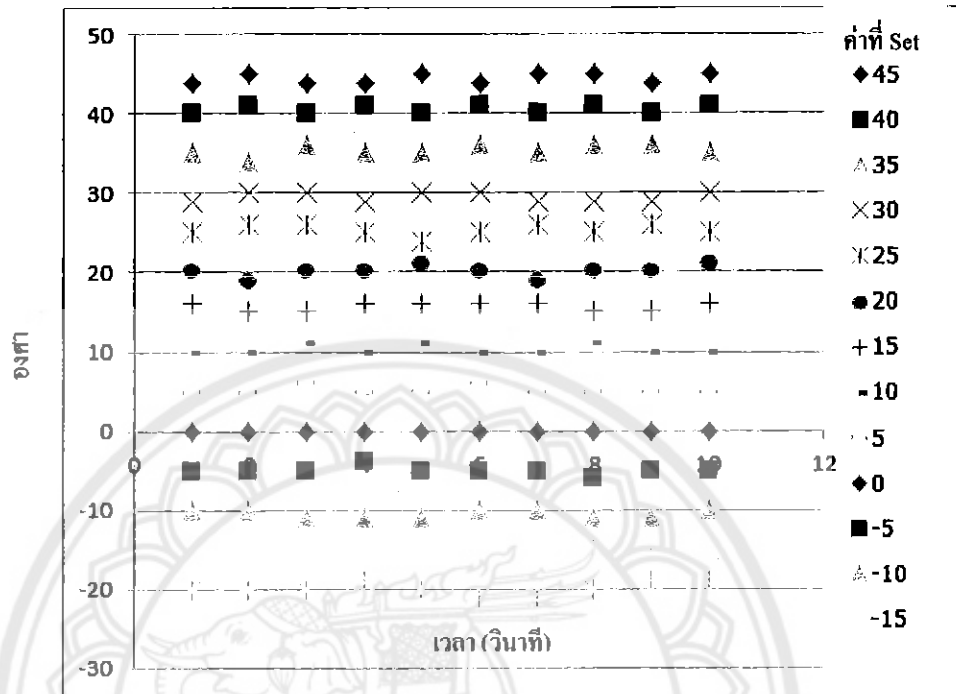
รูปที่ 4.2 วิธีการทดลองวัดมุมเอียงในแนวแกน X

4.2.2 ผลการทดลอง

ระบบลูกตุ้มผกผันจะใช้ค่ามุมเอียงที่ได้จากเซนเซอร์ตั้งแต่ -45 องศา ถึง 45 องศา เพราะถ้า Rod ทำมุมเอียงมากกว่า -45 องศา หรือ 45 องศา Rod จะไม่สามารถทรงตัวในแนวตั้งได้ ผู้จัดทำโครงการได้ทำการทดลองวัดค่ามุมเอียง -45 องศา ถึง 45 องศา ค่าที่ได้แสดงในตารางที่ 4.1

มุมเอียง (องศา)	ค่ามุมที่วัดได้(องศา)										ค่าเฉลี่ย	ค่าความคลาด เคลื่อน (องศา)
	1 sec	2 sec	3 sec	4 sec	5 sec	6 sec	7 sec	8 sec	9 sec	10 sec		
-45	-45	-46	-45	-47	-45	-45	-46	-46	-45	-45	-45.5	0.5
-40	-40	-41	-40	-39	-39	-40	-40	-39	-40	-39	-39.7	0.3
-35	-35	-36	-34	-35	-35	-36	-35	-36	-34	-35	-35.1	0.1
-30	-29	-30	-30	-31	-30	-31	-31	-29	-31	-30	-30.2	0.2
-25	-25	-25	-24	-24	-25	-25	-24	-24	-24	-25	-24.5	0.5
-20	-20	-20	-20	-19	-20	-21	-21	-20	-19	-19	-19.9	0.1
-15	-15	-15	-15	-14	-14	-14	-15	-14	-15	-15	-14.6	0.4
-10	-10	-10	-11	-11	-11	-10	-10	-11	-11	-10	-10.5	0.5
-5	-5	-5	-5	-4	-5	-5	-5	-6	-5	-5	-5	0
0	0	0	0	0	0	0	0	0	0	0	0	0
5	5	5	6	5	5	6	5	5	5	5	5.2	0.2
10	10	10	11	10	11	10	10	11	10	10	10.3	0.3
15	16	15	15	16	15	16	16	15	15	16	15.5	0.5
20	20	19	20	20	21	20	19	20	20	21	20	0
25	25	26	26	25	24	25	26	25	26	25	25.3	0.3
30	29	30	30	29	30	30	29	29	29	30	29.5	0.5
35	35	34	36	35	35	36	35	36	36	35	35.3	0.3
40	40	41	40	41	40	41	40	41	40	41	40.5	0.5
45	44	45	44	44	45	44	45	45	44	45	44.5	0.5
ค่าความคลาดเคลื่อนเฉลี่ย(องศา)												0.3

ตารางที่ 4.1 แสดงผลการทดลอง ค่าเฉลี่ย และค่าความคลาดเคลื่อนของการทดลอง ของการวัดมุมเอียง
ในแนวแกน X



รูปที่ 4.3 แสดงความสัมพันธ์ของค่ามุมเชิงในแนวแกน X ณ วินาทีที่ 1-10

จากตารางที่ 4.1 และรูปที่ 4.3 นั้นจะเห็นว่าค่าที่วัดได้จากเซนเซอร์นั้นจะนิ่ง เมื่อทำการวัดมุมจากวินาทีที่ 1 ถึง 10 นั้น ค่าจะแกว่งขึ้นหรือแกว่งลง 1 องศา ทุกๆ 2-3 วินาที จะเห็นว่าเซนเซอร์นั้นจะมีความแม่นยำในการวัดจะไม่เกิน 0.5 องศา และประสิทธิภาพโดยรวมของเซนเซอร์แนวแกน X มุม -45 ถึง 45 องศา มีค่าผิดพลาดที่น้อย เนื่องจากค่าความผิดพลาดเฉลี่ยมีค่าเพียง 0.3 องศา

4.3 ผลการทดลองปรับค่า K_p , K_i และ K_d ด้วยวิธีการลองผิดลองถูก (Trial and Error)

4.3.1 วิธีการทดลอง

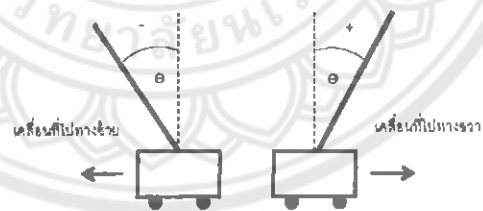
การทดลองนั้นจะใช้ตัวต้านทานปรับค่าได้ $10\text{ k}\Omega$ เป็นจำนวน 3 ตัว เพื่อใช้ในการปรับค่า P, I และ D โดยค่าที่ได้จะส่งไปยังบอร์ดไมโครคอนโทรลเลอร์ จากนั้นก็จะทำการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิทัล ซึ่งมีค่าเท่ากับ 0 ถึง 1023 เพื่อใช้คำนวณในสมการ PID ดังแสดงในรูป 4.4



รูปที่ 4.4 ตัวด้านทวนปรับค่าได้ที่ใช้ในการปรับค่า K_p , K_i และ K_d

การปรับค่าจะแบ่งเป็น 3 ช่วง

- ช่วงแรกจะปรับค่า I , D เป็นศูนย์ จากนั้นเพิ่มค่า P จนกระทั่งตัวรถเคลื่อนที่ถูกต้องและมีการตอบสนองเร็วพอตามทิศทางของเซนเซอร์ เช่น ค่ามุมที่เป็นลบให้ตัวรถเคลื่อนที่ไปทางซ้าย ในทำนองเดียวกัน ค่ามุมที่เป็นบวกให้รถเคลื่อนที่ไปทางขวา ดังแสดงในรูปที่ 4.5 เป็นต้น

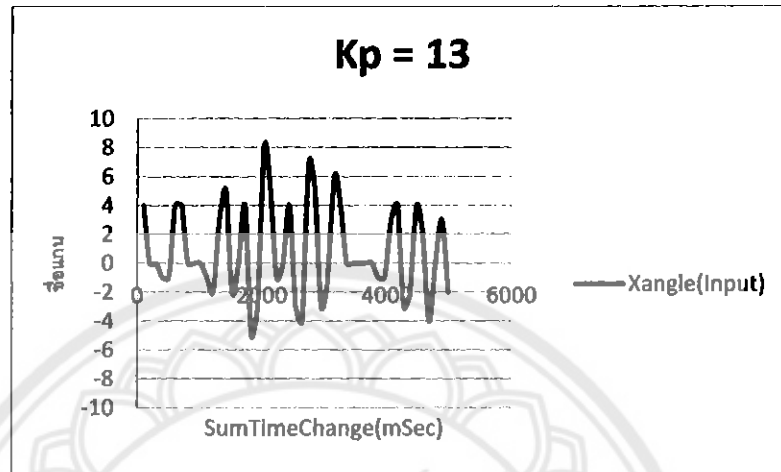


รูปที่ 4.5 แสดงการเคลื่อนที่ของรถตามทิศทางมุมเอียง

- ช่วงสองปรับค่า I จนกระทั่งรถเคลื่อนที่เข้าสู่จุดสมดุลที่ Rod สามารถทรงตัวในแนวตั้งได้ในเวลาที่เร็วขึ้น
- ช่วงสามปรับค่า D จนกระทั่งรถเคลื่อนที่เข้าสู่จุดสมดุลที่ Rod สามารถทรงตัวในแนวตั้งได้ในเวลาที่เร็วที่สุด

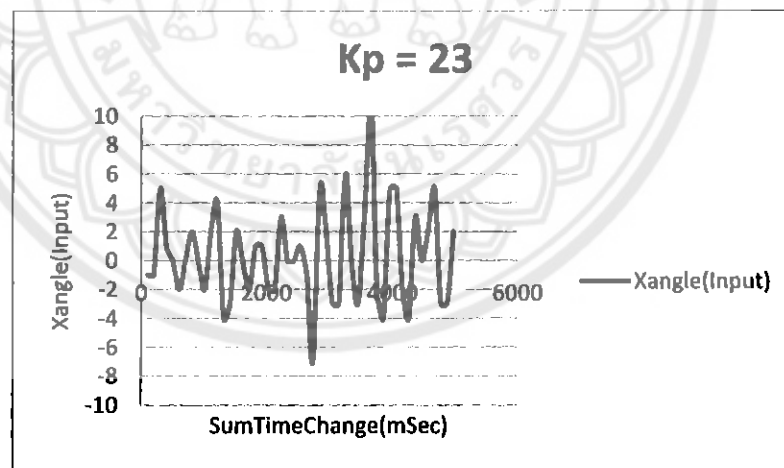
4.3.2 ผลการทดลอง

การปรับค่า K_p เพียงอย่างเดียว



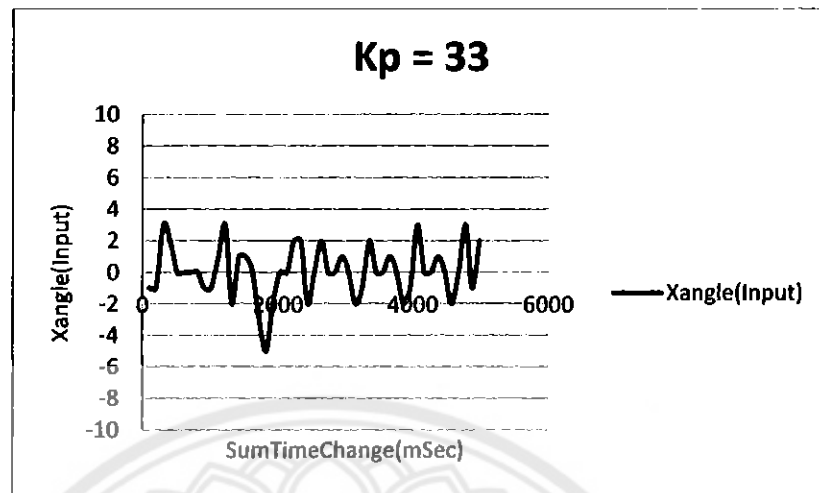
รูปที่ 4.6 กราฟแสดงค่า $K_p = 13$, $K_i = 0$, $K_d = 0$

จากรูปที่ 4.6 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -4 องศา ถึง 8 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวก จากการทดลองจะเห็นว่ารถยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 13



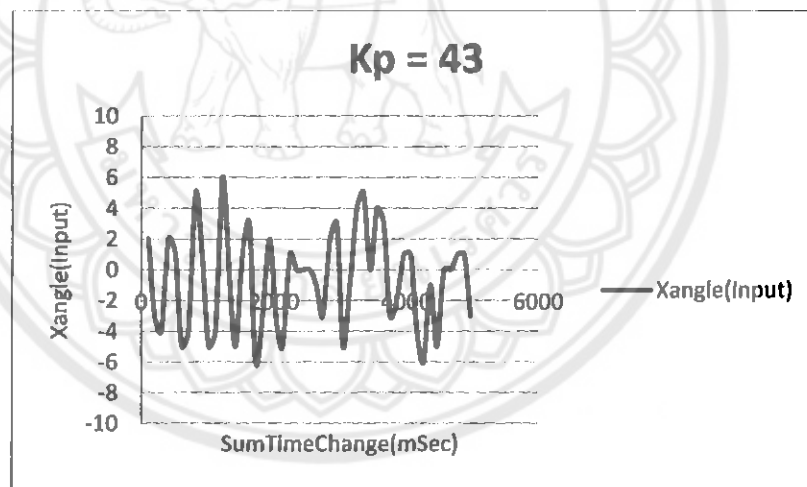
รูปที่ 4.7 กราฟแสดงค่า $K_p = 23$, $K_i = 0$, $K_d = 0$

จากรูปที่ 4.7 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -5 องศา ถึง 5 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวก จากการทดลองจะเห็นว่ารถยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 23



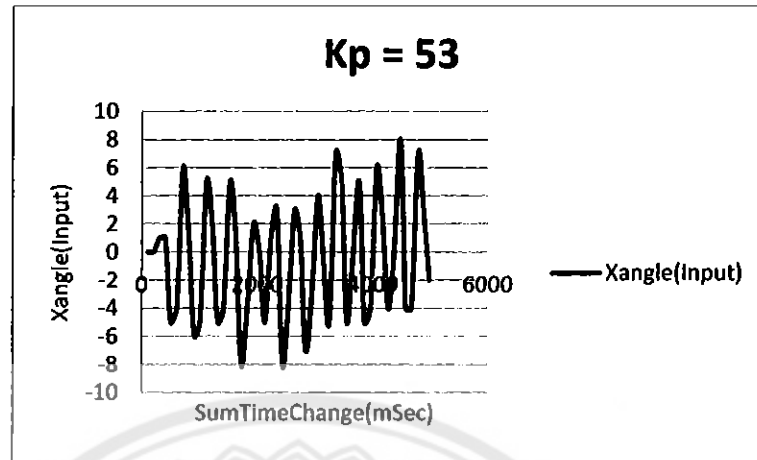
รูปที่ 4.8 กราฟแสดงค่า $K_p = 33$, $K_i = 0$, $K_d = 0$

จากรูปที่ 4.8 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -2 องศา ถึง 3 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวก จากการทดลองจะเห็นว่ารถยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 33



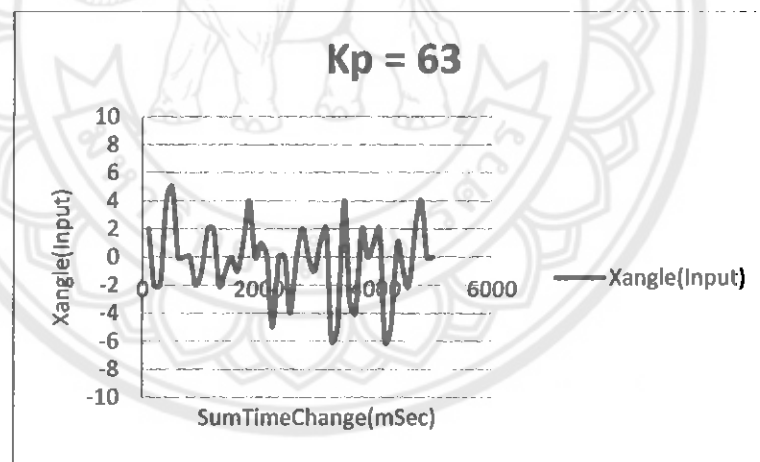
รูปที่ 4.9 กราฟแสดงค่า $K_p = 43$, $K_i = 0$, $K_d = 0$

จากรูปที่ 4.9 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -5 องศา ถึง 5 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ดีขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 43



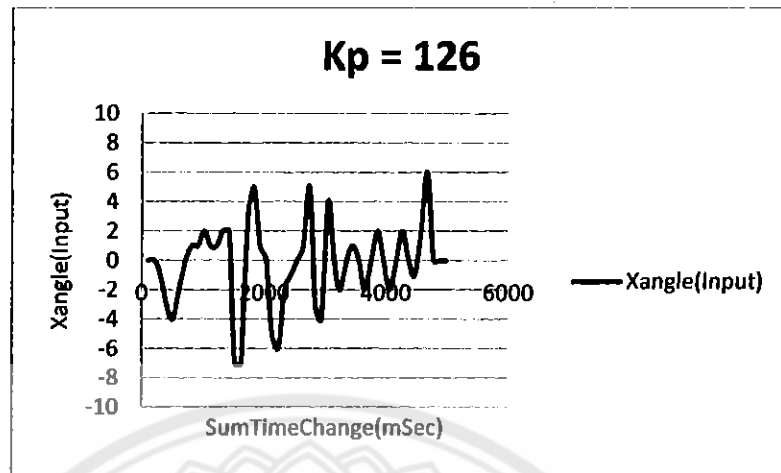
รูปที่ 4.10 กราฟแสดงค่า $K_p = 53$, $K_i = 0$, $K_d = 0$

จากรูปที่ 4.10 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -5 องศา ถึง 5 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ดีขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 53



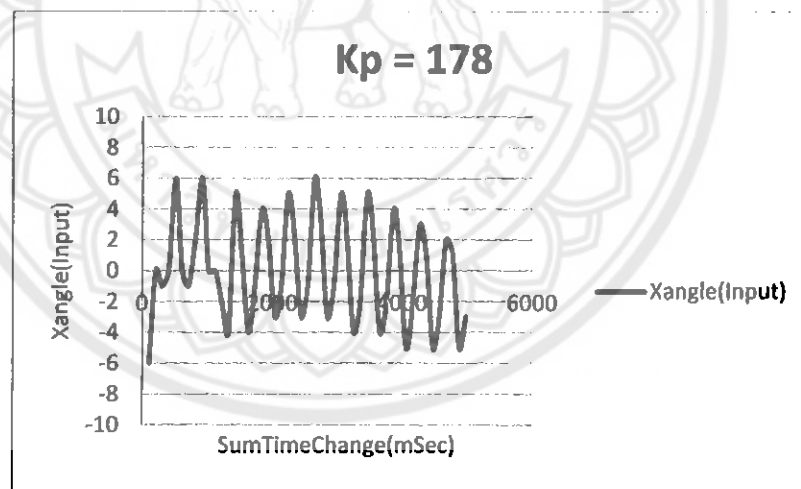
รูปที่ 4.11 กราฟแสดงค่า $K_p = 63$, $K_i = 0$, $K_d = 0$

จากรูปที่ 4.11 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -6 องศา ถึง 5 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ดีขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 63



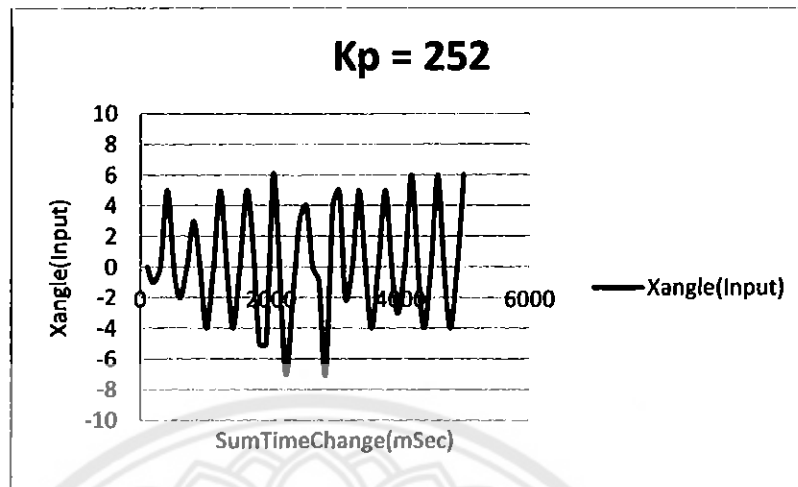
รูปที่ 4.12 กราฟแสดงค่า $K_p = 126$ $K_i = 0$, $K_d = 0$

จากรูปที่ 4.12 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -6 องศา ถึง 6 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้คืบขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 126



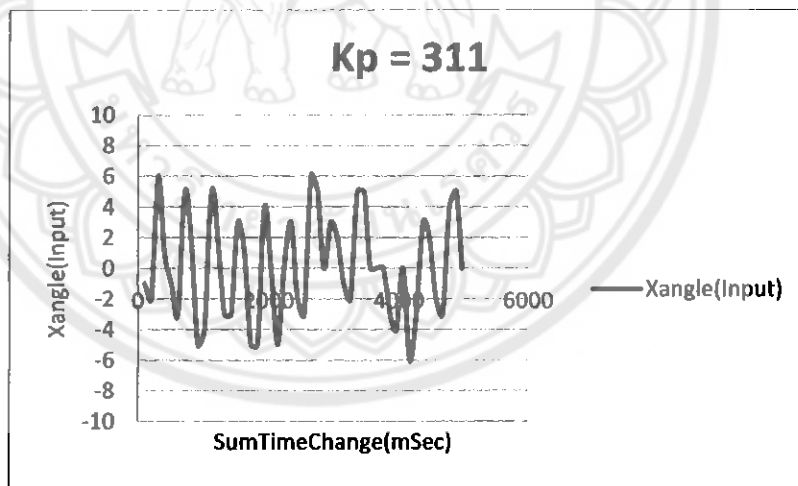
รูปที่ 4.13 กราฟแสดงค่า $K_p = 178$ $K_i = 0$, $K_d = 0$

จากรูปที่ 4.13 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -5 องศา ถึง 6 องศา ซึ่งเป็นการแกว่งที่ไม่สม่ำเสมอและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้คืบขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 178



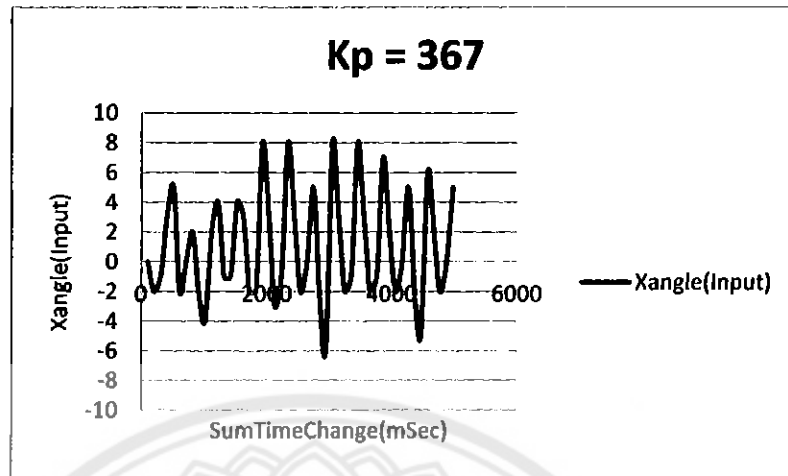
รูปที่ 4.14 กราฟแสดงค่า $K_p = 252$ $K_i = 0$, $K_d = 0$

จากรูปที่ 4.14 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -5 องศา ถึง 5 องศา ซึ่งเป็นการแกว่งที่ดีขึ้นจากค่า K_p ที่ผ่านมาและมีทิศทางแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ดีขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 252



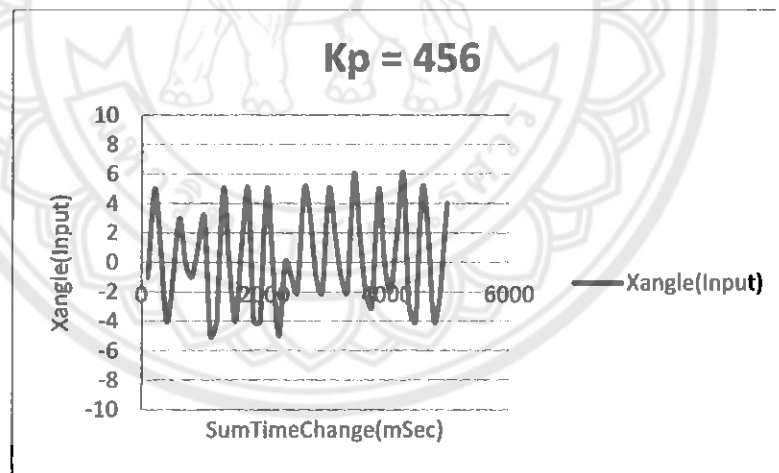
รูปที่ 4.15 กราฟแสดงค่า $K_p = 311$ $K_i = 0$, $K_d = 0$

จากรูปที่ 4.15 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -6 องศา ถึง 6 องศา ซึ่งเป็นการแกว่งที่ดีขึ้นจากค่า K_p ที่ผ่านมาและมีทิศทางแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ดีขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 311



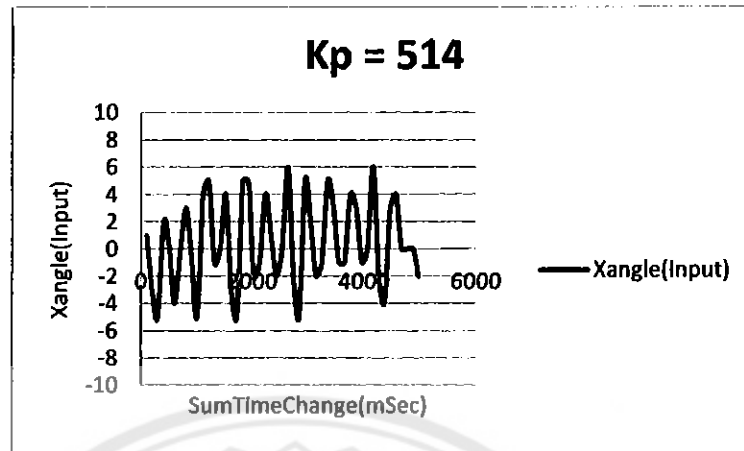
รูปที่ 4.16 กราฟแสดงค่า $K_p = 367$ $K_i = 0$, $K_d = 0$

จากรูปที่ 4.16 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -6 องศา ถึง 8 องศา ซึ่งเป็นการแกว่งที่ดีขึ้นจากค่า K_p ที่ผ่านมาและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ดีขึ้น แต่ยังไม่สามารถเคลื่อนที่ตามทิศทางของ Rod หัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 367



รูปที่ 4.17 กราฟแสดงค่า $K_p = 456$ $K_i = 0$, $K_d = 0$

จากรูปที่ 4.17 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -5 องศา ถึง 6 องศา ซึ่งเป็นการแกว่งที่ดีขึ้นกว่าค่า K_p ที่ผ่านมาและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ เป็นการแกว่งที่สม่ำเสมอ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ดีขึ้น และสามารถเคลื่อนที่ตามทิศทางของ Rod หัน เหมาะสมที่ใช้ค่า K_p เท่ากับ 456

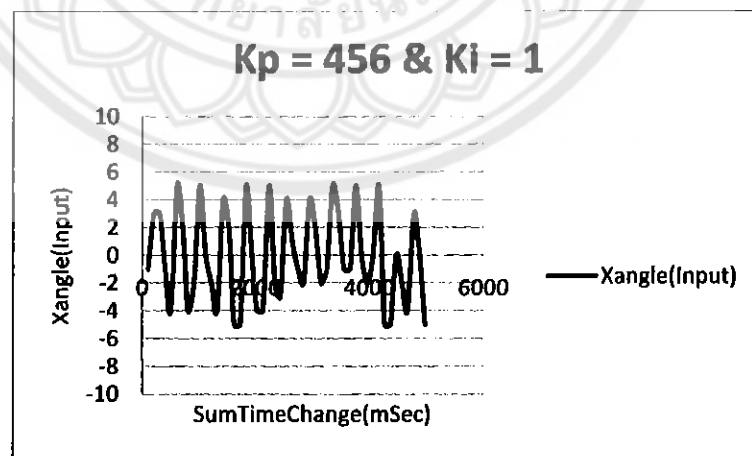


รูปที่ 4.18 กราฟแสดงค่า $K_p = 456$ $K_i = 0$, $K_d = 0$

จากรูปที่ 4.14 จะเห็นว่าช่วงของกราฟจะแกว่งอยู่ในช่วง -5 องศา ถึง 6 องศา ซึ่งเป็นการแกว่งที่มีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้ไม่ดีเท่ากับ K_p เท่ากับ 456 รูปที่ 4.17 จากการทดลองจะเห็นว่ารถมีการเคลื่อนที่เร็วเกิน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 514

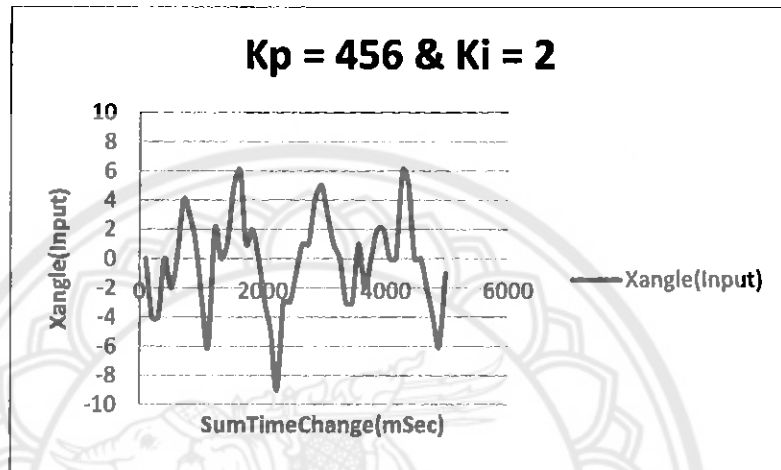
จากการทดลองจะเห็นว่าค่า K_p เท่ากับ 456 รูปที่ 4.17 ระบบจะมีการตอบสนองที่ดีที่สุด เพราะฉะนั้นจึงเลือกใช้ค่า K_p เท่ากับ 456

การปรับค่า K_p และ K_i



รูปที่ 4.19 กราฟแสดงค่า $K_p = 456$ $K_i = 1$, $K_d = 0$

จากรูปที่ 4.19 จะเห็นว่า การปรับค่า K_i เท่ากับ 1 จะมีผลทำให้กราฟมีการแกว่งอยู่ในช่วงค่ามุมเอียงเท่ากับ -5 องศา ถึง 5 องศา มีช่วงการแกว่งที่น้อยลง และจากผลการทดลองทำให้รถมีการตอบสนองให้เคลื่อนที่ไปมาได้เร็วขึ้นกว่าเดิม จึงเหมาะสมที่จะใช้ค่า K_p เท่ากับ 456 และ K_i เท่ากับ 1

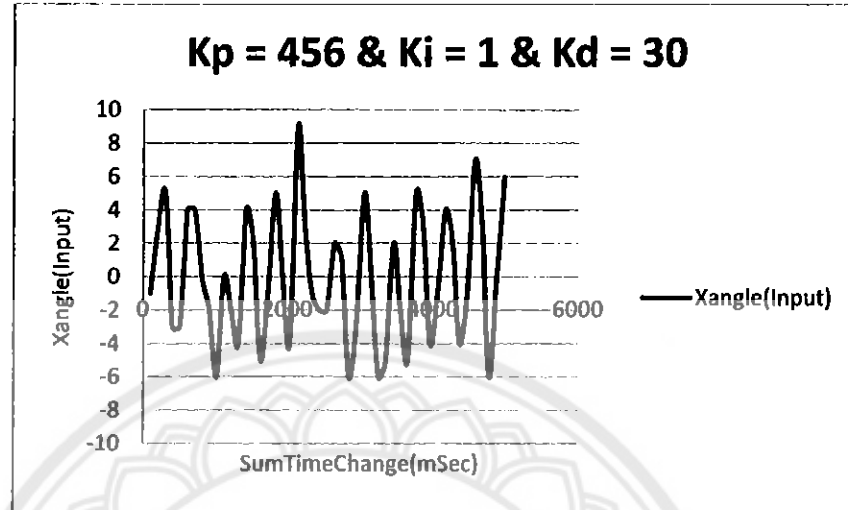


รูปที่ 4.20 กราฟแสดงค่า $K_p = 456$ $K_i = 2$, $K_d = 0$

จากรูปที่ 4.20 จะเห็นว่า การปรับค่า K_i เท่ากับ 2 มีผลทำให้ช่วงกราฟเกิดการสั่นไม่สามารถแกว่งให้ค่ามุมเอียงเป็นค่าบวกหรือค่าลบได้ มีผลทำให้รถไม่เคลื่อนที่ไปในทิศทางเดียวกับ Rod เอียงได้ ซึ่งทำให้ระบบไม่มีความเสถียรภาพเกิดขึ้น จึงไม่เหมาะสมที่จะใช้ค่า K_p เท่ากับ 456 และ K_i เท่ากับ 2 ได้

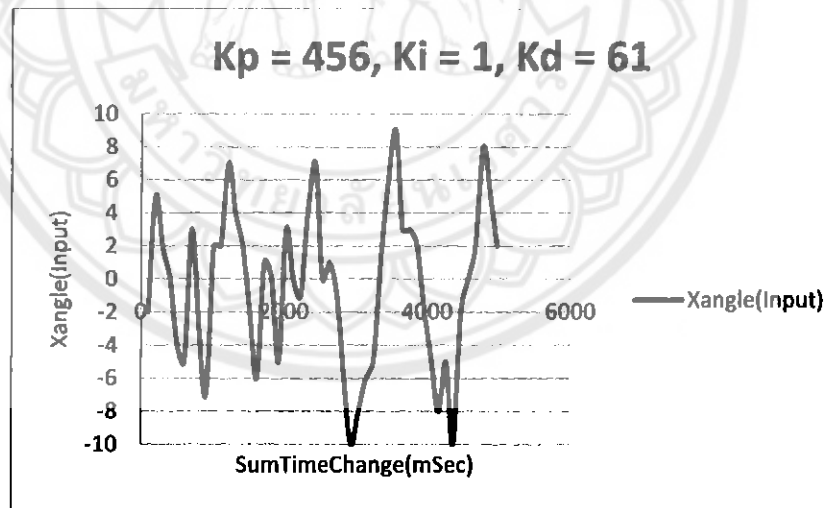
จากการทดลองจะเห็นว่า ถ้าค่า K_i มากกว่า 2 มีผลทำให้ระบบไม่มีความเสถียรภาพ จึงสรุปได้ว่าค่าที่เหมาะสมที่สุด คือค่า K_p เท่ากับ 456 และ K_i เท่ากับ 1

การปรับค่า K_p , K_i และ K_d



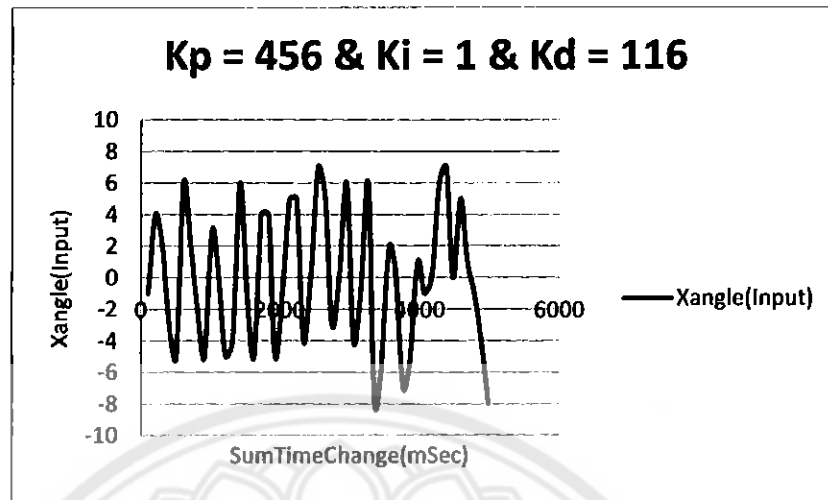
รูปที่ 4.21 กราฟแสดงค่า $K_p = 456$ $K_i = 1$, $K_d = 30$

จากรูปที่ 4.21 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 30 การตอบสนองของระบบยังคงไม่มีการเปลี่ยนแปลงที่ชัดเจน ซึ่งยังมีค่ามุมเอียงบางค่าไม่อยู่ในช่วงค่ามุมเอียงเท่ากับ -6 องศา ถึง 6 องศา



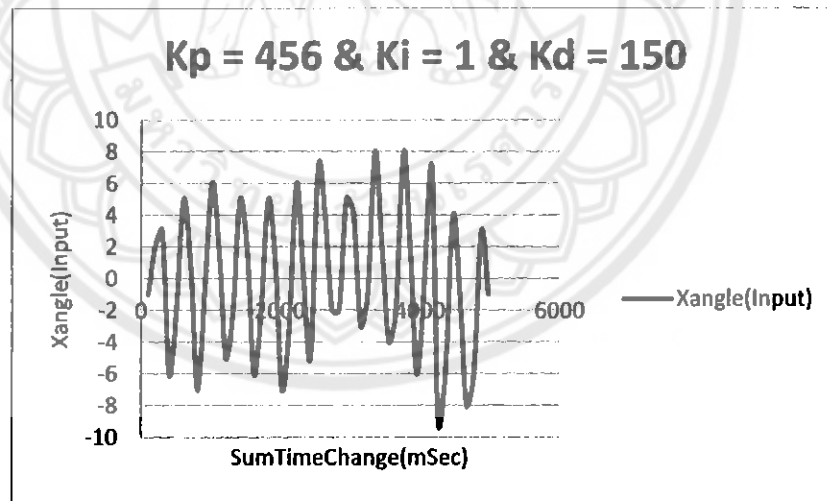
รูปที่ 4.22 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 61$

จากรูปที่ 4.22 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 61 ช่วงของค่ามุมเอียงเกิดการสั่น มีผลทำให้ระบบตอบสนองไม่ทัน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 456 K_i เท่ากับ 1 และ K_d เท่ากับ 61



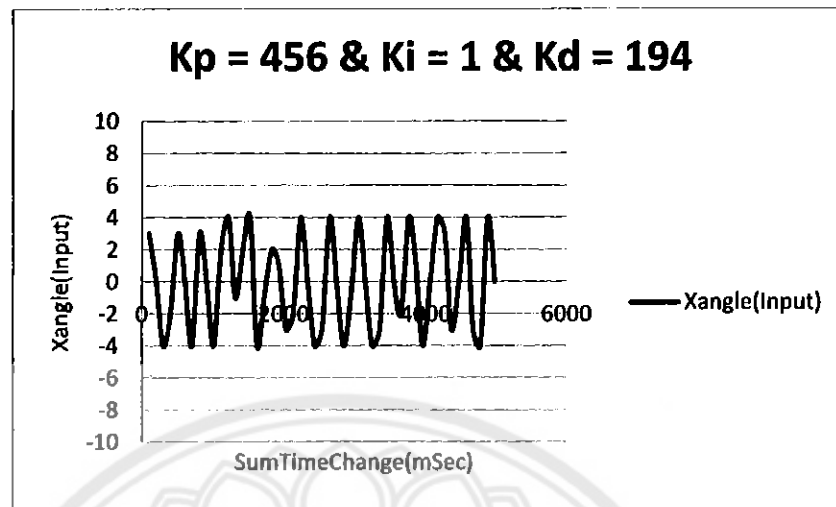
รูปที่ 4.23 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 116$

จากรูปที่ 4.23 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 116 ช่วงของค่ามุมเอียงบ้างค่ายังเกิดการสั่น เหมือนกับรูปที่แล้ว แต่มีการแกว่งค่ามุมเอียงเป็นค่าบวกค่าลบที่ถี่ขึ้นในช่วงแรก มีผลทำให้ระบบตอบสนองได้ดีในช่วงแรกของระบบเริ่มทำงาน จึงไม่เหมาะสมที่ใช้ค่า K_p เท่ากับ 456 K_i เท่ากับ 1 และ K_d เท่ากับ 116



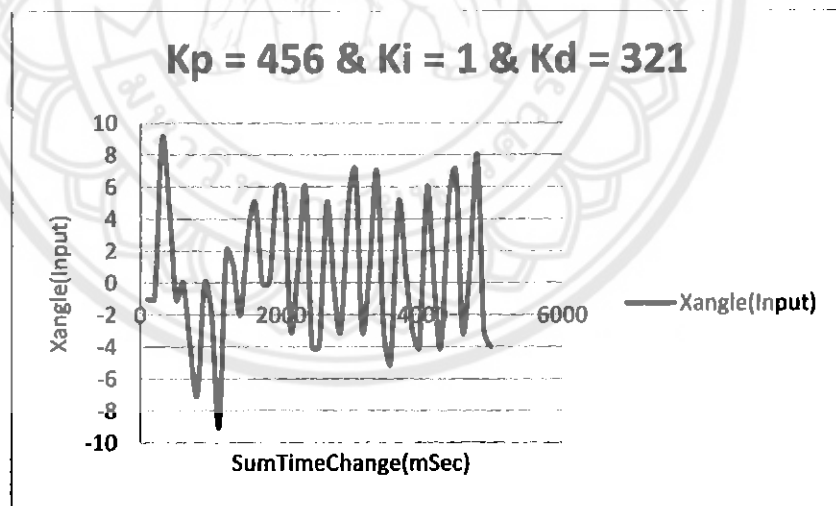
รูปที่ 4.24 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 150$

จากรูปที่ 4.24 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 150 มีผลทำให้ไม่มีการสั่นของช่วงมุมเอียง เหมือนกับรูปที่ 4.23 ซึ่งช่วงของกราฟอยู่ในช่วง -9 องศา ถึง 8 องศา



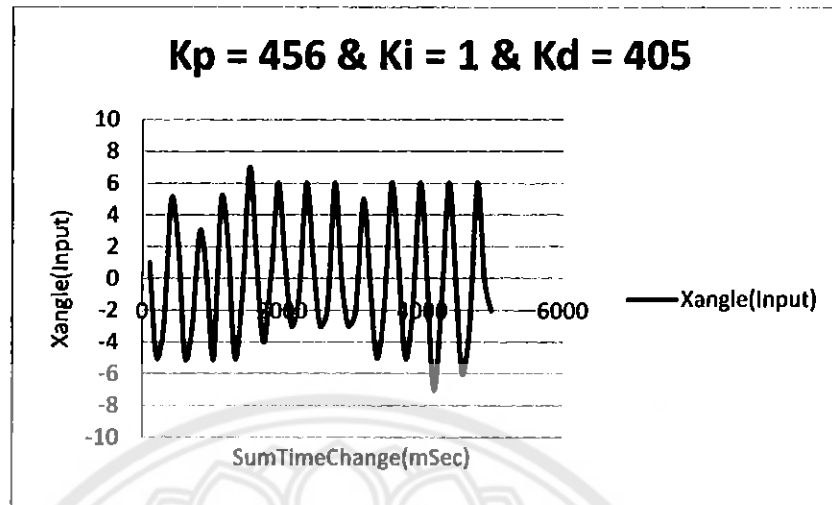
รูปที่ 4.25 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 194$

จากรูปที่ 4.25 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 194 มีผลทำให้การแกว่งของค่ามุมเอียงมีการแกว่งที่คืบขึ้นอยู่ในช่วงที่น้อยลงคือช่วง -4 องศา ถึง 4 องศา ซึ่งจากการทดลองจริงจะเคลื่อนที่ไปตามทิศทาง Rod ได้ดีขึ้น และได้เร็วพอสำหรับรองรับน้ำหนักของ Rod



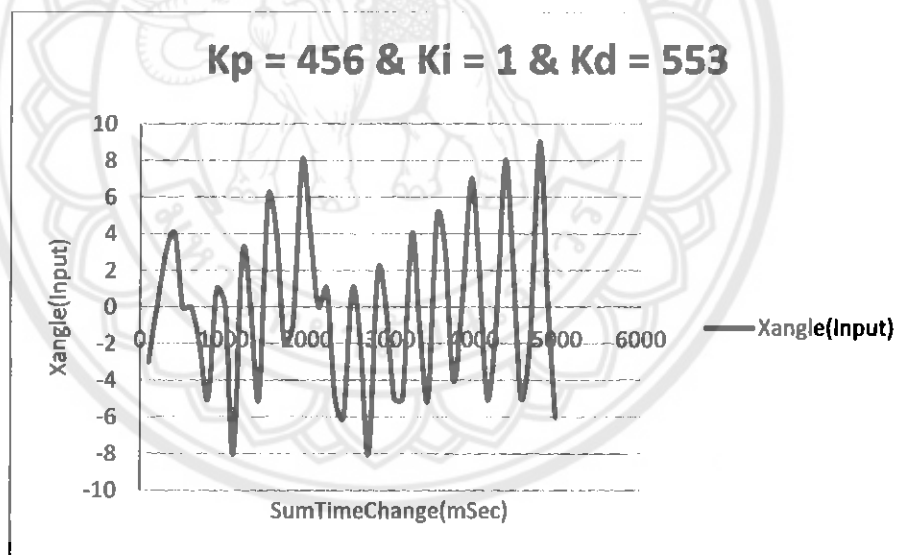
รูปที่ 4.26 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 321$

จากรูปที่ 4.26 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 321 มีผลทำให้การแกว่งของค่ามุมเอียงมีการแกว่งไม่สม่ำเสมออยู่ในช่วง -5 องศา ถึง 5 องศา และมีบางค่ามีค่าเท่ากับ 9 ประสิทธิภาพการตอบสนองลดลง เมื่อเทียบกับ รูปที่ 4.25



รูปที่ 4.27 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 405$

จากรูปที่ 4.26 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 405 มีผลทำให้การแกว่งของค่ามุมเอียงมีการแกว่งในช่วง -6 องศา ถึง 6 องศา ซึ่งมีช่วงการแกว่งที่เพิ่มมากขึ้น



รูปที่ 4.28 กราฟแสดงค่า $K_p = 456$, $K_i = 1$, $K_d = 553$

จากรูปที่ 4.26 จะเห็นว่าเมื่อปรับค่า K_d เท่ากับ 553 มีผลทำให้การแกว่งของค่ามุมเอียงมีการแกว่งอยู่ในช่วง -8 องศา ถึง 8 องศา ซึ่งมีช่วงการแกว่งที่เพิ่มมากขึ้น และเกิดการแกว่งไม่สม่ำเสมอ

4.3.3 วิเคราะห์ผลการทดลอง การปรับค่า K_p , K_i และ K_d

จากการปรับเฉพาะค่า K_p เพียงค่าเดียว เมื่อปรับค่าเพิ่มขึ้นจะทำให้ระบบเกิดการแกว่ง แกว่งช้า หรือแกว่งเร็วขึ้นอยู่กับค่า K_p ที่ปรับ เปรียบเสมือนการปรับค่าแบบหยาบ จะเห็นว่า รูปที่ 4.17 คีที่สุด มีค่า K_p เท่ากับ 456 จากรูปจะเห็นว่าช่วงของค่ามุมเอียงจะแกว่งอยู่ในช่วงที่คี่ขึ้นกว่าค่า K_p ที่ผ่านมาและมีทิศทางการแกว่งไปทางด้านค่ามุมเอียงที่เป็นบวกและลบ เป็นการแกว่งที่สม่ำเสมอ ซึ่งจากการทดลองจะเห็นว่ารถมีเคลื่อนไปเคลื่อนที่มาได้คี่ขึ้น และสามารถเคลื่อนที่ตามทิศทางของ Rod ทัน

การปรับค่า K_i เป็นการปรับค่าที่ทำให้ระบบตอบสนองเข้าสู่จุดสมดุลได้เร็วขึ้น คือเทอมปริพันธ์ เป็นผลรวมของความผิดพลาดในทุกช่วงเวลา เมื่อรวมกับเทอมสัดส่วนจะเร่งระบบเข้าสู่จุดสมดุลหรือจุดที่ต้องการเร็วขึ้น จากการปรับค่า K_p และ K_i จะเห็นว่ารูปที่ 4.19 คีที่สุด ซึ่งการปรับค่า K_i เท่ากับ 1 จะมีผลทำให้กราฟมีการแกว่งอยู่ในช่วงค่ามุมเอียงเท่ากับ -5 องศา ถึง 5 องศา มีช่วงการแกว่งที่น้อยลง และจากผลการทดลองทำให้รถมีการตอบสนองให้เคลื่อนที่ไปมาได้เร็วขึ้นกว่าเดิม และจากผลการทดลองค่า K_i มากกว่า 1 มีผลทำให้ระบบไม่เสถียร เพราะตัวควบคุมแบบอินทิกรัลนั้นเกิด $i = K_i * \text{errSum}$ ถ้าค่า K_i มากคูณกับค่าที่เกิดจากการสะสมของสัญญาณผิดพลาดหรือค่ามุมขึ้นเรื่อยๆเป็นเวลานาน จะทำให้ค่านี้นิ่งขึ้นเรื่อยๆ จะส่งผลทำให้ระบบควบคุมไม่เสถียร

การปรับค่า K_d เป็นการชะลออัตราการเปลี่ยนแปลงของสัญญาณขาออกของระบบควบคุม เนื่องจากเทอมอนุพันธ์ เป็นผลต่างระหว่างความผิดพลาดปัจจุบันและค่าผิดพลาดก่อนหน้า เมื่อเกิดการหักล้างระหว่างความผิดพลาดปัจจุบันน้อยกว่าความผิดพลาดก่อนหน้านี้นั้นมาก จะทำให้เทอมอนุพันธ์มีค่าเป็นลบ ส่งผลให้เมื่อรวมกับเทอมสัดส่วนและเทอมปริพันธ์ มีค่าลดลงทำให้เกิดการชะลออัตราการเปลี่ยนแปลงของสัญญาณขาออกของระบบควบคุมและด้วยผลนี้จะช่วยให้ระบบควบคุมเข้าสู่จุดที่ต้องการ การเหตุผลดังกล่าวทำให้เทอมอนุพันธ์เปรียบเสมือนการปรับค่าแบบละเอียด จากการปรับค่า K_p , K_i และ K_d จะเห็นว่ารูปที่ 4.25 คีที่สุด คือค่ามุมเอียงในกราฟมีการแกว่งที่คี่ขึ้นอยู่ในช่วงที่น้อยลงคือช่วง -4 องศา ถึง 4 องศา ซึ่งจากการทดลองรถจะเคลื่อนที่ไปมาตามทิศทาง Rod ได้คี่ขึ้น และเร็วพอสำหรับรองรับน้ำหนักของ Rod

4.4 บทสรุป

ในบทนี้ได้กล่าวถึงวิธีการทดลองและผลการทดลองของเซนเซอร์ MEMSIC 2125 โดยที่เซนเซอร์จะให้ค่าในการวัดที่ไม่นิ่ง มีอัตราการแกว่งขึ้นลงของค่าอยู่ที่ประมาณ 1-2 วินาทีแต่ค่าในการวัดมีความแม่นยำสูง โดยที่ค่าความผิดพลาดของการวัดในแนวแกน X เฉลี่ยมีค่าเท่ากับ 0.3 และค่า K_p , K_i และ K_d ที่ได้จากการปรับมีค่าเท่ากับ $K_p = 456$, $K_i = 1$, $K_d = 194$ เป็นรูปที่แกว่งเข้าใกล้จุดสมดุลมากที่สุด



บทที่ 5

สรุปผลการทดลอง

ระบบลูกคัมพคัมมีความสำคัญในด้านการศึกษาพื้นฐานของระบบควบคุม เนื่องจากระบบลูกคัมพคัมเป็นระบบที่มีข้อศึกษาอยู่หลายประการ เช่น เป็นระบบที่ไม่เป็นเชิงเส้น เป็นระบบที่ไม่มีเสถียรภาพ เป็นต้น ดังนั้นระบบดังกล่าวจึงเหมาะแก่การนำมาศึกษาเพื่อออกแบบตัวควบคุม ซึ่งระบบลูกคัมพคัมเป็นระบบที่ไม่มีเสถียรภาพ ดังนั้นผู้ดำเนินโครงการได้นำเอาทฤษฎีตัวควบคุมแบบพีไอดีมาใช้ เพื่อช่วยให้การทำงานของระบบลูกคัมพคัมมีความเสถียรภาพในการทรงตัวมากขึ้น สำหรับการทำงานของระบบควบคุมนั้นจะต้องทำการรักษาสมดุลของลูกคัม เพื่อให้ลูกคัมสามารถตั้งอยู่ได้ในแนวคิงตลอดเวลา โดยระบบจะต้องส่งแรงให้ฐานของลูกคัมมีการเคลื่อนที่ด้วยแรงความเร็วและเวลาที่เหมาะสม หากลูกคัมจะล้มไปทางด้านซ้าย ฐานของลูกคัมก็จะเคลื่อนตัวไปทางด้านซ้ายเพื่อเลี้ยงให้ลูกคัมตั้งอยู่ในแนวคิงดั้งเดิม ในทำนองเดียวกันเมื่อลูกคัมจะล้มไปทางด้านขวาฐานของลูกคัมก็จะเคลื่อนตัวไปทางด้านขวาเพื่อเลี้ยงให้ลูกคัมตั้งอยู่ในแนวคิงดั้งเดิม เมื่อฐานของลูกคัมมีการเคลื่อนที่กลับไปกลับมา ก็จะเกิดแรงเหวี่ยงของตัวลูกคัม แรงเหวี่ยงนี้จะช่วยให้ลูกคัมสามารถตั้งอยู่ในแนวคิงได้

5.1 สรุปผลการทดลอง

จากการทดลองบทที่ 4 นั้น แยกสรุปได้เป็น 2 ส่วนดังนี้

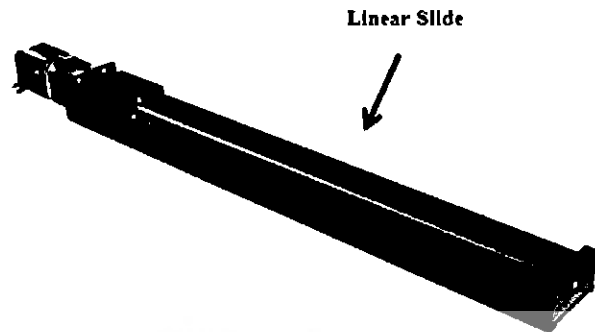
1. เซนเซอร์ MEMSIC 2125 ให้ความแม่นยำในการวัดสูง กล่าวคือมีการผิดพลาดในการวัดไม่เกิน 1 องศา ในแนวแกน X แต่ค่าไม่นิ่ง กล่าวคือค่าจะแกว่งขึ้นลง ประมาณ 1 องศา ทุกๆ 1 ถึง 2 วินาที
2. การปรับค่า PID จากการทดลองค่าที่ได้คือ $P=456$, $I=1$, $D=194$ นี้ไม่สามารถทำให้ลูกคัมพคัมตั้งอยู่ในแนวคิงได้

5.2 วิจัยรณัผลการทดลอง

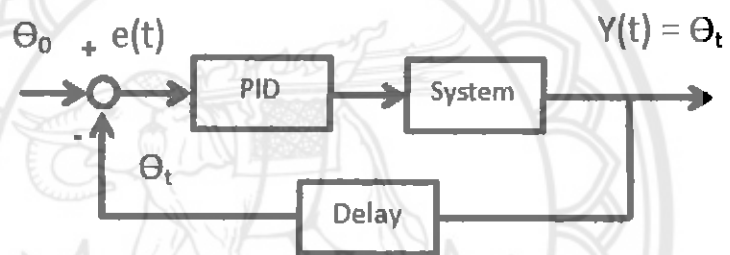
1. ความถูกต้องที่ได้จากการวัดจากเซนเซอร์ MEMSIC 2125 ในการวัดมุมเอียงในแนวแกน X นั้นจะมีความถูกต้องและแม่นยำสูง จากคู่มือรายละเอียดของเซนเซอร์ MEMSIC 2125 นั้นมุมที่เกิน -60 ถึง 60 องศา จะมีความถูกต้องของค่าที่ได้จากการวัดมุมเอียงนั้นจะลดลง [6]
2. สาเหตุที่ทำให้ระบบลูกตุ้มผกผันสามารถตั้งในแนวตั้งได้ในเวลาสั้นๆ เท่านั้น สามารถสรุปได้ดังนี้
 - 2.1 รวงเลื้อนมีขนาดสั้น วิธีแก้ไขคือการเปลี่ยนไปใช้รวงที่มีขนาดยาวเหมาะสมกับขนาดความยาวของ Rod
 - 2.2 รดบนรวงเลื้อนเคลื่อนที่ได้ไม่ดี เพราะรวงเลื้อนและตัวรวงเลื้อนวัสดุคนละชนิดกัน วิธีแก้ไขคือการเปลี่ยนไปใช้รวงเลื้อนเชิงเส้น ที่ใช้ในโรงงานอุตสาหกรรม
 - 2.3 ความเร็วในการทำงานของเซนเซอร์ MEMSIC2125 ช้า วิธีแก้ไขคือการเปลี่ยน ไปใช้ 9 Degrees of Freedom - Razor IMU (SEN-10736) [7] ซึ่งมีความเร็วในการวัดมุมเอียงเร็วกว่า MEMSIC2125
 - 2.4 ความเร็วของระบบซึ่งประมวลผลโดยไม่โครคอนโทรลเลอร์ช้า เพราะ Coding ไม่ดีพอทำให้ระบบเกิด Delay มีผลทำให้การประมวลผลโดยไม่โครคอนโทรลเลอร์ช้า

5.3 การพัฒนาโครงการต่อไปในอนาคต

จากการวิจัยผลการทดลองทำให้คณะผู้จัดทำได้เสนอแนวทางการพัฒนาโครงการต่อไปในอนาคต โดยการพัฒนารวงเลื้อนลูกตุ้มผกผันให้ดีขึ้น โดยการใชรวงเลื้อนเชิงเส้น ที่ได้มาตรฐานตามโรงงานอุตสาหกรรมที่มีความยาวเหมาะสมกับขนาดความยาวของ Rod ดังแสดงในรูปที่ 5.1 เปลี่ยนเซนเซอร์วัดมุมเอียงจาก MEMSIC2125 เป็น 9 Degrees of Freedom - Razor เพราะ 9 Degrees of Freedom - Razor เป็นเซนเซอร์วัดมุมเอียงที่มีความเร็วสูงและมีการประมวลผลวัดมุมเอียงในตัวเซนเซอร์ และหาวิธีการกำจัด Delay ของระบบ โดยสร้างแบบจำลองทางคณิตศาสตร์ ดังแสดงในรูปที่ 5.2



รูปที่ 5.1 รางเลื่อนเชิงเส้น (Linear slide)



รูปที่ 5.2 ภาพรวมการทำงานที่เกิด Delay ของระบบขึ้น

Θ_0 คือ มุมเอียงปัจจุบัน

Θ_t คือ มุมเอียงรอบที่แล้ว

$e(t)$ คือ ผลต่างระหว่างมุมเอียงปัจจุบันกับมุมเอียงรอบที่แล้ว

$Y(t)$ คือ เอาต์พุตของระบบ

เอกสารอ้างอิง

- [1] การทดลอง CS06 ลักษณะสมบัติของตัวควบคุม PID สืบค้นเมื่อวันที่ 22 กรกฎาคม 2555, จาก jitkomut.lecturer.eng.chula.ac.th/ee331/lab6_datasheet.pdf
- [2] Wikipedia.(5 มีนาคม 2555) ระบบควบคุมพีไอดี สืบค้นเมื่อวันที่ 22 กรกฎาคม 2555, จาก <http://th.wikipedia.org/wiki/ระบบควบคุมพีไอดี>
- [3] Mr.Adisak Chinawong (8 มกราคม 2543) มอเตอร์กระแสตรง(DC Motor) สืบค้นเมื่อวันที่ 23 กรกฎาคม 2555, จาก <http://www.adisak51.com/page21.html>
- [4] devesh@electroons.com DC motor direction control using L293d สืบค้นเมื่อวันที่ 15 พฤษภาคม 2556, จาก http://www.electroons.com/electroons/dc_motor_control.html
- [5] Canon Thailand เครื่องพิมพ์ Canon รุ่น PIXMA iX6560 สืบค้นเมื่อวันที่ 15 พฤษภาคม 2556, จาก <http://www.canon.co.th/personal/products/printers/inkjet-printers/pixma-ix6560?languageCode=TH#specificationAnchor>
- [6] MEMSIC2125 โมดูลวัดความเร่งแบบ 2 แกน สืบค้นเมื่อวันที่ 15 พฤษภาคม 2556, จาก <https://docs.google.com/viewer?a=v&q=cache:LNTi-QZ0RW0J:www.es.co.th/schematic/pdf/9000056.pdf+&hl=th&gl=th&pid=bl&srcid=ADGEESg0YFn05zah9FGaFCi9vgpCDfvU1m3gPq-On67K0ezhVDFV83W7UJZ4DuCs0qwrrer13xqKAT6ZHPKUm5rKFG-GcDVjAp2949u87Sxbo3fAdoV3l8JLZRnQtzOMDxvdyMgRkMorS&sig=AHIEtbSXkCSuA7pD2oX-OQI3-SLq-Tz3tw>
- [7] SEN-10736 (9 Degrees of Freedom - Razor IMU) สืบค้นเมื่อวันที่ 15 พฤษภาคม 2556, จาก <https://www.sparkfun.com/products/10736>
- [8] ET-BASE AVR EASY328 สืบค้นเมื่อวันที่ 15 พฤษภาคม 2556, จาก <http://www.etteam.com/product2009/ET-AVR/ET-BASE%20AVR%20EASY328.html>
- [9] L293D สืบค้นเมื่อวันที่ 15 พฤษภาคม 2556, จาก <http://www.robotstorehk.com/L293D.PDF>



ภาคผนวก ก
ไฟล์โปรแกรม

```
#include <math.h>

int xPin = 12;

int yPin = 13;

int Xroll;

double accelerationX, Xangle;

int BAUD = 9600;

int CENTER = 0;

int LEFT = 1;

int RIGHT = 2;

int headstatus=0;

unsigned long lastTime;

double Input, Output, timeChange ;

double errSum = 0;

double lastErr;

double kp, ki, kd, p, i, d;

double Setpoint = 0;

int ledPin = 13; //ok tested its on board

int motorPins[] = {A3, A4, A5};

//int posCenter = A1; //ok tested

int butSxPin = 8; //ok tested

int butDxPin = 9; //ok tested

int potentiometerKP = A1;
```

```
int potentiometerKI = A2;

int potentiometerKD = A0 ;

int butSxState = 0;

int butDxState = 0;

int fall = 0;

//PID CODE *****

void Compute()
{
    /*How long since we last calculated*/
    unsigned long now = millis();
    timeChange = (double)(now - lastTime);

    //timeChange = 0.1;
    /*Compute all the working error variables*/
    double error = Setpoint - Input;
    errSum += (error * timeChange);
    double dErr = (error - lastErr) / timeChange;

    /*Compute PID Output*/
    p = kp * error;
    i = ki * errSum;
    d = kd * dErr;
    Output = p + i + d;

    /*Remember some variables for next time*/
    lastErr = error;
```

```
lastTime = now;

/*Set Limit Output*/
if(Output > 20000){
    Output = 20000;
}
if(Output < -20000){
    Output = -20000;
}
}

/*void SetTunings(double Kp, double Ki, double Kd)
{
    kp = Kp;
    ki = Ki;
    kd = Kd;
}*/

void autoGet(){
    if(fall == 4 || fall == -4){
        fall = 0;
        errSum = 0;
    }
}

void SetTunings(){
    double readPotenKP = analogRead(potentiometerKP);
```

```
kp = (double)readPotenKP;

//kp = 456;

int readPotenKI = analogRead(potentiometerKI);

ki = (int)readPotenKI;

//ki = 1;

int readPotenKD = analogRead(potentiometerKD);

kd = (int)readPotenKD;

//kd = 61;

}

/*****

void setup () {
  Serial.begin(BAUD);           //initialize serial port
  pinMode(butSxPin, INPUT);
  pinMode(butDxPin, INPUT);
  pinMode(motorPins[0], OUTPUT); //set motor pin as output
  pinMode(motorPins[1], OUTPUT);
  //pinMode(motorPins[2], OUTPUT);
  //pinMode(yPin, INPUT);
  pinMode(xPin, INPUT);
  //SetTunings(50,0,0); //Set PID
}

void loop () {

/*****

// Due to erroneous output of pulseIn();
```

```

// when x and y are calculated together

// pulseIn(); must be used twice.

Xroll = pulseIn (xPin, HIGH);

// Calculate G-force in Milli-g's.

accelerationX = (( Xroll / 10 ) - 500) * 8;

// uncomment next four lines for milli-g output.

// Calculate angle (radians) for both -x and -y axis.

Xangle = asin (accelerationX/ 1000.0 );

// Convert radians to degrees.

Xangle = Xangle * (360 / (2*M_PI));

Input = (int)Xangle; //Input Of PID
*****

//Serial.println("1.BEGIN DIGITAL READ");

butSxState = digitalRead(butSxPin);

butDxState = digitalRead(butDxPin);

if((butSxState == HIGH)&&(butDxState == LOW)) {

    headstatus=1; //MAX LEFT

    //ROLL BACK 2 STEP

    digitalWrite(motorPins[0], LOW);

    digitalWrite(motorPins[1], HIGH);

    analogWrite(motorPins[2], 255);

    digitalWrite(motorPins[0], LOW);

    digitalWrite(motorPins[1], HIGH);

    analogWrite(motorPins[2], 255);

} else if((butSxState == LOW)&&(butDxState == HIGH)) {

    headstatus=2; //MAX RIGHT

```

```
//ROLL BACK 2 STEP

digitalWrite(motorPins[0], HIGH);
digitalWrite(motorPins[1], LOW);
analogWrite(motorPins[2], 255);
digitalWrite(motorPins[0], HIGH);
digitalWrite(motorPins[1], LOW);
analogWrite(motorPins[2], 255);
} else if ((butSxState == LOW) && (butDxState == LOW)) {
    headstatus=3; //Both Click --> Impossible
} else if ((butSxState == HIGH) && (butDxState == HIGH)) {
    headstatus=4; //Not Click
}
autoGet();
SetTunings();
Compute(); //Compute PID

//*****Print value show Serial monitor*****a
Serial.print(" Xangle(Input) = ");
Serial.print(Input);

Serial.print(" Output=");
Serial.print(Output);

Serial.print(" Kp=");
Serial.print(kp);
```



```
Serial.print(" Ki=");  
Serial.print(ki);  
Serial.print(" Kd=");  
Serial.print(kd);  
Serial.print(" timeChange = ");  
Serial.print(timeChange);  
//Serial.print(" headstatus: ");  
//Serial.println(headstatus,DEC);  
Serial.print("\n");  
//*****  
if(Input > 4) fall = fall + 1;  
if(Input < -4) fall = fall -1;  
if(headstatus==4){  
  if (Output > 0) {  
    digitalWrite(motorPins[0], LOW);  
    digitalWrite(motorPins[1], HIGH);  
    int PWM = (int)map(Output, 1 , 20000, 130, 255);  
    analogWrite(motorPins[2], PWM);  
    //Serial.println("Left");  
  } else if (Output < 0){  
    digitalWrite(motorPins[0], HIGH);  
    digitalWrite(motorPins[1], LOW);  
    Output = abs(Output);  
    int PWM = (int)map(Output, 1 , 20000, 130, 255);  
    analogWrite(motorPins[2], PWM);  
    //Serial.println("Right");
```

```
}else if(Output == 0){  
    digitalWrite(motorPins[0], LOW);  
    digitalWrite(motorPins[1], LOW);  
    analogWrite(motorPins[2], 0);  
    //Serial.println("Center");  
}  
}  
}
```



ภาคผนวก ข

MEMSIC2125 โมดูลวัดความเร่งแบบ 2 แกน

โมดูล MXD2125 เป็นโมดูลวัดความเร่ง (accelerometer) ผลงานของ Memsic Inc (www.memsic.com) ภายในบรรจุตัวตรวจจับความเร่งแบบ 2 แกน สามารถวัดได้ทั้งความเร่งในเชิงพลศาสตร์ (dynamic acceleration) หรือความสั่นสะเทือน (vibration) และเชิงสถิตยศาสตร์ (static acceleration) ซึ่งก็คือความเร่งจากแรงโน้มถ่วงโลก (gravity)

คุณสมบัติของโมดูล MXD2125

- วัดความเร่งในย่าน $\pm 2g$
- ความละเอียดสูงกว่า $1mg$ ($g = 9.81 \text{ m/s}^2$)
- โครงสร้างตรวจจับความเร่งแบบ 2 แกน (ตั้งฉากกัน) ด้วยเทคโนโลยี CMOS
- คิดตั้งวงจรประมวลผลสัญญาณภายในตัวถึงเดียวกัน
- ทนการเปลี่ยนแปลงแบบเฉียบพลันที่ความเร่งถึง 50,000 g
- รับแรงดันไฟเลี้ยงได้ตั้งแต่ +3 ถึง +5.25 V

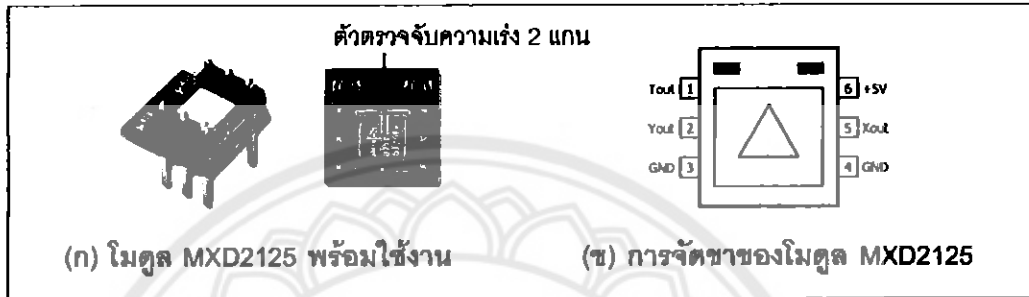
เนื่องจากตัวถังของ MXD2125 เป็นแบบ SMD ทำให้การนำไปใช้งานอาจไม่สะดวกนัก จึงได้นำ MXD2125GL มาติดตั้งบนแผ่นวงจรพิมพ์ จากนั้นต่อขาออกมาให้สามารถติดตั้งบนแผงต่อวงจรหรือเบรคบอร์ดได้เพื่ออำนวยความสะดวกในการทดลองและใช้งานในรูปแบบที่ 1 แสดงรูปร่างของโมดูลและการจัดขา

หลักการทำงาน

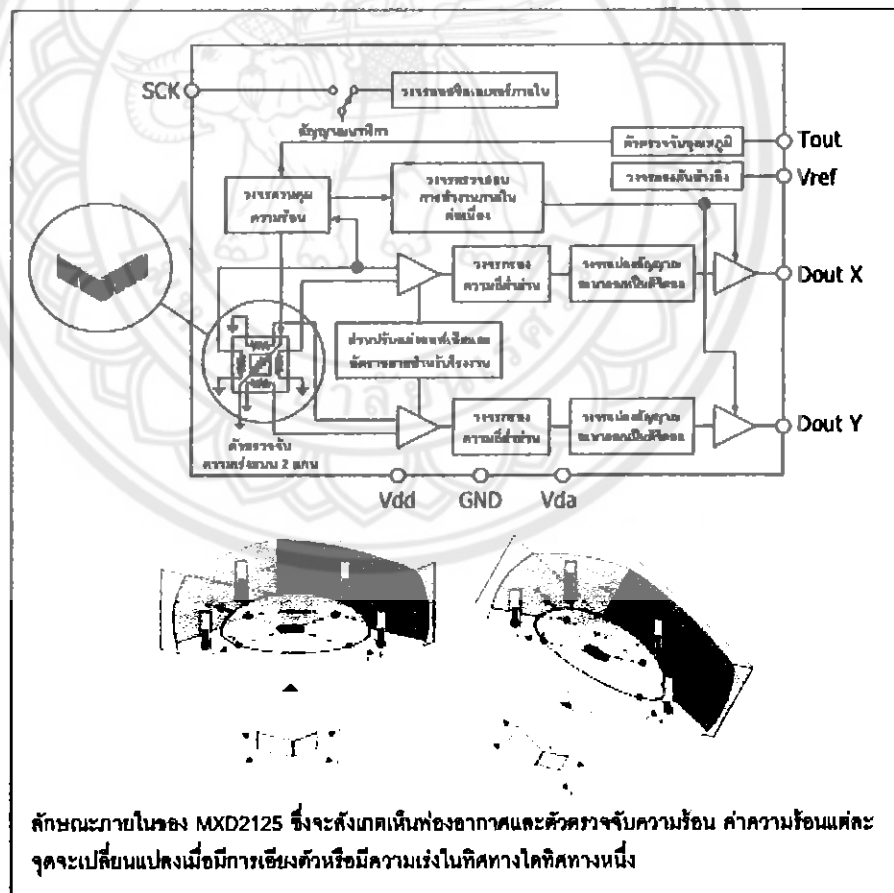
ภายใน MXD2125 ได้บรรจุตัวตรวจจับความเร่งแบบ 2 แกนที่ผลิตด้วยเทคโนโลยี CMOS ตัวตรวจจับความเร่งนี้ใช้หลักการนำพาความร้อนจากก๊าซที่บรรจุอยู่ภายในตัวตรวจจับความเร่งแบบ 2 แกน และมีตัวกำเนิดความร้อนที่แกนกลางสังเกตจากรูปที่ 2

ภายในตัวตรวจจับความเร่งแบบ 2 แกนมีชุดเทอร์โมคัปเปิลถูกจัดเรียงทั้งหมด 4 ชุด ในแต่ละแกน (แกน x และ y) ถูกเชื่อมต่อเข้าสู่วงจรประมวลผลสัญญาณเพื่อเปรียบเทียบความสมดุลในการกระจายตัวของความร้อน ซึ่งก็คือ การกระจายตัวของก๊าซภายในตัวตรวจจับ โดยเอาค่าหุตุจากการ

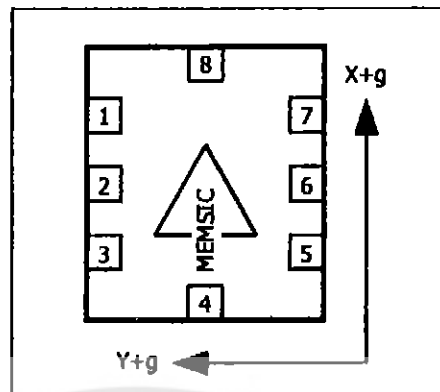
ตรวจจับความเร่งในแต่ละแกนจะถูกส่งออกทางขาสัญญาณ Dout X สำหรับแกน X และ Dout Y สำหรับแกน Y ในรูปแบบสัญญาณ PWM ที่มีค่าความถี่ที่เปลี่ยนแปลงแตกต่างกันไปตามสัดส่วนของปริมาณความเร่งในแต่ละแกน โดยทิศทางความเร่งจากการตรวจจับของแกน X และ Y เมื่อมองตัวตรวจจับจากด้านบนแสดงตามรูปที่ 3



รูปที่ 1 รูปร่างและการจัดขาของ MXD2125 โมดูลวัดความเร่ง



รูปที่ 2 แสดง โค้ดอะแกรมการทำงานภายใน โมดูล MXD2125



รูปที่ 3 แสดงทิศทางการเร่งในแนวแกน X และ Y ที่ตัวตรวจจับความเร่งวัดได้ เมื่อมองจากด้านบนของตัวตรวจจับ

เมื่อตัวตรวจจับ ไม่ได้เคลื่อนที่และอยู่ในแนวระนาบสัญญาณ PWM เอาต์พุตที่ได้ในแต่ละแกน จะมีค่าควิตซ์ไซเกิลเป็น 50% คาบเวลาของสัญญาณเท่ากับ 10 หรือ 2.5 มิลลิวินาทีขึ้นอยู่กับเบอร์ของตัวตรวจจับ สำหรับในที่นี้ใช้เบอร์ MXD2125GL จึงให้สัญญาณที่มีคาบเวลา 10 มิลลิวินาที มีสัดส่วนความไวต่อค่าควิตซ์ไซเกิลที่ 12.5 %/g

นอกจากการนำโมดูล MXD2125 ไปใช้ในการวัดความเร่งในแนวระวัดแล้ว นักพัฒนายังนิยมนำไปประยุกต์ใช้สำหรับงานตรวจจับความลาดเอียงของอุปกรณ์ต่างๆ ที่ติดตั้งเข้ากับโมดูล MXD2125 เพื่อควบคุมแนวการเคลื่อนที่ เช่น ควบคุมหุ่นยนต์ไม่ให้เอียงตัวมากเกินไป หรือเพื่อเพิ่มกำลังขับของหุ่นยนต์เมื่อเดินทางผ่านที่ลาดชัน หรือแม้กระทั่งควบคุมทิศทางการฉายภาพของโปรเจ็กเตอร์ในตำแหน่งมุมต่างๆ

MEMSIC2125 ใช้งานอะไรได้บ้าง

ใช้งานกับหุ่นยนต์

เพื่อวัดความเร่งและการเอียงของหุ่นยนต์

ระบบการตรวจวัดแรงสั่นสะเทือน

เพื่อวัดความสั่นสะเทือนจากเครื่องจักรขนาดใหญ่

ระบบการเตือนและตรวจจับการเคลื่อนไหว

ในโทรศัพท์เคลื่อนที่

เพื่อปรับเปลี่ยนหน้าจอให้หันทิศทางที่ถูกต้องกับผู้ใช้ตลอดเวลา

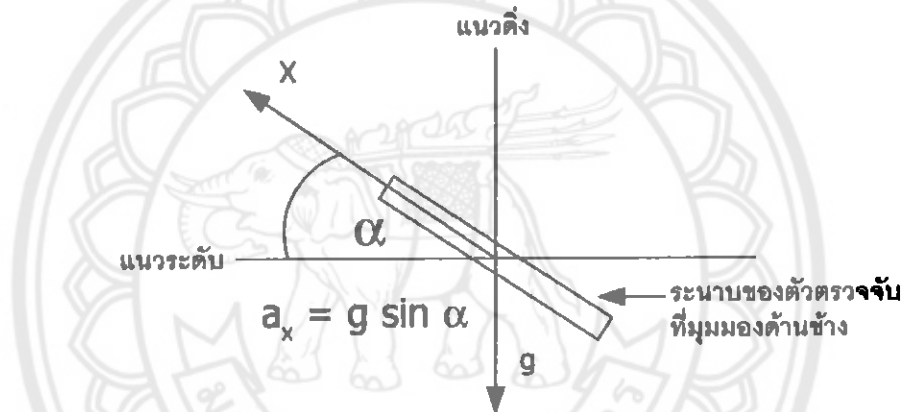
ในเครื่องบินบังคับวิทยุ

เพื่อตรวจสอบการยกตัวหรือเอียงตัวของเครื่องบิน (pitch and yaw)

ใช้ในไฟหน้าของรถยนต์เพื่อปรับระดับการเอียงของหลอดไฟหน้าอัตโนมัติ

ใช้ในระบบกันขโมยรถยนต์ ด้วยวิธีการตรวจจําการเอียงหรือการสั่นสะเทือน

การประยุกต์ใช้งานโมดูล MXD2125 ตรวจจําความลาดเอียง



เมื่อแกนใดๆ ของตัวตรวจจําเกิดการเอียงตัวทำมุมกับแนวตั้งของโลก จะส่งผลให้เกิดความไม่สมดุลของกลุ่มอนุภาคก๊าซที่บรรจุอยู่ภายในตัวตรวจจํา ทำให้การนำพาความร้อนภายในไม่สมดุล ส่งผลต่อการเปลี่ยนแปลงค่าควิตี้ไซเกิลของสัญญาณ PWM ที่ขาเอาต์พุตของแต่ละแกนที่เกิดการเอียงตัว โดยค่าควิตี้ไซเกิลที่เปลี่ยนแปลงไปจากสภาวะปกตินั้นจะสัมพันธ์กับมุมที่เอียงไปในแต่ละแกน

มมเอียง (องศา)	เอาต์พุตที่ขา Xout หรือ Yout (g)
90	1.000
85	0.996
80	0.985
70	0.940
60	0.866
45	0.707
30	0.500
20	0.342
10	0.174
5	0.087
0	0.000

ตารางที่ 1 แสดงค่า g ที่เอาต์พุต Xout หรือ Yout ของโมดูล MXD2125 ที่มุมเอียงค่าต่างๆ

การคำนวณมุมเอียงจากตัวตรวจจับ

สำหรับ โมดูลวัดความเร่ง MXD2125GL จะให้สัญญาณพัลส์ที่เอาต์พุตสำหรับขา X-out และ Y-out ที่มีคิวดั้ไซเกิล 50% ความถี่ 100Hz หรือมีความกว้างพัลส์ 10 มิลลิวินาที ในภาวะที่แกนนั้นๆ อยู่ในแนวระดับพอดี (ไม่เอียง) และจะให้ค่าคิวดั้ไซเกิลเปลี่ยนแปลงไป 12.5% เมื่อความเร่งเปลี่ยนแปลงไปเท่ากับ g พอดี โดยขนาดของมุมเอียงซึ่งพอจะเทียบเคียงกับความเร่งในหน่วย g ที่เอาต์พุต X-out และ Y-out แสดงไว้ตามตารางที่ 1

คั้งนั้นความเร่งเนื่องจากแกนใดแกนหนึ่งในหน่วยจำนวนเต็มเท่าของค่า g โดยสมมติให้เป็น a จะสัมพันธ์กับความกว้างของสัญญาณพัลส์บวกลบในหน่วยมิลลิวินาที ซึ่งกำหนดเป็น t ซึ่งสามารถสรุปเป็นสมการทางคณิตศาสตร์ได้คั้งนี้

$$a = \frac{\frac{t}{10} - 0.5}{0.125} \dots\dots\dots(1)$$

การคำนวณมุมเอียงเมื่อเทียบกับแกน X หรือ Y เอียงทำมุม α กับแนวระดับ จากรูปตัวอย่าง จากรูป เมื่อแกน X ของตัวตรวจจับทำมุมเงย α องศากับแนวระดับ สัญญาณพัลส์ที่อ่านได้จาก ขาสัญญาณ X-out เนื่องจากการเอียงตัวจะเทียบเคียงได้กับปริมาณความเร่ง a_x

โดย $a_x = g \sin \alpha$ ดังนั้น $\frac{a_x}{g} = \sin \alpha$

จะได้ $\alpha = \sin^{-1} \left(\frac{a_x}{g} \right) \dots\dots\dots(2)$

ช่วงมุม ในการวัด	k(arc/g)	ความผิดพลาด
±10	57.50	±0.02%
±20	58.16	±0.16%
±30	59.04	±0.48%
±40	60.47	±1.13%
±50	62.35	±2.24%

ตารางที่ 2 แสดงค่าความผิดพลาดที่เกิดขึ้นจากการตรวจจับความลาดเอียงของโมดูล MXD2125

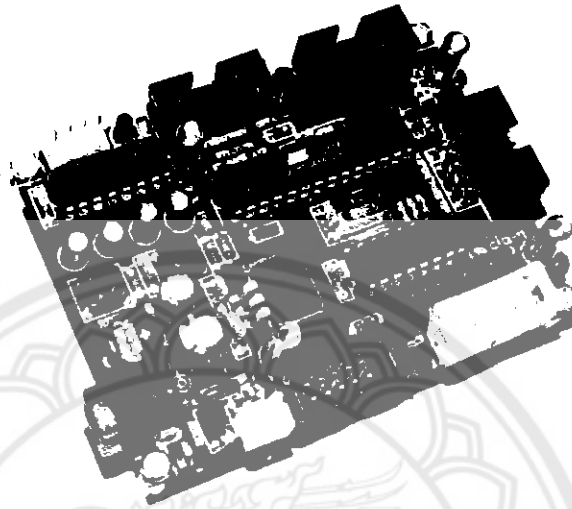
ค่าของ a_x สามารถคำนวณจากสมการ (1) โดยค่าที่ได้จะเป็นจำนวนเต็มเท่าของค่า g ซึ่งจะเป็นการหักล้างตัวส่วน g ในสมการ (2) ไปในตัว

การคำนวณหาค่ามุมเอียงจะให้ผลลัพธ์ได้แม่นยำในช่วง ±60 องศา ซึ่งถือว่ายังพอประมาณเป็นเชิงเส้นได้ เมื่อต้องการประมาณมุมเอียงจากสมการ (2) ให้เป็นสมการเชิงเส้นเพื่อเหตุผลบางประการ เช่น ประหยัดหน่วยความจำในการเขียน โปรแกรม หรือช่วยในการพัฒนาด้วยภาษาแอสเซมบลีซึ่งยากต่อการเขียน โปรแกรมย่อย $\arcsin(\sin^{-1})$ ตามสมการ(2) จึงลดรูปได้เป็น

$$\alpha = k a_x \dots\dots\dots(3)$$

ภาคผนวก ก

ET-BASE AVR EASY88



ET-BASE AVR EASY88 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR โดยบอร์ดเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATMEGA88 ของ ATMEL เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ในตัวถังแบบ 28 Pin DIP โดย MCU ตัวนี้จะมีจุดเด่น คือเป็นไมโครคอนโทรลเลอร์ขนาดเล็กแต่เทียบพร้อมไปด้วยทรัพยากรพื้นฐานต่างๆอย่างครบถ้วน เหมาะแก่การใช้ในการศึกษาเรียนรู้สำหรับผู้เริ่มต้น และยังสามารถนำไปประยุกต์ใช้งานต่างๆได้โดยง่าย ซึ่ง MCU สามารถทำงานได้ด้วยความถี่สูงสุด 20MHz ที่ 1 Clock / Machine Cycle นอกจากนี้แล้วยังมีความเทียบพร้อมด้วยอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งาน ไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลแบบ EEPROM ขนาด 512 Byte และหน่วยความจำใช้งานแบบ SRAM อีก 1 K Byte ส่วนในด้านของอุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วนเหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดีโดยจะมีทั้งระบบฮาร์ดแวร์ของ SPI, UART, I2C, Watchdog, Timer/Counter, PWM และ ADC ฯลฯ

โดยการออกแบบโครงสร้างของบอร์ดนั้น จะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งาน และสะดวกต่อการพัฒนาโปรแกรม โดยลักษณะของบอร์ดจะเน้นความคุ้มค่าและมีความอ่อนตัว ทั้งด้านของการศึกษาทดลอง และ การนำไปประยุกต์ใช้งานจริงๆ โดยในด้านของการศึกษาทดลองนั้น สามารถเลือกซื้อบอร์ดทดลอง Input / Output ขนาดเล็กของ ET-MINI

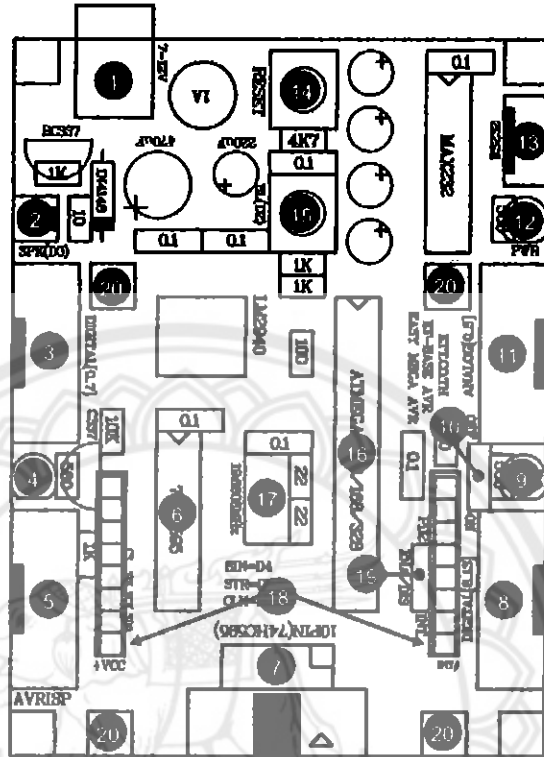
I/O แบบต่างๆที่ทาง อีทีที ออกแบบและผลิตขึ้นมาสนับสนุนเพื่อเป็นทางเลือกให้ผู้ใช้ได้เลือกชุดอุปกรณ์ที่ตนเองสนใจศึกษาทดลอง เพื่อนำมาติดตั้งใช้งานร่วมกับบอร์ดของ ET-BASE AVR EASY88 ได้อย่างง่ายดาย สำหรับกรณีที่จะนำบอร์ดไปเขียน โปรแกรมเพื่อสร้างเป็นชิ้นงานจริงนั้น ก็สามารถนำไปตัดแปลง หรือเชื่อมต่อกับอุปกรณ์ต่างๆได้โดยง่ายตามความเหมาะสม ซึ่งเรียกได้ว่า บอร์ดเคียวใช้ได้ทั้งเรียนรู้และใช้งาน

คุณสมบัติของบอร์ด

- เลือกใช้ MCU ตระกูล AVR เบอร์ ATMEGA88 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Crystal Oscillator ค่า 19.6608 MHz เพื่อให้สามารถใช้งานพอร์ตสื่อสารอนุกรม RS232 ได้อย่างลงตัว
- สามารถเปลี่ยนไปติดตั้งใช้งาน MCU ตระกูล AVR ขนาด 28 PIN เบอร์อื่นๆในอนุกรมเดียวกันได้เช่น ATMEGA8, ATMEGA48, ATMEGA168 และ ATMEGA328 เป็นต้น โดย MCU เหล่านี้จะมีตำแหน่งขาที่เข้ากันได้สามารถติดตั้งใช้งานในบอร์ดได้ทันทีโดยไม่ต้องตัดแปลงแก้ไขวงจร
- มีหน่วยความจำ 8 KByte Flash / 512 Byte EEPROM / 1024 Byte SRAM
- มีพอร์ต I/O ขนาด 20 บิต จำนวน 3 พอร์ต (PB(6บิต), PC(6บิต), PD(8Bit))
 - มีวงจรสื่อสารอนุกรม UART จำนวน 1 พอร์ต
 - มีวงจรสื่อสาร SPI จำนวน 1 พอร์ต
 - มีวงจรสื่อสาร I2C จำนวน 1 พอร์ต
 - มีวงจร Timer/Counter ขนาด 16 บิต 1 ชุด และ Timer/Counter ขนาด 8 บิต 2 ชุด
 - มีวงจร ADC ขนาด 10บิต จำนวน 6 ช่อง
- MCU ประจำบอร์ดได้รับการติดตั้ง Bootloader สำหรับใช้ Upload Code ให้บอร์ดผ่านทางพอร์ตสื่อสารอนุกรม RS232 ได้ทันที โดยไม่ต้องใช้เครื่องโปรแกรมภายนอก สามารถใช้การพัฒนาโปรแกรมได้ทั้ง ภาษาแอสเซมบลีภาษาซีภาษาเบสิกและ ภาษา C++ ของ Arduino ได้ทันที
- มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header 2x5 จำนวน 3 ชุด (PB, PC และ PD)

- มีสวิตช์ RESET หรือมวงจร External Reset แบบ RC Reset ภายในบอร์ด
- มีสวิตช์ BL(PD2) สำหรับใช้รีเซ็ตบอร์ดเข้าทำงานใน Bootloader และใช้ทดสอบ Input
- มีขั้วต่อ RS232 สำหรับใช้งาน และ Upload Code ด้วย Bootloader ผ่าน RS232
- มีขั้วต่อ AVRISP แบบ 10PIN IDE มาตรฐาน AVRISP สำหรับใช้โปรแกรม Code ให้บอร์ด ในกรณีที่ไม่ต้องการใช้การ Upload Code ผ่านทาง RS232 ของ Bootloader
- มีวงจรมหา Output ด้วย 74HC595 พร้อมขั้วต่อสัญญาณ Output แบบ IDE 10 Pin จำนวน 1 ชุด
- มี LED แสดงสถานะแหล่งจ่าย Power และ 2 LED สำหรับใช้ทดสอบการทำงานของบอร์ด
- ใช้กับแหล่งจ่ายไฟ 7 ถึง 10VAC/DC หรือมวงจร Bridge Rectifier และ Regulate 1A(Low Drop)
- มีฐานสำหรับใช้ติดตั้งบอร์ดทดลอง I/O ขนาดมาตรฐาน ET-MINI I/O SET เพื่อใช้งานร่วมกับบอร์ด
- ทดลองต่างๆเช่น DC Motor, Stepping Motor, I2C RTC,I2C I/O,I2C EEPROM, SPI ฯลฯ
- ขนาด PCB Size เล็กเพียง 8 x 6 Cm.

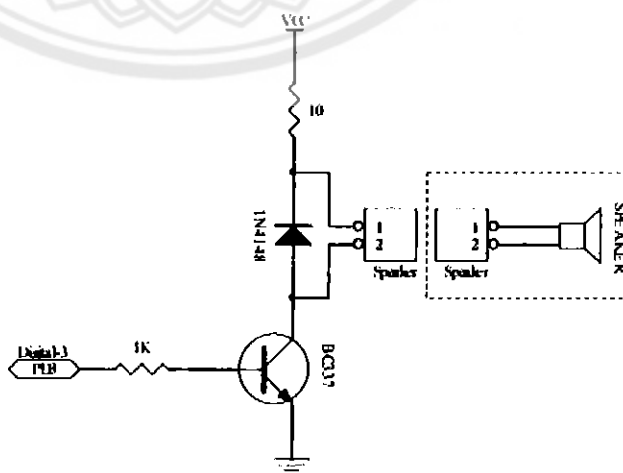
โครงสร้างบอร์ด ET-BASE AVR EASY88

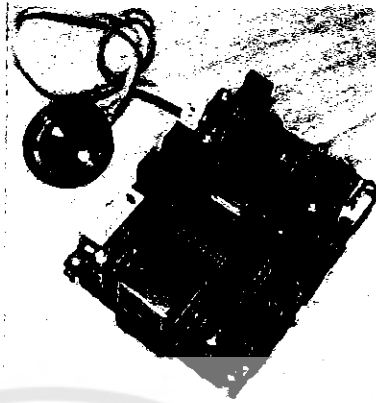


รูปที่ 1 แสดง โครงสร้างของบอร์ด ET-BASE AVR EASY88

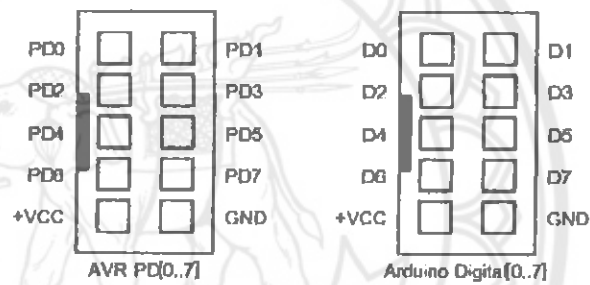
- หมายเลข 1 คือขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่าย 7-10VAC/DC
- หมายเลข 2 เป็นขั้วต่อสำหรับใช้ต่อกับ ลำโพงซึ่งถูก Drive ผ่านทรานซิสเตอร์ BC337 ดัง

วงจร

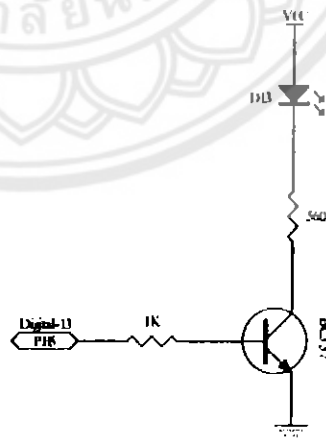




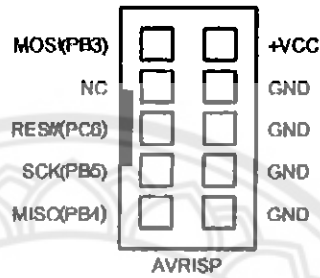
- หมายเลข 3 เป็นขั้วต่อสัญญาณจาก PD[0..7] ซึ่งในกรณีนี้ใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Digital[0..7]



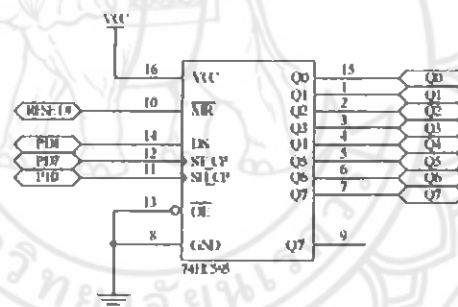
- หมายเลข 4 เป็น LED ใช้แสดงสถานะของขาสัญญาณ PB[5] หรือ Digital[13] ของ Arduino ซึ่ง LED นี้จะถูก Drive ผ่านทรานซิสเตอร์ BC337 ดังวงจร



- หมายเลข 5 เป็นขั้วต่อ AVRISP ใช้สำหรับ Download Code ให้กับ MCU ในกรณีที่ใช้การพัฒนาโปรแกรมของบอร์ดเป็นแบบ MCU ของ AVR ตามปกติโดยไม่ผ่านระบบ Bootloader โดยขั้วต่อ AVRISP นี้จะสามารถใช้งานได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ ATMEGA88 และใช้ขั้วต่อ ตรงตามมาตรฐาน AVRISP ดังรูป

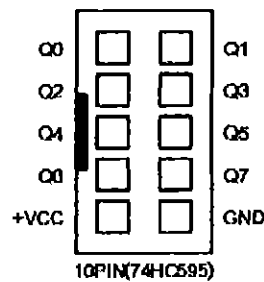


- หมายเลข 6 เป็นไอซีเบอร์ 74HC595 ซึ่งใช้ขยาย Output Port ขนาด 8 บิต โดยมีการต่อวงจรร่วมกับ MCU ที่ใช้ในบอร์ด ดังวงจร

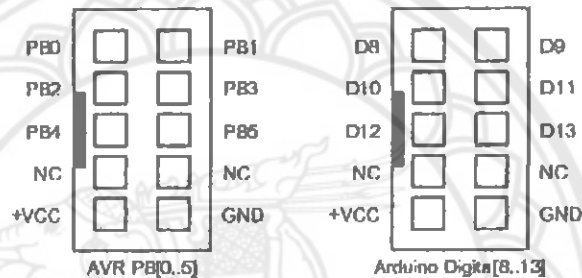


โดย Output ของ 74HC595 นี้สามารถนำไปประยุกต์ใช้งานเพื่อทำหน้าที่เป็น Output ทั่วไป หรือใช้สำหรับเชื่อมต่อกับ Character LCD ในแบบ 4 Bit Mode ก็ได้เช่นเดียวกัน

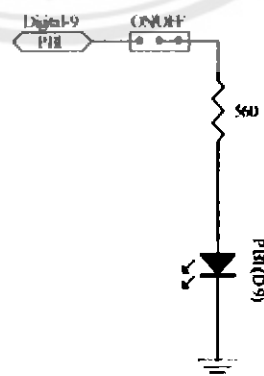
- หมายเลข 7 เป็นขั้วต่อสัญญาณ Output จาก 74HC595 ซึ่งมีขนาด 8บิต ก็คือ Q[0..7] โดยมี การจัดเรียงขาสัญญาณดังนี้



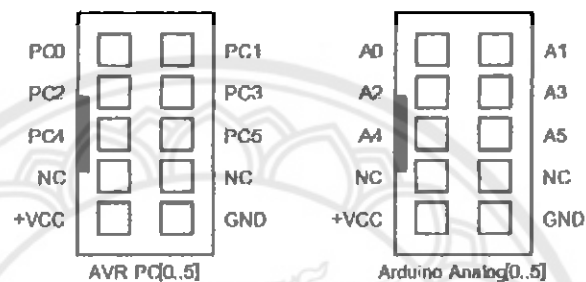
- หมายเลข 8 เป็นขั้วต่อสัญญาณจาก PB[0..5] ซึ่งในกรณีนี้ใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Digital[8..13]



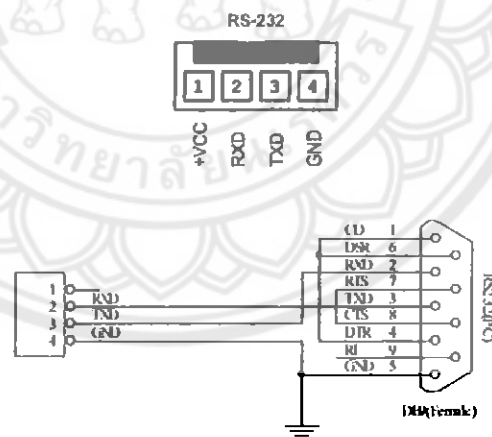
- หมายเลข 9 เป็น LED ใช้แสดงสถานะของขาสัญญาณ PB[1] หรือ Digital[9] ของ Arduino ซึ่ง LED นี้จะถูกดึงวงจรแบบ Source Current จากขาสัญญาณของ MCU โดยมี Jumper เป็นตัวตัดต่อสัญญาณระหว่าง PB[1] กับ LED ซึ่ง LED นี้สามารถแสดงผลได้ 2 แบบ คือ ใช้ทดสอบการแสดงผลแบบ ON/OFF เมื่อกำหนดขาสัญญาณ PB[1] เป็นแบบ Digital Output และใช้ทดสอบการแสดงผลแบบ Dimmer เมื่อกำหนดขาสัญญาณ PB[1] เป็นแบบ Output PWM ดังวงจร



- หมายเลข 10 เป็น Jumper สำหรับใช้ในการติดต่อสัญญาณ PB[1] กับ LED โดยเมื่อเลือกไว้ด้าน ON จะเป็นการต่อสัญญาณ PB[1] เข้ากับ LED แต่เมื่อเลือก OFF จะเป็นการตัดการเชื่อมต่อของ PB[1] ออกจากวงจรแสดงผลของ LED
- หมายเลข 11 เป็นขั้วต่อสัญญาณจาก PC[0..5] ซึ่งในกรณีใช้การพัฒนาโปรแกรมด้วย Arduino จะเป็นขาสัญญาณของ Analog[0..5]



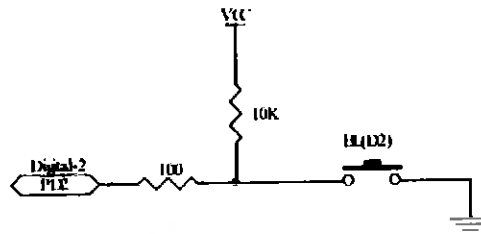
- หมายเลข 12 เป็น LED Power ใช้แสดงสถานะของแหล่งจ่ายไฟ +5VDC
- หมายเลข 13 คือขั้วต่อ RS232 สำหรับใช้งานทั่วไป และ Upload Code ให้กับ MCU ผ่านระบบ Bootloader โดยมีการจัดเรียงสัญญาณดังนี้



รูปแสดง วงจรการต่อสายของ RS232 สำหรับใช้งานกับบอร์ด

- หมายเลข 14 คือ สวิตช์ RESET ใช้สำหรับรีเซ็ตการทำงานของ MCU
- หมายเลข 15 คือ สวิตช์ BL(Bootloader) โดยต่อผ่านขาสัญญาณ PD[2] ใช้สำหรับสร้างสัญญาณ โลกิก LOW ให้กับขาสัญญาณ PD[2] เพื่อทดสอบการรับค่า Input รวมทั้งการ

สร้างสัญญาณ Trigger Interrupt ของ INT0 รวมทั้งการใช้สิ่งให้ MCU เข้าทำงานใน Bootloader โดยใช้งานร่วมกับสวิทช์ RESET โดยสวิทช์ BL มีการต่อวงจรรูป



- หมายเลข 16 เป็น MCU ประจําบอร์ด ซึ่งสามารถใช้ได้กับ AVR ขนาด 28ขาได้หลายเบอร์ เช่น ATMEGA8, ATMEGA88, ATMEGA168 และ ATMEGA328
- หมายเลข 17 เป็น Crystal Oscillator ค่าความถี่ 19.6608 MHz
- หมายเลข 18 เป็น Header สำหรับรองรับการเชื่อมต่อสัญญาณกับบอร์ด ET-MINI ENC28J60 ของ บริษัท อีทีที จำกัด สำหรับใช้พัฒนาโปรแกรมใช้งานกับระบบ Ethernet LAN
- หมายเลข 19 เป็น Jumper สำหรับใช้ตัดต่อขาสัญญาณของ PD2(INT0) ที่เชื่อมต่อระหว่าง PD2(INT0) ของบอร์ด ET-BASE AVR EASY88 กับ INT ของบอร์ด ET-MINI ENC28J60 ซึ่งถ้าเลือกไว้ด้าน ENA หมายถึง Enable ซึ่งจะเป็นการเชื่อมต่อขา INT จาก ENC28J60 เข้ากับขา PD2 หรือ INT0 ของ ATMEGA88 แต่เมื่อเลือกไว้ทางด้าน DIS จะหมายถึง Disable ซึ่งเป็นการตัดการเชื่อมต่อขา INT ของ ENC28J60 ออกจากขา PD2(INT0) ของ ATMEGA88 ซึ่งตามปรกติควรเลือกไว้ที่ด้าน DIS เสมอ
- หมายเลข 20 เป็นตำแหน่งฐานรองสำหรับบอร์ดทดลองขนาดเล็กของ บริษัท อีทีที จำกัด ที่มีขนาดมาตรฐาน ในขนาด MINI I/O Size ซึ่งผู้ใช้สามารถนำบอร์ดชุด ET-MINI I/O ต่างๆมาต่อทดลองร่วมกับบอร์ด ET-BASE AVR EASY88 ได้ทันที

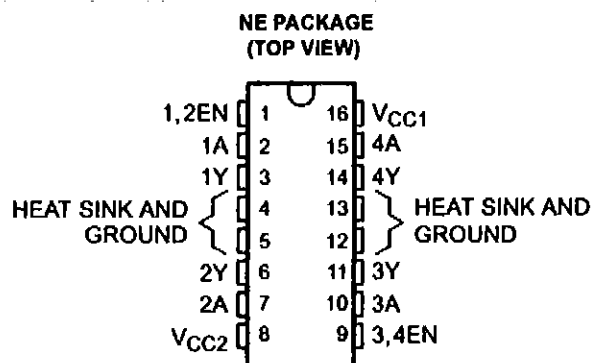
ภาคผนวก ง

L293D

- 600-mA Output Current Capability Per Driver
 - Pulsed Current 1.2-A Per Driver
 - Output Clamp Diodes for Inductive Transient Suppression
 - Wide Supply Voltage Range 4.5 V to 36 V
 - Separate Input-Logic Supply
 - Thermal Shutdown
 - Internal ESD Protection
 - High-Noise-Immunity Inputs
- Functional Replacement for SGS L293D

Description

The L293D is a quadruple high-current half-H driver designed to provide bidirectional drive currents of up to 600 mA at voltages from 4.5 V to 36 V. It is designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL-compatible. Each output is a complete totem-pole drive circuit with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4 EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. External high-speed output clamp diodes should be used for inductive transient suppression. When the enable input is low, those drivers are disabled, and their outputs are off and in a high-impedance state. With the proper data inputs, each pair of drivers form a full-H (or bridge) reversible drive suitable for solenoid or motor applications. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293D is designed for operation from 0°C to 70°C



รูปที่ 1 IC L293D

FUNCTION TABLE
(each driver)

INPUTS [†]		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low level,
 X = irrelevant, Z = high-impedance (off)
[†] In the thermal shutdown mode, the
 output is in the high-impedance state
 regardless of the input levels.

รูปที่ 2 ตารางแสดงการทำงานของ IC L293D