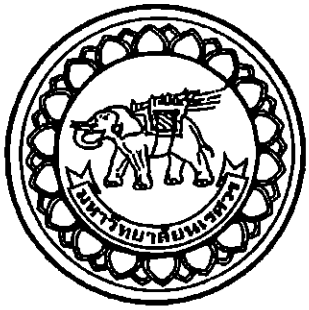


ระบบสาธิตควบคุมการเคลื่อนที่
ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

DEMONSTRATION SYSTEM FOR MOTION CONTROL
WITH REDUCED OSCILLATION OF ATTACHED PENDULUM

นายจักรวรรดิ บุตรดา รหัส 49363960

นายฟีรดนย์ ทองดี รหัส 52362076

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 20 ก.ค. 2559
เลขทะเบียน 1686 1416
เลขเรียกหนังสือ ป.ร.
มหาวิทยาลัยนเรศวร จ.พิษณุโลก 2556

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน ระบบสาธิตควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

ผู้ดำเนินโครงงาน นายจักรวรรดิ บุตรดา รหัส 49363960

นายพีรดนย์ ทองดี รหัส 52362076

ที่ปรึกษาโครงงาน ดร. สุวิทย์ กิระวิทยา

สาขาวิชา วิศวกรรรมไฟฟ้า

ภาควิชา วิศวกรรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2555

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

Suvit Kiravittaya .....ที่ปรึกษาโครงงาน

(ดร. สุวิทย์ กิระวิทยา)

.....กรรมการ

(ดร. มุฑิตา สงฆ์จันทร์)

.....กรรมการ

(อาจารย์ เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	**ระบบสาธิตควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่**
ผู้ดำเนินโครงงาน	**นายจักรวรรดิ บุตรดา รหัส 49363960** **นายพีรดนย์ ทองดี รหัส 52362076**
ที่ปรึกษาโครงงาน	**ดร. สุวิทย์ กิระวิทยา**
สาขาวิชา	**วิศวกรรรมไฟฟ้า**
ภาควิชา	**วิศวกรรรมไฟฟ้าและคอมพิวเตอร์**
ปีการศึกษา	2555

บทคัดย่อ

วัตถุประสงค์ของโครงงานนี้ คือ การสาธิตระบบควบคุมการเคลื่อนที่เพื่อลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่ และศึกษาลักษณะการสั่นของลูกตุ้มในแต่ละแบบที่ได้ทำการทดลองโดยการเคลื่อนที่ของมอเตอร์ แบ่งเป็น 3 แบบ คือ การให้มอเตอร์เคลื่อนที่แบบความเร็วคงที่ การให้มอเตอร์เคลื่อนที่แบบเร็วบ้างช้าบ้าง และการให้มอเตอร์เคลื่อนที่ด้วยความเร็วสูงสุดแล้วค่อยๆ ลดความเร็วจนถึงจุดหมาย จากการทดลองพบว่า การให้มอเตอร์เคลื่อนที่แบบเร็วบ้างช้าบ้างด้วยจังหวะที่เหมาะสมทำให้ลูกตุ้มสั่นน้อยลงตลอดช่วงการเคลื่อนที่ โดยผลการทดลองนี้สอดคล้องกับการคำนวณที่ได้จากการแก้สมการสถานะด้วยระเบียบวิธีเชิงเลขด้วยโปรแกรม MATLAB

Project title	Demonstration System for Motion Control with Reduced of Attached Oscillation	
Name	Mr. Chakkawat Butda	ID. 49363960
	Mr. Peeradon Tongdee	ID. 49363960
Project advisor	Dr. Suwit Kiravittaya	
Major	Electrical Engineering	
Department	Electrical and Computer Engineering	
Academic year	2012	

Abstract

The objective of this project is to demonstrate the movement control and study on the oscillation of pendulum attached to the mover. It is aimed to reduce the oscillation of the pendulum. Three kinds of control modes are studied. They are 1. Constant speed control, 2. two-speed control and 3. gradually-reduced speed control.

Experimental results show that the two-speed control with suitable control pattern give minimum oscillating amplitude. State equation of this system has been formulated. By numerical solving this equation within MATLAB program, we can show that the experimental result is consistent with the numerical calculation.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาช่วยเหลืออย่างดียิ่งของ ดร.สุวิทย์ กิระวิทยา อาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำแนะนำ แนวคิด ตลอดจนแก้ไขปัญหาข้อบกพร่องต่างๆ มาโดยตลอดจนโครงงานเล่มนี้เสร็จสมบูรณ์ จึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณ ดร. มุฑิตา สงฆ์จันทร์ และ อาจารย์ เศรษฐา ตั้งค้าวานิช ที่ปรึกษาร่วมโครงงาน ที่ให้คำปรึกษาที่มีค่ายิ่งเกี่ยวกับการเขียนโปรแกรมสนับสนุนโดยดร.สุวิทย์ กิระวิทยา

สุดท้ายนี้ขอขอบพระคุณทุกๆ ท่านที่ได้สนับสนุนการทำงาน และให้กำลังใจแก่คณะผู้ดำเนินโครงงานเสมอมา กระทั่งการดำเนินโครงงานครั้งนี้สำเร็จลุล่วงด้วยดี และความดีอันเกิดจากการดำเนินโครงงานครั้งนี้ คณะผู้ดำเนินโครงงานขอมอบแด่บิดา มารดา ครู อาจารย์ และผู้มีพระคุณทุกท่าน คณะผู้ดำเนินโครงงานมีความซาบซึ้งในความกรุณาอันดียิ่งจากทุกท่านที่ได้กล่าวนามมา และขอกราบขอบพระคุณมา ณ โอกาสนี้

นายจักรวรรดิ บุตรดา
นายพีรดนย์ ทองดี

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ฌ
สารบัญรูป ..... ญ

บทที่ 1 บทนำ 1
1.1 ความเป็นมา และความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 2
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนการทำงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 สถานที่ในการดำเนินโครงงาน 3
1.7 ระยะเวลาในการดำเนินโครงงาน 3
1.8 ขั้นตอน และแผนการดำเนินโครงงาน 3
1.9 งบประมาณ 4

บทที่ 2 หลักการ และทฤษฎีเบื้องต้น 5
2.1 บทนำ 5
2.1.1 ความเป็นมา 5
2.1.2 ชนิดของการเคลื่อนที่ 5
2.2 องค์ประกอบในระบบควบคุมการเคลื่อนที่ 5

สารบัญ (ต่อ)

หน้า

2.3 บอร์ด ET-OPTO DC MOTOR 6
2.4 หลักการทำงานของมอเตอร์กระแสตรง 8
2.5 Microcontroller Arduino Duemilanove AT MEGA 328 9
2.5.1 การทำงานของ Arduino Duemilanove AT MEGA 328 9
2.6 การควบคุมแบบ PWM (Pulse Width Modulation) 11
2.6.1 การทำงานของสัญญาณ PWM 11
2.6.2 การใช้ PWM ในการควบคุมความเร็วมอเตอร์ 12
2.7 ลิมิตสวิตช์ (Limit Switch) 12
2.7.1 ข้อดีของลิมิตสวิตช์ (Limit Switch) 12
2.8 MATLAB 13
2.8.1 ลักษณะทั่วไปของ MATLAB 13
2.9 การเคลื่อนที่ของลูกตุ้ม 15
2.9.1 การเคลื่อนที่ของลูกตุ้ม 15
2.9.2 ความสัมพันธ์ของความเร็วเชิงมุมหรือความเร่งเชิงมุม 15

บทที่ 3 วิธีดำเนินโครงงาน 18
3.1 ศึกษาและเก็บข้อมูล 18
3.1.1 ศึกษาการใช้งาน 18
3.1.2 ศึกษาวิธีดำเนินงานจัดทำ 18
3.2 ออกแบบการทดลอง 18
3.3 เขียนโปรแกรม 19
3.4 ทดสอบ และปรับปรุง 19
3.5 ประเมินผล 19
3.6 สรุป และจัดทำรูปเล่มโครงงาน 19

สารบัญ (ต่อ)

หน้า

3.7 อุปกรณ์ในการดำเนินงาน 19

3.8 การเชื่อมต่ออุปกรณ์ 20

3.9 สร้างอุปกรณ์ในการทดลอง 20

3.10 ขั้นตอนการทดลองโดยแบ่งการทดลองเป็น 3 แบบ 21

3.10.1 การทดลองแบบที่ 1 22

3.10.2 การทดลองแบบที่ 2 23

3.10.3 การทดลองแบบที่ 3 24

3.10.4 สังเกตการสั่นของลูกตุ้มขณะมอเตอร์หยุดเคลื่อนที่ที่จุดหมาย 24

บทที่ 4 ผลการทดลอง 25

4.1 ผลการทดลองรูปแบบที่ 1 25

4.1.1 สรุปผลการทดลองรูปแบบที 1 29

4.2 ผลการทดลองรูปแบบที่ 2 30

4.2.1 สรุปผลการทดลองรูปแบบที่ 3 33

4.3 ผลการทดลองรูปแบบที่ 3 34

4.3.1 สรุปผลการทดลองรูปแบบที่ 3 37

4.4 เปรียบเทียบผลการทดลอง 38

บทที่ 5 สรุปผลของการทดลอง และปัญหาที่พบ 39

5.1 สรุปผลการทดลอง 39

5.2 ปัญหาที่พบ 39

5.3 การพัฒนา และปรับปรุงแก้ไข 40

สารบัญ (ต่อ)

หน้า

ภาคผนวก ก 41

ภาคผนวก ข 57

เอกสารอ้างอิง.....73

ประวัติผู้ดำเนินโครงงาน.....74

สารบัญตาราง

ตารางที่ หน้า

1.1 ขั้นตอน และแผนการดำเนินโครงงาน 3

2.1 ตารางแสดงการส่งสัญญาณควบคุมมอเตอร์ไฟตรง 7

4.1 ตารางแสดงระยะการแกว่งของลูกตุ้มในการทดลองแบบที่ 1 27

4.2 ตารางแสดงระยะการแกว่งของลูกตุ้มในการทดลองแบบที่ 2 31

4.3 ตารางแสดงระยะการแกว่งของลูกตุ้มในการทดลองแบบที่ 3 35

4.4 ตารางแสดงแอมพลิจูดสูงสุดของการแกว่งของลูกตุ้ม 38

สารบัญรูป

รูปที่ หน้า

1.1 รูปแบบโครงสร้าง....1
2.1 รูปโครงสร้างระบบสาธิตการควบคุม....6
2.2 ET-OPTO DC MOTOR....6
2.3 วงจรภายในของมอเตอร์กระแสตรง....8
2.4 การทำงานของ Arduino....9
2.5 แสดงสัญญาณ PWM ซึ่งแสดงค่า Duty Cycles ที่ต่างๆกัน....11
2.6 สัญลักษณ์ สวิตช์จำกัดระยะ (Limit Switch)....12
2.7 การสั่นของลูกตุ้ม....15
3.1 การเชื่อมต่ออุปกรณ์....20
3.2 กราฟแสดงความเร็วของมอเตอร์ในการทดลองที่ 1....22
3.3 กราฟแสดงความเร็วของมอเตอร์ในการทดลองที่ 2....23
3.4 กราฟแสดงความเร็วของมอเตอร์ในการทดลองที่ 3....24
4.1 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 1....25
4.2 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 2....25
4.3 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 3....26
4.4 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 4....26
4.5 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 5....27
4.6 กราฟความเร็วเทียบกับเวลาของการทดลองแบบที่ 1....27
4.7 กราฟแสดงระยะทาง แอมพลิจูดการสั่น และความเร็วของลูกตุ้ม ในการทดลองแบบที่ 1....28
4.8 กราฟแสดงแอมพลิจูดการสั่นของลูกตุ้ม ในการทดลองแบบที่ 1....28
4.9 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 1....30
4.10 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 2....30
4.11 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 3....30
4.12 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 4....31

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.13 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 5.....31
4.14 กราฟความเร็วเทียบกับเวลาของการทดลองแบบที่ 2.....32
4.15 กราฟแสดงระยะทาง แอมพลิจูดการสั่น และความเร็วของลูกตุ้ม ในการทดลองแบบที่ 2.....32
4.16 กราฟแสดงแอมพลิจูดการสั่นของลูกตุ้ม ในการทดลองแบบที่ 2.....33
4.17 การแกว่งของลูกตุ้มในการทดลองแบบที่ 3 ครั้งที่ 1.....34
4.18 การแกว่งของลูกตุ้มในการทดลองแบบที่ 3 ครั้งที่ 2.....34
4.19 การแกว่งของลูกตุ้มในการทดลองแบบที่ 3 ครั้งที่ 3.....34
4.20 การแกว่งของลูกตุ้มในการทดลองแบบที่ 3 ครั้งที่ 4.....35
4.21 การแกว่งของลูกตุ้มในการทดลองแบบที่ 3 ครั้งที่ 5.....35
4.22 กราฟความเร็วเทียบกับเวลาของการทดลองแบบที่ 3.....36
4.23 กราฟแสดงระยะทาง แอมพลิจูดการสั่น และความเร็วของลูกตุ้ม ในการทดลองแบบที่ 3.....36
4.24 กราฟแสดงแอมพลิจูดการสั่นของลูกตุ้ม ในการทดลองแบบที่ 3.....37
4.25 แอมพลิจูดการทดลองที่ 1.....38
4.26 แอมพลิจูดการทดลองที่ 2.....38
4.27 แอมพลิจูดการทดลองที่ 3.....38

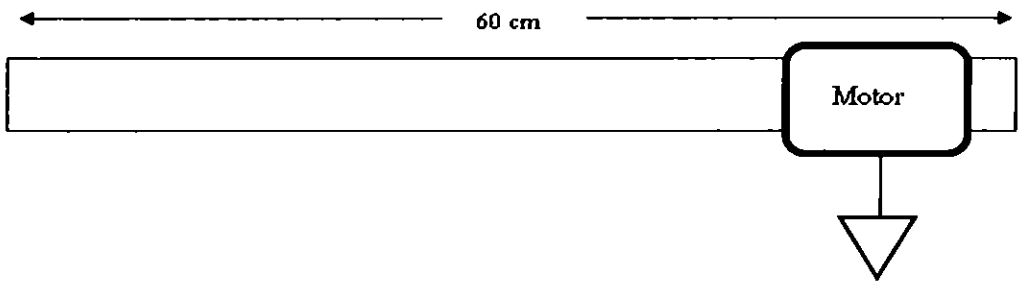
บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของโครงงาน

โครงงานนี้เป็นการสร้างระบบสาธิตควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่ โดยใช้วงจรขับเคลื่อนมอเตอร์ในการทำงานของระบบควบคุมจะต้องรักษาสมดุลของลูกตุ้มเพื่อให้ลูกตุ้มมีการสั่นลดลง ระบบจะทำการควบคุมความเร็วในการเคลื่อนที่ และตำแหน่งที่หยุดเคลื่อนที่ ผู้ดำเนินโครงงานได้ใช้วงจรขับเคลื่อนมอเตอร์เป็นตัวขับเคลื่อน ตัวเคลื่อนที่จะเคลื่อนที่ด้วยความเร็วที่ระบบกำหนด และทำการลดความเร็วลงจนหยุดนิ่งในตำแหน่งที่ต้องการให้หยุดนิ่ง พิจารณาการลดความเร็วว่าการลดความเร็วในลักษณะใดจะทำให้ลูกตุ้มที่ติดอยู่กับตัวเคลื่อนที่มีการสั่นลดลง ซึ่งถ้าหยุดนิ่งในความเร็วที่ไม่สัมพันธ์กับตำแหน่งที่จะหยุดลูกตุ้มก็จะมีการสั่นที่มาก แต่ถ้าควบคุมให้หยุดนิ่งในความเร็วที่สัมพันธ์กับตำแหน่งที่จะหยุดลูกตุ้มก็จะมีการสั่นที่น้อยลง

ในทางปฏิบัติการควบคุมการเคลื่อนที่โดยใช้วงจรขับเคลื่อนมอเตอร์ ผู้ดำเนินโครงงานมีความสนใจที่จะนำทฤษฎีระบบรูปที่ 1.1 รูปแบบโครงสร้างควบคุมมาประยุกต์ใช้ในการออกแบบตัวควบคุมเพื่อให้การทำงานของระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่ ให้มีเสถียรภาพ (Stable) ของการทรงตัวให้มากขึ้น



รูปที่ 1.1 รูปแบบโครงสร้าง

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ออกแบบ และสร้างระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

1.2.2 ทดสอบการทำงานของระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่อยู่บนตัวเคลื่อนที่

1.2.3 นำความรู้ทางด้านการควบคุมระบบมาใช้ในระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

1.3 ขอบเขตของโครงงาน

1.3.1 ออกแบบ และสร้างระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

1.3.2 สร้างระบบควบคุมที่สามารถควบคุมความเร็ว และกำหนดตำแหน่งในการเคลื่อนที่ได้

1.3.3 สาธิตการลดการสั่นของลูกตุ้มขณะหยุดเคลื่อนที่

1.4 ขั้นตอนการทำงาน

1.4.1 ศึกษาการทำงานของวงจรขับเคลื่อนมอเตอร์

1.4.2 สร้างวงจรขับเคลื่อนมอเตอร์

1.4.3 ออกแบบ และสร้างระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

1.4.4 ทดสอบการทำงานของระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่ และปรับปรุงระบบ

1.4.5 ศึกษาการสั่นของลูกตุ้ม

1.4.6 จัดทำเอกสาร และคู่มือการใช้งาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้ระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

1.5.2 ได้เรียนรู้การสร้างระบบควบคุม และวงจรขับเคลื่อนมอเตอร์

1.5.3 นำไปประยุกต์ใช้งานได้จริง

1.6 สถานที่ในการดำเนินโครงงาน

ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

1.7 ระยะเวลาในการดำเนินโครงงาน

ตั้งแต่เดือน กรกฎาคม พ.ศ. 2556 ถึง กุมภาพันธ์ พ.ศ. 2557

1.8 ขั้นตอน และแผนการดำเนินโครงงาน

ตารางที่ 1.1 ขั้นตอน และแผนการดำเนินโครงงาน

ลำดับ	การดำเนินโครงงาน	ช่วงเวลา							
		ก.ค. 56	ส.ค. 56	ก.ย. 56	ต.ค. 56	พ.ย. 56	ธ.ค. 56	ม.ค. 57	ก.พ. 57
1.8.1	ศึกษาการทำงานของวงจรขับเคลื่อนมอเตอร์	■	■						
1.8.2	สร้างวงจรขับเคลื่อนมอเตอร์		■	■	■	■			
1.8.3	ออกแบบ และสร้างระบบสาธิต				■	■	■		
1.8.4	ทดสอบการทำงานของระบบแลปรับปรุงระบบ					■	■	■	
1.8.5	ศึกษาการสั่นของลูกตุ้ม					■	■	■	
1.8.6	จัดทำปริญญานิพนธ์ฉบับสมบูรณ์		■	■	■	■	■	■	■

1.9 งบประมาณ

1.9.1 ค่าอุปกรณ์ และเครื่องมือในการทำฮาร์ดแวร์	6,480 บาท
1.9.2 ค่าจัดทำปริญญานิพนธ์	700 บาท
1.9.3 รวมเป็นเงินทั้งสิ้น (เจ็ดพันหนึ่งร้อยแปดสิบบ้านถ้วน)	7,180 บาท

1.9.4 หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2

หลักการและทฤษฎีเบื้องต้น

2.1 บทนำ

2.1.1 ความเป็นมา

การควบคุมการเคลื่อนที่ (Motion Control) วิทยาการในการควบคุมระบบเครื่องกลไฟฟ้า (Electromechanical System) ให้มีการตอบสนองในรูปแบบการเคลื่อนที่ (Motion) ตามที่ต้องการ โดยในระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่นี้จะใช้วงจรขับเคลื่อนมอเตอร์เป็นตัวขับเคลื่อน

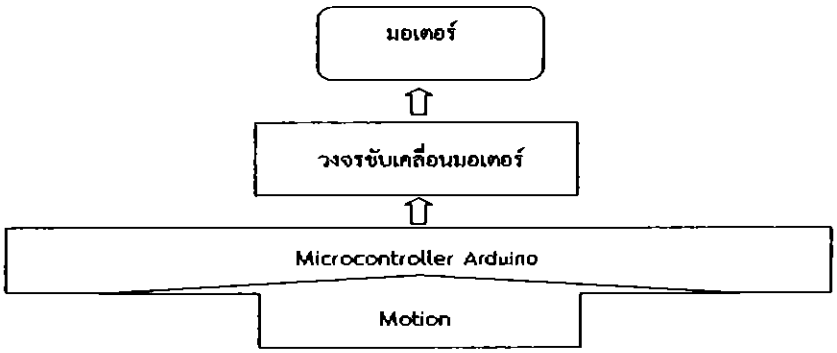
2.1.2 ชนิดของการเคลื่อนที่

2.1.2.1 การเคลื่อนที่แบบจุดต่อจุด (Point To Point Motion) เป็นการเคลื่อนที่ ซึ่งสนใจเฉพาะตำแหน่งที่ต้องการให้กลไกเคลื่อนไปถึงโดยไม่บังคับเส้นทาง

2.1.2.2 การเคลื่อนที่แบบต่อเนื่อง (Continuous Motion) เป็นการเคลื่อน ซึ่งบังคับเส้นทางผ่านไปตลอดจนถึงตำแหน่งที่ต้องการให้กลไกเคลื่อนที่ไปถึง

2.2 องค์ประกอบในการควบคุมการเคลื่อนที่

การควบคุมการเคลื่อนที่ (Motion Control Performance) จะขึ้นอยู่กับองค์ประกอบ 4 ส่วนหลัก ดังแสดงในรูปที่ 2.1 คือ มอเตอร์ (motor) การเคลื่อนที่ (Motion) วงจรขับเคลื่อนมอเตอร์ (Motor Cycle) และตัวควบคุม (Controller) โดยองค์ประกอบแต่ละส่วนมีความสัมพันธ์กันตามโครงสร้างระบบควบคุมการเคลื่อนที่

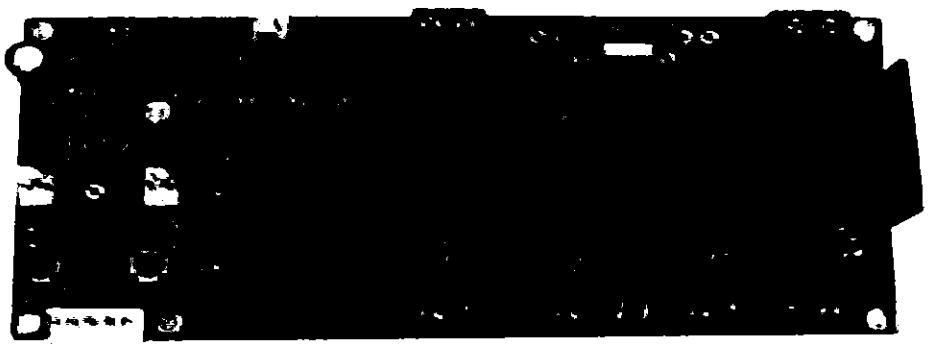


รูปที่ 2.1 โครงสร้างระบบควบคุมการเคลื่อนที่

2.3 บอร์ด ET-OPTO DC MOTOR

บอร์ดใช้งานอิสระ หรือต่อเข้ากับบอร์ด Microcontroller ต่างๆ ใช้ควบคุมการทำงานของตัวมอเตอร์ไฟตรงให้หมุน ซ้าย, ขวา และควบคุมความเร็วของตัวมอเตอร์ไฟตรงออกแบบใช้กับมอเตอร์ไฟตรงกระแสสูงๆได้

- ใช้ Power มอสเฟส N-Channel เบอร์ RFP50N06 ขนาด 60V/50A จำนวน 4 ตัวในการใช้งาน
- ใช้ 5 Pin ต่อควบคุมจากบอร์ดภายนอก DIR1, DIR2, ENA
- มีวงจรภายในต่อใช้งานอิสระโดยตรง ไม่ต้องใช้บอร์ดควบคุม โดยมี 2 Switch สั่งหมุน ซ้าย, ขวา และ VR ปรับความเร็วหรือใช้ Pulse Logic จากไมโครคอนโทรลเลอร์ควบคุมการทำงานได้เอง เมื่อเลือก Jumper ให้ควบคุมผ่านการ Control
- สามารถต่อกับมอเตอร์ไฟตรงได้ 1 ตัว ขนาดไม่เกิน 24 V DC กระแส ไม่เกิน 5 Amp (สามารถใช้กับกระแสได้สูงกว่านี้ โดยย้ายวงจร มอสเฟส ออกภายนอกต่อสายตรง)
- แหล่งจ่ายไฟใช้งานวงจร 5 V DC 2 Pin
- PCB Size 12.7x5.6 cm. การทำงานของวงจร



รูปที่ 2.2 ET-OPTO DC MOTOR

เมื่อทำการเชื่อมต่อขาสัญญาณเรียบร้อยแล้วในการเขียนโปรแกรมผู้ใช้สามารถส่งสัญญาณมาควบคุมการหมุนของมอเตอร์ไฟตรงได้ตามตารางที่ 2.1 ให้สังเกตว่า ในการป้อน PWM ควบคุมความเร็วของมอเตอร์นั้นจะต้องป้อนเข้าที่ขา ENA ถ้าไม่ต้องการควบคุมความเร็ว และยังคงให้มอเตอร์หมุนทำงานได้ที่ความเร็วเต็มที่จะต้องกำหนดให้ขา ENA เป็น 1 ไว้เสมอ ถ้าขา ENA เป็น 0 มอเตอร์จะหยุดหมุน

ตารางที่ 2.1 ตารางแสดงการส่งสัญญาณควบคุมมอเตอร์ไฟตรง

ENA/PWM	DIR1	DIR2	STATUS DC MOTOR
0	X	x	SLOW STOP
1	0	0	SLOW STOP
1	0	1	ROTATE RIGHT
1	1	0	ROTATE LEFT
1	1	1	FAST STOP

เมื่อ ENA = 0 จะไม่สนใจสภาวะของ DIR1 และ DIR2 ทำให้ไอซีออปโต้ทั้ง 4 ตัวไม่ทำงานจึงไม่มีแรงดันไป ไบอัสให้กับขาของ Gate ของมอสเฟส ทั้ง 4 ตัว ทำให้มอสเฟสไม่ทำงาน มอเตอร์ก็จะไม่หมุน หรือถ้าหมุนอยู่มอเตอร์ก็จะหยุดแบบ Slow

เมื่อ ENA = 1 ส่วน DIR1, DIR2 = 0 แล้ว จะมีลักษณะการทำงานเหมือนกับเงื่อนไขแรก

เมื่อ ENA = 1 ส่วน DIR1 = 0, DIR2 = 1 จะทำให้ไอซีออปโต้ตัวที่ 1 กับ 4 ทำงานส่วนตัวที่ 2 และ 3 ไม่ทำงานส่งผลให้มีแรงดันไบอัสที่ขา Gate ของ Q1 และ Q4 ทำให้ มอสเฟส คู่นี้ On มีกระแสไหลจาก Q1 ผ่านไปยังมอเตอร์ไฟตรงครบวงจรที่ Q4 ลงกราวด์ ทำให้มอเตอร์หมุนขวา (ตามเข็มนาฬิกา)

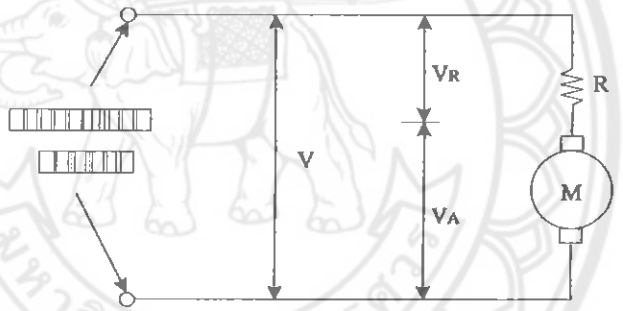
เมื่อ ENA = 1 ส่วน DIR1 = 1, DIR2 = 0 จะทำให้ไอซีออปโต้ตัวที่ 2 กับ 3 ทำงาน ส่วนตัวที่ 1 และ 4 ไม่ทำงานส่งผลให้มีแรงดันไบอัสที่ขา Gate ของ Q3 และ Q4 ทำให้ มอสเฟส คู่นี้ On ทำให้มีกระแสไหลผ่าน Q2 ผ่านมอเตอร์ไฟตรงครบวงจรที่ Q3 ลงกราวด์ ทำให้มอเตอร์หมุนซ้าย (ทวนเข็มนาฬิกา)

เมื่อ ENA = 1 ส่วน DIR1, DIR2 = 1 จะทำให้ไอซีออปโต้ตัวที่ 3 กับ 4 ทำงาน ส่วนตัวที่ 1 และ 2 ไม่ทำงานส่งผลให้มีแรงดันไบอัสที่ขา Gate ของ Q3 และ Q4 ทำให้ มอสเฟส คู่นี้ On ดึงกระแสที่ไหลผ่านมอเตอร์ไฟตรงอยู่ลงกราวด์ทันที ทำให้มอเตอร์หยุดทำงานทันที

2.4 หลักการทำงานของมอเตอร์กระแสตรง

เมื่อผ่านกระแสไฟฟ้าเข้าไปยังขดลวดในสนามแม่เหล็กจะทำให้เกิดแรงแม่เหล็ก ซึ่งมีสัดส่วนของแรงขึ้นกับกระแสแรงของสนามแม่เหล็ก โดยแรงจะเกิดขึ้นเป็นมุมฉากกับกระแส และสนามแม่เหล็กขณะที่ทิศทางของแรงกลับตรงข้ามกันกระแสของสนามแม่เหล็กไหลย้อนกลับจะทำให้เกิดการเปลี่ยนแปลงของกระแส ทำให้สนามแม่เหล็กมีทิศทางของแรงที่เปลี่ยนไป ทำให้มอเตอร์กระแสตรงกลับทิศทางการหมุนได้

สนามแม่เหล็กของมอเตอร์ส่วนหนึ่งเกิดขึ้นจากแม่เหล็กถาวรที่ถูกยึดติดกับแผ่นเหล็กโดยปกติส่วนนี้จะยึดอยู่กับที่ และขดลวดเหนี่ยวนำจะพันอยู่กับส่วนที่เป็นแกนหมุนของมอเตอร์ ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 วงจรภายในของมอเตอร์กระแสตรง

ในการทำงานผู้ปฏิบัติงานได้เลือกใช้ ไมโครคอนโทรลเลอร์ (Microcontroller Arduino) ในการควบคุมความเร็ว และทิศทางในการหมุนของมอเตอร์ในวงจรขับเคลื่อน

2.5 Microcontroller Arduino Duemilanove AT MEGA 328

คือ Microcontroller ชนิดหนึ่งสร้างมาจาก Microcontroller ชนิด AVR ปกติเราจะต้องเขียนโปรแกรมจากภาษาที่ยาก และบางตัวอาจจะต้องใช้กับระบบปฏิบัติการที่จำกัดโดย Arduino ทำให้การเขียนโปรแกรมนั้นง่ายขึ้น สามารถเขียนโปรแกรมด้วยภาษาที่ง่ายอย่างภาษา C ผ่านทาง Arduino ซึ่ง Arduino สามารถใช้ได้กับระบบปฏิบัติการหลายตระกูล

รูปที่ 2.4 การทำงานของ Arduino

2.5.1 การทำงานของ Arduino Duemilanove AT MEGA 328

Arduino คือ เครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอก และส่งสัญญาณไป ควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมาก ตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยว และมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน ผู้ปฏิบัติงานจึงใช้ Arduino เป็นตัวรับสัญญาณจากตัว Sensor ในที่นี้ผู้ปฏิบัติงานได้ใช้ Sensor วัดความเร็วรอบของมอเตอร์ (Infrared sensor) ในการตรวจจับจากนั้น Sensor จะทำการป้อนกลับสัญญาณมายังไมโครคอนโทลเลอร์เพื่อควบคุมการลดความเร็วในการหมุนของมอเตอร์เมื่อความเร็วต่อรอบของมอเตอร์เกินกว่าที่ผู้ปฏิบัติงานกำหนด จากนั้นเมื่อหยุดนิ่งได้สักพัก ก็จะสั่งการให้มอเตอร์หมุนย้อนกลับ เพื่อให้ตัวเคลื่อนที่เคลื่อนที่กลับมาที่เดิม

บอร์ด Arduino ใช้พัฒนาการใช้งาน Microcontroller ในตระกูล AVR เพราะว่าเป็น Open Source สามารถดัดแปลงไปใช้งานได้ทั้ง Hardware และ software ได้ทันที ภาษาที่ใช้กับบอร์ดนี้จะเป็นลักษณะของ C/C+ โดยมี Libaries ต่างๆ ให้

Arduino มีจุดเด่นเรื่องของ ความง่ายในการเรียนรู้ และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่างๆ ขึ้นมาสนับสนุนการใช้งาน ด้วยรูปแบบที่ง่ายไม่ซับซ้อน ถึงแม้ว่า Arduino เองจะมีรูปแบบการงานคล้ายๆ กับไมโครคอนโทลเลอร์อย่าง Basic Stamp ของ Parallax, BX-24 ของ Netmedias และ Handy Board ของ MIT แต่ก็มีจุดเด่นกว่ารายอื่นๆ หลายอย่าง เป็นต้นว่า

2.5.1.1 ราคาไม่แพง เนื่องจากมี Source Code และวงจรแจกให้ฟรี สามารถต่อวงจรขึ้นมาใช้งานได้เอง

2.5.1.2 โปรแกรมที่ใช้พัฒนาของ Arduino รองรับการทำงานทั้ง Windows, Linux และ Macintosh OSX

2.5.1.3 มีรูปแบบคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถนำไปใช้งานจริงๆ ที่มีความซับซ้อนมากๆ ได้ และยังสามารถสร้างคำสั่ง และ Library ใหม่ๆ ขึ้นมาใช้งานเองได้ เมื่อมีความชำนาญมากขึ้นแล้ว

2.5.1.4 มีการเปิดเผยวงจร และ Source Code ทั้งหมดทำให้สามารถนำไปพัฒนาต่อยอดเพิ่มเติมได้ตามความต้องการทั้ง Hardware และ Software

Arduino เป็นบอร์ดไมโครคอนโทลเลอร์ โดยใช้ AVR ขนาดเล็กเป็นตัวช่วยประมวลผล และสั่งงานเหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทลเลอร์ และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์ Input/Output ต่างๆ ได้มากมาย ทั้งในแบบที่เป็นการทำงานตัวเดียวแบบอิสระ หรือเชื่อต่อสั่งงานร่วมกับอุปกรณ์อื่นๆ เช่น คอมพิวเตอร์ PC ทั้งนี้ก็เนื่องมาจากว่า Arduino สนับสนุนการเชื่อต่อกับอุปกรณ์ Input/Output ต่างๆ ได้มากมาย ทั้งในแบบ Digital และ Analog เช่น การรับค่าจากสวิตช์ หรืออุปกรณ์ตรวจจับแบบต่างๆ รวมไปถึง การควบคุมอุปกรณ์ Output ต่างๆ ตั้งแต่หลอดไฟLED, มอเตอร์, รีเลย์ ฯลฯ โดยระบบฮาร์ดแวร์ของ Arduino สามารถสร้าง และประกอบขึ้นใช้งานได้เอง ในกรณีผู้ใช้พอมีความรู้ด้านอิเล็กทรอนิกส์อยู่บ้าง หรือ สามารถซื้อแผงวงจรสำเร็จรูปที่มีการผลิตออกจำหน่ายกันในราคาที่ไม่แพง สำหรับเรื่องของโปรแกรมที่จะใช้เป็นเครื่องมือในการพัฒนานั้น สามารถ Download มาใช้งานได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆ โดย Arduino มีจุดเด่นในเรื่องความง่ายของการพัฒนาโปรแกรม และยังมีเอกสารข้อมูลรวมทั้งตัวอย่างต่างๆ ให้ใช้เป็นแนวทางในการศึกษาเรียนรู้เป็นจำนวนมาก เนื่องจาก Arduino เป็นระบบพัฒนาไมโครคอนโทลเลอร์แบบ Open Source ซึ่งมีการตีพิมพ์เอกสารต่างๆ ที่เกี่ยวข้องมาเผยแพร่ให้ได้รับรู้เป็นระยะๆ รวมทั้งการเปิดเผย Source Code และตัวอย่างต่างๆ ให้ผู้ใช้งาน หรือพัฒนาดัดแปลงต่อยอดได้โดยไม่เสียค่าใช้จ่าย ด้วยเหตุนี้จึงมีผู้คนทั่วไปให้ความสนใจ และนำไปศึกษาทดลองใช้งานกันมากมาย มีการดัดแปลง และสร้าง

เป็นโครงงาน แบบต่างๆ กันเป็นจำนวนมาก จึงเป็นประโยชน์อย่างยิ่งสำหรับผู้เริ่มต้นที่สามารถนำเอาตัวอย่าง และโครงงานต่างๆที่คนอื่นทำไว้แล้ว มาใช้อ้างอิงเป็นแนวทางหาการศึกษาเรียนรู้ได้โดยง่าย และที่สำคัญ คือ ฟรีไม่เสียค่าใช้จ่าย

2.6 การควบคุมแบบ PWM (Pulse Width Modulation)

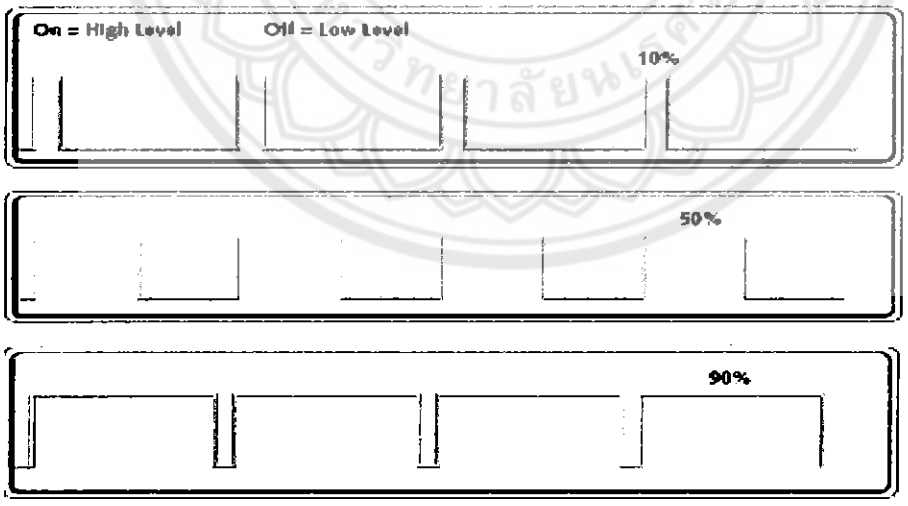
Pulse width modulation (PWM) คือ เทคนิคสำหรับควบคุมวงจรทางด้านฮาร์ดแวร์โดยใช้สัญญาณเอาท์พุตแบบดิจิตอลของไมโครโปรเซสเซอร์ควบคุม

2.6.1 การทำงานของสัญญาณ PWM

2.6.1.1 โดยรูปที่ 2.5.1 แสดงสัญญาณ PWM ที่ร้อยละ 10 Duty Cycle คือ สัญญาณในการออนจะเป็นร้อยละ 10 ของ คาบสัญญาณ และจะออฟเป็นร้อยละ 90 ของคาบสัญญาณ

2.6.1.2 โดยรูปที่ 2.5.2 แสดงสัญญาณ PWM ที่ร้อยละ 50 Duty Cycle คือ สัญญาณในการออนจะเป็นร้อยละ 10 ของคาบสัญญาณ และจะออฟเป็นร้อยละ 50 ของคาบสัญญาณ

2.6.1.3 โดยรูปที่ 2.5.3 แสดงสัญญาณ PWM ที่ร้อยละ 90 Duty Cycle คือ สัญญาณในการออนจะเป็นร้อยละ 10 ของ คาบสัญญาณและจะออฟเป็น ร้อยละ 10 ของคาบสัญญาณ



รูปที่ 2.5.1

รูปที่ 2.5.2

รูปที่ 2.5.3

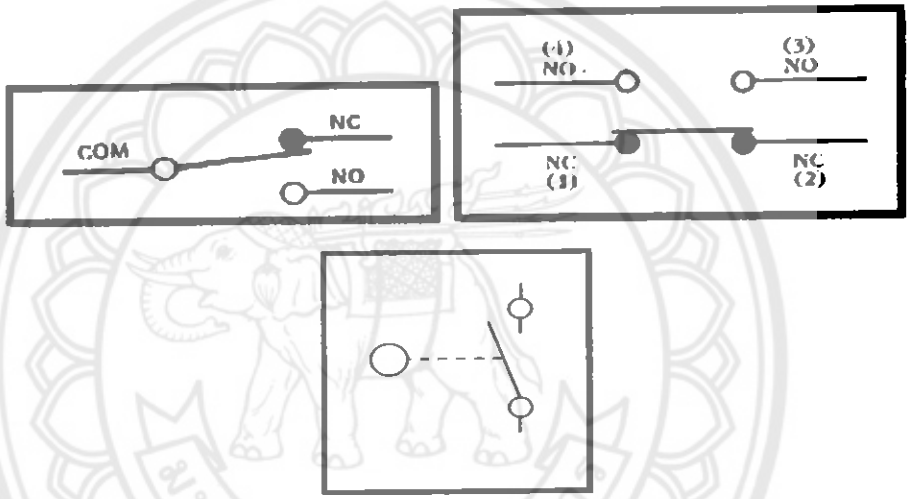
รูปที่ 2.5 แสดงสัญญาณ PWM ซึ่งแสดงค่า Duty Cycles ที่ต่างๆกัน

2.6.2 การใช้ PWM ในการควบคุมความเร็วมอเตอร์

PWM ง่ายในการอินเตอร์เฟสกับไมโครคอนโทรลเลอร์ และใช้เพียงแค่เอาท์พุตสัญญาณเดียวในการควบคุมความเร็ว

2.7 ลิมิตสวิตช์ (Limit Switch)

เป็นสวิตช์ที่จำกัดระยะทางการทำงานอาศัยแรงกดภายนอกมากระทำเช่น วางของทับที่ปุ่มกดหรือลูกเบี้ยวมาชนที่ปุ่มกด และเป็นผลทำให้หน้าสัมผัสที่ต่ออยู่กับก้านชน เปิด-ปิด ตามจังหวะของการชน



รูปที่ 2.6 สัญลักษณ์ สวิตช์จำกัดระยะ (Limit Switch)

ดังนั้น จึงมีการนำไปใช้ประโยชน์ได้อย่างมากมาย เช่น ลิฟท์โดยสาร, ลิฟท์ขนของ, ประตูที่ทำงานด้วยไฟฟ้า, ระบบสายพานลำเลียง เป็นต้น และลิมิตสวิตช์ (Limit Switch) สามารถเชื่อมต่อได้หลายอัน มีเชื่อมต่อปกติปิด และปกติเปิดมี โครงสร้างคล้ายสวิตช์ ปุ่มกด

2.7.1 ข้อดีของลิมิตสวิตช์ (Limit Switch)

2.7.1.1 ติดตั้งง่าย สะดวกต่อการใช้งาน

2.7.1.2 ไม่ต้องมีไฟเลี้ยงวงจรในการทำงาน

2.7.1.3 การทำงานเชื่อถือได้ มีความแม่นยำในการทำงาน

2.7.1.4 ราคาต่ำกว่าอุปกรณ์ตรวจจับชนิดอื่น

2.8 MATLAB

เป็นซอฟท์แวร์ (Software) สำหรับการวิเคราะห์เชิงตัวเลข (Numerical Analysis) เขียนขึ้นโดย Dr. Cleve Moler ตั้งแต่ ค.ศ. 1982 ในตอนแรกเป็นการรวบรวมโปรแกรมการคำนวณเกี่ยวกับชุดข้อมูลตาราง (Matrices) ที่เขียนขึ้นโดยใช้ภาษาฟอร์แทรน (FORTRAN) แล้วเรียบเรียงเป็นหนังสือชื่อ Demonstration of Matrix Library โดย C.B Moler ซึ่งต่อมามีการพัฒนาจนกลายเป็นโปรแกรมขนาดใหญ่ ที่เขียนโดยใช้ภาษาซี (C) และแอสเซมเบลอร์ (Assembler) รากฐานของ MATLAB นั้นมาจากโครงการ EISPACK (Eigen System Package) และ LINPACK (Linear System Package) ณ Argonne National Laboratory สหรัฐอเมริกา โดย EISPACK เป็นโปรแกรมย่อยภาษาฟอร์แทรน (FORTRAN SUBROUTINES) ซึ่งจะคำนวณ Eigen Values และ Eigen Vectors สำหรับ Matrices ประเภทต่างๆ ซึ่งจะอธิบายโดยละเอียดในหนังสือ Matrix Eigen System Routines : EISPACK Guide โดย Smith, Boyle, Dongarra, Garbow, Ikebe, Klema และ Moler ส่วน LINPACK เป็นโปรแกรมย่อยภาษาฟอร์มแทนสำหรับการวิเคราะห์ และแก้ระบบสมการพีชคณิตเชิงเส้น (Simultaneous Linear Algebraic Equations) และปัญหาด้าน Linear Least Square ซึ่งมีคำอธิบายใน The LINPACK User's Guide โดย Dongarra, Bunch, Moler และ Stewart ปัจจุบันมีการใช้ MATLAB ในการเรียนการสอน ทางวิศวกรรมศาสตร์ในมหาวิทยาลัยทั่วไปในสหรัฐอเมริกา ซอฟท์แวร์เป็นลิขสิทธิ์ของ The MathWorks, Inc., 21 Eliot St., South Natick MA01760

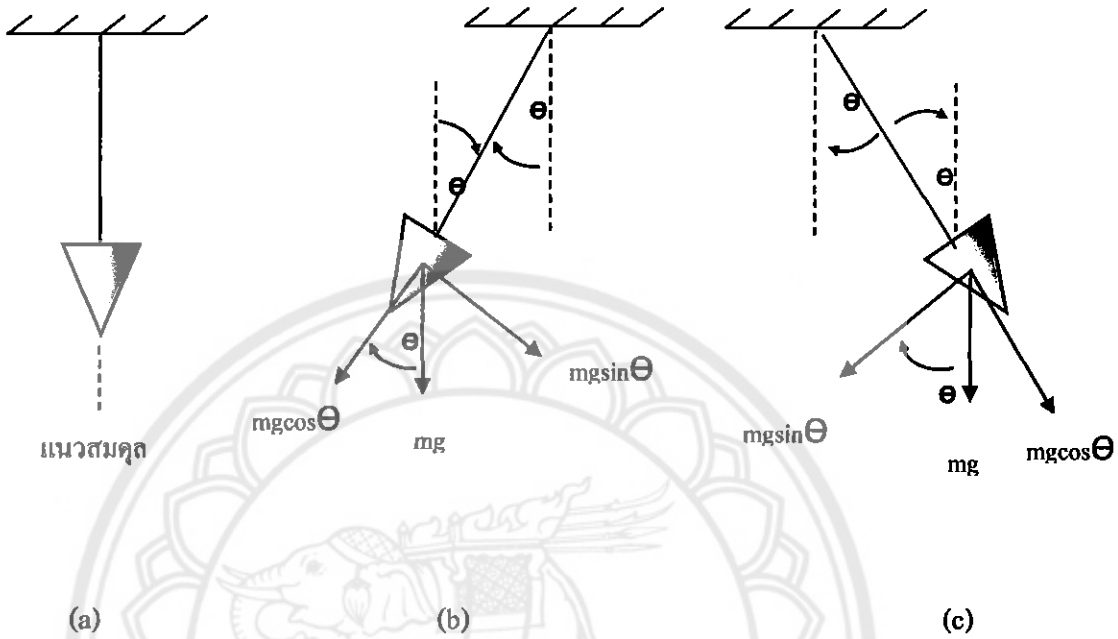
2.8.1 ลักษณะโดยทั่วไปของ MATLAB

การทำงานของ MATLAB ในขั้นตอนต่างๆ สามารถเขียนเป็นลำดับต่อเนื่องกันในแฟ้มหรือไฟล์ (File) ต่างหาก เรียกว่า M – File เพราะจะต่อท้ายชื่อแฟ้มด้วยจุดเอ็ม (.m) แล้วนำมาใช้โดยการเรียกชื่อ M – File นั้นๆ ซึ่งจะทำให้สามารถบันทึกการทำงานทั้งหมดออกมาในรูปของโปรแกรม ที่สามารถนำมาใช้ใหม่ได้อีกเรื่อยๆ เช่นเดียวกับโปรแกรมที่เขียนไว้ในภาษาอื่นๆ เช่น ฟอร์แทรน, C หรือป่าสคาล (Pascal) MATLAB มาตรฐานจะมี M – Files ที่ใช้งานประเภทต่างๆ กันรวมมาเป็นชุดเครื่องมือ (Toolbox) ส่วน Toolbox ที่มีผู้จัดทำไว้แล้วมีมากมาย เช่น Statistical Toolbox, Optimization Toolbox, Neural Network Toolbox, Financial Toolbox และอื่นๆ ซึ่งจะหาได้จากบริษัท Mathworks การแสดงผลใน MATLAB สามารถแสดงออกมาได้ในลักษณะทั้ง กราฟ 2 มิติ และ 3 มิติ สีต่างๆ และรูปภาพ ซึ่งจะเป็นประโยชน์ในการสื่อความหมายกับผู้ใช้ข้อมูล ซึ่งถือว่าเป็นคุณสมบัติที่เด่น เนื่องจาก

การเขียนโปรแกรมโดยใช้ภาษาต่างๆ มักจะขาดเครื่องมือแสดงผลที่เห็นได้ชัดเจนนอกจากนี้ยังสามารถใช้ MATLAB ร่วมกับภาษาคอมพิวเตอร์อื่นๆ เช่น FORTRAN และ C++ หากมีความจำเป็น แต่มักเป็นเรื่องยุ่งยาก เพราะจำเป็นที่จะต้องมีความเชี่ยวชาญในการเขียนโปรแกรมภาษานั้นๆ ด้วยซึ่งหากโปรแกรมภาษานั้นๆ ไม่มีขนาดที่ใหญ่เกินไปก็อาจจะนำมาเขียนใหม่ให้เป็นโปรแกรมภาษา MATLAB ให้หมดในสมัยหลัง MATLAB มีความสามารถด้านรูปภาพ (Graphical Capabilities) เพิ่มขึ้นจึงเพิ่มเติมด้านการติดต่อกับผู้ใช้ (User Interface) เข้าไปโดยจะมีการสร้างแผนภาพ เพื่อที่จะป้อนข้อมูล หรือเปลี่ยนแปลงข้อมูลโดยใช้ช่องเติมข้อมูล (Message Box) ปุ่มกด (Button) ปุ่มลาก (Slider) และอื่นๆ เข้าช่วยซึ่งจะเหมาะกับโปรแกรมที่ใช้อยู่เสมอๆ โดยผู้ที่มีความคุ้นเคยกับการเขียนโปรแกรมค่อนข้างน้อย แต่ก็จะมักทำให้ความเร็วของการทำงานลดลง เพราะการทำงานด้านรูปภาพจะเพิ่มภาระการทำงานให้กับไมโครโปรเซสเซอร์ของระบบประมวลผลส่วนกลาง (Central Processing Unit, CPU) สำหรับด้านการจำลองการทำงานด้วยคอมพิวเตอร์ มีการใช้โปรแกรม Simulink ซึ่งสามารถที่จะเขียนโปรแกรมในลักษณะเป็นหน่วย (Unit) ซึ่งแทนด้วยรูปภาพของก้อนวัตถุ (Block Diagram) แล้วนำข้อมูลจากตัวแปรต่างๆ ของ MATLAB, Built – In Functions และ M – Files มาใช้ในด้านคำนวณได้ แต่เนื่องจากยังมีการใช้ที่ยุ่งยากเพราะต้องประสมประสานทักษะในหลายๆ ด้าน และการทำงานค่อนข้างช้า จึงอยู่ในระหว่างการปรับปรุงให้มีประสิทธิภาพดีขึ้นลักษณะอื่นๆ ที่จะมีประโยชน์กับผู้ใช้ในระดับสูง ได้แก่ การแปลอัตโนมัติจากภาษา MATLAB ให้เป็นภาษา C++ ซึ่งจะทำให้สามารถนำ M – Files ที่เขียนขึ้นไว้แล้วไปสร้างเป็นซอฟท์แวร์อิสระ (Stand – Alone Application) ได้ต่อไป

2.9 การเคลื่อนที่ของลูกตุ้ม

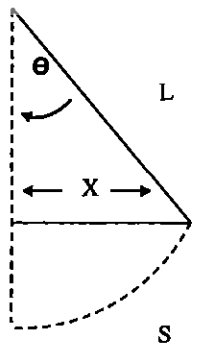
2.9.1 การเคลื่อนที่ของลูกตุ้ม



รูปที่ 2.7 การสั่นของลูกตุ้ม

เมื่อลูกตุ้มเคลื่อนที่ออกจากแนวสมดุลในรูป (a) จะมีแรงดึงกลับที่ทำให้วัตถุรักษาสภาพเดิมไว้ ในที่นี้คือแรง mgsinΘ เมื่อ Θ เป็นมุมระหว่างแนวเชือกกับแนวดิ่งและแรงที่กระทำกับวัตถุในที่นี้คือ –mgsinΘ

2.9.2 ความสัมพันธ์ของความเร็วเชิงมุมหรือความเร่งเชิงมุม



จาก $-mg\sin\Theta = -m\omega^2 s$

$g\sin\Theta = \omega^2 s$

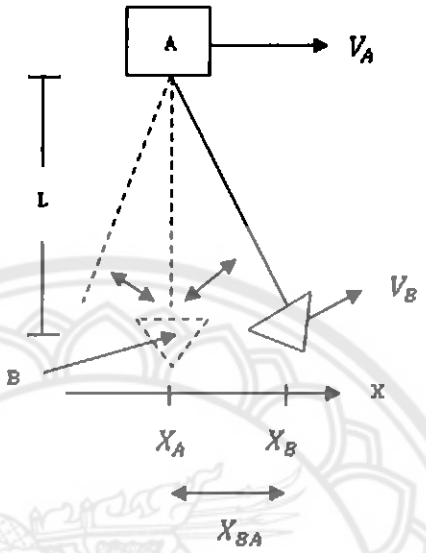
จากรูป ถ้า Θ เป็นมุม เล็กๆ $s = x$

และ $\frac{X}{Y} = \sin\Theta$

จึงได้ว่าเมื่อเป็นมุมเล็กๆ $\frac{X}{Y} = \sin\Theta$

หรือ $g\frac{S}{L} = \omega^2 s$ ดังนั้น $\omega = \sqrt{\frac{g}{L}}$

หากการสั่นมีการหน่วงเนื่องจากการมีความสูญเสียในระบบ ก็จะมีเทอมหน่วง $2\zeta\omega_n \frac{dx}{dt}$ ด้วย โดยจะได้ว่า สมการการแกว่ง มีรูปแบบ คือ



เนื่องจากแรงดันไฟฟ้าแปรผันตรงกับความเร็วมอเตอร์

กำหนดให้ $u(t) = k \cdot v_A$ (2.1)

A = มอเตอร์

B = ลูกตุ้ม

ζ = Damping Coefficient (ค่าสัมประสิทธิ์)

ω_n = ความถี่เชิงมุมธรรมชาติของการแกว่ง $\sqrt{\frac{g}{L}}$

จากสมการการแกว่งของวัตถุ แบบมี Damping

$$\ddot{x}_{BA} + 2\zeta\omega_n\dot{x}_{BA} + \omega^2{}_n x_{BA} = 0 \quad (2.2)$$

x_{BA} คือ ตำแหน่งของ B เทียบกับ $A = x_B - x_A$

$\dot{x}_{BA} = \dot{x}_B - \dot{x}_A = v_B - v_A$ = ความเร็วของ B เทียบกับ A

$\ddot{x}_{BA} = \dot{v}_B - \dot{v}_A$ = ความเร่งของ B เทียบกับ A

หากกำหนดให้ แรงดันขาเข้าเป็นแบบค่าคงที่ $(\dot{u}(t) = \dot{v}_A = 0)$ จะได้ว่า (2.2) เขียนได้เป็น

$$\dot{v}_B - \dot{v}_A + 2\zeta\omega_n(v_B - v_A) + \omega^2{}_n x_{BA} = 0 \tag{2.3}$$

จาก (2.1) และ (2.3) กำหนดตัวแปรสถานะคือ

$$x = \begin{bmatrix} x_A \\ x_{BA} \\ v_B \end{bmatrix}$$

จะได้ว่า

$$\dot{x}_A = v_A = \frac{1}{k} \cdot u(t) \tag{2.4}$$

$$\dot{x}_{BA} = v_B - v_A = v - \frac{1}{k} \cdot u(t) \tag{2.5}$$

$$\dot{v}_B = -2\zeta\omega_n(v_B - v_A) + \omega^2{}_n . x_{BA}$$

$$\dot{v}_B = -\omega^2{}_n . x_{BA} - 2\zeta\omega_n v_B + \frac{2\zeta\omega_n}{k} \cdot u(t) \tag{2.6}$$

จาก (2.4)-(2.6) เขียนเป็นสมการสถานะได้ คือ

$$\frac{d}{dt}\begin{bmatrix} x_A \\ x_{BA} \\ v_B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega^2{}_n & -2\zeta\omega_n \end{bmatrix}\begin{bmatrix} x_A \\ x_{BA} \\ v_B \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 2\zeta\omega_n \end{bmatrix} u(t) \tag{2.7}$$

บทที่ 3

วิธีดำเนินโครงงาน

3.1 ศึกษา และเก็บข้อมูล

3.1.1 ศึกษาการใช้งานฟังก์ชั่นพื้นฐานโปรแกรม Microcontroller Arduino และเครื่องมือและอุปกรณ์ในการปฏิบัติงาน และการเชื่อมต่อระบบไฟฟ้า คณะผู้ดำเนินโครงงานวิศวกรรมได้ทำการศึกษาการเขียนโปรแกรมลงในบอร์ด Microcontroller Arduino เพิ่มเติมจากหนังสือการใช้งาน Arduino เพื่อนำไปใช้ในการเขียนโปรแกรม และทำการสร้างโครงสร้างขึ้น เพื่อทำเป็นรางให้มอเตอร์เคลื่อนที่ โดยจะติดตัวเคลื่อนที่ คือ ลูกตุ้ม เพื่อสังเกตการสั่น

3.1.2 ศึกษาวิธีดำเนินงาน จัดทำโครงสร้างมอเตอร์ และการเขียนโปรแกรม Microcontroller Arduino ในส่วนของรูปอุปกรณ์ผู้ดำเนินโครงงานได้นำไปไว้ในส่วนของภาคผนวก

3.1.2.1 จัดทำโครงสร้างของระบบสาธิต

3.1.2.2 จัดทำรางมอเตอร์

3.1.2.3 สร้างบอร์ดสวิตช์

3.1.2.4 การเชื่อมต่อของระบบไฟฟ้ากับอุปกรณ์

3.1.2.5 เขียนโปรแกรมลงในบอร์ด Microcontroller Arduino

3.1.2.6 บันทึกผล และวิเคราะห์ผลการทดลอง

3.2 ออกแบบการทดลอง

ออกแบบขั้นตอน กระบวนการ วิธีการทดลอง

3.3 เขียนโปรแกรม

3.3.1 เขียนโปรแกรมโดยใช้ Microcontroller Arduino ในการควบคุมระบบ และใช้โปรแกรม MATLAB ในการคำนวณหาผลลัพธ์ที่ได้จากการทดลอง

3.3.2 ในส่วนของโค้ดโปรแกรมผู้ดำเนินโครงงานได้นำไปไว้ในส่วนของภาคผนวก

3.4 ทดสอบและปรับปรุง

ทดสอบการใช้งานของอุปกรณ์ว่ามีข้อบกพร่องประการใด แล้วนำมาปรับปรุง

3.5 ประเมินผล

ประเมินผลโดยอาจารย์ผู้สอนรายวิชา

3.6 สรุปและจัดทำรูปเล่มโครงงาน

ประเมินผลโดยอาจารย์ผู้สอนรายวิชา นำข้อมูลที่ได้จากการทดลองมาวิเคราะห์ถึงความแตกต่างและสาเหตุของการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

ในการสร้างระบบสาธิตควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่มีอุปกรณ์มากมายทั้งในส่วนของซอฟต์แวร์และฮาร์ดแวร์ ดังนี้

3.7 อุปกรณ์ในการดำเนินงาน

3.7.1 มอเตอร์ไฟตรง 24 V

3.7.2 เฟืองสะพาน

3.7.3 เฟืองขบ

3.7.4 หม้อแปลง 5V

3.7.5 แหล่งจ่ายไฟตรง 24 V

3.7.6 คอมพิวเตอร์

3.7.7 ชุดลูกตุ้มและตัวเคลื่อนที่

3.7.8 วงจรขับเคลื่อนมอเตอร์ ET-OPTO DC MOTOR

3.7.9 บอร์ด Arduino

3.7.10 โครงเหล็ก

3.7.11 รางเหล็ก

3.7.12 ไม้บรรทัด

3.7.13 ลิมิตสวิตช์

3.7.14 สายไฟ

3.7.15 สวิตช์

3.7.16 แผงวงจร

3.8 การเชื่อมต่ออุปกรณ์

รูปที่ 3.1 การเชื่อมต่ออุปกรณ์

3.9 สร้างอุปกรณ์ในการทดลอง

3.9.1 สั่งซื้อมอเตอร์ไฟตรง 24 V

3.9.2 สั่งซื้อเฟืองสะพานความยาว 60 เซนติเมตร

3.9.3 สั่งซื้อเฟืองขบ ขนาด 30 ฟังร่อง รูแกน 10 มิลลิเมตร

3.9.4 สั่งซื้อบอร์ด Arduino

3.9.5 สั่งซื้อแหล่งจ่ายไฟตรง 24 V

3.9.6 สั่งซื้อตัวแปลงไฟ 5 Amp

3.9.7 สั่งซื้อบอร์ด ET-OPTO DC MOTOR

3..9.8 เชื่อมโครงเหล็กเข้าด้วยกัน

3.9.9 สร้างรางเลื่อนไว้รองรับส่วนท้ายของมอเตอร์

3.9.10 ต่อบอร์ดสวิตช์

3.9.11 ติดตั้งลิมิตสวิตช์

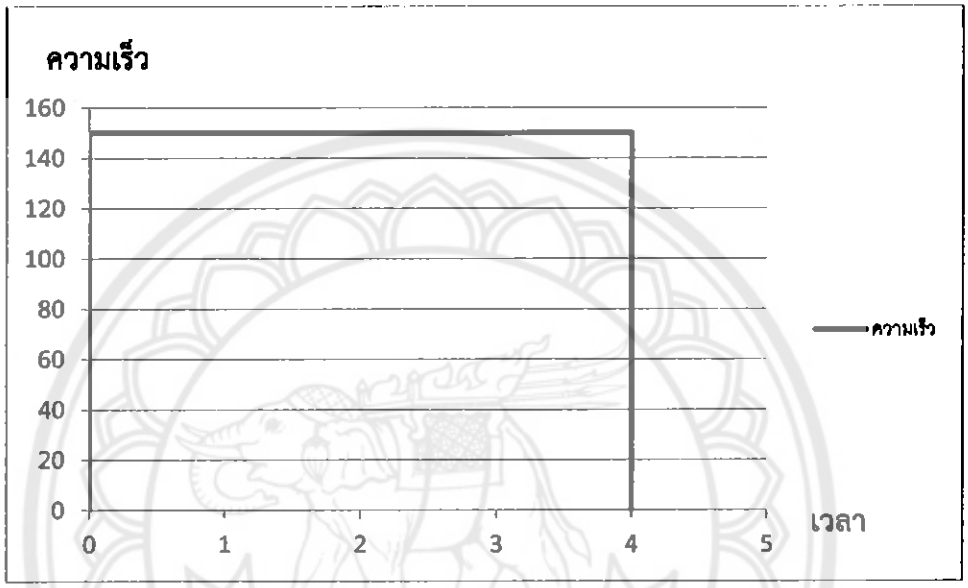
3.9.12 ต่อสายไฟเข้ากับอุปกรณ์ทุกตัว

3.10 ขั้นตอนการทดลองโดยแบ่งการทดลองเป็น 3 แบบ

ในการทดลองระบบสาธิตการควบคุมการเคลื่อนที่ของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่ ผู้ทำการทดลองได้เขียนโปรแกรมให้มอเตอร์หาจุดเริ่มต้น ถ้าต่ออุปกรณ์ครบระบบมอเตอร์ไม่อยู่ในตำแหน่งเริ่มต้น โปรแกรมจะสั่งให้มอเตอร์เคลื่อนที่ไม่ยังจุดเริ่มต้น โดยจะแบ่งการทดลองเป็น 3 รูปแบบ แต่ละรูปแบบมีขอบเขตเวลา คือ ใช้เวลาเคลื่อนที่จากจุดเริ่มต้นไปยังจุดหมายใช้เวลา 4 วินาที โดยประมาณ ซึ่งเป็นระยะเวลาที่ได้จากการเคลื่อนที่ด้วยความเร็วเฉลี่ยร้อยละ 50 ของความเร็วเต็มกำลังของมอเตอร์ คือ 150

3.10.1 การทดลองแบบที่ 1

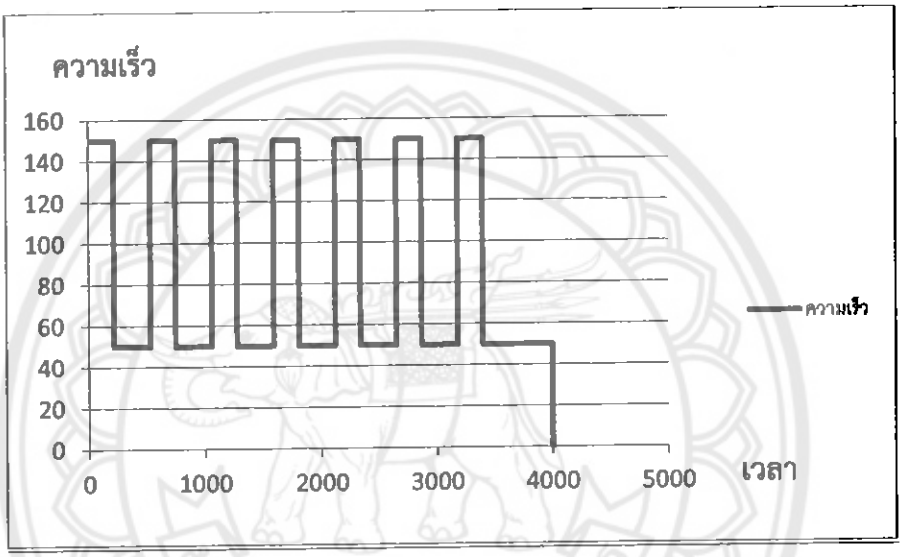
คณะผู้ดำเนินโครงงานวิศวกรรมได้เขียนโปรแกรมลงในบอร์ด Arduino ให้เมื่อกดสวิตช์ 1 มอเตอร์เคลื่อนที่ด้วยความเร็วคงที่จากจุดเริ่มต้นไปยังจุดหมาย โดยใช้ความเร็วเฉลี่ยร้อยละ 50 ของความเร็วเต็มกำลังของมอเตอร์ คือ 150 ใช้เวลา 4 วินาทีโดยประมานไปถึงจุดหมาย ดังรูปที่ 3.2



รูปที่ 3.2 กราฟแสดงความเร็วของมอเตอร์ในการทดลองที่ 1

3.10.2 การทดลองแบบที่ 2

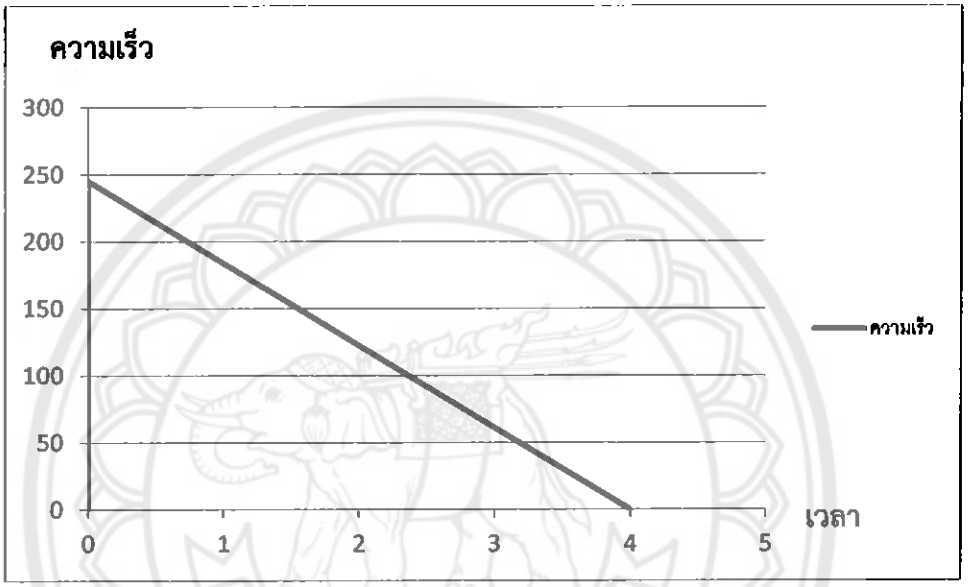
คณะผู้ดำเนินโครงงานวิศวกรรมได้เขียนโปรแกรมลงในบอร์ด Arduino ให้เมื่อกดสวิตช์ 2 มอเตอร์วิ่งด้วยความเร็วไม่คงที่จากจุดเริ่มต้นไปถึงจุดหมาย คือ ช้าบ้างเร็วบ้างสลับกันไป โดยกำหนดให้วิ่งด้วยความเร็ว 150 เป็นระยะเวลา 215 msec และวิ่งด้วยความเร็ว 50 เป็นระยะเวลา 215 msec ใช้เวลาจากจุดเริ่มต้นไปถึงจุดหมาย 4 วินาที โดยประมาณ ดังรูปที่ 3.3



รูปที่ 3.3 กราฟแสดงความเร็วของมอเตอร์ในการทดลองที่ 2

3.10.3 การทดลองแบบที่ 3

คณะผู้ดำเนินโครงงานวิศวกรรมได้เขียนโปรแกรมลงในบอร์ด Arduino ให้เมื่อกดสวิตช์ 3 มอเตอร์วิ่งด้วยความเร็วเกือบเต็มกำลัง จากนั้นสั่งให้ลดความเร็วลงเรื่อยๆ คือ ใช้ความเร็ว 245 แล้วลดลงที่ละ 0.49 จนหยุดนิ่งที่จุดหมาย โดยใช้เวลา 4 วินาที โดยประมาณ ดังรูปที่ 3.4



รูปที่ 3.4 ความเร็วของมอเตอร์ในการทดลองที่ 3

3.10.4 สังเกตการสั่นของลูกตุ้มขณะมอเตอร์หยุดเคลื่อนที่ที่จุดหมาย

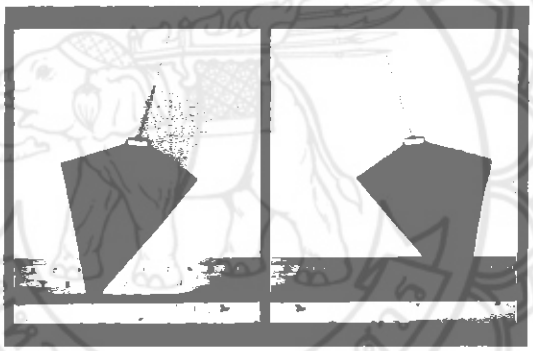
สังเกตการแกว่งในแต่ละรูปแบบของการทดลอง ว่าลูกตุ้มมีการแกว่งมากน้อยเพียงใด

บทที่ 4

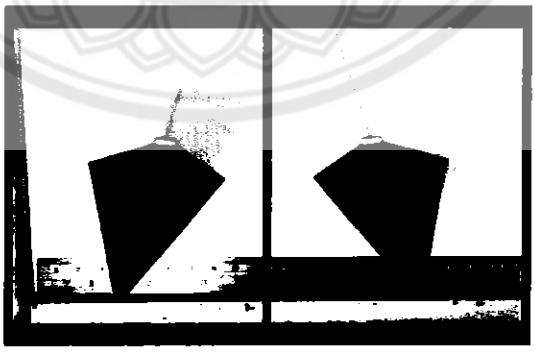
ผลการทดลอง

4.1 ผลการทดลองรูปแบบที่ 1

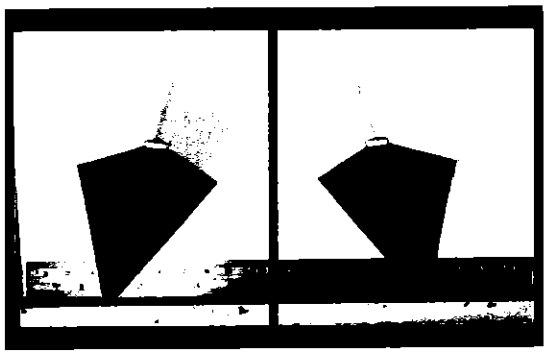
จากการทดลองในบทที่ 3 เมื่อกดสวิตช์ที่ 1 การเคลื่อนที่ของลูกตุ้ม ด้วยความเร็วคงที่จากจุดเริ่มต้นไปยังจุดหมาย ผลที่ได้ คือ การแกว่งของลูกตุ้มขณะที่มอเตอร์หยุดเคลื่อนที่นั้นมีการแกว่งที่มากที่สุดจากทั้ง 3 การทดลอง เนื่องจากความเร่งสุดท้ายก่อนจะหยุดนิ่งนั้น มีความเร่งมากที่สุด ซึ่งระยะการแกว่งของลูกตุ้มนั้นแสดงได้ดังตาราง 4.1



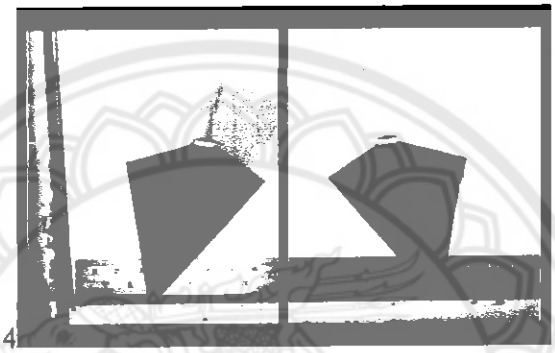
รูปที่ 4.1 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 1



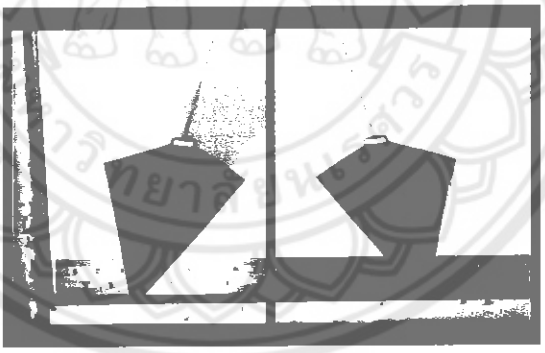
รูปที่ 4.2 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 2



รูปที่ 4.3 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 3



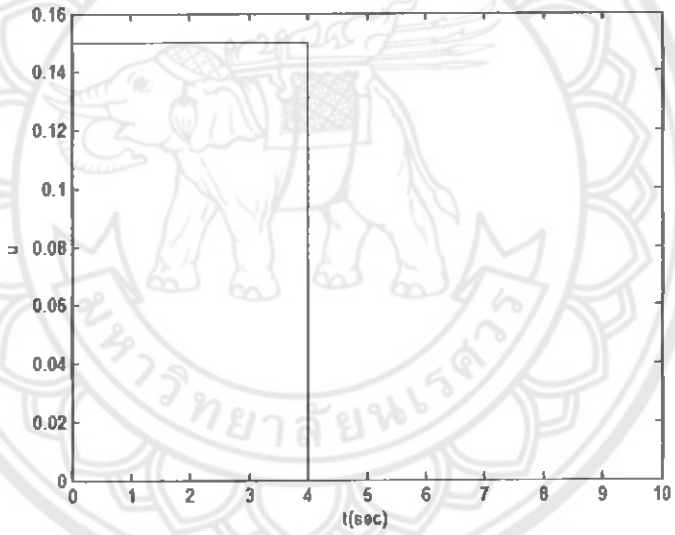
รูปที่ 4.4 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 4



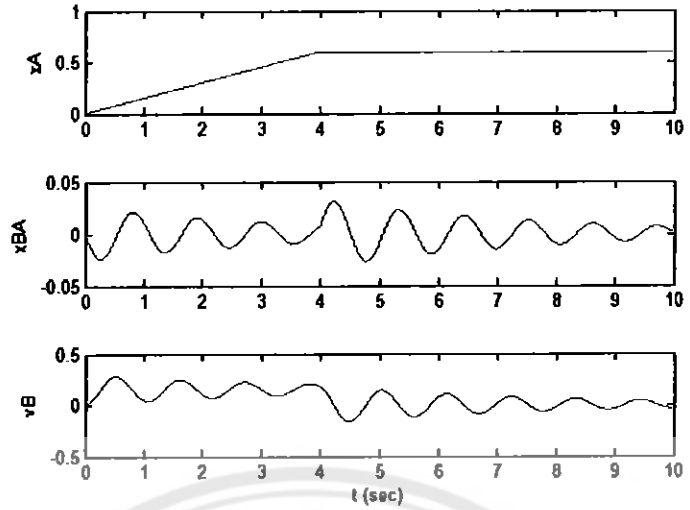
รูปที่ 4.5 การแกว่งของลูกตุ้มในการทดลองแบบที่ 1 ครั้งที่ 5

ตารางที่ 4.1 ตารางแสดงระยะการแกว่งของลูกตุ้มในการทดลองแบบที่ 4

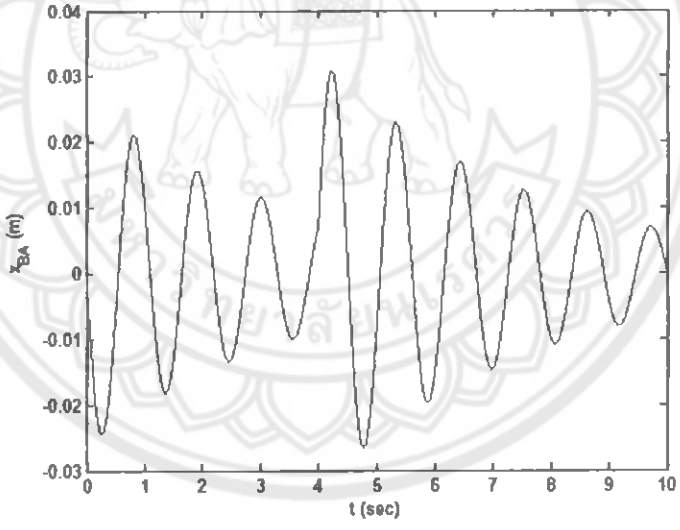
ครั้งที่	ระยะการแกว่งไปทางซ้าย	ระยะการแกว่งไปทางขวา
1	6.7 cm	6.8 cm
2	6.6 cm	6.7 cm
3	6.6 cm	6.6 cm
4	6.7 cm	6.8 cm
5	6.6 cm	6.7 cm
เฉลี่ย	6.64 cm	6.72 cm



รูปที่ 4.6 กราฟความเร็วเทียบกับเวลาของการทดลองแบบที่ 1



รูปที่ 4.7 กราฟแสดงระยะทาง แอมพลิจูดการสั่น และความเร็วของลูกตุ้ม ในการทดลองแบบที่ 1



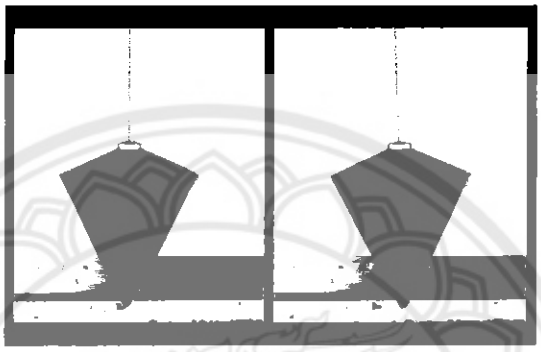
รูปที่ 4.8 กราฟแสดงแอมพลิจูดการสั่นของลูกตุ้ม ในการทดลองแบบที่ 1

4.1.1 สรุปผลการทดลองรูปแบบที่ 1

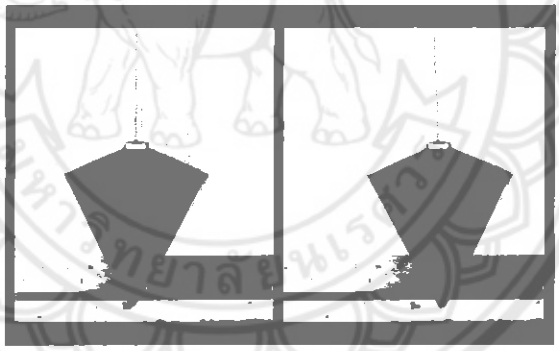
จากการทดลองในแบบที่ 1 ผู้ทำการทดลองได้นำค่าเงื่อนไขอินพุตของการทดลองไปจำลองในโปรแกรมแมทแลบดังในภาคผนวก ได้ผลการจำลองดังรูป 4.3 และ 4.4 จากตารางที่ 4.1 จะพบว่าผลการทดลองกับทฤษฎีสอดคล้องกัน แต่มีค่าที่คลาดเคลื่อนไปบ้าง เมื่อเทียบกับรูปที่ 4.3 แอมพลิจูดของการแกว่งของลูกตุ้มนั้นมีค่าไม่เท่ากัน และจากรูปที่ 4.4 แสดงว่าการเคลื่อนที่ด้วยความเร็วคงที่จากจุดเริ่มต้นไปยังจุดหมายนั้น ขณะที่มอเตอร์หยุดนิ่งพบว่าลูกตุ้มมีการแกว่งที่มากที่สุด

4.2 ผลการทดลองรูปแบบที่ 2

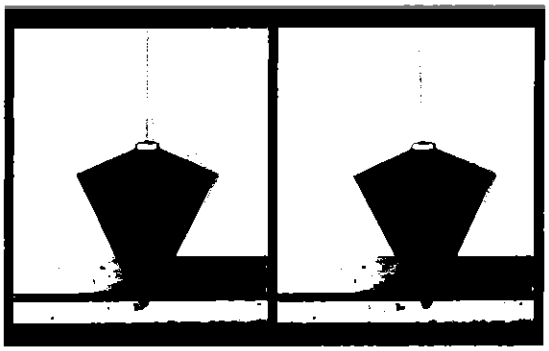
จากการทดลองในบทที่ 3 เมื่อกดสวิตช์ที่ 2 การเคลื่อนที่ของลูกตุ้ม ด้วยความเร็วไม่คงที่จากจุดเริ่มต้นไปยังจุดหมาย โดยใช้ความเร็วช้าบ้าง เร็วบ้างสลับกันไป ผลที่ได้ คือ การแกว่งของลูกตุ้ม ขณะที่มอเตอร์หยุดเคลื่อนที่นั้นมีการแกว่งที่น้อยที่สุดจากทั้ง 3 การทดลอง เนื่องจากความเร่งสุดท้ายก่อนจะหยุดนิ่งนั้น มีความเร่งน้อยที่สุด ซึ่งระยะการแกว่งของลูกตุ้มนั้นแสดงได้ดังตาราง 4.2



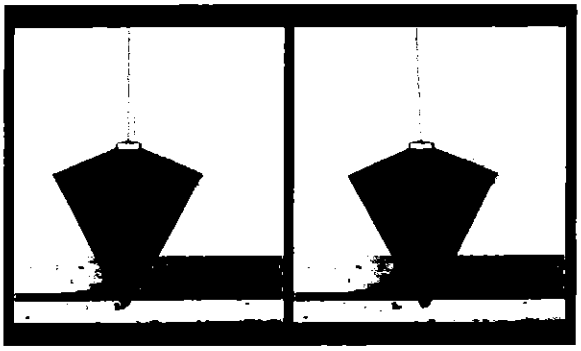
รูปที่ 4.9 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 1



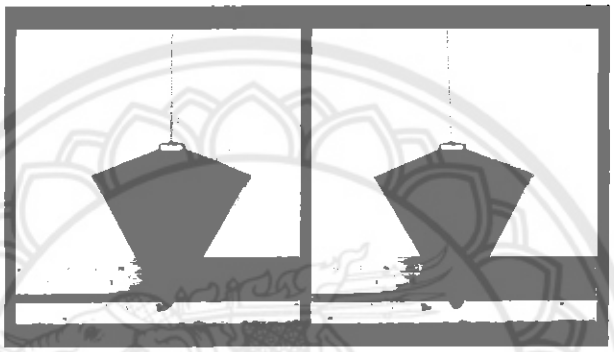
รูปที่ 4.10 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 2



รูปที่ 4.11 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 3



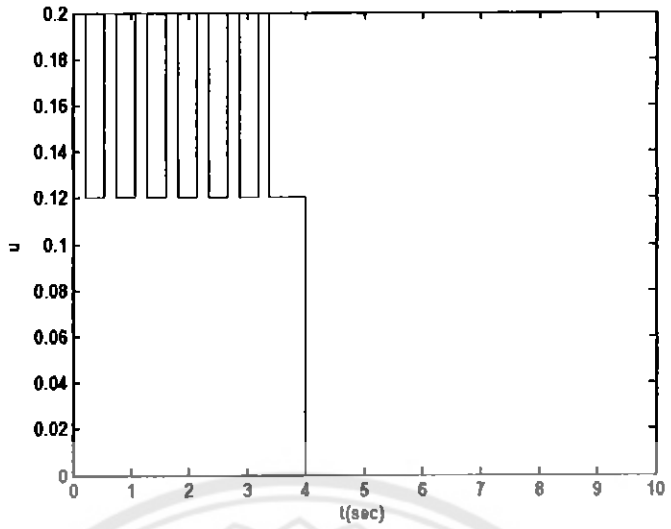
รูปที่ 4.12 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 4



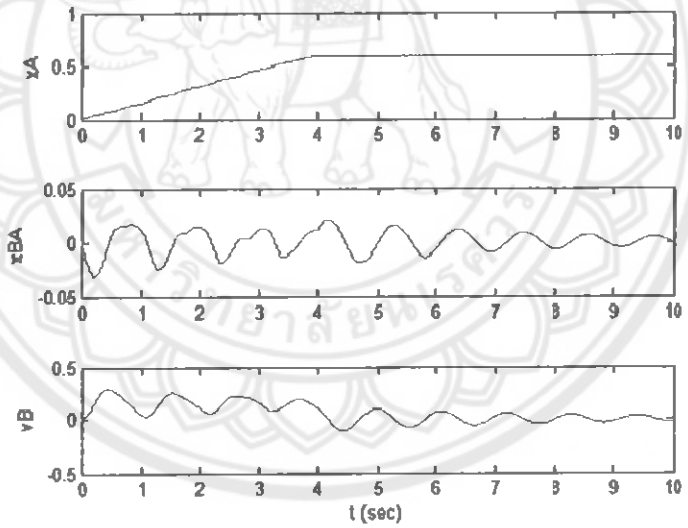
รูปที่ 4.13 การแกว่งของลูกตุ้มในการทดลองแบบที่ 2 ครั้งที่ 5

ตารางที่ 4.2 ตารางแสดงระยะการแกว่งของลูกตุ้มในการทดลองแบบที่ 2

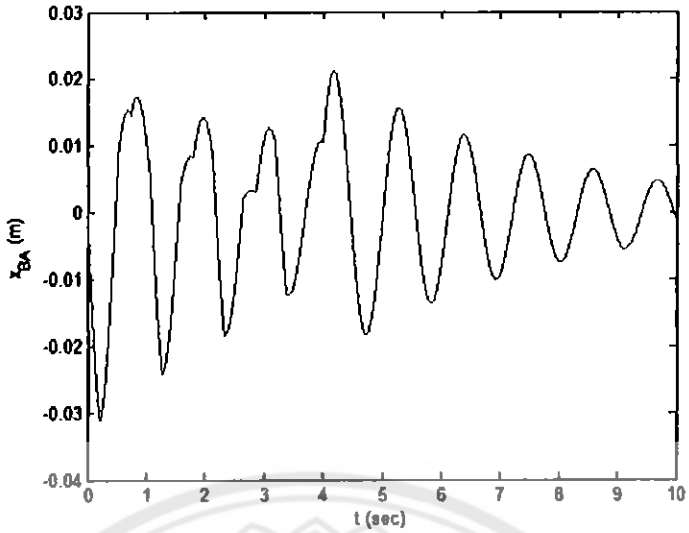
ครั้งที่	ระยะการแกว่งไปทางซ้าย	ระยะการแกว่งไปทางขวา
1	1 cm	1 cm
2	0.8 cm	1 cm
3	1 cm	1 cm
4	1 cm	0.8 cm
5	1 cm	1 cm
เฉลี่ย	0.96 cm	0.96 cm



รูปที่ 4.14 กราฟความเร็วเทียบกับเวลาของการทดลองแบบที่ 2



รูปที่ 4.15 กราฟแสดงระยะทาง แอมพลิจูดการสั่น และความเร็วของลูกตุ้ม ในการทดลองแบบที่ 2



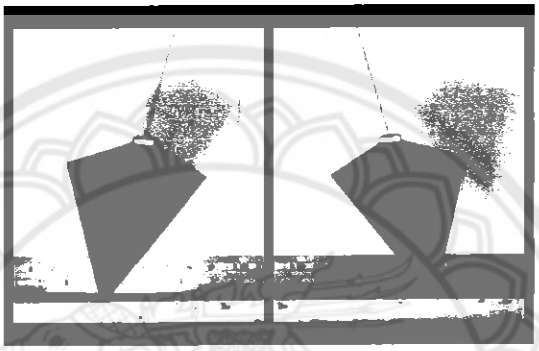
รูปที่ 4.16 กราฟแสดงแอมพลิจูดการสั่นของลูกตุ้ม ในการทดลองแบบที่ 2

4.2.1 สรุปผลการทดลองแบบที่ 2

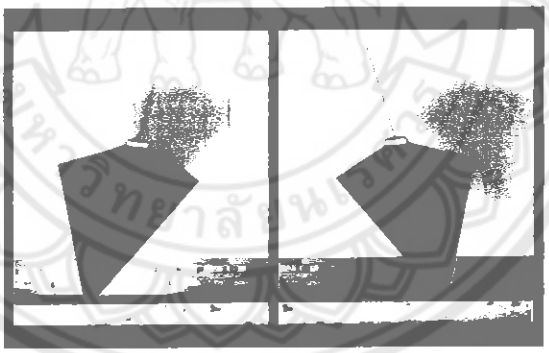
จากการทดลองในแบบที่ 2 ผู้ทำการทดลองได้นำค่าเงื่อนไขอินพุตของการทดลองไปจำลองในโปรแกรมแมทแลบดังในภาคผนวก ได้ผลการจำลองดังรูป 4.7 และ 4.8 จากตารางที่ 4.8 จะพบว่าผลการทดลองกับทฤษฎีมีความสอดคล้องกันอยู่ โดยมีค่าที่คลาดเคลื่อนไปบ้าง เมื่อเทียบกับรูปที่ 4.7 แอมพลิจูดของการแกว่งของลูกตุ้มนั้นมีค่าไม่เท่ากัน จากรูปที่ 3.3 เทียบกับรูปที่ 4.6 จะพบว่าเราจะต้องเปลี่ยนค่าความเร็วในการคำนวณให้สอดคล้องกับผลการทดลอง คือ ให้ได้ระยะทางการเคลื่อนที่ไป ตรงกับความเป็นจริง คือ 0.6 เมตร และจากรูปที่ 4.8 แสดงให้เห็นว่าการทดลองในแบบที่ 2 มีแอมพลิจูดของการแกว่งที่น้อยที่สุด

4.3 ผลการทดลองรูปแบบที่ 3

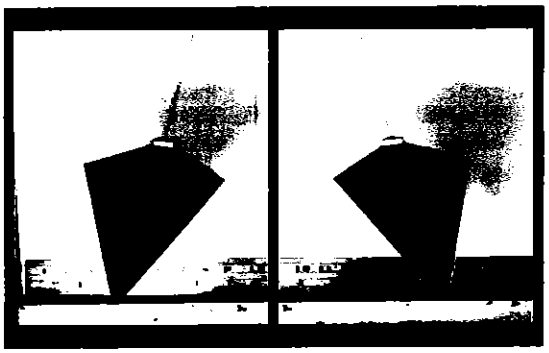
จากการทดลองในบทที่ 3 เมื่อกดสวิตช์ที่ 3 การเคลื่อนที่ของมอเตอร์ด้วยความเร็วไม่คงที่จากจุดเริ่มต้นไปยังจุดหมาย คือ เริ่มเคลื่อนที่ด้วยความเร็วเกือบเต็มกำลังจากนั้นค่อยๆ ลดความเร็วลงเรื่อยๆ จนหยุดนิ่ง ผลที่ได้ คือ การแกว่งของลูกตุ้มขณะที่มอเตอร์หยุดเคลื่อนที่นั้นมีการแกว่งที่น้อยกว่าการทดลองที่ 1 แต่มากกว่าการทดลองที่ 2 เนื่องจากก่อนถึงจุดหยุดนิ่งยังคงมีความเร่งอยู่ ซึ่งระยะการแกว่งของลูกตุ้มนั้นแสดงได้ดังตาราง 4.3



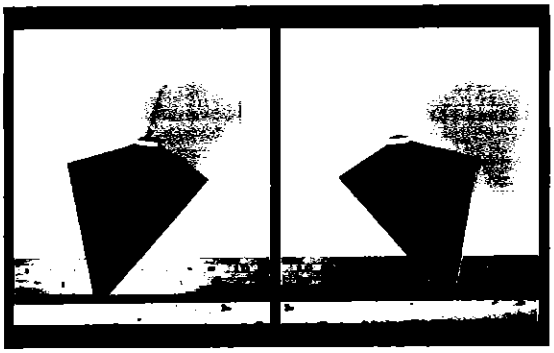
รูปที่ 4.17 การแกว่งของลูกตุ้มในการทดลองแบบ 3 ครั้งที่ 1



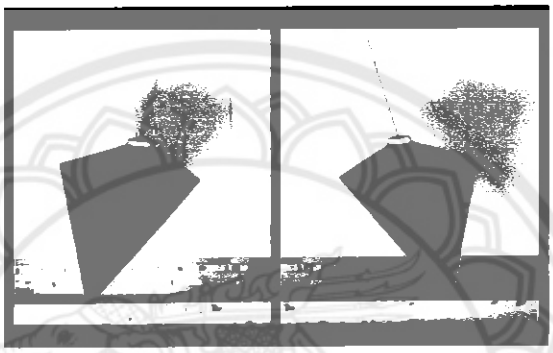
รูปที่ 4.18 การแกว่งของลูกตุ้มในการทดลองแบบ 3 ครั้งที่ 2



รูปที่ 4.19 การแกว่งของลูกตุ้มในการทดลองแบบ 3 ครั้งที่ 3



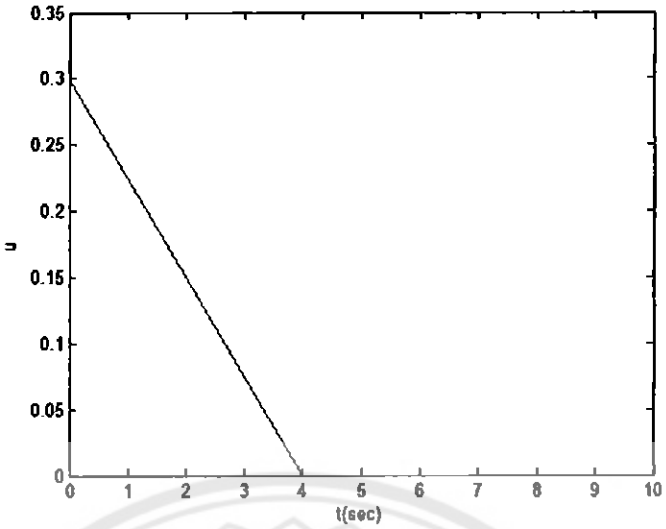
รูปที่ 4.20 การแกว่งของลูกตุ้มในการทดลองแบบ 3 ครั้งที่ 4



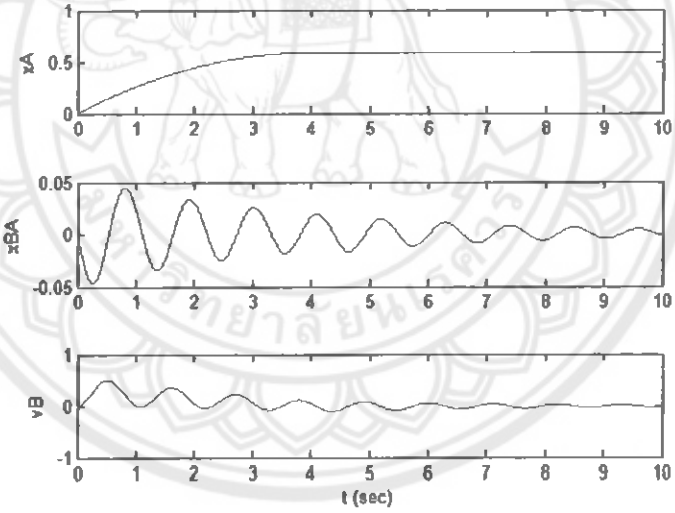
รูปที่ 4.21 การแกว่งของลูกตุ้มในการทดลองแบบ 3 ครั้งที่ 5

ตารางที่ 4.3 ตารางแสดงระยะการแกว่งของลูกตุ้มในการทดลองแบบที่ 3

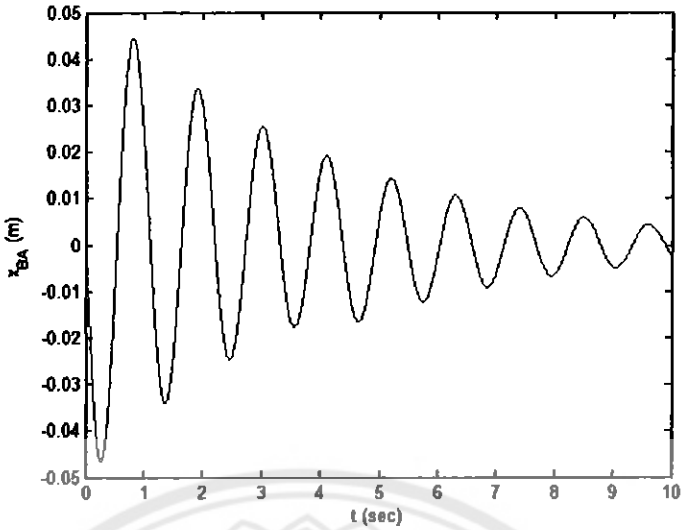
ครั้งที่	ระยะการแกว่งไปทางซ้าย	ระยะการแกว่งไปทางขวา
1	5.6 cm	5.7 cm
2	5.8 cm	6 cm
3	5.6 cm	5.8 cm
4	5.8 cm	5.6 cm
5	5.7 cm	5.7 cm
เฉลี่ย	5.7 cm	5.76 cm



รูปที่ 4.22 กราฟความเร็วเทียบกับเวลาของการทดลองแบบที่ 3



รูปที่ 4.23 กราฟแสดงระยะทาง แอมพลิจูดการสั่น และความเร็วของลูกตุ้ม ในการทดลองแบบที่ 3

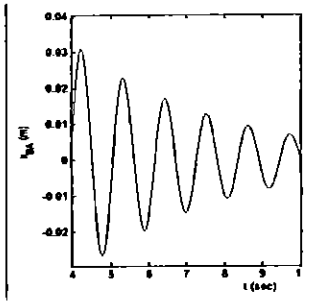


รูปที่ 4.24 กราฟแสดงแอมพลิจูดการสั่นของลูกตุ้ม ในการทดลองแบบที่ 3

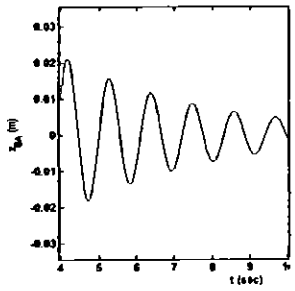
4.3.1 สรุปผลการทดลองแบบที่ 3

จากการทดลองในแบบที่ 3 ผู้ทำการทดลองได้นำค่าเงื่อนไขอินพุตของการทดลองไปจำลองในโปรแกรมแมทแลบ ดังในภาคผนวก ได้ผลการจำลองดังรูป 4.11 และ 4.12 จากตารางที่ 4.3 จะพบว่าผลการทดลองกับทฤษฎีมีความสอดคล้องกัน แต่มีค่าที่คลาดเคลื่อนไปบ้าง เมื่อเทียบกับรูปที่ 4.11 แอมพลิจูดของการแกว่งของลูกตุ้มนั้นมีค่าไม่เท่ากัน จากรูปที่ 3.4 เทียบกับรูปที่ 4.10 พบว่าเราต้องเปลี่ยนค่าความเร็วเพื่อให้ทฤษฎีมีความสอดคล้องกับค่าความจริงในการทดลอง ให้ได้ระยะการเคลื่อนที่จริง คือ 0.6 เมตร และจากรูปที่ 4.12 แสดงให้เห็นว่าการทดลองในแบบที่ 3 มีแอมพลิจูดของการแกว่งที่น้อยกว่าการทดลองแบบที่ 1 แต่มากกว่าแบบที่ 2

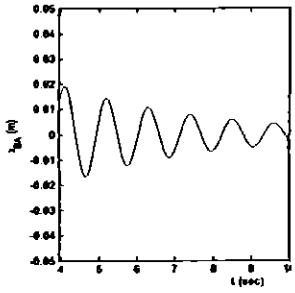
4.4 เปรียบเทียบผลการทดลอง



รูปที่ 4.25 แอมพลิจูดการทดลองที่ 1



รูปที่ 4.26 แอมพลิจูดการทดลองที่ 2



รูปที่ 4.27 แอมพลิจูดการทดลองที่ 3

ตารางที่ 4.4 ตารางแสดงแอมพลิจูดสูงสุดของการแกว่งของลูกตุ้ม

แอมพลิจูดการสั่นครั้งที่	แอมพลิจูดสูงสุด
การทดลองที่ 1	0.0308
การทดลองที่ 2	0.0211
การทดลองที่ 3	0.0192

จากการเปรียบเทียบผลการคำนวณกับผลการทดลอง พบว่ามีความสอดคล้องกัน คือ หลังจากที่มอเตอร์หยุดนิ่งในช่วงวินาทีที่ 4 พบว่า แอมพลิจูดสูงสุดของการแกว่งในการทดลองที่ 1 มีค่าที่มากที่สุด คือ 3.08 เซนติเมตร แต่เมื่อเปลี่ยนรูปแบบความเร็วในการเคลื่อนที่เป็นแบบที่ 2 คือ 2.11 เซนติเมตร และแบบที่ 3 คือ 1.92 เซนติเมตร พบว่าแอมพลิจูดสูงสุดของการแกว่งของลูกตุ้มนั้นลดลง เมื่อเวลาผ่านไปแอมพลิจูดของการแกว่งนั้นก็ค่อยๆ ลดลงด้วยเช่นกัน ดังรูปที่ 4.13 รูปที่ 4.14 และรูปที่ 4.15 แสดงให้เห็นว่าความเร็วในการเคลื่อนที่ของมอเตอร์มีผลต่อการแกว่งของลูกตุ้ม

บทที่ 5

สรุปผลของการทดลอง และปัญหาที่พบ

5.1 สรุปผลการทดลอง

ในการดำเนินโครงงานนี้ผู้ดำเนินโครงงานมีจุดประสงค์ที่จะสร้างระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่ จากนั้นศึกษาความแตกต่างของการสั่นของลูกตุ้ม โดยในโครงงานนี้ได้เลือกใช้ Microcontroller Arduino เป็นตัวควบคุมการเคลื่อนที่ของมอเตอร์โดยแบ่งการควบคุมเป็น 3 แบบ คือ การให้มอเตอร์เคลื่อนที่แบบความเร็วคงที่ การให้มอเตอร์เคลื่อนที่แบบเร็วบ้างช้าบ้าง และการให้มอเตอร์เคลื่อนที่ด้วยความเร็วสูงสุดแล้วค่อยๆ ลดความเร็วจนถึงจุดหมาย จากนั้นวิเคราะห์ผลการทดลอง ในการวิเคราะห์ผลการทดลองผู้ดำเนินโครงงานได้เลือกใช้โปรแกรม MATLAB เป็นตัวช่วยคำนวณค่าที่ได้ แสดงผลออกมาเป็นกราฟ

จากการทดลองพบว่าแอมพลิจูดของการสั่นของลูกตุ้มในการเคลื่อนที่แบบเร็วบ้างช้าบ้าง มีค่าน้อยที่สุด โดยผลการทดลองนี้สอดคล้องกับการคำนวณที่ได้จากการแก้สมการสถานะด้วยระเบียบวิธีเชิงเลข

5.2 ปัญหาที่พบ คือ

- พบการวิ่งตกรางของเฟืองซึ่งทำให้เกิดการสั่นของมอเตอร์ขณะเคลื่อนที่ไปบนเฟืองสะพาน ผู้ทำการทดลองจึงต้องปรับปรุงรางเหล็ก โดยยึดรางเหล็กติดกับตัวมอเตอร์ และเชื่อมเฟืองสะพานเข้ากับฐานเพื่อไม่ให้เกิดการตกรางของมอเตอร์ขณะเคลื่อนที่

- ตัวต้านทานบนบอร์ด ET-OPTO DC MOTOR DRIVER 6-24V/5A เกิดการช็อต จึงต้องนำตัวต้านทาน 5 โอห์ม 10 วัตต์ มาต่อเข้าไปแทนที่ตัวต้านทานเดิม

- โครงเหล็กมีลักษณะที่ไม่เที่ยงตรง คือ มีการคดงอ ผู้ทำการทดลองจึงต้องมาดัดเพื่อให้ได้ระนาบพอที่จะทำการทดลอง ให้มอเตอร์เคลื่อนที่ไปได้โดยไม่เกิดการติด

- พอร์ท USB ไม่รองรับกับคอมพิวเตอร์ที่ใช้ จึงต้องเปลี่ยนคอมพิวเตอร์จนกว่าพอร์ท USB จะรองรับ เนื่องจากถ้าพอร์ท USB ไม่รองรับกับคอมพิวเตอร์จะทำให้ไม่สามารถซิมิวเลตโปรแกรมลงไปยังบอร์ด Arduino ได้

- การกำหนดทิศทางการวิ่งของมอเตอร์ และตำแหน่งที่จะหยุด ในช่วงแรกมีปัญหา คือ มอเตอร์วิ่งเลยระยะราง หรือเมื่อเริ่มการทดลองมอเตอร์ไม่อยู่ในจุดเริ่มต้น ผู้ทำการทดลองจึงได้เพิ่มลิมิตสวิทช์เพื่อเป็นตัวป้อนกลับสัญญาณมาสั่งให้มอเตอร์หยุด และได้เขียนโปรแกรมเมื่อเริ่มต้นไม่ว่ามอเตอร์อยู่ในตำแหน่งใดต้องเคลื่อนที่กลับมายังตำแหน่งเริ่มต้น

- สมการที่ได้ไม่สามารถคำนวณได้ด้วยมือจึงต้องใช้โปรแกรม MATLAB ในการช่วยคำนวณ โดยการพล็อตกราฟ ออกมาเป็นค่า

- ผลการวิเคราะห์ต่างจากการทดลอง เนื่องจากเวลาที่จับนั้นไม่มีความละเอียดพอ ความเร็วคลาดเคลื่อน เฟืองของระบบมีความไม่สม่ำเสมอ และลูกตุ้มแกว่งแบบ 2 มิติไม่ใช่ 1 มิติ

5.3 การพัฒนา และปรับปรุงแก้ไข

- ต้องมีความละเอียดในการจับเวลา
- เพิ่มรูปแบบการทดลอง คือ อาจเพิ่มรูปแบบการเคลื่อนที่ของมอเตอร์
- มีการวัดการสั่นของที่ตุ้มที่ละเอียดขึ้น

ภาคผนวก ก

1. บอร์ด ET-OPTO DC MOTOR

รูปที่ 1 ET-OPTO DC MOTOR

1.1 คุณสมบัติทั่วไปของบอร์ด ET-OPTO DC MOTOR DRIVER 6-24 V/5 A

เป็นบอร์ดใช้งานอิสระ หรือต่อเข้ากับบอร์ด Microcontroller ต่างๆ ใช้ควบคุมการทำงานของตัวมอเตอร์ไฟตรงให้หมุน ซ้าย ขวา และควบคุมความเร็วของตัวมอเตอร์ไฟตรงออกแบบใช้กับมอเตอร์ไฟตรงกระแสสูงๆได้

- ใช้ Power มอสเฟส N-Channel เบอร์ RFP50N06 ขนาด 60 V/50 A จำนวน 4 ตัวในการใช้งาน

- ใช้ 5 Pin ต่อควบคุมจากบอร์ดภายนอก DIR1, DIR2, ENA

- มีวงจรภายในต่อใช้งานอิสระโดยตรง ไม่ต้องใช้บอร์ดควบคุม โดยมี 2 Switch สั่งหมุน ซ้าย, ขวา และ VR ปรับความเร็วหรือใช้ Pulse Logic จากไมโครคอนโทรลเลอร์ควบคุมการทำงานได้เองเมื่อเลือก Jumper ให้ควบคุมผ่านการ Control

- สามารถต่อกับมอเตอร์ไฟตรงได้ 1 ตัว ขนาดไม่เกิน 24 V DC กระแส ไม่เกิน 5 Amp (สามารถใช้กับกระแสได้สูงกว่านี้ โดยย้ายวงจร Mofet ออกภายนอกต่อสายตรง)

- Power Supply ใช้งานวงจร 5 V DC 2 Pin

- PCB Size 12.7x5.6 เซนติเมตร การทำงานของวงจร

- ใช้มอสเฟสทำหน้าที่เป็นตัว Drive สามารถใช้ขับมอเตอร์ไฟตรงได้ตั้งแต่ 6 V-24 V ที่กระแส 5 Amp โดยประมาณ

- แบ่งแหล่งจ่ายไฟเลี้ยงบอร์ดออกเป็น 2 ชุด คือ แหล่งจ่ายไฟตรง 5 V สำหรับไอซีและแหล่งจ่ายไฟตรง 6 V – 24 V สำหรับเลี้ยงมอเตอร์ไฟตรงโดยตรง

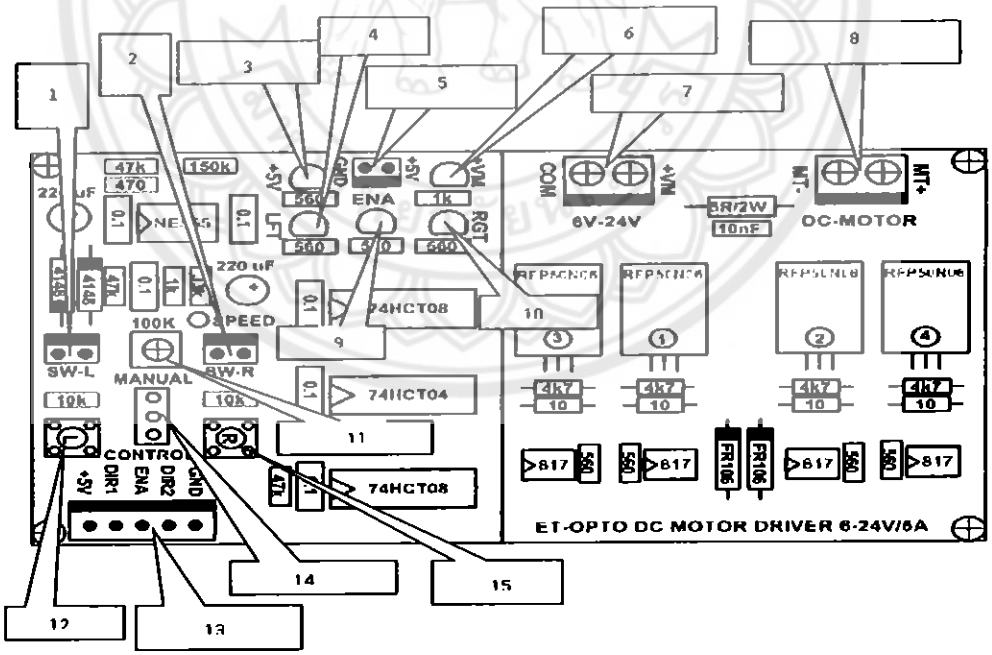
- ใช้ไอซีออปโต้เป็นตัวแยกกราวด์ ระหว่างแหล่งจ่ายทั้ง 2 ชุด เพื่อป้องกันความเสียหายของอุปกรณ์บนบอร์ด

- ใช้เป็นตัวแยกกราวด์ ระหว่างแหล่งจ่ายทั้ง 2 ชุด เพื่อป้องกันความเสียหายของอุปกรณ์บนบอร์ด

- สามารถควบคุมความเร็วของมอเตอร์ไฟตรงด้วยสัญญาณ PWM ได้จากบอร์ดโดยตรง หรือจะเลือกส่งสัญญาณ PWM จากภายนอกบอร์ดเข้ามาควบคุมความเร็วแทนก็ได้

- สามารถเปลี่ยนทิศทางการหมุนของมอเตอร์ไฟตรงได้จาก Switch Left, Switch Right ที่ต่ออยู่บนบอร์ด และมีขั้วสำหรับต่อสวิตช์จากภายนอกเข้ามาควบคุมแทนสวิตช์บนบอร์ดก็ได้

- มีขั้วต่อ 5 PIN สำหรับใช้ต่อกับไมโครคอนโทรลเลอร์ เพื่อใช้ส่งสัญญาณภายนอกมาควบคุมทิศทาง และความเร็วการหมุน ของมอเตอร์ไฟตรงได้โดยตรง และสามารถควบคุม การหยุดของมอเตอร์ไฟตรงได้ 2 แบบ คือ แบบ Fast Stop และแบบ Slow Stop



รูปที่ 2 แสดงโครงสร้างบอร์ด ET-OPTO DC MOTOR DRIVER 6-24 V/5 A

1.2 คุณสมบัติแต่ละหมายเลข

1.2.1 หมายเลข 1 Connector Switch Left

เป็นขั้วต่อสวิตช์ควบคุมการหมุนซ้ายของมอเตอร์ไฟตรงซึ่งทำไว้ให้ผู้ใช้สามารถต่อสวิตช์ ได้โดยตรงจากภายนอกเข้ามาควบคุมการหมุนของมอเตอร์ เมื่อไม่ต้องการใช้งานสวิตช์ที่จัดไว้ให้บนบอร์ด โดยขั้วต่อนี้จะต่อขนานอยู่กับ Switch Left

1.2.2 หมายเลข 2 Connector Switch Right

เป็นขั้วต่อสวิตช์ควบคุมการหมุนขวาของมอเตอร์ไฟตรงซึ่งทำไว้ให้ผู้ใช้สามารถต่อสวิตช์ ได้โดยตรงจากภายนอกเข้ามาควบคุมการหมุนของมอเตอร์ เมื่อไม่ต้องการใช้งานสวิตช์ที่จัดไว้ให้บนบอร์ด โดยขั้วต่อนี้จะต่อขนานอยู่กับ Switch Right

1.2.3 หมายเลข 3 หลอดไฟ LED Vin 5 V

จะเป็นหลอดไฟ LED แสดงสถานการณ์ทำงานของแหล่งจ่ายไฟตรง +5V ที่ป้อนเข้ามาทางขั้วต่อ Vin 5 V ถ้าหลอดไฟ LED นี้ติดแสดงว่ามีแรงดันมาเลี้ยงในส่วนของไอซีที่อยู่บนบอร์ดแล้ว

1.2.4 หมายเลข 4 หลอดไฟ LED LFT

จะเป็นหลอดไฟ LED แสดงทิศทางการหมุนของมอเตอร์ว่ามีทิศการหมุนทางด้านซ้าย (ทวนเข็ม)

1.2.5 หมายเลข 5 Connector Vin 5 V

เป็น Connector สำหรับต่อไฟ +5 V ภายนอกบอร์ดเข้ามาเลี้ยงในส่วนของไอซี

1.2.6 หมายเลข 6 หลอดไฟ LED VM 6 V-24 V

เป็นหลอดไฟ LED แสดงสถานการณ์ทำงานของแหล่งจ่ายไฟเลี้ยงมอเตอร์ไฟตรงที่ป้อนเข้ามาทางขั้วต่อ VM 6 V-24 V

1.2.7 หมายเลข 7 Connector VM 6 V-24 V

เป็นขั้วต่อสำหรับต่อไฟตรง 6 V-24 V ภายนอกบอร์ดเข้ามาเลี้ยงมอเตอร์ไฟตรงโดยค่าแรงดันต่อเข้ามานั้นจะขึ้นอยู่กับ ค่าแรงดันการใช้งานของมอเตอร์ไฟตรงที่ผู้ใช้นำมาทำการต่อ

1.2.8 หมายเลข 8 Connector มอเตอร์ไฟตรง

เป็นขั้วสำหรับต่อมอเตอร์ไฟตรงโดยควรต่อบวกของมอเตอร์เข้าที่ MT+ และขั้วลบเข้าที่ MT- หรือจะกลับขั้วกันก็ได้ แต่จะทำให้หลอดไฟ LED ที่ใช้แสดงทิศทางการหมุน แสดงผิดไปจากความเป็นจริงมอเตอร์ไฟตรงที่นำมาต่อนั้นควรมีขนาดไม่เกิน 24 V/5 A เพื่อความปลอดภัยของชุด Driver (มอสเฟส)

1.2.9 หมายเลข 9 หลอดไฟ LED ENA

เป็นหลอดไฟ LED ที่ใช้แสดงสถานการณ์ทำงานของขา ENA สำหรับ Enable มอเตอร์ไฟตรงโดยถ้าหลอดไฟ LED นี้ดับ นั่นคือ มอเตอร์ไฟตรงจะไม่สามารถทำงานได้ แต่ถ้าหลอดไฟ LED นี้ติดแสดงว่ามอเตอร์จะทำงานทันทีเมื่อมีการส่งสัญญาณที่ขา DR1 หรือ DR2

1.2.10 หมายเลข 10 หลอดไฟ LED RGT

จะเป็นหลอดไฟ LED แสดงทิศทางการหมุนของมอเตอร์ว่ามีทิศทางการหมุนทางด้านขวา (ตามเข็ม)

1.2.11 หมายเลข 11 VR Control Speed

เป็น VR ปรับค่าสำหรับควบคุมความเร็วของมอเตอร์ไฟตรงภายในบอร์ด โดยจะมีลักษณะเป็นการปรับค่า PWM เพื่อใช้ในการควบคุมความเร็วของมอเตอร์ เมื่อจะควบคุมความเร็วจะต้อง Set Jumper มาทางด้าน Manual เวลาปรับ VR สังเกตที่หลอดไฟ LED ENA จะหรี่ลงเมื่อความเร็วของมอเตอร์ลดลง หรือสว่างขึ้นเมื่อความเร็วมอเตอร์เพิ่มขึ้น

1.2.12 หมายเลข 12 Switch Left

เป็นสวิตช์ใช้สำหรับควบคุมให้มอเตอร์หมุนมาทางด้านซ้าย โดยการกดสวิตช์ Left นี้ค้างไว้ มอเตอร์จะหมุนมาทางด้านซ้าย ถ้าปล่อยมอเตอร์ก็จะ Stop แบบ Fast

1.2.13 หมายเลข 13 Connector External Control

เป็นขั้วต่อขนาด 5 PIN ใช้สำหรับต่อสัญญาณจากภายนอก เช่น ต่อจากไมโครคอนโทรลเลอร์ มาควบคุมทิศทาง และความเร็วของมอเตอร์ไฟตรงโดยตรง ซึ่งจะต้อง Set Jumper มาทางด้าน Control

1.2.14 หมายเลข 14 Jumper Control

เป็น Jumper ใช้สำหรับเลือกการควบคุมมอเตอร์ไฟตรงว่าต้องการเลือกควบคุมจากบนบอร์ดโดยตรง หรือจะเขียนโปรแกรมจากภายนอกมาควบคุมโดยผ่านทางขั้วต่อ 5 Pin ในหมายเลข 13 ถ้าต้องการควบคุมจากบอร์ดก็ให้ Set Jumper มาทางด้าน Manual ซึ่งจะสามารถควบคุมทิศทางการหมุนได้จาก Switch Left หรือ Switch Right และควบคุมความเร็วได้จากการปรับ VR ถ้าต้องการควบคุมจากภายนอกก็ให้ Set Jumper มาทางด้าน Control ดังรูป

ก. เมื่อต้องการควบคุมมอเตอร์ไฟตรงจากบอร์ด ข. เมื่อต้องการควบคุมมอเตอร์ไฟตรงจากภายนอก

รูปที่ 3 แสดงการ Set Jumper Manual/control

1.2.15 หมายเลข 15 Switch Right

เป็นสวิตช์ใช้สำหรับควบคุมให้มอเตอร์หมุนมาทางขวาโดยการกด Switch Right นี้ค้างไว้ มอเตอร์ก็จะหมุนมาทางด้านขวา ถ้าปล่อยมอเตอร์ก็จะหยุดทันที

1.3 โหมดการใช้งาน

1.3.1 โหมด MANUAL

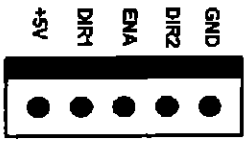
การใช้งานในโหมดนี้จะต้อง Set Jumper มาทางด้าน Manual ซึ่งในโหมดนี้จะเป็นการควบคุมการหมุน และทิศทางของมอเตอร์ไฟตรงได้จาก Switch Left และ Switch Right ที่จัดไว้ให้บน

บอร์ดได้โดยตรง รวมทั้งยังสามารถปรับความเร็วของมอเตอร์ได้ด้วยการหมุน VR 100K ที่อยู่บนบอร์ดได้อีกด้วยโดยไม่จำเป็นต้องอาศัยการควบคุมจาก ไมโครคอนโทรลเลอร์ เมื่อต้องการให้มอเตอร์หมุน ก็กดสวิตช์ใดสวิตช์หนึ่งมอเตอร์ก็จะหมุนไปตามทิศทางที่ต้องการ ในขณะที่กดสวิตช์ให้มอเตอร์หมุนอยู่นั้นผู้ใช้ก็สามารถปรับ VR เพื่อควบคุมความเร็วของมอเตอร์ให้หมุนช้าหรือเร็วได้ตามต้องการ เมื่อต้องการให้มอเตอร์หยุดหมุนก็เพียงแค่ปล่อยสวิตช์ที่กดอยู่ มอเตอร์ก็จะหยุดหมุนทันที (Fast Stop) หรือถ้าต้องการให้ Stop แบบ Slow คือ ค่อยๆ ให้มอเตอร์หยุดหมุน ก็ให้กดสวิตช์อีกตัวหนึ่งร่วมด้วย นั่นคือ สวิตช์ทั้ง 2 ตัวถูกกด ก็จะเป็นการให้มอเตอร์หยุดหมุนเช่นกันแต่จะเป็นแบบ Slow Stop

1.3.2 โหมด Control

การใช้งานในโหมดนี้จะต้อง Set Jumper มาทางด้าน Control ซึ่งในโหมดนี้จะเป็นการควบคุมความเร็วการหมุนของมอเตอร์ไฟตรงโดยใช้สัญญาณพัลส์จากภายนอก เช่น จากไมโครคอนโทรลเลอร์ ป้อนเข้ามาที่ขา ENA ส่วนทิศทางการหมุนของมอเตอร์จะยังคงใช้สวิตช์ที่อยู่บนบอร์ดได้เหมือนเดิม หรือ ถ้าต้องการใช้ไมโครคอนโทรลเลอร์ควบคุมทั้งความเร็วการหมุน และทิศทางการหมุนของมอเตอร์ โดยไม่ใช้สวิตช์ หรือ VR ที่จัดไว้ให้บนบอร์ด ก็จะต้องต่อขาสัญญาณ O/P ของไมโครคอนโทรลเลอร์เข้าที่ขา DIR, ENA และ

DIR2 ที่ขั้วต่อ 5 PIN บนบอร์ด รวมทั้งต่อ GND เข้าด้วยกันส่วนขา +5 V ที่ขั้วต่อ 5 PIN ของบอร์ด จะเป็นขาที่จ่าย หรือรับแรงดัน O/P +5 V ถ้าไมโครคอนโทรลเลอร์ที่นำมาใช้มีไฟเลี้ยงอยู่แล้วก็ไม่จำเป็นต้องใช้ขานี้ต่อเพียงขา GND ร่วมกันอย่างเดียว แต่ถ้าไม่มีก็สามารถดึงไฟเลี้ยงไปที่ไมโครคอนโทรลเลอร์ที่นำมาใช้ได้ (+5 V)



รูปที่ 4 แสดงการจัดเรียงขาขั้วต่อ 5 PIN บนบอร์ด

1.4 การทำงานของวงจร

เมื่อทำการเชื่อมต่อขาสัญญาณกับมอเตอร์ไฟตรงเรียบร้อยแล้วในการเขียนโปรแกรมผู้ใช้สามารถส่งสัญญาณมาควบคุมการหมุนของมอเตอร์ไฟตรงได้ตามตารางด้านล่าง ให้สังเกตว่า ในการป้อน PWM ควบคุมความเร็วของมอเตอร์นั้นจะต้องป้อนเข้าที่ขา ENA ถ้าไม่ต้องการควบคุมความเร็วและยังคงให้มอเตอร์หมุนทำงานได้ที่ความเร็วเต็มที่จะต้องกำหนดให้ขา ENA เป็น 1 ไว้เสมอ ถ้าขา ENA เป็น 0 เมื่อไหร่มอเตอร์จะไม่หมุน หรือ หยุดหมุน

ตารางที่ 1 แสดงการส่งสัญญาณควบคุมมอเตอร์ไฟตรง

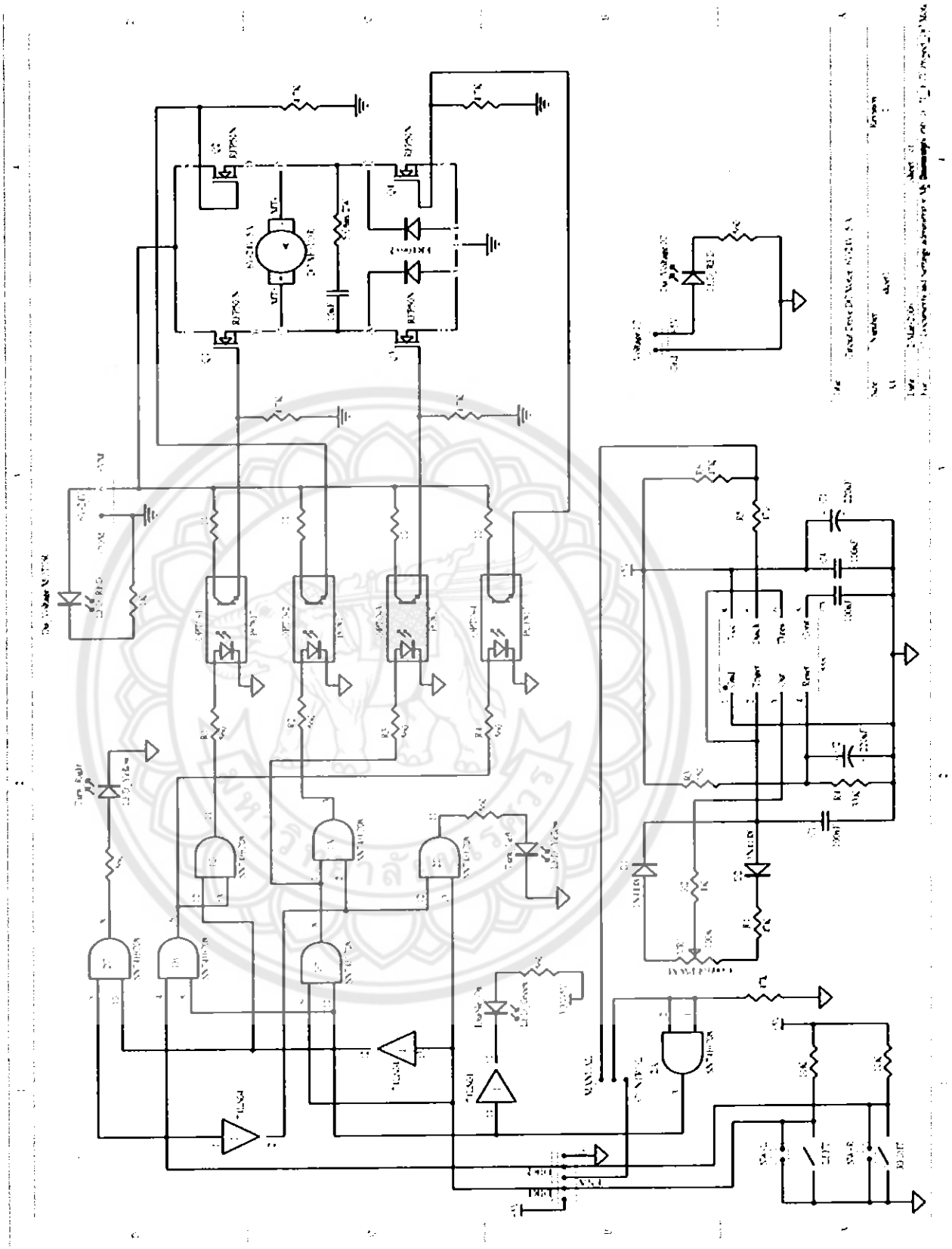
ENA/PWM	DIR1	DIR2	STATUS DC MOTOR
0	x	x	SLOW STOP
1	0	0	SLOW STOP
1	0	1	ROTATE RIGHT
1	1	0	ROTATE LEFT
1	1	1	FAST STOP

- เมื่อ ENA = 0 จะไม่สนใจสภาวะของ DIR1 และ DIR2 ทำให้ไอซีออปโต้ทั้ง 4 ตัวไม่ทำงาน จึงไม่มีแรงดันไปไบอัสให้กับขาของ Gate ของมอสเฟสทั้ง 4 ตัว ทำให้มอสเฟสไม่ทำงาน มอเตอร์ก็จะไม่หมุนหรือถ้าหมุนอยู่มอเตอร์ก็จะหยุดแบบ Slow

- เมื่อ ENA = 1 ส่วน DIR1, DIR2 = 0 แล้วจะมีลักษณะการทำงานเหมือนกับเงื่อนไงแรกเมื่อ ENA = 1 ส่วน DIR1 = 0, DIR2 = 1 จะทำให้ไอซีออปโต้ตัวที่ 1 กับ 4 ทำงานส่วนตัวที่ 2 และ 3 ไม่ทำงานส่งผลให้มีแรงดันไบอัสที่ขา Gate ของ Q1 และ Q4 ทำให้มอสเฟสคู่นี้ On มีกระแสไหลจาก Q1 ผ่านไปยังมอเตอร์ไฟตรงครบวงจรที่ Q4 ลงกราวด์ ทำให้มอเตอร์หมุนขวา (ตามเข็ม)

- เมื่อ ENA = 1 ส่วน DIR1 = 1, DIR2 = 0 จะทำให้ไอซีออปโต้ตัวที่ 2 กับ 3 ทำงาน ส่วนตัวที่ 1 และ 4 ไม่ทำงานส่งผลให้มีแรงดันไบอัสที่ขา Gate ของ Q3 และ Q4 ทำให้มอสเฟสคู่นี้ ON ทำให้มีกระแสไหลผ่าน Q2 ผ่านมอเตอร์ไฟตรงครบวงจรที่ Q3 ลงกราวด์ ทำให้มอเตอร์หมุนซ้าย (ทวนเข็มนาฬิกา)

- เมื่อ ENA = 1 ส่วน DIR1, DIR2 = 1 จะทำให้ไอซีออปโต้ตัวที่ 3 กับ 4 ทำงาน ส่วนตัวที่ 1 และ 2 ไม่ทำงานส่งผลให้มีแรงดันไบอัสที่ขา Gate ของ Q3 และ Q4 ทำให้มอสเฟสคู่นี้ ON ดึงกระแสที่ไหลผ่านมอเตอร์ไฟตรงอยู่ ลงกราวด์ทันที ทำให้มอเตอร์หยุดทันที



รูปที่ 5 วงจร ET-OPTO DC MORTOR DRIVER 6-24 V/5 A

2. เฟืองขบ และเฟืองสะพาน

ใช้เป็นส่วนที่สำหรับให้มอเตอร์เคลื่อนที่ไปบนเฟืองสะพาน โดยมีเฟืองขบเป็นตัวขับเคลื่อนมอเตอร์

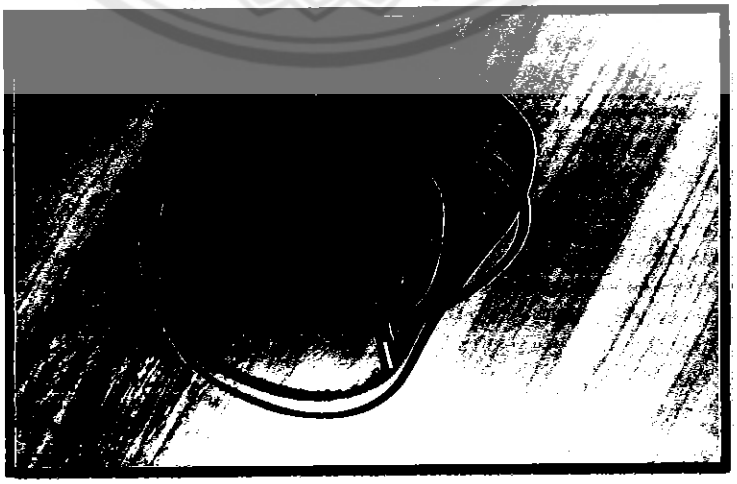
รูปที่ 6 เฟืองสะพาน และเฟืองขบ

- เฟืองขบ เป็นเฟืองเหล็กใหม่ สำหรับทดรอบมอเตอร์ให้มีกำลังมากขึ้น หรือใช้วิ่งบนเฟืองสะพาน ขนาดรูแกนมีตั้งแต่ 10 มิลลิเมตร ระยะร่องฟัน 4 มิลลิเมตร ขนาด 30 ฟัน เส้นผ่าศูนย์กลาง 3 เซนติเมตร

- เฟืองสะพาน เป็นเหล็กขนาด 60 เซนติเมตร สำหรับทำตัวขับเคลื่อนในทางราบ หรือ แนวดิ่ง เกาะแน่น ระยะระหว่างฟันเฟือง ร่องฟัน 4 มิลลิเมตร ขนาดแกน 1x1 เซนติเมตร มีเฟืองขบทั้งเหล็ก

3. หม้อแปลง 5 V

ใช้สำหรับจ่ายไฟ 5 V ให้กันบอร์ดไมโครคอนโทรลเลอร์



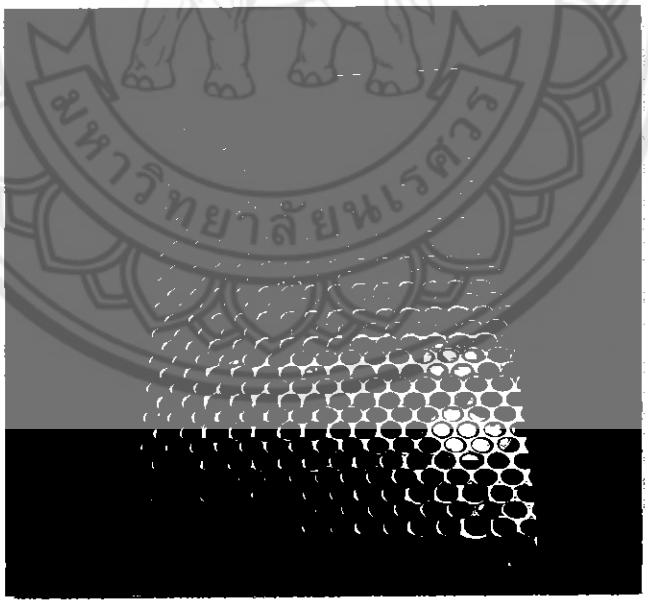
รูปที่ 7 หม้อแปลง 5 V

4. มอเตอร์ไฟตรง 24 V

รูปที่ 8 มอเตอร์ไฟตรง 24 V

5. แหล่งจ่ายไฟตรง 24 V

เนื่องจากผู้ทดลองใช้มอเตอร์ไฟตรง 24 V ในการทดลองจึงต้องใช้แหล่งจ่าย 24 V เช่นกัน

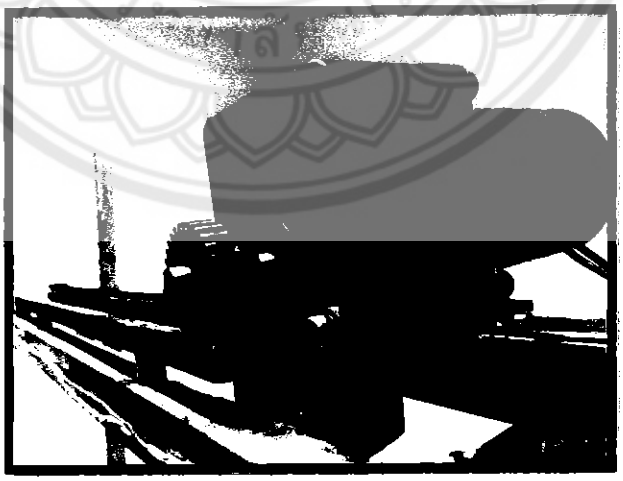


รูปที่ 9 แหล่งจ่ายไฟตรง 24 V

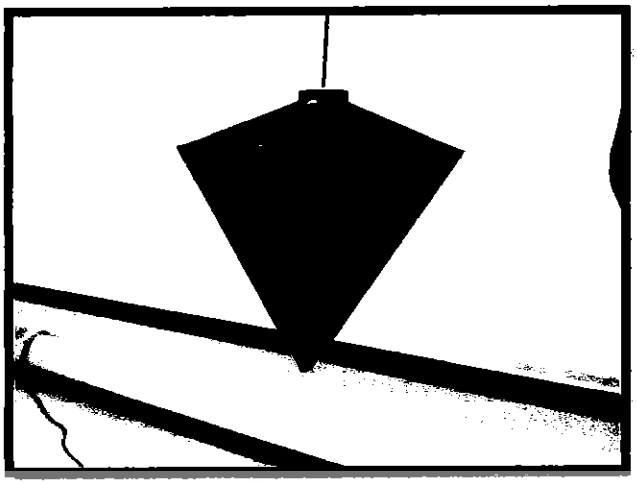
6. ชุดลูกตุ้ม และตัวเคลื่อนที่

ใช้มอเตอร์เป็นตัวเคลื่อนที่ไปบนเฟืองสะพานโดยมีเฟืองขบเป็นตัวขับเคลื่อน โดยได้ติดลูกตุ้มไว้กับมอเตอร์ ซึ่งเป็นตัวเคลื่อนที่

รูปที่ 10 มอเตอร์ และลูกตุ้ม



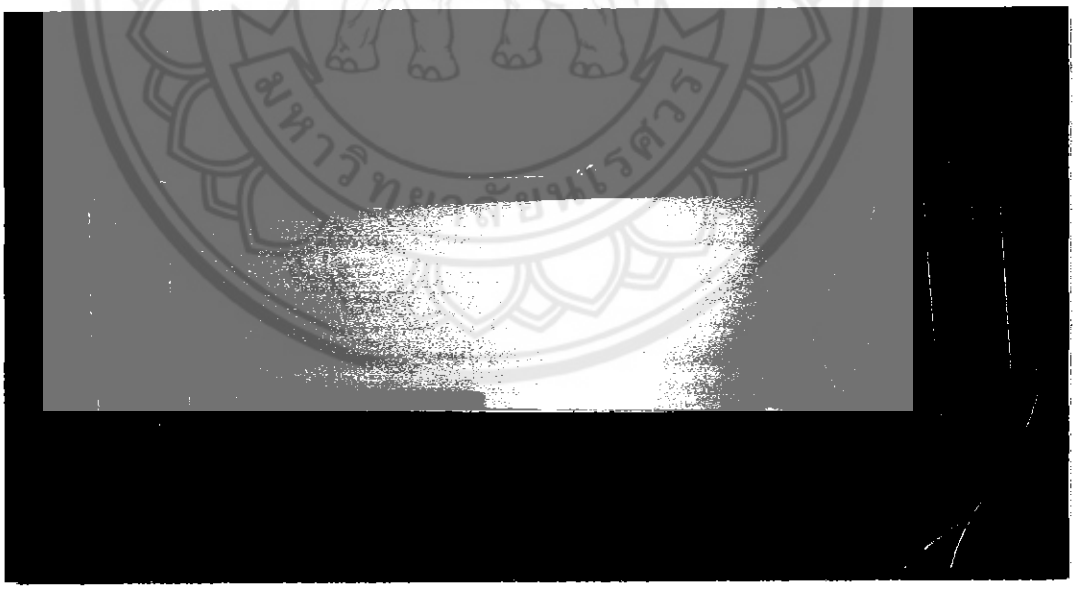
รูปที่ 11 มอเตอร์



รูปที่ 12 ลูกตุ้ม

7. โครงเหล็ก

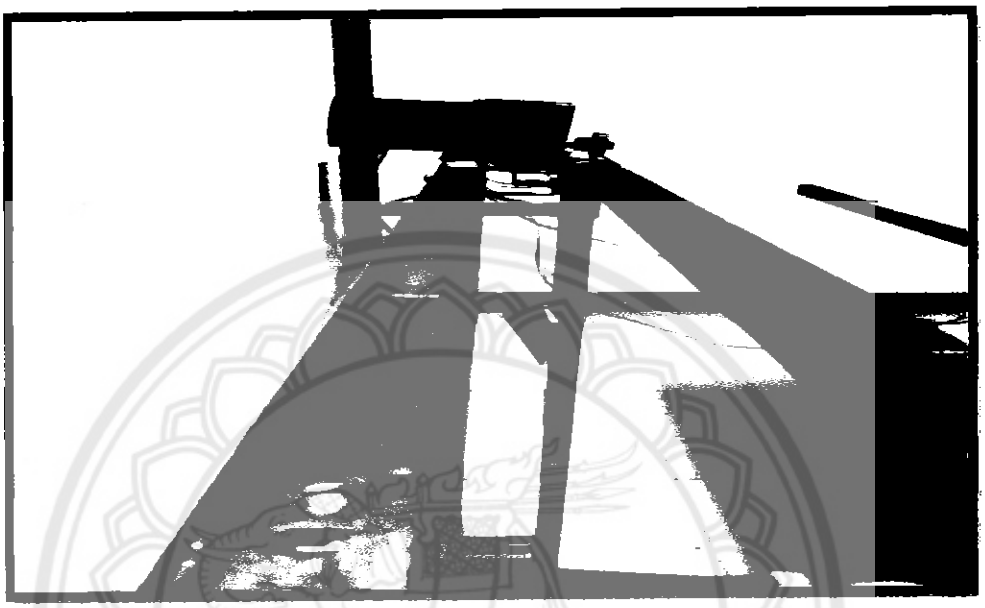
เป็นแท่งเหล็กยาวประมาณ 70 เซนติเมตร นำมาทำเป็นฐานให้เฟืองสะพานแข็งแรง และที่เหลือทำเป็นโครงสร้าง สามารถนำมาประกอบได้



รูปที่ 13 โครงเหล็กที่ใช้สำหรับติดตั้งตัวเคลื่อนที่

8. รางเหล็ก

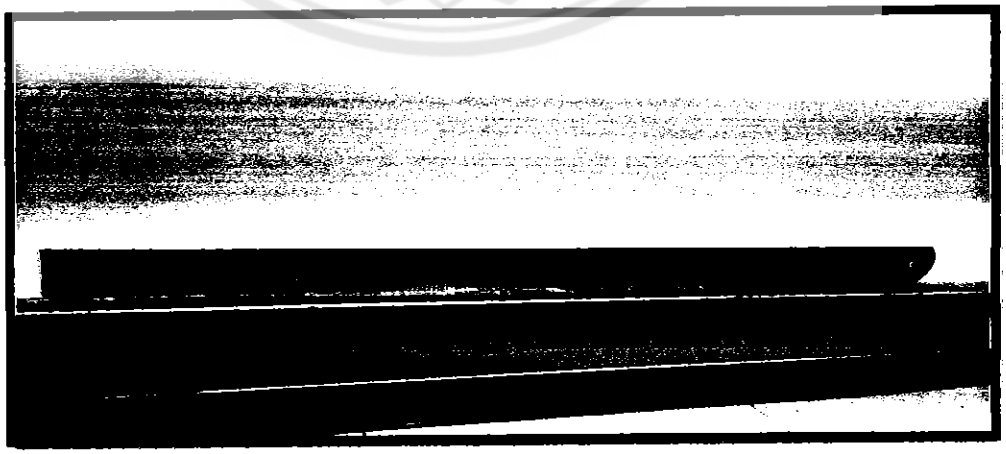
รางเหล็กนี้ใช้เป็นส่วนยึดมอเตอร์เพื่อในขณะขับเคลื่อนที่นั้นมอเตอร์หมุนทำให้เฟืองขบไม่ตกเฟืองสะพาน เพราะถ้าเฟืองขบตกร่องเฟืองสะพานจะทำให้เกิดการสั่นขึ้น



รูปที่ 14 รางเหล็ก

9. ไม้บรรทัด

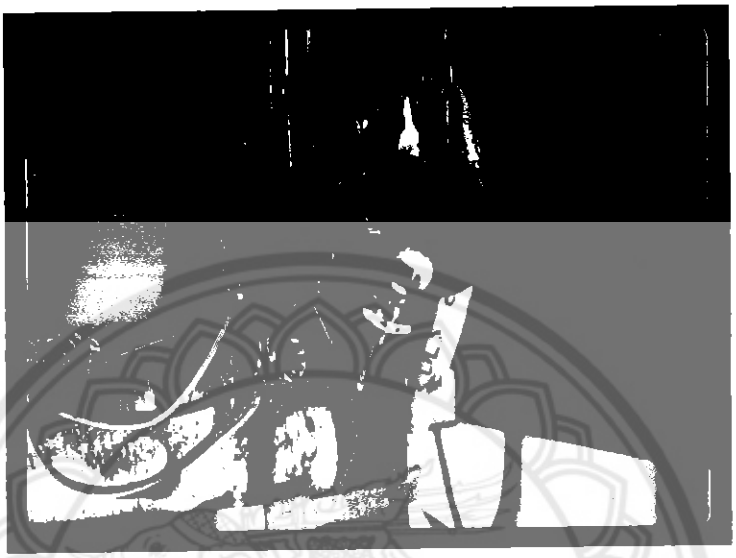
ใช้เป็นตัววัดระยะการแกว่งจากสายตาว่ามีความแตกต่างกันมากน้อยเพียงใด



รูปที่ 15 ไม้บรรทัด

10. ลิมิตสวิตช์ (Limited switch)

เป็นสวิตช์ที่อาศัยแรงกดภายนอกมากระทำ เมื่อมอเตอร์สัมผัสกับสวิตช์ มอเตอร์ก็จะหยุดการทำงาน



รูปที่ 16 Limit switch

11. ระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

รูปที่ 17 ระบบสาธิตการควบคุมการเคลื่อนที่ที่สามารถลดการสั่นของลูกตุ้มที่ติดอยู่บนตัวเคลื่อนที่

ภาคผนวก ข

1.1 โค้ดโปรแกรม Arduino

```
int Right=2;
int EN=3;
int Left=4;
int i=1,k=0;
int SW_Right=6;
int SW_Left=7;
int SW_01=8;
int SW_02=9;
int SW_03=10;
float j=255,jj=0;
float time_speed=0,time_speed2=0;
void setup()
{
  Serial.begin(9600);
    pinMode(Right,OUTPUT);
    pinMode(EN,OUTPUT);
    pinMode(Left,OUTPUT);
    pinMode(SW_Right,INPUT);
    pinMode(SW_Left,INPUT);
 }
```

```
void Stop ()
{
   digitalWrite(Right, HIGH);
   digitalWrite(EN, HIGH);
   digitalWrite(Left, HIGH);
   Delay_Time();
}
void Stop_1 ()
{
   digitalWrite(Right, HIGH);
   digitalWrite(EN, HIGH);
   digitalWrite(Left, HIGH);
}
void Right_Motor ()
{
   digitalWrite(Right, HIGH);
   digitalWrite(EN,HIGH);
   digitalWrite(Left, LOW);
}
void Right_Motor_1 ()
{
   digitalWrite(Right,HIGH);
```

```
  analogWrite(EN,150);
  digitalWrite(Left,LOW );
}
void Right_Motor_2 ()
{
 for(i=0;i<7;i++)
 {
  digitalWrite(Right, HIGH);
  analogWrite(EN,150);
  digitalWrite(Left, LOW);
  delay(215);
  digitalWrite(Right, HIGH);
  analogWrite(EN,50);
  digitalWrite(Left, LOW);
  delay(315);
 }
 while(digitalRead(SW_Right)==HIGH)
 {
  digitalWrite(Right, HIGH);
  analogWrite(EN,50);
  digitalWrite(Left, LOW);
 }
```

```
}
void Right_Motor_3 ()
{
 while(digitalRead(SW_Right)==HIGH)
 {
 jj=j-0.49;
 j=jj;
 digitalWrite(Right, HIGH);
  analogWrite(EN,jj);
 digitalWrite(Left, LOW);
 delay(8);
 }
Stop();
}
void Left_Motor ()
{
  digitalWrite(Right, LOW);
  analogWrite(EN,100);
  digitalWrite(Left, HIGH);
}
void Startt()
{
```

```
  if(digitalRead(SW_Left)==1)
  {
    Left_Motor ();
     while(digitalRead(SW_Left)==1);
    Stop ();
     }
}
void Delay_Time()
{
  for(j=0;j<30;j++)
  {
    delay(1000);
  }
}
void loop()
{
   Startt();
  while(1)
 {
  if(digitalRead(SW_01)==LOW)
     {
        while(digitalRead(SW_01)==LOW);
```

```
      delay(1);
      Right_Motor_1 ();
        while(digitalRead(SW_Right)==HIGH)
        {
         delay(1);
         time_speed=time_speed+1;
        }
       Stop ();
        Left_Motor ();
         delay(1);
       while(digitalRead(SW_Left)==HIGH);
        delay(1);
        Stop_1 ();
         if(!(time_speed==0))
{
  time_speed=(time_speed/1000)+2;
  Serial.println(time_speed);
 time_speed=0;
}
   }
////////////////////////////////////////////////////////
    if(digitalRead(SW_02)==LOW)
```

```
    {
      while(digitalRead(SW_02)==LOW);
      delay(1);
      Right_Motor_2 ();
      Stop ();
        Left_Motor ();
        delay(1);
        while(digitalRead(SW_Left)==HIGH);
       delay(1);
       Stop_1 ();
    }
////////////////////////////////////////////////////////

if(digitalRead(SW_03)==LOW)
    {
      while(digitalRead(SW_03)==LOW);
      delay(1);
      Right_Motor_3 ();
      Stop ();
      Left_Motor ();
        delay(1);
        while(digitalRead(SW_Left)==HIGH);
```

```
        delay(1);
        Stop_I ();
    }
}
  }
```

อธิบายโปรแกรม

- ถ้ามอเตอร์ไม่อยู่ที่จุดเริ่มต้นโปรแกรมจะทำการสั่งให้มอเตอร์เคลื่อนที่มายัง**จุดเริ่มต้นทุกครั้ง**เมื่อเปิดระบบ
- หมุนขวาแบบที่ 1 คือ หมุนด้วยความเร็ว 150 (จากความเร็วสูงสุด 255) จา**กจุดเริ่มต้นไปยัง**จุดหมาย โดยใช้เวลาประมาณ 4 วินาที
- หมุนขวาแบบที่ 2 หมุนด้วยความเร็ว 2 แบบ คือ หมุนด้วยความเร็ว 150 **เป็นระยะเวลา** 215 msec และวิ่งด้วยความเร็ว 50 เป็นระยะเวลา 315 msec จะทำการหมุนแบบนี้**สลับกับเป็น**จำนวน 7 รอบใช้เวลา 3710 msec จากนั้นเวลาที่เหลือ 290 msec จะวิ่งด้วย**ความเร็ว 50 จนถึง**จุดหมาย ใช้เวลาประมาณ 4 วินาที
- หมุนขวาแบบที่ 3 ด้วยความเร็วเกือบเต็มกำลัง คือ 245 จากนั้นค่อยๆลด**ลงทีละ** 0.49 **เป็น**จำนวน 500 รอบ จะไปหยุดนิ่งที่จุดหมายพอดี

2.1 โค้ดโปรแกรม MATLAB การทดลองแบบที่ 1

```
clear; clc;
close all;

% Parameters
w = sqrt(9.81/0.3);  % Natural angular frequency
zeta = 0.05;        % damping coef
xmax = 0.6;         % maximum distance
dt = 0.001;
tmax = 10;          % maximum time

u = [0.15*ones(4000,1); zeros(6000,1)];

A = [0    0   0;
    0    0   1;
    0 -w^2 -2*zeta*w];
B = [1 -1 2*zeta*w]';

% Initial condition
xA = 0;
xBA = 0;
vB = 0;

X = [xA xBA vB]';
t = 0;
tsav = [];
xAsav = [];
xBAsav = [];
vBsav = [];

for i=1:10000
```

```
  t = t+dt;
  Xnew = X + dt*(A*X+B*u(i));

  tsav = [tsav t];
  xAsav = [xAsav Xnew(1)];
  xBAsav = [xBAsav Xnew(2)];
  vBsav = [vBsav Xnew(3)];

  X = Xnew;
end
figure; plot(tsav,u); ylabel('u'); xlabel('t(sec)');

figure;
subplot(3,1,1); plot(tsav,xAsav); ylabel('xA');
subplot(3,1,2); plot(tsav,xBAsav); ylabel('xBA');
subplot(3,1,3); plot(tsav,vBsav); ylabel('vB'); xlabel('t (sec)');

figure; plot(tsav,xBAsav); ylabel('x_{BA} (m)'); xlabel('t (sec)');
```

2.2 โค้ดโปรแกรม MATLAB แการทดลองแบบที่ 2

```
clear; clc;
close all;

% Parameters
w = sqrt(9.81/0.3);  % Natural angular frequency
zeta = 0.05;        % damping coef
xmax = 0.6;         % maximum distance
dt = 0.001;
tmax = 10;          % maximum time
umax = 0.20;
umin = 0.12;

u = zeros(10000,1);
u(1:215) = umax;
u(216:530)= umin;
u(531:745)= umax;
u(746:1060)= umin;
u(1061:1275)= umax;
u(1276:1590)= umin;
u(1591:1805)= umax;
u(1806:2120)= umin;
u(2121:2335)= umax;
u(2336:2650)= umin;
u(2651:2865)= umax;
u(2866:3180)= umin;
u(3181:3395)= umax;
u(3376:3710)= umin;
u(3711:4000)=umin;
```

```
A = [0    0   0;
     0    0   1;
     0 -w^2 -2*zeta*w];
B = [1 -1 2*zeta*w]';

% Initial condition
xA = 0;
xBA = 0;
vB = 0;

X = [xA xBA vB]';
t = 0;
tsav = [];
xAsav = [];
xBAsav = [];
vBsav = [];

for i=1:10000
  t = t+dt;
  Xnew = X + dt*(A*X+B*u(i));

  tsav = [tsav t];
  xAsav = [xAsav Xnew(1)];
  xBAsav = [xBAsav Xnew(2)];
  vBsav = [vBsav Xnew(3)];

  X = Xnew;
end

figure; plot(tsav,u); ylabel('u'); xlabel('t(sec)');

figure;
```

```
subplot(3,1,1); plot(tsav,xAsav); ylabel('xA');
subplot(3,1,2); plot(tsav,xBAsav); ylabel('xBA');
subplot(3,1,3); plot(tsav,vBsav); ylabel('vB'); xlabel('t (sec)');

figure; plot(tsav,xBAsav); ylabel('x_{BA} (m)'); xlabel('t (sec)');
```

2.3 โค้ดโปรแกรม MATLAB การทดลองแบบที่ 3

```
clear; clc;
close all;

% Parameters
w = sqrt(9.81/0.3);  % Natural angular frequency
zeta = 0.05;        % damping coef
xmax = 0.6;         % maximum distance
dt = 0.001;
tmax = 10;          % maximum time
umax = 0.20;
umin = 0.12;

u = zeros(10000,1);

for i=1:500
   u( (8*i+1):(8*i+8) ) = 0.00122*(245 - 0.49*i);
end
A = [0   0   0;
    0   0   1;
    0 -w^2 -2*zeta*w];
B = [1 -1 2*zeta*w]';

% Initial condition
xA = 0;
xBA = 0;
vB = 0;

X = [xA xBA vB]';
t = 0;
```

```
tsav = [];
xAsav = [];
xBAsav = [];
vBsav = [];

for i=1:10000
  t = t+dt;
  Xnew = X + dt*(A*X+B*u(i));

  tsav = [tsav t];
  xAsav = [xAsav Xnew(1)];
  xBAsav = [xBAsav Xnew(2)];
  vBsav = [vBsav Xnew(3)];

  X = Xnew;
end

figure; plot(tsav,u); ylabel('u'); xlabel('t(sec)');

figure;
subplot(3,1,1); plot(tsav,xAsav); ylabel('xA');
subplot(3,1,2); plot(tsav,xBAsav); ylabel('xBA');
subplot(3,1,3); plot(tsav,vBsav); ylabel('vB'); xlabel('t (sec)');

figure; plot(tsav,xBAsav); ylabel('x_{BA} (m)'); xlabel('t (sec)');
```

เอกสารอ้างอิง

[1] สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.ฟิสิกส์1.ถนนพญาไท เขตปทุมวันกรุงเทพฯ

www.ChulaPupress.com

[2] โศรฎา แข็งการและกันต์ธร ชำนิประศาสน์. การใช้ MATLAB สำหรับงานทางวิศวกรรม.

[3] สืบค้นเมื่อ 1 มิถุนายน 2556, จากftp://ftp.ee.psu.ac.th/pub/matlab/help/matlab_guide_thai.pdf

[4] โอภาส ศิริดรรชิตถาวร. Arduino และโมดูล POP-MPU

[5] สุวิทย์ กิระวิทยา วิศวกรรมไฟฟ้าเบื้องต้น (Introduction to Electrical Engineering)