

เครื่องถ่วงสมดุลแผ่นเวียงหมุนแนวตั้ง



นายศิริพงษ์ ปะวะโก

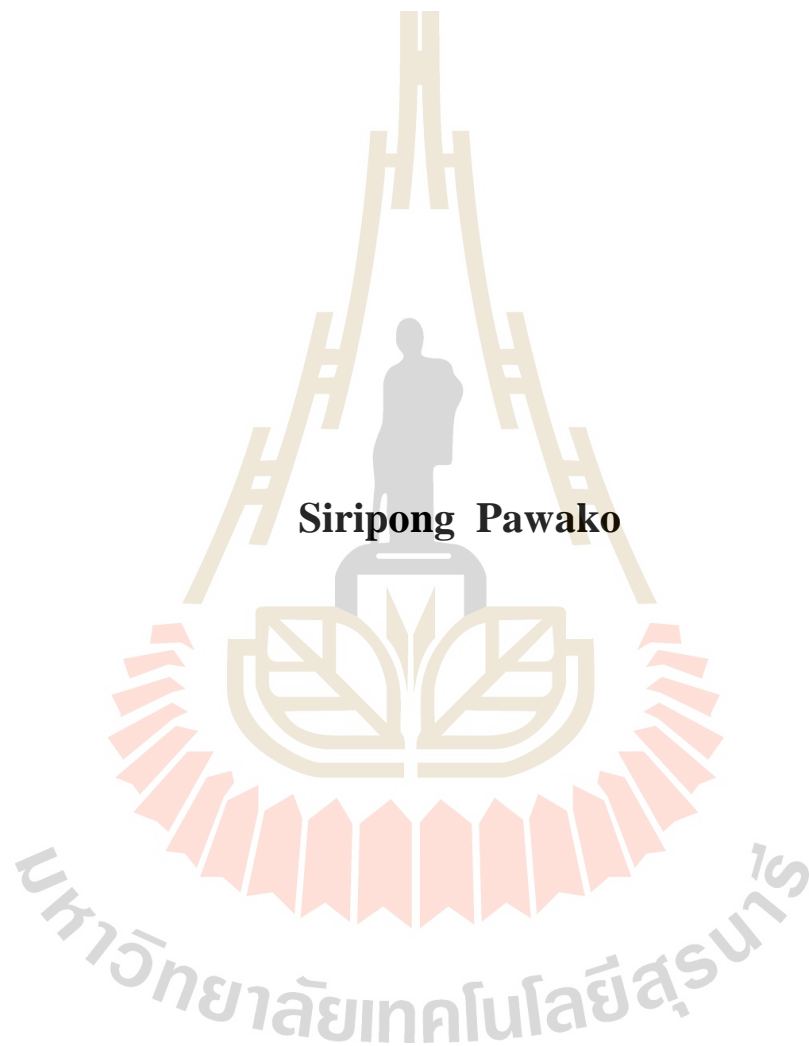
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2560

**VERTICAL BALANCING MACHINE FOR
CENTRIFUGE DISC**



Siripong Pawako

**A Thesis Submitted in Partial Fulfillment of the Requirement for the
Degree of Master of Engineering in Mechatronics Engineering**

Suranaree University of Technology

Academic Year 2017

เครื่องถ่วงสมดุลแผ่นเหยียงหมุนแนวตั้ง

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้หน่วยวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาโทบริหารธุรกิจ

คณะกรรมการสอบวิทยานิพนธ์



(อ. ดร. โสรกฐา แจ็งการ)

ประธานกรรมการ



(รศ. ดร. จิระพล ศรีเสริฐผล)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(อ. ดร. ชิตพงษ์ เวชไรสงค์)

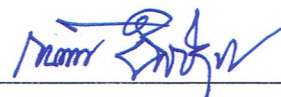
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี



(ศ. ดร. สันติ แม่นศิริ)

รองอธิการบดีฝ่ายวิชาการและพัฒนาคณาจารย์



(รศ. ร.อ. ดร. กนต์ธร ชานีประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

ศิริพงษ์ ปะวะโก : เครื่องถ่วงสมดุลแผ่นเหวี่ยงหมุนแนวตั้ง (VERTICAL BALANCING MACHINE FOR CENTRIFUGE DISC) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร. จิระพล ศรีเสริฐผล, 121 หน้า.

ในปัจจุบันเครื่องหมุนเหวี่ยงได้ถูกนำมาใช้มากมายในวงการทางวิทยาศาสตร์และทางการแพทย์ ซึ่งหน้าที่ของเครื่องนี้คือสร้างแรงหนีศูนย์กลางที่เกิดจากการหมุนรอบจุดหมุนในความเร็วรอบสูง โดยประโยชน์จากการทำงานดังกล่าวก็จะแตกต่างกันออกไปตามชนิดของเครื่อง เช่น ใช้เร่งอัตราการตกตะกอนของอนุภาคสารที่ไม่ละลายในของเหลว แยกของเหลวหลาย ๆ ชนิดที่มีความถ่วงจำเพาะต่างกันหรือช่วยทำสารละลายให้เข้มข้นขึ้น และถูกพัฒนาให้มีความหลากหลายในการใช้งานและมีประสิทธิภาพมากขึ้น ปัญหาที่พบในเครื่องจักรหรืออุปกรณ์ที่มีการหมุนส่งกำลังนั้นส่วนใหญ่จะเป็นการสั่นสะเทือน ที่มีสาเหตุจากความไม่สมดุลในการหมุน สำหรับการสร้างเครื่องปั่นเลือด (Hematocrit Centrifuge Machine) ที่ทำงานด้วยความเร็วรอบสูงถึง 12,000 rpm ทำงานได้อย่างมีประสิทธิภาพ จำเป็นต้องมีการควบคุมการผลิตของแผ่นเหวี่ยงหมุน (Centrifuge Disc) ให้มีค่าความสมดุลไม่เกินค่าที่กำหนด งานวิจัยนี้นำเสนอการออกแบบเครื่องถ่วงสมดุลแผ่นเหวี่ยงหมุนแบบแนวตั้งเพื่อวิเคราะห์และแก้ปัญหาความไม่สมดุลของแผ่นเหวี่ยงหมุนสำหรับเครื่องปั่นเลือด ในช่วงความเร็วรอบ 5,000-6,000 rpm ซึ่งจะสามารถลดการสั่นสะเทือนของเครื่องปั่นเลือดได้



สาขาวิชา วิศวกรรมเครื่องกล
ปีการศึกษา 2560

ลายมือชื่อนักศึกษา ศิริพงษ์ ปะวะโก
ลายมือชื่ออาจารย์ที่ปรึกษา ดร. จิระพล ศรีเสริฐผล

SIRIPONG PAWAKO : VERTICAL BALANCING MACHINE FOR
CENTRIFUGE DISC. THESIS ADVISOR : ASSOC. PROF. JIRAPHON
SRISERTPHOL, Ph.D., 121 PP.

HEMATOCRIT CENTRIFUGE/VIBRATION/BALANCING

At present, the centrifuge machine has been used extensively in the scientific and medical field. The centrifugal force is created by rotation of these machine at high speed rotation. The benefits of centrifugal force will vary depending on the type of machine such as accelerate the rate of settling of particulate matter in insoluble in liquid, separate multiple liquid types with different specific gravity and make solution more concentrated. It was developed for multipurpose and more efficient. The problems encountered in the rotating machinery or equipment are mostly vibrations caused by the imbalance in the rotation. For efficient operation, The Hematocrit Centrifuge Machine which is operated at a speed of up to 12,000 rpm, the controlling of the production of centrifuge disks is required to achieve an unbalance occur no more than acceptable values. This research presents the design of vertical balancing machines to analyze and solve the unbalanced problems of centrifuge disk for Hematocrit Centrifuge Machine. At rotational speed of 5,000-6,000 rpm. The results showed that the vertical balancing machines can help user to analyze and solve the unbalanced problems to reduce the vibration of the Hematocrit Centrifuge Machine.

School of Mechanical Engineering

Academic Year 2017

Student's Signature Pawako S.

Advisor's Signature Sritpol J.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จล่วงตามวัตถุประสงค์ทุกประการ ผู้จัดทำขอขอบพระคุณบุคคลต่างๆ ที่ให้คำปรึกษา แนะนำ และช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการและด้านการดำเนินงาน โครงการงาน ดังนี้

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.จิระพล ศรีเสริฐผล อาจารย์ที่ปรึกษา วิทยานิพนธ์ ที่ให้ความรู้และคำปรึกษาแนะนำในการทำงานด้วยเมตตาตลอด

ขอบคุณ คุณวรรณรักษ์ จวีร์วรรณ และเจ้าหน้าที่ประจำสาขาวิศวกรรมเครื่องกล (หลักสูตร นอกเวลา) ที่อำนวยความสะดวกในด้านเอกสารต่างๆ และคอยให้คำปรึกษามาตลอด

ขอบคุณพี่ๆ เพื่อนๆ น้องๆ บัณฑิตศึกษาที่ใกล้ชิดทุกคนที่มีส่วนทำให้การเรียนในระดับปริญญาโทเป็นช่วงเวลาที่มีความสุข

และที่ขาดไม่ได้ขอกราบขอบพระคุณพ่อสายรุ้ง ปะวะโก และคุณแม่ทองถวิล ปะวะโก และพี่เอกพงษ์ ปะวะโก ที่ให้กำลังใจ ความรัก การอบรมเลี้ยงดู และส่งเสริมการศึกษาเป็นอย่างดี มาโดยตลอดจนทำให้ผู้จัดทำวิทยานิพนธ์ประสบความสำเร็จในชีวิตตลอดมา

สุดท้ายนี้ขออาราธนา สิ่งศักดิ์สิทธิ์ทั้งหลายจงดลบันดาลให้บุคคลทั้งหลายที่ได้กล่าวนาม และไม่ได้กล่าวนาม จงมีแต่ความสุขและความเจริญในชีวิตตลอดไป

ศิริพงษ์ ปะวะโก

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
คำอธิบายสัญลักษณ์และคำย่อ.....	ฎ
บทที่	
1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหาการวิจัย.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 ขอบเขตของการวิจัย.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 สถานที่ดำเนินงานวิจัย.....	2
2 ปรัชญาบรรณกรรมและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 เลือด.....	3
2.1.1 ส่วนประกอบของเลือด.....	3
2.2 เครื่องหมุนเหวี่ยง.....	3
2.2.1 การหมุนเหวี่ยงแบบดิฟเฟอเรนเชียลเกรดเดียนต์.....	4
2.2.2 การหมุนเหวี่ยงแบบเดนซิทีเกรดเดียนต์.....	5
2.2.3 ชนิดของเครื่องหมุนเหวี่ยง.....	5
2.2.4 องค์ประกอบและคุณสมบัติของเครื่องหมุนเหวี่ยง.....	6
2.3 เครื่องปั่นเลือด.....	6
2.4 มอเตอร์ไฟฟ้า.....	7

สารบัญ (ต่อ)

หน้า

2.4.1	มอเตอร์ไฟฟ้ากระแสตรง.....	7
2.4.2	มอเตอร์ไฟฟ้ากระแสสลับ.....	8
2.5	มอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน.....	8
2.6	แบบจำลองทางคณิตศาสตร์ของมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน.....	10
2.7	ระบบควบคุมความเร็วรอบของมอเตอร์กระแสตรงไร้แปรงถ่าน.....	12
2.7.1	ตัวควบคุม PID.....	13
2.7.2	ตัวควบคุมแบบสัดส่วน.....	13
2.7.3	ตัวควบคุมแบบปริพันธ์.....	14
2.7.4	ตัวควบคุมแบบอนุพันธ์.....	14
2.7.5	การทำงานของ PID.....	14
2.8	การสันสะเทือนทางกล.....	15
2.8.1	เครื่องมือวัดการสันสะเทือน.....	15
2.8.2	การจัดการสัญญาณการสันสะเทือน.....	18
2.9	ความไม่สมดุลทางกล.....	19
2.9.1	ขั้นตอนการทำสมดุลแบบกราฟิก.....	23
2.9.2	ขั้นตอนการทำสมดุลแบบคำนวณสมการ.....	25
2.10	ปริทัศน์วรรณกรรมที่เกี่ยวข้อง.....	25
3	วิธีดำเนินการวิจัย.....	28
3.1	เครื่องปั้นงาน.....	28
3.1.1	Arduino Uno R3.....	28
3.1.2	Liquid Crystal Display (LCD).....	30
3.1.3	Brushless Direct Current Motor.....	30
3.1.4	Driver สำหรับ Brushless Direct Current Motor.....	31
3.1.5	แผนภาพและวงจรควบคุมเครื่องปั้นงาน.....	32
3.2	เครื่องถ่วงสมดุล.....	32
3.2.1	Arduino Uno R3.....	33

สารบัญ (ต่อ)

หน้า

3.2.2	Liquid crystal display (LCD).....	34
3.2.3	Accelerometer.....	35
3.2.4	Keypad.....	35
3.2.5	แผนภาพและวงจรควบคุมเครื่องปั่นงาน.....	36
3.3	แนวทางและวิธีดำเนินงานวิจัย.....	37
4	ผลการดำเนินงานวิจัย.....	38
4.1	ออกแบบและสร้างเครื่องปั่นงาน.....	38
4.2	สร้าง Math Model ของระบบมอเตอร์ Brushless DC และตัวควบคุม.....	41
4.3	ออกแบบสร้างเครื่องถ่วงสมดุลเพื่อทดสอบถ่วงสมดุลการหมุน.....	44
4.4	ทดสอบ Closed-Down Speed เพื่อดู Frequency.....	53
4.5	ผลการทดลอง.....	54
4.5.1	เครื่องปั่นงาน.....	54
4.5.2	เครื่องถ่วงสมดุล.....	54
5	บทสรุปและข้อเสนอแนะ.....	59
5.1	สรุปผลการวิจัย.....	59
5.2	ข้อเสนอแนะ.....	59
	รายการอ้างอิง.....	61
	ภาคผนวก.....	
	ภาคผนวก ก. คู่มือการใช้งานเครื่องถ่วงสมดุล.....	62
	ภาคผนวก ข. อุปกรณ์เพิ่มเติมในการทดลอง.....	71
	ภาคผนวก ค. บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่.....	108
	ประวัติผู้เขียน.....	121

สารบัญตาราง

ตารางที่	หน้า
2.1	ลักษณะการสั่นสะเทือนที่เกิดขึ้นตามเครื่องจักร.....17
4.1	พารามิเตอร์ของ BLDC มอเตอร์.....43
4.2	พารามิเตอร์ของระบบควบคุม PID.....44
4.3	เงื่อนไขการ Trial Weight ของแต่ละงาน.....56
4.4	ผลจากการถ่วงสมดุลในแต่ละเงื่อนไขของการ Trial Weight.....57

สารบัญรูป

รูปที่	หน้า
2.1	ค่าโมโนแกรมสำหรับการหาแรงหนีศูนย์กลาง..... 4
2.2	เครื่องปั่นเลือด..... 7
2.3	โครงสร้างภายใน BLDC Motor..... 9
2.4	อุปกรณ์การขับเคลื่อน BLDC Motor..... 9
2.5	วงจรควบคุมการจ่ายไฟให้กับมอเตอร์..... 10
2.6	วงจรภายในมอเตอร์ BLDC..... 10
2.7	การจ่ายสัญญาณให้กับมอเตอร์ในรูปแบบ Six Step..... 11
2.8	ระบบควบคุมแบบปิด..... 11
2.9	Controller แบบ PID..... 13
2.10	มาตรฐาน ISO 10816-1..... 16
2.11	เปรียบเทียบการสั่นสะเทือนต่อหน่วยการวัด..... 17
2.12	ตำแหน่งการวัดการสั่นสะเทือน..... 18
2.13	การจัดการสัญญาณเพื่อนำไปใช้วิเคราะห์ความเสียหาย..... 18
2.14	สัญญาณก่อนเข้ากระบวนการและหลังเข้ากระบวนการ..... 19
2.15	สัญญาณจากความไม่สมดุลของจานหมุนในโดเมนเวลา..... 19
2.16	สัญญาณจากความไม่สมดุลจานหมุนในโดเมนความถี่..... 20
2.17	ความไม่สมดุลของแผ่นจานหมุน..... 21
2.18	ความไม่สมดุลเนื่องจากมีมวลถ่วง..... 22
2.19	แก้ไขโดยการติดตามขนาดเท่ากันฝั่งตรงข้าม..... 22
2.20	แก้ไขโดยการตัดมวลที่ทำให้เกิดความไม่สมดุลออก..... 22
2.21	การวัดการสั่นสะเทือนและเทียบเฟสสองเสาการสั่นสะเทือน..... 23
2.22	การหาความไม่สมดุลในแผ่นเหวี่ยงหมุน..... 24

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.1 Arduino Uno R3.....	28
3.2 วงจรไฟฟ้าของ Arduino Uno R3.....	29
3.3 จอ LCD แสดงผล 16x2.....	30
3.4 Brushless Direct Current Motor.....	30
3.5 Driver สำหรับ Brushless Direct Current Motor.....	31
3.6 แผนภาพของการเชื่อมต่ออุปกรณ์ของเครื่องปั่นงาน.....	32
3.7 แผนภาพวงจรไฟฟ้าที่ออกแบบสำหรับเครื่องปั่นงาน.....	32
3.8 Arduino Uno R3.....	33
3.9 วงจรไฟฟ้าของ Arduino Uno R3.....	33
3.10 จอ LCD แสดงผล 20x4.....	34
3.11 Accelerometer ADXL335.....	35
3.12 Keypad 1x4.....	35
3.13 แผนภาพของการเชื่อมต่ออุปกรณ์ของเครื่องปั่นงาน.....	36
3.14 แผนภาพวงจรไฟฟ้าที่ออกแบบสำหรับเครื่องปั่นงาน.....	36
4.1 แบบเครื่องปั่นงานมุมมองไอโซเมตริก (Isometric).....	38
4.2 แบบเครื่องปั่นงานมุมมองด้านบน (Top View).....	38
4.3 แบบเครื่องปั่นงานมุมมองด้านหน้า (Front View).....	39
4.4 เครื่องปั่นงาน.....	39
4.5 วงจรควบคุมเครื่องปั่นงาน.....	40
4.6 สถานะติดตั้ง (Setup) ของเครื่องปั่นงาน.....	40
4.7 สถานการณ์ทำงาน (Run) ของเครื่องปั่นงาน.....	41
4.8 Model Closed-Loop ที่จะใช้ประมาณค่าพารามิเตอร์.....	42
4.9 การเปรียบเทียบความเร็วรอบจากการประมาณค่ากับความเร็วยุติจริง.....	42
4.10 โมเดลการ Optimization PID Controller.....	43
4.11 แบบเครื่องถ่วงสมดุล.....	44
4.12 เครื่องถ่วงสมดุล.....	45

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.13 ติดตั้ง Accelerometer ที่มอเตอร์ BLDC.....	46
4.14 เครื่องถ่วงสมดุลแสดงผล Setup Speed.....	46
4.15 เครื่องถ่วงสมดุลแสดงผล Run Machine.....	47
4.16 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้.....	47
4.17 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่เก็บค่า.....	48
4.18 เครื่องถ่วงสมดุลแสดงค่า Trial Weight ที่ต้องติดบนงานเลียด.....	48
4.19 ติดมวล 0.5 กรัมที่มุม 0 องศาบนใบงาน.....	49
4.20 เครื่องถ่วงสมดุลแสดงผล Run Machine.....	49
4.21 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้.....	50
4.22 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่เก็บค่า.....	50
4.23 ขนาดมวลแก้ไขและมุมที่ต้องติดมวลแก้ไข.....	51
4.24 ทำการติดตั้งมวลตามที่เครื่องถ่วงสมดุลคำนวณ.....	51
4.25 เครื่องถ่วงสมดุลแสดงผล Run Machine.....	52
4.26 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้.....	52
4.27 ค่าการสั่นสะเทือนที่ลดลงหลังจากการแก้ไขความไม่สมดุลของงานเลียด.....	53
4.28 Frequency Response ของระบบเครื่องปั่นงานในรูปของ Amplitude การสั่นสะเทือน.....	53
4.29 Frequency Response ของระบบเครื่องปั่นงานในรูปของมุม Phase.....	54
4.30 การปรับตั้งบนเครื่องทดสอบมุมเอียง.....	78
4.31 กราฟแสดงความเร็วรอบของมอเตอร์ BLDC ตลอดการทำงาน 5 นาที.....	55
4.32 กราฟสัญญาณการสั่นสะเทือนเปรียบเทียบกับสัญญาณความเร็วรอบ.....	56
4.33 สัญญาณการสั่นสะเทือนที่สมดุลกับไม่สมดุลของงานใบที่ 1.....	57
4.34 สัญญาณการสั่นสะเทือนที่สมดุลกับไม่สมดุลของงานใบที่ 2.....	58
ก.1 ชุดอุปกรณ์ถ่วงสมดุลงานปั่นเหวี่ยงเลียด.....	63
ก.2 ชุดอุปกรณ์ส่วนมอเตอร์.....	63
ก.3 ชุดอุปกรณ์ส่วนควบคุม.....	64
ก.4 ชุดเครื่องถ่วงสมดุล.....	65

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก.5 แผนภาพการเชื่อมต่อของอุปกรณ์ถ่วงสมดุล.....	65
ก.6 เครื่องถ่วงสมดุล.....	66
ก.7 จอแสดงผลและปุ่มกดของเครื่องถ่วงสมดุล.....	66
ก.8 จอแสดงผลให้เดินเครื่อง.....	67
ก.9 Setup ความเร็วรอบที่ชุดควบคุม.....	67
ก.10 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้.....	67
ก.11 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่เก็บค่า.....	68
ก.12 เครื่องถ่วงสมดุลแสดงค่า Trial Weight ที่ต้องติดบนจานเลือด.....	68
ก.13 แสดงความเร็วรอบขณะทำงาน.....	68
ก.14 ตีคมวล 0.5 กรัมที่มุม 0 องศาบนใบจาน.....	69
ก.15 แสดงขนาดมวลแก้ไขและมุมที่ต้องตีคมวลแก้ไข.....	69
ก.16 ทำการติดตั้งมวลตามที่เครื่องถ่วงสมดุลคำนวณ.....	70
ก.17 แสดงถึงค่าการสันสะเทือนที่ลดลงหลังจากการแก้ไขความไม่สมดุลของจานเลือด.....	70

คำอธิบายสัญลักษณ์และคำย่อ

PID	คือ	การควบคุมชนิดป้อนกลับแบบสัดส่วน-ปริพันธ์-อนุพันธ์
K_p	คือ	ค่าอัตราการขยายแบบสัดส่วน
K_i	คือ	ค่าอัตราการขยายแบบปริพันธ์
K_d	คือ	ค่าอัตราการขยายแบบอนุพันธ์
PWM	คือ	สัญญาณที่มีการปรับความกว้างของพัลส์
ESC	คือ	ชุดอิเล็กทรอนิกส์ควบคุมความเร็วรอบ
RPM	คือ	ความเร็วรอบในการหมุน (Rev/min)

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหาการวิจัย

ในปัจจุบันเครื่องหมุนเหวี่ยงได้ถูกนำมาใช้มากมายในวงการทางวิทยาศาสตร์และทางการแพทย์ ซึ่งหน้าที่ของเครื่องนี้คือสร้างแรงหนีศูนย์กลางที่เกิดจากการหมุนรอบจุดหมุนในความเร็วรอบสูง โดยประโยชน์จากการทำงานดังกล่าวก็จะแตกต่างกันออกไปตามชนิดของเครื่อง เช่น ใช้เร่งอัตราการตกตะกอนของอนุภาคสสารที่ไม่ละลายในของเหลว แยกของเหลวหลาย ๆ ชนิดที่มีความถ่วงจำเพาะต่างกันหรือช่วยทำสารละลายให้เข้มข้นขึ้น ซึ่งเทคโนโลยีปัจจุบันนี้ยังสามารถหาน้ำหนักโมเลกุลของสสารได้อีกด้วย และคาดว่าต่อไปการพัฒนาเกี่ยวกับเครื่องเหวี่ยงหมุนนั้น จะถูกพัฒนาให้มีความหลากหลายในการใช้งานและมีประสิทธิภาพมากขึ้น

ปัญหาที่พบในเครื่องหมุนเหวี่ยงส่วนใหญ่จะเป็นการสั่นสะเทือนที่มีสาเหตุมาจากการไม่สมดุลในการหมุนของจานเหวี่ยงหมุน ซึ่งเครื่องปั่นเลือด (Hematocrit Centrifuge Machine) ทำงานด้วยความเร็วรอบสูงถึง 12,000 rpm ใช้ในการแยกองค์ประกอบต่าง ๆ ภายในเลือด การทำงานได้อย่างมีประสิทธิภาพ จำเป็นต้องมีการควบคุมการผลิตของแผ่นเหวี่ยงหมุน (Centrifuge Disc) ให้มีค่าความสมดุลไม่เกินค่าที่กำหนดผู้ใช้งานมักจะพบปัญหาการสั่นสะเทือนเนื่องจากความไม่สมดุลในแผ่นเหวี่ยงหมุนของเครื่องนี้ ทำให้ประสิทธิภาพในการปั่นเหวี่ยงลดลง

งานวิจัยนี้จึงได้นำเสนอเกี่ยวกับการสร้างเครื่องถ่วงสมดุลแผ่นเหวี่ยงหมุนแบบแนวตั้ง เพื่อวิเคราะห์และแก้ปัญหาความไม่สมดุลในแผ่นเหวี่ยงหมุนที่ความเร็วรอบ 5,000-6,000 rpm ซึ่งเหมาะสำหรับการแก้ปัญหาให้กับแผ่นเหวี่ยงหมุนจากเครื่องปั่นเลือด สามารถลดการสั่นสะเทือนของเครื่องปั่นเลือดได้

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 สร้างเครื่องถ่วงสมดุลแผ่นปั่นเหวี่ยงแบบหมุนแนวตั้ง เพื่อลดผลการสั่นสะเทือนเนื่องจากการความไม่สมดุลของแผ่นปั่นเหวี่ยง

1.3 ขอบเขตของการวิจัย

1.3.1 ใช้ Brushless Direct Current Motor ในการหมุนแผ่นปั่นเหวี่ยง

- 1.3.2 ความเร็วรอบในการหมุนอยู่ระหว่าง 5,000-6,000 rpm
- 1.3.3 ใช้ Accelerometer ในการวัดสัญญาณการสั่นสะเทือน
- 1.3.4 ใช้โปรแกรม Arduino ในการควบคุม

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ระบบควบคุมสามารถคุมความเร็วรอบของเครื่องปั่นเลือดได้
- 1.4.2 เครื่องถ่วงสมดุลสามารถวิเคราะห์และแก้ไขปัญหาในแผ่นเหวี่ยงหมุนได้
- 1.4.3 สามารถลดปัญหาการสั่นสะเทือนของเครื่องปั่นเลือดได้

1.5 สถานที่ดำเนินงานวิจัย

มหาวิทยาลัยเทคโนโลยีสุรนารี



บทที่ 2

ปรีทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 เลือด

เลือดหรือโลหิต (Blood) เป็นของเหลวสีแดงที่ไหลเวียนอยู่ภายในเส้นเลือดทั่วร่างกาย โดยอาศัยการสูบฉีดของหัวใจและอวัยวะสำคัญที่ร่างกายมนุษย์ใช้ในการสร้างเม็ดโลหิต คือ ไชกระดูก ในร่างกายของคนเรามีโลหิตมากน้อยตามน้ำหนักของแต่ละคนคิดโดยประมาณ 80 ซีซี ต่อน้ำหนัก 1 กิโลกรัม น้ำที่สำคัญของเลือด คือ ขนส่งก๊าซออกซิเจนจากปอด ไปยังเซลล์ต่าง ๆ ทั่วร่างกาย และขนส่งก๊าซคาร์บอนไดออกไซด์จากเซลล์เนื้อเยื่อมายังปอด เพื่อขับถ่ายออกจากร่างกายต่อไป นอกจากนี้ เลือดยังทำหน้าที่ขนส่งสารต่าง ๆ เช่น กรดอะมิโน ฮอโมน วิตามินไปยังเซลล์ต่าง ๆ ทั่วร่างกาย และนำของเสียต่าง ๆ จากเซลล์ไปขับออกจากร่างกาย เช่น นำยูเรียไปขับออกที่ไต เป็นต้น

2.1.1 ส่วนประกอบของเลือด

เลือด ประกอบด้วยองค์ประกอบหลัก 2 ส่วน ได้แก่

2.1.1.1 ส่วนที่เป็นของเหลว เรียกว่า พลาสมา (Plasma) ซึ่งมีอยู่ประมาณร้อยละ 55 ของปริมาณเลือดทั้งหมด

2.1.1.2 ส่วนที่เป็นของแข็งคือเซลล์เม็ดเลือด (Cellular Components) ซึ่งมีอยู่ประมาณร้อยละ 45 ของปริมาณเลือดทั้งหมด แบ่งออกเป็น 3 ชนิด ได้แก่

1) เม็ดเลือดแดง มีหน้าที่หลักในการนำออกซิเจน และอาหารที่ย่อยแล้ว ส่งไปยังเซลล์ต่าง ๆ ของร่างกายและรับของเสีย รวมทั้งคาร์บอนไดออกไซด์ออกมาจากเซลล์

2) เม็ดเลือดขาว มีหน้าที่ป้องกันร่างกายจากสิ่งแปลกปลอม มีส่วนสำคัญในระบบภูมิคุ้มกันของร่างกาย

3) เกล็ดเลือด มีความจำเป็นต่อการแข็งตัวของเลือด ในเวลาที่มีบาดแผล ส่วนพลาสมา คือ ส่วนที่เป็นของเหลวสีเหลืองทำให้เม็ดเลือดลอยตัวอยู่ได้

2.2 เครื่องหมุนเหวี่ยง

เครื่องหมุนเหวี่ยง (Centrifuge) เป็นเครื่องมือพื้นฐานสำหรับเร่งอัตราการตกตะกอนของอนุภาค (Particle) ที่ไม่ละลายออกจากของเหลวหรือใช้แยกของเหลวหลาย ๆ ชนิดที่มีความถ่วง

จำเพาะ (Specific Gravity) ต่างกันออกจากกันและทำสารละลายให้เข้มข้นขึ้น โดยการหมุนเหวี่ยง จะมีลักษณะการทำงาน 2 แบบ ดังนี้

2.2.1 การหมุนเหวี่ยงแบบดิฟเฟอเรนเชียลเกรดเดียนต์

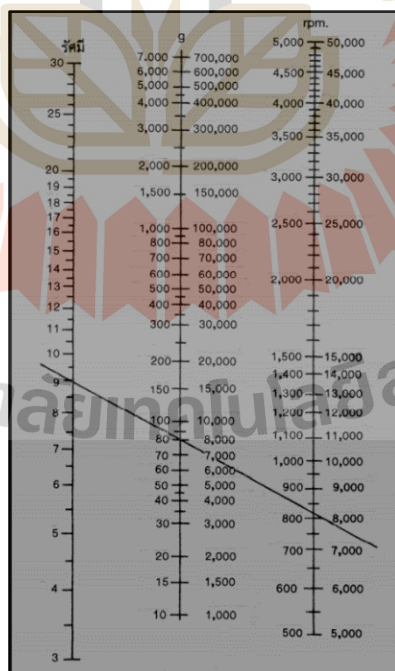
เครื่องหมุนเหวี่ยงสร้างแรงหนีศูนย์กลางหรือแรงหมุนเหวี่ยง (Centrifugal Force, CF) ขึ้น เพื่อเร่งให้อนุภาคตกตะกอนเร็วขึ้น ดังนั้นภายใต้สนามของแรงเหวี่ยงหนีศูนย์กลาง แรงนอนก้นของอนุภาคจะเป็นสัดส่วนโดยตรงกับแรงหนีศูนย์กลาง ทำให้อนุภาคนอนก้นด้วยอัตราเร็วที่ต่างกัน โดยแรงหนีศูนย์กลางนิยมนวัดเปรียบเทียบกับแรงโน้มถ่วงของโลก (Earth's Gravitational Force, g) โดยคำนวณจากจำนวนรอบการหมุนที่สมบูรณ์ของหัวหมุนต่อนาที (Revolution Per Minute, rpm) ดังสมการที่ 2.1

$$RCF = 1.12 \times 10^{-6} \omega^2 r \quad (2.1)$$

โดย RCF คือ แรงหนีศูนย์กลางสัมพัทธ์ (g)

ω คือ ความเร็วรอบที่ใช้ในการปั่น (rpm)

r คือ รัศมีสูงสุดของเลือดที่นำมาปั่น (mm)



รูปที่ 2.1 ค่าโมโนแกรมสำหรับการหาแรงหนีศูนย์กลาง
(ที่มา: ชูชาติ อารีจิตรานุสรณ์, 2544)

จากสมการความสัมพันธ์ดังกล่าวข้างต้น สามารถคำนวณหาแรงเหวี่ยงหนีศูนย์กลางได้จาก โนโมแกรม เมื่อทราบความเร็วรอบและรัศมีของหัวหมุน ดังรูปที่ 2.1

ภายใต้สนามแรงหนีศูนย์กลางอนุภาคจะตกตะกอนด้วยอัตราเร็วที่ไม่เท่ากัน การปั่นแยก ตะกอนจึงต้องใช้เวลานานพอเพียงที่อนุภาคขนาดเล็กจะนอนก้นหมด จนกลายเป็นก้อนตะกอน (Pellet) และของเหลวเหนือตะกอน (Supernatant)

2.2.2 การหมุนเหวี่ยงแบบเดนซิตีเกรเดียนต์

การหมุนเหวี่ยงแบบเดนซิตีเกรเดียนต์ เป็นการแยกอนุภาคของสารออกจาก นอนก้น โดยอาศัยความแตกต่างของอัตราเร็วในการนอนก้นหรือแยกออกจากกัน โดยอาศัย ความแตกต่างของความหนาแน่น โดยใช้ตัวกลางที่เหมาะสมและมีความหนาแน่นต่าง ๆ กัน จึงนิยมใช้สำหรับการแยกสารหลายชนิดออกจากกันโดยมีความบริสุทธิ์สูง สามารถแบ่งย่อยออกได้ เป็น 2 วิธีตามหลักการแยก คือ

- 1) การแยกโดยอาศัยความแตกต่างของอัตราเร็วในการนอนก้น ในกรณีที่อนุภาค หรือสารประกอบที่ต้องการแยกมีอัตราเร็วในการนอนก้นแตกต่างกันมาก
- 2) การแยกโดยอาศัยความแตกต่างของความหนาแน่นของอนุภาค ในกรณีที่ อนุภาคหรือสารประกอบที่ต้องการแยกมีอัตราเร็วในการนอนก้นใกล้เคียงกันมาก

2.2.3 ชนิดของเครื่องหมุนเหวี่ยง

เครื่องหมุนเหวี่ยงมีรูปแบบแตกต่างกันมีทั้งขนาดเล็ก ขนาดปานกลางที่สามารถ ตั้งบน โต๊ะได้ (Bench Model) และขนาดใหญ่ที่สามารถหมุนเหวี่ยงสารละลายได้ครั้งละมาก ๆ ซึ่งต้องตั้งบนพื้นในขณะใช้งาน (Floor Model) แต่ด้วยเทคโนโลยีการผลิตทำให้ขีดความสามารถ ไม่ได้เพิ่มมากขึ้นตามขนาดของเครื่องหมุนเหวี่ยงดังนั้นการแบ่งชนิดที่ค่อนข้างชัดเจนจึงแบ่งตาม แรงหนีศูนย์กลางออกได้เป็น 3 ชนิด ดังนี้

- 1) เครื่องหมุนเหวี่ยงความเร็วรอบต่ำ (Low Speed Centrifuge) เป็นเครื่องหมุนเหวี่ยงขนาดเล็กเป็นส่วนใหญ่ นิยมใช้ในงานทั่ว ๆ ไปในห้องปฏิบัติการมีความเร็วรอบไม่เกิน 6,000 รอบต่อนาที มีแรงหนีศูนย์กลางสูงสุดในช่วง 1,800-7,000 กรัม
- 2) เครื่องหมุนเหวี่ยงความเร็วรอบสูง (High Speed Centrifuge) มีความเร็วรอบไม่เกิน 28,000 รอบต่อนาที มีแรงเหวี่ยงหนีศูนย์กลางสูงสุดถึง 80,000 กรัม จึงนิยมใช้เฉพาะงานที่ต้องการความแรงในการปั่นแยกปานกลาง ตัวอย่างเช่น การแยกอนุภาคขนาดเล็ก ๆ หรือมีน้ำหนั กเบาออกจากของเหลว

3) เครื่องหมุนเหวี่ยงความเร็วรอบสูงมาก (Ultra Speed Centrifuge) เป็นเครื่องหมุนเหวี่ยงที่มีขนาดใหญ่ที่มีความเร็วรอบของการหมุนสูงถึง 150,000 รอบต่อนาที สามารถสร้างแรงหนีศูนย์กลางได้สูงถึง 800,000 กรัม

2.2.4 องค์ประกอบและคุณสมบัติของเครื่องหมุนเหวี่ยง

องค์ประกอบพื้นฐานที่สำคัญของเครื่องหมุนเหวี่ยงมีดังนี้

1) ตัวถัง (Body) สร้างขึ้นเพื่อป้องกันอันตรายแก่ผู้ใช้ และป้องกันอุปกรณ์ภายในตัวถัง

2) หัวหมุน (Rotor หรือ Head) เป็นส่วนที่ทำหน้าที่ยึดหรือใส่กระบอกใส่หลอดปั่น (Bucket, Shield, Carrier) นิยมทำด้วยโลหะที่แข็งแต่มีน้ำหนักเบา เช่น อะลูมิเนียม เหล็กกล้า ไรซินิม ไททานเนียม และโลหะผสม เป็นต้น

3) กระบอกใส่หลอดปั่น ผลิตด้วยเหล็กกล้า ไรซินิมหรือพลาสติกแข็ง กระบอก 1 อันอาจใส่หลอดปั่นได้มากกว่า 1 อันขึ้นอยู่กับจำนวนรูที่เจาะ ที่กั้นกระบอกใส่หลอดปั่นมียางกันแตก (Cushion) ซึ่งมีรูปร่าง 3 แบบตามชนิดของกันหลอดปั่น คือ แบบกรวย แบบกลม และแบบแบน มีอุปกรณ์บางอย่างที่อาจต้องใช้ร่วมกับกระบอกใส่หลอดปั่นคือยางปรับขนาด ซึ่งจะใช้เมื่อหลอดปั่นมีขนาดเล็กกว่ากระบอกใส่หลอดปั่นมาก และแหวนตรึงเนียน (Trunnion Ring) ซึ่งใช้สวมกระบอกใส่หลอดปั่นก่อนนำไปคล้องแขนของหัวหมุนแบบมุมกว้าง

4) มอเตอร์ (Motor) เป็นอุปกรณ์ส่วนที่อยู่ถัดจากหัวหมุนลงมา แกนหมุนของมอเตอร์จะหมุนหัวหมุน โดยตรง ดังนั้นถ้าเกิดความไม่สมดุลในการหมุนของหัวหมุนจึงทำให้ลูกปืนรอบแกนหมุนแตก หรือแกนหมุนของมอเตอร์คดงได้ง่าย ซึ่งสามารถป้องกันและแก้ไขได้โดยการติดตั้งตัวมอเตอร์บนฐานที่ยึดหยุ่นได้ ซึ่งอาจจะเป็นสปริงหรือแท่นยาง ระบบดังกล่าวอาจเรียกชื่อว่า “ระบบป้องกันการไม่สมดุล (Unbalance Protection System)” หรือ “ระบบปรับสมดุลอัตโนมัติ (Autobalance System)” ถึงแม้ว่าระบบดังกล่าวมีขีดจำกัดในการต่อต้านความไม่สมดุลในช่วงไม่เกิน 45 กรัม แต่ช่วยลดเวลาในการซั่งหลอดปั่นพร้อมของเหลวด้วยเครื่องซั่งได้ เพราะสามารถประมาณน้ำหนักที่ใกล้เคียงกันด้วยสายตาแทนการซั่ง แต่ในเครื่องหมุนเหวี่ยงความเร็วสูงและความเร็วสูงมากต้องซั่งน้ำหนักให้เท่ากันทุกครั้งถึงแม้ว่าเครื่องหมุนเหวี่ยงจะมีระบบป้องกันการไม่สมดุล

2.3 เครื่องปั่นเลือด

เครื่องปั่นเลือด (Hematocrit Centrifuge) คือ เครื่องที่ใช้ทำการปั่นเลือด โดยหลักการจะทำการปั่นเลือดที่ความเร็วรอบ 12,000 รอบต่อนาที เป็นเวลา 5 นาที เพื่อสร้างแรงหนีศูนย์กลางในการที่จะแยกองค์ประกอบของเลือดเพื่อดูผลของเลือดที่ได้มาจากการบริจจาค์เลือดที่ได้มามีส่วน

ชนิดใด โดยเลือดที่ได้รับบริจาคนั้นจะเป็นเลือดรวม ซึ่งจะมีส่วนประกอบภายในเลือดต่าง ๆ จะแบ่งหลัก ๆ ออกเป็น 3 ชนิด ได้แก่ เม็ดเลือดแดง พลาสมา และเกล็ดเลือด ซึ่งเลือดแต่ละชนิดที่แยกส่วนประกอบออกจะถูกนำไปใช้สำหรับผู้ป่วยที่มีความต้องการต่างกัน คือ เลือดรวมจะใช้สำหรับผู้ป่วยที่มีการเสียเลือดมากจากอุบัติเหตุ ผ่าตัดใหญ่ เม็ดเลือดแดงใช้สำหรับผู้ป่วยโรคโลหิตจาง โรคมะเร็งเม็ดเลือดขาว โรคธาลัสซีเมีย เกล็ดเลือด ใช้สำหรับผู้ป่วยโรคไข้เลือดออกหรือผู้ป่วยที่มีเกล็ดเลือดต่ำกว่าปกติ และพลาสมา ใช้สำหรับผู้ป่วยโรคฮีโมฟีเลีย โรคตับแข็ง



รูปที่ 2.2 เครื่องปั่นเลือด

2.4 มอเตอร์ไฟฟ้า

มอเตอร์ (Motor) เป็นเครื่องใช้ไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล ประกอบด้วยขดลวดที่พันรอบแกนโลหะที่วางอยู่ระหว่างขั้วแม่เหล็ก โดยเมื่อผ่านกระแสไฟฟ้าเข้าไปยังขดลวดที่อยู่ระหว่างขั้วแม่เหล็ก จะทำให้ขดลวดหมุนไปรอบแกน และเมื่อสลับขั้วไฟฟ้า การหมุนของขดลวดจะหมุนกลับทิศทางเดิม ซึ่งสามารถแบ่งได้เป็น 2 ประเภท คือ

2.4.1 มอเตอร์ไฟฟ้ากระแสตรง

เมื่อมีการผ่านกระแสไฟฟ้าเข้าไปยังขดลวดในสนามแม่เหล็กจะทำให้เกิดแรงแม่เหล็กซึ่งมีส่วนของแรงขึ้นกับกระแสแรงของสนามแม่เหล็ก โดยแรงจะเกิดขึ้นเป็นมุมฉากกับกระแสและสนามแม่เหล็กขณะที่ทิศทางของแรงกลับตรงกันข้ามกัน ถ้าหากกระแสของ

สนามแม่เหล็กไหลย้อนกลับจะทำให้เกิดการเปลี่ยนแปลงของกระแส และสนามแม่เหล็กเป็นผลทำให้ทิศทางของแรงเปลี่ยนไป ด้วยคุณสมบัตินี้ทำให้มอเตอร์กระแสตรงกลับทิศทางการทำงานได้

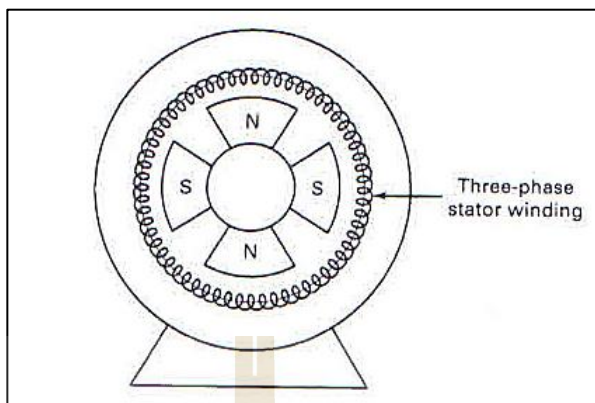
สนามแม่เหล็กของมอเตอร์ส่วนหนึ่งเกิดขึ้นจากแม่เหล็กถาวร ซึ่งจะถูกยึดติดกับแผ่นเหล็กหรือเหล็กกล้า โดยปกตินี้จะเป็นส่วนที่ยึดอยู่กับที่และขดลวดเหนี่ยวนำจะพันอยู่กับส่วนที่เป็นแกนหมุนของมอเตอร์

2.4.2 มอเตอร์ไฟฟ้ากระแสสลับ

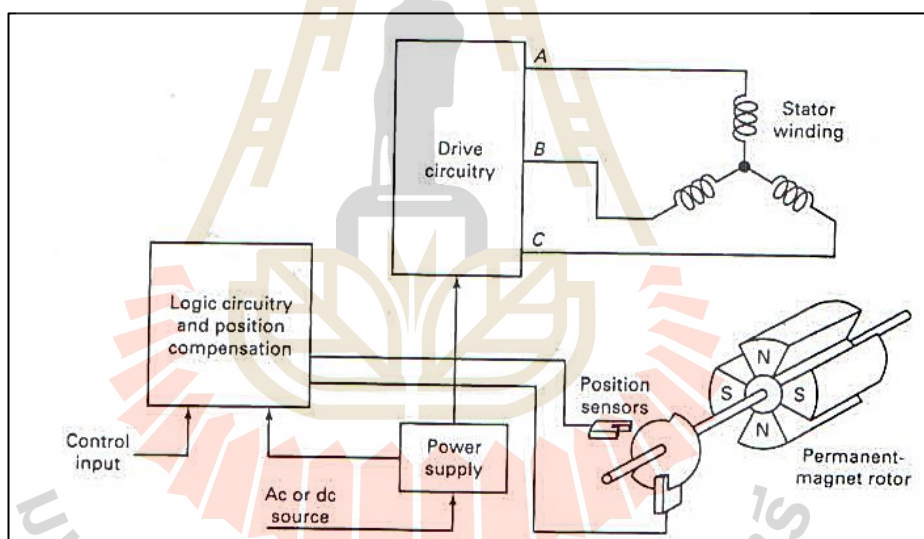
มอเตอร์ไฟฟ้ากระแสสลับ หมายถึง มอเตอร์ที่ใช้กับระบบไฟฟ้ากระแสสลับ เป็นเครื่องกลไฟฟ้าที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล ส่วนที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้า คือ ขดลวดในสเตเตอร์ และส่วนที่ทำหน้าที่ให้พลังงานกล คือ ตัวหมุนหรือโรเตอร์ ซึ่งเมื่อขดลวดในสเตเตอร์ได้รับพลังงานไฟฟ้าก็จะสร้างสนามแม่เหล็กขึ้นมาในตัวที่อยู่กับที่หรือสเตเตอร์ ซึ่งสนามแม่เหล็กที่เกิดขึ้นนี้จะมีการเคลื่อนที่หรือหมุนไปรอบ ๆ สเตเตอร์ เนื่องจากการต่างเฟสของกระแสไฟฟ้าในขดลวดและการเปลี่ยนแปลงของกระแสไฟฟ้า ในขณะที่สนามแม่เหล็กเคลื่อนที่ไปสนามแม่เหล็กจากขั้วเหนือก็จะพุ่งเข้าหาขั้วใต้ ซึ่งจะไปตัดกับตัวนำที่เป็นวงจรรปิดหรือขดลวดกรงกระรอกของตัวหมุนหรือโรเตอร์ ทำให้เกิดการเหนี่ยวนำของกระแสไฟฟ้าขึ้นในขดลวดของโรเตอร์ ซึ่งสนามแม่เหล็กของโรเตอร์นี้จะเคลื่อนที่ตามทิศทางเคลื่อนที่ของสนามแม่เหล็กที่สเตเตอร์ก็จะทำให้โรเตอร์ของมอเตอร์เกิดพลังงานกลสามารถนำไปขับภาระโหลดที่ต้องการหมุนได้

2.5 มอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน

มอเตอร์กระแสตรงแบบไร้แปรงถ่าน (Brushless Direct Current Motor) ชนิดนี้จะเป็นมอเตอร์กระแสตรงที่ไม่ใช้แปรงถ่านในการถ่ายพลังงาน ด้านหยุดเป็นขดลวดอาร์เมเจอร์ ส่วนด้านขับเคลื่อนที่เป็นขั้วแม่เหล็กถาวร การทำงานของมอเตอร์ชนิดนี้จะให้ขดลวดอาร์เมเจอร์ต่อเชื่อมเข้ากับวงจรสวิตซ์ซึ่งอิเล็กทรอนิกส์ เป็นผลทำให้ทิศทางไหลของกระแสในขดลวดอาร์เมเจอร์เกิดการเปลี่ยนแปลงตามความถี่ของการสวิตซ์ของทรานซิสเตอร์กำลัง ทำให้โรเตอร์ที่เป็นแม่เหล็กถาวรหมุนตามการเปลี่ยนแปลงของสนามแม่เหล็กที่ขดลวดอาร์เมเจอร์ ส่วนตำแหน่งในการตรวจจับที่เพล่าจะใช้ในคอนเริ่มตัน เพื่อให้ได้เวลาในการสวิตซ์ที่มีความเหมาะสม ซึ่งวงจรอิเล็กทรอนิกส์ที่ใช้กับมอเตอร์ กระแสตรงแบบไร้แปรงถ่านเป็นส่วนที่มีความยุ่งยากซับซ้อนพอสมควร อย่างไรก็ตาม หลักการทำงานโดยทั่วไปของมอเตอร์กระแสตรงแบบไร้แปรงถ่านซึ่งแสดงให้เห็นได้ดังรูปที่ 2.3



รูปที่ 2.3 โครงสร้างภายใน BLDC Motor
(ที่มา: G Prasad และคณะ, 2012)

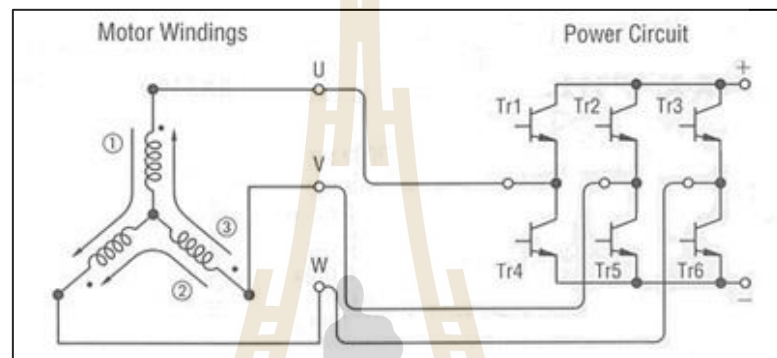


รูปที่ 2.4 อุปกรณ์การขับเคลื่อน BLDC Motor
(ที่มา: G Prasad และคณะ, 2012)

จากรูปที่ 2.4 เป็นระบบการควบคุมมอเตอร์กระแสตรงแบบไร้แปรงถ่านชนิด 3 เฟส ซึ่งประกอบด้วยสเตเตอร์ที่มีขดลวดอาร์เมเจอร์จำนวน 3 ชุด โดยได้รับการกระตุ้นจากวงจรขับเคลื่อนอิเล็กทรอนิกส์และมีสัญญาณตรวจจับตำแหน่งอยู่ที่เพลลา ซึ่งโครงสร้างของมอเตอร์กระแสตรงแบบไร้แปรงถ่านมีลักษณะคล้ายกันกับของมอเตอร์ซิงโครนัส จำนวนขั้วแม่เหล็กของขดลวดอาร์เมเจอร์ที่สเตเตอร์จะถูกกำหนดตามจำนวนขั้วของแม่เหล็กถาวรของโรเตอร์

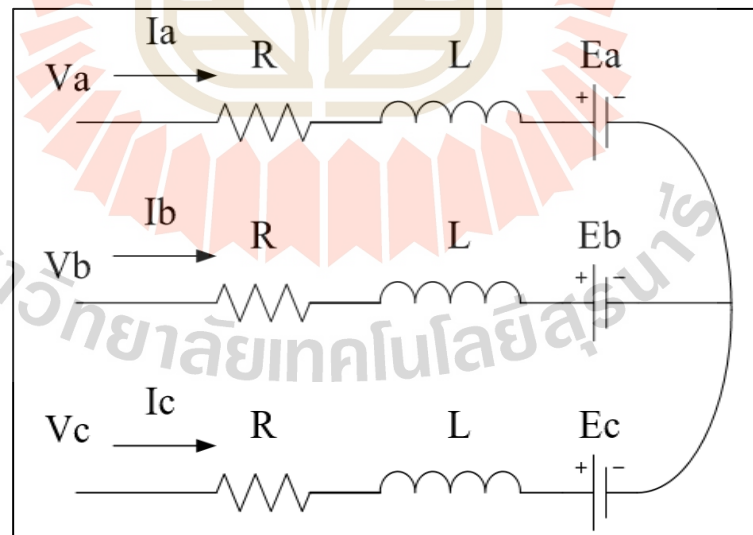
2.6 แบบจำลองทางคณิตศาสตร์ของมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน

มอเตอร์กระแสตรงไร้แปรงถ่าน (Brushless Direct Current Motor) เป็นมอเตอร์กระแสตรงที่มีการวางขดลวด 3 ชุดซึ่งในแต่ละชุดวางห่างกัน 120 องศาต่อกันแบบ Y Circuit โดยจะมีการป้อนสัญญาณจากไฟฟ้าสลับไปมา เพื่อเหนี่ยวนำให้เกิดสนามแม่เหล็กทำให้มอเตอร์เกิดการหมุนขึ้น



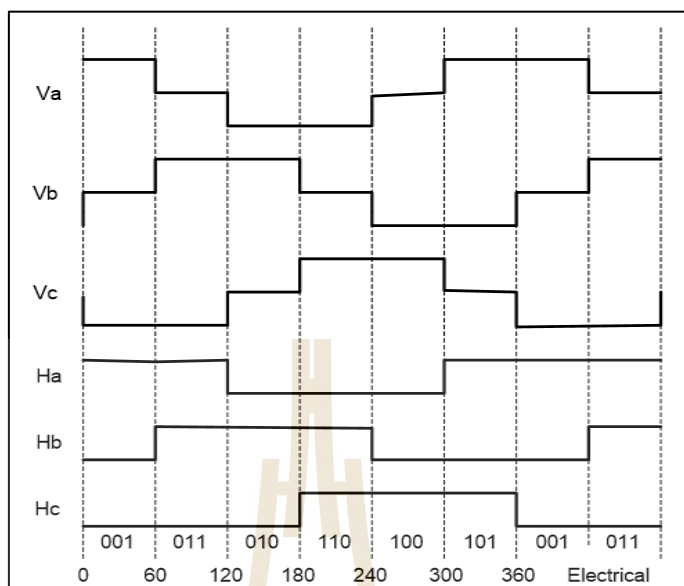
รูปที่ 2.5 วงจรควบคุมการจ่ายไฟให้กับมอเตอร์

(ที่มา: Pawin Jawayon และ Jiraphon Srisertpol, 2012)



รูปที่ 2.6 วงจรภายในมอเตอร์ BLDC

(ที่มา: Pawin Jawayon และ Jiraphon Srisertpol, 2012)



รูปที่ 2.7 การจ่ายสัญญาณให้กับมอเตอร์ในรูปแบบ Six Step
(ที่มา: Pawin Jawayon และ Jiraphon Srisertpol, 2012)

Plant ของระบบนี้ก็คือ มอเตอร์ Brushless DC ซึ่งการหา Mathematic Model ของ BLDC นั้นจะสามารถหาได้จากสมการ Model ทางไฟฟ้า โดยการจ่ายไฟฟ้าให้กับมอเตอร์จะได้ตามสมการ

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = R \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} + \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} \frac{di}{dt} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} E_a \\ E_b \\ E_c \end{bmatrix} \quad (2.2)$$

เมื่อมอเตอร์เกิดการเหนี่ยวนำสนามแม่เหล็กจะทำให้เกิดแรงบิดของมอเตอร์ตามสมการ

$$T_e = \frac{E_a I_a - E_b I_b - E_c I_c}{\omega} \quad (2.3)$$

Model ทางกล เมื่อเกิดการบิดหมุนของมอเตอร์จะเป็นไปตามสมการ

$$J \frac{d\omega}{dt} + B\omega = T_e - T_i \quad (2.4)$$

โดยที่ V_a, V_b, V_c คือ เฟสของความต่างศักย์ที่จ่ายให้กับมอเตอร์ (V)
 I_a, I_b, I_c คือ เฟสของกระแสที่จ่ายให้กับมอเตอร์ (A)

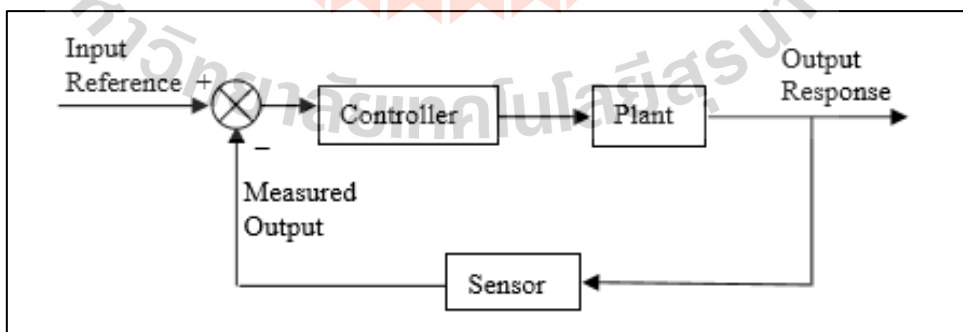
E_a, E_b, E_c	คือ เฟสของศักย์ไฟฟ้าต้านกลับ (V)
R	คือ ความต้านทานของแต่ละเฟส (Ω)
L	คือ ค่าเหนี่ยวนำไฟฟ้าภายในเฟส (H)
H	คือ ค่าเหนี่ยวนำไฟฟ้ารวม (H)
T_e	คือ แรงบิดที่เกิดขึ้นจากสนามแม่เหล็กไฟฟ้า (N.m)
T_i	คือ แรงบิดภายนอก (N.m)
ω	คือ ความเร็วรอบในการหมุน (rad/s)
B	คือ สัมประสิทธิ์การหน่วงในการหมุน (N.s/m)
J	คือ โมเมนต์ความเฉื่อยในการหมุน ($kg.m^2$)

จาก Model ข้างต้นสามารถหาแบบจำลองทางคณิตศาสตร์อย่างง่ายจากการ Optimization ในโปรแกรมคำนวณต่างๆ ซึ่งจะสามารถใช้ Model ดังนี้

$$\frac{\omega(s)}{v(s)} = \frac{K_t}{JLs^2 + (JR + BL)s + (BR + K_t K_b)} \quad (2.5)$$

2.7 ระบบควบคุมความเร็วรอบของมอเตอร์กระแสตรงไร้แปรงถ่าน

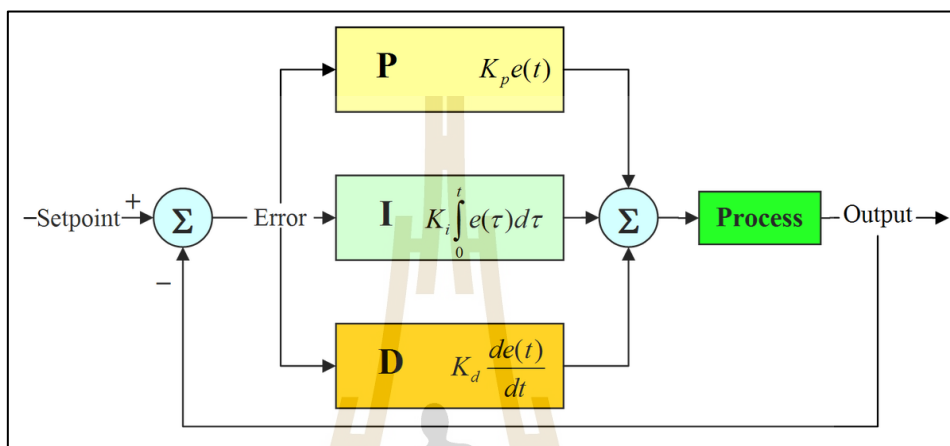
การควบคุมความเร็วรอบการหมุนจะอาศัยหลักการทางด้านระบบควบคุมอัตโนมัติ โดยจะใช้เป็นระบบควบคุมแบบปิด (Closed Loop Control) ในการทำงาน เหตุผลคือระบบควบคุมแบบปิดนั้นจะมีการวัดสัญญาณ Output ที่ได้ เพื่อที่จะนำมาเปรียบเทียบกับสัญญาณ Input และตัวควบคุม จะทำการชดเชยสัญญาณเพื่อที่จะให้ Output นั้นเข้าใกล้ Input มากที่สุด



รูปที่ 2.8 ระบบควบคุมแบบปิด

(ที่มา: Manjita Srivastava และ คณะ, 2009)

การออกแบบ Controller นั้นเป็นปัจจัยที่สำคัญมากในการควบคุมความเร็วรอบของมอเตอร์ ซึ่งผลตอบสนองที่ได้จาก Controller แต่ละแบบจะไม่เหมือนกัน ซึ่ง Controller จะมี Diagram ดังรูปที่ 2.9



รูปที่ 2.9 Controller แบบ PID

(ที่มา: Manjita Srivastava และคณะ, 2009)

2.7.1 ตัวควบคุม PID

ตัวควบคุมแบบ PID เป็นตัวควบคุมที่พบบ่อยมากที่สุดในงานอุตสาหกรรมทั่วไป เพราะเป็นตัวควบคุมที่ใช้งานง่าย การปรับค่าเกนอาศัยหลักการที่ไม่ได้ซับซ้อนมากนักก็ให้ผลตอบสนองเป็นที่ยอมรับได้สามารถปรับแต่งการควบคุมได้ง่ายเมื่อต้องการ ระบบควบคุมแบบพีไอดีมีตัวควบคุมย่อย 3 ตัว คือ

- ตัวควบคุมแบบสัดส่วนหรือตัวควบคุมพี (P Controller)
- ตัวควบคุมแบบปริพันธ์หรือตัวควบคุมไอ (I Controller)
- ตัวควบคุมแบบอนุพันธ์หรือตัวควบคุมดี (D Controller)

ในการควบคุมระบบทั่วไปมักใช้งานตัว ควบคุมร่วมกัน เช่น การควบคุมแบบพีไอ การควบคุมแบบพีดี และการควบคุมแบบพีไอดี รายละเอียดการทำงานที่ต่างกัน

2.7.2 ตัวควบคุมแบบสัดส่วน

ตัวควบคุมแบบนี้จะนำ เอาสัญญาณค่าความผิดพลาดระหว่างสัญญาณอ้างอิงกับสัญญาณเอาต์พุตมาเป็นอินพุตของตัวควบคุมแล้วตัวควบคุมจะทำการสร้างสัญญาณเอาต์พุตด้วยการขยายสัญญาณความผิดพลาดดังกล่าวด้วยค่าเกนของตัวควบคุม จุดเด่นของตัวควบคุมแบบนี้เมื่อนำไปใช้งานก็คือ การปรับค่าเกนให้สูงขึ้นจะมีผลทำให้ระบบมีผลตอบสนองที่เร็วขึ้น ปัญหาที่

อาจจะเกิดขึ้นในการนำไปใช้งานก็คือ ถ้านำไปใช้กับระบบ ชนิด 0 (System Type 0) ตัวควบคุมแบบนี้จะไม่สามารถขจัดค่าความผิดพลาดในสถานะคงตัวได้แต่ก็สามารถทำให้ความผิดพลาดดังกล่าวมีค่าน้อยลงได้ด้วยการปรับค่าเกนให้สูง ซึ่งในทางปฏิบัติแล้วการปรับค่าเกนให้สูงมากขนาดไหนเอาต์พุตที่ออกจริง ๆ จากตัวควบคุมมักมีค่าจำกัด และการปรับเกนให้มีค่าสูงสำหรับระบบที่มีอันดับสูง อาจจะทำให้ได้ผลตอบสนองที่ไม่เป็นที่พึงประสงค์ เช่น การปรับเกนให้สูงขึ้นสำหรับระบบอันดับสอง ผลที่ตามมาก็คือ ค่าพุงเกินก็จะสูงขึ้นตามด้วย ซึ่งอาจจะเป็นอันตรายต่อระบบได้

2.7.3 ตัวควบคุมแบบปริพันธ์

ตัวควบคุมแบบนี้จะนำเอาสัญญาณความผิดพลาดระหว่างสัญญาณอ้างอิงกับสัญญาณเอาต์พุตมาเป็นอินพุตของตัวควบคุม แล้วตัวควบคุมจะทำการสร้างสัญญาณเอาต์พุตด้วยการอินทิเกรตสัญญาณความผิดพลาดดังกล่าวแล้วคูณด้วยค่าเกนของตัวควบคุม จุดเด่นของตัวควบคุมแบบนี้เมื่อนำไปใช้งานก็คือ ถ้านำไปใช้กับระบบชนิด 0 (System type 0) ตัวควบคุมแบบนี้จะสามารถขจัดค่าความผิดพลาดในสถานะคงตัวได้ ข้อดีที่อาจจะเกิดขึ้นในการนำไปใช้งานก็คือ ตัวควบคุมแบบนี้ไม่สามารถลดผลของการพุงเกินของผลตอบสนองได้และการปรับเกนให้มีค่าสูงจะทำให้ได้ผลตอบสนองที่ไม่เป็นที่พึงประสงค์เช่นการปรับเกนให้สูงขึ้น อาจจะมีผลทำให้ผลตอบสนองของระบบเกิดการแกว่งตัวได้

2.7.4 ตัวควบคุมแบบอนุพันธ์

ตัวควบคุมแบบนี้จะนำเอาสัญญาณความผิดพลาดระหว่างสัญญาณอ้างอิงกับสัญญาณเอาต์พุตมาเป็นอินพุตของตัวควบคุม แล้วตัวควบคุมจะทำการสร้าง สัญญาณเอาต์พุตด้วยการอนุพันธ์สัญญาณความผิดพลาดดังกล่าว แล้วคูณด้วยค่าเกนของตัวควบคุม จุดเด่นของตัวควบคุมแบบนี้เมื่อนำไปใช้งานก็คือ ตัวควบคุมแบบนี้ใช้สำหรับลดผลของการพุงเกินของผลตอบสนองได้ ลดผลตอบสนองที่มีการเปลี่ยนแปลงไปมาได้แต่ต้องปรับค่าเกนให้เหมาะสมด้วยไม่เช่นนั้นอาจจะทำให้ได้ผลตอบสนองที่ไม่เป็นที่พึงประสงค์ ปัญหาที่อาจจะเกิดขึ้นในการนำเอาตัวควบคุมนี้ไปใช้งานก็คือ ตัวควบคุมแบบนี้จะไม่สามารถนำไปใช้กับระบบที่มีสัญญาณรบกวนได้ เพราะตัวควบคุมนี้จะขยายสัญญาณรบกวนทำให้ระบบทำงานเพี้ยนได้

2.7.5 การทำงานของ PID

เมื่อทำการหมุนเครื่องปั่นด้วยความเร็วที่กำหนดไว้ ระบบจะเริ่มหมุนจากหยุดนิ่งไปจนถึงค่าความเร็วที่กำหนดไว้โดยมี Sensor วัดความเร็วรอบที่ได้จะส่งค่ากลับไปยัง Controller เพื่อเปรียบเทียบความเร็วที่ได้กับความเร็วที่กำหนดจะได้ค่าความคลาดเคลื่อนดังสมการ

$$Error = Setpoint - Output \quad (2.6)$$

ซึ่งค่าความคลาดเคลื่อนที่ได้จะถูกคำนวณใหม่โดย PID Controller ก่อนที่จะส่งสัญญาณสั่งงาน Input ครึ่งถัดไปดังสมการ

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (2.7)$$

โดยที่ e คือ Error ที่เกิดขึ้น
 u คือ Input ที่จ่ายให้กับระบบ (U)
 t คือ เวลาที่วัดและคำนวณ ณ ขณะนั้น (s)

2.8 การสั่นสะเทือนทางกล

การสั่นสะเทือน (Vibration) คือ การแกว่งหรือการสั่นของวัตถุรอบ ๆ จุดสมดุล ยกตัวอย่างเช่น การแกว่งของลูกตุ้ม การสั่นสะเทือนของปั้มน้ำ หรือแม้กระทั่งการสั่นสะเทือนของลำโพงมอเตอร์ เครื่องซักผ้า เป็นต้น เห็นได้ว่าการสั่นสะเทือนมีทั้งคุณประโยชน์และข้อเสีย โดยส่วนใหญ่เราตรวจสอบการสั่นสะเทือนของเครื่องมือและอุปกรณ์ เพื่อนำไปวิเคราะห์และป้องกันความเสียหายของเครื่องจักรกล ซึ่งการสั่นสะเทือนแบ่งเป็น 2 ประเภทได้แก่

- การสั่นสะเทือนแบบอิสระ (Free Vibration) คือ ปรากฏการณ์ทางกลที่วัตถุหนึ่งได้รับพลังงาน (พลังงานจล, พลังงานศักย์) จากนั้นจึงเกิดการสั่นสะเทือนอย่างอิสระ เช่น การดึงเก้าอี้ชิงช้าแล้วปล่อยให้แกว่งอย่างอิสระ การตีระฆังให้มีเสียงดังและสั่น
- การสั่นสะเทือนแบบบังคับ (Forced Vibration) การสั่นที่มีแรงกระทำจากภายนอกอย่างต่อเนื่องตลอดเวลา ความถี่ของการสั่นจะเท่ากับความถี่ของแรงกระทำในระบบ แรงกระทำนี้เป็นได้ทั้งแรงเสียดทานหรือหน่วง (Damp) และแรงเสริม (Force) เช่น ผู้เฒ่าอาจจะค่อย ๆ ออกแรงสวนทิศทางการแกว่งซึ่งก็จะทำให้ชิงช้าค่อย ๆ ช้าลง เป็นต้น แต่หากผู้เฒ่าจะออกแรงเสริมในจังหวะที่พอดีกับการแกว่งของชิงช้าก็จะทำให้ชิงช้าแกว่งแรงขึ้นและสูงขึ้นเรื่อย ๆ (คาบการให้แรงพอดีกับคาบการสั่นแบบอิสระ) ลักษณะเช่นนี้ก็คือ “เรโซแนนซ์”

2.8.1 เครื่องมือวัดการสั่นสะเทือน

ปัจจุบันเครื่องมือวัดการสั่นสะเทือนแบบพกพา (Vibration Meter) เป็นเครื่องมือที่ถูกนำมาใช้ในการตรวจสอบความผิดปกติของเครื่องจักร โดยอาศัยขนาดการสั่นสะเทือน (Vibration Amplitude) กับมาตรฐานการสั่นสะเทือนแล้วนำมาประเมินว่าเครื่องจักรนั้นทำงานปกติหรือผิดปกติ เพื่อจะได้วางแผนงานบำรุงรักษาต่อไป โดยเครื่องมือวัดการสั่นสะเทือนแบบพกพา (Vibration Meter) หรือที่หลาย ๆ คนรู้จักกันในชื่อ Vibration Meter เป็นเครื่องมือที่ใช้งานง่าย

และสะดวก จึงเหมาะกับผู้เริ่มต้นตรวจสอบสภาพเครื่องจักร แต่ถึงกระนั้นผู้ที่ใช้เครื่องมือวัดชนิดนี้ควรเตรียมตัวเบื้องต้น เพื่อจะได้ใช้เครื่องมือได้อย่างมีประสิทธิภาพ ดังนี้

2.8.1.1 มาตรฐาน ISO 10816-1

ควรมีความรู้เกี่ยวกับมาตรฐานการวัดการสั่นสะเทือน อาทิเช่น ISO 2373:1974, ISO 10816-1 หรือ ISO 10816-3 ในที่นี้ขอแนะนำมาตรฐาน ISO 10816-1 เป็นมาตรฐานในการบอกระดับความรุนแรงด้านการสั่นสะเทือน โดยมีเงื่อนไขการใช้งานเบื้องต้น คือ

- ย่านความถี่ 10-1,000 Hz
- แอมพลิจูด หน่วย mm/sec, rms (ความเร็ว)
- เครื่องจักรต้องมีความเร็วของเพลาน้อยกว่า 120 รอบต่อนาที

โดยค่าระดับของ ISO 10816-1 ดังรูปที่ 2.10

R.M.S Vibration Velocity RMS	Class I Up to 15 kW	Class II	Class III	Class IV
0.28	A	A	A	A
0.45				
0.71	B	B	B	
1.12				
1.8	C	C	C	
2.8				
4.5	D	D	D	B
7.1				
11.2	D	D	D	C
18				
28	D	D	D	D
45				

รูปที่ 2.10 มาตรฐาน ISO 10816-1

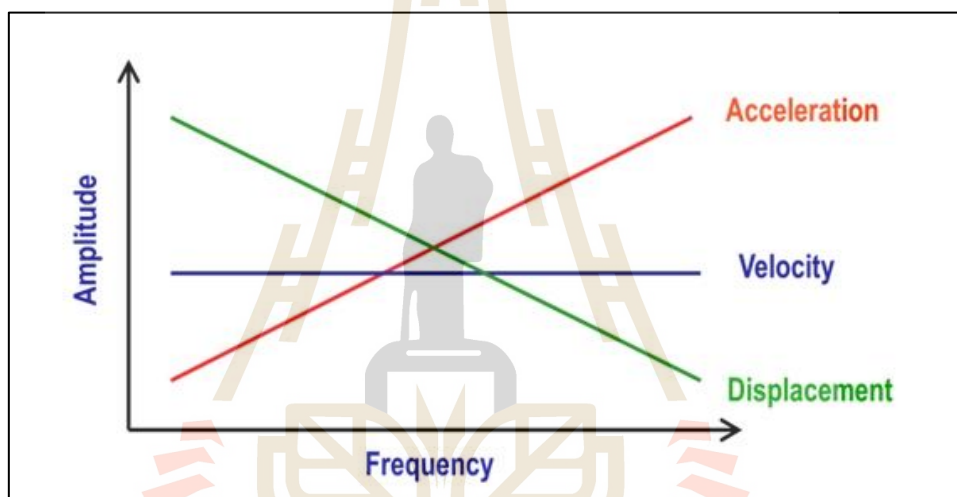
(ที่มา: อัยกฤษ รอดพ่าย, 2559)

มาตรฐาน ISO 10816-1 จะจำแนกเครื่องจักรตามกำลัง (Power, kW) ของเครื่องจักร โดยแบ่งเป็น 4 Class

- Class I: สำหรับเครื่องจักรที่มีกำลังไม่เกิน 15 kW
- Class II: สำหรับเครื่องจักรขนาดกลางที่มีกำลังตั้งแต่ 15 kW ถึง 75 kW
- Class III: สำหรับเครื่องจักรขนาดใหญ่ที่มีฐานแบบ Rigid มีกำลังไม่เกิน 300 kW
- Class III: สำหรับเครื่องจักรขนาดใหญ่ที่มีฐานเป็นแบบ Soft อาทิเช่น Gas Turbine, Turbo Generator ที่มีกำลังไม่เกิน 10 MW

ตารางที่ 2.1 ลักษณะการสั่นสะเทือนที่เกิดขึ้นตามเครื่องจักร

A	เครื่องจักรใหม่เพิ่งใช้งาน
B	เครื่องจักรเริ่มมีการสั่นสะเทือน จะส่งผลเสียในระยะยาว
C	เครื่องจักรมีการสั่นสะเทือนรุนแรง ถ้าปล่อยไว้จะมีผลเสียกับเครื่องจักรในระยะเวลาอันสั้น
D	เครื่องจักรมีการสั่นสะเทือนรุนแรงมาก อาจทำให้เครื่องจักรเสียหายแบบฉุกฉิน



รูปที่ 2.11 เปรียบเทียบการสั่นสะเทือนต่อหน่วยการวัด
(ที่มา: อัยญาช รอดพ่าย, 2559)

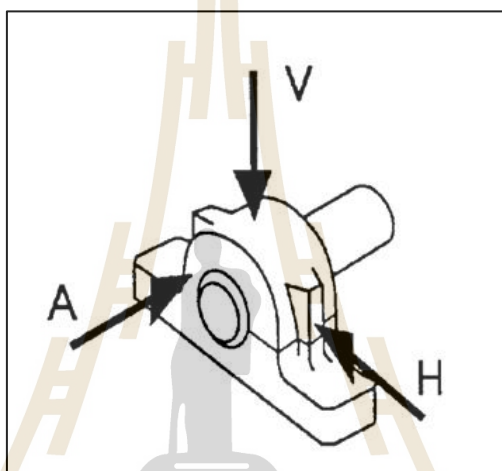
2.8.1.2 หน่วยการวัดการสั่นสะเทือน

การมีความเข้าใจพื้นฐานการตรวจวัดการสั่นสะเทือน โดยเฉพาะเรื่องหน่วยในการวัดการสั่นสะเทือนว่ามีความเหมาะสมกับอุปกรณ์ หรือสภาพความเสียหายแบบใด ดังรูปที่ 2.11 ในการประเมินความเสียหายของเครื่องจักรด้วยการตรวจวัดการสั่นสะเทือน ในหน่วยความเร็วเหมาะสำหรับการประเมินระดับความเสียหายตามมาตรฐาน ISO 10816 เนื่องจากความถี่ไม่มีผลกับขนาดการสั่นสะเทือน ส่วนหน่วยความเร่งใช้ตรวจสอบความผิดปกติของตลับลูกปืน และชุดเกียร์ว่ามีความเสียหายเกิดขึ้นหรือยัง เพราะความเสียหายของตลับลูกปืนและชุดเกียร์เมื่อเริ่มเสียหายจะส่งสัญญาณการสั่นสะเทือนที่ความถี่สูง ซึ่งจะสูงกว่า 500 Hz ออกมา ถ้าสามารถ

ตรวจพบได้ก่อนจะทำให้เราเตรียมตัววางแผนงานได้ดียิ่งขึ้น ถ้าผู้ใช้ใช้แต่หน่วยความเร็วอาจจะไม่ทันการในการป้องกันความเสียหายของคลัสเตอร์ก็เป็นได้

2.8.1.3 ตำแหน่งการวัดการสั่นสะเทือน

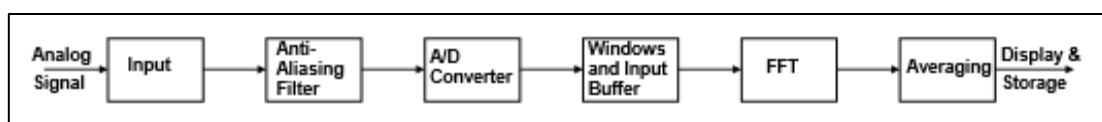
กำหนดตำแหน่งการวัดและทิศทางการวัดให้เหมาะสม ซึ่งปกติตำแหน่งการวัดการสั่นสะเทือนจะทำการวัดที่ตำแหน่งแบร์ริงในทิศทางแบบ 3 แกน คือ แกนแนวนอน (Horizontal), แกนแนวตั้ง (Vertical) และแนวแกน (Axial) ดังรูปที่ 2.12



รูปที่ 2.12 ตำแหน่งการวัดการสั่นสะเทือน
(ที่มา: อัยยุทธ รอดพ่าย, 2559)

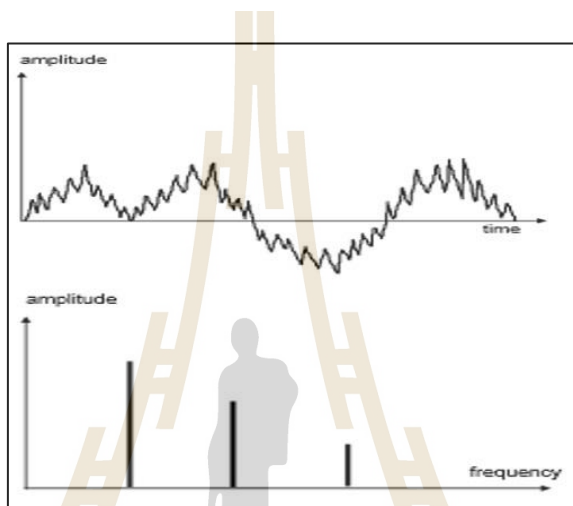
2.8.2 การจัดการสัญญาณการสั่นสะเทือน

จากสัญญาณนั้นจะยังไม่สามารถนำมาวิเคราะห์ปัญหาการสั่นสะเทือนได้โดยตรง เนื่องจากว่าสัญญาณเหล่านั้นเต็มไปด้วยค่าความถี่หลากหลายที่รวมกันอยู่ ซึ่งจะต้องทำการจัดการกับสัญญาณดังกล่าวด้วยวิธีการดังนี้



รูปที่ 2.13 การจัดการสัญญาณเพื่อนำไปใช้วิเคราะห์ความเสียหาย
(ที่มา: Dennis H. Shreve, 1995)

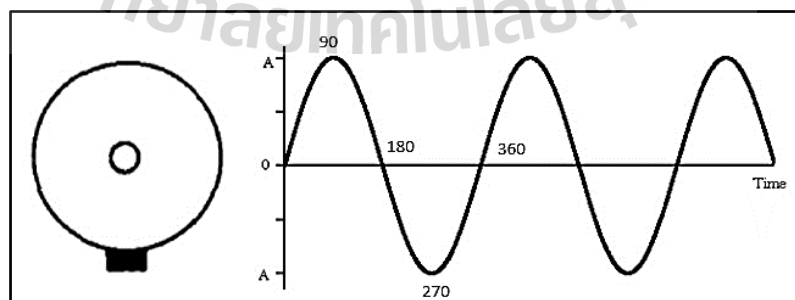
จากรูปที่ 2.13 นั้นเมื่อสัญญาณได้ Analog เข้ามาจะต้องทำการกรองความถี่ที่ไม่ต้องการออกไปก่อน ซึ่งจะเป็นจำพวกความถี่ของสัญญาณรบกวนต่าง ๆ ที่เข้ามา จากนั้นจะทำการแปลงสัญญาณจาก Analog ให้กลายเป็น Digital เพื่อที่จะทำการเก็บข้อมูลของสัญญาณแบบ Real Time ต่อมาจะทำการแยกความถี่ด้วย Fast Fourier Transform เพื่อที่จะทำให้สามารถแสดงผลการสั่นสะเทือนในแต่ละความถี่ได้



รูปที่ 2.14 สัญญาณก่อนเข้ากระบวนการและหลังเข้ากระบวนการ
(ที่มา: Dennis H. Shreve, 1995)

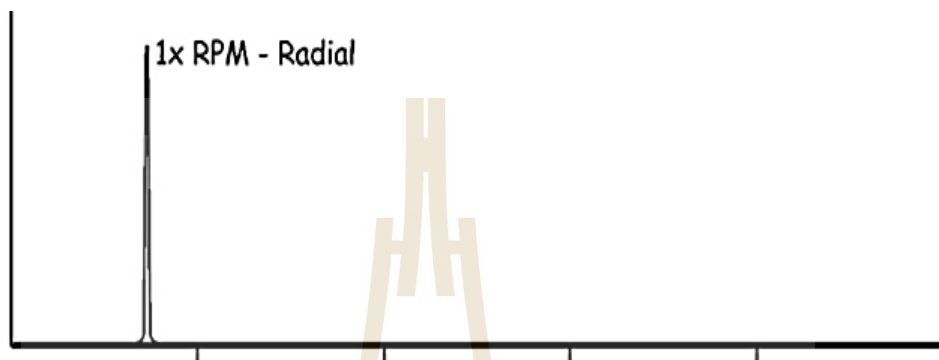
2.9 ความไม่สมดุลทางกล

เมื่อเกิดความไม่สมดุลในแผ่นเหยียงหมุนจานหมุนจะส่งแรงหนีศูนย์กลางออกมาตลอดการทำงานดังรูปที่ 2.15



รูปที่ 2.15 สัญญาณจากความไม่สมดุลของจานหมุนในโดเมนเวลา

แรงหนีศูนย์กลางจากความไม่สมดุลจะส่งผลให้เกิดการสั่นสะเทือน โดยการสั่นที่ได้จะตรงกับความถี่ในการปั่นเลือดเสมอ นั่นคือการสั่นที่ 1 เท่าของความถี่รอบในแนว Radial เสมอ เมื่อทำการจัดการสัญญาณให้อยู่ในโดเมนความถี่แล้วจะได้ดังรูปที่ 2.16

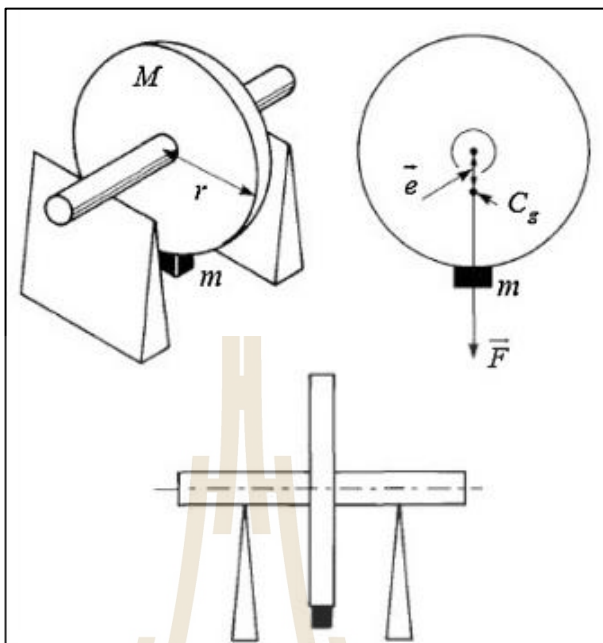


รูปที่ 2.16 สัญญาณจากความไม่สมดุลจานหมุนในโดเมนความถี่
(ที่มา: Md. Abdul Saleem และคณะ, 2012)

จากปัญหาที่เกิดขึ้นจะมีการสั่นสะเทือนเกิดขึ้นเนื่องจากมีความไม่สมดุล ซึ่งความไม่สมดุลในการหมุน (Unbalance) จะเป็นความไม่สมดุลแบบสถิตย์ (Static Unbalance) เกิดได้หลายกรณี เช่น เกิดจากกระบวนการผลิตที่ผิดพลาดจากโรงงาน เกิดจากการใช้งานเป็นเวลานาน เป็นต้น ซึ่งความไม่สมดุลในการหมุนของเครื่องปั่นเลือดนี้จะเกิดจากจุดศูนย์กลางมวลของจานหมุนไม่ตรงกับจุดศูนย์กลางของเพล่าจะคำนวณได้จากสมการ

$$U = mr \quad (2.8)$$

โดยที่ U คือ ความไม่สมดุลที่เกิดขึ้นบนจานหมุน (g.mm)
 m คือ ขนาดของมวลที่ทำให้เกิดความไม่สมดุลในการหมุน (g)
 r คือ ระยะห่างของมวลที่ทำให้เกิดความไม่สมดุลกับศูนย์กลางการหมุน (mm)



รูปที่ 2.17 ความไม่สมดุลของแผ่นจานหมุน

(ที่มา: Md. Abdul Saleem และคณะ, 2012)

เมื่อเครื่องจักรมีความไม่สมดุลในงานหมุนจะทำให้เกิดแรงเหวี่ยงหนีศูนย์กลาง เนื่องจากความไม่สมดุลนี้ สามารถคำนวณได้จากสมการ

$$F = U\omega^2 \sin \omega t \quad (2.9)$$

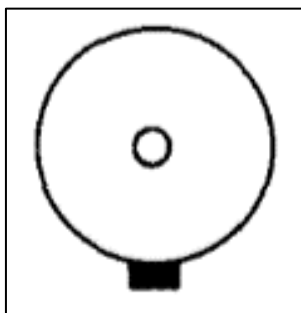
โดย F คือ แรงหนีศูนย์กลางที่เกิดจากความไม่สมดุลในงานหมุน (N)

U คือ ความไม่สมดุลที่เกิดขึ้นบนงานหมุน (g.mm)

ω คือ ความเร็วรอบในการปั่นเลือด (rad/s)

แรงหนีศูนย์กลางจากความไม่สมดุลในการหมุนนั้นจะส่งผลทำให้เกิดการสั่นสะเทือนขึ้น และจากสมการที่ 2.9 จะเห็นได้ว่าเมื่อความเร็วรอบสูงขึ้น ทำให้มีแรงหนีศูนย์กลางมากขึ้น การสั่นสะเทือนที่เกิดขึ้นก็จะสูงขึ้นตามความเร็วรอบในการหมุน

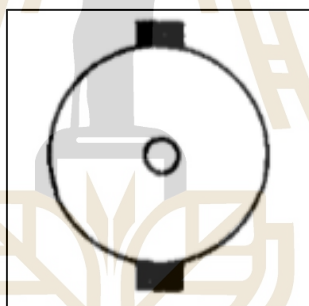
การแก้ไขปัญหาความไม่สมดุลในแผ่นปั่นเหวี่ยงนั้นเนื่องจากความไม่สมดุลที่เกิดขึ้นในงานหมุนจะเป็นแบบสถิต



รูปที่ 2.18 ความไม่สมดุลเนื่องจากมีมวลถ่วง

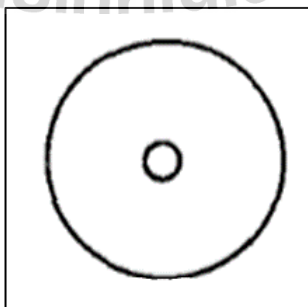
เพราะฉะนั้นจะสามารถทำการแก้ไขได้ 2 แบบคือ

- ตัดมวลเพิ่มในฝั่งตรงข้ามที่มีขนาดเท่ากันเพื่อให้เกิดแรงหนีศูนย์กลางในฝั่งตรงข้ามหักล้างกับแรงหนีศูนย์กลางที่ทำการเกิดความไม่สมดุล



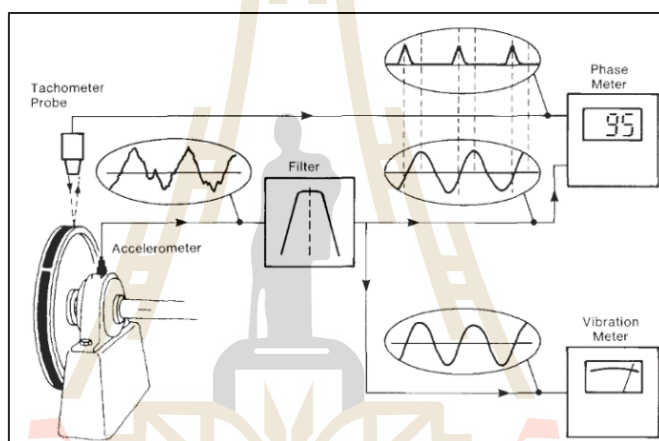
รูปที่ 2.19 แก้ไขโดยการตัดมวลขนาดเท่ากันฝั่งตรงข้าม

- ตัดมวลส่วนที่เกินในจานหมุนนั้นออกเพื่อกำจัดแรงหนีศูนย์กลาง



รูปที่ 2.20 แก้ไขโดยการตัดมวลที่ทำให้เกิดความไม่สมดุลออก

ในความเป็นจริงการที่จะทราบถึงตำแหน่งของมวลที่ทำให้เกิดความไม่สมดุลนั้น ไม่สามารถมองหรือคาดการณ์ได้จากการมองด้วยตาเปล่า จะต้องอาศัยอุปกรณ์ในการช่วยนั่นคือ Sensor วัดการสั่นสะเทือน และ Sensor วัดเฟสการหมุนของจานหมุน โดยจะติด Sensor วัดการสั่นสะเทือนเพื่อที่จะวัดการสั่นที่เกิดขึ้นไว้บริเวณแบร์ริงของอุปกรณ์ซึ่งเป็นจุดที่ส่งการสั่นสะเทือนมายัง Sensor ได้มากที่สุด และ Sensor วัดเฟสจะติดตั้งเพื่อวัดและเก็บค่าเฟสของการหมุน จากนั้นจะนำสัญญาณ 2 ชนิดที่วัดได้มาประมวลผลการสั่นสะเทือนและแรงที่เกิดขึ้นในเฟสการหมุน ซึ่งความถี่ที่วัดได้จากการสั่นสะเทือนจะต้องเท่ากับกับความถี่ของความถี่รอบการหมุนด้วย



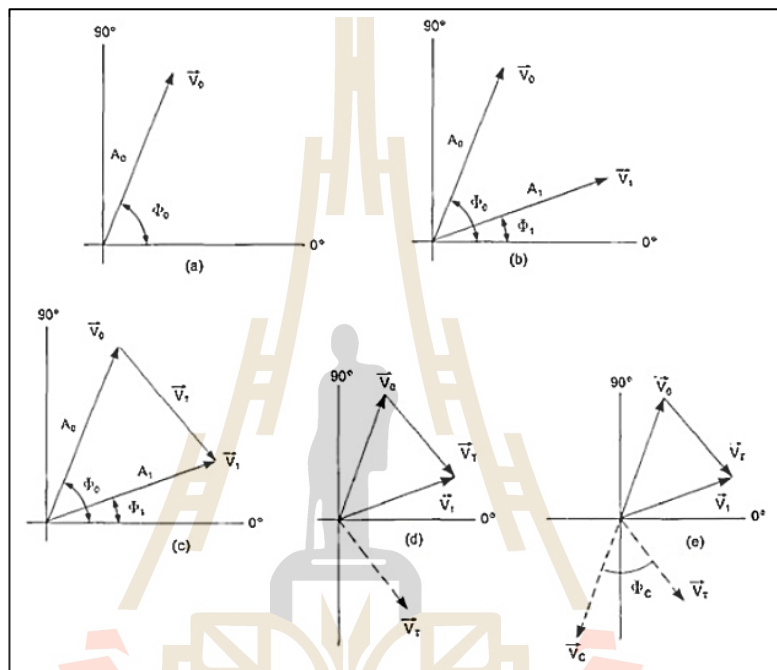
รูปที่ 2.21 การวัดการสั่นสะเทือนและเทียบเฟสของสการสั่นสะเทือน
(ที่มา: Guilherme Kenji Yamamoto และคณะ, 2016)

2.9.1 ขั้นตอนการทำสมดุลแบบกราฟิก

เมื่อติดตั้งอุปกรณ์การวัดเรียบร้อยแล้วจะทำการแก้ไขความไม่สมดุลในแผ่นหมุนเหวี่ยงนั้นจะทำการแก้ไขด้วยวิธีการดังต่อไปนี้

- ทำการเดินเครื่องที่ความเร็วรอบที่ใช้งานหรือความเร็วรอบต่ำที่ไม่ตรงกับความเร็วธรรมชาติของเครื่องแล้ววัดการสั่นสะเทือนและมุมเฟสของแรงเหวี่ยงที่เกิดขึ้น จะได้ Vector ของแรงขนาด V_0 แล้วบันทึก ลงใน Polar Plot
- ทำการถ่วงมวลที่ทราบน้ำหนักและติดลงในแผ่นจานในมุมที่เราทราบจากนั้นทำการเดินเครื่องที่ความเร็วรอบเดิม วัดการสั่นสะเทือนและมุมเฟสของแรงเหวี่ยงที่เกิดขึ้น จะได้ Vector ของแรงขนาด V_T แล้วบันทึก ลงใน Polar Plot
- ทำการลาก Vector จากปลายแรง V_0 ไปยัง V_T จะได้ Vector V_1 มา

- ย้าย Vector V_1 มายังจุดศูนย์กลางทั้งขนาดและทิศทาง
- ทำการกลับ Vector V_0 ให้อยู่ตรงข้ามแล้ววัดมุมจาก Vector V_0 ถึง V_1 ตามทิศทางการหมุนของแผ่นเหวี่ยง จะทำให้เราทราบมุมที่ต้องถ่วงแก้ไขความไม่สมดุลในแผ่นเหวี่ยงหมุน



รูปที่ 2.22 การหาความไม่สมดุลในแผ่นเหวี่ยงหมุน

(ที่มา: Guilherme Kenji Yamamoto และคณะ, 2016)

- คำนวณค่าของมวลที่ทำให้เกิดความไม่สมดุลโดยจะคำนวณได้จากสมการ

$$M_0 = M_T \left(\frac{|V_T|}{|V_0|} \right) \quad (2.10)$$

โดยที่	M_0	คือ มวลที่จะต้องถ่วงแก้ไข (g)
	V_0	คือ ขนาดของแรงที่ไม่สมดุล (mm/s)
	V_T	คือ ขนาดของแรงลัพธ์ (mm/s)
	M_T	คือ มวลที่ใช้ในการถ่วง Trial (g)

2.9.2 ขั้นตอนการทำสมดุลแบบคำนวณสมการ

- ควบคุมความเร็วรอบของมอเตอร์ให้คงที่
- เปิดเครื่องปั่นหมุนเหวี่ยงทำการวัด Amplitude (V_0) การสั่นสะเทือน และ Phase (ϕ_0) ของการสั่นสะเทือนของเครื่องปั่นหมุนเหวี่ยง
- ปิดเครื่องปั่นหมุนเหวี่ยง ทำการ Trial weight โดยการติดมวลขนาดเท่าใดก็ได้ (M_T) ที่มุมเท่าใดก็ได้ (θ_T)
- เปิดเครื่องปั่นหมุนเหวี่ยง ทำการวัด Amplitude (V_1) การสั่นสะเทือน และ Phase (ϕ_1) ของการสั่นสะเทือนของเครื่องปั่นหมุนเหวี่ยง
- คำนวณหามวล (M_c) และมุม (θ_b) แก้ไขความไม่สมดุลจากสมการ

$$V_T = \sqrt{(V_x)^2 + (V_y)^2} \quad (2.11)$$

โดยที่ $V_x = V_1 \cos \phi_1 - V_0 \cos \phi_0$ และ $V_y = V_1 \sin \phi_1 - V_0 \sin \phi_0$

$$\phi_T = \frac{V_y}{V_x} \quad (2.12)$$

$$M_c = M_T \left(\frac{|V_T|}{|V_0|} \right) \quad (2.13)$$

$$\theta_c = \theta_T + \phi_T - \phi_B \quad \begin{cases} \phi_B = \phi_0 + 180^\circ & ; \quad 0^\circ \leq \phi_0 < 179^\circ \\ \phi_B = \phi_0 - 180^\circ & ; \quad 180^\circ \leq \phi_0 < 359^\circ \end{cases} \quad (2.14)$$

โดย θ_c คือ มุมที่ใช้ถ่วงแก้ไข

- ปิดเครื่องปั่นเหวี่ยงหมุน ถอดมวล Trial ออกแล้วติดมวล M_c ที่ θ_b บนใบจาน

2.10 ปรัชญ่วรรณกรรมที่เกี่ยวข้อง

การทำเครื่องถ่วงสมดุลสำหรับเครื่องปั่นเลือดนั้นจะต้องใช้องค์ความรู้ต่าง ๆ ทั้งในด้านทางทฤษฎีของการแยกตัวของเลือด เทคนิคของการปั่นแยกเลือด อุปกรณ์ที่เหมาะสมในการนำมาใช้งาน เทคนิคการใช้งานอุปกรณ์ภายใน หลักการของความไม่สมดุลที่เกิดขึ้น

เพื่อนำมาออกแบบและสร้างขึ้นสำหรับเครื่องถ่วงสมดุลสำหรับเครื่องปั่นเลือด จึงได้ศึกษา งานวิจัยที่เกี่ยวข้องดังนี้

ชูชาติ อารีจิตรานุสรณ์ (2544) ศึกษาทฤษฎีเกี่ยวกับเครื่องหมุนเหวี่ยงชนิดต่าง ๆ เทคนิค และประเภทของเครื่องหมุนเหวี่ยงเพื่อแยกสสารออกจากกัน โดยจะอาศัยแรงเหวี่ยงหนีศูนย์กลาง ระหว่างการเหวี่ยงหมุนซึ่งแรงเหวี่ยงหนีศูนย์กลางนี้สามารถคำนวณได้จากโมโนแกรม เมื่อทราบ ความเร็วรอบและรัศมีของอุปกรณ์ที่ใช้หมุนเหวี่ยง

Manjita Srivastava, M C Srivastava, Smriti Bhatnagar (2009) ศึกษาทฤษฎีเกี่ยวกับระบบ ควบคุมอัตโนมัติ โดยจะกล่าวถึงหลักการเกี่ยวกับระบบควบคุมแบบปิด พร้อมทั้งกรรมวิธีใน การออกแบบ Controller สำหรับระบบต่าง ๆ ที่นำมาใช้งาน

G Prasad, Venkateswara Reddy M, Dr. P V N Prasad, Dr. G Tulasi Ram Das (2012) ศึกษา เกี่ยวกับการทำงานของมอเตอร์ BLDC และอุปกรณ์ควบคุมที่ใช้กับมอเตอร์ BLDC โดยงานวิจัยนี้ จะนำเสนอเกี่ยวกับการพัฒนาระบบควบคุมของมอเตอร์ BLDC โดยใช้ชุดควบคุมของสัญญาณ Digital (ชุด kit MCK28335) และเปรียบเทียบผลตอบสนองที่ได้จริงกับผลการจำลองผ่านโปรแกรม

Pawin Jawayon, Jiraphon Srisertpol (2013) ศึกษาเกี่ยวกับแบบจำลองทางคณิตศาสตร์ของ มอเตอร์ BLDC และตัวแปรที่เกี่ยวข้อง โดยงานวิจัยนี้จะนำเสนอเกี่ยวกับการประมาณความเร็วของ มอเตอร์ BLDC โดยเทียบความสัมพันธ์ระหว่างสัญญาณที่สื่อสารกับแรงเคลื่อนไฟฟ้าด้านกลับ โดยสัญญาณทั้งคู่จะสามารถวัดได้โดยตรงจากแต่ละเฟสของขั้วมอเตอร์ ซึ่งฟังก์ชันความสัมพันธ์ ระหว่างสัญญาณทั้งสองจะถูกประมาณและกรองหา Half of Phase Voltage-Crossing Point ใช้เพื่อ ประมาณค่าสัมพันธ์ของสมการพหุนามระหว่างตัวแปร และระยะเวลาของสัญญาณที่เปลี่ยน ซึ่งผลลัพธ์ของความเร็วโดยประมาณจะถูกต้องมากกว่า 97% และค่าเฉลี่ยของช่วงการจ่ายแรงดันจะ ต่ำกว่า 9 มิลลิวินาที

Ms.Manjusha Patil (2014) ศึกษาเกี่ยวกับการเลือกใช้และออกแบบ PID Controller เพื่อที่ จะนำมาใช้ในงานการควบคุมมอเตอร์ BLDC ซึ่งงานวิจัยนี้นำเสนอเกี่ยวกับระบบควบคุมที่มี ความแม่นยำและมีการทำงานที่เสถียรโดยจะพัฒนาระบบควบคุม PI และ PID และพิจารณา เพื่อเลือกชนิดของระบบควบคุมซึ่งผลที่ได้จากการทดสอบกับมอเตอร์ DC เปรียบเทียบระหว่าง ระบบควบคุม PI และ PID นั้นมีผลตอบสนองที่ใกล้เคียงกันสำหรับมอเตอร์ DC โดย PID จะมี ผลตอบสนองชั่วคราวดีกว่า

Viral K. Patel, Maitri N. Patel (2017) ศึกษาเกี่ยวกับการใช้งาน Accelerometer กับ Arduino Microcontroller เพื่อนำมาใช้สร้างเครื่องถ่วงสมดุล โดยงานวิจัยนี้จะนำเสนอเกี่ยวกับการทดสอบ การทำงานของ Accelerometer ที่ใช้กับ Arduino Microcontroller ในการวัดความถี่สูงซึ่งจะติดตั้งวัด การตั้งอยู่ที่ชุดแปริ่งในเครื่องจักร โดยจะวัดการสั่นสะเทือนและส่งไปยังคอมพิวเตอร์แสดงผล

ซึ่งผลที่ได้คืออุปกรณ์ Arduino Microcontroller ที่มีความเร็วสัญญาณนาฬิกา 16 MHz สามารถจับสัญญาณของการสั่นสะเทือนที่ความถี่สูงจาก Accelerometer ได้

Dennis H. Shreve (1995) ศึกษาเกี่ยวกับกรรมวิธีการจัดการสัญญาณการสั่นสะเทือนที่ได้จาก Accelerometer เพื่อนำไปใช้งาน ซึ่งงานวิจัยนี้จะนำเสนอเกี่ยวกับการจัดการสัญญาณการสั่นสะเทือนที่วัดจากเครื่องจักรโดยจะกรองความถี่ของสัญญาณด้วย High-Pass และ Low-Pass ขึ้นอยู่กับผลลัพธ์ที่ต้องการ ต่อมาทำการวิเคราะห์หรัสคลื่นเวลาในเครื่องมือ Digital ซึ่งต้องกำหนดจำนวนและอัตราของตัวอย่างข้อมูลที่เข้ามา ประมวลผลแบบ FFT (Fast Fourier Transform) เพื่อแบ่งความถี่ในแต่ละส่วนและแสดงผล

Md. Abdul Saleem, G. Diwakar, Dr. M.R.S. Satyanarayana (2012) ศึกษาเกี่ยวกับการสั่นเนื่องจากความไม่สมดุลในการหมุน ซึ่งงานวิจัยนี้จะนำเสนอเกี่ยวกับการเปลี่ยนรูปทรงของแกนหมุนซึ่งทำให้เกิดความไม่สมดุลในการหมุนและเกิดการสั่นสะเทือนขึ้น โดยจะทดลองเปลี่ยนรูปทรงของการ โกงของเพลลาเพื่อคุณลักษณะของการเกิดความไม่สมดุล และดูข้อผิดพลาดที่เกิดขึ้นเพื่อเป็นแนวทางในการตรวจหาความไม่สมดุลของเครื่องจักรในสายการผลิต

Guilherme Kenji Yamamoto, Cesarda Costa, João Sinoharada Silva Sousa (2016) ศึกษาเกี่ยวกับกรรมวิธีการแก้ไขความไม่สมดุลในการหมุน โดยงานวิจัยนี้จะนำเสนอเกี่ยวกับการสั่นสะเทือนของโรเตอร์ที่ไม่สามารถสร้างความสมดุลได้อย่างสมบูรณ์เนื่องจากความผิดพลาดในการผลิตหรือเกิดรูพรุนในวัสดุ ซึ่งจะเสนอวิธีการทดลองการตรวจวัดแบบอัจฉริยะสำหรับการตรวจวัดการสั่นสะเทือนและการตรวจสอบความผิดปกติเนื่องจากความไม่สมดุลในเครื่องจักรในการหมุน

Chigbogu C. Ozoegwu, Christian. C. Nwangwu, Chigozie F. Uzoh, Arinze V. Ogunoh (2012) ศึกษาเกี่ยวกับสมการ Balancing ซึ่งงานวิจัยนี้จะนำเสนอเกี่ยวกับสาเหตุของการสั่นสะเทือนของเครื่องจักรซึ่งอาจเกิดจากความไม่สมดุล การเยื้องศูนย์ หรือการหลวมคลอนทางกล โดยการทดสอบเรื่องความไม่สมดุลจะหลีกเลี่ยงวิธีแบบกราฟิกที่ยากต่อการเข้าใจและเกิดข้อผิดพลาดได้ง่าย ซึ่งจะเขียนโปรแกรม MATLAB เพื่อคำนวณการถ่วงสมดุลจากสมการ Balancing

บทที่ 3

วิธีดำเนินการวิจัย

การสร้างเครื่องถ่วงสมดุลเพื่อที่จะแก้ไขความไม่สมดุลในแผ่นงานปั้นเลือดนั้นจะใช้หลักการเดียวกันกับการแก้ไขความความไม่สมดุลในเครื่องจักรอื่น ๆ โดยจะต้องอาศัยอุปกรณ์หลัก 2 อย่าง ได้แก่

3.1 เครื่องปั้นงาน

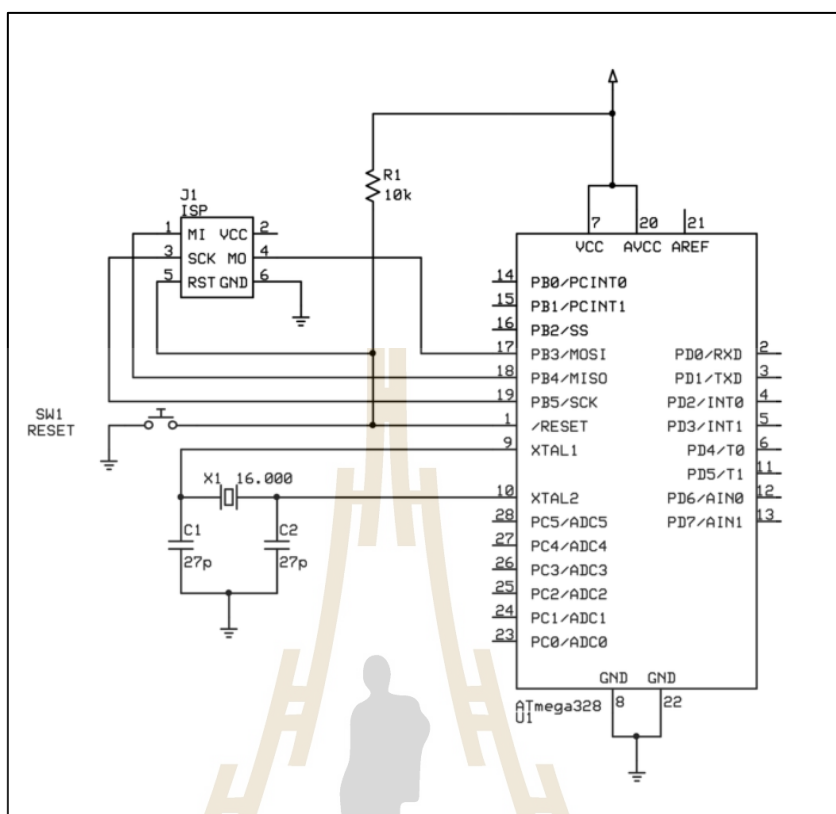
เครื่องปั้นงานเป็นเครื่องที่ใช้สำหรับปั้นงานเลือดที่ใช้ในเครื่องปั้นเลือด เพื่อตรวจสอบปัญหาการสันสะเก็ดเนื่องจากความไม่สมดุลของงานเลือด ซึ่งต้นกำลังจะใช้เป็นมอเตอร์ที่ออกแบบระบบควบคุมไว้เพื่อที่จะควบคุมการควบคุมความเร็วรอบในการหมุนของงานเลือด โดยใช้ Microcontroller เป็นตัวควบคุมเครื่องปั้นงานนี้ ซึ่งจะใช้อุปกรณ์ดังนี้

3.1.1 Arduino Uno R3



รูปที่ 3.1 Arduino Uno R3

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source หรือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ซึ่ง Arduino รุ่น Uno R3 จะใช้ IC เบอร์ Atmega328PU ซึ่งภายในบอร์ดจะมีวงจรไฟฟ้า ดังภาพที่ 3.1



รูปที่ 3.2 วงจรไฟฟ้าของ Arduino Uno R3

การใช้งานสำหรับ Arduino Uno R3 จะมีคุณสมบัติดังนี้

- ใช้ IC เบอร์ ATmega328P
- ทำงานที่แรงดัน 5 V
- แรงดันเข้าออร์ดอยู่ที่ 7-12V
- แรงดันที่บอร์ดรับได้อยู่ที่ 6-20V
- Digital Input 14 Port
- Digital Output 14 Port
- PWM Output 6 Port
- Analog Input 6 Port
- น้ำหนัก 25 กรัม

3.1.2 Liquid Crystal Display (LCD)

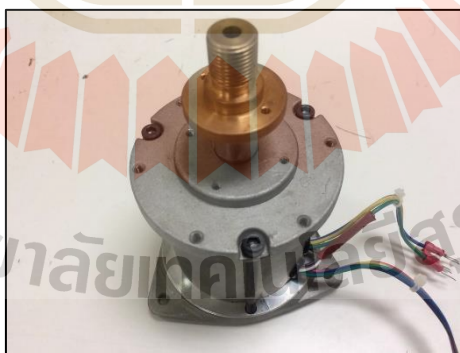


รูปที่ 3.3 จอ LCD แสดงผล 16x2

จอ LCD เป็นจอที่ใช้แสดงผลของความเร็วรอบตั้งต้นที่จะให้เครื่องปั่นงานทำงาน และทำการแสดงผลของความเร็วรอบระหว่างการปั่นงานแบบ Real Time โดยมีคุณสมบัติดังนี้

- ขนาดตัวอักษร 16x2 Character
- ทำงานที่แรงดัน 5 V
- การสื่อสารเป็นแบบ I2C

3.1.3 Brushless Direct Current Motor

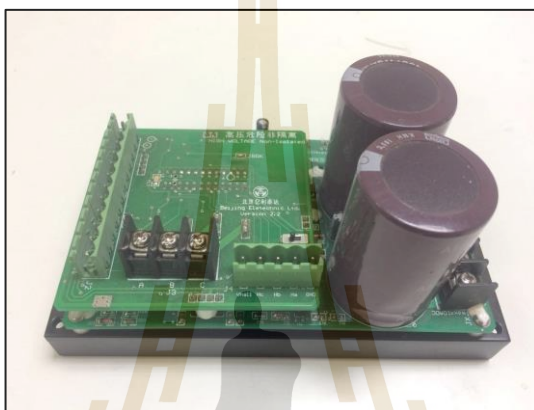


รูปที่ 3.4 Brushless Direct Current Motor

Brushless Direct Current Motor ใช้รุ่น F72-201105-220 เป็นมอเตอร์กระแสตรงที่ไม่มีแปรงถ่านซึ่งจะนำมาใช้ในการปั่นงานเพื่อที่จะวัดการสั่นสะเทือนและถ่วงสมดุล โดยมีคุณสมบัติดังนี้

- ความเร็วสูงสุด 25,000 รอบ/นาที
- ใช้งานที่อุณหภูมิ 0-70 องศาเซลเซียส
- มี Hall Effect Sensor ในตัว
- ควบคุมผ่าน BLMD-8TC39-1P-S14 Driver

3.1.4 Driver สำหรับ Brushless Direct Current Motor



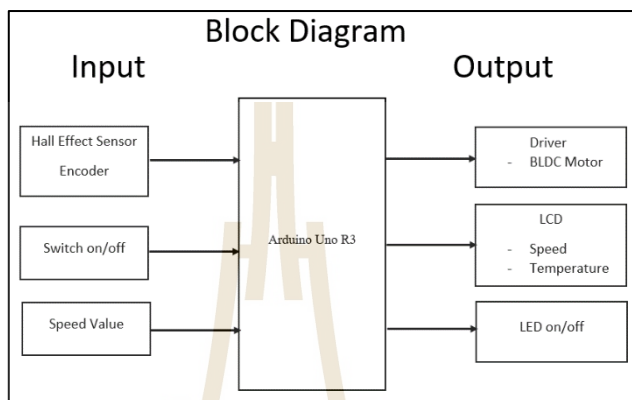
รูปที่ 3.5 Driver สำหรับ Brushless Direct Current Motor

Driver สำหรับ Brushless Direct Current Motor จะใช้รุ่น BLMD-8TC39-1P-S14 โดยอุปกรณ์นี้จะใช้ควบคุมมอเตอร์ Brushless Direct Current Motor ให้สามารถหมุนได้ ซึ่งสามารถควบคุมได้ทั้งแบบ Manual และแบบ Auto โดยสั่งจาก Microcontroller ซึ่งมีคุณสมบัติการใช้งานดังนี้

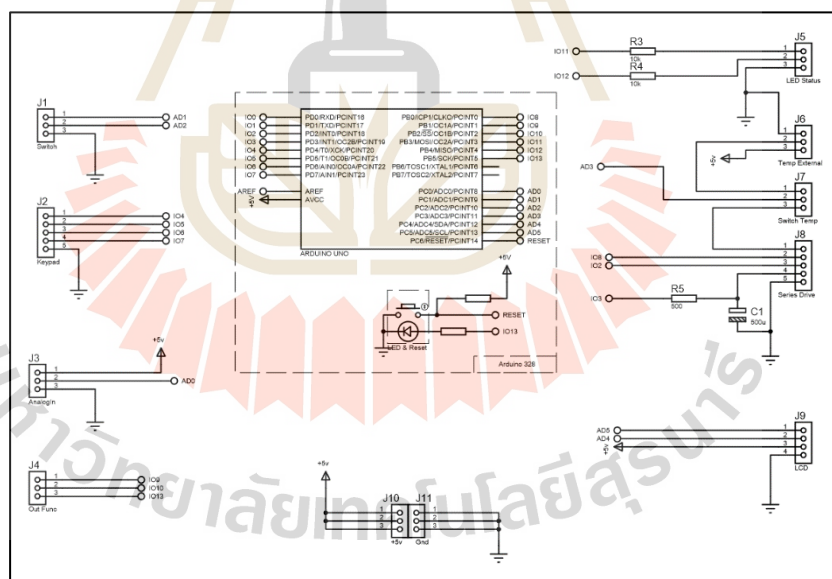
- ใช้ไฟเลี้ยง 220 VAC
- แปลงสัญญาณเป็น Six Step จ่ายควบคุมมอเตอร์
- ใช้สัญญาณ Analog เพื่อควบคุมความเร็วรอบของมอเตอร์
- สามารถกลับทิศทางมอเตอร์ได้
- ทำงานที่อุณหภูมิ -20 ถึง 85 องศาเซลเซียส
- จ่ายไฟเลี้ยง Hall Effect Sensor 5.5 V
- รับสัญญาณจาก Hall Effect และแปลงค่าเป็นความถี่โดยตรง
- มีระบบป้องกันภายในตัวเองเมื่อเกิด Over Load
- สามารถวัดอุณหภูมิภายในตัวได้

3.1.5 แผนภาพและวงจรควบคุมเครื่องปั่นงาน

จากอุปกรณ์ข้างต้นที่กล่าวมาจะต้องนำมาประกอบการใช้งานเพื่อสร้างเครื่องปั่นงาน ซึ่งแผนภาพของการเชื่อมต่ออุปกรณ์จะเป็นดังรูปที่ 3.6 และวงจรไฟฟ้าจะเป็นดังรูป 3.7



รูปที่ 3.6 แผนภาพของการเชื่อมต่ออุปกรณ์ของเครื่องปั่นงาน



รูปที่ 3.7 แผนภาพวงจรไฟฟ้าที่ออกแบบสำหรับเครื่องปั่นงาน

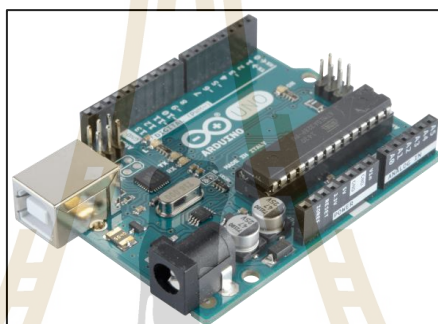
3.2 เครื่องถ่วงสมดุล

เครื่องถ่วงสมดุลจะเป็นเครื่องที่ใช้วัดและวิเคราะห์ความไม่สมดุลในงานปั่นเลือดและคำนวณหาจุดที่ทำให้งานปั่นเลือดมีความสมดุลพร้อมมวลแก้ไข ซึ่งจะต้องมีการวัดการสั่นสะเทือน

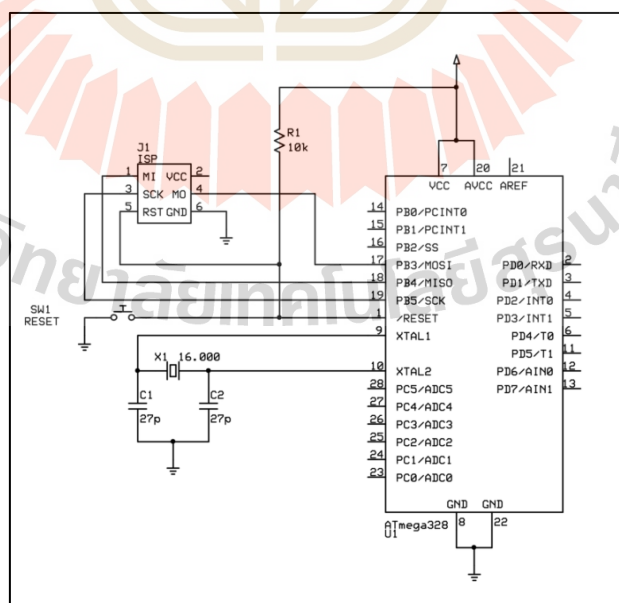
ที่เกิดจากงานปั่นเลือดพร้อมทั้งวัดความเร็วรอบ และคำนวณหาเฟสการสั้นสะเทือนที่สูงที่สุดที่เกิดขึ้น โดยจะมี Microcontroller เป็นตัวรับค่า Sensor และประมวลผลคำนวณหาผลและมุมที่ติด เพื่อให้งานปั่นเลือดเกิดความสมดุล โดยจะใช้อุปกรณ์ดังนี้

3.2.1 Arduino Uno R3

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source หรือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ซึ่ง Arduino รุ่น Uno R3 จะใช้ IC เบอร์ Atmega328PU ซึ่งภายในบอร์ดจะมีวงจรไฟฟ้าง่ายๆที่ 3.9



รูปที่ 3.8 Arduino Uno R3



รูปที่ 3.9 วงจรไฟฟ้าของ Arduino Uno R3

การใช้งานสำหรับ Arduino Uno R3 จะมีคุณสมบัติดังนี้

- ใช้ IC เบอร์ ATmega328P
- ทำงานที่แรงดัน 5V
- แรงดันเข้าบอร์ดอยู่ที่ 7-12 V
- แรงดันที่บอร์ดรับได้อยู่ที่ 6-20 V
- มี Digital Input 14 Port
- มี Digital Output 14 Port
- มี PWM Output 6 Port
- มี Analog Input 6 Port
- น้ำหนัก 25 กรัม

3.2.2 Liquid Crystal Display (LCD)

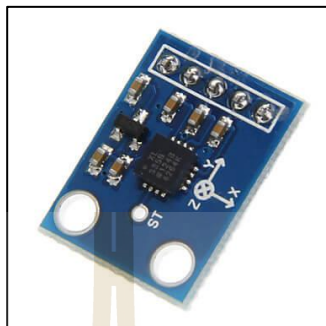


รูปที่ 3.10 จอ LCD แสดงผล 20x4

จอ LCD เป็นจอที่ใช้แสดงผลของความเร็วรอบตั้งต้นที่จะให้เครื่องปั่นงานทำงาน และทำการแสดงผลของความเร็วรอบระหว่างการปั่นงานแบบ Real Time โดยมีคุณสมบัติดังนี้

- ขนาดตัวอักษร 20x4 Character
- ทำงานที่แรงดัน 5 V
- การสื่อสารเป็นแบบ I2C

3.2.3 Accelerometer



รูปที่ 3.11 Accelerometer ADXL335

Accelerometer จากเครื่องถ่วงสมดุลจะใช้รุ่น ADXL335 หรืออีกชื่อหนึ่งคือ GY-61 ซึ่งเซ็นเซอร์ชนิดนี้จะใช้วัดการสั่นสะเทือนของเครื่องปั้นงานที่ยึดติดกับงานปั้นเลือดก่อนส่งให้ Microcontroller วิเคราะห์ผลต่อไป ซึ่งจะมีคุณสมบัติดังนี้

- ทำงานที่แรงดัน 3.3 V ให้ Output เป็น Analog แรงดัน 0-3 V
- สามารถวัดได้ 3 แกน ได้แก่ แกน X, แกน Y และ แกน Z
- Sensitivity 300 mV/g
- วัดการสั่นได้สูงสุด 3 g
- แกน X กับ แกน Y วัดการสั่นสะเทือนที่ความถี่ไม่เกิน 1,600 Hz
- แกน Z วัดการสั่นสะเทือนที่ความถี่ไม่เกิน 550 Hz

3.2.4 Keypad



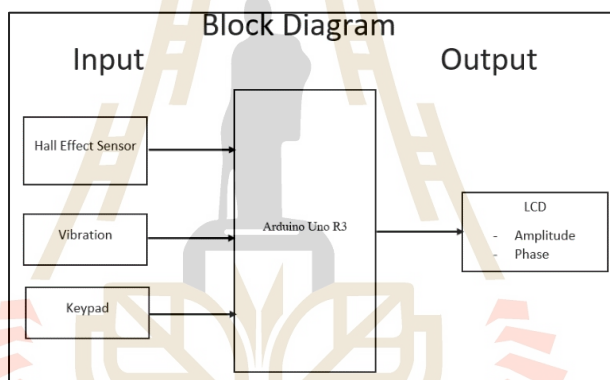
รูปที่ 3.12 Keypad 1x4

Keypad จะเปรียบเสมือนเป็น Switch ชนิดหนึ่งซึ่งในเครื่องถ่วงสมดุลจะใช้ขนาด 1x4 Switch ใช้สำหรับกดดำเนินการตามขั้นตอนของการถ่วงสมดุลงานปั่นเลือด โดย ในแต่ละปุ่มจะมีคุณสมบัติดังนี้

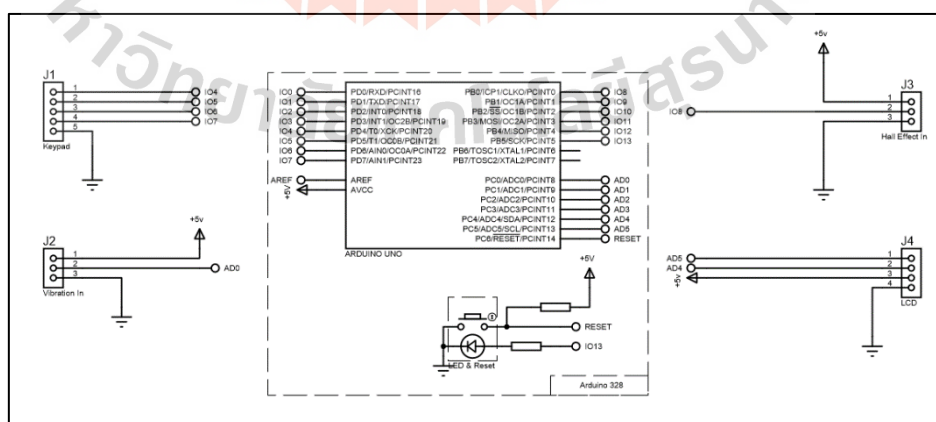
- ปุ่ม 1 จะใช้กดเพื่อถอยหลังไปยังขั้นตอนก่อนหน้า
- ปุ่ม 2 จะใช้กดเพื่อเพิ่มความเร็รรอบในขั้นตอนการตั้งค่า
- ปุ่ม 3 จะใช้กดเพื่อลดความเร็รรอบในขั้นตอนการตั้งค่า
- ปุ่ม 4 จะใช้กดเพื่อเดินหน้าไปยังขั้นตอนต่อไป

3.2.5 แผนภาพและวงจรควบคุมเครื่องปั่นงาน

จากอุปกรณ์ข้างต้นที่กล่าวมาจะต้องนำมาประกอบการใช้งานเพื่อสร้างเครื่องถ่วงสมดุล ซึ่งแผนภาพของการเชื่อมต่ออุปกรณ์จะเป็นดังรูปที่ 3.12 และวงจรไฟฟ้าจะเป็นดังรูป 3.13



รูปที่ 3.13 แผนภาพของการเชื่อมต่ออุปกรณ์ของเครื่องปั่นงาน



รูปที่ 3.14 แผนภาพวงจรไฟฟ้าที่ออกแบบสำหรับเครื่องปั่นงาน

3.3 แนวทางและวิธีดำเนินงานวิจัย

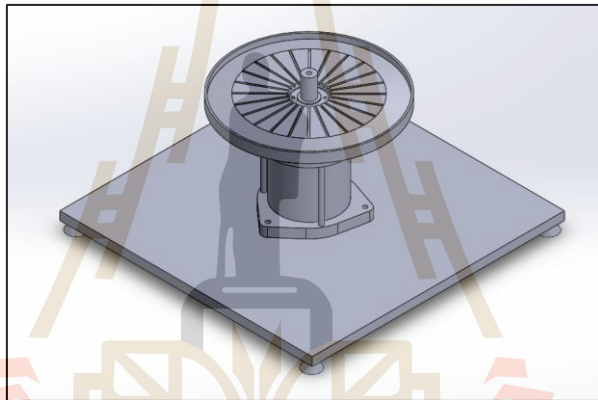
- สำรวจปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง
- ออกแบบและสร้างเครื่องป้อนงาน
- สร้าง Math Model ของระบบมอเตอร์ Brushless DC และตัวควบคุม
- ออกแบบสร้างเครื่องถ่วงสมดุลเพื่อทดสอบถ่วงสมดุลการหมุน
- ทดสอบ Closed-Down Speed เพื่อดู Frequency Response ของระบบ
- ทดสอบการทำงาน
- วิเคราะห์ข้อมูลและสรุปผลที่ได้
- จัดทำเอกสารและรายงานวิจัย

บทที่ 4

ผลการดำเนินงานวิจัย

4.1 ออกแบบและสร้างเครื่องปั่นงาน

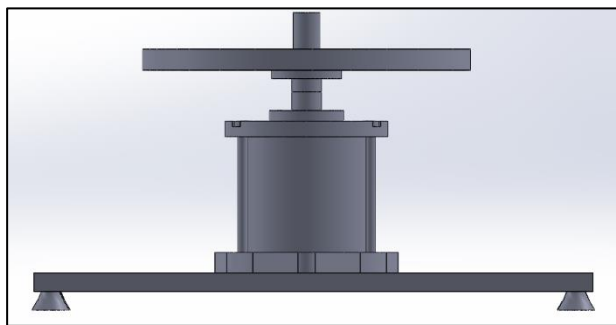
การออกแบบเครื่องปั่นงานนั้นจะใช้มอเตอร์ BLDC รุ่น F72-201105-220 ซึ่งจะถูกนำไปขึ้นรูปด้วยโปรแกรม SolidWorks จากนั้นทำการออกแบบโดยยึดฐาน โดยใช้วัสดุโลหะขนาด 300x300 ตารางมิลลิเมตร จึงทำให้ได้แบบการสร้างเครื่องปั่นงานดังรูปที่ 4.1-4.3



รูปที่ 4.1 แบบเครื่องปั่นงานมุมมองไอโซเมตริก (Isometric)

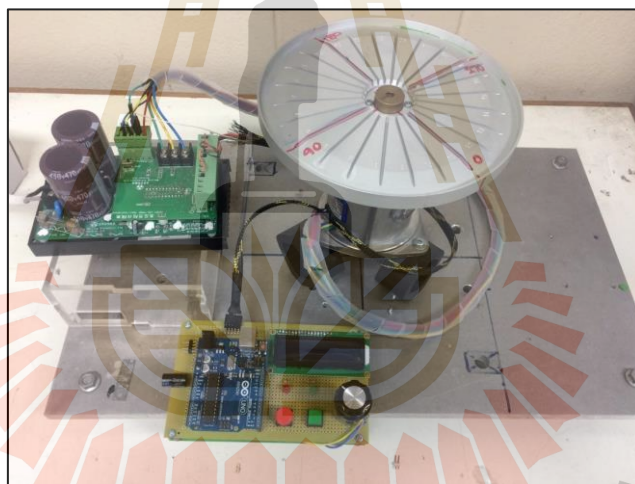


รูปที่ 4.2 แบบเครื่องปั่นงานมุมมองด้านบน (Top View)



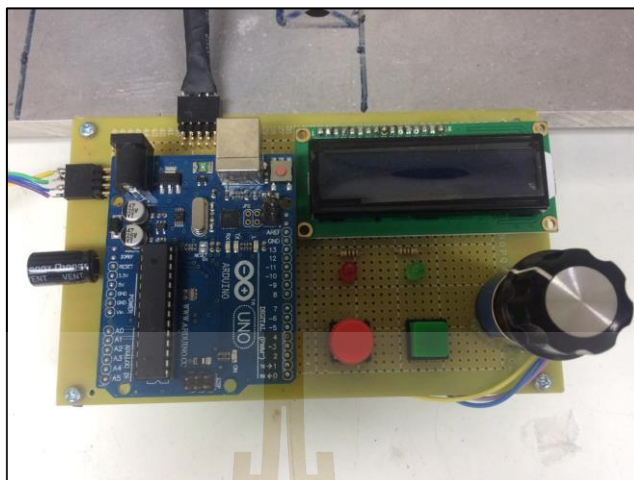
รูปที่ 4.3 แบบเครื่องปั่นงานมุมมองด้านหน้า (Front View)

จากแบบที่ได้ออกแบบมานั้นจะนำมาสร้างเครื่องจริง โดยการซื้ออุปกรณ์แต่ละชิ้นมาประกอบกันและเชื่อมต่อวงจรไฟฟ้าควบคุม เมื่อสร้างเครื่องสำเร็จจะมีลักษณะดังรูปที่ 4.4



รูปที่ 4.4 เครื่องปั่นงาน

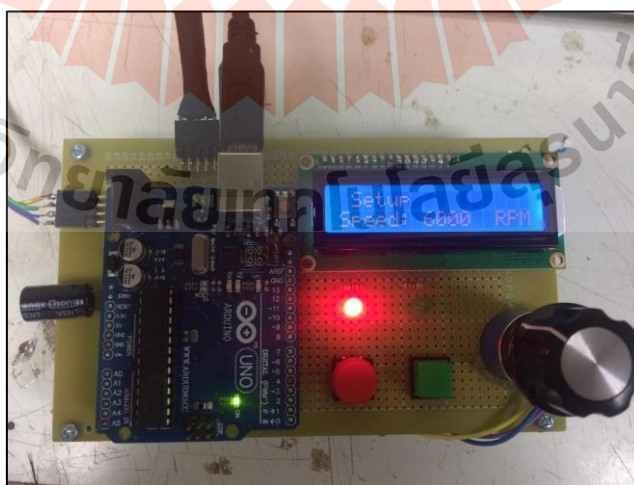
จากวงจรไฟฟ้าที่สร้างขึ้นจากแบบที่ได้ออกแบบไว้จะเป็นวงจรที่ใช้ในการควบคุมความเร็วรอบที่เราสามารถกำหนดเองได้จาก Potentiometer ซึ่งความเร็วรอบนี้จะอยู่ระหว่าง 1,000-12,000 รอบต่อนาที ตามข้อกำหนดของเครื่องปั่นเลือด โดยจะมี Switch สีเขียวเพื่อใช้ในการสั่งหมุนมอเตอร์ BLDC และมี Switch สีแดงเพื่อสั่งหยุดการทำงานของมอเตอร์ ซึ่งการทำงานนี้จะมีไฟ LED แสดงผลที่วงจร โดย LED สีแดงจะแสดงสถานะของมอเตอร์ว่าหยุดทำงาน และ LED สีเขียวจะแสดงสถานะของมอเตอร์ว่ากำลังทำงานอยู่ ซึ่งวงจรไฟฟ้าควบคุมที่สร้างขึ้นจะมีลักษณะดังรูปที่ 4.5



รูปที่ 4.5 วงจรควบคุมเครื่องปั่นงาน

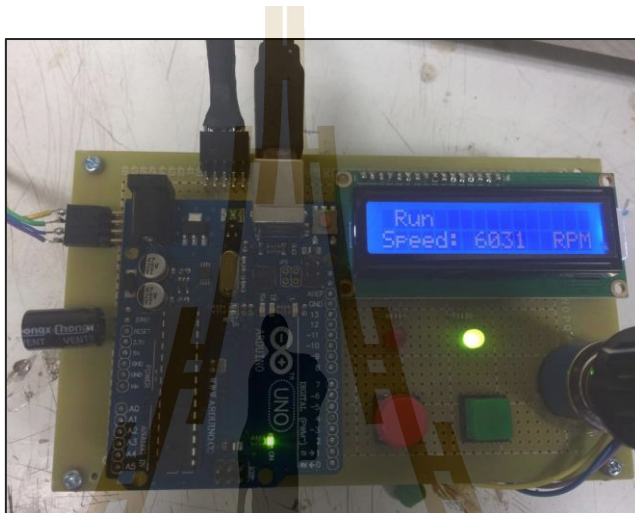
จากการสร้างเครื่องปั่นงานขึ้นมานั้นเพื่อที่จะปั่นงานเลือดในความเร็วรอบที่กำหนดเพื่อให้เครื่องถ่วงสมดุลวัดค่าการสั่นสะเทือนและเฟสของสภาวะการสั่นสะเทือนของงานเลือดได้ขณะที่ทำงาน โดยวิธีการใช้งานของเครื่องปั่นเลือดนั้นจะสามารถใช้งานได้ตามขั้นตอนดังนี้

1) ขั้นตอนแรกทำการต่อไฟเข้ากับระบบเครื่องปั่นงานจะมีไฟสถานะ LED สีแดงปรากฏขึ้น ซึ่งหมายความว่ามอเตอร์หยุดการทำงานหน้าจอ LCD จะแสดงสถานะ Setup ในบรรทัดที่ 1 และจะแสดงผลค่าความเร็วที่ต้องการ Setup ในช่วง 1,000-12,000 รอบต่อนาที ซึ่งจากรูปที่ 4.6 จะ Setup ความเร็วรอบไว้ที่ 6,000 รอบต่อนาที ตามการทดลองที่ตั้งไว้



รูปที่ 4.6 สถานะติดตั้ง (Setup) ของเครื่องปั่นงาน

2) ขั้นตอนต่อมาทำการกด Switch สีเขียวเพื่อทำให้ Microcontroller สั่งให้มอเตอร์ BLDC ทำงานผ่าน Driver เมื่อกดปุ่มแล้วไฟสถานะ LED จะเปลี่ยนจากสีแดงเป็นสีเขียวพร้อมกับมอเตอร์ BLDC เริ่มหมุนจาก 0 รอบต่อนาทีไปจนถึงค่าความเร็วรอบที่ Setup ไว้ จอ LCD เปลี่ยนสถานะจาก Setup เป็น Run ในบรรทัดที่ 1 และจะแสดงความเร็วรอบของมอเตอร์ BLDC ที่กำลังหมุนใน บรรทัดที่ 2 ดังรูปที่ 4.7

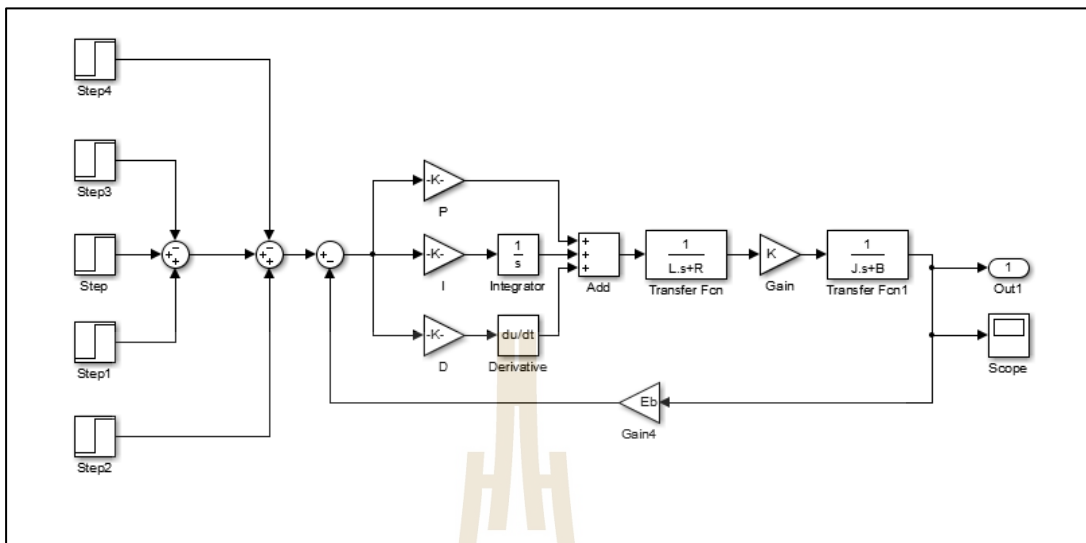


รูปที่ 4.7 สถานการณ์ทำงาน (Run) ของเครื่องปั่นงาน

3) ขั้นตอนสุดท้ายเมื่อต้องการหยุดมอเตอร์ BLDC แล้ว Setup หรือรอการ Run รอบต่อไป ให้ทำการกดปุ่มสีแดง จะทำให้ Driver หยุดจ่ายไฟฟ้าให้กับมอเตอร์ ทำให้มอเตอร์หยุดลง LED แสดงสถานะเปลี่ยนจากเขียวเป็นแดง หน้าจอ LCD จะแสดงสถานะ Setup ในบรรทัดที่ 1 และจะแสดงผลค่าความเร็วที่ต้องการ Setup ในบรรทัดที่ 2 ซึ่งสามารถรันครั้งต่อไปได้โดยการ กดปุ่มสีเขียวได้ทันที

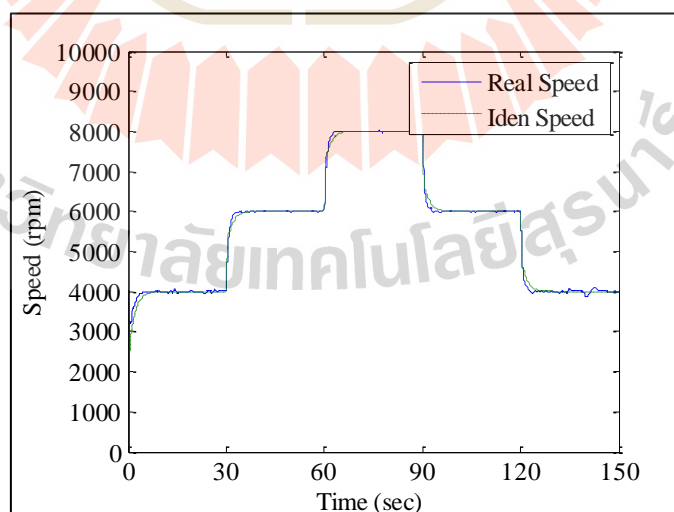
4.2 สร้าง Math Model ของระบบมอเตอร์ Brushless DC และตัวควบคุม

การสร้างแบบจำลองทางคณิตศาสตร์ของมอเตอร์ BLDC ในการทดลองนี้จะทำการประมาณค่าในระบบที่เป็น Closed loop Control โดยใช้ฟังก์ชัน Parameter Estimation โดยจำเป็นต้องเก็บข้อมูลของความเร็วรอบที่ได้ต่อความเร็วรอบที่ตั้งค่าเอาไว้ เพื่อให้ MATLAB คำนวณและประมาณค่าพารามิเตอร์แต่ละตัวออกมาซึ่งจะสามารถเขียน Model การหา Parameter ใน MATLAB/Simulink ได้ดังภาพที่ 4.8 ซึ่งเมื่อทำการประมาณค่าและคำนวณหาตัวควบคุมแล้วจะได้พารามิเตอร์ต่าง ๆ ดังตารางที่ 4.1



รูปที่ 4.8 Model Closed-Loop ที่จะใช้ประมาณค่าพารามิเตอร์

การประมาณค่าจะเป็นการประมาณโดยใช้ Closed Loop Identification ปรับความเร็วรอบในช่วงที่ต้องการใช้งานคือ 4,000-8,000 รอบต่อนาที ซึ่งจะตั้งตัวแปรเริ่มต้นของพารามิเตอร์ให้มีค่าเท่ากับ 1 จากนั้นเริ่มต้นการประมาณค่าโดยตัวโปรแกรมจะเริ่มประมาณจากการเทียบผลที่ได้จากพารามิเตอร์ก่อนหน้ากับค่าความเร็วรอบจริงและจะค่อย ๆ ปรับพารามิเตอร์ต่าง ๆ ที่ทำให้ค่าผลของความเร็วยุทธที่ได้ใกล้เคียงกับความเร็วยุทธจริงที่เก็บข้อมูลมาดังรูปที่ 4.9



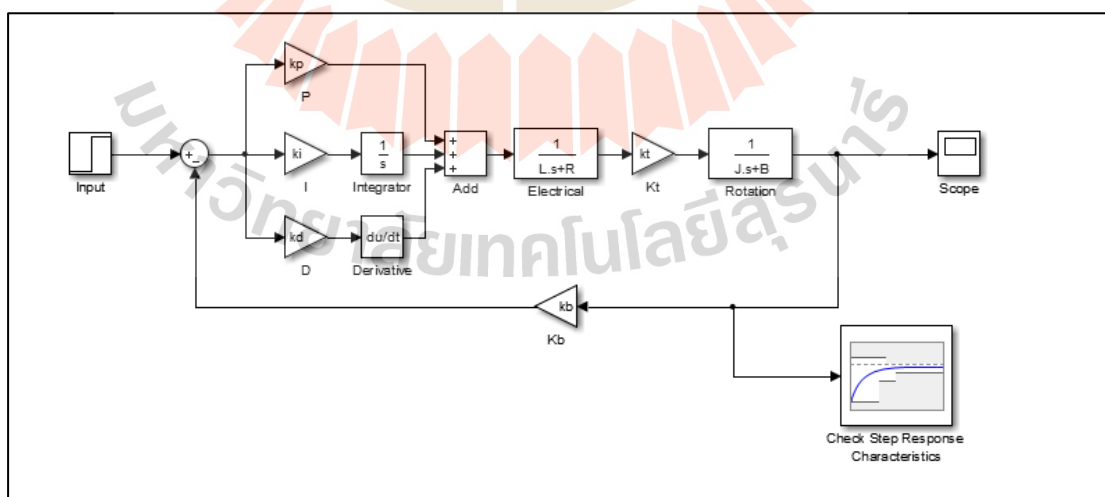
รูปที่ 4.9 การเปรียบเทียบความเร็วยุทธจากการประมาณค่ากับความเร็วยุทธจริง

เมื่อทำการประมาณค่าจากโปรแกรมแล้วจะทำให้ได้ค่าพารามิเตอร์ของมอเตอร์ BLDC ที่สามารถนำไปจำลองผลตอบสนองได้ดังตารางที่ 4.1

ตารางที่ 4.1 พารามิเตอร์ของ BLDC มอเตอร์

Name of Parameter	Parameter	Value	Unit
Moment of Inertia	J	4.37	Kg.m^2
Friction Coefficient	B	4.88	Nm.s/rad
Resistance	R	28.94	Ω/phase
Inductance	L	27.17	H/phase
Back-EMF	K_b	0.1×10^{-3}	V/1000 rpm
Torque Constant	K_t	0.912	kNm/A

เมื่อได้พารามิเตอร์ของมอเตอร์ BLDC แล้วก็จะทำการ Optimization ตัว PID Controller เพื่อที่จะหาค่าที่เหมาะสมที่สุดที่จะใช้ในการควบคุมมอเตอร์ BLDC ที่ความเร็วรอบในการใช้ถ่วงสมดุล ซึ่งจะใช้ Block Check Step Response เป็นตัวช่วยในการ Optimization PID Controller ดังรูปที่ 4.10



รูปที่ 4.10 โมเดลการ Optimization PID Controller

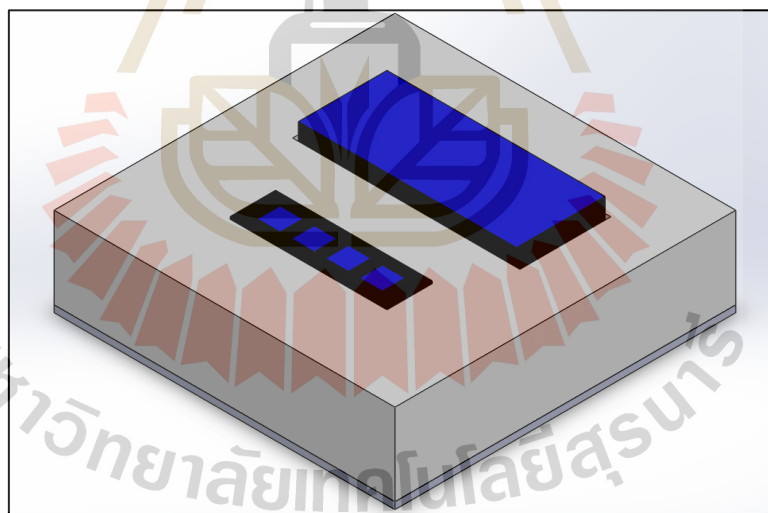
เมื่อทำการ Optimization PID Controller แล้วจะทำให้ได้ค่า Controller ต่างดังตารางที่ 4.2

ตารางที่ 4.2 พารามิเตอร์ของระบบควบคุม PID

Name of Parameter	Parameter	Value	Unit
Proportional Gain	K_p	0.0100	-
Integral Gain	K_i	0.0045	sec^{-1}
Derivative Gain	K_d	0.0010	sec

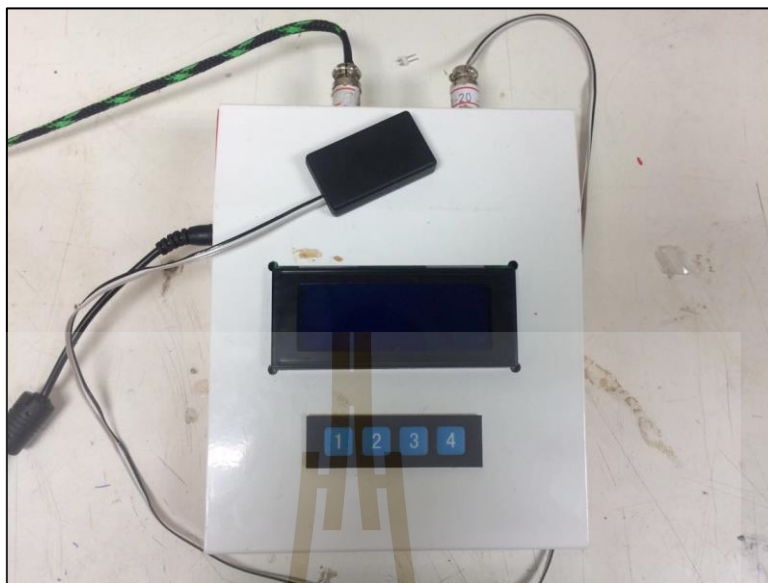
4.3 ออกแบบสร้างเครื่องถ่วงสมดุลเพื่อทดสอบถ่วงสมดุลการหมุน

การออกแบบเครื่องถ่วงสมดุลนั้นจะต้องออกแบบทั้งวงจรไฟฟ้าและออกแบบตัวเครื่อง ซึ่งตัวเครื่องนั้นจะใช้วัสดุที่ทำจากโลหะขนาด 150x150 ตารางมิลลิเมตร สูง 40 มิลลิเมตร โดยภายในเป็นพื้นที่สำหรับใส่วงจรของเครื่องนี้จากการออกแบบจะได้แบบ ดังรูปที่ 4.11



รูปที่ 4.11 แบบเครื่องถ่วงสมดุล

จากแบบที่ออกแบบไว้เมื่อนำมาสร้างจริงจะมีลักษณะดังรูปที่ 4.12



รูปที่ 4.12 เครื่องถ่วงสมดุล

จากเครื่องถ่วงสมดุลที่สร้างขึ้นนั้นจะเป็นเครื่องถ่วงสมดุลที่ใช้คู่กับเครื่องปั่นงาน โดยจะมีช่องเสียบรับ Accelerometer 1 ตัว ใช้ติดตั้งสำหรับวัด Vibration ที่เครื่องปั่นงาน โดยเมื่อได้ค่าการสั่นสะเทือนตัว Accelerometer ก็จะส่งไปยัง Microcontroller เพื่อที่จะทำการประมวลผลต่อไป และจะมีช่องเสียบรับ Hall Effect Sensor อีก 1 ตัวเพื่อที่จะตรวจจับความเร็วรอบจากมอเตอร์ BLDC เมื่อได้สัญญาณ 2 ชนิดเข้ามาที่เครื่องแล้ว เครื่องก็จะประมวลผลกับ 2 สัญญาณและหามุมองศาการสั่นสะเทือนสูงที่สุดที่วัดได้ก่อนที่จะเข้าสู่กระบวนการถ่วงสมดุลต่อไป โดยขั้นตอนการถ่วงสมดุลนั้นจะมีขั้นตอนดังนี้

- 1) ติดตั้ง Accelerometer ไปยังมอเตอร์ BLDC ดังรูปที่ 4.11
- 2) เปิด Switch ข้างเครื่องถ่วงสมดุล หน้าจอ LCD จะแสดง “Welcome to HC Balance” ซึ่งจะต้องรอกวนว่าจะแสดงผล Setup Speed ดังรูปที่ 4.12
- 3) ทำการ Setup Speed โดย LCD ที่แสดงผลด้านล่างสุดจะกำกับการทำงานของปุ่ม โดยจากรูปนี้ เมื่อกดปุ่มเลข 2 จะทำให้ Speed ที่ Setup เพิ่มขึ้นทีละ 100 รอบต่อนาที เมื่อกดปุ่มเลข 3 จะทำให้ Speed ที่ Setup ลดลงทีละ 100 รอบต่อนาที และเมื่อกดปุ่มเลข 4 จะทำให้เก็บค่า Setup Speed และเข้าสู่ขั้นตอนการถ่วงสมดุลต่อไป โดยขั้นตอนนี้ผู้ใช้งานจะต้องทำการ Setup Speed ตามที่ต้องการจะถ่วงสมดุลซึ่งในการทดลองนี้จะ Setup Speed ไว้ที่ 6,000 รอบต่อนาที แล้วกดปุ่ม 4 เพื่อเก็บค่าและดำเนินการขั้นตอนถัดไป



รูปที่ 4.13 ติดตั้ง Accelerometer ที่มอเตอร์ BLDC



รูปที่ 4.14 เครื่องถ่วงสมดุลแสดงผล Setup Speed

- 4) เมื่อมาถึงขั้นตอน Run Machine ดังรูปที่ 4.15 ให้ทำการหมุนมอเตอร์ BLDC ที่เครื่องปั่นจานด้วยความเร็วรอบตรงกับที่เครื่องถ่วงสมดุลกำหนด แล้วทำการกดปุ่ม 4 เพื่อดำเนินขั้นตอนถัดไป



รูปที่ 4.15 เครื่องถ่วงสมดุลแสดงผล Run Machine

- 5) เมื่อถึงขั้นตอนนี้ตัวเครื่องจะทำการวัดค่า Amplitude กับความเร็วรอบจากมอเตอร์เพื่อมาคำนวณหาเฟสการสั่นสะเทือนที่สูงที่สุดที่เกิดขึ้นขณะทำงาน ขั้นตอนนี้ผู้ใช้งานจะต้องรอนกว่า Speed เข้าใกล้กับความเร็วรอบที่ตั้งค่าเอาไว้ แล้วจึงกดปุ่ม 4 เพื่อดำเนินการ Store ค่า Amplitude และ Phase ที่วัดได้ดังรูปที่ 4.16 เมื่อเครื่องทำการ Store ค่าให้เราเครื่องจะแสดงค่าที่ Store ค้างไว้ดังรูปที่ 4.17 ซึ่งค่า Amplitude ที่วัดได้จะเป็นค่าการสั่นสะเทือนที่ส่งผลมาจากความไม่สมดุลบนงานเลื่อย และ ค่า Phase ที่วัดได้จะเป็นค่าเฟสองศาที่ทำให้เกิดการสั่นสะเทือนของงานเลื่อย



รูปที่ 4.16 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้

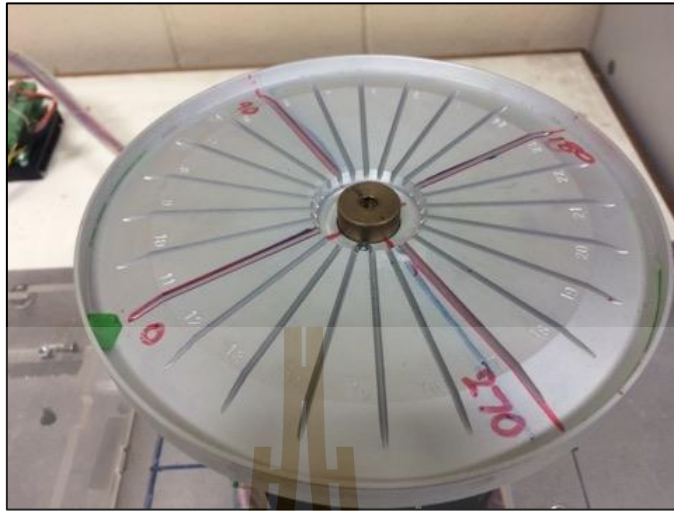


รูปที่ 4.17 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่เก็บค่า

- 6) เมื่อทำการ Store ค่า Amplitude และ Phase ที่เกิดจากความไม่สมดุลแล้วจะต้องทำการ Trial Weight โดยเครื่องถ่วงสมดุลก็จะให้ทำการติดมวลในงานเลือดที่น้ำหนัก 0.5 กรัมที่มุม 0 องศาบนใบจานหมุนดังรูปที่ 4.18 ให้ทำการหยุดมอเตอร์ BLDC แล้วติดมวลขนาด 0.5 กรัมที่มุม 0 องศาบนใบจานเลือด ดังรูปที่ 4.19 จากนั้นทำการกดปุ่ม 4 เพื่อดำเนินการขั้นตอนถัดไป

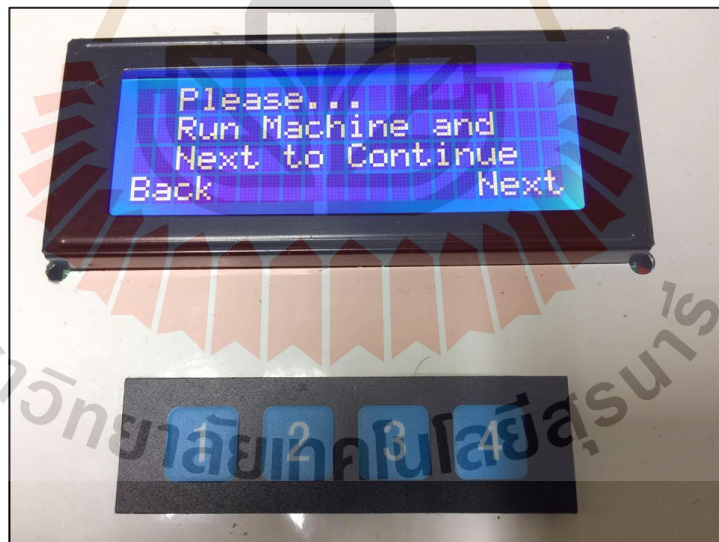


รูปที่ 4.18 เครื่องถ่วงสมดุลแสดงค่า Trial Weight ที่ต้องติดบนงานเลือด



รูปที่ 4.19 ตีคมวล 0.5 กรัมที่มุม 0 องศาบนใบจาน

- 7) เมื่อมาถึงหน้า Run Machine ดังรูปที่ 4.20 ต่อมาทำการ Run มอเตอร์ BLDC และกดปุ่ม 4 เพื่อดำเนินการวัดในขั้นตอนถัดไป



รูปที่ 4.20 เครื่องถ่วงสมดุลแสดงผล Run Machine

- 8) เมื่อถึงขั้นตอนนี้ตัวเครื่องจะทำการวัดค่า Amplitude กับความเร็วรอบจากมอเตอร์ เพื่อมาคำนวณหาเฟสการสั่นสะเทือนที่สูงที่สุดที่เกิดขึ้นขณะทำงาน ขั้นตอนนี้ผู้ใช้งาน จะต้องรอนกว่า Speed เข้าใกล้กับความเร็วรอบที่ตั้งค่าเอาไว้ แล้วจึงกดปุ่ม 4 เพื่อดำเนินการ Store ค่า Amplitude และ Phase ที่วัดได้ดังรูปที่ 4.21 เมื่อเครื่องทำการ Store ค่าให้เราเครื่องจะแสดงค่าที่ Store ค้างไว้ ดังรูปที่ 4.22 ซึ่งค่า Amplitude ที่วัดได้ จะเป็นค่าการสั่นสะเทือนจากความไม่สมดุลของงานเลื่อยผสมกับมวลที่ทำการติด Trial และ ค่า Phase ที่วัดได้จะเป็นค่าเฟสองศาที่ทำให้เกิดการสั่นสะเทือนของงาน เลื่อยผสมกับมวลที่ทำการติด Trial



รูปที่ 4.21 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้



รูปที่ 4.22 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่เก็บค่า

- 9) เมื่อทำการ Store ค่า Amplitude และ Phase ที่เกิดจากความไม่สมดุลร่วมกับมวล Trial แล้วเครื่องถ่วงสมดุลจะคำนวณและแสดงขนาดมวลที่จะต้องติดแก้ไขและมุมที่ต้องติดมวลแก้ไขดังรูปที่ 4.23 ให้ทำการหยุดมอเตอร์ BLDC และถอดมวล Trial ออก พร้อมทั้งติดตั้งมวลแก้ไขไปที่ตำแหน่งตามที่เครื่องถ่วงสมดุลคำนวณให้ ดังรูปที่ 4.24



รูปที่ 4.23 ขนาดมวลแก้ไขและมุมที่ต้องติดมวลแก้ไข



รูปที่ 4.24 ทำการติดตั้งมวลตามที่เครื่องถ่วงสมดุลคำนวณ

- 10) เมื่อติคมวลแก้ไขไปยังงานเลือดแล้วมาถึงหน้า Run Machine ดังรูปที่ 4.25 ต่อมาทำการ Run มอเตอร์ BLDC และกดปุ่ม 4 เพื่อดำเนินการวัดในขั้นตอนถัดไป



รูปที่ 4.25 เครื่องถ่วงสมดุลแสดงผล Run Machine

- 11) เมื่อถึงขั้นตอนนี้ตัวเครื่องจะทำการวัดค่า Amplitude กับความเร็วรอบจากมอเตอร์ เพื่อมาคำนวณหาเฟสการสั่นสะเทือนที่สูงที่สุดที่เกิดขึ้นขณะทำงาน ขั้นตอนนี้ผู้ใช้งานจะต้องรอกว่า Speed เข้าใกล้กับความเร็วรอบที่ตั้งค่าเอาไว้ แล้วจึงกดปุ่ม 4 เพื่อดำเนินการ Store ค่า Amplitude และ Phase ที่วัดได้ ดังรูปที่ 4.26 จากนั้นเครื่องจะคำนวณว่าในขั้นตอนการแก้ไขความไม่สมดุลที่ผ่านมาสามารถลดการสั่นสะเทือนของงานได้ขนาดไหน ดังรูปที่ 4.27 ซึ่งจะจบกระบวนการถ่วงสมดุลของงานเลือด



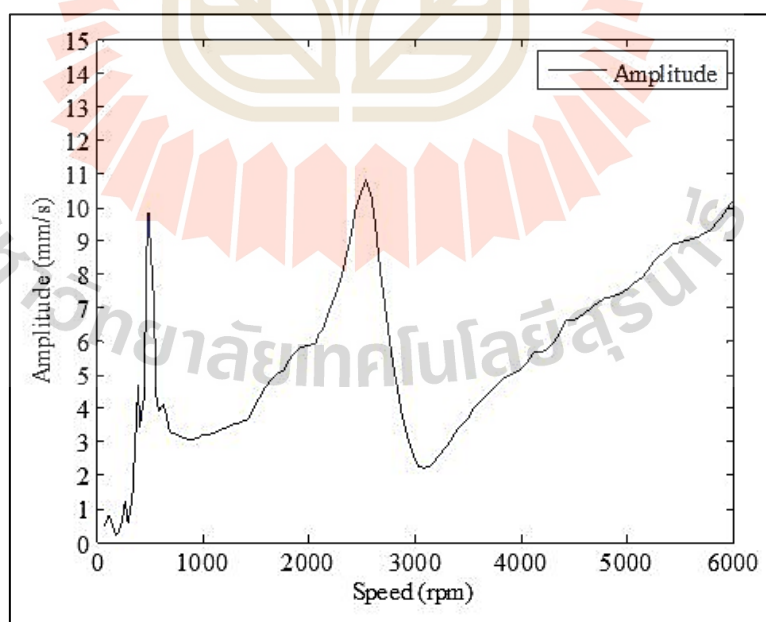
รูปที่ 4.26 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้



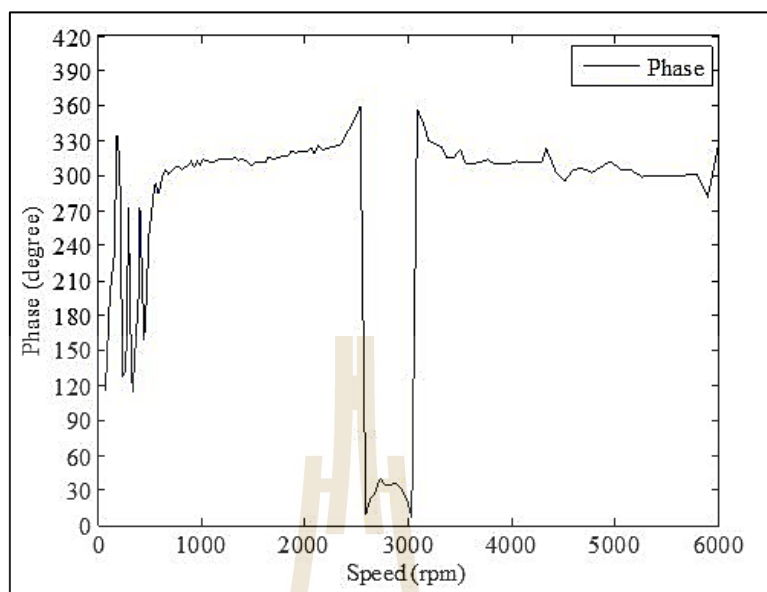
รูปที่ 4.27 ค่าการสั่นสะเทือนที่ลดลงหลังจากการแก้ไขความไม่สมดุลของจานเลือด

4.4 ทดสอบ Closed-Down Speed เพื่อดู Frequency

เมื่อสร้างเครื่องถ่วงสมดุลเสร็จก็จะต้องทำการทดสอบระบบโดยการทำ Closed-Down Speed เพื่อที่จะตรวจสอบ Frequency Response ในระบบเครื่องปั่นงาน ซึ่งจะต้องเลือก Speed ให้เหมาะสมเพื่อที่จะถ่วงสมดุลในการทดลอง ซึ่งจากผลของการทำ Close-Down Speed ได้ผล ดังรูปที่ 4.28 ในรูปของ Amplitude การสั่นสะเทือน และดังรูปที่ 4.29 ในรูปของมุมเฟส



รูปที่ 4.28 Frequency Response ของระบบเครื่องปั่นงานในรูปของ Amplitude การสั่นสะเทือน



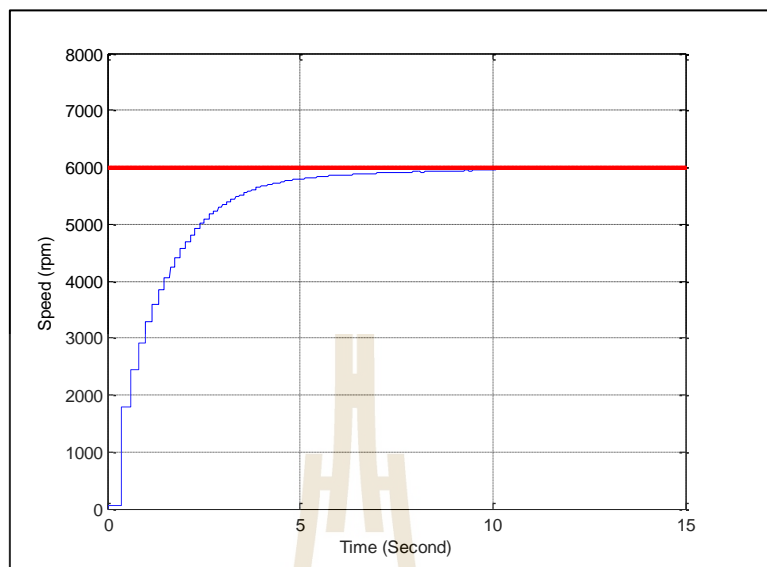
รูปที่ 4.29 Frequency Response ของระบบเครื่องปั่นงานในรูปของมุม Phase

จากรูปที่ 4.28 จะพบ Resonance ที่ทำให้เกิดการสั่นสูงที่สุด 2 ช่วง คือช่วง 600-1,000 rpm และ 1500-3,200 rpm เมื่อพิจารณาจากรูปที่ 4.29 แล้ว Resonance จะเกิดการกลับเฟสกะทันหันในช่วงดังกล่าวจึงทำให้ไม่สามารถที่จะถ่วงสมดุลได้ในช่วงนี้ การทดลองนี้จะต้องถ่วงสมดุลที่ความเร็วรอบ 3,200 rpm ขึ้นไป ซึ่งผู้ทดลองได้ทำการเลือกความเร็วรอบที่ 6,000 rpm เพื่อใช้ในการทดลองถ่วงสมดุลนี้

4.5 ผลการทดลอง

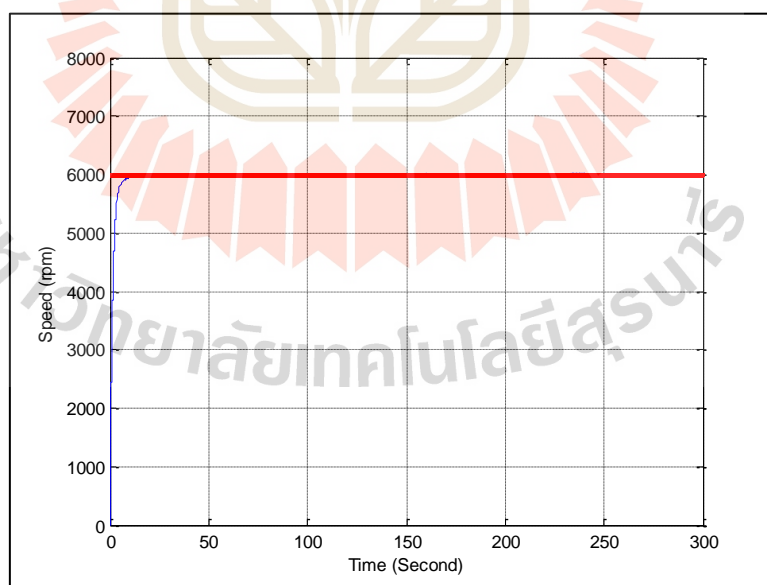
4.5.1 เครื่องปั่นงาน

จากการออกแบบ Controller ในโปรแกรม Matlab แล้วนำมาใช้จริงกับมอเตอร์ BLDC โดย Setup ความเร็วรอบตามการทดลองที่ 6,000 รอบต่อนาที จากนั้นทำการ Run มอเตอร์ BLDC แล้วจะเห็นได้ว่าผลตอบสนอง Rise Time เมื่อมอเตอร์เริ่มหมุนจากหยุดนิ่งไปจนถึงความเร็วรอบที่ 6,000 รอบต่อนาทีนั้น Controller จะควบคุมความเร็วรอบไม่ให้เกิด Overshoot ในการหมุนของมอเตอร์ BLDC ซึ่งจะใช้เวลาเข้าสู่ Steady State 3 วินาที โดยมี %error ของความเร็วรอบอยู่ที่ 0.25% ดังรูปที่ 4.30



รูปที่ 4.30 กราฟแสดงความเร็วรอบของมอเตอร์ BLDC

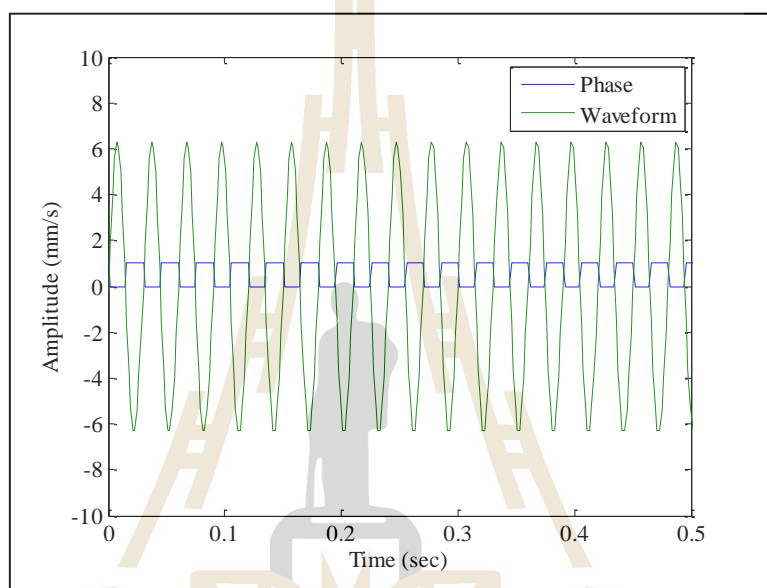
การใช้งานจริงของเครื่องปั้นเลื้อคนั้นจะต้องทำการปั้นเลื้อคโดยใช้เวลา 5 นาที ซึ่งเมื่อนำ Controller ตัวนี้มาใช้กับมอเตอร์ BLDC โดยการ Run 5 นาทีตามเงื่อนไข Controller ก็ยังสามารถควบคุมความเร็วรอบที่ 6,000 รอบต่อนาที ตลอดการทำงานของกระบวนการ ดังรูปที่ 4.31



รูปที่ 4.31 กราฟแสดงความเร็วรอบของมอเตอร์ BLDC ตลอดการทำงาน 5 นาที

4.5.2 เครื่องถ่วงสมดุล

การทดลองของเครื่องถ่วงสมดุลนั้นจะทำการวัดการสั่นสะเทือนเนื่องจากความไม่สมดุลของจานเลือด และวัดความเร็วรอบของมอเตอร์เพื่อที่จะนำสัญญาณไปเทียบกับการสั่นสะเทือนในการหามุมเฟสที่มีการสั่นสะเทือนที่สูงที่สุดซึ่งเมื่อเทียบสัญญาณระหว่างการสั่นสะเทือนกับความเร็วรอบเพื่อหามุมเฟส ดังรูปที่ 4.32



รูปที่ 4.32 กราฟสัญญาณการสั่นสะเทือนเปรียบเทียบกับสัญญาณความเร็วรอบ

ตารางที่ 4.3 เงื่อนไขการ Trial Weight ของแต่ละงาน

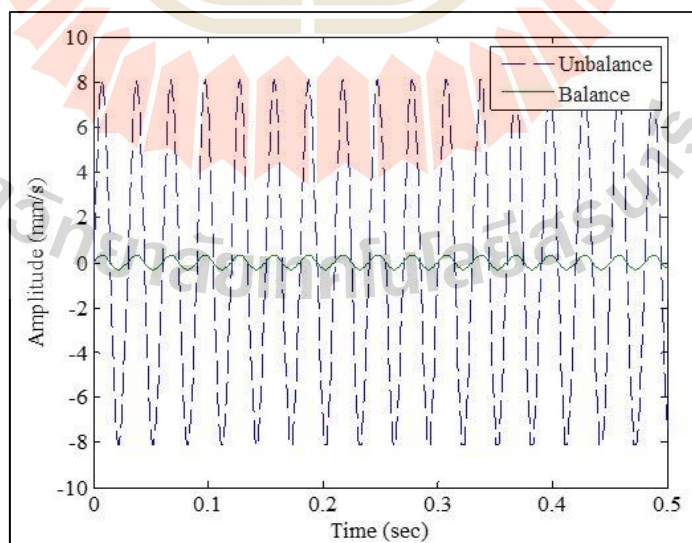
	งานเลือดใบที่ 1		งานเลือดใบที่ 2	
	M_T (grams)	θ_T (degree)	M_T (grams)	θ_T (degree)
กรณีที่ 1	0.3	0	0.4	0
กรณีที่ 2	0.5	180	0.6	270
กรณีที่ 3	1.0	0	0.9	0

งานเล็อดแต่ละงานจะทำการถ่วงสมดุลเพื่อลดการสั่นสะเทือนลง โดยเปลี่ยนเงื่อนไขของการ Trial Weight ซึ่งเมื่อทำการถ่วงสมดุลแล้วจะได้ผลดังตารางที่ 4.4

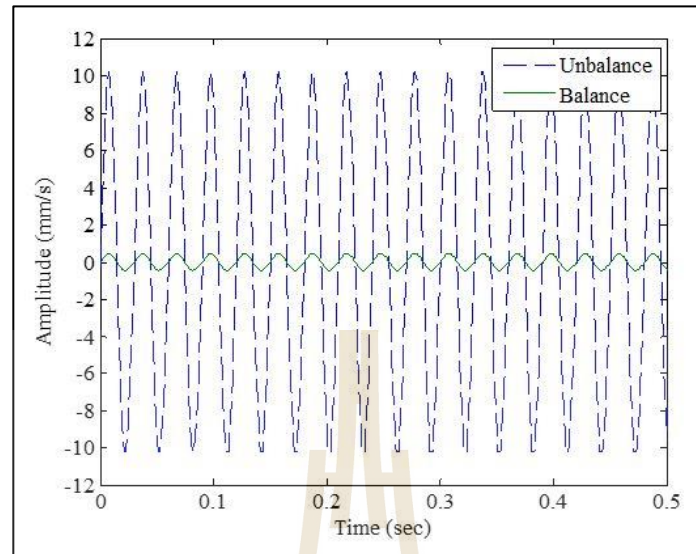
ตารางที่ 4.4 ผลจากการถ่วงสมดุลในแต่ละเงื่อนไขของการ Trial Weight

	A_0 (mm/s)	ϕ_0 ($^{\circ}$)	Trial Weight ในแต่ละกรณี						แก้ไข		ผลจากการถ่วงสมดุล	
			A_1 (mm/s)			ϕ_1 ($^{\circ}$)			M_B (g)	θ_B ($^{\circ}$)	A_B (mm/s)	% สั่น ลดลง
			I	II	III	I	II	III				
งาน 1	8.15	328	7.19	11.19	8.44	347	310	34	0.9	60	0.33	95.95
งาน 2	10.19	354	11.08	17.45	14.33	21	358	356	0.8	86	0.41	95.98

จากการถ่วงสมดุลของงานหมุน 2 ใบในแต่ละกรณีการ Trial Weight นั้นเมื่อทำการถ่วงสมดุลไม่ว่าจะ Trial Weight กรณีใด การคำนวณของเครื่องถ่วงสมดุลสุดท้ายจะส่งผลให้สามารถบอกขนาดของมวลและมุมของการติดมวลแก้ไขความไม่สมดุลบนงานที่จุดเดียวกัน โดยผลจากการติดมวลแก้ไขความไม่สมดุลของงานเล็อดใบที่ 1 จะลดการสั่นสะเทือนได้ 95.95% ดังรูปที่ 4.33 และงานเล็อดใบที่ 2 จะลดการสั่นสะเทือนได้ 95.98% ดังรูปที่ 4.34



รูปที่ 4.33 สัญญาณการสั่นสะเทือนที่สมดุลกับไม่สมดุลของงานใบที่ 1



รูปที่ 4.34 สัญญาณการสั่นสะเทือนที่สมดุลกับไม่สมดุลของงานใบที่ 2

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้เป็นงานวิจัยเพื่อสร้างเครื่องถ่วงสมดุลแผ่นปั่นเลือดที่เหวี่ยงหมุนในแนวตั้ง ซึ่งจะแสดงถึงกรรมวิธีในการออกแบบ และสร้างเครื่องถ่วงสมดุลจริงในการใช้ถ่วงสมดุลแผ่นปั่นเลือด เพื่อลดการสั่นสะเทือนของเครื่องปั่นเลือด โดยจะประกอบไปด้วยอุปกรณ์ 2 ชั้น ได้แก่

1. เครื่องปั่นงาน

เครื่องปั่นงานที่ได้ถูกออกแบบและสร้างขึ้นใช้ทดลองในการปั่นงานเลือดจริง เพื่อกระตุ้นการสั่นสะเทือนของงานเลือด โดยจะใช้มอเตอร์ BLDC เป็นอุปกรณ์ในการหมุนเหวี่ยง ซึ่งควบคุมโดย Driver ผ่าน Microcontroller ซึ่งจากการที่ได้สร้างเครื่องขึ้นมาใช้งานจริง โดยทดลองปั่นที่ 6,000 รอบต่อนาที ระบบควบคุมที่ออกแบบมาสามารถควบคุมความเร็วรอบเครื่องปั่นงานได้อย่างมีประสิทธิภาพ ซึ่งระบบไม่มี Overshoot ใช้เวลาเข้าสู่ Steady State 3 วินาที และมีค่าความคลาดเคลื่อนสูงสุดที่วัดได้ตลอดการทำงาน 0.25%

2. เครื่องถ่วงสมดุล

เครื่องถ่วงสมดุลที่ได้ถูกออกแบบและสร้างขึ้นใช้ในการวัดการสั่นสะเทือนของงานเลือดที่เครื่องปั่นงาน โดยใช้สัญญาณการสั่นสะเทือนจาก Accelerometer และสัญญาณความเร็วรอบจาก Hall Effect Sensor เพื่อวัด Amplitude กับ Phase การสั่นสะเทือน จากการทดลอง โดยใช้เครื่องถ่วงสมดุลนี้แก้ไขความไม่สมดุลในงานเลือดจำนวน 2 ใบงาน ที่เงื่อนไขการคิด Trial Weight ที่ต่างกัน สรุปได้ว่าที่กรณีการถ่วง Trial Weight ต่าง ๆ ของใบงานแต่ละใบ จะสามารถบอกขนาดของมวลและตำแหน่งการติดตั้งมวลได้ตรงกัน ซึ่งเมื่อทำการติดมวลตามเครื่องถ่วงสมดุลได้ คำนวณและบอกไว้แล้วสามารถลดการสั่นสะเทือนของงานเลือดซึ่งการสั่นสะเทือนของงานเลือดลดลงจากเดิม 91.45 %

5.2 ข้อเสนอแนะ

สำหรับข้อเสนอแนะในการใช้งานเครื่องถ่วงสมดุล เพื่อให้สามารถถ่วงสมดุลของงานเลือดได้อย่างแม่นยำ และไม่เกิดอันตรายต่อผู้ใช้งานจะมีข้อเสนอแนะดังนี้

1. ควรตรวจเช็คการเชื่อมต่อของสายไฟระหว่าง Arduino กับ Driver ให้ละเอียดว่ามี การเชื่อมต่อกันหรือไม่ เนื่องจากว่า Driver จะสั่งมอเตอร์ทำงานก็ต่อเมื่อ Active Low เมื่อไม่ได้ เชื่อมต่อกันระหว่าง Arduino กับ Driver จะเปรียบเหมือนเป็น Active Low มอเตอร์จะปรับความเร็ว รอบไปที่สูงที่สุดซึ่งอาจทำให้เกิดอันตรายต่อผู้ใช้งานได้
2. เมื่อใช้งานเครื่องถ่วงสมดุลโดยรับสัญญาณจาก Driver ที่มาจากเครื่องปั่นงาน ควรเปิด เครื่องถ่วงสมดุลก่อนที่จะ Run มอเตอร์ BLDC เพื่อไม่ให้เกิดการรบกวนการควบคุมความเร็วรอบ เนื่องจากการเดินไฟไม่สมบูรณ์
3. ในระหว่างการถ่วงสมดุลงานเลือดควรเช็คการติดมวล์ให้ละเอียด เมื่อความเร็วรอบของ การหมุนสูงขึ้นแรงหนีศูนย์กลางจะสามารถทำให้มวล์ที่ติดตั้งอยู่หลุดออกจากงานเลือดและจะเกิด อันตรายต่อผู้ใช้งานได้

รายการอ้างอิง

- ชูชาติ อารีจิตรานุสรณ์ (2544). เครื่องมือวิทยาศาสตร์, ขอนแก่น:คลังน่านาวิทยา.
- Manjita Srivastava, M C Srivastava and Smriti Bhatnagar (2009), “Control System”, New Delhi:Tata McGraw-Hill.
- G Prasad, Venkateswara Reddy M, P V N Prasad and G Tulasi Ram Das (2012). “ Speed control of brushless DC motor with DSP controller using Matlab”, International Journal of Engineering Research and Applications, vol. 2, pp.2120-2125.
- Pawin Jawayon, Jiraphon Srisertpol (2013), “Speed Estimation Of 3-Phase BLDC Motor Using Genetic Algorith”, International Journal of Engineering Science and Innovative Technology, vol. 2.
- Ms.Manjusha Patil (2014), “Modelling and simulation of dc drive using PI and PID controller”, International Journal of Innovative Research in Electrical, Electronic, vol. 2.
- Viral K. Patel, Maitri N. Patel (2017), “Development of Smart Sensing Unit for Vibration Measurement by Embedding Accelerometer with the Arduino Microcontroller”, International Journal of Instrumentation Science.
- Dennis H. Shreve (1995), “Signal Processing for Effective Vibration Analysis”, IRD Mechanalysis, Inc Columbus, Ohio.
- Md. Abdul Saleem, G. Diwakar, Dr. M.R.S. Satyanarayana (2012), “Detection of Unbalance in Rotating Machines Using Shaft Deflection Measurement during Its Operation”, Journal of Mechanical and Civil Engineering, vol. 3, pp.08-20.
- Guilherme Kenji Yamamoto, Cesarda Costa, João Sinoharada Silva Sousa. (2016) “A smart experimental setup for vibration measurement and imbalance fault detection in rotating machinery”, Control and Automation Engineering, vol. 4, pp.8-18.
- Chigbogu C. Ozoegwu, Christian. C. Nwangwu, Chigozie F. Uzoh, Arinze V. Ogunoh, (2012) “Pure Analytical Approach to Rotational Balancing”, Journal of Safety Engineering 2012, pp.50-56.



ภาคผนวก ก

คู่มือการใช้งานเครื่องถ่วงสมดุล

1.1 อุปกรณ์ชุดถ่วงสมดุล



รูปที่ ก.1 ชุดอุปกรณ์ถ่วงสมดุลงานปั้นเหรียญเลือด

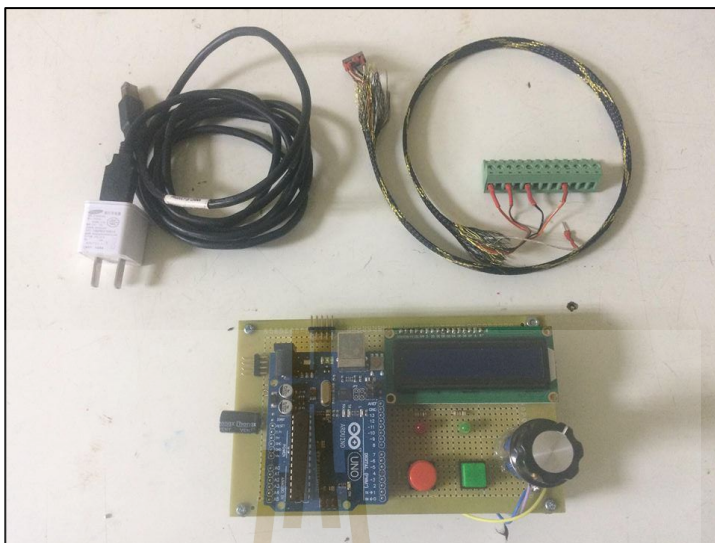
จากรูปที่ ก.1 คือ ชุดอุปกรณ์ทั้งหมดที่ใช้ในการถ่วงสมดุลงานปั้นเหรียญเลือด ซึ่งจะประกอบด้วย 2 ส่วนหลัก ๆ คือ เครื่องปั้นงาน และเครื่องถ่วงสมดุล โดยจะมีรายละเอียดย่อยต่าง ๆ ดังนี้

1.1.1 เครื่องปั้นงาน

เครื่องปั้นงานจะเป็นเครื่องที่ใช้ในการปั้นเหรียญงานเพื่อสร้างแรงดันสะท้อนให้กับเครื่องถ่วงสมดุลวิเคราะห์และแก้ไขปัญหาความสมดุลต่อไป โดยจะมีอุปกรณ์ดังรูปที่ ก.2 และรูปที่ ก.3



รูปที่ ก.2 ชุดอุปกรณ์ส่วนมอเตอร์



รูปที่ ก.3 ชุดอุปกรณ์ส่วนควบคุม

จากรูปจะเป็นชุดอุปกรณ์ของเครื่องปั่นงาน ซึ่งจะประกอบไปด้วยส่วนของมอเตอร์และส่วนควบคุม โดยจะมีรายละเอียดดังนี้

1.1.1.1 ส่วนของมอเตอร์มีรายละเอียดดังนี้

- ฐานเครื่องปั่นงาน
- มอเตอร์ BLDC
- บอร์ดขับมอเตอร์ BLDC

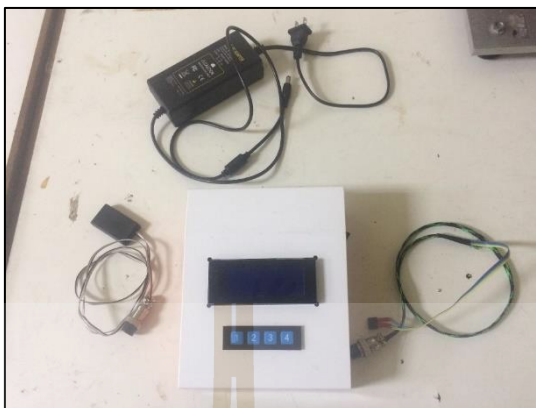
1.1.1.2 ส่วนของชุดควบคุมมีรายละเอียดดังนี้

- ชุด Controller ควบคุมและแสดงผล
- สายเชื่อมต่อระหว่างบอร์ดขับมอเตอร์กับชุด Controller
- Power Supply 5V

1.2 เครื่องถ่วงสมดุล

เครื่องถ่วงสมดุลเป็นเครื่องที่ใช้ในการวิเคราะห์การสั่นสะเทือนของเครื่องปั่นงานที่เกิดการสั่นสะเทือนเนื่องจากการเกิดความไม่สมดุลของงานปั่นเลือดในระหว่างการหมุนและคำนวณหาจุดการติดมวลแก้ไขความไม่สมดุลเพื่อลดการสั่นสะเทือนของเครื่องปั่นเลือดในขณะที่น่างานไปใช้งาน ซึ่งจะมีรายละเอียดดังนี้

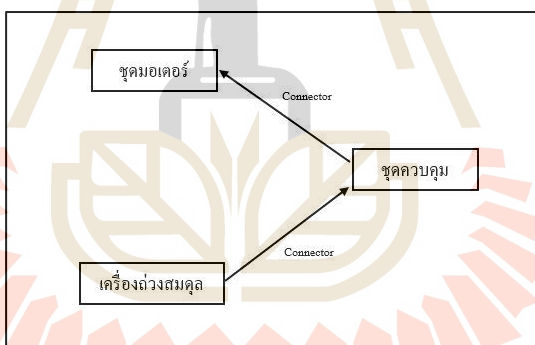
- เครื่องถ่วงสมดุล
- สายเชื่อมต่อระหว่างเครื่องถ่วงสมดุลกับชุด Controller
- Power Supply 12V



รูปที่ ก.4 ชุดเครื่องถ่วงสมดุล

2.1 การเชื่อมต่ออุปกรณ์ถ่วงสมดุล

การเชื่อมต่ออุปกรณ์ถ่วงสมดุลจะเชื่อมต่อกัน ดังรูปที่ ก.5



รูปที่ ก.5 แผนภาพการเชื่อมต่อของอุปกรณ์ถ่วงสมดุล

จากรูปจะต้องเชื่อมต่อชุดมอเตอร์กับชุดควบคุมเข้าด้วยกัน เพื่อควบคุมความเร็วรอบมอเตอร์และรับสัญญาณความเร็วรอบส่งต่อไปให้กับเครื่องถ่วงสมดุล และต้องเชื่อมต่อเครื่องถ่วงสมดุลกับชุดควบคุมเพื่อรับสัญญาณความเร็วรอบจากมอเตอร์มาคำนวณหาเฟืองสองขาของการสั้นสะเทือน

3.1 การใช้งานเครื่องถ่วงสมดุล

เมื่อทำการเชื่อมต่ออุปกรณ์ทั้งหมดเข้าด้วยกันแล้วให้ทำการเปิดเครื่องถ่วงสมดุลโดยจะอยู่ทางด้านข้างขวาของเครื่อง ดังรูปที่ ก.6



รูปที่ ก.6 เครื่องถ่วงสมดุล

จากนั้นเครื่องก็จะแสดงผล “Welcome for Balancing Machine” จากนั้นให้รอจนกระทั่งถึงหน้าต่าง Setup ความเร็วรอบที่จะใช้ Balance ดังรูปที่ ก.7

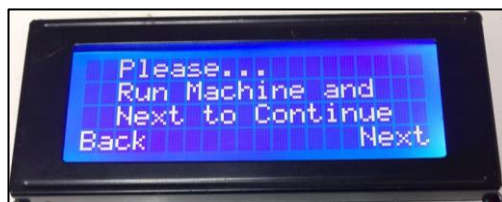


รูปที่ ก.7 จอแสดงผลและปุ่มกดของเครื่องถ่วงสมดุล

จากรูปที่แสดงจะให้ทำการ Setup ความเร็วรอบที่จะใช้ Balance โดยมีปุ่มให้ 4 ปุ่มตามหมายเลขซึ่งในแต่ละปุ่มจะบ่งบอกลักษณะการทำงานดังนี้

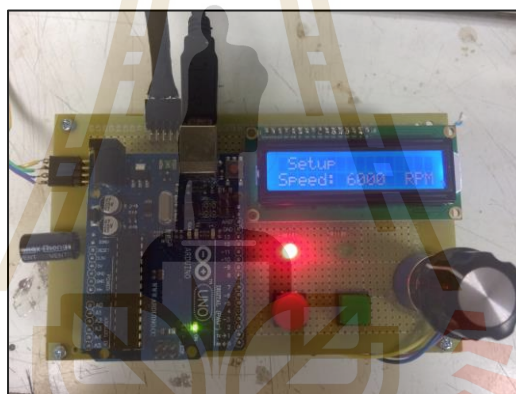
- ปุ่ม 1 จะกดเพื่อย้อนกลับไปยังขั้นตอนที่แล้ว
- ปุ่ม 2 จะกดเพิ่มความเร็วรอบในการ Balance
- ปุ่ม 3 จะกดลดความเร็วรอบในการ Balance
- ปุ่ม 4 จะกดเพื่อเข้าสู่ขั้นตอนถัดไปของการทำงาน

ซึ่งเมื่อทำการเลือกความเร็วรอบได้แล้วก็ให้กดปุ่ม 4 เพื่อดำเนินการถัดไป



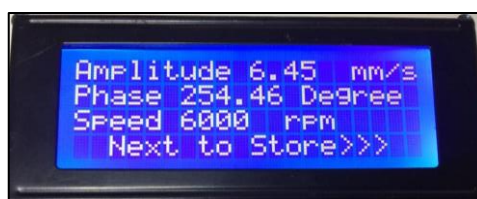
รูปที่ ก.8 จอแสดงผลให้เดินเครื่อง

จากรูปที่ ก.8 เมื่อถึงขั้นตอนนี้ให้ทำการตั้งค่าความเร็วรอบของมอเตอร์ปั่นงานให้ตรงกับความเร็วรอบที่ตั้งไว้และทำการ Run เครื่อง โดยกดปุ่มเขียวที่ชุดควบคุม ดังรูปที่ ก.9 และกดถัดไปที่เครื่องถ่วงสมดุล

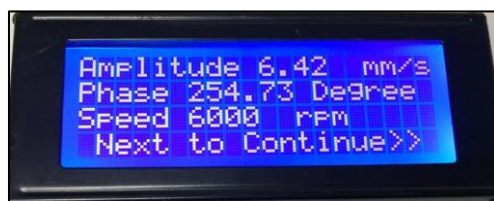


รูปที่ ก.9 Setup ความเร็วรอบที่ชุดควบคุม

เมื่อเดินเครื่องปั่นแล้วเครื่องถ่วงสมดุลก็จะทำการอ่านค่าการสั่นสะเทือนและมุมเฟสของการสั่นสะเทือนจากใบงาน ดังรูปที่ ก.10 เมื่อค่าที่อ่านเข้าสู่สถานะคงที่แล้วให้ทำการกดปุ่ม 4 เพื่อเก็บค่าการสั่นสะเทือนและมุมเฟสไว้ ดังรูปที่ ก.11

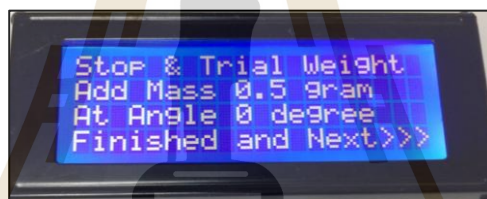


รูปที่ ก.10 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่วัดได้



รูปที่ ก.11 เครื่องถ่วงสมดุลแสดงผล Amplitude, Phase และ Speed ที่เก็บค่า

เมื่อทำการเก็บค่าแล้วให้ทำการกดปุ่ม 4 เพื่อเข้าสู่ขั้นตอนต่อไปซึ่งจะเป็นขั้นตอนการ Trial Weight ซึ่งตัวเครื่องนี้จะให้ผู้ใช้งานใส่มวลเข้าไปที่งาน 0.5 กรัม ที่มุม 0 องศาของใบงาน ดังรูปที่ ก.12 ให้ผู้ใช้ทำการหยุดเครื่องปั่นโดยกดปุ่มสีแดง ดังรูปที่ ก.13 และรอนกว่าเครื่องจะหยุดสนิท จากนั้นทำการติคมวลตามที่ตัวเครื่องได้บอกไว้ ดังรูปที่ ก.14



รูปที่ ก.12 เครื่องถ่วงสมดุลแสดงค่า Trial Weight ที่ต้องติคบนงานเลือด

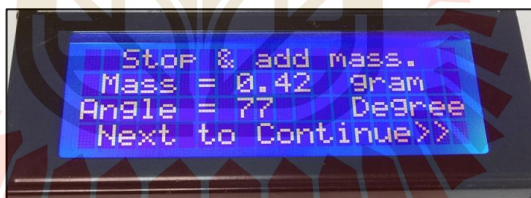


รูปที่ ก.13 แสดงความเร็วรอบขณะทำงาน



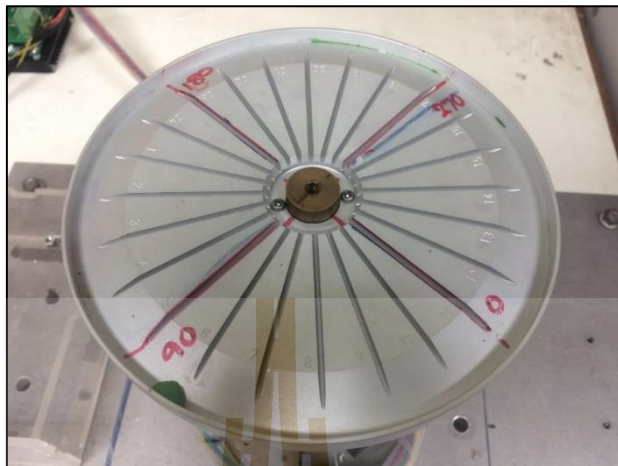
รูปที่ ก.14 ดิจิตอล 0.5 กรัมที่มุม 0 องศาบนใบงาน

เมื่อดิจิตอล Trial แล้วให้ทำการ Run เครื่องอีกครั้งเพื่อวัดและเก็บค่าการสั่นสะเทือนและมุมเฟสหลังจาก Trial Weight โดยกดปุ่ม 4 เพื่อทำการวัดตามขั้นตอนด้านบน ซึ่งเมื่อทำการวัดและเก็บค่าแล้ว เครื่องถ่วงสมดุลจะทำการคำนวณหามวลและมุมที่ใช้ในการถ่วงสมดุลบนใบงานเลือด ดังรูปที่ ก.15



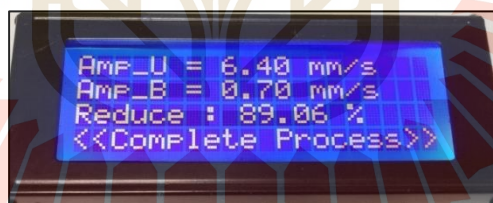
รูปที่ ก.15 แสดงขนาดมวลแก้ไขและมุมที่ต้องดิจิตอลแก้ไข

ให้ทำการ Stop เครื่องขึ้น ถอดมวล Trial ออก และทำการดิจิตอลขนาดและมุมตามที่เครื่องได้คำนวณมาลงบนงานเลือด ดังรูปที่ ก.16



รูปที่ ก.16 ทำการติดตั้งมวลตามที่เครื่องถ่วงสมดุลคำนวณ

เมื่อทำการติดมวลถ่วงสมดุลลงบนใบจานเลือดแล้ว ให้ทำการ Run เครื่องจักรอีกครั้ง เพื่อวัดการสั่นสะเทือนหลังจากการถ่วงสมดุลตามขั้นตอนเดิม ซึ่งค่าที่วัดจะถูกนำมาเปรียบเทียบกับของเดิมว่าเมื่อทำการถ่วงสมดุลงานเลือดแล้วการสั่นสะเทือนลดลงไปเท่าไร ดังรูปที่ ก.17 ก็ถือว่าการเสร็จสิ้นขั้นตอนการถ่วงสมดุลงานเลือด



รูปที่ ก.17 แสดงถึงค่าการสั่นสะเทือนที่ลดลงหลังจากการแก้ไขความไม่สมดุลของงานเลือด



ภาคผนวก ข

โปรแกรม Arduino

ตัวอย่างการเขียน โปรแกรมของเครื่องปั่นงาน

```
#include <FreqMeasure.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2);
int rpm = 0;
long Times, Time1, Time2, Time3;
int n = 0;
int j = 0;
int z = 0;
int STT, ST, SH, S, SF;
double sum = 0;
int count = 0;
int a = 169;
int b = 181;
int c = 192;
long C = 0;
int R = 0;
int x = 0;
int T;
long TimeSet = 60000; //time run centi second (10^-2)
void(*resetFunc)(void) = 0;
String Speed;
char Sp[10];
```

```
#include <PID_v1.h>
#define PIN_INPUT 0
#define PIN_OUTPUT 3
double Setpoint, Input, Output;
double Kp = 0.01, Ki = 0.0045, Kd = 0.001;//Load
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
void setup() {
  myPID.SetMode(AUTOMATIC);
  Serial.begin(115200);
  pinMode(A2, INPUT_PULLUP);
  pinMode(A3, INPUT_PULLUP);
  pinMode(2, OUTPUT);
  pinMode(A0, INPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  FreqMeasure.begin();
  lcd.begin();
  digitalWrite(2, LOW);
  analogWrite(PIN_OUTPUT, 0);
  digitalWrite(12, HIGH);
  digitalWrite(11, LOW);
}
void loop() {
  while (1) {
    digitalWrite(2, LOW);
    Times = millis();
    Time1 = Times / 10;
    Time2 = Times / 100;
    Time3 = Times / 1000;
    int Poten = analogRead(A0);
```



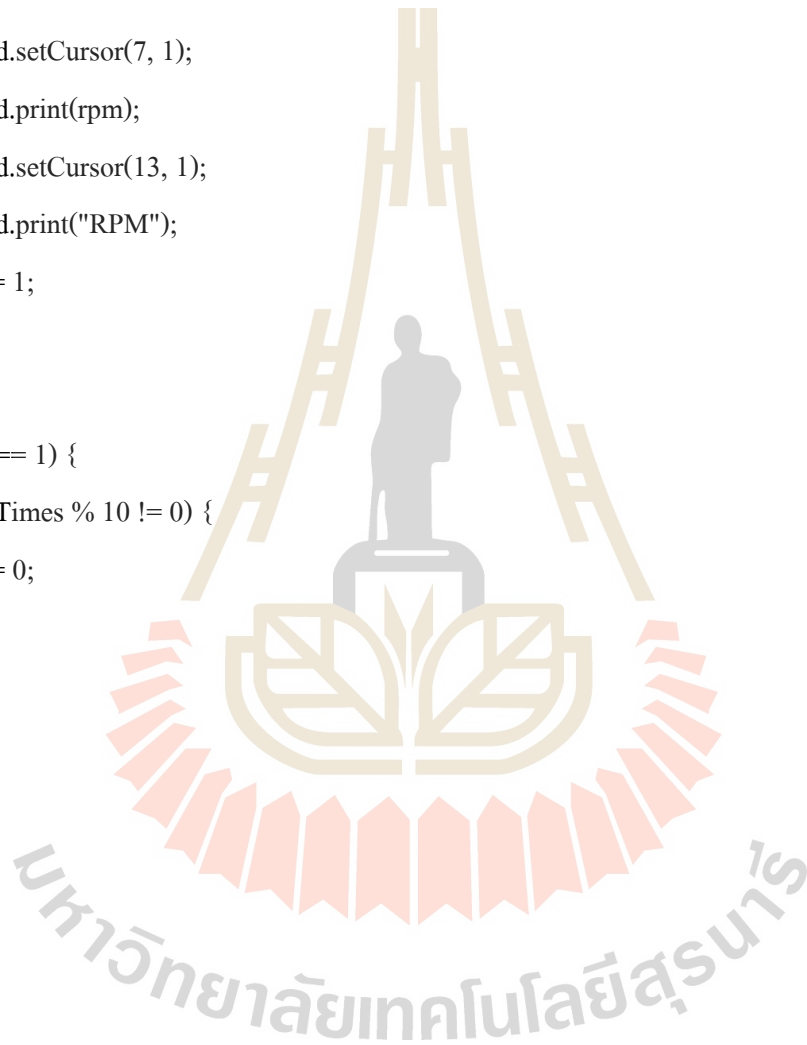
```
int Sw5 = digitalRead(A3);
unsigned int Set = map(Poten, 0, 1023, 0, 12000);
Set = (Set / 100) * 100;
if (Sw5 == LOW) {
    n = 1;
    C = Time3;
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    lcd.setCursor(0, 0);
    lcd.print("      ");
    lcd.setCursor(0, 1);
    lcd.print("      ");
}
lcd.setCursor(1, 0);
lcd.print("Setup");
if (Set < 10000) {
    lcd.setCursor(11, 0);
    lcd.print(" ");
}
if (Set < 1000) {
    lcd.setCursor(10, 0);
    lcd.print(" ");
}
if (Set < 100) {
    lcd.setCursor(9, 0);
    lcd.print(" ");
}
if (Set < 10) {
    lcd.setCursor(8, 0);
    lcd.print(" ");
}
```

```
}  
lcd.setCursor(7, 0);  
lcd.print(Set);  
lcd.setCursor(13, 0);  
lcd.print("RPM");  
while (n == 1) {  
    Times = millis();  
    Time1 = Times / 10;  
    Time2 = Times / 100;  
    Time3 = Times / 1000;  
    digitalWrite(2, HIGH);  
    int SwRe = digitalRead(A2);  
    int Poten = analogRead(A0);  
    unsigned int Set = map(Poten, 0, 1023, 0, 12000);  
    Set = (Set / 100) * 100;  
    if (SwRe == LOW) {  
        resetFunc();  
    }  
    Input = rpm;  
    Setpoint = Set;  
    myPID.Compute();  
    if (x == 0) {  
        if ((T) % TimeSet == 0) {  
            R++;  
            x = 1;  
        }  
    }  
    if (x == 1) {  
        if (T % TimeSet != 0) {  
            x = 0;  
        }  
    }  
}
```

```
}  
}  
if (R == 1) {  
    analogWrite(3, Output);  
}  
if (R == 2) {  
    digitalWrite(2, LOW);  
    analogWrite(3, 0);  
    Serial.println(rpm);  
    resetFunc();  
}  
//analogWrite(3,a);  
if (FreqMeasure.available()) {  
    // average several reading together  
    sum = sum + FreqMeasure.read();  
    count = count + 1;  
    if (count > 30) {  
        double frequency = F_CPU / (sum / count);  
        sum = 0;  
        count = 0;  
        rpm = (frequency * 60) / 3;  
    }  
}  
if (j == 0) {  
    if (Times % 10 == 0) {  
        T++;  
        lcd.setCursor(1, 0);  
        lcd.print("Setup");  
        if (Set < 10000) {  
            lcd.setCursor(11, 0);
```

```
    lcd.print(" ");
}
if (Set < 1000) {
    lcd.setCursor(10, 0);
    lcd.print(" ");
}
if (Set < 100) {
    lcd.setCursor(9, 0);
    lcd.print(" ");
}
if (Set < 10) {
    lcd.setCursor(8, 0);
    lcd.print(" ");
}
lcd.setCursor(7, 0);
lcd.print(Set);
lcd.setCursor(13, 0);
lcd.print("RPM");
lcd.setCursor(1, 1);
lcd.print("Run");
if (rpm < 10000) {
    lcd.setCursor(11, 1);
    lcd.print(" ");
}
if (rpm < 1000) {
    lcd.setCursor(10, 1);
    lcd.print(" ");
}
if (rpm < 100) {
    lcd.setCursor(9, 1);
```

```
    lcd.print(" ");
}
if (rpm < 10) {
    lcd.setCursor(8, 1);
    lcd.print(" ");
}
lcd.setCursor(7, 1);
lcd.print(rpm);
lcd.setCursor(13, 1);
lcd.print("RPM");
j = 1;
}
}
if (j == 1) {
    if (Times % 10 != 0) {
        j = 0;
    }
}
}
}
```



ตัวอย่างการเขียนโปรแกรมของเครื่องถ่วงสมดุล

```

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x23, 20, 4);

int Next, Back, Up, Down, n = 0;

int S_Speed = 1000;

int T_x = 0, T_y = 0, T_z = 0, T_L = 0, Speed, count = 0, c = 0;

double Z0, P0, Z1, P1, Po, Mt, Pt, Zv, Pv, Pz0p, Mc, Pc;

double Zx0, Zy0, Zx1, Zy1, Zxv, Zyv, Zxvn, Zyvvn, Zx0p, Zy0p;

double MassTrial = 0.5;

double TrialAngle = 0;

double DTR = (3.14159 / 180);

double RTD = (180 / 3.14159);

float Fac=31.1;

unsigned long Timemil, Timemic, T0, T1 = 0, T2, Ttrig, Tc;

float Z, Zf, Zm, P, Zmax, Zp = 0, Pp = 0, ZP = 0, PP = 0, Amp_U, Amp_B, Zpeak, Phase;

void setup() {
  Serial.begin(115200);
  lcd.begin();
  lcd.setCursor(3, 0);
  lcd.print("Welcome to...");
  delay(2000);
  lcd.setCursor(1, 1);
  lcd.print("HC Balance Machine");
  delay(2000);
  lcd.setCursor(2, 2);
  lcd.print("By.");
  delay(2000);
  lcd.setCursor(6, 2);
  lcd.print("Suranaree");

```

```
lcd.setCursor(1, 3);  
lcd.print("Medical Equipment");  
delay(5000);  
Clear();  
Standby();  
n = 0;  
pinMode(2, INPUT_PULLUP);  
pinMode(3, INPUT_PULLUP);  
pinMode(6, INPUT_PULLUP);  
pinMode(7, INPUT_PULLUP);  
pinMode(A0, INPUT);  
pinMode(8, INPUT);  
}  
void loop() {  
  while (1) {  
    while (n == 0) {  
      SetSw();  
      lcd.setCursor(3, 0);  
      lcd.print("Setup Speed...");  
      lcd.setCursor(0, 1);  
      lcd.print("<<<Speed Balance.>>>");  
      lcd.setCursor(2, 2);  
      lcd.print("Speed: ");  
      if (S_Speed < 10000) {  
        lcd.setCursor(13, 2);  
        lcd.print(" ");  
      }  
      lcd.setCursor(9, 2);  
      lcd.print(S_Speed);  
      lcd.setCursor(15, 2);
```

```
lcd.print("RPM");
lcd.setCursor(7, 3);
lcd.print("Up");
lcd.setCursor(11, 3);
lcd.print("Dn");
lcd.setCursor(16, 3);
lcd.print("Next");
if (T_x == 0) {
  if (Next == 0) {
    n = n + 1;
    Clear();
    T_x = 1;
  }
  if (Back == 0) {
    T_x = 1;
  }
  if (Up == 0) {
    S_Speed = S_Speed + 500;
    if (S_Speed > 10000) {
      S_Speed = 10000;
    }
    T_x = 1;
  }
  if (Down == 0) {
    S_Speed = S_Speed - 500;
    if (S_Speed < 1000) {
      S_Speed = 1000;
    }
    T_x = 1;
  }
}
```



```

}
if(T_x == 1) {
    if((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
        T_x = 0;
    }
}
}
while (n == 1) {
    SetSw();
    StandbyStart();
    if(T_x == 0) {
        if(Next == 0) {
            n = n + 1;
            Clear();
            T_x = 1;
        }
        if(Back == 0) {
            n = n - 1;
            Clear();
            T_x = 1;
        }
    }
    if(T_x == 1) {
        if((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
            T_x = 0;
        }
    }
}
while (n == 2) {
    Standby();

```

```

}
while (n == 3) {
    Timemil = millis() / 100;
    Timemic = micros();
    SetSw();
    Z = analogRead(A0);
    Speed = digitalRead(8);
    Zf = 0.2079 * Zf + 0.7921 * Z;
    if (T_y == 0) {
        if (Speed == HIGH) {
            T0 = Timemic;
            T2 = T0 - T1;
            T1 = T0;
            T_y = 1;
        }
    }
    if (T_y == 1) {
        if (Speed == LOW) {
            T_y = 0;
        }
    }
    if (T_z == 0) {
        if (Zf >= Zm) {
            Zm = Zf;
        }
        if (Zf < Zm) {
            Ttrig = Timemic;
            Tc = Ttrig - T1;
            P = (Tc * 360.00) / T2;
            //P=(Tc*360.00)/10000;
        }
    }
}

```

```

T_z = 1;
Zmax = Zm;
Zp = Zp + Zmax;
Pp = Pp + P;
count = count + 1;
if (count == 35) {
    Zp = Zp / count;
    Pp = Pp / count;
    ZP = ZP + Zp;
    PP = PP + Pp;
    Pp = 0;
    Zp = 0;
    count = 0;
    c = c + 1;
    if (c == 5) {
        Zpeak = ZP / c;
        Phase = PP / c;
        Zpeak = ((Zpeak*1.2659)-461.97)-Fac;
        if (Zpeak < 0) {
            Zpeak = 0;
        }
        PP = 0;
        ZP = 0;
        c = 0;
    }
}
}
}
if (T_z == 1) {
    if (Zf < Zm) {

```

```
Zm = Zf;
}
if (Zf > Zm) {
    Zm = Zf;
    T_z = 0;
}
}
if (Phase > 360) {
    Phase = 0;
}
if (T_L == 0) {
    if (Timemil % 10 == 0) {
        lcd.setCursor(0, 0);
        lcd.print("Amplitude ");
        if (Zpeak < 10) {
            lcd.setCursor(14, 0);
            lcd.print(" ");
        }
        lcd.setCursor(10, 0);
        lcd.print(Zpeak);
        lcd.setCursor(16, 0);
        lcd.print("mm/s");
        lcd.setCursor(0, 1);
        lcd.print("Phase ");
        if (Phase < 100) {
            lcd.setCursor(11, 1);
            lcd.print(" ");
        }
        if (Phase < 10) {
            lcd.setCursor(10, 1);
```

```

    lcd.print(" ");
}
lcd.setCursor(6, 1);
lcd.print(Phase);
lcd.setCursor(13, 1);
lcd.print("Degree");
lcd.setCursor(0, 2);
lcd.print("Speed ");
lcd.setCursor(6, 2);
lcd.print(S_Speed);
lcd.setCursor(12, 2);
lcd.print("rpm");
lcd.setCursor(2, 3);
lcd.print("Next to Store>>>");
T_L = 1;
}
}
if (T_L == 1) {
    if (Timemil % 10 != 0) {
        T_L = 0;
    }
}
if (T_x == 0) {
    if (Next == 0) {
        n = n + 1;
        Clear();
        Amp_U = Zpeak;
        Z0 = Zpeak;
        P0 = Phase;
        T_x = 1;
    }
}

```

```
}  
if (Back == 0) {  
    n = n - 2;  
    Clear();  
    T_x = 1;  
}  
}  
if (T_x == 1) {  
    if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {  
        T_x = 0;  
    }  
}  
}  
while (n == 4) {  
    SetSw();  
    lcd.setCursor(0, 0);  
    lcd.print("Amplitude ");  
    lcd.setCursor(10, 0);  
    lcd.print(Z0);  
    lcd.setCursor(16, 0);  
    lcd.print("mm/s");  
    lcd.setCursor(0, 1);  
    lcd.print("Phase ");  
    lcd.setCursor(6, 1);  
    lcd.print(P0);  
    lcd.setCursor(13, 1);  
    lcd.print("Degree");  
    lcd.setCursor(0, 2);  
    lcd.print("Speed ");  
    lcd.setCursor(6, 2);
```

```

lcd.print(S_Speed);
lcd.setCursor(12, 2);
lcd.print("rpm");
lcd.setCursor(1, 3);
lcd.print("Next to Continue>>");
if (T_x == 0) {
  if (Next == 0) {
    n = n + 1;
    Clear();
    T_x = 1;
  }
  if (Back == 0) {
    n = n - 3;
    Clear();
    T_x = 1;
  }
}
if (T_x == 1) {
  if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
    T_x = 0;
  }
}
}
while (n == 5) {
  SetSw();
  lcd.setCursor(0, 0);
  lcd.print("Stop & Trial Weight");
  lcd.setCursor(0, 1);
  lcd.print("Add Mass 0.5 gram");
  lcd.setCursor(0, 2);

```

```
lcd.print("At Angle 0 degree");  
lcd.setCursor(0, 3);  
lcd.print("Finished and Next>>>");  
if (T_x == 0) {  
    if (Next == 0) {  
        n = n + 1;  
        Clear();  
        T_x = 1;  
    }  
    if (Back == 0) {  
        n = n - 1;  
        Clear();  
        T_x = 1;  
    }  
}  
if (T_x == 1) {  
    if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {  
        T_x = 0;  
    }  
}  
while (n == 6) {  
    SetSw();  
    StandbyStart();  
    if (T_x == 0) {  
        if (Next == 0) {  
            n = n + 1;  
            Clear();  
            T_x = 1;  
        }  
    }  
}
```



```

}
}
if(T_x == 1) {
    if((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
        T_x = 0;
    }
}
while (n == 7) {
    Standby();
}
while (n == 8) {
    Timemil = millis() / 100;
    Timemic = micros();
    SetSw();
    Z = analogRead(A0);
    Speed = digitalRead(8);
    Zf = 0.2079 * Zf + 0.7921 * Z;
    if (T_y == 0) {
        if (Speed == HIGH) {
            T0 = Timemic;
            T2 = T0 - T1;
            T1 = T0;
            T_y = 1;
        }
    }
    if (T_y == 1) {
        if (Speed == LOW) {
            T_y = 0;
        }
    }
}

```

```

}
if (T_z == 0) {
    if (Zf >= Zm) {
        Zm = Zf;
    }
    if (Zf < Zm) {
        Ttrig = Timemic;
        Tc = Ttrig - T1;
        P = (Tc * 360.00) / T2;
        T_z = 1;
        Zmax = Zm;
        Zp = Zp + Zmax;
        Pp = Pp + P;
        count = count + 1;
        if (count == 35) {
            Zp = Zp / count;
            Pp = Pp / count;
            ZP = ZP + Zp;
            PP = PP + Pp;
            Pp = 0;
            Zp = 0;
            count = 0;
            c = c + 1;
            if (c == 5) {
                Zpeak = ZP / c;
                Phase = PP / c;
                Zpeak = ((Zpeak*1.2659)-461.97)-Fac;
                if (Zpeak < 0) {
                    Zpeak = 0;
                }
            }
        }
    }
}

```

```
    PP = 0;
    ZP = 0;
    c = 0;
  }
}
}
}
if (T_z == 1) {
  if (Zf < Zm) {
    Zm = Zf;
  }
  if (Zf > Zm) {
    Zm = Zf;
    T_z = 0;
  }
}
if (Phase > 360) {
  Phase = 0;
}
if (T_L == 0) {
  if (Timemil % 10 == 0) {
    lcd.setCursor(0, 0);
    lcd.print("Amplitude ");
    if (Zpeak < 10) {
      lcd.setCursor(14, 0);
      lcd.print(" ");
    }
    lcd.setCursor(10, 0);
    lcd.print(Zpeak);
    lcd.setCursor(16, 0);
```

```

lcd.print("mm/s");
lcd.setCursor(0, 1);
lcd.print("Phase ");
if (Phase < 100) {
    lcd.setCursor(11, 1);
    lcd.print(" ");
}
if (Phase < 10) {
    lcd.setCursor(10, 1);
    lcd.print(" ");
}
lcd.setCursor(6, 1);
lcd.print(Phase);
lcd.setCursor(13, 1);
lcd.print("Degree");
lcd.setCursor(0, 2);
lcd.print("Speed ");
lcd.setCursor(6, 2);
lcd.print(S_Speed);
lcd.setCursor(12, 2);
lcd.print("rpm");
lcd.setCursor(2, 3);
lcd.print("Next to Store>>>>");
T_L = 1;
}
}
if (T_L == 1) {
    if (Timemil % 10 != 0) {
        T_L = 0;
    }
}

```

```

}
if (T_x == 0) {
    if (Next == 0) {
        n = n + 1;
        Clear();
        Z1 = Zpeak;
        P1 = Phase;
        T_x = 1;
    }
    if (Back == 0) {
        n = n - 2;
        Clear();
        T_x = 1;
    }
}
if (T_x == 1) {
    if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
        T_x = 0;
    }
}
}
while (n == 9) {
    SetSw();
    lcd.setCursor(0, 0);
    lcd.print("Amplitude ");
    lcd.setCursor(10, 0);
    lcd.print(Z1);
    lcd.setCursor(16, 0);
    lcd.print("mm/s");
    lcd.setCursor(0, 1);

```

```
lcd.print("Phase ");
lcd.setCursor(6, 1);
lcd.print(P1);
lcd.setCursor(13, 1);
lcd.print("Degree");
lcd.setCursor(0, 2);
lcd.print("Speed ");
lcd.setCursor(6, 2);
lcd.print(S_Speed);
lcd.setCursor(12, 2);
lcd.print("rpm");
lcd.setCursor(1, 3);
lcd.print("Next to Continue>>");
if (T_x == 0) {
  if (Next == 0) {
    n = n + 1;
    Clear();
    T_x = 1;
  }
  if (Back == 0) {
    n = n - 3;
    Clear();
    T_x = 1;
  }
}
if (T_x == 1) {
  if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
    T_x = 0;
  }
}
```

```

}
while (n == 10) {
    SetSw();
    Po = 0;
    Mt = MassTrial;
    Pt = TrialAngle;
    Zx0 = Z0 * cos(P0 * DTR);
    Zy0 = Z0 * sin(P0 * DTR);
    Zx1 = Z1 * cos(P1 * DTR);
    Zy1 = Z1 * sin(P1 * DTR);
    Zv = sqrt((pow((Zx1 - Zx0), 2)) + (pow((Zy1 - Zy0), 2)));
    if (((Zy1 - Zy0) < 0 && (Zx1 - Zx0) < 0) || ((Zy1 - Zy0) < 0 && (Zx1 - Zx0) > 0)) {
        Pv = 360 + (atan2((Zy1 - Zy0), (Zx1 - Zx0)) * RTD);
    }
    else {
        Pv = atan2((Zy1 - Zy0), (Zx1 - Zx0)) * RTD;
    }
    if (((Zy0 - 2 * Zy0) < 0 && (Zx0 - 2 * Zx0) < 0) || ((Zy0 - 2 * Zy0) < 0 && (Zx0 - 2 *
Zx0) > 0)) {
        Pz0p = 360 + (atan2(Zy0 - 2 * Zy0, Zx0 - 2 * Zx0) * RTD);
    }
    else {
        Pz0p = atan2((Zy0 - 2 * Zy0), (Zx0 - 2 * Zx0)) * RTD;
    }
    Mc = Mt * (Z0 / Zv);
    if (Pv > Pz0p) {
        if (Po == 0) {
            Pc = Pt - abs(Pv - Pz0p);
            if (Pc < 0) {
                Pc = Pc + 360;
            }
        }
    }
}

```

```

    }
}
else {
    Pc = Pt + abs(Pv - Pz0p);
    if (Pc > 360) {
        Pc = Pc - 360;
    }
}
else {
    if (Po == 0) {
        Pc = Pt - (360 - abs(Pz0p - Pv));
        if (Pc < 0) {
            Pc = Pc + 360;
        }
    }
    else {
        Pc = Pt + (360 - abs(Pz0p - Pv));
        if (Pc > 360) {
            Pc = Pc - 360;
        }
    }
}
n = n + 1;
}

while (n == 11) {
    SetSw();
    lcd.setCursor(2, 0);
    lcd.print("Stop & add mass.");
    lcd.setCursor(1, 1);

```



```

lcd.print("Mass = ");
lcd.setCursor(8, 1);
lcd.print(Mc);
lcd.setCursor(14, 1);
lcd.print("gram");
lcd.setCursor(0, 2);
lcd.print("Angle = ");
lcd.setCursor(8, 2);
lcd.print(Pc);
lcd.setCursor(14, 2);
lcd.print("Degree");
lcd.setCursor(1, 3);
lcd.print("Next to Continue>>");
if (T_x == 0) {
  if (Next == 0) {
    n = n + 1;
    Clear();
    T_x = 1;
  }
  if (Back == 0) {
    n = n - 2;
    Clear();
    T_x = 1;
  }
}
if (T_x == 1) {
  if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
    T_x = 0;
  }
}
}

```

```

}
while (n == 12) {
    SetSw();
    StandbyStart();
    if (T_x == 0) {
        if (Next == 0) {
            n = n + 1;
            Clear();
            T_x = 1;
        }
    }
    if (T_x == 1) {
        if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {
            T_x = 0;
        }
    }
}
while (n == 13) {
    Standby();
}
while (n == 14) {
    Timemil = millis() / 100;
    Timemic = micros();
    SetSw();
    Z = analogRead(A0);
    Speed = digitalRead(8);
    Zf = 0.2079 * Zf + 0.7921 * Z;
    if (T_y == 0) {
        if (Speed == HIGH) {
            T0 = Timemic;

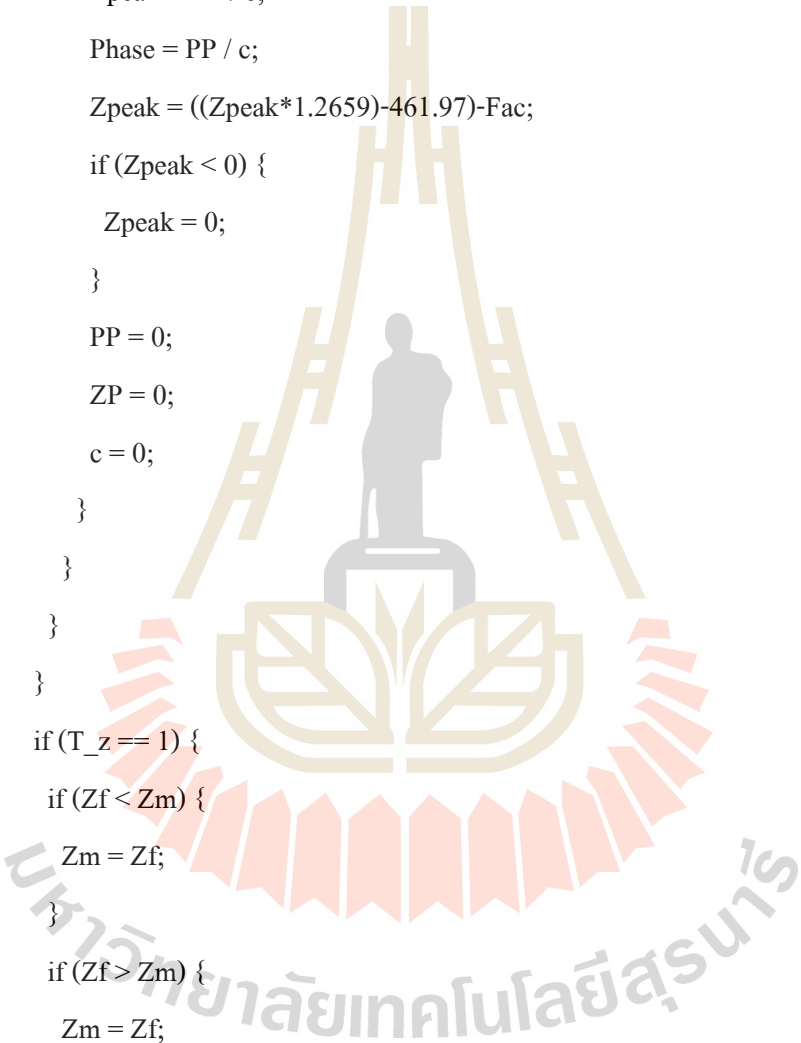
```

```

T2 = T0 - T1;
T1 = T0;
T_y = 1;
}
}
if (T_y == 1) {
    if (Speed == LOW) {
        T_y = 0;
    }
}
if (T_z == 0) {
    if (Zf >= Zm) {
        Zm = Zf;
    }
    if (Zf < Zm) {
        Ttrig = Timemic;
        Tc = Ttrig - T1;
        P = (Tc * 360.00) / T2;
        //P=(Tc*360.00)/10000;
        T_z = 1;
        Zmax = Zm;
        Zp = Zp + Zmax;
        Pp = Pp + P;
        count = count + 1;
    }
    if (count == 35) {
        Zp = Zp / count;
        Pp = Pp / count;
        ZP = ZP + Zp;
        PP = PP + Pp;
    }
}

```

```
Pp = 0;
Zp = 0;
count = 0;
c = c + 1;
if (c == 5) {
    Zpeak = ZP / c;
    Phase = PP / c;
    Zpeak = ((Zpeak*1.2659)-461.97)-Fac;
    if (Zpeak < 0) {
        Zpeak = 0;
    }
    PP = 0;
    ZP = 0;
    c = 0;
}
}
}
}
if (T_z == 1) {
    if (Zf < Zm) {
        Zm = Zf;
    }
    if (Zf > Zm) {
        Zm = Zf;
        T_z = 0;
    }
}
}
if (Phase > 360) {
    Phase = 0;
}
}
```



```
if (T_L == 0) {  
    if (Timemil % 10 == 0) {  
        lcd.setCursor(0, 0);  
        lcd.print("Amplitude ");  
        if (Zpeak < 10) {  
            lcd.setCursor(14, 0);  
            lcd.print(" ");  
        }  
        lcd.setCursor(10, 0);  
        lcd.print(Zpeak);  
        lcd.setCursor(16, 0);  
        lcd.print("mm/s");  
        lcd.setCursor(0, 1);  
        lcd.print("Phase ");  
        if (Phase < 100) {  
            lcd.setCursor(11, 1);  
            lcd.print(" ");  
        }  
        if (Phase < 10) {  
            lcd.setCursor(10, 1);  
            lcd.print(" ");  
        }  
        lcd.setCursor(6, 1);  
        lcd.print(Phase);  
        lcd.setCursor(13, 1);  
        lcd.print("Degree");  
        lcd.setCursor(0, 2);  
        lcd.print("Speed ");  
        lcd.setCursor(6, 2);  
        lcd.print(S_Speed);  
    }  
}
```

```
lcd.setCursor(12, 2);  
lcd.print("rpm");  
lcd.setCursor(2, 3);  
lcd.print("Next to Store>>>");  
T_L = 1;  
}  
}  
if (T_L == 1) {  
    if (Timemil % 10 != 0) {  
        T_L = 0;  
    }  
}  
if (T_x == 0) {  
    if (Next == 0) {  
        n = n + 1;  
        Clear();  
        Amp_B = Zpeak;  
        T_x = 1;  
    }  
    if (Back == 0) {  
        n = n - 2;  
        Clear();  
        T_x = 1;  
    }  
}  
if (T_x == 1) {  
    if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {  
        T_x = 0;  
    }  
}
```

```
}  
while (n == 15) {  
    SetSw();  
    lcd.setCursor(0, 0);  
    lcd.print("Amp_U = ");  
    lcd.setCursor(8, 0);  
    lcd.print(Amp_U);  
    lcd.setCursor(13, 0);  
    lcd.print("mm/s");  
    lcd.setCursor(0, 1);  
    lcd.print("Amp_B = ");  
    lcd.setCursor(8, 1);  
    lcd.print(Amp_B);  
    lcd.setCursor(13, 1);  
    lcd.print("mm/s");  
    lcd.setCursor(0, 2);  
    lcd.print("Reduce : ");  
    lcd.setCursor(9, 2);  
    lcd.print(((1 - (Amp_B / Amp_U)) * 100));  
    lcd.setCursor(15, 2);  
    lcd.print("%");  
    lcd.setCursor(0, 3);  
    lcd.print("<<Complete Process>>");  
    if (T_x == 0) {  
        if (Next == 0) {  
            n = 0;  
            Clear();  
            T_x = 1;  
        }  
    }  
}
```

```
if (T_x == 1) {  
    if ((Next == 1) && (Back == 1) && (Up == 1) && (Down == 1)) {  
        T_x = 0;  
    }  
}  
}  
}  
}  
}  
}  
void Clear() {  
    lcd.setCursor(0, 0);  
    lcd.print(" ");  
    lcd.setCursor(0, 1);  
    lcd.print(" ");  
    lcd.setCursor(0, 2);  
    lcd.print(" ");  
    lcd.setCursor(0, 3);  
    lcd.print(" ");  
}  
void Standby() {  
    lcd.setCursor(1, 1);  
    lcd.print("Please wait");  
    delay(1000);  
    lcd.setCursor(12, 1);  
    lcd.print(".");  
    delay(1000);  
    lcd.setCursor(13, 1);  
    lcd.print(".");  
    delay(1000);  
    lcd.setCursor(14, 1);  
    lcd.print(".");
```



```

delay(1000);
lcd.setCursor(15, 1);
lcd.print(".");
delay(1000);
lcd.setCursor(16, 1);
lcd.print(".");
delay(1000);
lcd.setCursor(17, 1);
lcd.print(".");
delay(1000);
lcd.setCursor(18, 1);
lcd.print(".");
delay(1000);
Clear();
n = n + 1;
}
void SetSw() {
  Next = digitalRead(6);
  Back = digitalRead(3);
  Up = digitalRead(2);
  Down = digitalRead(7);
}
void StandbyStart() {
  lcd.setCursor(2, 0);
  lcd.print("Please... ");
  lcd.setCursor(2, 1);
  lcd.print("Run Machine and");
  lcd.setCursor(2, 2);
  lcd.print("Next to Continue");
  lcd.setCursor(0, 3);

```

```
lcd.print("Back");  
lcd.setCursor(16, 3);  
lcd.print("Next");  
}
```





รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

ศิริพงษ์ ปะวะโก, อภิสิทธิ์ ทวีอภิรดีรัตน์, สุนทร โอบฐงาม และ จิระพล ศรีเสริฐผล. การออกแบบควบคุมความเร็วรอบของมอเตอร์กระแสตรงแบบไม่มีแปลงถ่านสำหรับเครื่องปั่นเม็ด เลือดแดง. การประชุมวิชาการระดับชาติ “นวัตกรรมและเทคโนโลยีวิชาการ 2017” ITC-2017, 26-27 ธันวาคม 2560, จังหวัดสุรินทร์.

A Taweepiradeerattana, S Pawako, A Rodpai and J Srisertpol. **Analytical design the vertical balancing device for hematocrit centrifuge machine.** International Conference on Mechanical, Electric and Industrial Engineering (MEIE2018), Hangzhou, China, May 26-28, 2018.

การประชุมวิชาการระดับชาติ "นวัตกรรมและเทคโนโลยีวิชาการ 2017"
"วิจัยจากองค์ความรู้สู่การพัฒนาที่ยั่งยืน"



การออกแบบระบบควบคุมความเร็วรอบของมอเตอร์กระแสตรงแบบไม่มีแปรงถ่าน สำหรับเครื่องปั่นเม็ดเลือดแดง

Speed Control System Design of the Brushless DC Motor for Hematocrit Centrifuge Machine

ศิริพงษ์ ปะวะโก^{1*}, อภิสิทธิ์ ทวีภักดิ์รัตน์², สุนทร โอษะงาม¹ และ จิระพล ศรีเสริมผล²

Siripong Pawako^{1*}, Apisit Taweapiradeeratana², Sontorn Odngam¹ and Jiraphon Srisertpool²

¹วิศวกรรมเมคคาทรอนิกส์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

²วิศวกรรมเครื่องกล สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

Email: siripong.p@gmail.com ผู้ประพันธ์หลัก

บทคัดย่อ

เครื่องปั่นเม็ดเลือดแดง(Hematocrit Centrifuge Machine: HCM) นั้นเป็นเครื่องที่สำคัญชนิดหนึ่ง ที่ใช้หลักการของแรงเหวี่ยงหนีศูนย์กลาง ในการแยกองค์ประกอบที่มีความถ่วงจำเพาะต่าง ๆ ภายในเลือดออกจากกันเป็นชั้น ๆ ตามลำดับ เพื่อสามารถนำตัวอย่างเลือดที่ผ่านการแยกองค์ประกอบไปวิเคราะห์ผลทางการแพทย์ต่อไป ปัจจุบันเครื่อง HCM ส่วนมากนำเข้าจากต่างประเทศและมีราคาแพง เนื่องจากเครื่อง HCM นั้นทำงานด้วยความเร็วรอบที่สูง ในช่วง 10,000-13,000 รอบต่อนาที (rpm) โดยความเร็วรอบคลาดเคลื่อนไม่เกิน 10% และค่าแรงเหวี่ยงหนีศูนย์กลางสัมพัทธ์(Relative Centrifugal Force: RCF) ต้องมีค่ามากกว่า 1,000 เท่าของค่าความเร่งโน้มถ่วงโลกที่ระดับน้ำทะเล ($g = 9.81 \text{ m/s}^2$) เป็นเวลา 5 นาที และสามารถทำเวลาจาก 0 – 12,000 rpm ไม่เกิน 10 sec งานวิจัยนี้จึงนำเสนอการออกแบบระบบควบคุมความเร็วรอบของมอเตอร์กระแสตรงแบบไม่มีแปรงถ่านสำหรับเครื่องปั่นเม็ดเลือดแดง โดยทำการเลือกมอเตอร์แบบไม่มีแปรงถ่าน(Brushless DC Motor: BLDC motor) อุปกรณ์ขับเคลื่อนมอเตอร์ แผ่นปั่นเหวี่ยงเม็ดเลือดแดง และออกแบบระบบควบคุมความเร็วรอบแบบปิด(Closed-loop) ที่มีตัวควบคุมแบบพีไอดี (PID controller) สำหรับเครื่อง HCM โดยใช้บอร์ดอาร์ดูโน(Arduino UNO) เป็นไมโครคอนโทรลเลอร์ และวัดความเร็วรอบจาก Hall-effect sensor ผลที่ได้จากการทดลองที่ออกแบบขึ้นนั้นสามารถทำงานได้เป็นไปตามข้อกำหนดของเครื่อง HCM ที่ความเร็วรอบ 12,000 rpm ค่าความคลาดเคลื่อนสูงสุดน้อยกว่า 5% เข้าสู่ความเร็วรอบที่ต้องการได้ในเวลา 7 วินาที สามารถให้ค่า RCF สูงสุด 13,000 g (ช่วง 2,000-13,000 g)

คำสำคัญ เครื่องปั่นเม็ดเลือดแดง, มอเตอร์กระแสตรงแบบไร้แปรงถ่าน และตัวควบคุมแบบพีไอดี

Abstract

The Hematocrit Centrifuge Machine (HCM) is an important device that uses centrifugal force to separate components of specific gravity within the blood from one layer to the next. To be able to take blood samples that have undergone separate components to further analyze the results. Currently, most of HCM's products are imported from abroad and expensive. Because the HCM has a high working speed range of 10,000-13,000 rpm of the RCF must be greater than 1,000 times the Earth's gravitational acceleration at sea level ($g = 9.81 \text{ m/s}^2$) for a period of 5 minutes and an effective control system for high speed control. So to reducing import cost. This research is offered "Speed Control System Design of the Brushless DC Motor for Hematocrit Centrifuge Machine" to design the speed control system of HCM motor using PID controller and Hall-effect sensor is a feedback measure to adjust the speed to meet the needs. By virtue of the Board of Directors (Arduino UNO) is a microcontroller. The result is that the designed HCM can operate at a speed of 12,000 rpm. The maximum tolerance of 5% (steady state error) to the desired speed in 7 seconds gives RCF a maximum of 13,000 g (1,200-13,000 g).

Keywords Hematocrit centrifuge machine, Brushless DC motor and PID controller

บทนำ

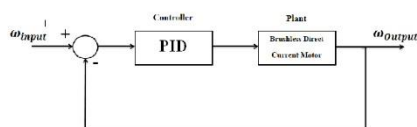
เครื่องปั่นเม็ดเลือดแดง (Hematocrit Centrifuge Machine: HCM) นั้นเป็นเครื่องสำคัญที่ใช้หลักการของแรงเหวี่ยงหนีศูนย์กลาง ในการแยกเม็ดเลือดแดงออกจากพลาสมาและเม็ดเลือดขาวภายในเลือด ซึ่งเป็นเครื่องมือที่มีความจำเป็นอย่างมากทั้งในทางการแพทย์และวิทยาศาสตร์

ปัจจุบันเครื่องHCM ส่วนมากที่ใช้ภายในประเทศนั้นนำเข้าจากต่างประเทศซึ่งมีราคาแพง เนื่องจาก เครื่องHCM มีความเร็วรอบในการทำงานที่สูง อยู่ในช่วง 10,000-13,000 rpm โดยมีความคลาดเคลื่อนไม่เกิน 10%ของความเร็วรอบที่กำหนด และค่าแรงเหวี่ยงหนีศูนย์กลาง(RCF) ต้องมีค่ามากกว่า 1,000 g เป็นระยะเวลาอย่างน้อย 5 นาที เพื่อให้เม็ดเลือดแยกตัวออกจากองค์ประกอบอื่น กล่าวโดยสรุปแล้วหนึ่งในปัจจัยสำคัญที่ทำให้เครื่องHCMมีมูลค่าสูง นั่นก็คือระบบควบคุมความเร็วรอบของเครื่องHCM ซึ่งเป็นที่มาของบทความวิจัยนี้ โดยมีวัตถุประสงค์คือ เพื่อออกแบบระบบควบคุมความเร็วรอบของมอเตอร์กระแสตรงแบบไม่มีแปรงถ่าน (Brushless DC Motor: BLDC motor) สำหรับเครื่อง HCM โดยมีเงื่อนไขข้อหนึ่งคือสามารถควบคุมความเร็วรอบที่ 12,000 rpm และมีค่าความคลาดเคลื่อน (steady state error)ไม่เกิน 5% โดยใช้เวลาในการเข้าสู่สภาวะคงตัว(setting time) ไม่เกิน 10 วินาที และใช้ตัวควบคุมแบบพีไอดี (PID controller) เป็นตัวควบคุม

วัสดุ อุปกรณ์ และวิธีดำเนินการวิจัย

วัสดุอุปกรณ์ที่ใช้ในงานวิจัยได้แก่

1. เครื่องคอมพิวเตอร์ควบคุมการทำงานและแสดงผล,
2. มอเตอร์ Brushless DC พร้อมชุดควบคุม,
3. จานปั่นเหวี่ยง(Centrifuge disc)สำหรับเครื่อง HCM,
4. โปรแกรม Arduino พร้อมบอร์ด UNO-R3 สำหรับเขียนโปรแกรมควบคุม,
5. โปรแกรม Matlab สำหรับช่วยในการคำนวณทางคณิตศาสตร์



รูปที่ 1 แบบจำลองทางคณิตศาสตร์ของระบบ BLDC motor

เบื้องต้นการใช้โปรแกรม Matlab/Simulink จำลอง แบบจำลองทางคณิตศาสตร์ของ BLDC motor โดยตัดแปลงจากบทความ Speed Estimation Of 3-Phase BLDC Motor Using Genetic Algorithm (Pawin Jawayon& Jiraphon Srisertpol,2013) เพื่อใช้ในการออกแบบระบบควบคุมต่อไป โดยรูปแบบการหาค่าส่วนขยาย(gain Kp, Ki, Kd) ที่เหมาะสมของ PID controller ในงานวิจัยนี้ใช้การลองผิดลองถูก(Trial and error)ในการหาค่าที่เหมาะสม และใช้โปรแกรม Arduino ในการเขียนชุดคำสั่งควบคุม

ตารางที่ 1 แสดงค่าอัตราขยายของ PID-controller

Gain ของ PID Controller	Value
ค่า Kp	0.0100
ค่า Ki	0.0045
ค่า Kd	0.0010

หลังจากทำการปรับแต่งจนได้ค่าที่ทำให้ผลการตอบสนองเป็นไปตามเงื่อนไขแล้ว(แสดงในผลการทดลอง) พร้อมเก็บข้อมูลการตอบสนองของ BLDC motor แล้วนำไปแสดงผลเป็นกราฟด้วยโปรแกรม Matlab เพื่อนำไปพิจารณาคต่อไป

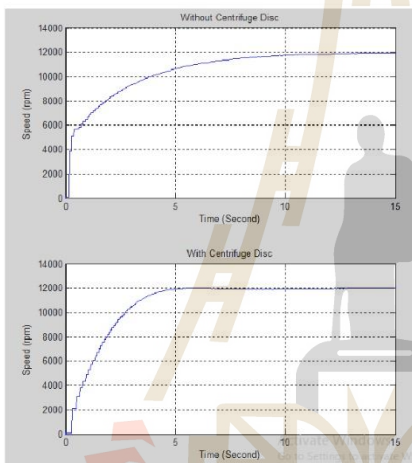
ในการทดลองเพื่อเก็บข้อมูลจะแบ่งออกเป็น 3

1. ผลการตอบสนอง BLDC Motor ในช่วงเวลา 0-15 วินาที ที่ความเร็วรอบ 12,000 rpm เพื่อพิจารณาผลการตอบสนองชั่วคราว (transient response) โดยทดสอบทั้งแบบปราศจากCentrifuge discและแบบติดตั้ง Centrifuge disc
2. ผลการตอบสนอง BLDC Motor ในช่วงเวลา 0-300 วินาที (หรือ 0 - 5 นาที) ที่ความเร็วรอบ 12,000 rpm เพื่อดูผลการตอบสนองที่สภาวะคงตัว (steady state response) ว่าสามารถรักษาเสถียรภาพความเร็วรอบได้ตลอดการทำงานหรือไม่ และเป็นการ

จำลองสภาวะการทำงานจริงอีกด้วย โดยทดสอบทั้งแบบปราศจาก Centrifuge disc และแบบติดตั้ง Centrifuge disc

3. ผลการตอบสนองของ BLDC Motor แบบ Multi-step input ที่ความเร็วรอบ 5,000 rpm, 13,000 rpm และ 9,000 rpm โดยใช้เวลา 60 วินาทีในแต่ละความเร็วรอบ เพื่อดูประสิทธิภาพในการควบคุมความเร็วรอบของ PID controller และทดสอบโดยปราศจาก Centrifuge disc

ผลการวิจัย

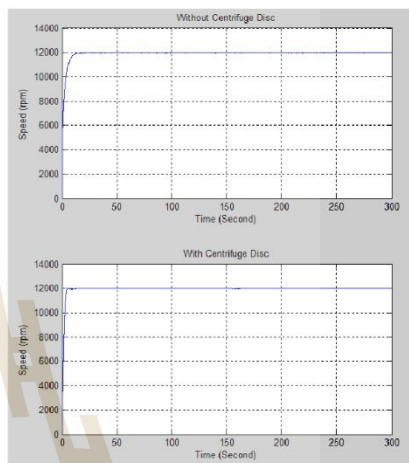


รูปที่ 2 ผลการตอบสนองการทดสอบที่ 1

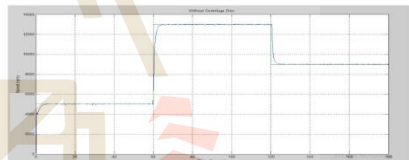
จากรูปที่ 2 แสดงกราฟผลการตอบสนองของ BLDC Motor ในช่วงเวลา 0-15 sec ที่ความเร็วรอบเริ่มจากหยุดนิ่งถึง 12,000rpm เพื่อพิจารณาผลการตอบสนองชั่วคราวพบว่า ที่การทดสอบแบบ Without Centrifuge Disc มี Setting time 10 sec, ไม่มีค่า Overshoot และที่การทดสอบแบบ With Centrifuge Disc มี Setting time 5 sec และมี Overshoot ไม่ถึง 1%

จากรูปที่ 3 ผลการตอบสนอง BLDC Motor ในช่วงเวลา 0 - 5 นาที ที่ความเร็วรอบ 12,000 rpm เพื่อดูผลการตอบสนองที่สภาวะคงตัว ว่าสามารถรักษาเสถียรภาพความเร็วรอบได้ตลอดการทำงานพบว่า ที่การทดสอบแบบ Without Centrifuge Disc

เริ่มพิจารณาผลการตอบสนองที่สภาวะคงตัวที่เวลา 10 sec ถึง 300 sec และที่การทดสอบแบบ With Centrifuge Disc เริ่มพิจารณาผลการตอบสนองที่สภาวะคงตัวที่เวลา 5 sec ถึง 300 sec โดยการทดลองทั้งสองมีค่าไม่ถึง 1% ตลอดการทำงาน



รูปที่ 3 ผลการตอบสนองการทดสอบที่ 2



รูปที่ 4 ผลการตอบสนองการทดสอบที่ 3

จากรูปที่ 4 แสดงผลการตอบสนอง BLDC Motor แบบ Multi-step input ที่ความเร็วรอบ 5,000 rpm, 13,000 rpm และ 9,000 rpm โดยใช้เวลา 60 วินาทีในแต่ละความเร็วรอบ เพื่อทดสอบประสิทธิภาพในการควบคุมความเร็วรอบของ PID controller และทดสอบโดยปราศจาก Centrifuge disc พบว่า ทุกการเปลี่ยนแปลงความเร็วรอบ ระบบสามารถเข้าสู่ค่าความเร็วรอบที่กำหนดได้ในเวลาไม่ถึง 5 sec และมีค่าความคลาดเคลื่อนที่สภาวะคงตัวไม่เกิน 2%

อภิปรายผล

จากผลการทดลองพบว่า ในการทดลองที่ 1 ทดสอบผลการตอบสนอง BLDC Motor ในช่วงเวลา 0-15 sec ที่ความเร็วรอบ เริ่มจาก หยุดนิ่งถึง 12,000rpm เพื่อพิจารณาผลการตอบสนองชั่วคราว ซึ่ง Setting time ของแบบ with centrifuge disc ใช้เวลานานกว่าแบบ without centrifuge disc เนื่องจาก แบบ with centrifuge disc มีมวลในระบบมากกว่าทำให้เกิดความเฉื่อย (inertial) controller ตัวนี้ถูกออกแบบมาเพื่อใช้กับ BLDC motor with centrifuge disc พอนำไปทดสอบแบบ without centrifuge disc

สรุปผลการวิจัย

ผลที่ได้จากการทดลองที่ออกแบบขึ้นนั้น สามารถทำงานได้เป็นไปตามข้อกำหนดของเครื่อง HCM ที่ความเร็วรอบ 12,000 rpm มีค่า Overshoot ไม่ถึง 1% เข้าสู่ความเร็วรอบที่ต้องการได้ในเวลา 5 วินาที (With Centrifuge Disc) และในช่วงการทำงาน ค่าความคลาดเคลื่อนสูงสุดต้องน้อยกว่า 5%

กิตติกรรมประกาศ

บริษัท สุรนารีแพทยภัณฑ์ จำกัด
มหาวิทยาลัยเทคโนโลยีสุรนารี

เอกสารอ้างอิง

- [1] ชูชาติ อารีจิตรานุสรณ์. (2544). "เครื่องมือวิทยาศาสตร์. ขอนแก่น:คลังน่านาวิทยา.
- [2] ศูนย์การแพทย์กาญจนาภิเษก มหาวิทยาลัยมหิดล. (2556). การศึกษาค่าเคมีคลินิกในพลาสมาที่ "ได้จากการปั่นแยกเม็ดเลือด. นครปฐม.
- [3] Manjita Srivastava, M C Srivastava, Smriti Bhatnagar, (2009), " Control System ", New Delhi:Tata McGraw-Hill
- [4] G Prasad, Venkateswara Reddy M, Dr. P V N Prasad, Dr. G Tulasi Ram Das, " Speed control of Brushless DC motor with DSP controller using Matlab ", International Journal of Engineering Research and Applications , Vol. 2, pp.2120-2125 , May-Jun 2012

[5] Pawin Jawayon, Jiraphon Srisertpol, " Speed Estimation Of 3-Phase BLDC Motor Using Genetic Algorith ", International Journal of Engineering Science and Innovative Technology , Vol. 2, January 2013

[6] Ms.Manjusha Patil, "Modelling and simulation of dc drive using PI and PID controller", International Journal of Innovative Research in Electrical, Electronic, Vol. 2, December 2014

Analytical design of the vertical balancing device for hematocrit centrifuge machine

Apisit Taweeapiradeerattana, Siripong Pawako, Autsadayut Rodpai, Nitisak Numanoy and Jiraphon Srisertpol¹

School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand

¹ E-mail: jiraphon@sut.ac.th

Abstract Hematocrit Centrifuge Machines (HCM) apply centrifugal force to separate suspended particles from a liquid or liquids which have different densities. Centrifuges of which the forces are greater than gravity can greatly accelerate separations that occur naturally as a result of density differences. HCM were intended to separate red blood cells from plasma in order to facilitate later diagnosis process. Main components include a motor that supplies power to the disk to quickly attain required speed. An unbalance found on a disk which creates vibration on the structure of HCM was caused by either non-homogeneous material or manufacturing process. Therefore, to balance the disk, this paper presents the analysis and design of the Vertical Balancing Device (VBD) for HCM using inexpensive accelerometer to measure amplitude with Hall-effect sensor to estimate rotating speed and maximum phase trigger for balancing. The experimental results show that VBD can balance the disk which in turn effectively reduces vibration to the determined dynamic response criteria effectively of unbalance vibration.

1. Introduction and problem statement

The measuring hematocrit technique was discussed of the method which consists of the components of whole blood, indications for measuring hematocrit and measuring technologies [1]. Those processes require a microhematocrit centrifugation technique of the HCM that used to help diagnose blood loss. The HCM devices intended to determine the blood's hematocrit, the ratio of red-cell volume to whole blood volume, expressed as a decimal, a fraction, or a percentage. Figure 1 (a) shows the components including a motor that supplies power to the shaft and disk, centrifuge heads that spin on the disk, and a lid latch with lid. The centrifuge head contains shields that cover the disk and turns on a spindle. A safety shield in the chamber surrounds the disk. This model is equipped with an LCD display, a keypad, a timer, a braking system, and a tachometer, enabling the user to program a specific speed up to 20,000 rpm and view timer status. Principles of operation, centrifuge heads apply centrifugal force to separate liquids of different densities in the tubes. By generating forces many times greater than gravity, centrifuges can greatly accelerate separations [2]. The HCM, a capillary tube pocket slot of a centrifuge head unit, quickly attains required speed to spin microcapillary tube samples as shown in figure 1(b). These tubes require only small blood samples taken from a puncture site or from venous blood specimen.



However, configurations of many disks are manufactured in the ways that produce dissymmetry, rough surfaces on turning or milling, shown in figure 1(c). As mentioned previously, the rotating disk at high speed is affected by the vibration. The operation of the HCM may be noisy from vibration of being dragged into the tube cracking. It will also shorten the life of the machine. It causes a problem of unbalance which is often simply defined as the unequal distribution of the weight of a disk about its rotating centerline. The International Standard ISO 10816 in the Class 1 at Zone A [3] is general used as conditions and procedures for the measurement. The general evaluation criteria, presented in terms of both vibration magnitude and changing vibration, are related to both operational monitoring and acceptance testing. Therefore, it is necessary to correct the rotating disk unbalance before use. Nowadays, the tools used to find the unbalance for high performance rotary disk are very expensive and unsuitable for use in small and medium-size industries with limited operating costs.

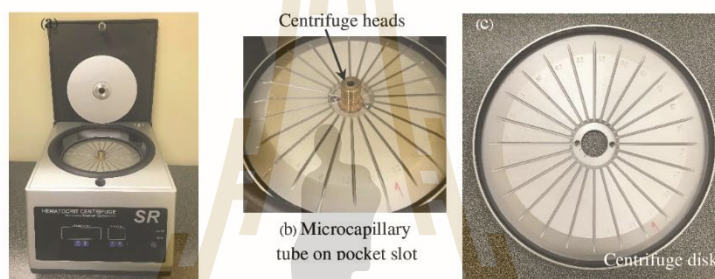


Figure 1. Hematocrit Centrifuge Machine (HCM).

This project is a research and development study jointly with a private company in Thailand who is a manufacturer of the vertical balancing device for HCM. VBD determines an analytical design qualification for balancing HCM, balancing the disks to reduce unbalance to minimum. Select a balance tolerance within this range that is cost effective and amount of unbalance allowed in a product can easily be determined by the manufacturer.

2. HCM system and controller identification

Controller is a HCM which monitors and alters the operating conditions of a given dynamical system. The control algorithms are usually divided into two main types: open-loop for parameter identification and closed-loop for designing PID controller [4]. So the mathematical model of the BLDC motor is controlled by the DC link voltage v . The transfer function of the BLDC motor can be described by the following equation:

$$\frac{\omega(s)}{v(s)} = \frac{K_t}{JLs^2 + (JR + BL)s + (BR + K_t K_b)} \quad (1)$$

where ω is the reference speed. Specifications and system parameters used Parameter Estimation Toolbox in MATLAB for the parameters identification of the HCM system and controller identification are listed in table 1. Parameter Estimation toolbox is a Simulink-based product for estimating parameters of model from experimental data [5]. The investigation of the coefficients in the mathematical models equation (1) is a very simple problem because of restricted information. Input voltage and speed can be measured, where $\hat{\omega}$ is the speed from the mathematical model and e is the error with difference between two outputs as shown in figure 2(a).

In practice, the PID controller has been widely used in industrial applications owing to its simplicity, robustness, reliability and easy tuning parameters. PID commonly interpreted as

proportional, integral and derivative controller possesses the step response conditions following at operating speed 2,000 rpm: the steady state error is less than 0.5%, maximum overshoot is 5% and settling time at 1.0% error is less than 15 sec. As a result, PID Tuner provides a fast and widely applicable single-loop PID tuning method for the Simulink® PID Controller blocks [6], as listed in table 1. This method, PID controller parameters can be tuned to achieve a robust design with the desired response time as shown in figure 2(b).

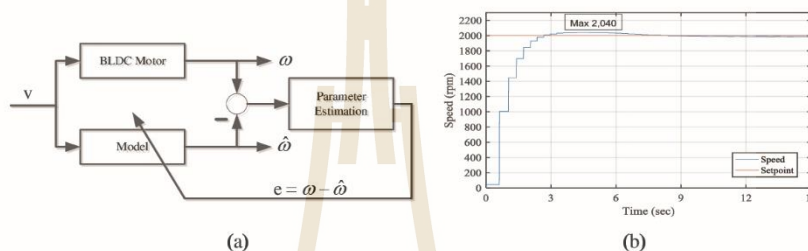


Figure 2. (a) Parameter estimation diagram (b) Response of PID controller at operating speed.

Table 1. Parameter of the HCM system and controller identification.

Description	Parameters	Value	Unit
BLDC motor			
Moment of Inertia	J	4.37	Kg.m ²
Friction Coefficient	B	4.88	Nm.s/rad
Resistance	R	28.94	Ω/phase
Inductance	L	27.17	H/phase
Back EMF	K _b	0.1x10 ⁻³	V/1000 rpm
Torque Constant	K _t	0.912	kNm/A
PID Controller			
Proportional Gain	K _p	0.0100	-
Integral Gain	K _i	0.0045	sec ⁻¹
Derivative Gain	K _d	0.0010	sec

3. Magnitude and phase processing

A vibration is the movement of a physical quantity in relation to a reference location in a cyclically increasing and decreasing manner as a function of time. The most important features of machine vibration change according to equation (2).

$$x(t) = A \sin(\omega t + \phi) \tag{2}$$

where *A* is amplitude of vibration. In this paper Arduino UNO, which is a microcontroller with 3-axis accelerometer (ADXL-335) was measured signal voltage output from magnitude of vibration transmission. It is suitable because of the VBD speed operated under our condition. Hall-effect sensor was applied for referenced signal between speed signal of motor and acceleration signal to find phase angle that can bring about balancing HCM disk. The response to compare signal between accelerometer and Hall-effect for finding the phase of maximum vibration by

$$\phi = \frac{(T_m - T_0) \times 360}{T_w} \tag{3}$$

where ϕ is phase angle of maximum vibration, T_m is maximum vibration period in each cycle, T_0 is starting time, T_1 is ending time and $T_w = T_1 - T_0$ is period time between T_0 and T_1 .

Signal data of the vibration and speed rotor from the accelerometer and Hall-effect sensor are acquired for signal processing. Vibration due to the unbalance is seen as a maximum peak-amplitude in the harmonic at the vibration frequency. The vibration level and the phase of the rotational frequency of the rotor signal could be read directly from the display. Figure 3 shows the proposed methodology, which combines the amplitude of vibration and the trigger pulse reference signals from rotor analyses to precisely determine the motor condition according to the following procedure:

1. Holding vibration and rotational frequency signals during BLDC motor operation;
2. Processing vibration and rotation frequency signals in the time domain to determine the phase angle between the signals from (2) and (3)
3. Accepting whether the disk is balanced or unbalanced from the results of the above analyses;
4. Predetermining the trial mass and its position for correcting unbalance if any.

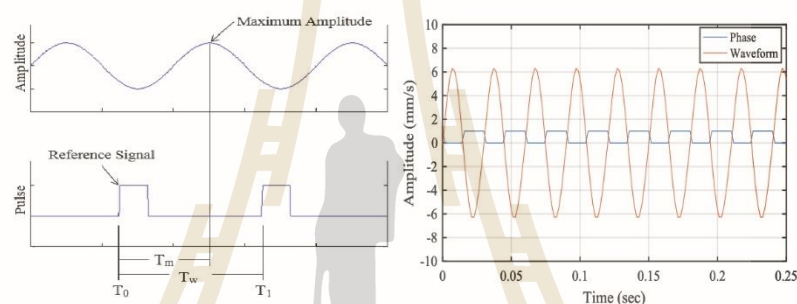


Figure 3. Signal of accelerometer and Hall-effect sensor, left (Simulation) and right (Experimental).

4. Unbalance correction algorithm

The unbalance correction algorithm used herein is based on vector diagram calculations for single-plane balancing [7]. The magnitude and angular position of the correction mass can be determined by representing in figure 4 as follows:

- i. Measuring; A_0 is the initial amplitude, ϕ_0 is initial phase of vibration, and its direction is given in polar coordinate.
- ii. Installing trial the given mass, M_T is known as trial mass and θ_T is trial phase to install on disk. After that, executing and measuring amplitude and phase again in terms A_1 and ϕ_1 respectively.
- iii. Transforming amplitudes and phases in (i) and (ii) into rectangular coordinate, calculating the net amplitude, A_T between A_0 and A_1 given by

$$A_T = \sqrt{(A_x)^2 + (A_y)^2} \quad (4)$$

where $A_x = A_1 \cos \phi_1 - A_0 \cos \phi_0$ and $A_y = A_1 \sin \phi_1 - A_0 \sin \phi_0$.

- iv. Calculating the net phase ϕ_T from quadrant condition and $A^\dagger = A_y / A_x$ is defined according to the following procedure in table 2.
- v. After that, calculating the vibration amplitude that is proportional to the unbalanced mass M_B which obtains the relationship

$$M_B = \frac{A_0}{A_T} M_T \quad (5)$$

vi. Determining the position of the mass relative to the position of the trial mass as,

$$\theta_B = \theta_T + \phi_T - \phi_B \begin{cases} \phi_B = \phi_0 + 180^\circ & ; 0^\circ \leq \phi_0 < 179^\circ \\ \phi_B = \phi_0 - 180^\circ & ; 180^\circ \leq \phi_0 < 359^\circ \end{cases} \quad (6)$$

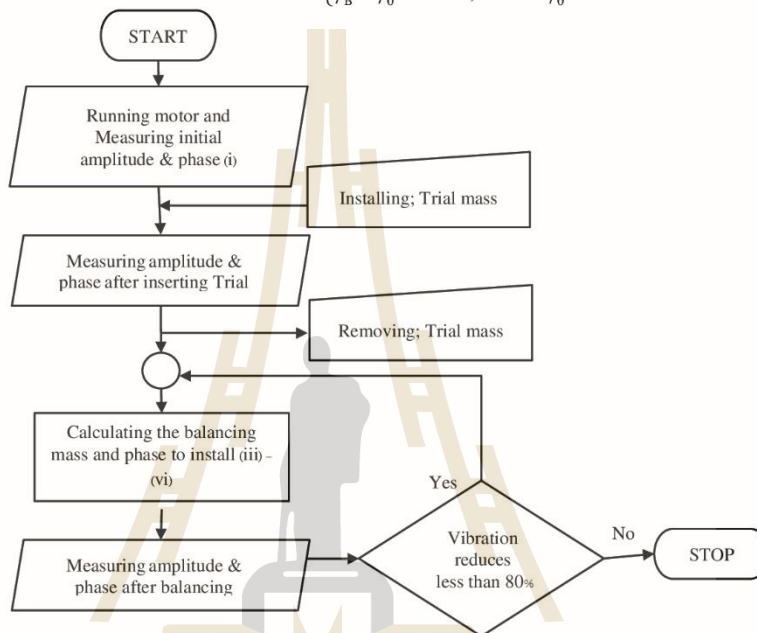


Figure 4. Block diagram of algorithm for automatic unbalance detection and correction.

The calculated angle is measured from the position marked on the rotor indicating the point where the trial mass was mounted. If it is a positive angle, it is measured in the direction of rotation. A negative angle is measured in the opposite sense from condition equation (6).

Table 2. Phase quadrant condition.

Condition	Phase formulas
$A_y > 0$ and $A_x > 0$	$\phi_T = \tan^{-1}(A')$
$A_y > 0$ and $A_x < 0$	$\phi_T = 180^\circ - \tan^{-1}(A')$
$A_y < 0$ and $A_x < 0$	$\phi_T = 180^\circ + \tan^{-1}(A')$
$A_y < 0$ and $A_x > 0$	$\phi_T = 360^\circ - \tan^{-1}(A')$

5. Experimental results and analysis

Experiments were also carried out to verify the correction algorithm. The balance correction algorithms were developed under the Arduino UNO-R3. A widely used open-source microcontroller board base on the Arduino software in Windows makes it easy to write a code and upload it to the board. The board was equipped with sets of digital and analog input/output (I/O) pins that were interfaced to various expansion boards or other circuits and experimental setup in figure 5. They were

composed of (a) Arduino UNO-R3 microcontroller board; (b) a brushless DC motor (300 Watt, 220-240 VDC with drive board); (c) -3axis accelerometer to measure acceleration with a minimum full-scale range of $3\pm$ g (Arduino-ADXL335); (d) Hematocrit Centrifuge disk setup on head of motor (diameter 175 mm., weight 420 g., material made of Aluminum 6063); (e) laptops running the application software in Arduino IDE. The BLDC motor was operated at a constant speed of 33.33 Hz (2,000 rpm).



Figure 5. Vertical Balancing Device (VBD) experimental.

In the first trial, the HCM was operated without adding any mass to the disk and was therefore considered balanced. Figure 6 shows the signal acquired (red line) by the accelerometer, which was used to determine both the amplitudes and phases of the vibration signal before unbalancing. The signal's peak amplitude was approximately 6.40 mm/sec. for disk I and 7.60 mm/sec. for disk II respectively. The experimentation installed, a trial mass of pre-determined weight was introduced for testing 3 cases of the trial mass and phase which was the functional validation of VBD and tests precision of correction algorithm, shown in table 3. The unbalance thus created to produce a mechanical vibration in the HCM's structure. Vibration signal of the faulty unbalanced machine; the peak amplitude and phase signal increased, shown in table 4. This unbalance produced an increase in the HCM's vibration level. In this unbalanced condition, the phase between the signals of the Hall-effect sensor and the accelerometer was calculated to be approximately from equation (3).

Table 3. Testing condition of trial.

	Disk I		Disk II	
	M_T (grams)	θ_T (degree)	M_T (grams)	θ_T (degree)
Case I	0.53	0	0.53	0
Case II	0.79	0	0.71	170
Case III	0.21	120	0.35	300

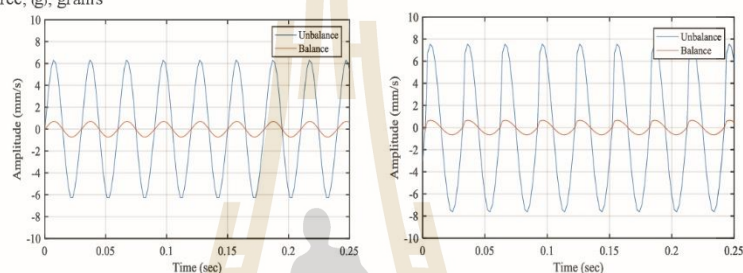
The proposed unbalance correction algorithm was implemented after identifying the correlation between unbalance and increase in the vibration amplitude. In addition, additional experiments were conducted for validating the proposed procedure by proving its effectiveness in balancing the system. Table 4 shows six experiments in which different disks were used as unbalance masses and trial masses.

The correction process included inserting a 0.42 g correction mass at 77° (as calculated by the system) for disk I and 0.48 g correction mass at 87° for disk II, respectively. The results of this are shown in figure 6 (blue line) which the A_B magnitude after balancing is reduced more than 80% both two disks.

Table 4. Experimentally obtained correction balance data.

	A_0 (mm/sec)	ϕ_0 ($^\circ$)	Trial in Case						Balancing Parameters		Results	
			A_1 (mm/sec)			ϕ_1 ($^\circ$)			M_B (g)	θ_B ($^\circ$)	A_B (mm/sec)	% Reduce
			I	II	III	I	II	III				
Disk I	6.40	255	9.10	14.70	4.80	315	334	186	0.42	77	0.70	89.06
Disk II	7.60	261	11.01	14.20	12.56	311	200	272	0.48	87	0.65	91.45

$^\circ$), degree, (g); grams

**Figure 6.** Behaviours of amplitude balancing, left (Disk I) and right (Disk II).

6. Conclusions

The implementation of an experimental setup with signal processor management that combines accelerometer vibration and Hall-effect sensor analyses yielded satisfactory results when detecting unbalance and manufacturing fault identification in disk of HCM. Practical experiments using the VBD showed that can effectively balance the disk. The effectiveness for reducing vibration caused by unbalance is shown through the levels similar to those of the corresponding balanced rotating machine. That was successfully tested and ready for quality control application in manufacturing process.

Acknowledgments

This research was supported by Suranaree University of Technology (SUT), and would like to thankfully acknowledge the research instruments from Suranaree Medical Equipment Company Limited and RAP Enterprise and Services Limited Partnership.

References

- [1] Gitte W 2004 Hematocrit-a review of different analytical method (Denmark: Acutecaretesting)
- [2] Max M S, Albert B S and Eleanor D H 1945 An improved micro hematocrit method **24** 1016
- [3] ISO 10816-1 1995 Mechanical vibration - Evaluation of machine vibration by measurements on non-rotating parts – Part 1: General guidelines (Switzerland: ISO)
- [4] Pawin J and Jiraphon S 2013 Int. J. of Eng. Sci. and Inno. Tech. **2** 254
- [5] Simeonov I, and Stoyanov S 2003 Chem. and Bio. Eng **17** 285
- [6] Åström, K J and Hägglund T 2006 Advanced PID Control (NC: ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park)
- [7] Guilherme K Y, Cesar daC and João Sinohara das S 2016 Case Study in Mech. Sys. and Sign. Proc. **4** 8

ประวัติผู้เขียน

นายศิริพงษ์ ปะวะโก เกิดเมื่อวันที่ 8 กรกฎาคม พ.ศ. 2536 ที่จังหวัดกรุงเทพมหานคร จบการศึกษาระดับประถมศึกษาจากโรงเรียนวัดบุญประดิษฐ์ จังหวัดกรุงเทพมหานคร จบการศึกษาระดับมัธยมศึกษาตอนต้นจากโรงเรียนวัดบุญประดิษฐ์ จังหวัดกรุงเทพมหานคร จบการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนราชวินิตบางแคปานขำ จังหวัดกรุงเทพมหานคร และสำเร็จการศึกษาได้รับปริญญาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมเครื่องกล) หลักสูตร 4 ปี ประจำปีการศึกษา 2558 มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2559 และหลังจากนั้นได้เข้ารับการศึกษต่อในระดับวิศวกรรมศาสตรมหาบัณฑิต (เมคคาทรอนิกส์) ประจำปีการศึกษา 2559 ณ สถาบันเดิม



มหาวิทยาลัยเทคโนโลยีสุรนารี