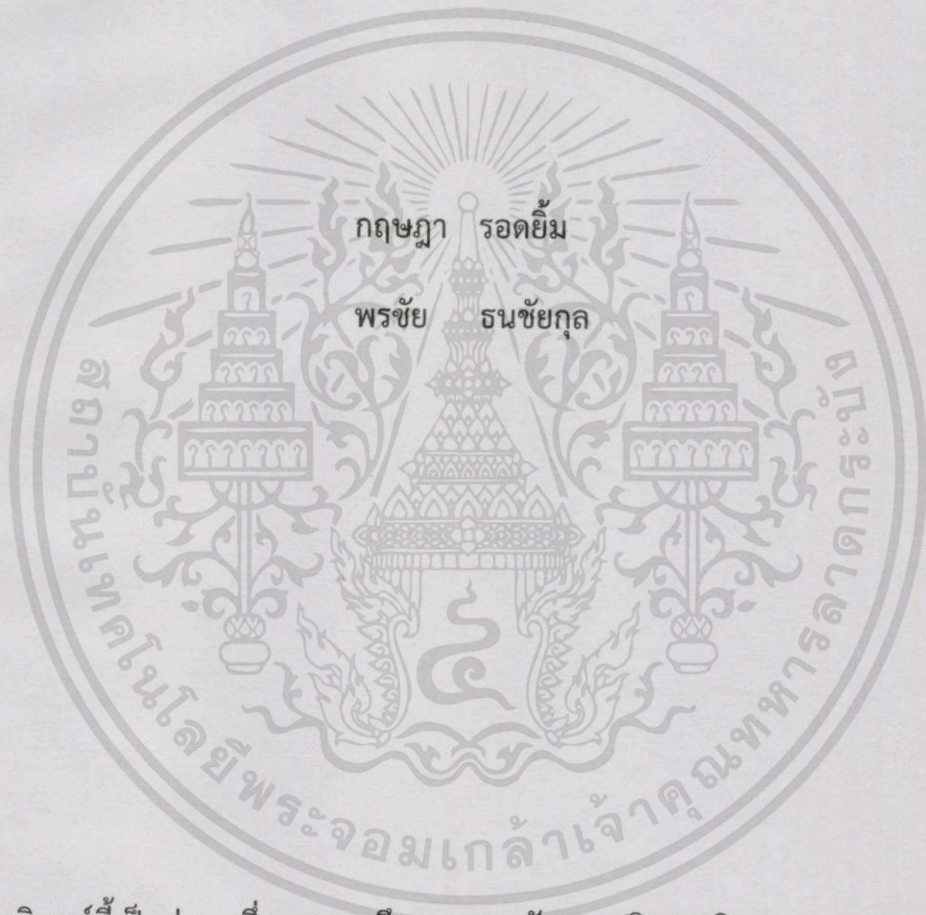


หุ่นยนต์จักรยานรักษาสมดุล
BALANCING BICYCLE ROBOT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

หุ่นยนต์จักรยานรักษาสมดุล
BALANCING BICYCLE ROBOT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BALANCING BICYCLE ROBOT



KRITSADA RODYIM

PORNCHAI TANACHAIKUN

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN CONTROL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

หุ่นยนต์จักรยานสมดุล

BALANCING BYCICLE ROBOT

ผู้จัดทำ

นายกฤษฎา

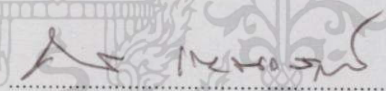
รอดยิ้ม

53010053

นายพรชัย

ธนชัยกุล

53011052



อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.ถาวร เบนจันราษฎร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์จักรยานรักษาสมดุล

โดย

นายภุชญา รอดยิ้ม 53010053

นายพรชัย ธนชัยกุล 53011052

อาจารย์ที่ปรึกษา

รศ. ดร.ถาวร เบนญจนาสุทธิ์

ปีการศึกษา 2556

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้นำเสนอการศึกษาและการออกแบบหุ่นยนต์จักรยานรักษาสมดุล โดยใช้จานหมุนเพื่อสร้างสมดุล ซึ่งหุ่นยนต์จักรยานรักษาสมดุลประกอบด้วย หุ่นยนต์จักรยานไมโครคอนโทรลเลอร์ และวงจรที่จำเป็น จุดมุ่งหมายของโครงการนี้ คือสามารถสร้างสมดุลให้กับหุ่นยนต์จักรยานรักษาสมดุลโดยใช้ระบบควบคุมแบบป้อนกลับ

ขั้นตอนดำเนินการ เริ่มจากออกแบบและสร้างโครงสร้างของหุ่นยนต์จักรยานรักษาสมดุล แล้วออกแบบและสร้างวงจรที่จำเป็น หลังจากนั้นได้เขียนโปรแกรมคอมพิวเตอร์ โดยใช้โปรแกรม Arduino และ LabVIEW จากการทดลองพบว่าการทำงานของหุ่นยนต์จักรยานรักษาสมดุลทำงานตามคำสั่งโปรแกรม อย่างไรก็ตามหุ่นยนต์จักรยานรักษาสมดุลยังไม่สามารถรักษาสมดุลได้ เมื่อใช้ตัวควบคุมแบบพีและพีดี

BALANCING BICYCLE ROBOT

By

Mr.Kritsada Rodyim 53010053

Mr.Pornchai Tanachaikun 53011052

Advisor

Assoc. Prof. Dr. Taworn Benjanarasuth

Academic Year 2013

ABSTRACT

This thesis presents theories and designs of a bicycle robot by using a flying wheel for balancing. The balancing bicycle robot consists of its body, a microcontroller and necessary circuit. The purpose of this project is to balance the balancing bicycle robot by feedback control system.

The project is conducted in the following steps. First, the structure of balancing bicycle robot is designed and constructed. Second, the necessary circuit is designed and implemented. Then, the computer programs are written in Arduino and LabVIEW. From the experimental results, the balancing bicycle robot operation can work accordingly to the program. However, the balancing bicycle robot still cannot balance itself by using P and PD controllers.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ที่ปรึกษารศ.ดร.ถาวร เบญจนราสุทธิ เป็นอย่างยิ่ง ที่คอยให้คำปรึกษาแนะนำวิธีการแก้ปัญหาต่าง ๆ ในโครงการทั้งทางทฤษฎี และทางปฏิบัติแก่คณะผู้จัดทำมาโดยตลอด ทำให้ผู้จัดทำเข้าใจถึงที่มาของปัญหา และสามารถแก้ไขปัญหาดังกล่าว และคอยถามความคืบหน้าของโครงการอยู่เสมอ เพื่อให้คณะผู้จัดทำมีความกระตือรือร้นต่อการทำโครงการ รวมทั้งเอื้อเพื่ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่น ๆ ที่เป็นประโยชน์ต่อโครงการ

ขอขอบพระคุณคณะอาจารย์สาขาวิชาวิศวกรรมระบบควบคุมที่คอยให้คำปรึกษา และคอยช่วยเหลือในส่วนที่คณะผู้จัดทำยังไม่มีความรู้เข้าใจเป็นอย่างดี

ขอขอบคุณสาขาวิชาวิศวกรรมระบบควบคุม ที่ได้เอื้อเพื่ออุปกรณ์ และสถานที่ในการทำวิจัยจนผู้จัดทำสามารถทำปริญญานิพนธ์ฉบับนี้จนเสร็จสิ้น

ขอขอบคุณเพื่อน ๆ ทุกคนที่คอยช่วยเหลือคณะผู้จัดทำมาโดยเฉพาะอย่างยิ่ง นายวิจิตร มินทรานทรวานันท์ โดยการสอน และแนะนำการใช้งานของโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับโครงการของคณะผู้จัดทำเอง

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยสนับสนุนการเงินสำหรับการศึกษา และคอยให้กำลังใจมาตลอด

คณะผู้จัดทำ

นายกฤษฎา รอดยิ้ม

นายพรชัย ธนชัยกุล

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์	1
1.3 ขั้นตอนการดำเนินงาน	1
1.4 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและความรู้พื้นฐาน	3
2.1 กลศาสตร์หุ่นยนต์จักรยานรักษาสมดุล	3
2.2 ระบบควบคุมแบบป้อนกลับ	7
2.2.1 ตัวบ่งชี้สมรรถนะของระบบ	7
2.2.2 ตัวควบคุมแบบพีไอดี	8
2.3 การควบคุมมอเตอร์ไฟฟ้า	11
2.3.1 การควบคุมทิศทางของมอเตอร์ไฟฟ้า	11
2.3.2 การควบคุมความเร็วของมอเตอร์ไฟฟ้า	12
2.4 หลักการทำงานของเซนเซอร์วัดความเร่ง	13
2.5 หลักการทำงานของไมโครคอนโทรลเลอร์	14
บทที่ 3 การออกแบบและสร้างชิ้นงาน	16
3.1 การออกแบบโครงสร้าง	16
3.1.1 โครงสร้างของหุ่นยนต์จักรยาน	16
3.1.2 งานหมุน	21
3.1.3 แป้นยึดงานหมุน	21
3.1.4 ล้อ	22
3.1.5 มอเตอร์ไฟฟ้า	22
3.1.6 แบตเตอรี่	23

สารบัญ (ต่อ)

	หน้า
3.1.7 ตัวเพิ่มน้ำหนักงานหมุน	24
3.1.8 เหล็กฉากรับน้ำหนัก	24
3.1.9 แผงวงจร	24
3.1.10 โครงสร้างหุ่นยนต์จักรยานเมื่อประกอบชิ้นส่วนเข้าด้วยกัน	25
3.2 การออกแบบวงจรไฟฟ้า	26
3.2.1 การออกแบบวงจรขับเคลื่อนมอเตอร์	26
3.2.2 วงจรไมโครคอนโทรลเลอร์และเซนเซอร์	26
3.2.3 การออกแบบวงจรประกอบโมดูล MPU6050	27
3.3 การออกแบบโปรแกรม	28
3.3.1 โปรแกรมป้อนเปอร์เซ็นต์สัญญาณพัลส์วิธึมอดูเลชัน	28
3.3.2 โปรแกรมอ่านเซนเซอร์วัดความเร็ว	30
3.3.3 โปรแกรมการควบคุมแบบพีไอดี	32
บทที่ 4 การทดลองและผลการทดลอง	36
4.1 การทดลองวัดค่ามุมพิทช์ก่อนและหลังการหาค่ามัธยฐาน	36
4.2 การทดลองวัดค่ามุมพิทช์เมื่อเอียงหุ่นยนต์จักรยานรักษาสมดุล	37
4.3 การทดลองขับเคลื่อนมอเตอร์เมื่อแปรค่าเปอร์เซ็นต์สัญญาณพัลส์วิธึมอดูเลชัน	40
4.4 การทดลองควบคุมหุ่นยนต์จักรยานรักษาสมดุล	41
4.4.1 การปรับค่าอัตราขยายเชิงสัดส่วน	41
4.4.2 การปรับค่าอัตราขยายเชิงอนุพันธ์	49
บทที่ 5 สรุปและวิเคราะห์ปัญหา	50
5.1 สรุปผลการดำเนินงาน	50
5.2 ปัญหาและอุปสรรค	50
5.3 แนวทางการพัฒนา	51
เอกสารอ้างอิง	52
ภาคผนวก	53
ภาคผนวก ก เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	54
ก.1 แผงฝังบอร์ด Arduino Uno R3	54
ก.2 เอกสารคู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmega328	55
ก.3 เอกสารคู่มือการใช้งาน L298N	59

สารบัญ (ต่อ)

	หน้า
ก.4 เอกสารคู่มือการใช้งาน MPU6050	61
ภาคผนวก ข แผงผังอุปกรณ์และการเชื่อมต่อวงจร	64
ข.1 แผงผังวงจรของหุ่นยนต์จักรยานรักษาสมดุล	64
ข.2 ลายวงจรสำหรับแผ่นวงจรพิมพ์	65
ภาคผนวก ค แบบโครงสร้างชิ้นส่วน	66
ค.1 ฐานกลาง	66
ค.2 แผ่นยึดมอเตอร์กับล้อหน้า	67
ค.3 งานหมุน	67
ค.4 แผ่นยึดมอเตอร์	68
ค.5 ก้านยึดล้อหลัง	68
ค.6 แผ่นรับน้ำหนักมอเตอร์	69
ค.7 ตัวล้อคแบบที่ 1	69
ค.8 ตัวล้อคแบบที่ 2	70
ค.9 ตัวล้อคแบบที่ 3	70
ภาคผนวก จ โปรแกรมควบคุมหุ่นยนต์จักรยานรักษาสมดุล	71

สารบัญรูป

รูปที่	หน้า
2.1 การวิเคราะห์หุ่นยนต์จักรยานรักษาสมดุลเมื่อมองจากด้านหน้า	5
2.2 การระบุสมรรถนะของระบบต่อสัญญาณขึ้นบันไดหนึ่งหน่วย	8
2.3 ระบบควบคุมแบบพีไอดี	9
2.4 วงจรเอชบริดจ์	11
2.5 ลักษณะการทำงานภายในของไอซี L298N	12
2.6 ลักษณะการปรับค่าดิฟฟ์ไจเคิ้ล	13
2.7 ทิศการวัดความเร่งของเซนเซอร์	13
2.8 โมดูล MPU6050	14
2.9 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	14
3.1 ภาพวาด 3 มิติ ของโครงสร้าง	16
3.2 ฐานกลาง	17
3.3 ก้านยึดล้อหลัง	17
3.4 แผ่นยึดมอเตอร์กับจานหมุน	18
3.5 แผ่นยึดมอเตอร์กับล้อหน้า	18
3.6 แผ่นรับน้ำหนักมอเตอร์	19
3.7 ตัวล็อกแบบที่ 1	19
3.8 ตัวล็อกแบบที่ 2	20
3.9 ตัวล็อกแบบที่ 3	20
3.10 จานหมุน	21
3.11 แผ่นยึดจานหมุน	21
3.12 ล้อ	22
3.13 มอเตอร์ไฟฟ้าความเร็วรอบสูง	22
3.14 มอเตอร์ไฟฟ้าแบบธรรมดา	23
3.15 แบตเตอรี่ลิเทียมโพลิเมอร์	23
3.16 ตัวเพิ่มน้ำหนักจานหมุน	24
3.17 เหล็กฉากรับน้ำหนัก	24
3.18 แผงวงจร	25
3.19 หุ่นยนต์จักรยานเมื่อประกอบชิ้นส่วนเข้าด้วยกัน	25
3.20 การเชื่อมต่อวงจรต่าง ๆ กับบอร์ด Arduino Uno R3	26
3.21 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I2C BUS	27

สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.22	บล็อกโปรแกรมบ็อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชัน	28
3.23	การทำงานของโปรแกรมบ็อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชัน	29
3.24	หน้าต่างโปรแกรมบ็อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชัน	30
3.25	บล็อกโปรแกรมอ่านเซนเซอร์วัดความเร่ง	30
3.26	การทำงานของโปรแกรมอ่านเซนเซอร์วัดความเร่ง	31
3.27	หน้าต่างโปรแกรมอ่านเซนเซอร์วัดความเร่ง	32
3.28	การทำงานของโปรแกรมควบคุมแบบพีไอดี	33
3.29	บล็อกโปรแกรมการควบคุมแบบพีไอดี	34
3.30	หน้าต่างโปรแกรมการควบคุมแบบพีไอดี	35
4.1	การจับหุ่นยนต์จักรยานรักษาสมดุลที่ตำแหน่ง 0 องศา	36
4.2	การเปรียบเทียบการอ่านค่ามุมพิทช์	37
4.3	การทดลองวัดค่ามุมพิทช์	37
4.4	หน้าต่างการอ่านค่ามุมพิทช์	38
4.5	การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 5 องศา	41
4.6	หน้าต่างโปรแกรมที่มุม 5 องศา	42
4.7	การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 10 องศา	42
4.8	หน้าต่างโปรแกรมที่มุม 10 องศา	43
4.9	การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 15 องศา	43
4.10	หน้าต่างโปรแกรมที่มุม 15 องศา	44
4.11	การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -5 องศา	44
4.12	หน้าต่างโปรแกรมที่มุม -5 องศา	45
4.13	การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -10 องศา	45
4.14	หน้าต่างโปรแกรมที่มุม -10 องศา	46
4.15	การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -15 องศา	46
4.16	หน้าต่างโปรแกรมที่มุม -15 องศา	47
4.17	การทดลองปล่อยหุ่นยนต์จักรยานรักษาสมดุลสร้างสมดุลเอง	48
4.18	หน้าต่างโปรแกรมเมื่อปล่อยหุ่นยนต์จักรยานรักษาสมดุลอาศัยตัวควบคุมแบบพี	48
4.19	หน้าต่างโปรแกรมเมื่อปล่อยหุ่นยนต์จักรยานรักษาสมดุลอาศัยตัวควบคุมแบบพีดี	49

สารบัญตาราง

ตารางที่	หน้า
2.1 ผลของการปรับเพิ่มอัตราขยายต่าง ๆ ของตัวควบคุมแบบพีไอตัวอย่างอิสระ	10
2.2 ข้อมูลบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	15
4.1 ข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์ครั้งที่ 1	38
4.2 ข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์ครั้งที่ 2	39
4.3 ข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์ครั้งที่ 3	39
4.4 ค่าเฉลี่ยข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์	39
4.5 การแปรค่าเปอร์เซ็นต์สัญญาณพัลส์วริธมอดูเลชันเทียบกับแรงดันเอาต์พุต	40



บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

การเริ่มหัดขี่จักรยานของทุกคนจะขี่ได้ในระยะเวลาสั้น ๆ เท่านั้น เนื่องจากผู้ขี่ยังไม่มีทักษะในการสร้างสมดุลตัวเองกับจักรยาน พอเริ่มขี่จักรยานไประยะเวลาหนึ่ง ผู้ขี่จะมีทักษะในการสร้างสมดุลระหว่างผู้ขี่กับจักรยานได้อย่างอัตโนมัติ โดยผู้ขี่นั้นไม่ทราบว่า การสร้างสมดุลนั้นมาจากตัวผู้ขี่เอง ซึ่งการสร้างสมดุลนั้นมาจากการบังคับคันเลี้ยว หรือการใช้น้ำหนักตัวผู้ขี่ทรงตัว

ทางผู้จัดทำสนใจการจัดทำหุ่นยนต์จักรยาน โดยทางผู้จัดทำคิดว่าการสร้างหุ่นยนต์จักรยานรักษาสมดุล โดยอาศัยงานหมุนในการสร้างสมดุลเป็นเรื่องน่าสนใจต่อการศึกษา และค้นคว้าหาข้อมูล ซึ่งแนวคิดนี้ นำมาจากการควบคุมสมดุลของหุ่นยนต์จักรยานด้วยระบบควบคุมพีไอดี (Balancing control of bicycle robot using PID control) [1] ซึ่งตัวหุ่นยนต์จักรยานมีการใช้งานหมุนในการสร้างสมดุล ซึ่งหุ่นยนต์จักรยานสามารถรักษาสมดุลได้ด้วยระบบควบคุมแบบพีไอดี ผู้จัดทำจึงเลือกศึกษาและจัดทำหุ่นยนต์จักรยานรักษาสมดุล โดยศึกษาแบบจำลองของการควบคุมสมดุลของหุ่นยนต์จักรยานด้วยระบบควบคุมพีไอดี [1] โดยผู้จัดทำหวังอย่างยิ่งว่าอนาคตความรู้จากการศึกษา และจัดทำโครงการจะสามารถนำไปประยุกต์ใช้กับจักรยานที่ใช้งานอยู่ทั่วไปได้ โดยผู้ขี่ไม่ต้องสร้างสมดุลกับจักรยานเอง แต่ใช้หลักการสร้างสมดุลโดยใช้งานหมุน หรือการบังคับคันเลี้ยวแทน และเพื่อเป็นประโยชน์แก่ผู้ที่อยากศึกษาเกี่ยวกับการสร้างสมดุลของหุ่นยนต์จักรยานเองด้วย

1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาและควบคุมหุ่นยนต์จักรยานรักษาสมดุล
- 1.2.2 เพื่อสร้างต้นแบบหุ่นยนต์จักรยานรักษาสมดุล
- 1.2.3 เพื่อศึกษาการใช้ไมโครคอนโทรลเลอร์ เพื่อควบคุมหุ่นยนต์จักรยานรักษาสมดุล

1.3 ขั้นตอนการดำเนินงาน

- 1.3.1 ศึกษาแนวคิดหุ่นยนต์จักรยานรักษาสมดุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.2 สร้างหุ่นยนต์จักรยาน

1.3.3 ศึกษาการใช้งานบอร์ดไมโครคอนโทรลเลอร์ และวงจรอิเล็กทรอนิกส์ ในการเชื่อมต่อกับโปรแกรม LabVIEW

1.3.4 ออกแบบระบบควบคุม โดยใช้ไมโครคอนโทรลเลอร์ในการรับค่าตัวแปรต่าง ๆ

1.3.5 ทดลองและสรุปผลการทำงาน

1.4 รายละเอียดของปฏิญานิพนธ์

ปฏิญานิพนธ์ฉบับนี้ แบ่งออกเป็น 5 บท ซึ่งครอบคลุมเนื้อหา ดังนี้

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ ขั้นตอนการศึกษา และรายละเอียดโดยย่อของปฏิญานิพนธ์แต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง ในบทนี้จะกล่าวถึง กลศาสตร์หุ่นยนต์จักรยานรักษาสมดุล ระบบควบคุมแบบป้อนกลับ การควบคุมมอเตอร์ไฟฟ้า หลักการทำงานของเซนเซอร์วัดความเร่ง และหลักการทำงานของไมโครคอนโทรลเลอร์

บทที่ 3 การออกแบบโครงสร้าง การออกแบบวงจรไฟฟ้า และการออกแบบโปรแกรมควบคุม

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 สรุปผลการทำงาน ปัญหาและอุปสรรค และแนวทางการพัฒนา

บทที่ 2

ทฤษฎีและความรู้พื้นฐาน

ก่อนจัดทำโครงการ ขั้นตอนแรกในการเริ่มการทำงานคือศึกษาทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับโครงการ เพื่อที่สามารถจัดทำโครงการได้อย่างถูกต้องและมีประสิทธิภาพ ดังนั้นการทำโครงการหุ่นยนต์จักรยานรักษาสมดุล จึงได้ศึกษาทฤษฎี และความรู้ที่เกี่ยวข้องดังต่อไปนี้

2.1 กลศาสตร์หุ่นยนต์จักรยานรักษาสมดุล

กลศาสตร์ของหุ่นยนต์จักรยานรักษาสมดุลมีการควบคุมทั้งแบบการบังคับคันเลี้ยว (Steering) และการควบคุมความเร็วจานหมุน (Flying wheel) ซึ่งการศึกษาจะใช้การควบคุมความเร็วจานหมุน โดยกลศาสตร์ของหุ่นยนต์จักรยานรักษาสมดุลจะใช้กลศาสตร์แบบลากรางจ์มาพิจารณา

กลศาสตร์แบบลากรางจ์ (Lagrangian mechanics) เป็นกลศาสตร์ที่อยู่ภายในขอบเขตของกลศาสตร์ดั้งเดิม (Classical mechanics) ซึ่งเสนอโดย โจเซฟ หลุยส์ ลากรางจ์ (Joseph Louis Lagrange) เป็นแนวทางในการวิเคราะห์พลวัตของระบบอีกแนวทางหนึ่ง นอกเหนือจากกลศาสตร์แบบนิวตัน (Newton mechanics)

ลากรางเจียนนิยามโดยผลต่างระหว่างพลังงานจลน์และพลังงานศักย์ภายในระบบ ดังสมการ (2.1)

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q) \quad (2.1)$$

โดย

L คือ ลากรางเจียน (Lagrangian)

T คือ พลังงานจลน์ของระบบ

V คือ พลังงานศักย์ของระบบ

q_i คือ พิกัดทั่วไป (Generalized coordinate) ที่ $i = 1, \dots, N$

N คือ องศาอิสระ (Degree of freedom) ของระบบ

พลังงานจลน์ คือพลังงานที่มีอยู่ในวัตถุที่กำลังเคลื่อนที่ จึงกล่าวได้ว่า "วัตถุที่กำลังเคลื่อนที่ ล้วนมีพลังงานจลน์ทั้งสิ้น ปริมาณพลังงานจลน์ในวัตถุจะมีมากหรือน้อยขึ้นอยู่กับมวล และความเร็วของวัตถุนั้น" อธิบายดังสมการ (2.2)

$$T = \frac{1}{2}mv^2 \quad (2.2)$$

โดย

m คือ มวลของวัตถุ

v คือ อัตราเร็วของวัตถุ

พลังงานศักย์ คือพลังงานกลในตัววัตถุที่สัมพันธ์กับตำแหน่งที่อยู่ของวัตถุ หรือพลังงานในวัตถุที่มีแรงต้านทาน เมื่อมันถูกแรงภายนอกกระทำให้ขนาด ปริมาตร หรือตำแหน่งของวัตถุไม่เป็นไปอย่างอิสระ ซึ่งหุ่นยนต์จักรยานรักษาสมดุลเกี่ยวข้องกับพลังงานศักย์โน้มถ่วง อธิบายดังสมการ (2.3)

$$V = mgh \quad (2.3)$$

โดย

g คือ ความเร่งโดยความโน้มถ่วง

h คือ ความสูงจากระดับอ้างอิง

กรณีระบบไม่มีแรงกระทำจากภายนอก และไม่มีการสูญเสียพลังงานในระบบ หรือเป็นระบบอนุรักษ์พลังงาน [4] มีความสัมพันธ์ตามสมการออยเลอร์-ลากรางจ์ (Euler-Lagrange equation) สามารถเขียนใหม่ ดังสมการ (2.4)

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0 \quad (2.4)$$

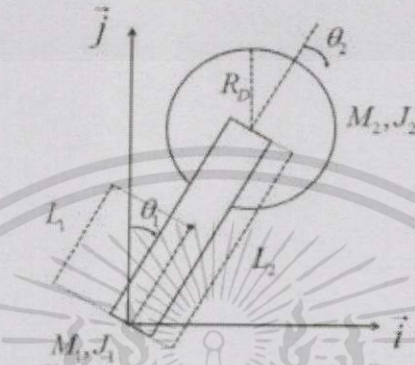
กรณีระบบมีแรงกระทำจากภายนอก และมีการสูญเสียพลังงานในระบบ หรือเป็นระบบไม่อนุรักษ์พลังงาน [4] มีความสัมพันธ์ตามสมการออยเลอร์-ลากรางจ์ สามารถเขียนใหม่ ดังสมการ (2.5)

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i \quad (2.5)$$

โดย

Q_i คือ แรงทั่วไป (Generalized force) สอดคล้องพิกัดทั่วไปที่ i

การประยุกต์ใช้สมการออยเลอร์-ลากรางจ์ในหุ่นยนต์จักรยานรักษาสมดุล จะพิจารณาโดยมองไปด้านหน้าของโครงสร้างหุ่นยนต์จักรยานรักษาสมดุลดังรูปที่ 2.1



รูปที่ 2.1 การวิเคราะห์หุ่นยนต์จักรยานรักษาสมดุลเมื่อมองจากหน้า [1]

โดย

L_1 คือ ระยะจากฐานตัวรถถึงจุดศูนย์กลางตัวรถ

L_2 คือ ระยะทางจากฐานตัวรถถึงจานหมุน

R_D คือ รัศมีของจานหมุน

θ_1 คือ มุมพิทซ์ หรือมุมการหมุนของตัวรถ

θ_2 คือ มุมการหมุนของจานหมุนเทียบกับตัวรถ

M_1 คือ น้ำหนักของตัวรถ

M_2 คือ น้ำหนักของจานหมุน

J_1 คือ โมเมนต์ความเฉื่อยการหมุนของตัวรถ

J_2 คือ โมเมนต์ความเฉื่อยการหมุนของจานหมุน

การพิจารณาสมการการเคลื่อนที่ของหุ่นยนต์จักรยานรักษาสมดุลนั้นใช้กลศาสตร์แบบลากรางจ์ ซึ่งสมการลากรางจ์ทั้งหมดของระบบประกอบไปด้วยพลังงานจลน์ และพลังงานศักย์ การหาพลังงานจลน์ และพลังงานศักย์ของหุ่นยนต์จักรยานรักษาสมดุลจะต้องกำหนดเวกเตอร์ตำแหน่ง

$$\vec{r}_1 = L_1 \sin\theta_1 \vec{i} + L_1 \cos\theta_1 \vec{j} \quad (2.6)$$

$$\vec{r}_2 = L_2 \sin\theta_2 \vec{i} + L_2 \cos\theta_2 \vec{j} \quad (2.7)$$

โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\vec{r}_1 คือ เวกเตอร์ตำแหน่งของตัวรถ

\vec{r}_2 คือ เวกเตอร์ตำแหน่งของจานหมุน

อัตราเร็วของหุ่นยนต์จักรยานรักษาสมดุลเกิดจากการอนุพันธ์เวกเตอร์ตำแหน่งจากสมการ (2.6) และสมการ (2.7) ดังนั้นจึงได้พลังงานจลน์ทั้งหมดตามสมการ (2.8)

$$T = \frac{1}{2}M_1(\vec{v}_1 \cdot \vec{v}_1) + \frac{1}{2}M_2(\vec{v}_2 \cdot \vec{v}_2) + \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}J_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (2.8)$$

และพลังงานศักย์ของระบบจะได้สมการ (2.9)

$$V = M_1gL_1\cos\theta_1 + M_2gL_2\cos\theta_1 \quad (2.9)$$

ดังนั้นจะได้ลากรางเจียน ดังสมการ (2.10)

$$\begin{aligned} L &= T - V \\ &= \frac{1}{2}M_1(\vec{v}_1 \cdot \vec{v}_1) + \frac{1}{2}M_2(\vec{v}_2 \cdot \vec{v}_2) + \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}J_2(\dot{\theta}_1 + \dot{\theta}_2)^2 - \\ &\quad M_1gL_1\cos\theta_1 - M_2gL_2\cos\theta_1 \end{aligned} \quad (2.10)$$

เมื่อได้สมการ (2.10) อาศัยสมการ (2.5) จะได้สมการลากรางจ์ (2.11) เมื่ออินทิเกรตคือค่า τ_q และพิกัดทั่วไป $q = [\theta_1 \theta_2]^T$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_q \quad (2.11)$$

นำลากรางเจียนในสมการ (2.10) แทนในสมการ (2.11) จะได้สมการกลศาสตร์ของหุ่นยนต์จักรยานรักษาสมดุล ดังสมการ (2.12) และสมการ (2.13)

$$(J_1 + L_1^2M_1 + L_2^2M_2)\ddot{\theta}_1 - g(L_1M_1 + L_2M_2)\sin\theta_1 = -\tau \quad (2.12)$$

$$J_2(\ddot{\theta}_1 + \ddot{\theta}_2) = \tau \quad (2.13)$$

และในการสร้างหุ่นยนต์จักรยานรักษาสมดุลจะใช้มอเตอร์ไฟฟ้าเป็นตัวขับเคลื่อนระบบ โดยแรงบิด (Torque) τ จะขึ้นกับแรงดันไฟฟ้า ดังสมการ (2.14) และสมการ (2.15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\tau = nK_t i + f_m(\dot{\theta}_2 - \dot{\theta}_1) \quad (2.14)$$

$$Li + R_m i = d + K_b(\dot{\theta}_2 - \dot{\theta}_1) \quad (2.15)$$

โดย

K_t คือ ค่าคงที่แรงบิดของมอเตอร์

f_m คือ องค์กรประกอบแรงเสียดทานของมอเตอร์

L คือ ค่าความเหนี่ยวนำของมอเตอร์

R_m คือ ค่าความต้านทานของมอเตอร์

K_b คือ ค่าคงที่ของ Back electromotive force

d คือ แรงดันไฟฟ้า หรือค่าพัลส์วิธึมอดูเลขัน

จากสมการ (2.14) และสมการ (2.15) สามารถหาความสัมพันธ์ระหว่างแรงบิดและแรงดันไฟฟ้า ดังสมการ (2.16)

$$\tau = \frac{nK_t}{R_m} d - \left(\frac{nK_t}{R_m} K_b + f_m \right) (\dot{\theta}_2 - \dot{\theta}_1) \quad (2.16)$$

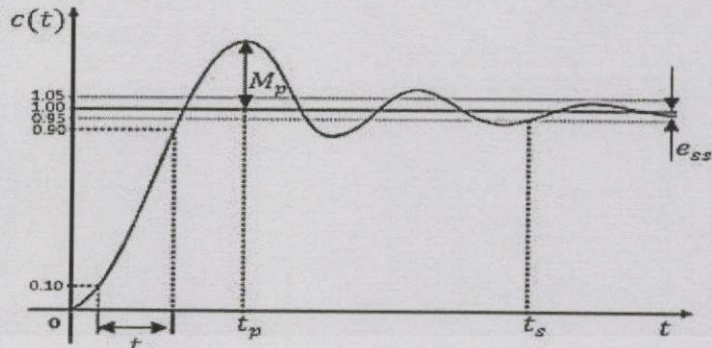
ดังนั้นพลวัตรวมของหุ่นยนต์จักรยานรักษาสมดุลสามารถอธิบายได้โดย สมการ (2.12) สมการ (2.13) และสมการ (2.16)

2.2 ระบบควบคุมแบบป้อนกลับ

ระบบควบคุมแบบป้อนกลับ (Feedback control system) [2] เป็นระบบควบคุมที่ใช้ค่าเอาต์พุตที่ได้จากการวัด แล้วทำการป้อนกลับไปเปรียบเทียบกับค่าอินพุตอ้างอิง เพื่อให้ระบบมีเสถียรภาพ และสมรรถนะตามความต้องการ ซึ่งสามารถวิเคราะห์ได้จากผลตอบสนองของระบบ โดยสามารถทราบได้จากตัวบ่งชี้สมรรถนะของระบบ

2.2.1 ตัวบ่งชี้สมรรถนะของระบบ

การระบุสมรรถนะ (Performance) [2] ของระบบนั้นนิยมพิจารณาจากผลตอบสนองต่อสัญญาณขั้นบันไดหนึ่งหน่วย (Unit step response) โดยทำการป้อนสัญญาณดังกล่าวเป็นอินพุตให้กับระบบที่ต้องการวิเคราะห์ จากนั้นตรวจสอบสัญญาณเอาต์พุตหรือผลตอบสนองที่ได้ โดยวิเคราะห์จากตัวบ่งชี้สมรรถนะต่าง ๆ ของระบบ ดังรูปที่ 2.2



รูปที่ 2.2 การระบุสมรรถนะของระบบต่อสัญญาณขั้นบันไดหนึ่งหน่วย [2]

จากรูปที่ 2.2 การวิเคราะห์สมรรถนะของระบบโดยอาศัยตัวบ่งชี้สมรรถนะที่นิยมพิจารณาดังนี้

ผลตอบสนองในสภาวะอยู่ตัว (Steady state response) นิยมพิจารณาค่าความผิดพลาดในสภาวะอยู่ตัว (Steady-state error) หรือ e_{ss} คือผลต่างระหว่างสัญญาณอินพุตอ้างอิงกับสัญญาณเอาต์พุตเมื่อระบบเข้าสู่สภาวะอยู่ตัว

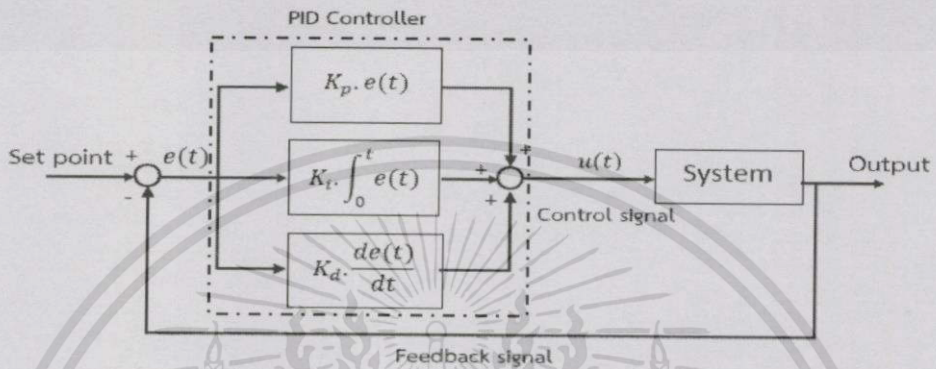
ผลตอบสนองในสภาวะชั่วคราว (Transient response) นิยมพิจารณาทัวบ่งชี้สมรรถนะดังนี้

- ช่วงเวลาขึ้น (Rise time) หรือ t_r คือช่วงเวลาตั้งแต่เอาต์พุตมีค่าเป็น 10 เปอร์เซ็นต์ จนถึง 90 เปอร์เซ็นต์ ของค่าเอาต์พุตในสภาวะอยู่ตัว
- ค่าพุ่งเกินสูงสุด (Maximum overshoot) หรือ M_p คือระดับผลตอบสนองที่มีค่าสูงสุดจากระดับค่าปลาย โดยมากนิยมระบุเป็นเปอร์เซ็นต์เทียบกับผลต่างระหว่างระดับตั้งต้น และระดับค่าปลาย
- เวลาพุ่งเกินสูงสุด (Peak time) หรือ t_p คือเวลาที่ผลตอบสนองใช้จากระดับเริ่มต้นจนถึงจุดซึ่งเกิดค่าพุ่งเกินสูงสุด
- ช่วงเวลาเข้าที่ (Settling time) หรือ t_s คือช่วงเวลาตั้งแต่เริ่มต้นจนกระทั่งการแกว่งของเอาต์พุตลดลงอยู่ในขอบเขตที่ยอมรับได้ว่าเข้าสู่สภาวะอยู่ตัว โดยปกติแล้วขอบเขตนี้จะอยู่ในช่วงไม่เกิน 5 เปอร์เซ็นต์ ของเอาต์พุตในสภาวะอยู่ตัว

2.2.2 ตัวควบคุมแบบพีไอดี

จากเป้าหมายที่ต้องการทำให้ระบบมีสมรรถนะเป็นไปตามตัวบ่งชี้สมรรถนะที่ต้องการ และมีเสถียรภาพ ดังนั้นจำเป็นต้องมีตัวควบคุมเพื่อช่วยในการควบคุมระบบ ซึ่งในโครงการนี้ได้เลือกใช้ตัวควบคุมแบบพีไอดี

ตัวควบคุมแบบพีไอดี (PID controller) [2] เป็นตัวควบคุมที่ใช้ในระบบป้อนกลับซึ่งใช้กันอย่างกว้างขวาง อธิบายแนวคิดอย่างง่ายได้ดังรูปที่ 2.3 ซึ่งอินพุตของตัวควบคุม คือค่าความผิดพลาด $e(t)$ ที่ได้มาจากความต่างค่าเป้าหมาย (Set point) [2] กับเอาต์พุตของกระบวนการ และเอาต์พุตของตัวควบคุมคือสัญญาณควบคุม (Control signal) $u(t)$



รูปที่ 2.3 ระบบควบคุมแบบพีไอดี [2]

จากรูปที่ 2.3 เห็นว่าตัวควบคุมแบบพีไอนั้น ประกอบไปด้วยการควบคุม 3 ส่วน ได้แก่ การควบคุมแบบสัดส่วน การควบคุมแบบปริพันธ์ และการควบคุมแบบอนุพันธ์ โดยมีสัญญาณควบคุม $u(t)$ ดังสมการ (2.17)

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.17)$$

โดย

K_p คือ อัตราขยายเชิงสัดส่วน (Proportional gain)

K_i คือ อัตราการขยายเชิงปริพันธ์ (Integral gain)

K_d คือ อัตราขยายเชิงอนุพันธ์ (Derivative gain)

T_i คือ ค่าเวลาปริพันธ์ (Integral time)

T_d คือ ค่าเวลาอนุพันธ์ (Derivative time)

โดยการปรับค่าพารามิเตอร์ของตัวควบคุม สามารถใช้หลักการดังนี้

การควบคุมแบบสัดส่วน (Proportional control action หรือ P action) เป็นการกำหนดให้เอาต์พุตของตัวควบคุม เป็นสัดส่วนกับค่าความผิดพลาด โดยปรับค่าอัตราขยายสัดส่วนที่คูณอยู่กับค่าความผิดพลาดที่ได้จากการนำเอาค่าเป้าหมายลบด้วยค่าที่วัดได้ ซึ่งในทางปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมแบบสัดส่วนจะเข้าใกล้ค่าหนึ่งที่ไม่ใช่ค่าเป้าหมายจริง ซึ่งค่าความผิดพลาดจากค่าเป้าหมายนิยมเรียกว่า ค่าออฟเซต (Offset)

การควบคุมแบบปริพันธ์ (Integral control action หรือ I action) ใช้ในการแก้ปัญหาออฟเซตของระบบ โดยการควบคุมแบบปริพันธ์จะเข้าไปช่วยกำจัดค่าออฟเซตที่ยังมีอยู่ ให้เอาต์พุตของระบบเข้าสู่ค่าเป้าหมายได้ โดยค่าเอาต์พุตที่ออกมาจะขึ้นอยู่กับค่าเวลาปริพันธ์ ที่กำหนดขึ้นมาตั้งแต่ต้น หากกำหนดให้ค่าเวลาปริพันธ์น้อย เอาต์พุตของระบบจะเข้าสู่เป้าหมายได้อย่างรวดเร็วแต่จะเกิดการกระเพื่อม (Hunting) ของสัญญาณมาก และหากกำหนดให้ค่าปริพันธ์มากจะเกิดการกระเพื่อมของสัญญาณน้อย แต่ใช้เวลานานกว่าที่เอาต์พุตของระบบจะเข้าสู่ค่าเป้าหมาย

การควบคุมแบบอนุพันธ์ (Derivative control action หรือ D action) ในกรณีที่มีการรบกวนจากระบบภายนอก (Disturbance) เป็นผลให้การประมวลผลของระบบมีการเปลี่ยนแปลงอย่างทันทีทันใดควรใช้การควบคุมแบบอนุพันธ์ ซึ่งจะมีการตอบสนองที่รวดเร็ว เป็นผลให้เอาต์พุตของระบบเข้าสู่ค่าเป้าหมายได้รวดเร็วขึ้น

โดยสรุปผลของการปรับเพิ่มอัตราขยายต่าง ๆ ของตัวควบคุมแบบพีไอดีอย่างอิสระจะส่งผลกระทบต่อสมรรถนะตามตารางที่ 2.1

ตารางที่ 2.1 ผลของการปรับเพิ่มอัตราขยายต่าง ๆ ของตัวควบคุมแบบพีไอดีอย่างอิสระ

ตัวแปร	ช่วงเวลาขึ้น	ค่าพุงเกิน สูงสุด	เวลาสู่สมดุล	ความผิดพลาด ในสภาวะคง ตัว	เสถียรภาพ
K_p	ลด	เพิ่ม	เปลี่ยนแปลง เล็กน้อย	ลด	ลด
K_i	ลด	เพิ่ม	เพิ่ม	ลดลงอย่างมี นัยสำคัญ	ลด
K_d	ลดลงเล็กน้อย	ลดลงเล็กน้อย	ลดลงเล็กน้อย	-	ดีขึ้นถ้า K_d มีค่าน้อย

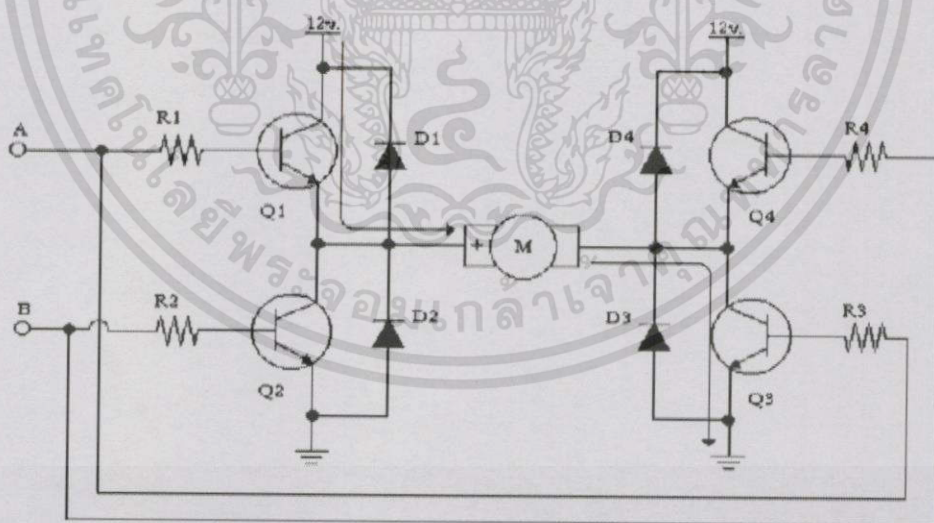
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การควบคุมมอเตอร์ไฟฟ้า

การควบคุมหุ่นยนต์จักรยานได้ทำการใช้มอเตอร์ไฟฟ้ากระแสตรงในการขับเคลื่อนตัวหุ่นยนต์ เพื่อรักษาสภาพสมดุลไว้ ซึ่งการควบคุมมอเตอร์ไฟฟ้ากระแสตรงมีการควบคุมความเร็ว และทิศทางการหมุนของมอเตอร์ มอเตอร์กระแสตรงเป็นเครื่องกลทางไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกลผ่านทางแกนหมุน หรือเพลามอเตอร์สามารถที่จะหมุนได้ เนื่องจากมีสนามแม่เหล็ก 2 แหล่งกระทำต่อกัน โดยที่สนามแม่เหล็กทั้ง 2 แหล่งอาจจะเป็นแบบที่ได้จากการจ่ายกระแสไฟฟ้าผ่านขดลวดสเตเตอร์ (Stator winding) และใช้การผ่านกระแสไฟฟ้าเข้าไปที่ขดลวดอาร์เมเจอร์ เนื่องจากลดความสูญเสียได้เพราะไม่มีขดลวดสนามแม่เหล็ก (Field winding) ทำให้มีประสิทธิภาพที่ดีขึ้น นอกจากนี้มอเตอร์กระแสตรงยังมีขนาดเล็ก และราคาถูก ซึ่งสนามแม่เหล็กถาวรเกิดจากการจ่ายกระแสไฟฟ้ากระแสตรงเข้าไปในขดลวดอาร์เมเจอร์ จะทำให้เกิดแรงบิดขึ้นที่โรเตอร์ ซึ่งจะทำให้เกิดการหมุนได้นั่นเอง

2.3.1 การควบคุมทิศทางของมอเตอร์ไฟฟ้า

การควบคุมทิศทางการหมุนของมอเตอร์กระแสตรงสามารถทำได้ โดยการควบคุมทิศทางการไหลของกระแสไฟฟ้าที่ไหลผ่านขดลวดอาร์เมเจอร์ ซึ่งสามารถทำได้โดยการกลับขั้วไฟฟ้าที่ป้อนให้กับขั้วมอเตอร์ โดยการควบคุมการไหลของกระแสไฟฟ้านั้น ส่วนใหญ่แล้วจะใช้วิธีการต่อวงจรอิเล็กทรอนิกส์ที่เรียกว่า วงจรเอชบริดจ์ (H-bridge) เข้ากับมอเตอร์ดังรูปที่ 2.4



รูปที่ 2.4 วงจรเอชบริดจ์

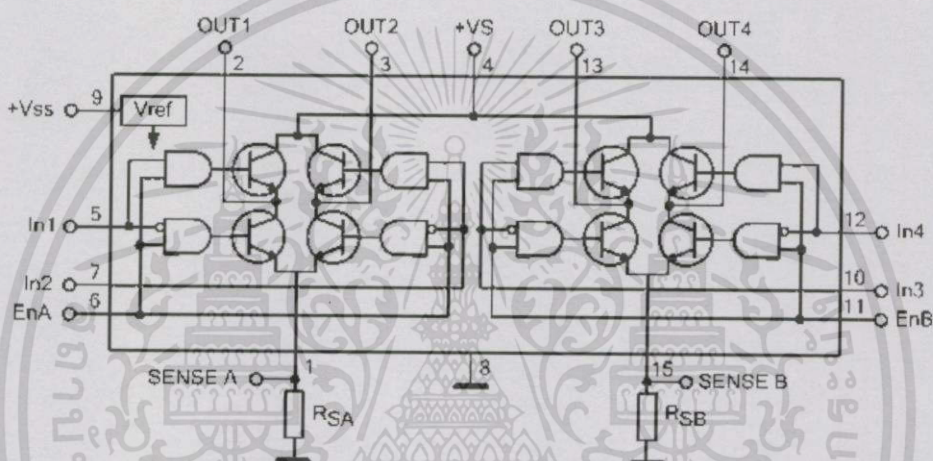
การทำงานของวงจรเอชบริดจ์ คือเมื่อจ่ายไฟที่จุด A จะทำให้ทรานซิสเตอร์ Q1 และ Q3 ทำงานส่งผลให้มีกระแสไหลจากแหล่งจ่ายผ่านขาคอลเล็กเตอร์ (Collector) และอีมิเตอร์ (Emitter) ของ Q1 ผ่านเข้าสู่ขั้วบวก (+) ของมอเตอร์ ผ่านไปยังขาคอลเล็กเตอร์ และอีมิเตอร์ของ Q3 ทำให้มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระแสไหลผ่านมอเตอร์ในทิศทางบวก และครบวงจร จึงทำให้มอเตอร์สามารถหมุนในทิศทางตามเข็มนาฬิกา

ในขณะเดียวกันหากจ่ายไฟที่จุด B จะทำให้ทรานซิสเตอร์ Q2 และ Q4 ทำงานส่งผลให้มีกระแสไหล จากแหล่งจ่ายผ่านชาคอลลีคเตอร์ และอิมิตเตอร์ของ Q4 ผ่านเข้าสู่ขั้วลบ (-) ของมอเตอร์ ผ่านไปยังชาคอลลีคเตอร์ และอิมิตเตอร์ของ Q2 ทำให้มีกระแสไหลผ่านมอเตอร์ในทิศทางบวก และครบวงจร จึงทำให้มอเตอร์สามารถหมุนในทิศทางทวนเข็มนาฬิกาได้

สำหรับโครงการนี้เลือกวงจรขับมอเตอร์เป็นไอซีขับมอเตอร์สำเร็จรูปแบบเอชบริดจ์เบอร์ L298N ซึ่งมีโครงสร้างดังรูปที่ 2.5

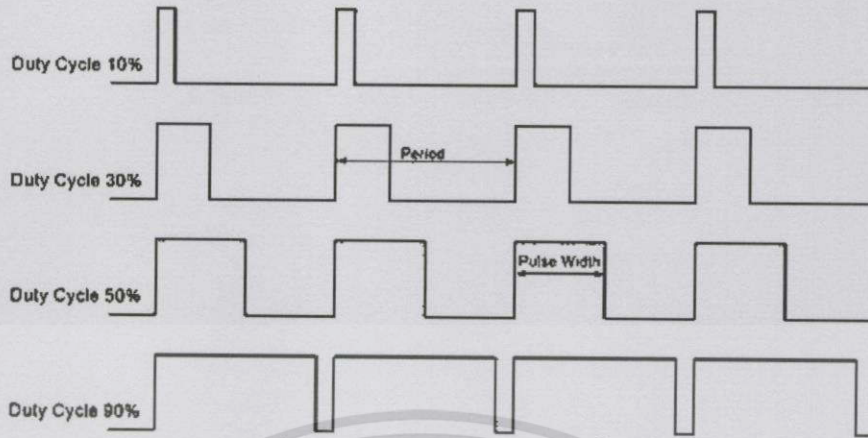


รูปที่ 2.5 ลักษณะการทำงานภายในของไอซี L298N

2.3.2 การควบคุมความเร็วของมอเตอร์ไฟฟ้า

เนื่องจากส่วนของขดลวดสเตเตอร์เป็นแม่เหล็กถาวร การควบคุมความเร็วจึงทำได้โดยการเปลี่ยนค่าแรงดันที่ขดลวดอาเมเจอร์ ซึ่งค่าแรงดันนี้จะแปรผันตรงกับความเร็วในการหมุนของมอเตอร์ โดยวิธีที่จะเปลี่ยนระดับแรงดัน จะใช้คลื่นรูปสี่เหลี่ยมที่สามารถเปลี่ยนแปลงช่วงเวลาในการเปิด และปิดแหล่งจ่ายได้จากไมโครคอนโทรลเลอร์ โดยจะเรียกสัญญาณนี้ว่า สัญญาณพัลส์วิธมอดูเลชัน (Pulse Width Modulation) ซึ่งในโครงการนี้ได้ใช้สัญญาณพัลส์วิธมอดูเลชันจากไมโครคอนโทรลเลอร์เป็นตัวบ่อนค่าให้กับขา ENABLE ของไอซี L298N โดยการปรับความกว้างของพัลส์ หรือที่เรียกว่า ค่าดิวตี้ไซเคิล (Duty cycle) ดังรูปที่ 2.6

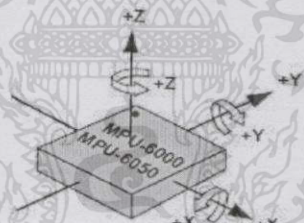
เมื่อความกว้างของพัลส์มีค่ามากจะทำให้ค่าเฉลี่ยของแรงดันมีค่ามาก หรือทำให้มอเตอร์หมุนเร็ว ในขณะเดียวกันถ้าหากว่าความกว้างของพัลส์มีค่าน้อยก็จะทำให้ค่าเฉลี่ยของแรงดันมีค่าน้อย หรือทำให้มอเตอร์หมุนได้ช้า



รูปที่ 2.6 ลักษณะการปรับค่าตัวใช้เคิล

2.4 หลักการทำงานของเซนเซอร์วัดความเร่ง

เซนเซอร์วัดความเร่งชนิดสามแกนนี้ใช้วัดความเร่งที่เกิดจากแรงภายนอกมากระทำ นอกจากนี้ยังสามารถนำมาประยุกต์เพื่อหาเวกเตอร์ลิฟท์ของความเร่ง วัดความโน้มเอียง และความสั่นสะเทือน จากความสามารถที่หลากหลายดังกล่าวจึงนำมาประยุกต์เพื่อใช้หามุมที่เปลี่ยนแปลงได้ ดังรูปที่ 2.7



รูปที่ 2.7 ทิศการวัดความเร่งของเซนเซอร์

เซนเซอร์วัดความเร่งที่เลือกใช้ คือโมดูล MPU6050 ซึ่งสามารถวัดความเร่งได้สามแกน ให้สัญญาณเอาต์พุตเป็นสัญญาณอนาล็อก โดยจะสามารถคำนวณหามุมพิทช์ (θ) ซึ่งจะได้สมการ (2.18) ซึ่งเป็นการเอียงของหุ่นยนต์จรวดยานรักษาสมดุรูปที่ 2.1

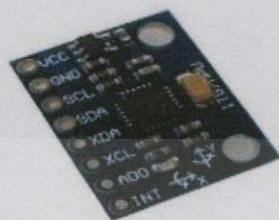
$$\theta = \arctan\left(\frac{a_y}{a_z}\right) \quad (2.18)$$

เมื่อ

A_y และ A_z คือค่าความเร่งของแกน Y และ Z ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

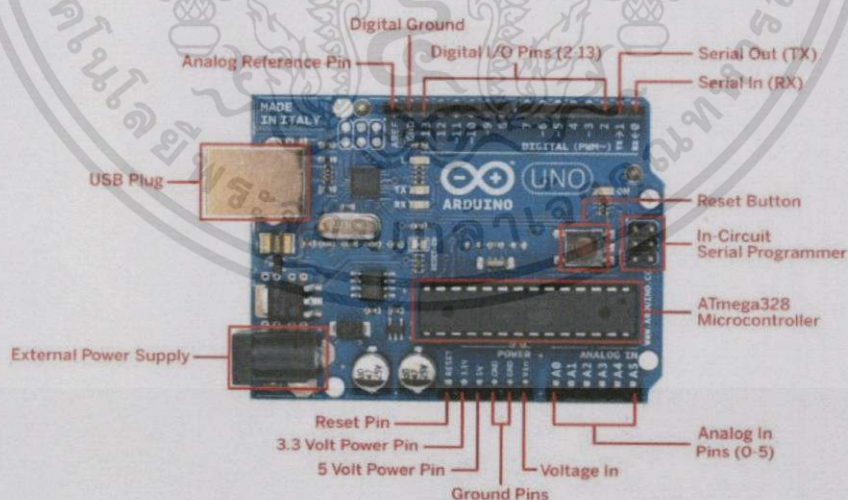
สำหรับโครงการนี้เลือกใช้เซนเซอร์วัดความเร่งแบบโมดูลสำเร็จรูปรุ่น MPU6050 ซึ่งมีลักษณะดังรูปที่ 2.8



รูปที่ 2.8 โมดูล MPU6050

2.5 หลักการทำงานของไมโครคอนโทรลเลอร์

ในโครงการนี้มีความจำเป็นที่จะต้องใช้การประมวลผลแบบตามเวลาจริง (Real time) เพราะว่าเป็นการทำงานที่เกี่ยวกับสถานะในช่วงเวลาหนึ่ง ๆ เท่านั้น หากประมวลผลช้าเกินไป จะทำให้ผลตอบสนองต่อสถานะนั้นผิดพลาดไปจากที่ต้องการ จึงต้องใช้หน่วยประมวลผลที่มีความเร็วสูง ในโครงการนี้เลือกใช้บอร์ดไมโครคอนโทรลเลอร์คือ Arduino Uno R3 ดังรูปที่ 2.9 และมีข้อมูลตามตารางที่ 2.2



รูปที่ 2.9 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ข้อมูลบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

บอร์ดไมโครคอนโทรลเลอร์	ATmega328
แรงดันในการทำงาน	5V
แรงดันอินพุต (แนะนำ)	7-12V
แรงดันอินพุต (จำกัด)	6-20V
ขาดิจิตอลอินพุต/เอาต์พุต	14 (มี 6 ขา สำหรับ PWM output)
ขานาลอกอินพุต	6
ไฟเลี้ยงขา อินพุต/เอาต์พุต	40 mA
ไฟเลี้ยงขา 3.3V อินพุต/เอาต์พุต	50 mA
หน่วยความจำ	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock speed	16 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้างชิ้นงาน

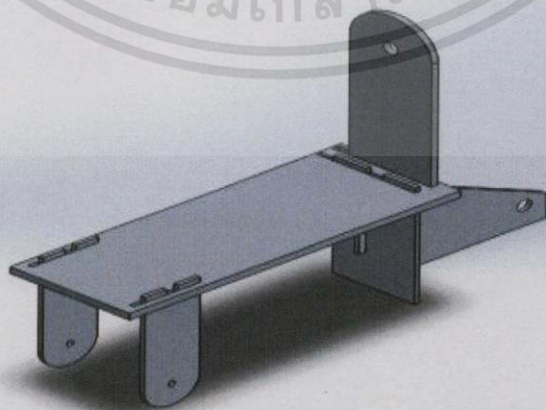
ในบทนี้กล่าวถึงการออกแบบโครงสร้างของหุ่นยนต์จักรยานรักษาสมดุล การออกแบบวงจรไฟฟ้าสำหรับควบคุมหุ่นยนต์จักรยานรักษาสมดุล และการออกแบบโปรแกรมสำหรับควบคุมเพื่อให้หุ่นยนต์จักรยานสามารถรักษาสมดุลได้

3.1 การออกแบบโครงสร้าง

การออกแบบโครงสร้างหุ่นยนต์จักรยานรักษาสมดุลต้องใช้แผ่นอะคริลิกในการสร้าง และนำชิ้นส่วนแต่ละส่วนเข้ามาประกอบกัน ดังนั้นคณะผู้จัดทำจึงใช้โปรแกรม SolidWorks ในการสร้างโครงสร้าง และใช้โปรแกรม SolidCam เพื่อเปลี่ยนการออกแบบจากโปรแกรม SolidWorks เป็น G-code แล้วให้เครื่อง CNC อ่านโค้ด และสั่งเจาะชิ้นงาน โดยหุ่นยนต์จักรยานรักษาสมดุลประกอบไปด้วยมอเตอร์ไฟฟ้า แบตเตอรี่ เป็นยึดจานหมุน ล้อ และแผงวงจร

3.1.1 โครงสร้างของหุ่นยนต์จักรยาน

โครงสร้างของหุ่นยนต์จักรยานรักษาสมดุลนี้ประกอบไปด้วยชิ้นส่วนจากอะคริลิกที่ออกแบบโดยโครงสร้างของหุ่นยนต์จักรยานรักษาสมดุลจะใช้โปรแกรม SolidWorks ออกแบบจะได้โครงสร้างภาพวาด 3 มิติ ดังรูปที่ 3.1

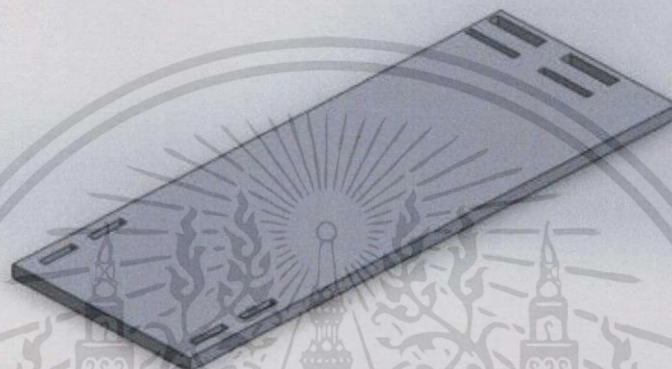


รูปที่ 3.1 ภาพวาด 3 มิติ ของโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานกลาง

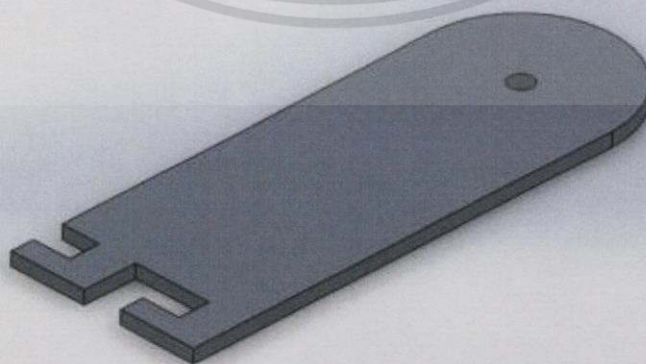
ฐานกลางของหุ่นยนต์จักรยานรักษาสมดุลทำขึ้นโดยใช้แผ่นอะคริลิก หนา 4 มิลลิเมตร ความกว้าง 5.45 เซนติเมตร ยาว 15 เซนติเมตร จำนวน 1 ชิ้น เป็นรูปทรงสี่เหลี่ยมผืนผ้า และทำการเจาะรูไว้เพื่อนำชิ้นส่วนต่าง ๆ เข้ามาประกอบกัน โดยชิ้นส่วนที่มาประกอบนั้นได้ออกแบบให้ด้านปลายมีลักษณะคล้ายรูปตัว L เพื่อนำมาสวม และทำการเลื้อนเพื่อยึดให้แน่น ดังรูปที่ 3.2



รูปที่ 3.2 ฐานกลาง

ก้านยึดล้อหลัง

ก้านยึดล้อหลังของหุ่นยนต์จักรยานรักษาสมดุลทำขึ้นโดยใช้แผ่นอะคริลิก หนา 2 มิลลิเมตร ความกว้าง 2.5 เซนติเมตร ยาว 6.95 เซนติเมตร เจาะรูมีเส้นผ่าศูนย์กลาง 3 มิลลิเมตร จำนวน 2 ชิ้น เป็นรูปทรงคล้ายกับขาตะเกียบจักรยาน ด้านปลายทำเป็นรูปตัว L เพื่อนำไปยึดกับฐานกลาง ดังรูปที่ 3.3



รูปที่ 3.3 ก้านยึดล้อหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผ่นยึดมอเตอร์กับจานหมุน

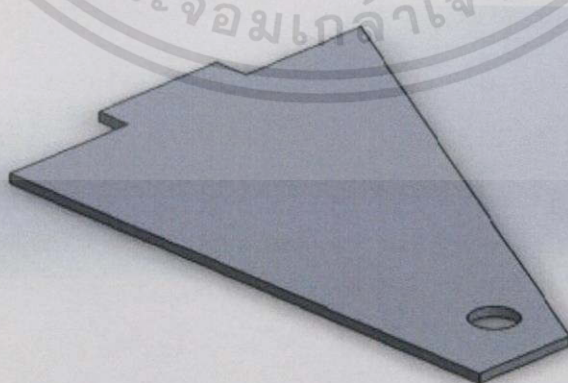
แผ่นยึดมอเตอร์กับจานหมุนของหุ่นยนต์จักรยานรักษาสมดุสทำขึ้นโดยใช้แผ่นอะคริลิก หนา 3 มิลลิเมตร ความกว้าง 3.75 เซนติเมตร ยาว 9.775 เซนติเมตร เจาะรูมีเส้นผ่าศูนย์กลาง 6.5 มิลลิเมตร จำนวน 1 ชิ้น เป็นรูปทรงคล้ายกับขาตะเกียบจักรยาน ด้านปลายทำเป็นรูปตัว L เพื่อนำไปยึดกับฐานกลางดังรูปที่ 3.4



รูปที่ 3.4 แผ่นยึดมอเตอร์กับจานหมุน

แผ่นยึดมอเตอร์กับล้อหน้า

แผ่นยึดมอเตอร์กับล้อหน้าของหุ่นยนต์จักรยานรักษาสมดุสทำขึ้นโดยใช้แผ่นอะคริลิก หนา 2 มิลลิเมตร ความกว้าง 8.5 เซนติเมตร ยาว 5.95 เซนติเมตร เจาะรูมีเส้นผ่าศูนย์กลาง 6.5 มิลลิเมตร จำนวน 1 ชิ้น ด้านข้างทำเป็นสี่เหลี่ยมยื่นออกมา 2 เซนติเมตร เพื่อนำไปยึดติดกับแผ่นรับน้ำหนักมอเตอร์ ดังรูปที่ 3.5

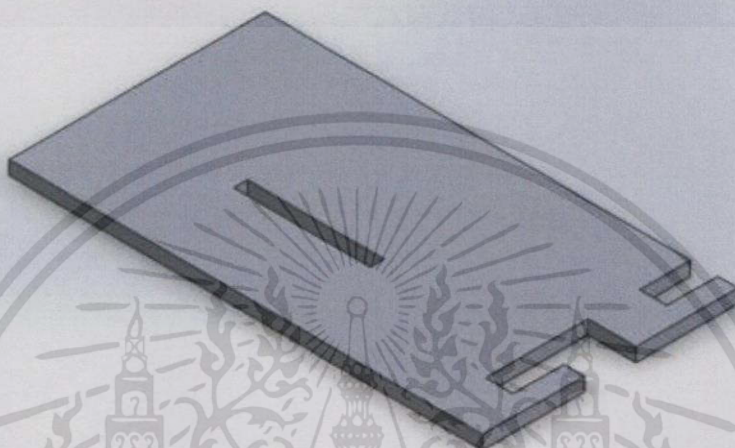


รูปที่ 3.5 แผ่นยึดมอเตอร์กับล้อหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผ่นรับน้ำหนักมอเตอร์

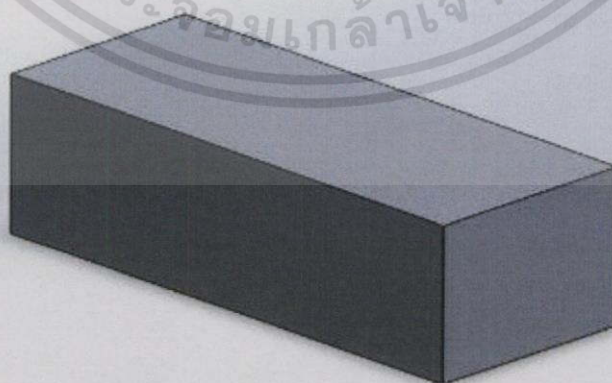
แผ่นรับน้ำหนักมอเตอร์ของหุ่นยนต์จักรยานรักษาสมดุลทำขึ้นโดยใช้แผ่นอะคริลิก หนา 2 มิลลิเมตร ความกว้าง 3.75 เซนติเมตร ยาว 6.95 เซนติเมตร เจาะรูมีความกว้าง 2 มิลลิเมตร ยาว 2 เซนติเมตร จำนวน 1 ชิ้น เพื่อรับน้ำหนักแผ่นยึดมอเตอร์กับล้อหน้า ดังรูปที่ 3.6



รูปที่ 3.6 แผ่นรับน้ำหนักมอเตอร์

ตัวล็อกแบบที่ 1

ตัวล็อกแบบที่ 1 ทำขึ้นโดยใช้แผ่นอะคริลิก หนา 4 มิลลิเมตร ความกว้าง 3 มิลลิเมตร ยาว 7.5 มิลลิเมตร จำนวน 2 ชิ้น เพื่อนำไปล็อกช่องว่างระหว่างฐานกลางกับแผ่นยึดมอเตอร์กับจานหมุน ดังรูปที่ 3.7

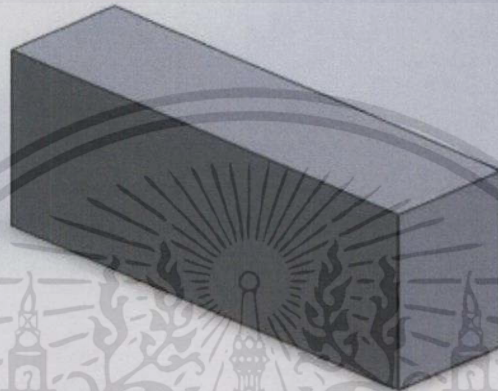


รูปที่ 3.7 ตัวล็อกแบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวล๊อคแบบที่ 2

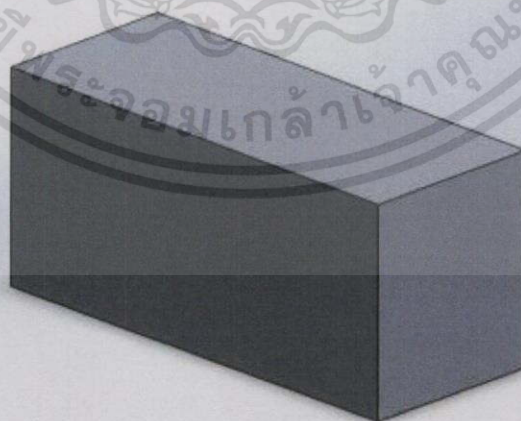
ตัวล๊อคแบบที่ 2 ทำขึ้นโดยใช้แผ่นอะคริลิก หนา 4 มิลลิเมตร ความกว้าง 2 มิลลิเมตร ยาว 7.5 มิลลิเมตร จำนวน 2 ชิ้น เพื่อนำไปล๊อคช่องว่างระหว่างฐานกลางกับแผ่นรับน้ำหนักมอเตอร์ ดังรูปที่ 3.8



รูปที่ 3.8 ตัวล๊อคแบบที่ 2

ตัวล๊อคแบบที่ 3

ตัวล๊อคแบบที่ 3 ทำขึ้นโดยใช้แผ่นอะคริลิก หนา 4 มิลลิเมตร ความกว้าง 2 มิลลิเมตร ยาว 5 มิลลิเมตร จำนวน 4 ชิ้น เพื่อนำไปล๊อคช่องว่างระหว่างฐานกลางกับก้านยึดล้อหลัง ดังรูปที่ 3.9

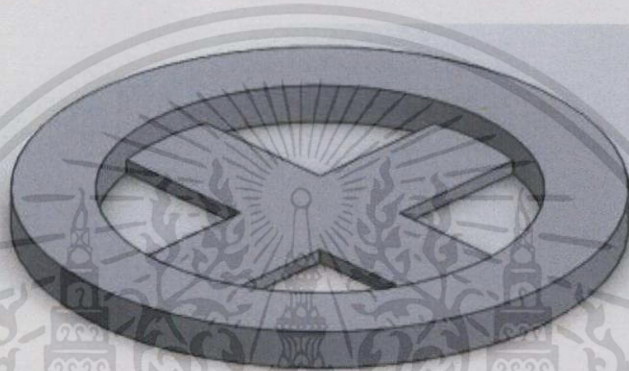


รูปที่ 3.9 ตัวล๊อคแบบที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 งานหมุน

งานหมุนของหุ่นยนต์จักรยานรักษาสมดุลทำขึ้นโดยใช้แผ่นอะคริลิก หนา 10 มิลลิเมตร เส้นผ่าศูนย์กลาง 11 เซนติเมตร จำนวน 1 ชิ้น เป็นรูปทรงคล้ายล้อรถยนต์ หลังจากนั้นก็นำมา เจาะรูเพื่อยึดกับแป้นยึดชิ้นงาน และอีกด้านของแป้นยึดชิ้นงานก็นำเพลามอเตอร์มาสวม โดยใช้น็อต ตัวหนอนยึดเพลามอเตอร์กับแป้นยึดชิ้นงาน เพื่อนำมาขับงานหมุน ดังรูปที่ 3.10



รูปที่ 3.10 งานหมุน

3.1.3 แป้นยึดงานหมุน

แป้นยึดงานหมุนที่เลือกใช้สำหรับโครงงานนี้คือ แป้นยึดงานหมุนแบบพลาสติกสำเร็จรูปมี ขนาดเส้นผ่าศูนย์กลาง 4 เซนติเมตร ขนาดความกว้างของแกนสวมเพล่า 6 มิลลิเมตร และมีรูสำหรับ นำน็อตมายึดกับงานหมุน ขนาดรู 3 มิลลิเมตร จำนวน 4 รู ดังรูปที่ 3.11



รูปที่ 3.11 แป้นยึดงานหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 ล้อ

ล้อที่เลือกใช้สำหรับโครงงานนี้ คือล้อสำหรับมอเตอร์เกียร์มีความหนา 10 มิลลิเมตร ขนาดเส้นผ่าศูนย์กลาง 6.9 เซนติเมตร และขนาดแกนสวมเพลาคือ 5.4 มิลลิเมตร ดังรูปที่ 3.12



รูปที่ 3.12 ล้อ

3.1.5 มอเตอร์ไฟฟ้า

การเลือกใช้มอเตอร์ไฟฟ้าสำหรับขับโครงงานนี้มี 2 ตัว คือ มอเตอร์ไฟฟ้าความเร็วรอบสูง และมอเตอร์ไฟฟ้าแบบธรรมดา

มอเตอร์ไฟฟ้าความเร็วรอบสูงที่ใช้ในโครงงานนี้ แสดงดังรูปที่ 3.13 ซึ่งใช้ขับจานหมุน โดยใช้ชุดทดเกียร์ 40:1 และมีรายละเอียดมอเตอร์ไฟฟ้าความเร็วรอบสูงดังนี้

รอบ : มากกว่า 35000 r/min (ขึ้นกับโหลดแรงบิด)

แรงดันไฟฟ้า : 3.0 V

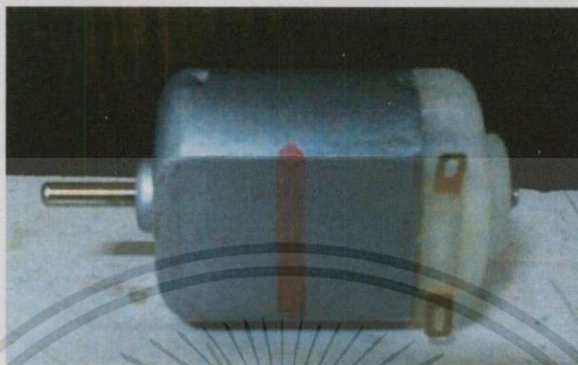


รูปที่ 3.13 มอเตอร์ไฟฟ้าความเร็วรอบสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ไฟฟ้าความเร็วแบบธรรมดาที่ใช้ในโครงงานนี้ แสดงดังรูปที่ 3.14 โดยใช้ชุดทดเกียร์

40:1



รูปที่ 3.14 มอเตอร์ไฟฟ้าแบบธรรมดา

3.1.6 แบตเตอรี่

แบตเตอรี่ที่เลือกใช้สำหรับโครงงานนี้คือ แบตเตอรี่ลิเทียมโพลิเมอร์ ยี่ห้อ MYSTORY ดังรูปที่ 3.15 จำนวน 1 ก้อน และมีรายละเอียดดังนี้

แรงดันไฟฟ้า : 7.4 V

ขนาดความจุของแบตเตอรี่ : 1500 mAh

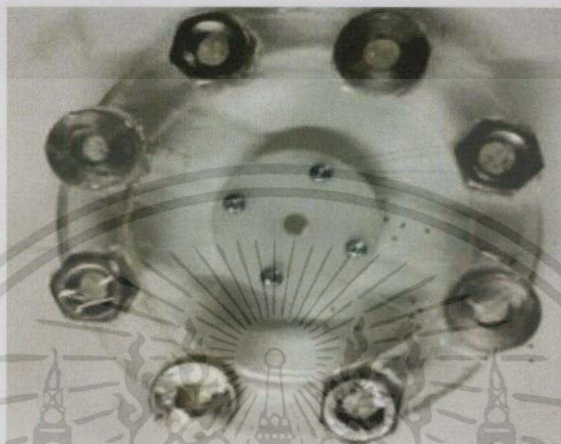


รูปที่ 3.15 แบตเตอรี่ลิเทียมโพลิเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.7 ตัวเพิ่มน้ำหนักรวม

ตัวเพิ่มน้ำหนักรวมมี 8 ชั้น นำมาติดกับจานหมุน เพื่อเพิ่มแรงการเหวี่ยงของจานหมุนให้มากขึ้น ดังรูปที่ 3.16



รูปที่ 3.16 ตัวเพิ่มน้ำหนักรวม

3.1.8 เหล็กฉากรับน้ำหนัก

เหล็กฉากรับน้ำหนักจำนวน 2 ชิ้น นำมารับน้ำหนักแผ่นรับน้ำหนักมอเตอร์เข้ากับฐานกลางเพื่อความแข็งแรง และป้องกันการหักของแผ่นรับน้ำหนักมอเตอร์ ดังรูปที่ 3.17



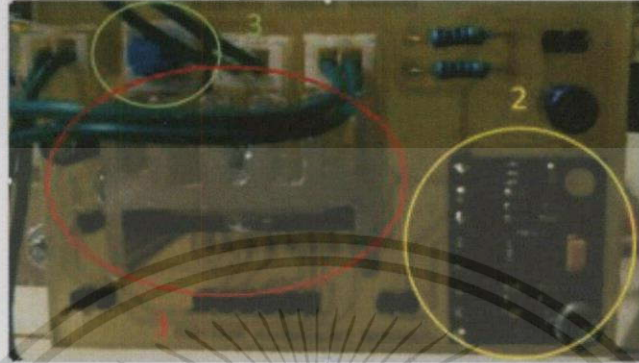
รูปที่ 3.17 เหล็กฉากรับน้ำหนัก

3.1.9 แผงวงจร

แผงวงจรที่ใช้ในการควบคุมหุ่นยนต์จักรยานรักษาสมดุล รวมถึงใช้สำหรับรับและส่งข้อมูลไปยังบอร์ด Arduino Uno R3 ถูกออกแบบให้มีขนาดความกว้าง 4.5 เซนติเมตร ความยาว 6.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนติเมตร และติดตั้งบนหุ่นยนต์จักรยานรักษาสมดุล โดยแผงวงจรประกอบมีลักษณะของแผงวงจร แสดงดังรูปที่ 3.18 และมีส่วนประกอบที่สำคัญดังนี้

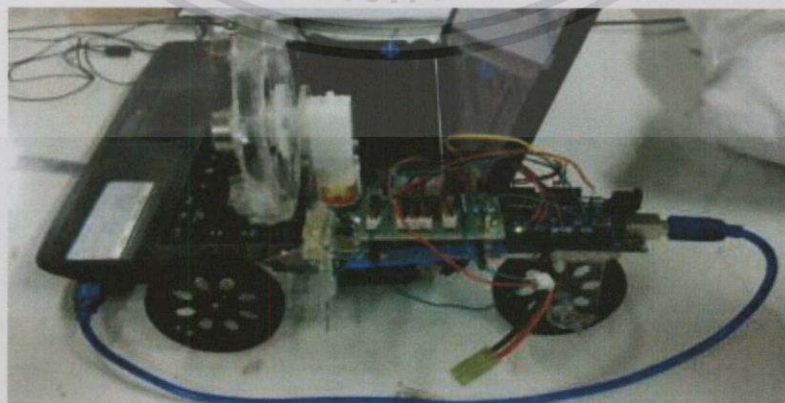


รูปที่ 3.18 แผงวงจร

- หมายเลขที่ 1 คือ วงจรขับเคลื่อนมอเตอร์โดยการใช้ไอซี L298N โดยจะใส่แผ่นระบายความร้อนติดกับไอซี L298N
- หมายเลขที่ 2 คือ โมดูล MPU6050 เป็นเซนเซอร์ตรวจสอบความเร่ง โดยการเชื่อมต่อ I2C ระหว่างบอร์ดควบคุมในการรับและส่งข้อมูล
- หมายเลขที่ 3 คือ สวิตช์เป็นตัวเปิด/ปิด ในการป้อนไฟเลี้ยงสู่บอร์ดควบคุมจากแบตเตอรี่ โดยโครงการนี้ไม่ได้ใช้สวิตช์มา เปิด/ปิด การป้อนไฟเลี้ยง

3.1.10 โครงสร้างหุ่นยนต์จักรยานเมื่อประกอบชิ้นส่วนเข้าด้วยกัน

หลังจากออกแบบชิ้นส่วนหุ่นยนต์จักรยาน และแผงวงจร จึงนำมาประกอบเข้าด้วยกันดังรูปที่ 3.19



รูปที่ 3.19 หุ่นยนต์จักรยานเมื่อประกอบชิ้นส่วนเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบวงจรไฟฟ้า

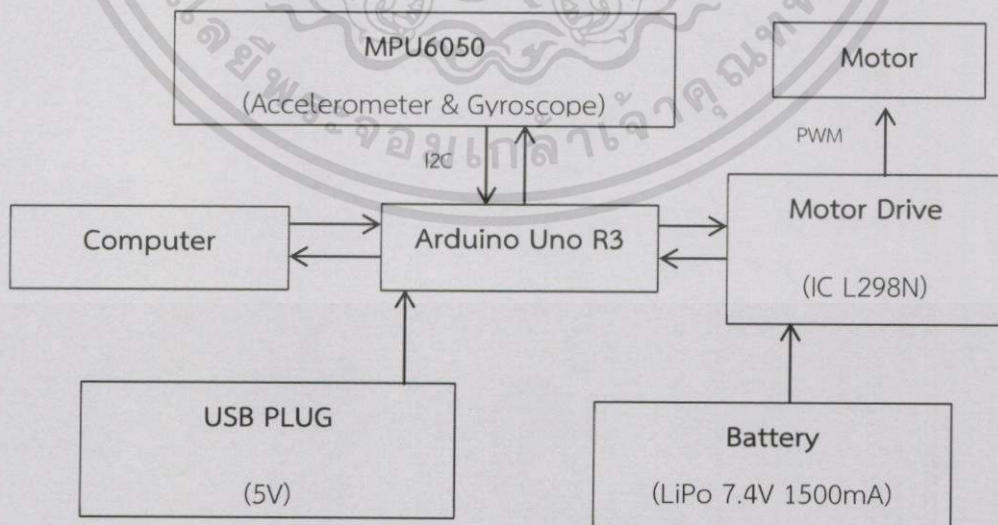
ในหัวข้อนี้จะกล่าวถึงการออกแบบวงจรไฟฟ้าที่ใช้ควบคุมหุ่นยนต์จักรยานรักษาสมดุ โดย การออกแบบวงจรไฟฟ้าแบ่งเป็น 3 ส่วน คือ วงจรขับเคลื่อนมอเตอร์ ไมโครคอนโทรลเลอร์และเซนเซอร์ และโมดูล MPU6050 เพื่อเชื่อมต่อเข้ากับบอร์ด Arduino Uno R3

3.2.1 การออกแบบวงจรขับเคลื่อนมอเตอร์

ไอซี L298N ดังรูปที่ 2.5 ต้องมีการนำตัวเก็บประจุ 100nF มาต่อเข้ากับขา 4 กับ 9 โดยจะ ทำหน้าที่กรองแรงดันหรือฟิลเตอร์ (Filter) เพราะในการเปลี่ยนแรงดันไฟกระแสสลับให้เป็น แรงดันไฟกระแสตรงจะมีการกระเพื่อมของแรงดันไฟตรง ที่เรียกว่า ริปเปิ้ล (Ripple) ดังนั้นจึงใช้ตัว เก็บประจุเพื่อลดค่าแรงดันริปเปิ้ลลงไป โดยการเก็บค่าประจุไว้เมื่อช่วงแรงดันสูง และจะจ่ายประจุ ให้กับโหลดเมื่อมีการกระเพื่อมทางด้านต่ำ ดังนั้นโหลดจะได้แรงดันที่ราบเรียบขึ้น

3.2.2 วงจรไมโครคอนโทรลเลอร์และเซนเซอร์

วงจรไมโครคอนโทรลเลอร์และเซนเซอร์ประกอบไปด้วยบอร์ดไมโครคอนโทรลเลอร์จำนวน 1 บอร์ด คือ Arduino Uno R3 โดยบอร์ด Arduino Uno R3 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ทำ หน้าที่รับค่าจากเซนเซอร์วัดความเร่ง โดยการสื่อสารแบบ I2C และนำค่าส่งไปให้บอร์ด และส่งไปยัง คอมพิวเตอร์ เพื่อประมวลผลสัญญาณ ซึ่งต่อที่ขานาฬิกาอินพุตของบอร์ดไมโครคอนโทรลเลอร์ และ วงจรขับเคลื่อนมอเตอร์ก็จะเชื่อมต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ที่ขาดิจิตอล ซึ่งบอร์ดก็จะประมวลผล และสร้างสัญญาณพัลส์วิธมอดดูเลชัน เพื่อไปควบคุมความเร็วมอเตอร์ไฟฟ้า ดังรูปที่ 3.20



รูปที่ 3.20 การเชื่อมต่อวงจรต่าง ๆ กับบอร์ด Arduino Uno R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบบวงจรประกอบโมดูล MPU6050

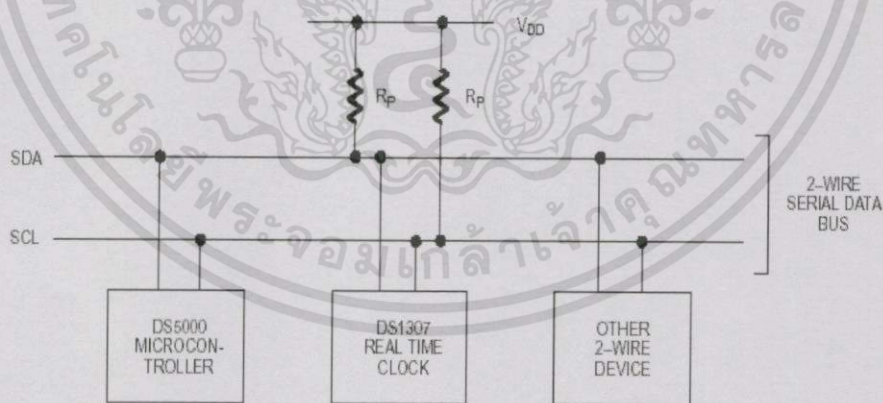
การใช้โมดูล MPU6050 นั้นจะต้องใช้ I2C เข้ามาเป็นตัวสื่อสารกับบอร์ด Arduino Uno R3 โดยการสื่อสารนั้นผ่านสายสัญญาณเพียง 2 เส้นเท่านั้น เพื่อลดพอร์ตการเชื่อมต่อของ Arduino Uno R3

การเชื่อมต่อแบบ I2C

I2C ย่อมาจาก Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้น ๆ I2C เป็นการสื่อสารอนุกรมแบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ Serial data (SDA) และสาย Serial clock (SCL) ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนหลาย ๆ ตัวเข้าด้วยกันได้ ทำให้ไมโครคอนโทรลเลอร์ (MCU) ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

การเชื่อมต่อผ่านสายสัญญาณ

I2C BUS ใช้สายสัญญาณ 2 เส้น คือ SCL, SDA สำหรับติดกับอุปกรณ์แบบ 2 ทิศทาง โดยที่ขาสัญญาณทั้ง 2 จะต้องต่อกับต่อกับ $2k\Omega < R_p < 10k\Omega$ เนื่องจากเอาต์พุตมีลักษณะเป็นแบบ Open Drain หรือเป็นแบบ Open Collector เพื่อให้เอาต์พุตเชื่อมต่อกันได้หลายตัว ซึ่งในโครงการนี้เลือกใช้ $R_p = 4.7k\Omega$ ลักษณะการเชื่อมต่อแสดงดังรูปที่ 3.21



รูปที่ 3.21 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I2C BUS

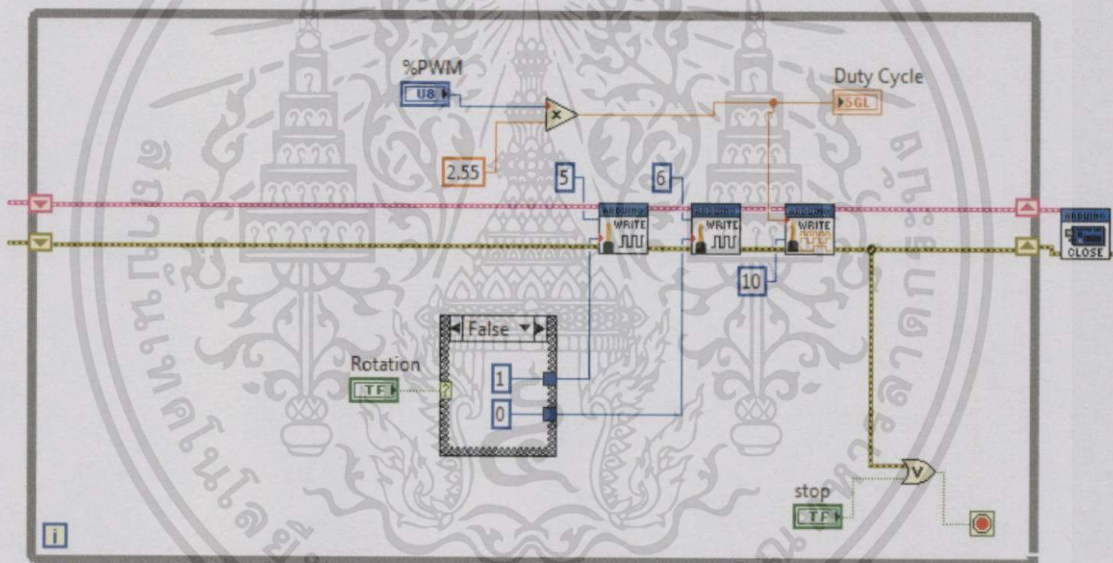
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบโปรแกรม

ในหัวข้อนี้จะกล่าวถึงแนวความคิดการออกแบบโปรแกรมควบคุมหุ่นยนต์จักรยานรักษาสมดุล ซึ่งจะมีอยู่ 3 ส่วนคือ โปรแกรมป้อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชัน โปรแกรมอ่านเซนเซอร์วัดความเร็ว และโปรแกรมควบคุมแบบพีไอดี โดยบอร์ด Arduino Uno R3 จะเชื่อมต่อกับโปรแกรม LabVIEW โดยการติดตั้งไดร์เวอร์ [3]

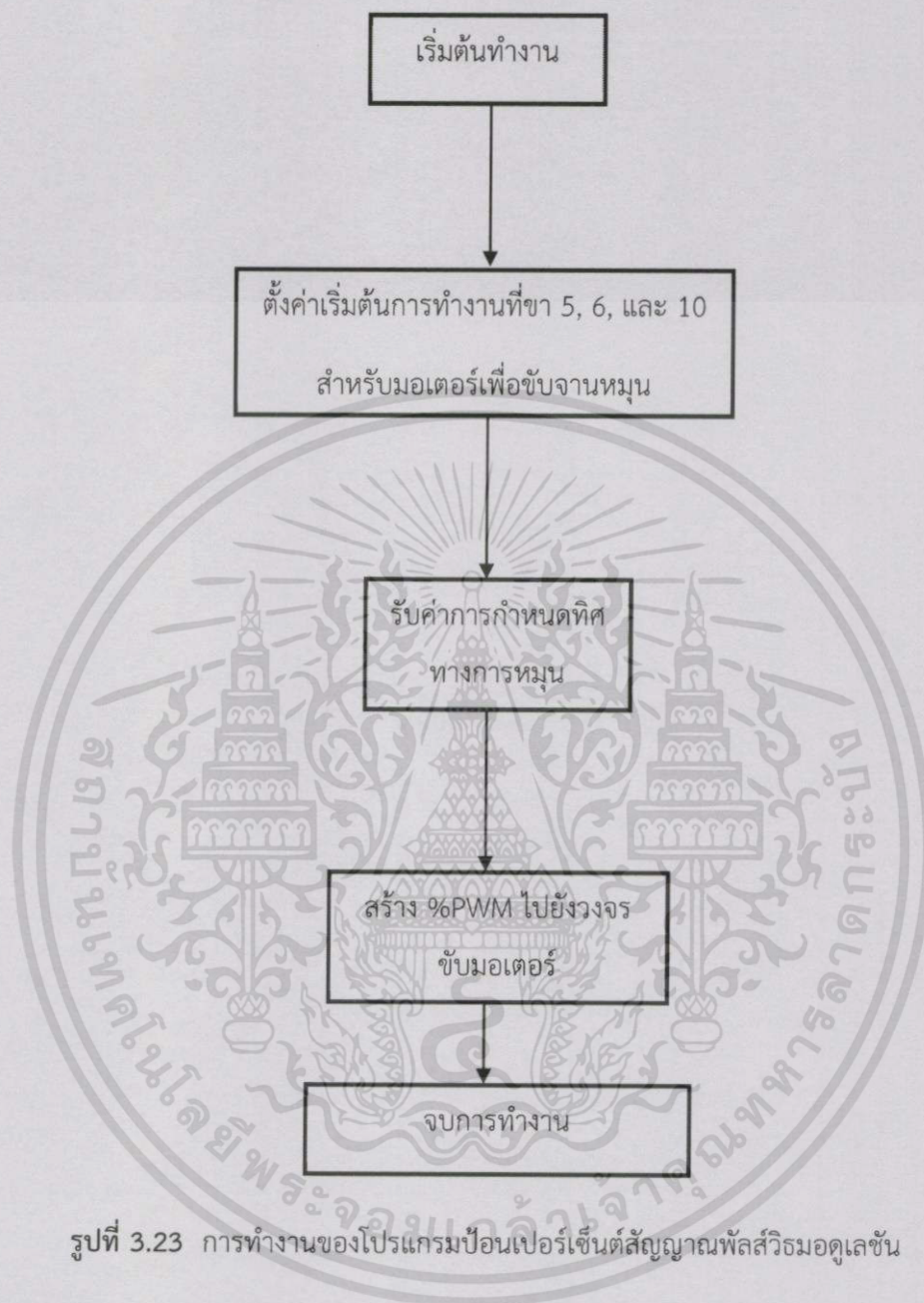
3.3.1 โปรแกรมป้อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชัน

การออกแบบโปรแกรมป้อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชัน จะทำการป้อนสัญญาณไปยังขาคิดจิตอลที่ 5, 6, และ 10 เพื่อสั่งให้มอเตอร์ทำการหมุน มีบล็อกการทำงานดังรูปที่ 3.22



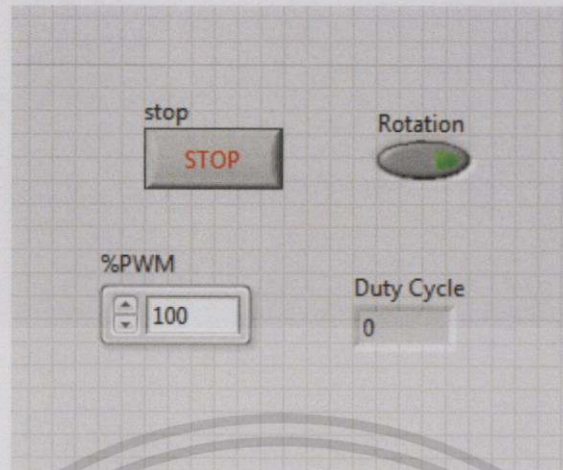
รูปที่ 3.22 บล็อกโปรแกรมป้อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชัน

ขั้นตอนการทำงานของบล็อกโปรแกรมป้อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชันจะทำการตั้งค่าเริ่มต้นการทำงานที่ขา 5, 6 และ 10 และทำการรับค่าการกำหนดทิศทาง ซึ่งการกำหนดทิศทางจะสร้างเงื่อนไขไว้ในโปรแกรม จากนั้นจะสร้างเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชันผ่านหน้าต่างโปรแกรม เพื่อทำการขับมอเตอร์ โดยจะมีแผนภาพการทำงานดังรูปที่ 3.23



รูปที่ 3.23 การทำงานของโปรแกรมป้อนเปอร์เซ็นต์สัญญาณพัลส์วีสวมอดูเลชัน

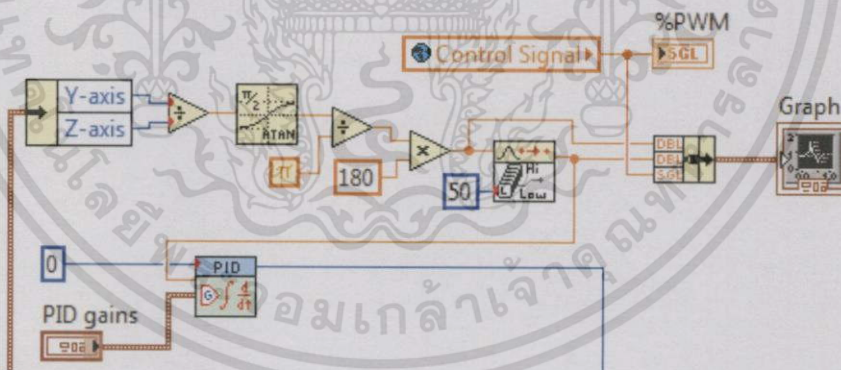
การทำงานสามารถกำหนดค่าเปอร์เซ็นต์สัญญาณพัลส์วีสวมอดูเลชัน และกำหนดทิศทางการหมุนจากหน้าต่างโปรแกรมดังรูปที่ 3.24



รูปที่ 3.24 หน้าต่างโปรแกรมบ็อนเปอร์เซ็นต์สัญญาณพัลส์วิธึมอดูลเซ็น

3.3.2 โปรแกรมอ่านเซนเซอร์วัดความเร็ว

การออกแบบโปรแกรมอ่านเซนเซอร์วัดความเร็วจะทำการหามุมพิทช์ และนำไปหาค่ามัธยฐานของข้อมูล เพราะหากเกิดการสั่นสะเทือนของมอเตอร์จากการขับจานหมุน ก็จะทำให้มุมพิทช์นั้นมีการผิดเพี้ยนได้ ทั้งที่ตัวรถยังไม่มีการเอียง ซึ่งมีบล็อกการทำงานดังรูปที่ 3.25



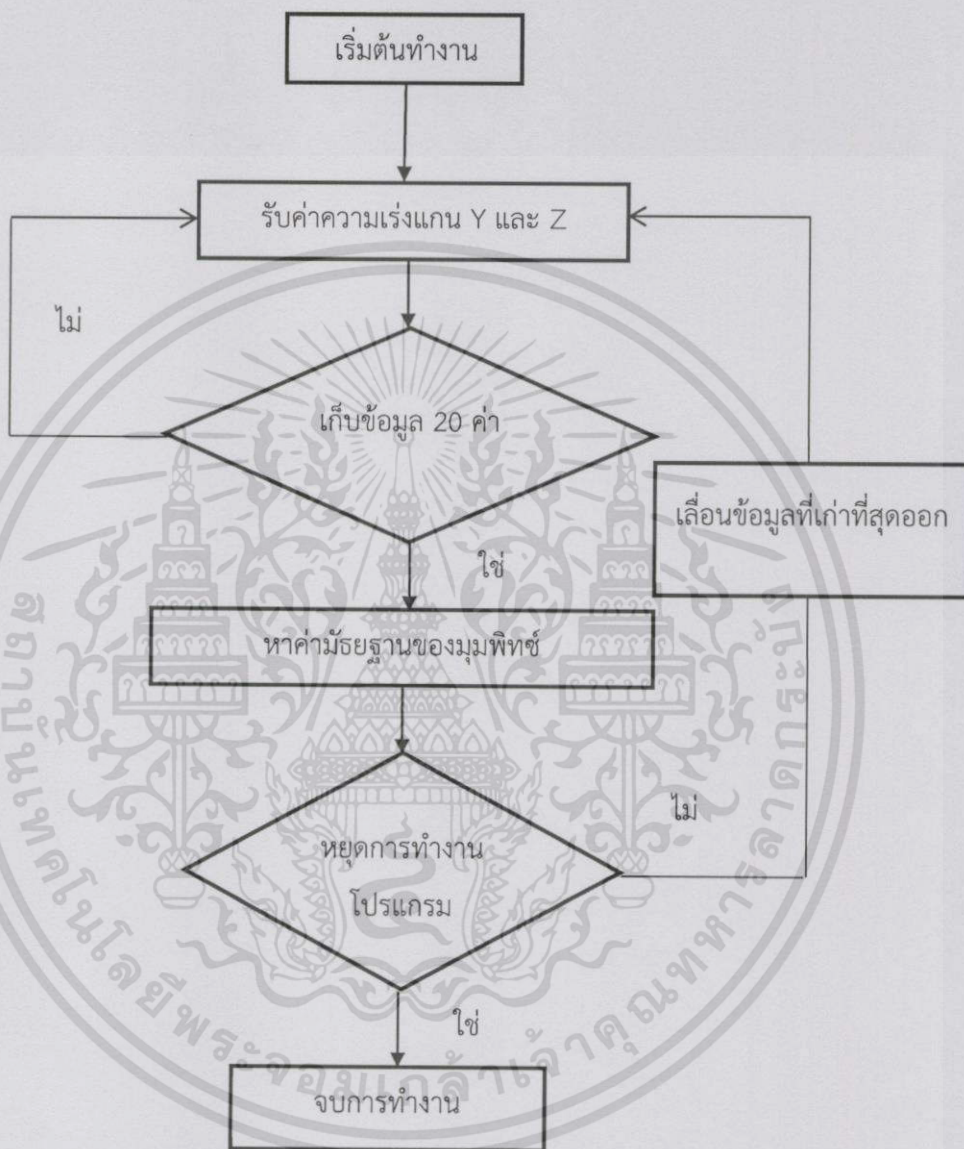
รูปที่ 3.25 บล็อกโปรแกรมอ่านเซนเซอร์วัดความเร็ว

ขั้นตอนการทำงานของบล็อกโปรแกรมอ่านเซนเซอร์วัดความเร็วจะรับค่าความเร็วในแกน Y และ Z ที่อ่านได้จากโมดูล MPU6050 มาแทนลงในสมการ (2.18) และนำมุมพิทช์มาแปลงเป็นมุมพิทช์ในหน่วยองศาตามสมการ (3.1)

$$\theta_{degree} = \theta_{radian} \cdot \frac{180}{\pi} \quad (3.1)$$

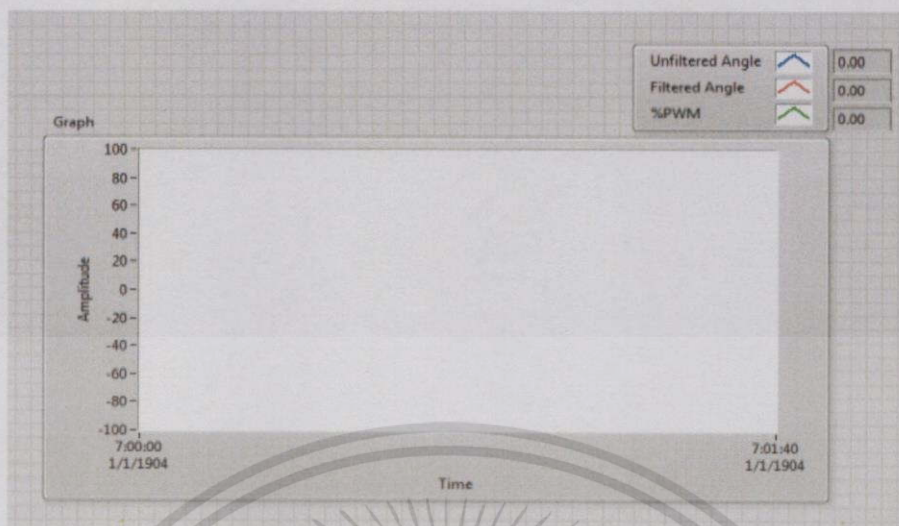
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นก็นำมัมพิทซ์ที่ได้มาทำการหาค่ามัธยฐาน โดยการช้กมาข้อมูล 20 ค่า แล้วนำการอ่านค่ามัมพิทซ์มาแสดงเป็นกราฟ โดยจะมีแผนภาพการทำงานดังรูปที่ 3.26



รูปที่ 3.26 การทำงานของโปรแกรมอ่านเซนเซอร์วัดความเร่ง

หน้าต่างโปรแกรมอ่านค่าเซนเซอร์วัดความเร่งจะแสดงการเปรียบเทียบของมัมพิทซ์ที่ทำการหาค่ามัธยฐาน โดยกราฟสีแดง คือมัมพิทซ์ที่ทำการหาค่ามัธยฐาน และกราฟสีน้ำเงิน คือมัมพิทซ์ที่ไม่ได้ทำการหาค่ามัธยฐาน ดังหน้าต่างโปรแกรมรูปที่ 3.27



รูปที่ 3.27 หน้าต่างโปรแกรมอ่านค่าเซนเซอร์วัดความเร็ว

3.3.3 โปรแกรมการควบคุมแบบพีไอดี

การออกแบบโปรแกรมการควบคุมแบบพีไอดีจะอาศัยบล็อกการควบคุมแบบพีไอดีสำเร็จรูปของโปรแกรม LabVIEW โดยเพิ่มส่วนปรับแต่งสัญญาณ การใช้งานผู้ใช้สามารถปรับค่าอัตราขยายเชิงสัดส่วน อัตราขยายเชิงปริพันธ์ อัตราขยายเชิงอนุพันธ์ และค่าเป้าหมาย (Set point: SP)

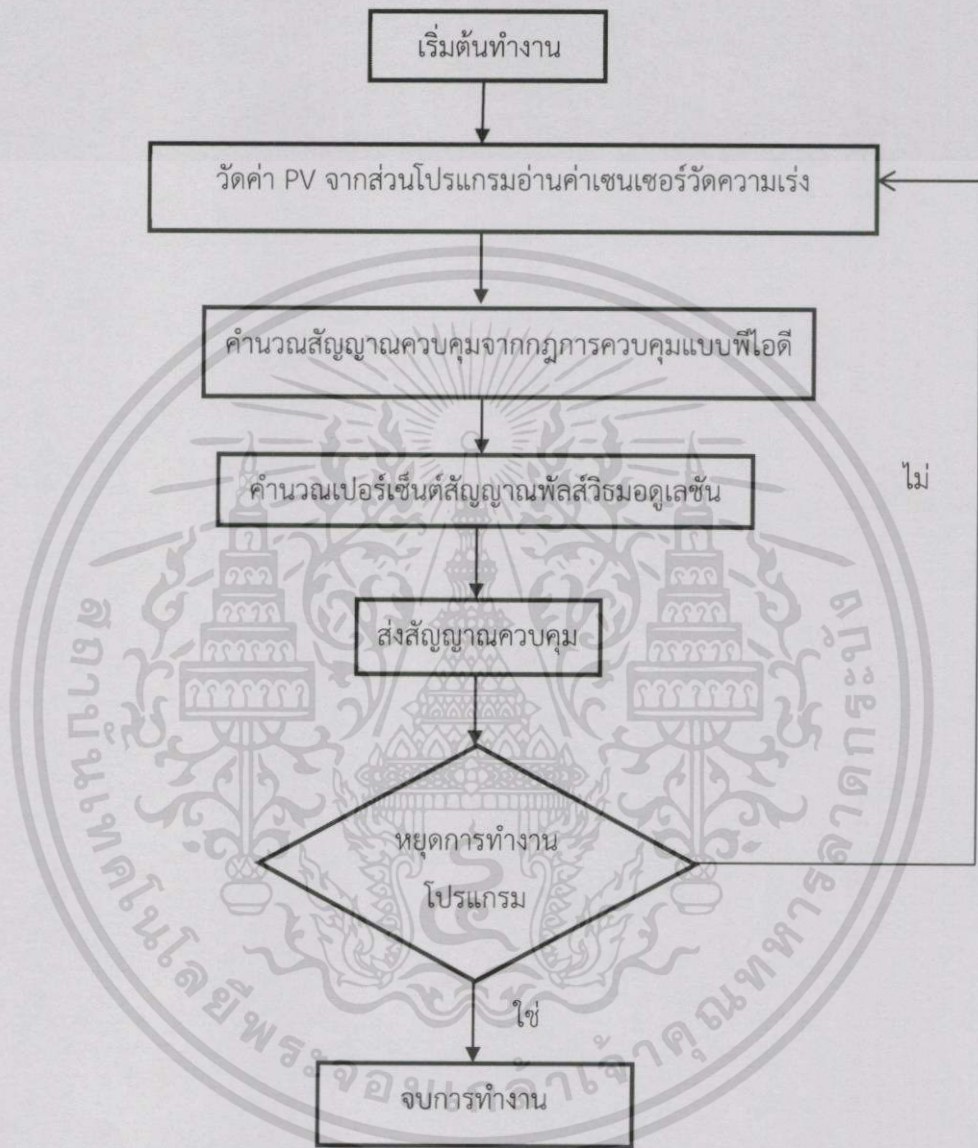
ขั้นตอนการทำงานของบล็อกโปรแกรมการควบคุมแบบพีไอดี แสดงได้ดังรูปที่ 3.28 กล่าวคือ เมื่อสามารถอ่านค่ามุมพิทช์ซึ่งเป็น PV (Process variable) [2] ของระบบสามารถคำนวณค่าความผิดพลาด โดยมาลบกับค่า SP ซึ่งโดยปกติมีค่าเท่ากับศูนย์ หรือคือจุดสมดุลของระบบ ค่าความผิดพลาดที่ได้จะผ่านกฎการควบคุมแบบพีไอดีตามสมการ (2.17) ได้สัญญาณควบคุม $u(t)$ จากสัญญาณควบคุมที่ได้ จึงคำนวณหาเปอร์เซ็นต์สัญญาณพัลส์วีสริมอดูเลชันตามสมการ (3.2) เมื่อสัญญาณค่าความผิดพลาดเป็นบวก ซึ่งจะทำให้หุ่นยนต์จักรยานรักษาสมดุลพยายามเอียงไปทางขวา

$$\%PWM = \left| \frac{u(t)}{255} \cdot 100 \right| + P \quad (3.2)$$

หรือสมการ (3.3) เมื่อสัญญาณค่าความผิดพลาดเป็นลบ ซึ่งจะทำให้หุ่นยนต์จักรยานรักษาสมดุลพยายามเอียงไปทางซ้าย

$$\%PWM = -\left(\left| \frac{u(t)}{255} \cdot 100 \right| + P \right) \quad (3.3)$$

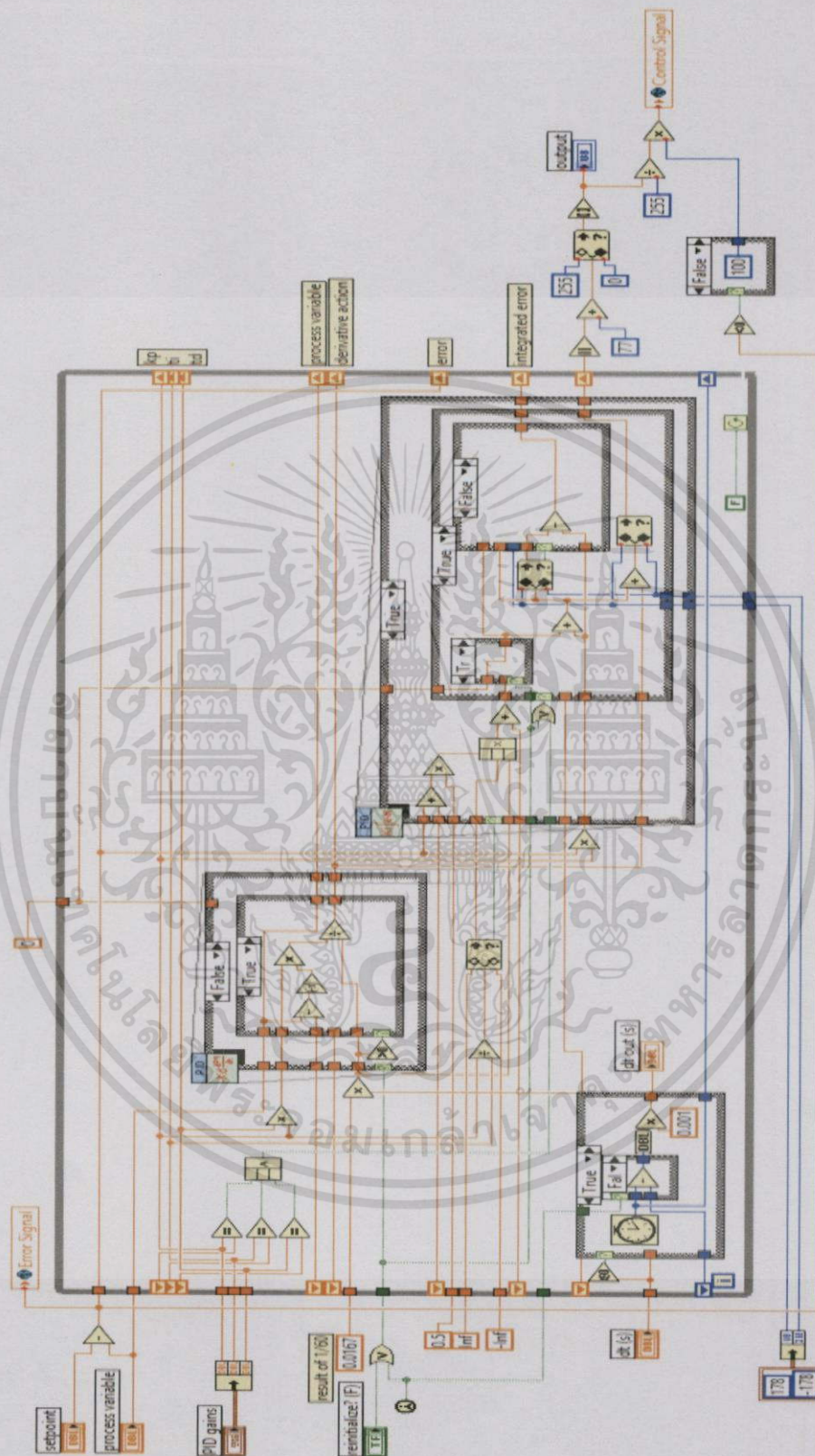
โดย P คือ เปอร์เซนต์สัญญาณพัลส์วริมอดูเลชันขั้นต่ำสุดที่ป้อนให้กับมอเตอร์ ก่อนมอเตอร์จะเริ่มหมุน ซึ่งมีค่าเท่ากับ 30 ดังจะกล่าวต่อไปในหัวข้อ 4.3



รูปที่ 3.28 การทำงานของโปรแกรมควบคุมแบบพีไอดี

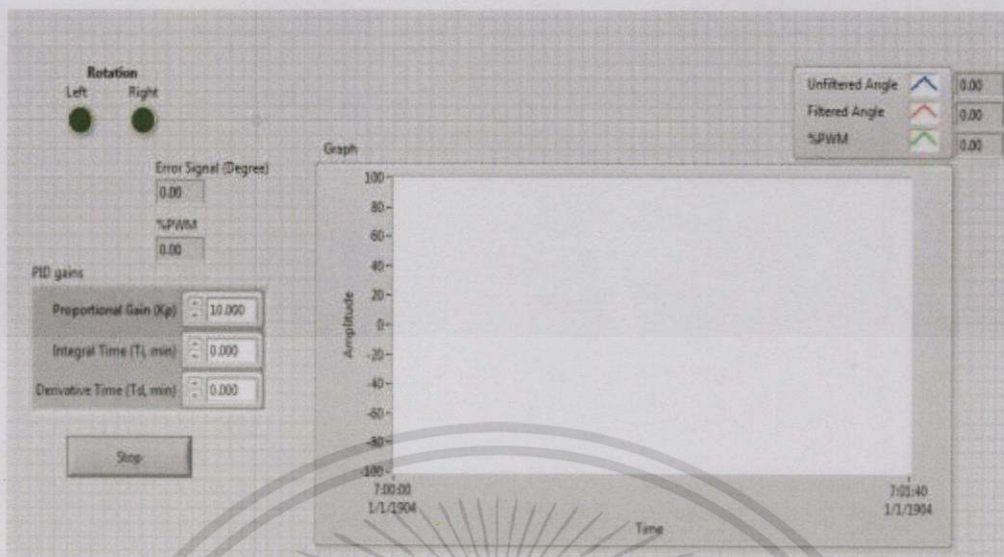
จากขั้นตอนการทำงานของโปรแกรมควบคุมแบบพีไอดีที่กล่าวข้างต้น สามารถสร้างบล็อกการทำงานของโปรแกรมควบคุมแบบพีไอดีด้วยโปรแกรม LabVIEW ดังรูปที่ 3.29 และมีการแสดงผลในหน้าต่างโปรแกรม ดังรูปที่ 3.30 โดยกราฟสีเขียว คือเปอร์เซนต์สัญญาณพัลส์วริมอดูเลชันที่ป้อนให้กับหุ่นยนต์จักรยานรักษาสมดุล กราฟสีแดง คือมุมพิทช์ที่ทำการหาค่ามัธยฐาน และกราฟสีน้ำเงิน คือมุมพิทช์ที่ไม่ได้ทำการหาค่ามัธยฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 บล็อกโปรแกรมการควบคุมแบบพีไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 หน้าต่างโปรแกรมการควบคุมแบบพีไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้นำเสนอการทดลองของโครงการระบบควบคุมหุ่นยนต์จักรยานรักษาสมดุล ซึ่งแบ่งออกเป็นการทดลองวัดค่ามุมพิทช์ก่อนและหลังการหาค่ามัธยฐาน การทดลองวัดค่ามุมพิทช์เมื่อเอียงหุ่นยนต์จักรยานรักษาสมดุล การทดลองขับมอเตอร์เมื่อแปรค่าเปอร์เซ็นต์สัญญาณพัลส์วิธึมอดูเลชัน และการทดลองควบคุมหุ่นยนต์จักรยานรักษาสมดุล

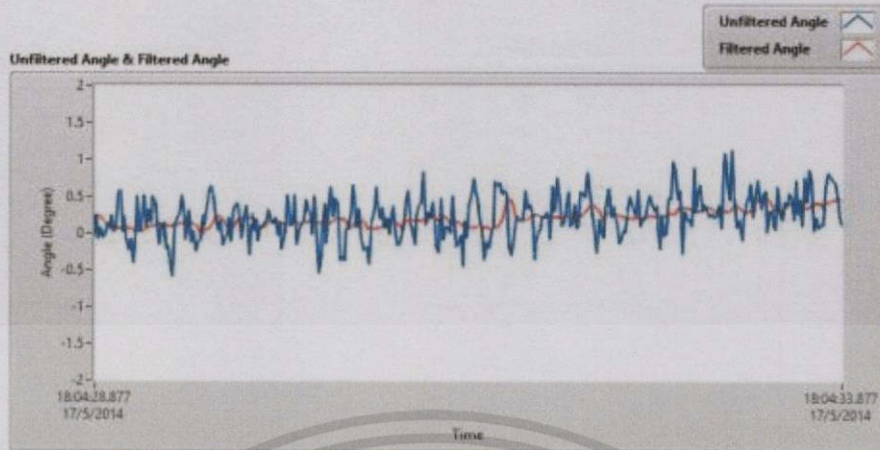
4.1 การทดลองวัดค่ามุมพิทช์ก่อนและหลังการหาค่ามัธยฐาน

ในการทดลองนี้เป็นการทดลองเพื่อเปรียบเทียบมุมพิทช์ที่ผ่านการหาค่ามัธยฐาน และมุมพิทช์ที่ไม่ได้ผ่านการหาค่ามัธยฐาน โดยการทดลองได้จับหุ่นยนต์จักรยานรักษาสมดุลตั้งฉากกับพื้นที่ตำแหน่ง 0 องศา ซึ่งลักษณะการทดลองเป็นดังรูปที่ 4.1



รูปที่ 4.1 จับหุ่นยนต์จักรยานรักษาสมดุลที่ตำแหน่ง 0 องศา

การอ่านค่ามุมพิทช์จะมีลักษณะดังรูปที่ 4.2

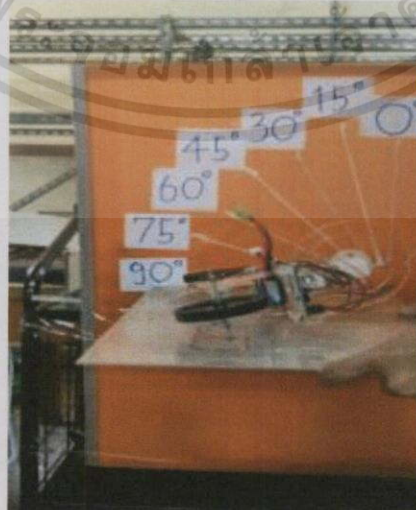


รูปที่ 4.2 การเปรียบเทียบการอ่านค่ามุมพิทช์

จากผลการทดลองพบว่ามุมพิทช์ที่ไม่ได้ผ่านการหาค่ามัธยฐาน ดังเส้นกราฟสีน้ำเงิน จะเกิดการแกว่งของสัญญาณเกิดขึ้น จึงไม่สามารถนำไปใช้งานได้ และมุมพิทช์ที่ผ่านการหาค่ามัธยฐาน ดังเส้นกราฟสีแดง จะมีสัญญาณที่ค่อนข้างใกล้เคียงกับมุมที่ทดลองคือ 0 องศา ซึ่งถือว่ายอมรับได้ และนำไปใช้งาน

4.2 การทดลองวัดค่ามุมพิทช์เมื่อเอียงหุ่นยนต์จักรยานรักษาสมดุล

ในการทดลองนี้เป็นการทดลองความแม่นยำของการวัดค่ามุมพิทช์ด้วยเซนเซอร์วัดความเร็ว โดยกำหนดให้มุมระนาบพื้นนั้นเป็น 90 องศา และเมื่อตั้งฉากกับพื้นให้เป็น 0 องศา ทำการเก็บข้อมูลดิบที่อ่านได้โดยการขยับมุมพิทช์ เพิ่มครั้งละ 15 องศา ครั้งละ 3 ข้อมูล และทำการหาค่าเฉลี่ยเป็นค่าที่เซนเซอร์อ่านได้ในกรณีการเปลี่ยนแปลงมุมแต่ละครั้ง ลักษณะการทดลองเป็นดังรูปที่ 4.3

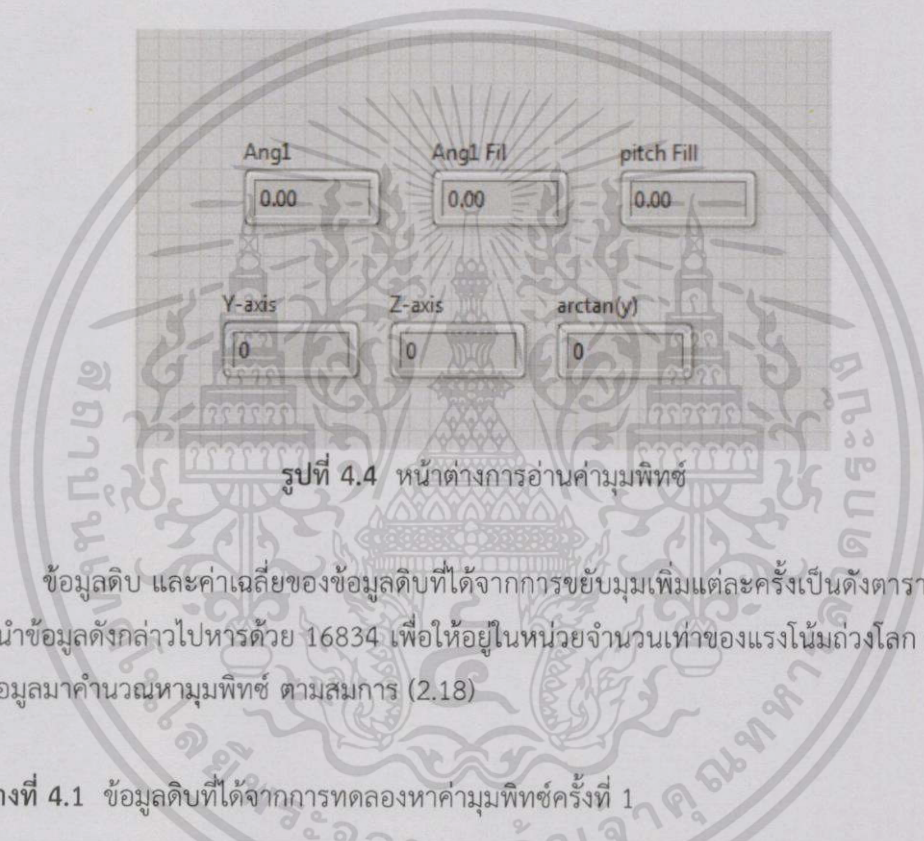


รูปที่ 4.3 การทดลองวัดค่ามุมพิทช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งหน้าตาการอ่านค่ามุมพิทช์จะมีบล็อกต่าง ๆ ดังนี้ ลักษณะดังรูปที่ 4.4

Ang1	คือ มุมในหน่วยเรเดียนที่ยังไม่ผ่านการหาค่ามัธยฐาน
Ang1 Fil	คือ มุมในหน่วยเรเดียนที่ผ่านการหาค่ามัธยฐาน
pitch Fill	คือ มุมในหน่วยองศาที่ผ่านการหาค่ามัธยฐาน
Y-axis	คือ ความเร่งในแกน Y ที่อ่านค่าได้
Z-axis	คือ ความเร่งในแกน Z ที่อ่านค่าได้
arctan(y)	คือ การคำนวณหาค่ามุมพิทช์ตามสมการ (2.18)



รูปที่ 4.4 หน้าตาการอ่านค่ามุมพิทช์

ข้อมูลดิบ และค่าเฉลี่ยของข้อมูลดิบที่ได้จากการขยับมุมเพิ่มแต่ละครั้งเป็นดังตารางที่ 4.1-4.4 นำข้อมูลดังกล่าวไปหารด้วย 16834 เพื่อให้อยู่ในหน่วยจำนวนเท่าของแรงโน้มถ่วงโลก และการนำข้อมูลมาคำนวณหามุมพิทช์ ตามสมการ (2.18)

ตารางที่ 4.1 ข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์ครั้งที่ 1

มุมทดสอบ	Ay	Ay(g)	Az	Az(g)	มุมพิทช์
0	92	0.005465	15512	0.921468	0.34
15	4776	0.283711	14676	0.871807	18.03
30	7936	0.471426	13372	0.794344	30.69
45	10752	0.638707	11448	0.680052	43.20
60	13968	0.829749	6968	0.413924	63.49
75	15460	0.918379	4608	0.273731	73.40
90	16264	0.966139	240	0.014256	89.15

ตารางที่ 4.2 ข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์ครั้งที่ 2

มุมทดสอบ	Ay	Ay(g)	Az	Az(g)	มุมพิทช์
0	36	0.00214	15644	0.92931	0.13
15	3940	0.23405	15112	0.89771	14.61
30	7916	0.44024	13412	0.79672	30.55
45	11188	0.66461	11324	0.67269	44.65
60	13948	0.82856	6948	0.41274	63.52
75	15708	0.93311	2820	0.16752	79.82
90	16184	0.96139	20	0.00119	89.93

ตารางที่ 4.3 ข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์ครั้งที่ 3

มุมทดสอบ	Ay	Ay(g)	Az	Az(g)	มุมพิทช์
0	180	0.01069	15596	0.92646	0.66
15	4184	0.24854	15032	0.89295	15.55
30	8276	0.49162	13220	0.78531	32.05
45	11272	0.66960	10924	0.64892	45.90
60	13976	0.83022	7088	0.42105	63.11
75	15428	0.91648	3428	0.20363	77.47
90	16236	0.96448	340	0.02020	88.80

ตารางที่ 4.4 ค่าเฉลี่ยข้อมูลดิบที่ได้จากการทดลองหาค่ามุมพิทช์

มุมทดสอบ	Ay	Ay(g)	Az	Az(g)	มุมพิทช์
0	102.66	0.00610	15584	0.92574	0.38
15	4300	0.25543	14940	0.88749	16.06
30	8042.66	0.47776	13334.66	0.79212	31.10
45	11070.66	0.65764	11232	0.66722	44.58
60	13964	0.82951	7001.33	0.41590	63.37
75	15532	0.92266	3618.66	0.21496	76.90
90	16228	0.96400	200	0.01188	89.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองพบว่าการอ่านค่าเซนเซอร์วัดความเร่ง โดยการหาค่ามัธยฐานของมุมพิทช์กับมุมที่เอียงหุ่นยนต์จักรยานรักษาสมดุลไว้มีความคลาดเคลื่อนเล็กน้อย และการทดลองอาจมีการผิดพลาดจากตัวผู้ทดลองเองด้วย ถือว่าค่าความผิดพลาดยอมรับได้ และนำเอาไปใช้งาน

4.3 การทดลองขับมอเตอร์เมื่อแปรค่าเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชัน

ในการทดลองนี้เป็นการทดลองขับมอเตอร์เมื่อแปรค่าเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชัน เพื่อหาเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันขั้นต่ำที่ทำให้มอเตอร์ขับงานหมุนได้ และความสัมพันธ์ระหว่างเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันต่อแรงดันเอาต์พุต โดยผลการทดลองแสดงดังตารางที่ 4.5

ตารางที่ 4.5 การแปรค่าเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันเทียบกับแรงดันเอาต์พุต

สัญญาณพัลส์วิธมอดูเลชัน (เปอร์เซ็นต์)	แรงดันเอาต์พุต (โวลต์)
100	2.12
90	1.97
80	1.82
70	1.60
60	1.39
50	1.09
40	0.60
30	0.29
20	0.09
10	0.04
0	0

จากผลการทดลองพบว่าหากจ่ายแรงดันไฟฟ้าด้วยเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันต่ำกว่า 30 เปอร์เซ็นต์ มอเตอร์จะยังไม่สามารถขับงานหมุนได้ ดังนั้นในการจ่ายแรงดันไฟฟ้าให้กับมอเตอร์จะต้องใช้เปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันที่มากกว่า 30 เปอร์เซ็นต์ขึ้นไป

4.4 การทดลองควบคุมหุ่นยนต์จักรยานรักษาสมดุล

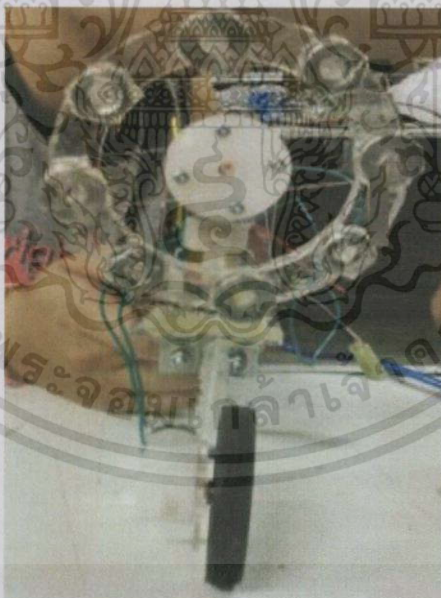
ในการทดลองนี้ได้ทดลองการสร้างสมดุลให้ตัวเองของหุ่นยนต์จักรยาน โดยเลือกใช้ตัวควบคุมแบบพีไอดีมาควบคุม

4.4.1 การปรับค่าอัตราการขยายเชิงสัดส่วน

การปรับค่าอัตราการขยายเชิงสัดส่วนจะแปรผันตรงกับสัญญาณความผิดพลาดของระบบ เมื่อสัญญาณความผิดพลาดมีค่ามาก การปรับค่าอัตราการขยายเชิงสัดส่วนก็จะมากตาม เมื่อการทดลองปรับอัตราการขยายเชิงสัดส่วนเท่ากับ 10 ที่การเอียงของหุ่นยนต์จักรยานรักษาสมดุล ที่ 5 องศา 10 องศา 15 องศา -5 องศา -10 องศา -15 องศา และปล่อยหุ่นยนต์จักรยานรักษาสมดุล สร้างสมดุลเองที่อัตราการขยายเชิงสัดส่วนเท่ากับ 40 จะได้ความสัมพันธ์ของมุมมอง และสัญญาณพัลส์วีธีมอดูเลชันตามผลการทดลองดังนี้

ที่มุม 5 องศา

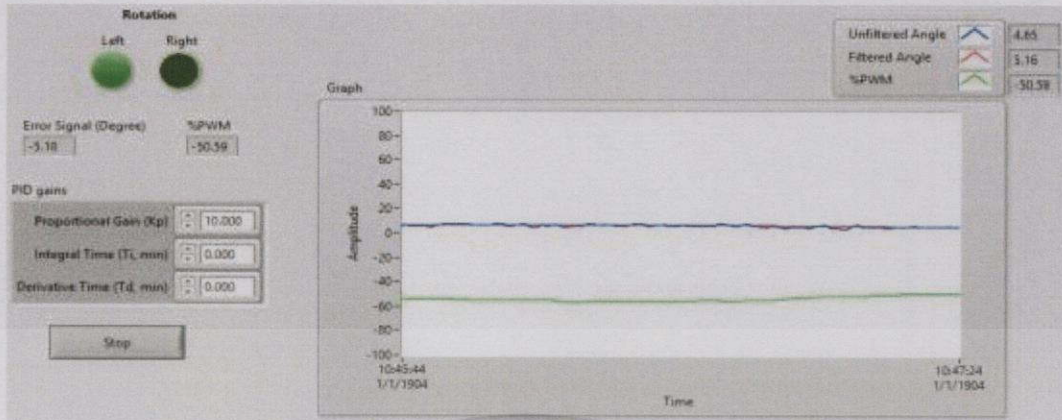
การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 5 องศา ดังรูปที่ 4.5



รูปที่ 4.5 การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 5 องศา

การแสดงผลผ่านหน้าต่างโปรแกรมจะเป็นดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 หน้าต่างโปรแกรมที่มุม 5 องศา

จากรูปที่ 4.6 เมื่อลองปรับค่าอัตราการขยายเชิงสัดส่วนเท่ากับ 10 ที่มุม 5.18 องศา พบว่าสัญญาณความผิดพลาดคือ -5.18 องศา การป้อนเปอร์เซ็นต์สัญญาณพัลส์วริธมอดูเลชันคือ -50.59% ซึ่งสอดคล้องกับสมการ (3.3) ดังเส้นกราฟสีเขียว โดยสั่งให้มอเตอร์ขับเคลื่อนไปทางซ้าย เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับมาสู่ตำแหน่งสมดุลคือ 0 องศา

ที่มุม 10 องศา

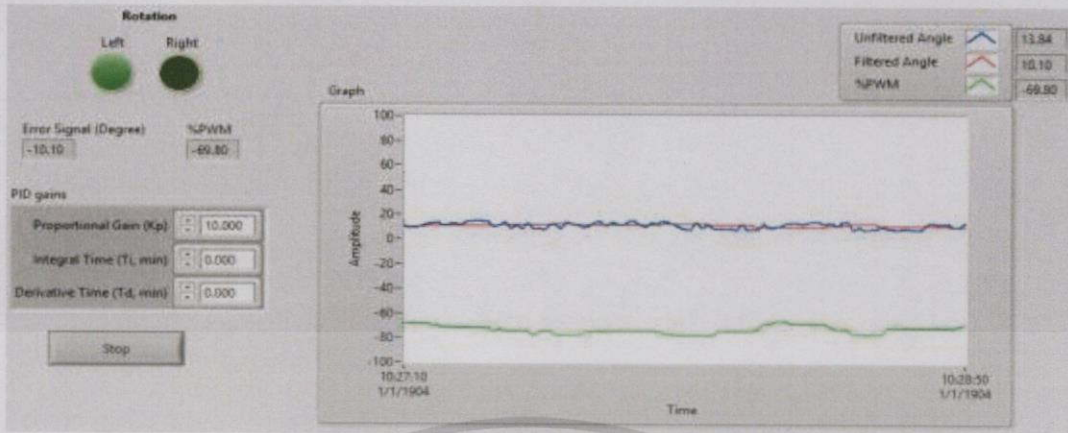
การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 10 องศา ดังรูปที่ 4.7



รูปที่ 4.7 การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 10 องศา

การแสดงผลผ่านหน้าต่างโปรแกรมจะเป็นดังรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าต่างโปรแกรมที่มุม 10 องศา

จากรูปที่ 4.8 เมื่อลองปรับค่าอัตราขยายเชิงสัดส่วนเท่ากับ 10 ที่มุม 10.10 องศา พบว่าสัญญาณความผิดพลาดคือ -10.10 องศา การป้อนเปอร์เซ็นต์สัญญาณพัลส์วิธิมอดูเลชันคือ -69.80% ซึ่งสอดคล้องกับสมการ (3.3) ตั้งเส้นกราฟสีเขียว โดยสั่งให้มอเตอร์ขับเคลื่อนไปทางซ้าย เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับมาสู่ตำแหน่งสมดุลคือ 0 องศา

ที่มุม 15 องศา

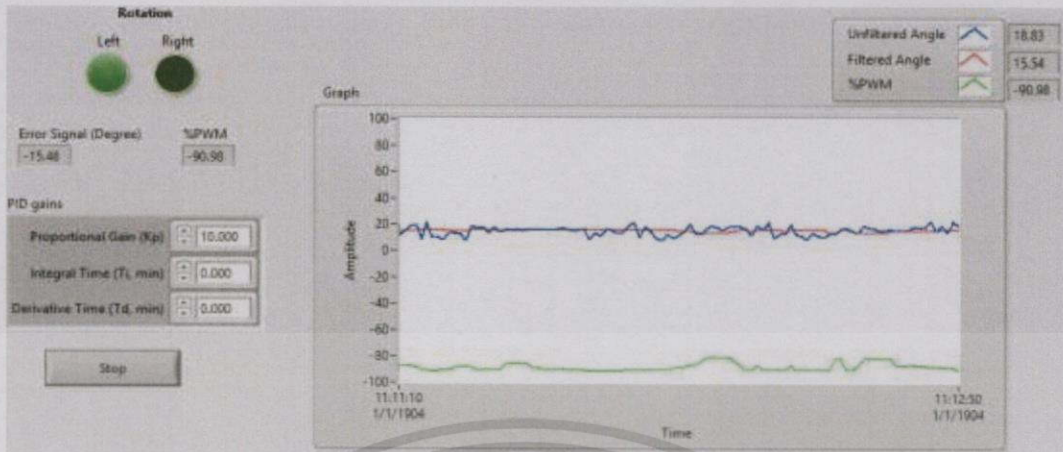
การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 15 องศา ดังรูปที่ 4.9



รูปที่ 4.9 การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม 15 องศา

การแสดงผลผ่านหน้าต่างโปรแกรมจะเป็นดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

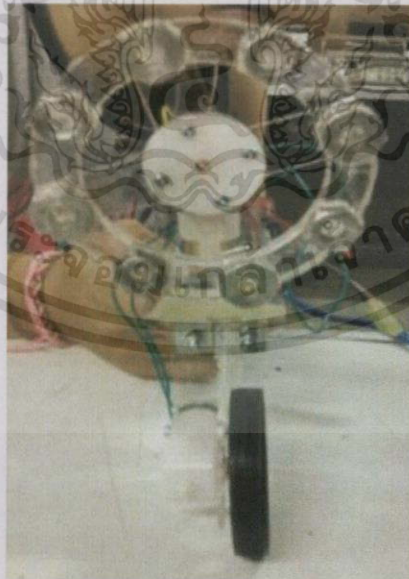


รูปที่ 4.10 หน้าต่างโปรแกรมที่มุม 15 องศา

จากรูปที่ 4.10 เมื่อลองปรับค่าอัตราขยายเชิงสัดส่วนเท่ากับ 10 ที่มุม 15.48 องศา พบว่าสัญญาณความผิดพลาดคือ -15.48 องศา การป้อนเปอร์เซ็นต์สัญญาณพัลส์ริชมอดูเลชันคือ -90.98% ซึ่งสอดคล้องกับสมการ (3.3) ดังเส้นกราฟสีเขียว โดยสั่งให้มอเตอร์ขับเคลื่อนไปทางซ้าย เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับมาสู่ตำแหน่งสมดุลคือ 0 องศา

ที่มุม -5 องศา

การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -5 องศา ดังรูปที่ 4.11



รูปที่ 4.11 การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -5 องศา

การแสดงผลผ่านหน้าต่างโปรแกรมจะเป็นดังรูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 หน้าต่างโปรแกรมที่มุม -5 องศา

จากรูปที่ 4.12 เมื่อลองปรับค่าอัตราขยายเชิงสัดส่วนเท่ากับ 10 ที่มุม -5.80 องศา พบว่าสัญญาณความผิดพลาดคือ 5.80 องศา การป้อนเปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลชันคือ 52.94% ซึ่งสอดคล้องกับสมการ (3.2) ดึงเส้นกราฟสีเขียว โดยสั่งให้มอเตอร์ขับเคลื่อนไปทางขวา เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับมาสู่ตำแหน่งสมดุลคือ 0 องศา

ที่มุม -10 องศา

การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -10 องศา ดังรูปที่ 4.13



รูปที่ 4.13 การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -10 องศา

การแสดงผลผ่านหน้าต่างโปรแกรมจะเป็นดังรูปที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

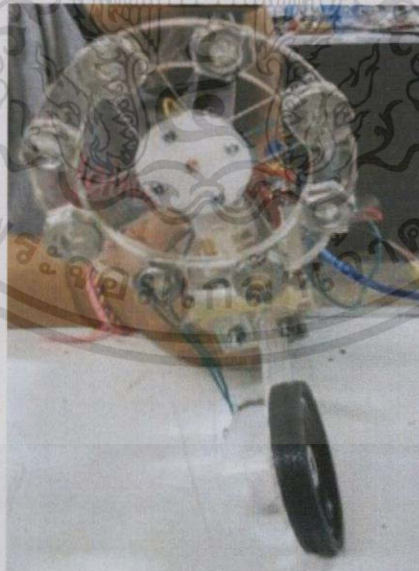


รูปที่ 4.14 หน้าต่างโปรแกรมที่มุม -10 องศา

จากรูปที่ 4.14 เมื่อลองปรับค่าอัตราขยายเชิงสัดส่วนเท่ากับ 10 ที่มุม -10.58 องศา พบว่าสัญญาณความผิดพลาดคือ 10.58 องศา การป้อนเบอร์เซ็นต์สัญญาณพัลส์วิธิมอดูเลขันคือ 71.76% ซึ่งสอดคล้องกับสมการ (3.2) ดังเส้นกราฟสีเขียว โดยสั่งให้มอเตอร์ขับเคลื่อนไปทางขวา เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับมาสู่ตำแหน่งสมดุลคือ 0 องศา

ที่มุม -15 องศา

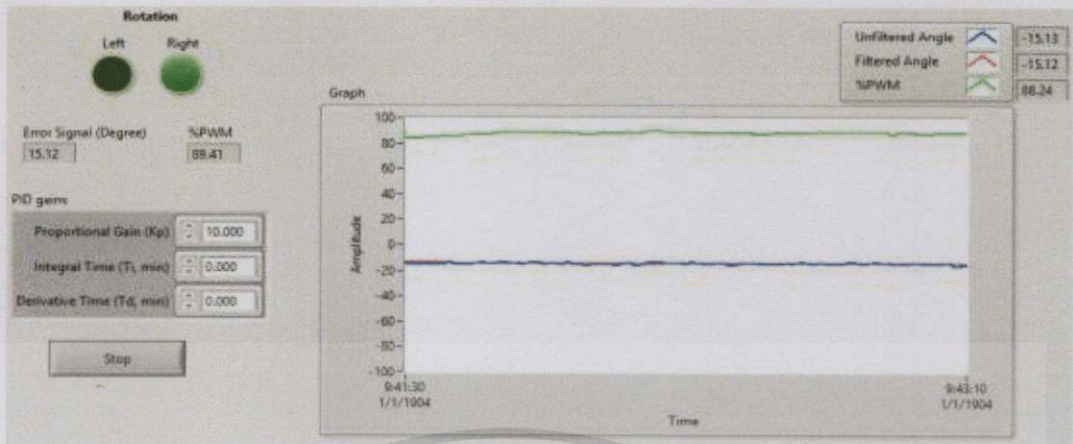
การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -15 องศา ดังรูปที่ 4.15



รูปที่ 4.15 การจับหุ่นยนต์จักรยานรักษาสมดุลที่มุม -15 องศา

การแสดงผลผ่านหน้าต่างโปรแกรมจะเป็นดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 หน้าต่างโปรแกรมที่มุม -15 องศา

จากรูปที่ 4.16 เมื่อลองปรับค่าอัตราการขยายเชิงสัดส่วนเท่ากับ 10 ที่มุม -15.12 องศา พบว่าสัญญาณความผิดพลาดคือ 15.12 องศา การป้อนเปอร์เซ็นต์สัญญาณพัลส์วิธิมอดูเลชันคือ 89.41% ซึ่งสอดคล้องกับสมการ (3.2) ดังเส้นกราฟสีเขียว โดยสั่งให้มอเตอร์ขับจานหมุนไปทางขวา เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับมาสู่ตำแหน่งสมดุลที่ 0 องศา

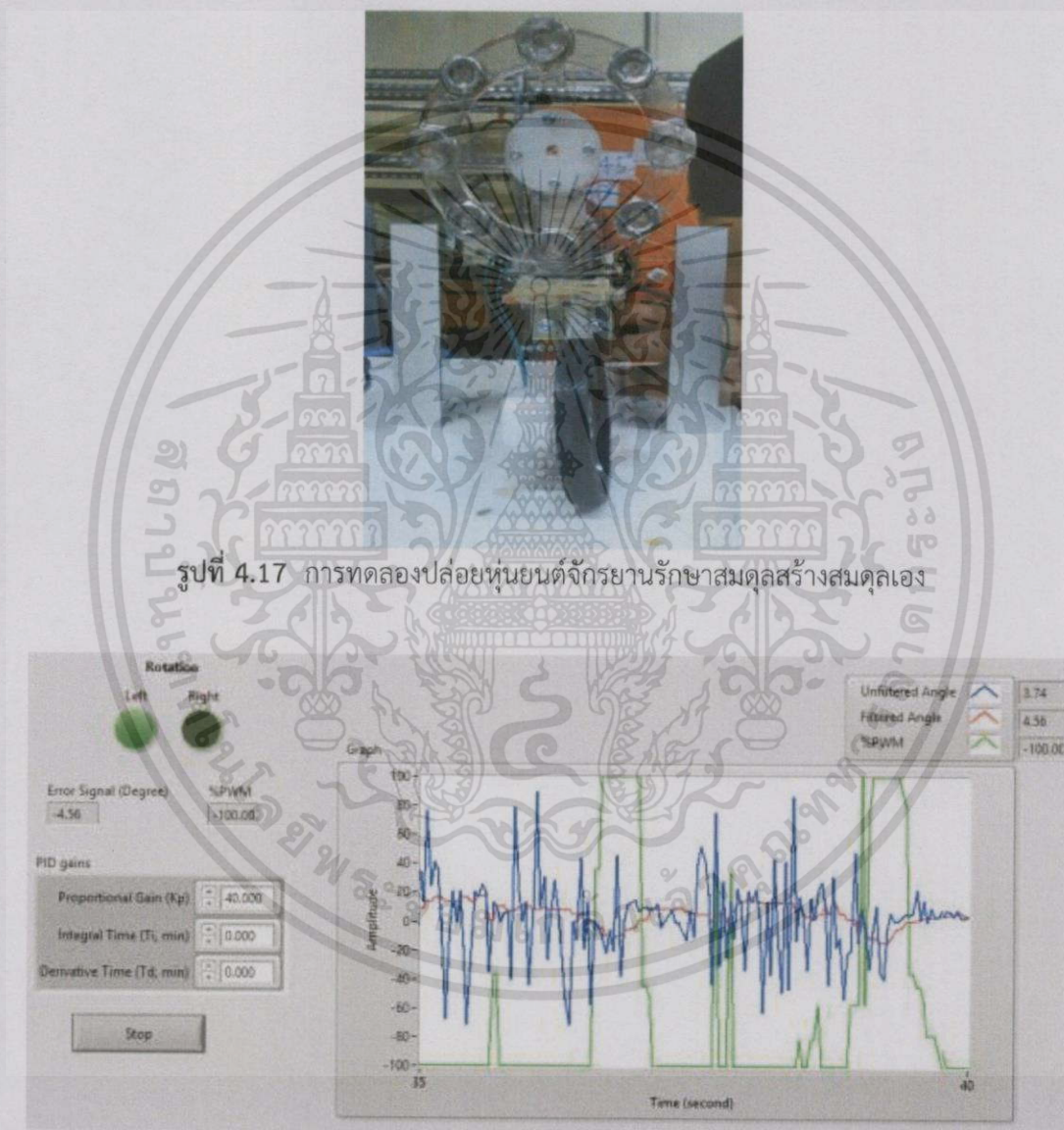
ปล่อยหุ่นยนต์จักรยานรักษาสมดุลสร้างสมดุลเอง

การปล่อยหุ่นยนต์จักรยานรักษาสมดุลสร้างสมดุลเองจะทดลองปรับค่าอัตราการขยายเชิงสัดส่วนเท่ากับ 40 เนื่องจากการทดลองก่อนหน้านี้ ซึ่งใช้ค่าอัตราการขยายเชิงสัดส่วนเท่ากับ 10 พบว่ามอเตอร์หมุนด้วยความเร็วรอบต่ำ เมื่อมุมพิทซ์น้อย ซึ่งไม่เพียงพอต่อการสร้างสมดุล นอกจากนั้นเพื่อป้องกันการล้มของหุ่นยนต์จักรยานรักษาสมดุล ซึ่งอาจทำให้อุปกรณ์เสียหายได้ จึงอาศัยหลักประคองทั้งสองฝั่ง เพื่อให้มุมพิทซ์ของหุ่นยนต์จักรยานรักษาสมดุลมีมุมพิทซ์จำกัดในย่าน $-15 \leq \theta \leq 15$ องศา โดยมีลักษณะการทดลองดังรูปที่ 4.17

เมื่อบันทึกผลการทดลองในกรณีที่มีมอเตอร์สามารถทำงานได้ตามปกติ ผลการทดลองสามารถบันทึกได้ดังหน้าต่างโปรแกรมในรูปที่ 4.18 จากรูปที่ 4.18 พบว่าเมื่อหุ่นยนต์จักรยานรักษาสมดุลเริ่มเอียงไปทางซ้าย หรือค่าความผิดพลาดเป็นบวก การป้อนเปอร์เซ็นต์สัญญาณพัลส์วิธิมอดูเลชันจะมีค่าเป็นลบ หรือส่งผลให้จานหมุนหมุนไปทางขวา เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับสู่จุดสมดุล อย่างไรก็ตามเนื่องจากเปอร์เซ็นต์สัญญาณพัลส์วิธิมอดูเลชันมีค่ามาก ทำให้หุ่นยนต์จักรยานรักษาสมดุลเอียงไปทางขวาเกินจุดสมดุลตามแรงบิดที่เกิดจากจานหมุน ทำให้ค่าความผิดพลาดเป็นลบ และเปอร์เซ็นต์สัญญาณพัลส์วิธิมอดูเลชันกลับค่าเป็นบวก ซึ่งทำให้หุ่นยนต์จักรยานรักษาสมดุลพยายามกลับสู่จุดสมดุล แต่ทำให้หุ่นยนต์จักรยานรักษาสมดุลเอียงไปทางซ้ายเกินจุดสมดุล และผลตอบสนองมีลักษณะเกิดการเคลื่อนที่สลับเอียงในมุมบวกและมุมลบดังข้างต้นอยู่ตลอดเวลาไม่สามารถ

ไม่สามารถรักษาสมดุลที่มุม 0 องศาได้ นอกจากนั้นสังเกตว่าหุ่นยนต์จักรยานรักษาสมดุลมีแนวโน้มเอียงไปทางขวา เนื่องจากน้ำหนักของหุ่นยนต์จักรยานรักษาสมดุลจะมีน้ำหนักด้านขวามาก

จากการทดลองนี้ในบางการทดลองพบว่ามอเตอร์ไม่สามารถทำงานได้ตามคำสั่งโปรแกรม โดยมอเตอร์ไม่สามารถขับเคลื่อนได้ ถึงแม้เปอร์เซ็นต์สัญญาณพัลส์วริมอดูเลขันมีค่าสูงสุด ส่งผลให้จักรยานรักษาสมดุลไม่ทำงาน โดยหยุดนิ่งชนกับเหล็กประคอง

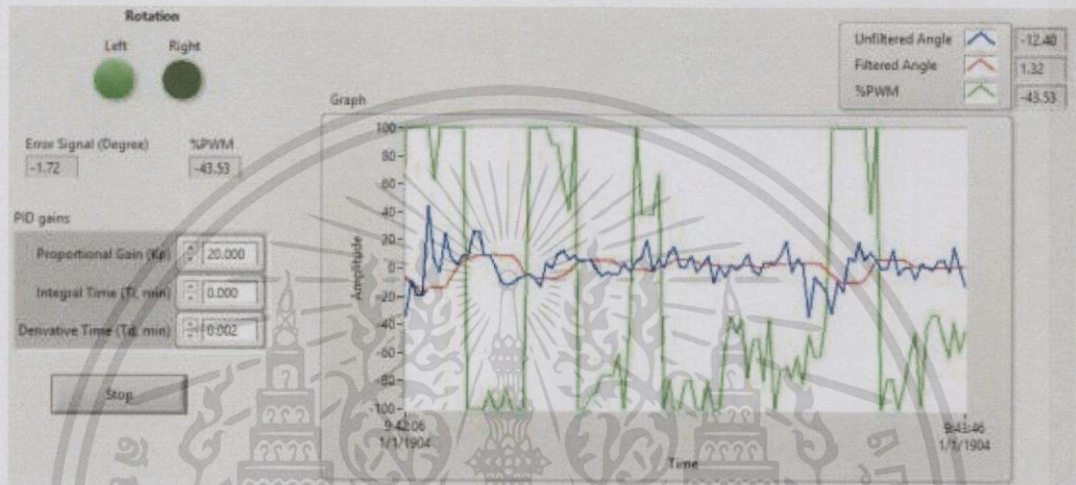


รูปที่ 4.18 หน้าต่างโปรแกรมเมื่อปล่อยหุ่นยนต์จักรยานรักษาสมดุลอาศัยตัวควบคุมแบบพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การปรับค่าอัตราขยายเชิงอนุพันธ์

การทดลองนี้ได้ทำการทดลองเช่นเดียวกับรูปที่ 4.17 ซึ่งทำการทดลองโดยเพิ่มการควบคุมแบบปรับค่าอัตราขยายเชิงอนุพันธ์เท่ากับ 0.002 โดยสัญญาณควบคุมแปรผันตรงกับอัตราการเปลี่ยนแปลงค่าความผิดพลาด เนื่องจากการปรับค่าอัตราขยายเชิงอนุพันธ์สามารถช่วยให้หุ่นยนต์จักรยานรักษาสมุมมีแนวโน้มต่อการตอบสนองของการเปลี่ยนแปลงของมุมพิทช์เร็วขึ้น



รูปที่ 4.19 หน้าต่างโปรแกรมเมื่อปล่อยหุ่นยนต์จักรยานรักษาสมุมอาศัยตัวควบคุมแบบพีดี

เมื่อบันทึกผลการทดลองในกรณีที่มีมอเตอร์สามารถทำงานได้ตามปกติ ผลการทดลองสามารถบันทึกได้ดังหน้าต่างโปรแกรมในรูปที่ 4.19 จากรูปที่ 4.19 พบว่าเมื่อหุ่นยนต์จักรยานรักษาสมุมเริ่มเอียงไปด้านซ้าย หรือค่าความผิดพลาดเป็นบวก การป้อนเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันจะมีค่าเป็นลบ หรือส่งผลให้จานหมุนหมุนไปทางขวา และป้อนเปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันมากเมื่อมีการเปลี่ยนแปลงมุมพิทช์เพียงเล็กน้อยทำให้หุ่นยนต์จักรยานรักษาสมุมมีแนวโน้มเอียงกลับมาทางซ้ายได้เร็วกว่าการปรับค่าอัตราขยายเชิงสัดส่วนเพียงค่าเดียว

จากการทดลองนี้ในบางการทดลองพบว่ามอเตอร์ไม่สามารถทำงานได้ตามคำสั่งโปรแกรม โดยมอเตอร์ไม่สามารถขับจานหมุนได้ ถึงแม้เปอร์เซ็นต์สัญญาณพัลส์วิธมอดูเลชันมีค่าสูงสุด ส่งผลให้จักรยานรักษาสมุมไม่ทำงาน โดยหยุดนิ่งชนกับเหล็กประคอง

สรุปและวิเคราะห์ปัญหา

5.1 สรุปผลการดำเนินงาน

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการศึกษา และการออกแบบหุ่นยนต์จักรยานรักษาสมดุล ซึ่งอาศัยหลักการของแรงบิดของจานหมุนในการสร้างสมดุล ต่อมาได้ออกแบบโครงสร้างหุ่นยนต์จักรยานรักษาสมดุลที่สร้างจากแผ่นอะคริลิก จากนั้นได้ออกแบบวงจรที่ใช้สำหรับการรับค่าเซนเซอร์ที่ใช้วัดค่าความเร่ง และวงจรขับเคลื่อน เพื่อไปขับจานหมุน หลังจากนั้นได้ออกแบบโปรแกรมเพื่อควบคุมการสร้างสมดุลของหุ่นยนต์จักรยาน ซึ่งกำหนดค่าอินพุตอ้างอิงของระบบเป็นตำแหน่งของรถที่ 0 องศา โดยค่าป้อนกลับเป็นมุมพิทซ์ของหุ่นยนต์จักรยานรักษาสมดุล และใช้ตัวควบคุมแบบพีไอดีในการควบคุม ซึ่งเลือกใช้แค่การปรับเพิ่มอัตราขยายเชิงสัดส่วนและอัตราขยายเชิงอนุพันธ์ จากการทดลองพบว่าระบบที่ได้ออกแบบสามารถควบคุมทิศทางการสร้างสมดุลได้ เช่น เมื่อหุ่นยนต์เอียงไปทางซ้ายก็สร้างแรงบิดไปทางซ้าย เพื่อให้หุ่นยนต์จักรยานรักษาสมดุลเอียงกลับมาทางขวา อย่างไรก็ตามหุ่นยนต์จักรยานรักษาสมดุลยังไม่สามารถรักษาสมดุลได้

5.2 ปัญหาและอุปสรรค

1. การประกอบกันระหว่างฐานกลางกับแผ่นรับน้ำหนักมอเตอร์นั้นไม่แข็งแรงพอ เนื่องจากแผ่นรับน้ำหนักมอเตอร์ต้องรับน้ำหนักของมอเตอร์ที่ต่อกับล้อด้านหน้า ซึ่งแผ่นรับน้ำหนักมีขนาดบางมากคือ 2 มิลลิเมตร ทำให้เกิดการหักของสลักลักษณะตัวแอล โดยการแก้ปัญหาของคณะผู้จัดทำคือนำฉากเหล็กขนาด 20*20 มิลลิเมตร มารับน้ำหนักของมอเตอร์ที่ต่อกับล้อด้านหน้าแทน โดยนำมาติดตั้งระหว่างฐานกลางกับแผ่นรับน้ำหนักมอเตอร์

2. การเลือกใช้ไอซี L293D ในครั้งแรกไม่เหมาะกับมอเตอร์ที่จะมาใช้ขับจานหมุน เนื่องจากกระแสที่ออกมาจากไอซีนี้น้อยเกินไป ทำให้จานหมุนนั้นหมุนตอบสนองต่อการสร้างสมดุลไม่ได้ โดยการแก้ปัญหาของคณะผู้จัดทำคือ เปลี่ยนมาใช้ไอซี L298N ซึ่งให้กระแสออกมาได้มากกว่า L293D สามารถขับจานหมุนให้เกิดแรงบิด เพื่อสร้างสมดุลให้หุ่นยนต์จักรยานรักษาสมดุล

3. การเลือกใช้มอเตอร์ไฟฟ้าครั้งแรกนั้นไม่เหมาะกับการมาขับจานหมุน เนื่องจากมอเตอร์มีความเร็วรอบต่ำ และแรงบิดไม่เพียงพอต่อกันสร้างสมดุลของหุ่นยนต์จักรยานรักษาสมดุล โดยการแก้ปัญหาของคณะผู้จัดทำ คือเปลี่ยนมอเตอร์มาเป็นมอเตอร์ความเร็วรอบสูงแทน

4. การใช้โปรแกรม LabVIEW ควบคุมหุ่นยนต์จักรยานรักษาสมดุสิตเกิดความผิดพลาดจากการอ่านข้อมูลเซนเซอร์ซึ่งไม่ทันต่อการประมวลผล และทำการควบคุม ซึ่งอาจจะเกิดจากการเชื่อมต่อระหว่างโปรแกรม LabVIEW กับบอร์ด Arduino Uno R3

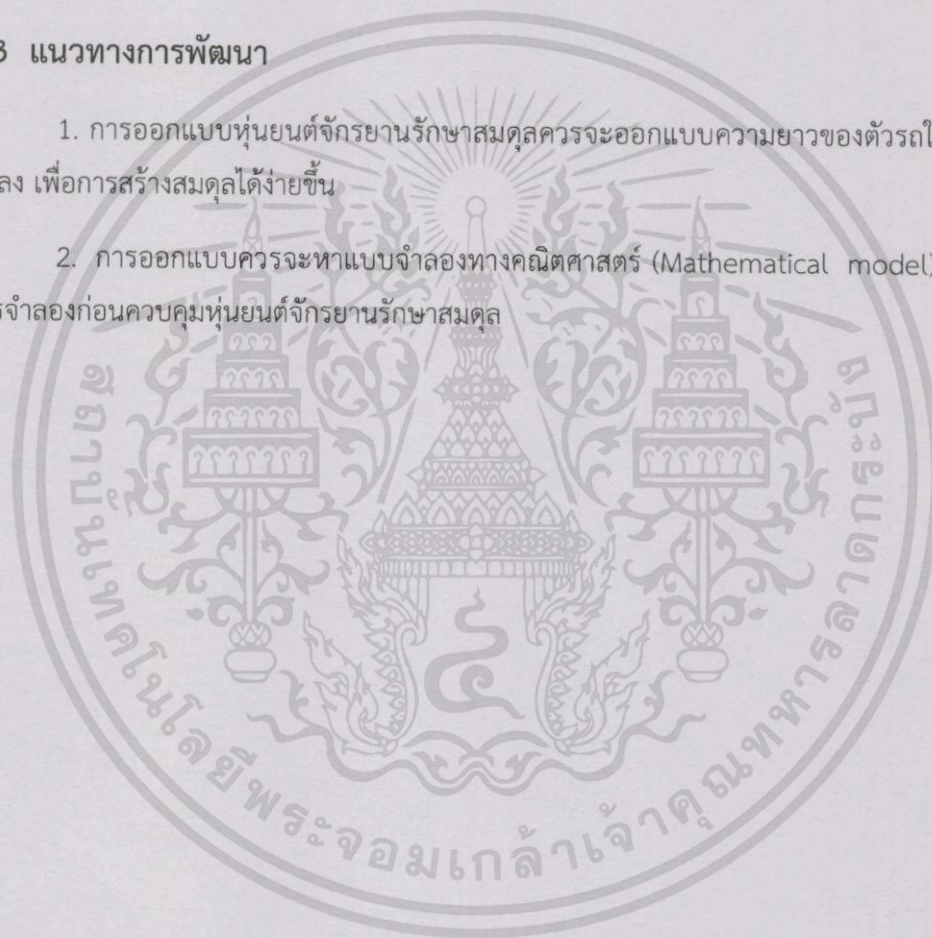
5. การทำงานของมอเตอร์ก็ไม่สามารถทำงานได้ตามคำสั่งโปรแกรมในบางครั้ง โดยมอเตอร์ไม่สามารถขับเคลื่อนได้ ถึงแม้เปอร์เซ็นต์สัญญาณพัลส์วามอเตอร์จะมีค่าสูงสุด

6. หุ่นยนต์จักรยานมีรูปร่างเอียงและยาวส่งผลให้การควบคุมทำได้ยาก

5.3 แนวทางการพัฒนา

1. การออกแบบหุ่นยนต์จักรยานรักษาสมดุสิตควรออกแบบความยาวของตัวรถให้มีขนาดสั้นลง เพื่อการสร้างสมดุสิตได้ง่ายขึ้น

2. การออกแบบควรหาแบบจำลองทางคณิตศาสตร์ (Mathematical model) และทำการจำลองก่อนควบคุมหุ่นยนต์จักรยานรักษาสมดุสิต



เอกสารอ้างอิง

- [1] Hyun-Woo Kim, Jae-Won An, Han Dong Yoo, Jang-Myung Lee. "Balancing Control of Bicycle Robot Using PID Control." **Proceeding of International Conference on Control.**, no.13, Oct.2013, pp. 145-147.
- [2] Katsuhiko Ogata. **Modern Control Engineering**, 4th ed., Prentice-Hall, New Jersey, USA, 2002.
- [3] **LabVIEW Interface for Arduino** [online] [cited 2014 May 22].
Available from: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/209835>
- [4] Javier E. Hasbun. **Classical Mechanics with Matlab Applications**, 1th ed., Massachusetts : Jones and bartlett publishers. 2008.
- [5] **MPU6050 Datasheet** [online] [cited 2014 May 9].
Available from: <http://www.alldatasheet.com/datasheetpdf/pdf/517744/ETC1/MPU-6050.html>
- [6] **L298N Datasheet** [online] [cited 2014 May 9].
Available from: <http://www.alldatasheet.com/view.jsp?Searchword=L298n>
- [7] **Arduino Uno R3** [online] [cited 2014 May 9].
Available from: <http://arduino.cc/en/Main/arduinoBoardUno>

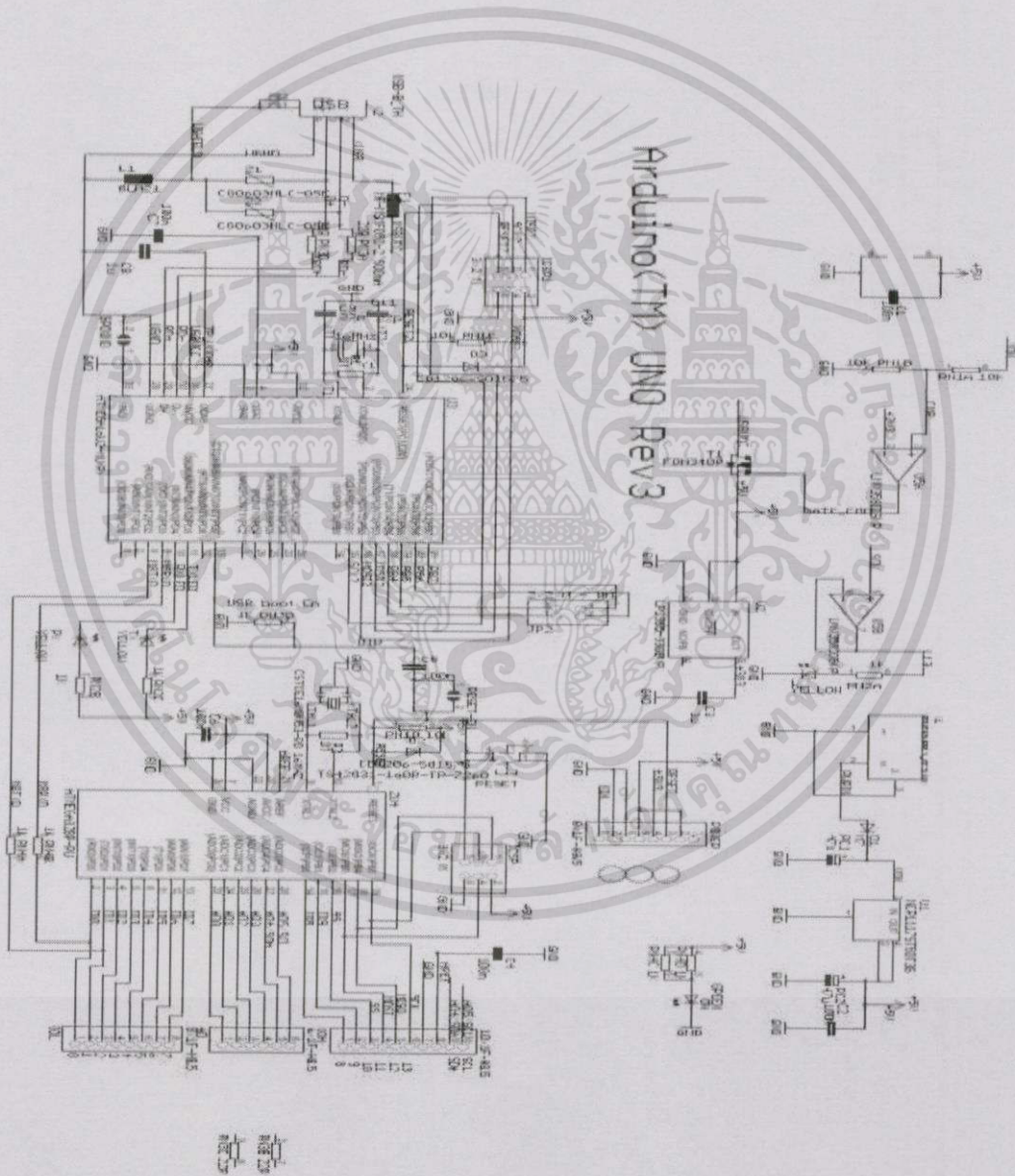


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ก.1 แผงผังบอร์ด Arduino Uno R3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 เอกสารคู่มือการใช้งานไมโครคอนโทรลเลอร์ Atmega328

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 μ A
 - Power-save Mode: 0.75 μ A (Including 32 kHz RTC)

1. Pin Configurations



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC} .

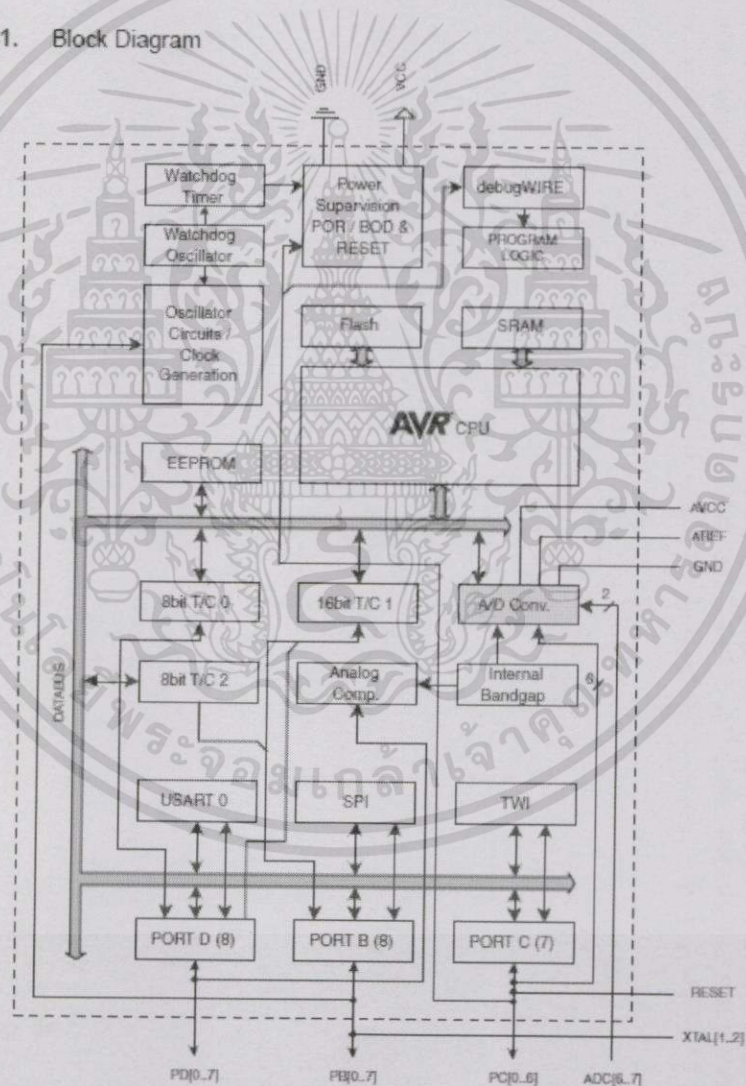
1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

Figure 2-1. Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3 เอกสารคู่มือการใช้งาน L298N

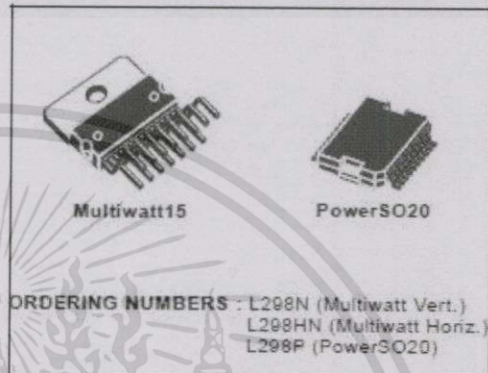
ไอซี L298N เป็นไอซีสำหรับการหมุนแบบสองทิศทางของมอเตอร์ไฟฟ้า มีรายละเอียดดังนี้

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

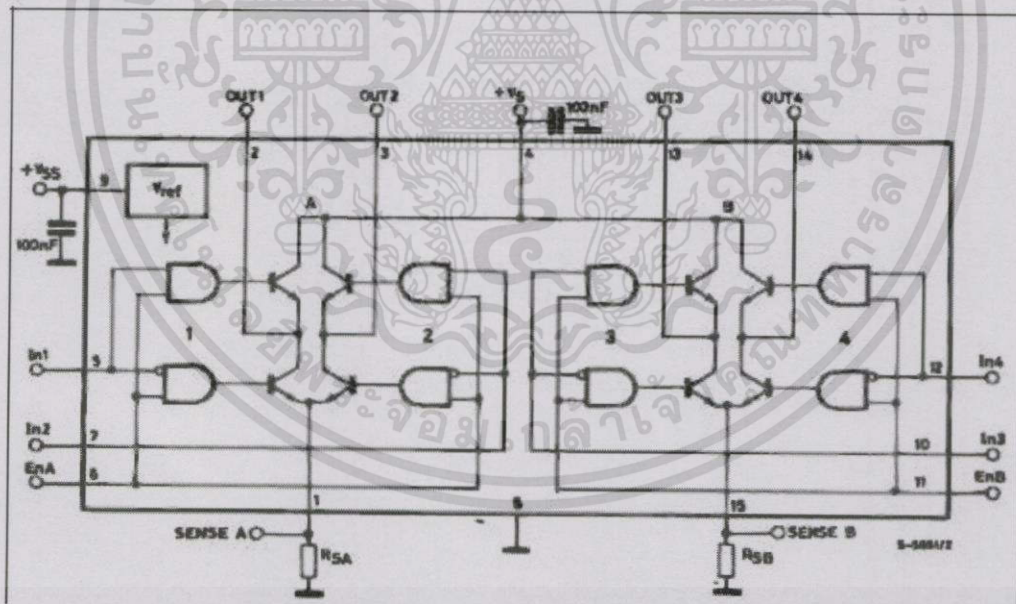
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

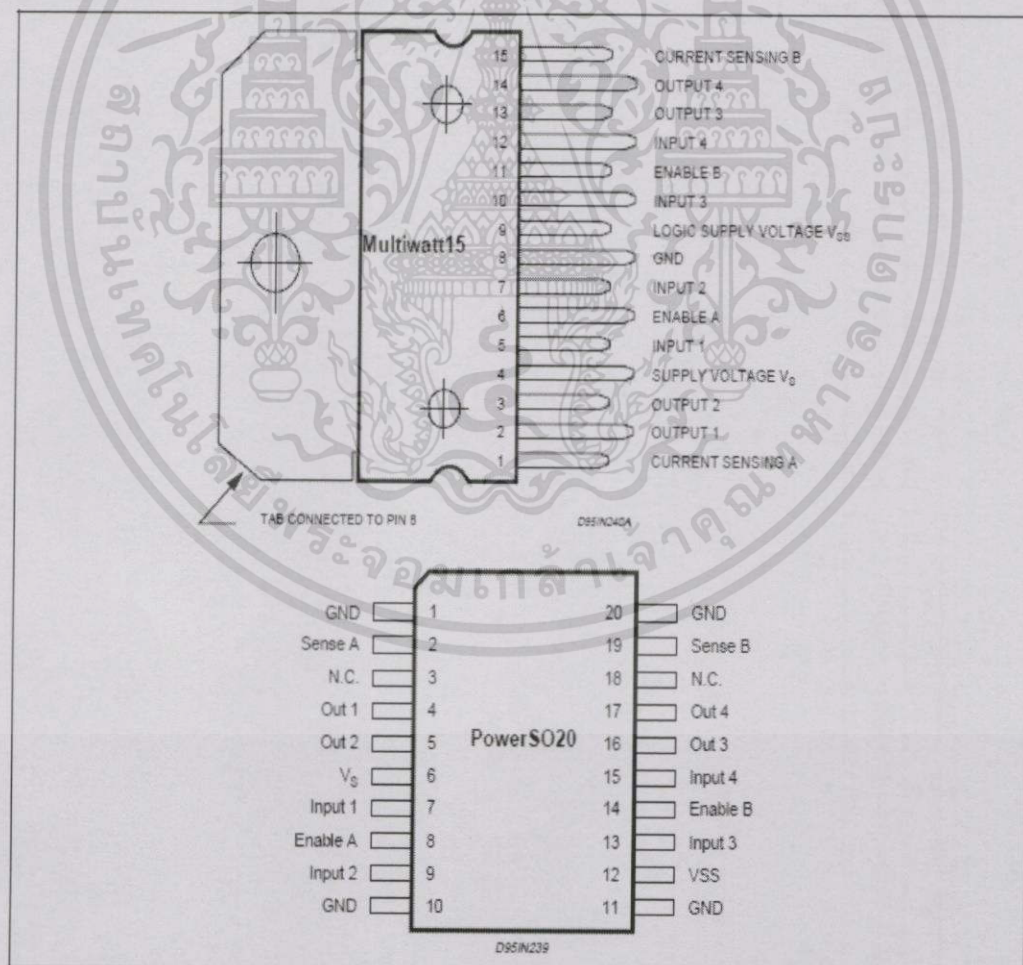
BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel) - Non Repetitive ($t = 100\mu s$) - Repetitive (80% on -20% off, $t_{on} = 10ms$) - DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _s	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ก.4 เอกสารคู่มือการใช้งาน MPU6050

เซนเซอร์ MPU6050 เป็นเซนเซอร์สำหรับตรวจวัดค่าความเร่ง มีรายละเอียดดังนี้

5 Features**5.1 Gyroscope Features**

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/\text{sec}$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5 μ A
- Factory calibrated sensitivity scale factor
- User self-test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500 μ A
- Low power accelerometer mode current: 10 μ A at 1.25Hz, 20 μ A at 5Hz, 60 μ A at 20Hz, 110 μ A at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

5.3 Additional Features

The MPU-60X0 includes the following additional features:

- 9-Axis MotionFusion by the on-chip Digital Motion Processor (DMP)
- Auxiliary master I²C bus for reading data from external sensors (e.g., magnetometer)
- 3.9mA operating current when all 6 motion sensing axes and the DMP are enabled
- VDD supply voltage range of 2.375V-3.46V
- Flexible VLOGIC reference voltage supports multiple I²C interface voltages (MPU-6050 only)
- Smallest and thinnest QFN package for portable devices: 4x4x0.9mm
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- 10,000 g shock tolerant
- 400kHz Fast Mode I²C for communicating with all registers
- 1MHz SPI serial interface for communicating with all registers (MPU-6000 only)
- 20MHz SPI serial interface for reading sensor and interrupt registers (MPU-6000 only)

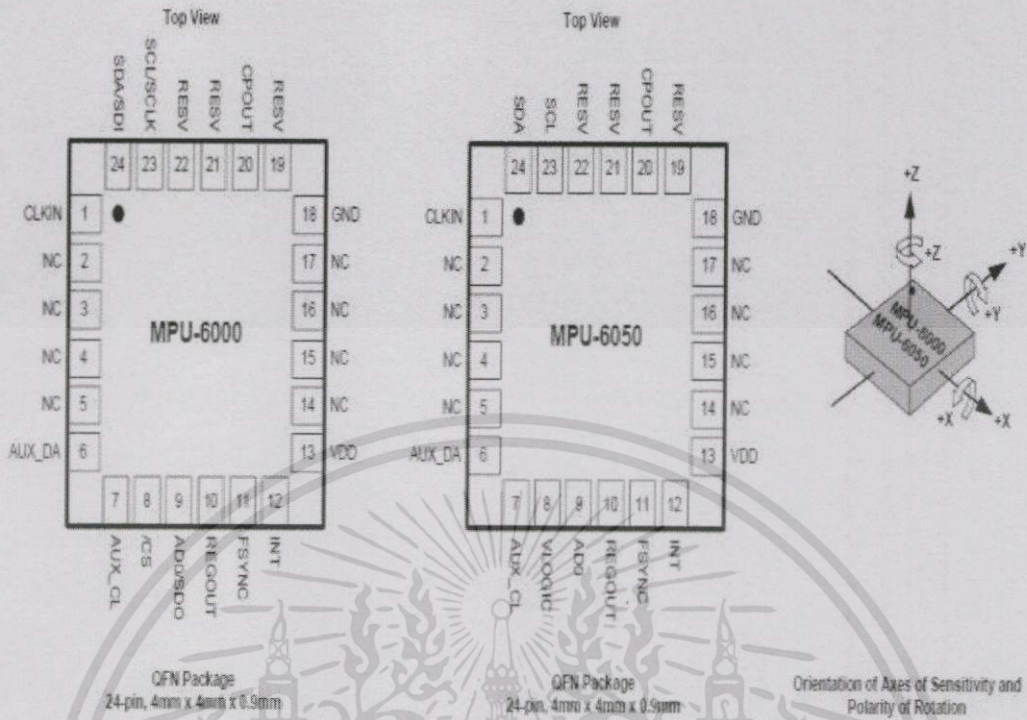
5.4 MotionProcessing

- Internal Digital Motion Processing™ (DMP™) engine supports 3D MotionProcessing and gesture recognition algorithms
- The MPU-60X0 collects gyroscope and accelerometer data while synchronizing data sampling at a user defined rate. The total dataset obtained by the MPU-60X0 includes 3-Axis gyroscope data, 3-Axis accelerometer data, and temperature data. The MPU's calculated output to the system processor can also include heading data from a digital 3-axis third party magnetometer.
- The FIFO buffers the complete data set, reducing timing requirements on the system processor by allowing the processor burst read the FIFO data. After burst reading the FIFO data, the system processor can save power by entering a low-power sleep mode while the MPU collects more data.
- Programmable interrupt supports features such as gesture recognition, panning, zooming, scrolling, tap detection, and shake detection
- Digitally-programmable low-pass filters
- Low-power pedometer functionality allows the host processor to sleep while the DMP maintains the step count.

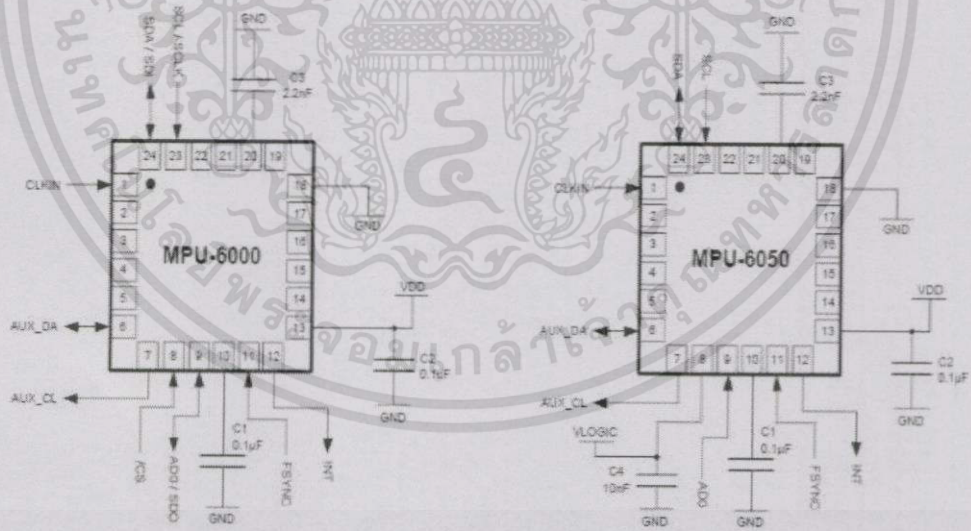
5.5 Clocking

- On-chip timing generator $\pm 1\%$ frequency variation over full temperature range
- Optional external clock inputs of 32.768kHz or 19.2MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



7.2 Typical Operating Circuit



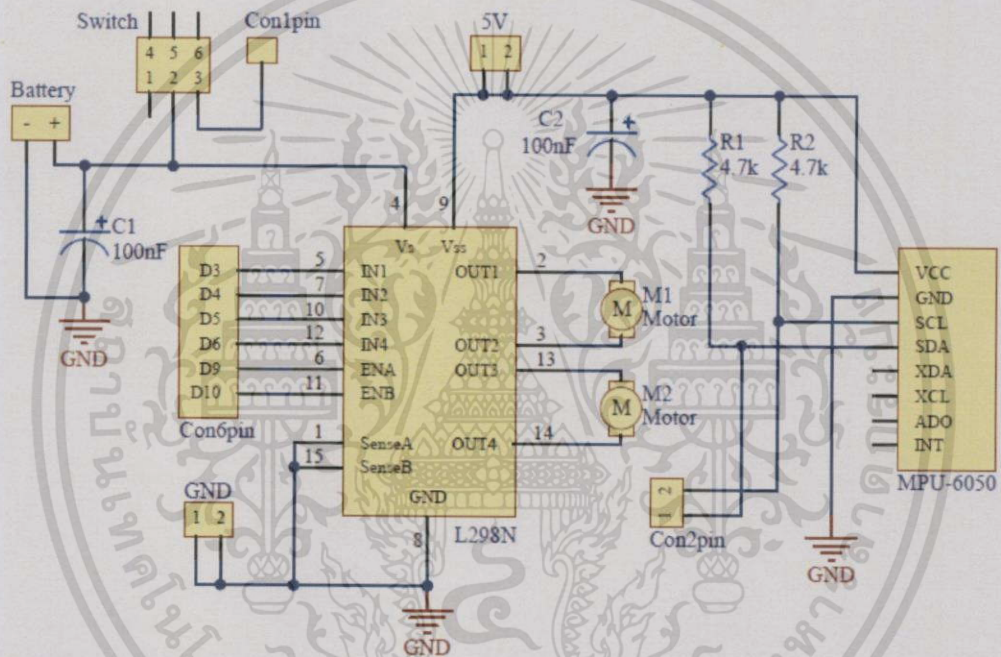
Typical Operating Circuits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

แผงผังอุปกรณ์และการเชื่อมต่อวงจร

ข.1 แผงผังวงจรของหุ่นยนต์จักรยานรักษาสมดุล



การเชื่อมต่อแผงวงจรกับบอร์ด Arduino Uno R3
MPU6050 จะเชื่อมต่อโดยแยกออกมาเป็น Con2pin ของขา SCL กับ SDA ไปเชื่อมต่อกับ
ขา A4, A5 ของบอร์ด Arduino Uno R3

IC298N จะเชื่อมต่อโดยแยกออกมาเป็น Con6pin ของขา 5, 7, 10, 12, 6 และ 11 ไป
เชื่อมต่อกับขาติจิตอล PWM ของบอร์ด Arduino Uno R3 ที่ D3,D4,D5,D6,D9 และ D10

IC298N จะเชื่อมต่อกับมอเตอร์ไฟฟ้า 2 ตัว ที่ขา 2, 3, 13, และ 14

ช่อง GND จะเชื่อมต่อกับบอร์ด Arduino Uno R3 ที่ขา GND

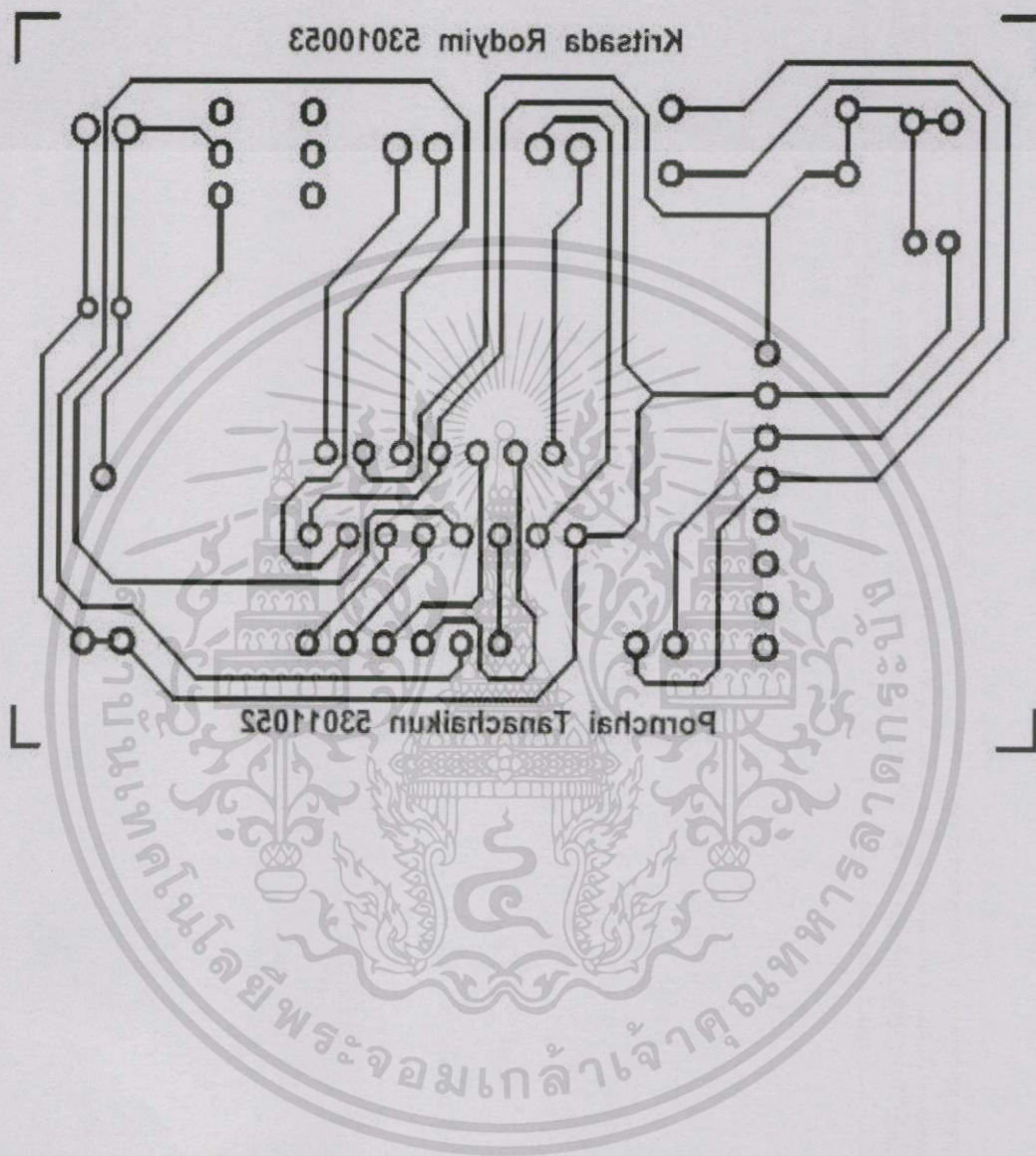
ช่อง Battery จะเชื่อมต่อกับแบตเตอรี่

ช่อง 5V จะเชื่อมต่อกับไฟเลี้ยงจากช่อง 5V จากบอร์ด Arduino Uno R3

Con1pin ออกแบบไว้จะใช้ไฟเลี้ยงจากบอร์ด Arduino Uno R3 (ไม่ได้ใช้งาน)

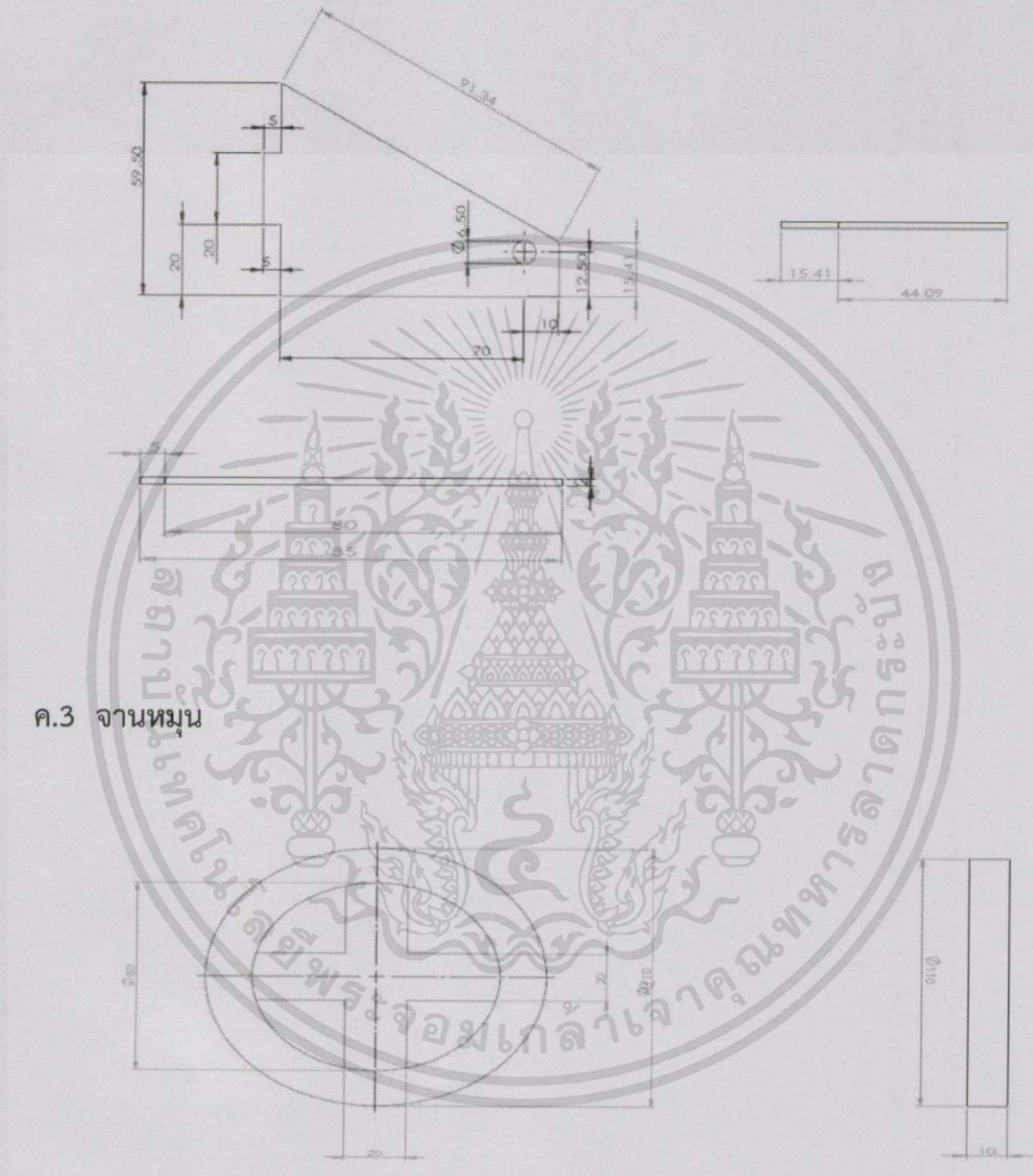
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 ลายวงจรสำหรับแผ่นวงจรพิมพ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

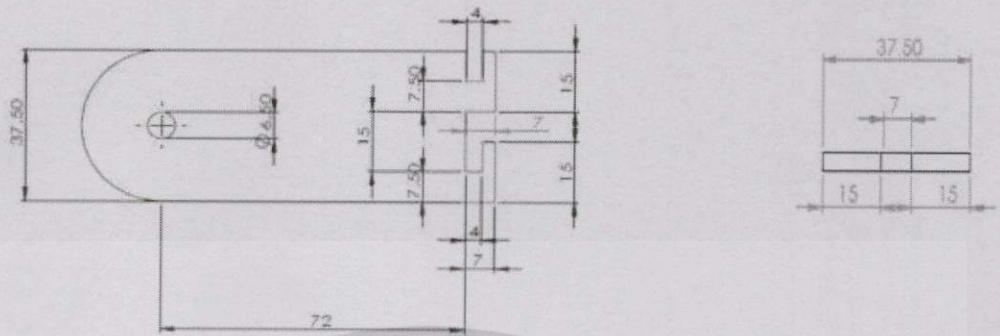
ค.2 แผ่นยึดมอเตอร์กับล้อหน้า



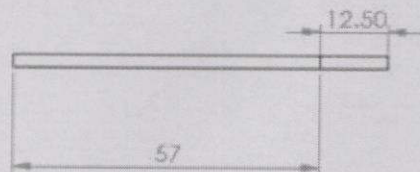
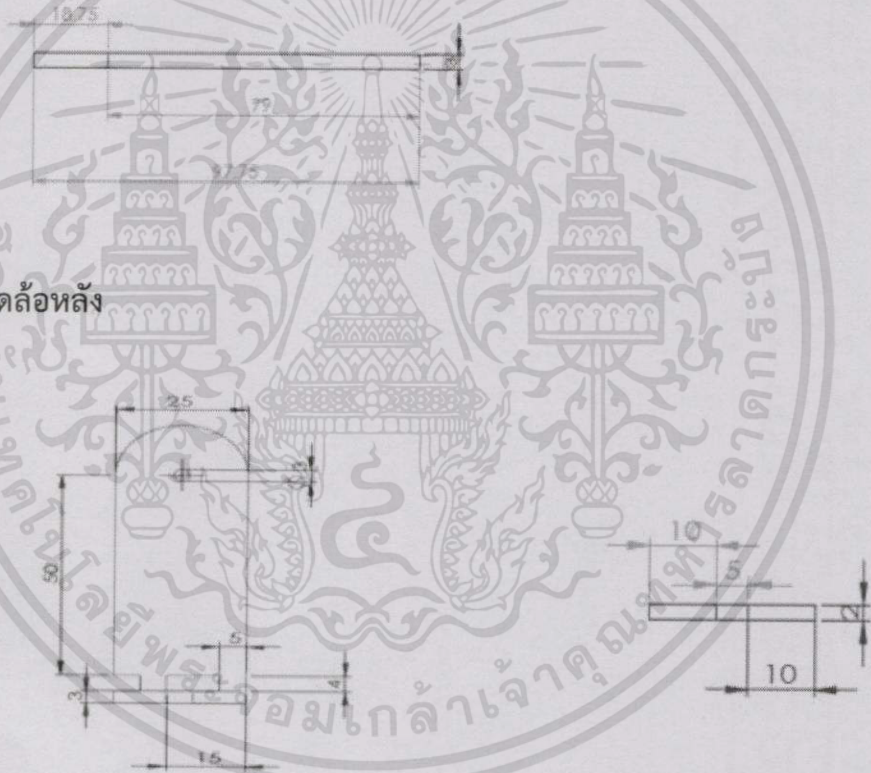
ค.3 งานหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.4 แผ่นยึดมอเตอร์

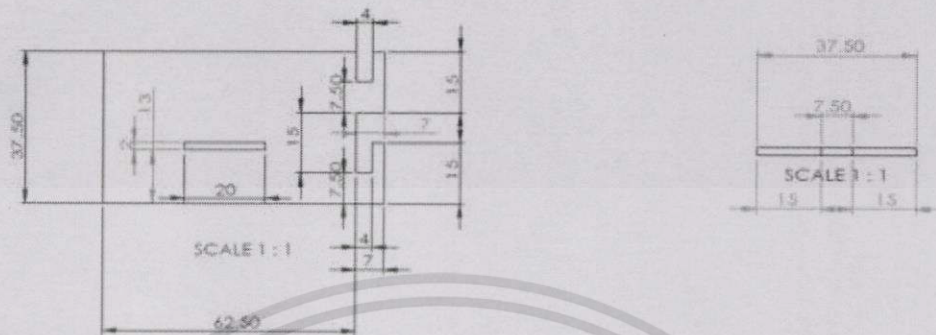


ค.5 ก้านยึดล้อหลัง

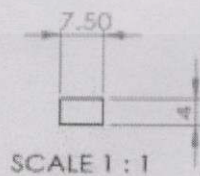
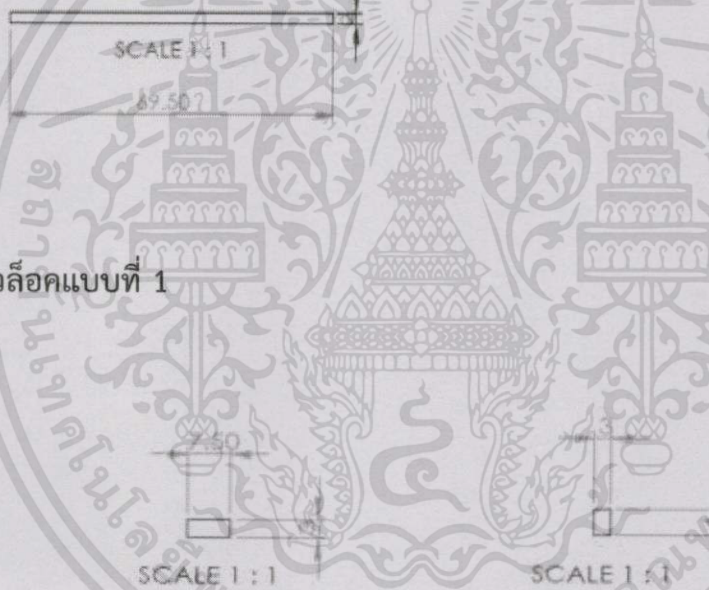


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.6 แผ่นรับน้ำหนักมอเตอร์

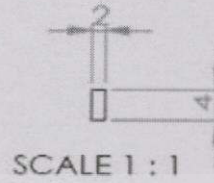
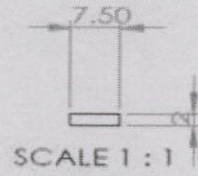


ค.7 ตัวล็อคแบบที่ 1

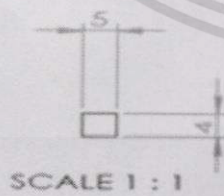
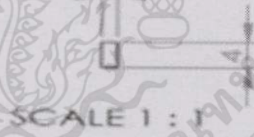
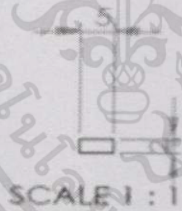
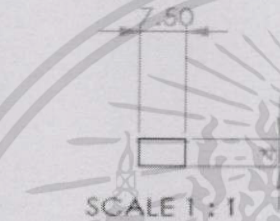


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.8 ตัวล๊อคแบบที่ 2



ค.9 ตัวล๊อคแบบที่ 3

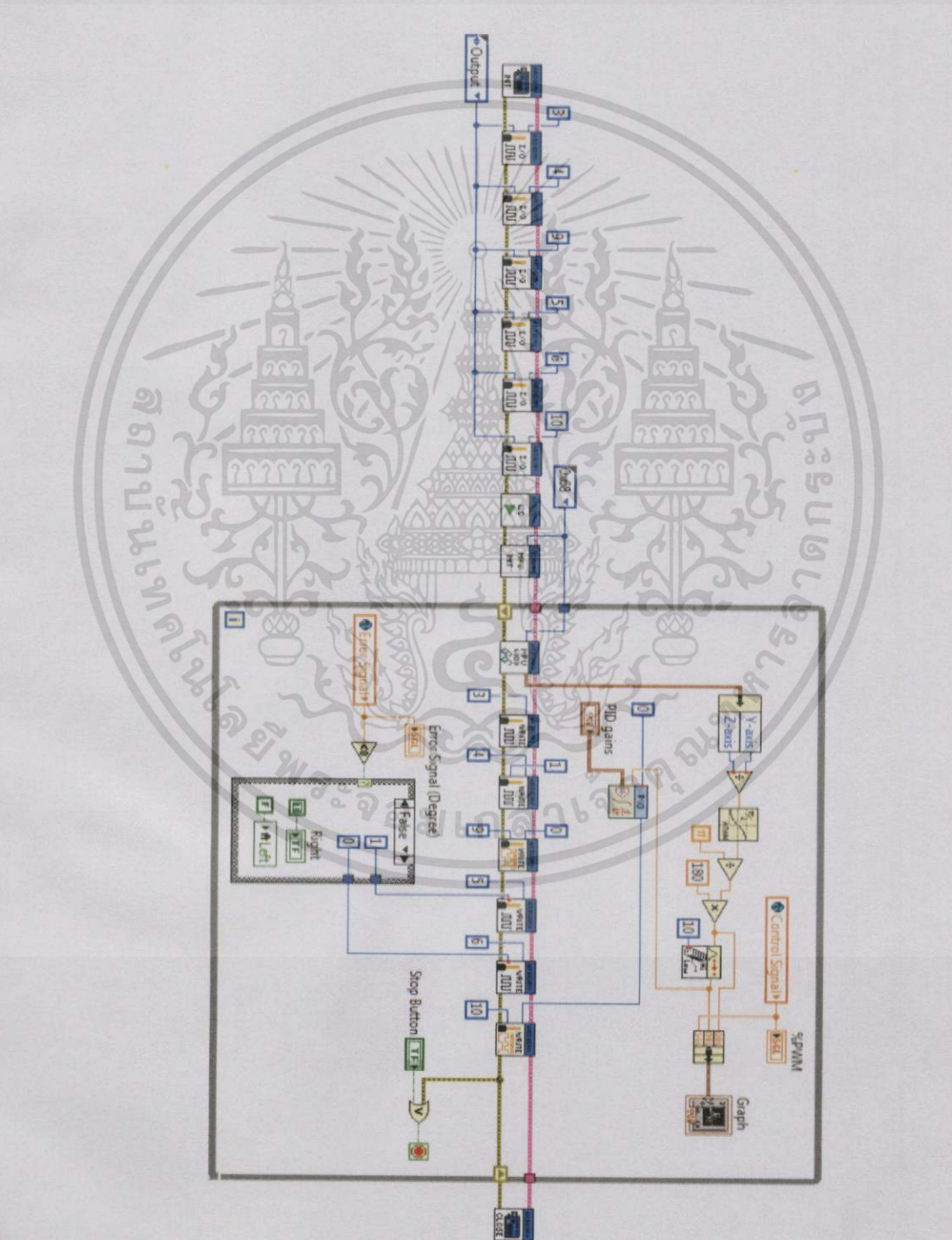


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

โปรแกรมควบคุมหุ่นยนต์จักรยานรักษาสมดุล

การเชื่อมต่อกันระหว่างโปรแกรม LabVIEW กับบอร์ด Arduino Uno R3 สามารถดาวน์โหลดไดร์เวอร์การติดตั้งจาก [3]



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้