

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์รักษาสมดุลย์ด้วยตนเอง

Self Balancing Robot



๒๖๖
๒ 62343
๑๐/๒๕

เลขหมู่.....
เลขทะเบียน.....**62343**
วัน,เดือน,ปี...16 ส.ค. 2549

.b.....11621345
.j.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมระบบควบคุม

สาขานแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง หุ่นยนต์รักษาสมดุลด้วยตนเอง

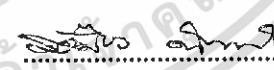
Self Balancing Robot

ผู้จัดทำ นายธวัช อุตัยเจริญพงษ์ 45010346

นายสุนทร แคล้วโยธา 45010206




.....อาจารย์ที่ปรึกษา
(อาจารย์ เกียรติวรรณ ทรงสัตย์)


.....อาจารย์ที่ปรึกษาร่วม
(อาจารย์ สองเมือง นันทขว้าง)

หัวข้อปริญญานิพนธ์	หุ่นยนต์รักษาสมดุทธ์ได้ด้วยตนเอง
นักศึกษา	นายธาวัน อุทัยเจริญพงษ์ นายฐนธร แก้วโยธา
รหัสประจำตัว	45010346 45010206
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมระบบควบคุม
สาขาวิชา	วิศวกรรมเมคคาทรอนิกส์
ปีการศึกษา	2548
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์เกียรติวรรณ ทรงสัถย์

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาระบบรักษาสมดุทธ์ได้ด้วยตนเอง โดยสร้างและออกแบบหุ่นยนต์ที่มีลักษณะการวางตัวบนล้อสองล้อขนานกันในแนวระนาบเดียวกัน ให้สามารถรักษาสมดุทธ์ได้ตลอดเวลาถึงแม้จะได้รับการรบกวนจากภายนอกก็ตาม หุ่นยนต์ทำงานโดยใช้หลักการควบคุมโดยอาศัยสัญญาณป้อนกลับจากอุปกรณ์ตรวจจับคือตัวต้านทานปรับค่าได้ ส่งไปยังไมโครคอนโทรลเลอร์เพื่อทำการคำนวณและส่งสัญญาณออกไปควบคุมการเคลื่อนที่ของหุ่นยนต์ต่อไป

Thesis Title Self Balancing Robot
Student Mr. Tawon Uthaicharoenpong
Mr. Thanathorn Kleawyotha
Student ID. 45010346
45010206
Degree Bachelor of Engineering
Major Mechatronics Engineering
Year 2005
Thesis Advisor Mr.Kerdtiwarn Drongsud

ABSTRACT

This thesis present the study,design and built of self balancing robot with its configuration of coaxial wheels to be able to balance it self even an external force has apply to it .The robots work on basis of feedback control theory , by measure an angle of the robot by variable resistor that act as an sensor in this project and send to microcontroller to calculate and control of posture and motion of robot to keep it stable in an upright position

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี ผู้เขียนขอกราบขอบพระคุณ อาจารย์เกียรติวรณทรหงส์ อาจารย์ที่ปรึกษาเป็นอย่างสูง ที่ได้ให้คำปรึกษาแนวทางในการแก้ปัญหา ความคิดริเริ่ม ตลอดจนการฝึกฝนให้ผู้เขียนมีความสามารถในการทำวิจัยได้อย่างมีประสิทธิภาพ รวมถึงให้ความสนใจใส่ดูแลสอบถามถึงความก้าวหน้าอย่างสม่ำเสมอ ทำให้ผู้เขียนทำงานอย่างมีระบบและใช้เวลาอย่างมีประสิทธิภาพ

ขอขอบพระคุณ อาจารย์ ดอน อิศรากร ที่ได้ให้คำปรึกษาแนะนำวิธีการแก้ปัญหาต่างๆ ที่เกิดขึ้นทั้งทางทฤษฎีและทางปฏิบัติแก่ผู้เขียนเสมอมา ทำให้ผู้เขียนเข้าใจปัญหานั้นๆ

ขอขอบพระคุณ อาจารย์ สองเมือง นันทขว้าง และนายสมรรถชัย ปัญญาที่ได้ช่วยเหลือให้คำปรึกษาและเป็นกำลังใจให้ ผู้เขียนด้วยดีเสมอมา

ขอขอบคุณ ภาควิชาวิศวกรรมระบบควบคุม ที่ได้เอื้อเฟื้ออุปกรณ์และเครื่องมือในการทำวิจัย ตลอดจนอำนวยความสะดวก ให้แก่ผู้เขียนจนสามารถทำวิทยานิพนธ์ฉบับนี้จนเสร็จสิ้น

ขอขอบคุณ โครงการสำนักวิจัยการสื่อสารและเทคโนโลยีสารสนเทศ (ReCCIT) และ ชุมชมอิเล็กทรอนิกส์ ที่ได้ให้การสนับสนุนอุปกรณ์และเครื่องมือในการทำวิจัยครั้งนี้

และท้ายที่สุดขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่รักและเคารพ ที่สนับสนุนให้โอกาสแก่ผู้เขียนได้เล่าเรียนจนถึงวันนี้ ตลอดจนเพื่อนๆ พี่ๆ และน้องๆ ในห้องวิจัยและชุมนุม ทุกคนที่ได้ให้ คำปรึกษา แนะนำ และเป็นกำลังใจให้ด้วยดีเสมอมาจนสามารถนำสิ่งต่างๆ ที่ได้รับมานั้นจัดทำปริญญานิพนธ์ฉบับนี้ให้เสร็จสิ้นสมบูรณ์

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้เขียนขอมอบแด่ผู้มีพระคุณทุกท่าน

ธาวัน อุทัยเจริญพงษ์
ฐนธร แคล้วโยธา

สารบัญ

	หน้า
บทคัดย่อ	I
กิตติกรรมประกาศ	III
สารบัญรูป	IV
บทที่ 1 บทนำ	1
บทที่ 2 ความรู้พื้นฐาน	2
2.1 ทฤษฎีระบบควบคุมพื้นฐาน	2
2.2 การวิจัยและความคืบหน้าในปัจจุบัน	4
2.2.1 รูปแบบของ Segway	4
2.2.2 รูปแบบของแกรสเซอร์(The Grasser Model)	11
2.2.3 รูปแบบของแบลคเวลล์(The Blackwell Model)	14
2.2.4 รูปแบบของ ชูจลีจจ์ (The Chudleigh Model)	17
2.2.5 รูปแบบของลาสัน (The Larson Model)	18
บทที่ 3 การออกแบบระบบการทำงาน	20
3.1 คุณสมบัติของ dsPIC30F2010	20
3.2 บล็อกไดอะแกรมวงจรการควบคุมการทำงานของหุ่นยนต์รักษาสมดุล	22
3.2.1 หน่วยการแปลงสัญญาณอนาล็อกเป็นดิจิทัล	23
3.2.2 หน่วยกำเนิดสัญญาณการมอดูเลตความกว้างพัลส์	25
บทที่ 4 โครงสร้างทางกล	30
รูปแบบที่ 1	30
รูปแบบที่ 2	32
รูปแบบที่ 3	33
รูปแบบที่ 4	34
รูปแบบที่ 5	35
บทที่ 5 หลักการในการควบคุมหุ่นยนต์และทฤษฎีของระบบควบคุม	39
5.1 หลักการในการควบคุมหุ่นยนต์	39
5.2 ตัวควบคุมแบบสัดส่วน(P Controller)	39
5.3 ตัวควบคุมแบบสัดส่วนบวกอินทิกรัล(PI Controller)	43
5.4 ตัวควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์(PD Controller)	45
5.5 ตัวควบคุมแบบพีไอดี(PID Controller)	46

สารบัญ(ต่อ)

	หน้า
บทที่ 6 การทดลองและผลการทดลอง	47
6.1 การทดลองระหว่างองศาที่เปลี่ยนไปของตัวต้านทานปรับค่าได้ กับแรงดันไฟฟ้า	47
6.2 การทดลองหาความสัมพันธ์ระหว่างองศาที่เปลี่ยนแปลงไปจากสมมูล และแรงดันที่จ่ายให้มอเตอร์	48
6.3 การทดลองหาค่าอัตราขยายที่เหมาะสม	50
6.3.1 การทดลองหาค่าอัตราขยายที่ใช้ในตัวควบคุมแบบพีดี	50
6.3.2 การหาค่าอัตราขยายที่ใช้ในตัวควบคุมแบบพีไอดี	51
บทที่ 7 สรุปและวิเคราะห์ปัญหา	53
7.1 สรุปผลการดำเนินงาน	53
7.2 ปัญหาและแนวทางแก้ไข	53
7.3 แนวทางการพัฒนาในอนาคต	54
เอกสารอ้างอิง	55
ภาคผนวก ก	56
ภาคผนวก ข	69
ภาคผนวก ค	76

สารบัญรูป

รูปที่	หน้า
2.1 แสดง แผนภาพอิสระ ของ Inverted pendulum (Sophia University Japan 2005)	2
2.2 แสดง Free body diagram ที่ถูกทำให้ง่ายขึ้น (University of Newcastle 2005)	3
2.3 แสดง Unit step response	4
2.4 ผลตอบสนองของPID	4
2.5 แสดงSegway i180	5
2.6 แสดงภาพตัดขวางของมอเตอร์ที่ใช้ใน Segway	5
2.7 แสดง ภาพอธิบายความถี่ของเฟืองที่ขบกัน	6
2.8 แสดงล้อและยางของSegway HT	7
2.9 แสดง แบตเตอรี่NI-MH	7
2.10 แสดงตำแหน่งที่วางวงจรรควบคุม	8
2.11 แสดงลักษณะการวางเซ็นเซอร์	8
2.12 แสดงลักษณะการวางเซ็นเซอร์	9
2.13 แสดง ตำแหน่งที่วางเซ็นเซอร์	9
2.14 แสดง ระบบสัญญาณและจอแสดงผลLCD	10
2.15 แสดงภาพกระจายของ Segway HT	11
2.16 แสดง JOE : A Mobile Inverted Pendulum, Grasser et al.(2002)	12
2.17 แสดงแผนภาพอิสระ JOE : A Mobile Inverted Pendulum, Grasser et al.(2002)	13
2.18 แสดงพาหนะของ Blackwell	14
2.19 แสดงระบบการเลี้ยวและอุปกรณ์การปรับอัตราขยาย	15
2.20 แสดงแพ็คแบตเตอรี่ที่อยู่ใต้ขึ้น	15
2.21 แสดงพาหนะของ Blackwellแบบที่2	16
2.22 แสดง The Chudleigh Model	17
2.23 แสดง The Chudleigh Model ในแบบ CAD	17
2.24 แสดง เซ็นเซอร์วัดระยะทางและเซ็นเซอร์อินฟราเรด	18

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.25 แสดงรูปแบบของลาตัน	19
3.1 แสดงบอร์ดทดลองเบอร์ ET- dsPIC30F2010 TRAINING KIT V1.0	21
3.2 แสดงวงจรส่วนไมโครคอนโทรลเลอร์และส่วนขับเคลื่อนมอเตอร์	22
3.3 แสดง บล็อกไดอะแกรมวงจรการควบคุมการทำงาน ของหุ่นยนต์รักษาสมดุลย์	22
3.4 แผนผังการทำงานของการใช้ช่องพัคตัวอย่างและ คงค่า2ช่องในรูปแบบของการรับสัญญาณแบบติดต่อกัน	23
3.5 แสดงแผนผังการต่อเชื่อมเพื่อให้ค่าอัตราการหมุนมากที่สุด	24
3.7 แสดงแผนผังการทำงานของ โมดูลMCPWM	25
3.8 แสดงรูปบอร์ดขับเคลื่อนมอเตอร์DMD2A	28
4.1 หุ่นยนต์รูปแบบที่1	31
4.2 แสดงเซอร์โวมอเตอร์	31
4.3 แสดงล้อที่ตัดแปลงจากแผ่นCD	31
4.4 แสดงหน้าต่างการทำงาน	31
4.5แสดงแผ่นติดตั้งเซอร์โวมอเตอร์	31
4.6 แสดงรูปของหุ่นยนต์รูปแบบที่2	32
4.7 แสดงมอเตอร์ไฟตรง	32
4.8แสดงเป็นมอเตอร์	32
4.9 แสดงหุ่นยนต์รูปแบบที่3	33
4.10 แสดงการต่อเพลาร่วม	33
4.11 แสดงชุดเกียร์ที่ทำงานพร้อมมอเตอร์	33
4.12 แสดงหุ่นยนต์รูปแบบที่4	34
4.13 แสดงตัวต้านทานปรับค่าได้ยี่ห้อเซินเซอร์ ติดกับเสาที่ยื่นออกมา	34
4.14 แสดงส่วนประกอบของชุด	34
4.15 แสดงหุ่นยนต์รูปแบบที่5	35

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.16 แสดงมอเตอร์เฟลาถู	36
4.17 แสดงล้อ	36
4.18 แสดงตัวยึดล้อ	36
4.19 แสดงแป้นช็อคมอเตอร์	36
4.20 แสดงหุ่นยนต์รูปแบบที่5ด้วยการกระจายออก	36
4.21 แสดงภาพจริงด้านหน้าของหุ่นยนต์	37
4.22 แสดงภาพจริงของด้านหลังหุ่นยนต์	37
4.23 แสดงภาพจริงด้านข้างของหุ่นยนต์	38
5.1 แสดงการทำงานโดยรวมของระบบ	39
5.2 แสดงแผนผังการทำงานของตัวควบคุมแบบสัดส่วน	39
5.3 แสดงระบบวงปิดของการใช้งานตัวควบคุมแบบสัดส่วน	40
5.4 แสดงความสัมพันธ์ระหว่างตำแหน่งโพลกับค่าอัตราส่วนการหน่วง	41
5.5 แสดงความสัมพันธ์ระหว่างความหน่วงที่สัมพันธ์กับเปอร์เซ็นต์ค่าพุ่งเกิน	42
5.6 แสดงตัวอย่างเส้นทางรอก	43
5.7 แสดงแผนผังการทำงานของตัวควบคุมแบบสัดส่วนบวกอินทิกรัล	44
5.8 แสดงเส้นทางการเดินรอกของระบบtype0 ซึ่งหลังจากการเพิ่ม โพลแล้วได้กลายเป็นระบบแบบtype1	44
5.9แสดงเส้นทางการเดินรอกหลังจากมีโพลที่จุดกำเนิดและซีโรที่ $s = -0.1$	44
6.1 แสดงวงจรถอดลอง	47
6.2แสดงกราฟระหว่างแรงดันเอาต์พุตและมุมของตัวต้านทานปรับค่าได้	48
6.3แสดงกราฟระหว่างองศาและแรงดันที่ได้จากการทดลอง	49
6.4 แสดงรูปการทดลอง	50

บทที่ 1

บทนำ

ดีน คาเมน(Dean Kamen) นักประดิษฐ์ และผู้ประกอบการ ได้ริเริ่มความคิดที่จะสร้างระบบขนส่งส่วนบุคคลแบบใหม่ในช่วงกลางของปี1990 ทุกวันนี้ Segway Human Transporter (HT) การประดิษฐ์ของเขา เป็นที่นิยมและมีขายอยู่ทั่วโลก Segway HT เป็นพาหนะที่มีสองล้อวางขนานในแนวเดียวกันแต่ละล้อ ขับเคลื่อนแยกกันอย่างอิสระ โดยจะมีตัวควบคุมที่ทำให้มันสมดุล ทั้งเมื่อมีหรือไม่มีผู้ขับขี่ การควบคุมการสมดุลนั้นทำโดยการตรวจสอบสัญญาณป้อนกลับ จากเซ็นเซอร์วัดมุมเอียง และ ไจโรสโคป มายังตัวควบคุมแบบปรีภูมิสถานะทำให้ระบบมีความคงทน และมีประสิทธิภาพ มากพอที่จะยอมรับผู้ขับขี่ที่มีน้ำหนักและรูปแบบการขับขี่ที่แตกต่างกันได้

วัตถุประสงค์ของโครงการนี้คือ วิจัย ศึกษาหลักการออกแบบ, หลักการทำงานของหุ่นยนต์รักษาสมดุลได้ด้วยตัวเอง และสร้างหุ่นยนต์ที่เลียนแบบการรักษาสมดุลได้ด้วยตัวมันเองบนล้อที่ขนานอยู่บนแกนเดียวกัน และมีหลักการทำงานที่คล้ายคลึงกับ Segway HT และยังสามารถสร้างได้ง่ายใช้ชิ้นส่วนที่หาได้โดยทั่วไปอีกด้วย

รายงานนี้จะเริ่มต้นด้วยการปูความรู้พื้นฐานที่เกี่ยวกับโครงการในบทที่2 ตามด้วยข้อมูลที่เกี่ยวข้องเพื่อที่จะนำไปเป็นตัวกำหนดแนวทางในการออกแบบระบบและการทำงานในบทที่3 ศึกษาโครงสร้างทางกลในบทที่4 ทฤษฎีระบบควบคุมในบทที่5 การทดลองบทที่6 และบทที่7จะเป็นบทสรุปและการวิเคราะห์ปัญหา

บทที่ 2

ความรู้พื้นฐาน

คณะของผู้จัดทำโครงการนี้ได้เริ่มต้นการทำโครงการโดยอ่านและศึกษา บทความที่เกี่ยวข้องกับโครงการ เช่นบทความเรื่อง Mobile inverted pendulum, the Segway HT, the Segway HT clones . self-balancing robot ซึ่งการศึกษาแบ่งได้เป็นสองส่วนคือ ส่วนตัวควบคุมที่จะใช้ในการรักษาสมดุล และส่วนที่เป็นโครงสร้างของ Segway HT บทที่ 2.1 จะกล่าวถึงพื้นฐานที่สำคัญของระบบควบคุมที่จะใช้ในการรักษาสมดุล โดยเฉพาะอย่างยิ่ง mobile inverted pendulum บทที่ 2.2 จะเป็นการวิเคราะห์บทความที่ได้ศึกษามา

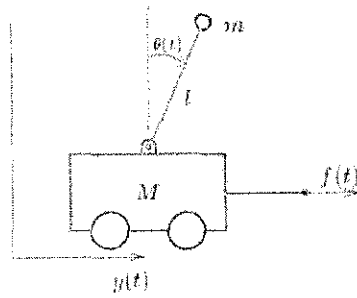
2.1 ทฤษฎีระบบควบคุมพื้นฐาน

การพัฒนาาระบบควบคุมอัตโนมัติเริ่มต้นในปี 1796 โดย เจมส์ วัตต์ (James Watt) เมื่อเขาได้ประดิษฐ์ flyball governor (Dorf & Bishop 2001) ของเขาขึ้น จากนั้นเป็นต้นมา ระบบต่างๆ ก็ได้เกิดขึ้นเพื่อที่ต่อมาเป็นพื้นฐานของระบบควบคุมเบื้องต้น มีหลายระบบที่ถูกหยิบยกขึ้นมา เพื่อแสดงแนวคิดของระบบควบคุมแบบอัตโนมัติ และหนึ่งในระบบเหล่านั้นก็คือ Inverted pendulum นั่นเอง

เริ่มจากการพยายามรักษาสมดุลของมวลที่ยึดติดกับปลายอีกด้านหนึ่งของแท่งไม้ที่อยู่บนมือของคุณ ดังรูปที่ 2.1 ได้ถูกทำให้อยู่ในรูปของรถที่อยู่บนรางดังรูป 2.2 เพื่อที่จะได้ตรวจสอบและนำเสนอออกมาในรูปของสมการการเคลื่อนที่ ที่เหมาะสมได้ง่ายและแม่นยำยิ่งขึ้น



รูปที่ 2.1 แสดง ภาพอิสรระ ของ Inverted pendulum (Sophia University Japan 2005)



รูปที่ 2.2 แสดง Free body diagram ที่ถูกทำให้ง่ายขึ้น (University of Newcastle 2005)

ในการที่จะเริ่มทำความเข้าใจว่า self-balancing robot จะทำงานอย่างไร จำเป็นต้องมีความรู้จักกับ mobile inverted pendulum ก่อน mobile inverted pendulum อย่างง่ายประกอบด้วยรถรางที่มีมวลคงที่และเสาที่มีมวลวางอยู่ที่ปลายด้านหนึ่ง เสานี้จะยึดติดอยู่กับราง โดยจุดหมุนที่อยู่เหนือแกน และสามารถเคลื่อนที่ได้อย่างอิสระในระนาบเดียวเท่านั้น แรงภายนอกที่มากระทำจะมีเพียงแรงในแนวราบที่กระทำกับรถรางเท่านั้น ซึ่งก็จะทำให้เกิดการเคลื่อนที่ของรถรางและเสา สมการการเคลื่อนที่ของระบบนี้ ได้มาจากการพิสูจน์ของสมมุติฐานแรงในแกนแนวนอน และ สมมุติฐานของแรงบิดที่เกิดขึ้นรอบจุดหมุนของเสา เพื่อที่จะทำให้การวิเคราะห์ระบบง่ายขึ้นเราก็จะทำให้ระบบเป็นเชิงเส้น โดยกำหนดให้อยู่ในลักษณะที่เสายู่ในสมมุติฐานแล้ว ทำให้พิจารณาได้ว่ามุมที่เสาเคลื่อนที่มีค่าน้อยพอที่จะทำให้มุมที่หมุนไปมีค่าน้อยมาก ซึ่งจะได้สมการดังสมการที่ 2.1 และ 2.2

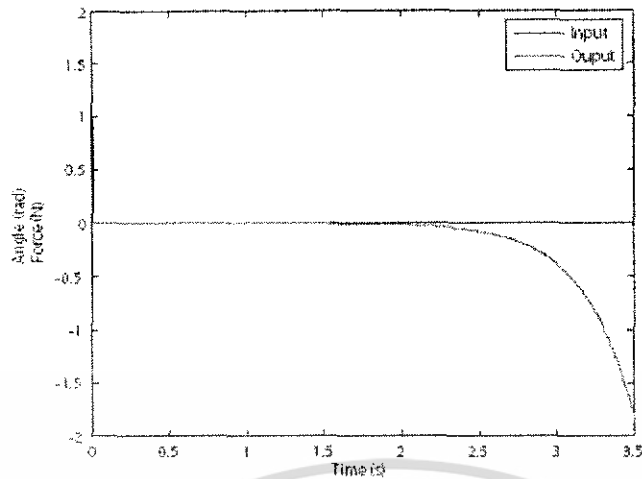
$$My + mL\ddot{\theta} - U(t) = 0 \quad (2.1)$$

$$mLy + mL^2\ddot{\theta} - mLg\theta = 0 \quad (2.2)$$

$$G(s) = \frac{-1/LM}{s^2 - g/L} \quad (2.3)$$

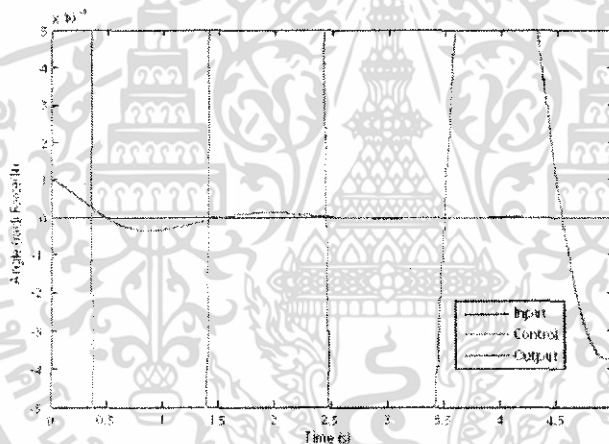
สมการถูกนำมารวมกันเพื่อที่จะกำจัดส่วนของความเร่งในแนวนอน และถูกนำมาเรียบเรียงใหม่ เพื่อให้ง่ายในการคำนวณจะสมมุติให้โดยสมมุติว่ามวลของรถรางมีค่ามากกว่ามวลที่ติดอยู่ที่ปลายมากๆ เราก็จะได้สมการถ่ายโอนที่แสดงถึงแรงที่กระทำต่อความเร่งเชิงมุมของเสาดังแสดงในสมการที่ 2.3

รูปที่ 2.3 แสดงผลตอบสนองต่อสัญญาณอิมพัลส์ มุมของเสามีค่ามากขึ้นเป็นแบบเอกซ์โปเนนเชียลเมื่อถูกแรงดึงคูดของ โลกมากระทำ และจะสมมุติว่าเป็นผลตอบสนองแบบเดียวกับ self-balancing robot เมื่อไม่มีแรงภายนอกมากระทำ



รูปที่ 2.3 แสดง Unit step response

เมื่อใช้ตัวควบคุมที่เหมาะสมเราก็จะสามารถใส่แรงให้รถรางเพื่อที่จะชดเชยแรงภายนอกที่มากระทำ เพื่อที่จะสามารถรักษาสถิตยของเสาได้ ผลตอบสนองของระบบเมื่อใช้ตัวควบคุมแบบ PID แสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 ผลตอบสนองของPID

2.2 การวิจัยและความคืบหน้าในปัจจุบัน

ในส่วนนี้จะอธิบายถึงความพยายามที่จะสร้างอุปกรณ์ที่สามารถรักษาสถิตยได้ด้วยตนเองแบบต่างๆที่เคยมีมาก่อนหน้านี้ เพื่อที่จะใช้เป็นพื้นฐานการออกแบบของ Self-balancing robot

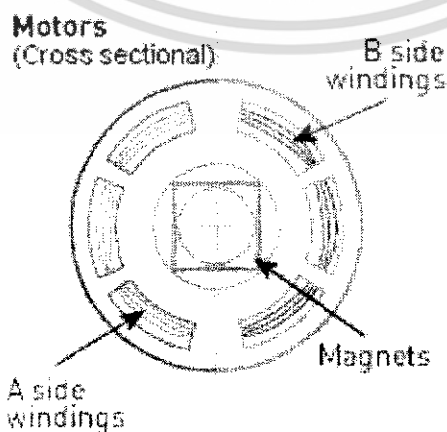
2.2.1 รูปแบบของ Segway

Segway ดังแสดงในรูปที่ 2.5 ได้ศึกษาและวิเคราะห์จากแหล่งข้อมูลสองแหล่ง คือ จากเว็บไซต์ของ Segway เอง (Segway Inc 2005) และ เว็บไซต์ How Stuff Works ที่อธิบายถึงการทำงานของ Segway (How Stuff Works Inc 2005)

Segway HT เป็นพาหนะที่รักษาสมดุลของตัวเองได้เพียงชนิดเดียวที่มีขายอย่างเป็นทางการทั่วโลกในขณะนี้ อ้างอิงจาก How Stuff Works Inc(2005) Segway HT ได้ถือกำเนิดขึ้นจากผู้ประกอบการและวิศวกร คีน คามน ได้พยายามเคลื่อนที่ไปข้างหลัง เพื่อที่จะพยายามรักษาสมดุลของเขาขณะล้มลง ขณะที่กำลังเดินออกจากที่อาบน้ำของเขา แม้ว่าเขาจะล้มลงในภายหลังก็ตาม เขาก็เริ่มคิดว่าหากร่างกายของมนุษย์สามารถตอบสนองต่อการเสียสมดุลได้เร็วขนาดนั้น เครื่องจักรก็น่าจะทำเช่นนั้นได้เหมือนกัน หลังจากจดสิทธิบัตรไต่เทียมแบบพกพาของเขาแล้ว คามนก็ระดมสมองคิดถึงเรื่องเครื่องจักรที่จะสามารถรักษาสมดุลของมันได้เป็นต้นมา เป็นที่รู้กันน้อยมากว่าการเติมฟันกับเครื่องจักรที่จะสามารถรักษาสมดุลได้เป็นครั้งแรกนั้นได้ผลลัพธ์ออกมาเป็น iBOT รถเงินที่มีมอเตอร์ที่สามารถปั่นปายบันไดได้โดยใช้ล้อที่วางขนานกันสองชุดสลับกัน โดยรักษาสมดุลโดยล้อชุดเดียวในขณะที่ล้ออีกชุดหนึ่งกำลังไต่บันไดขึ้นไป ในขณะที่กำลังพัฒนา iBOT คามนได้สังเกตเห็นความเป็นไปได้ว่า จะใช้หลักการรักษาสมดุลนี้เพื่อสร้างยานพาหนะที่ใช้ในชีวิตประจำวันได้ Segway HT ได้ถูกพัฒนาโดยบริษัทวิจัยของเขาเอง (DEKA Research and Development Corporation) ในปี2001 Segwayได้ปรากฏต่อสายตาคนภายนอกและพร้อมที่จะขายในปี2002



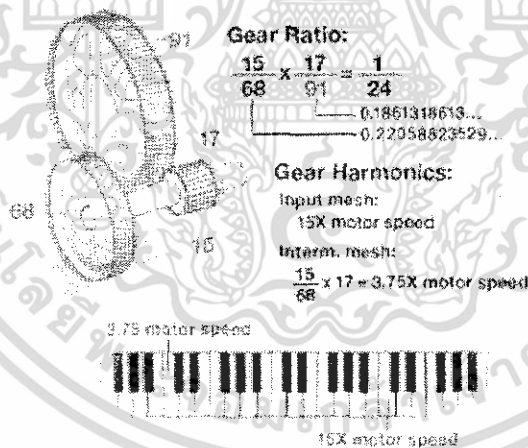
รูปที่ 2.5 แสดงSegway i180



รูปที่ 2.6 แสดงภาพตัดขวางของมอเตอร์ที่ใช้ใน Segway

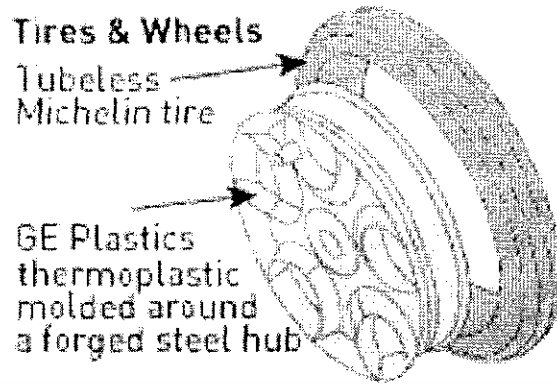
How Stuff Works Inc(2005)ได้อธิบายว่า มอเตอร์ที่ใช้ในSegway มีกำลัง1.88กิโลวัตต์ 8000 รอบต่อนาที เป็นมอเตอร์แบบ Brushless servo โดยใช้แม่เหล็กถาวร neodymium-iron-boron เป็นส่วนประกอบ ขับเคลื่อนโดย ใช้ฟลักซ์ออฟเฟคทรานซิสเตอร์ กำลังสูงและแรงดันสูง 12ชุด อ้างอิงจาก Segway Inc(2005)ว่ามอเตอร์ชนิดนี้เป็นมอเตอร์ที่มีกำลังมากที่สุดในขนาดเดียวกันที่เคยมีการผลิตเป็นจำนวนมากออกมา มันมีแรงบิดสูง ต้องการการดูแลรักษาน้อย และการใช้พลังงานไฟฟ้าของมอเตอร์แต่ละตัวจะถูกตรวจสอบโดยวงจรของมันเอง มอเตอร์สร้างขึ้นจากชุดของขดลวดสองชุดที่แยกออกจากกันดังแสดงในรูปที่2.6 เมื่อSegway HT ทำงานตามปกติ ขดลวดทั้งสองจะทำงานเพียงอย่างละครั้งหนึ่งของกำลังทั้งหมดเท่าๆกัน เมื่อขดลวดอันใดอันหนึ่งเกิดเสียหาย Segway HT จะยกเลิกการทำงานของขดลวดชนิดนั้นทันทีและใช้เพียงขดลวดที่เหลืออีกอันหนึ่งต่อไปเพื่อให้เกิดการหยุดที่ปลอดภัย

How Stuff Works Inc (2005) อธิบายว่า ระบบทดแรงสองขั้นมีอัตราทดเท่ากับ24:1 เพื่อให้ได้ประสิทธิภาพของแรงบิดสูงสุด และใช้เฟืองแบบ Helical เพราะว่ามันสามารถกำจัดเสียงรบกวนลงได้อย่างมาก ทีมงานที่พัฒนา Segway HT ได้ออกแบบให้ความถี่ที่ได้จากเฟืองสองตัวขบกันมีเสียงห่างกันเท่ากับเสียงสองคู่แปดพอดี ดังแสดงในรูปที่2.7 เฟืองนี้ยังถูกออกแบบให้มีอัตราส่วนของเฟืองไม่เป็นจำนวนเต็ม เพื่อที่เฟืองจะไม่ขบเข้าอยู่จุดเดิมทำให้มีการสึกกร่อนของฟันเฟืองน้อยลง



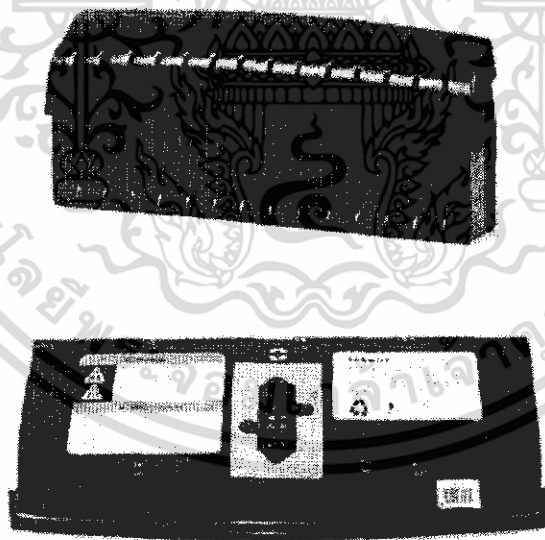
รูปที่ 2.7 แสดง ภาพอธิบายความถี่ของเฟืองที่ขบกัน

จากข้อมูลของ How Stuff Works Inc(2005) ล้อของมันประกอบด้วยคัมล้อที่ทำด้วยเหล็กหล่อ และขอบล้อทำด้วยเทอร์โมพลาสติกคุณภาพสูงเสริมใยแก้วดังแสดงในรูปที่2.8 ล้อแต่ละข้างจะยึดอยู่กับแกนเพนบาขับด้วยน็อตเพียงตัวเดียว ยางจากมิชลิน มีส่วนประกอบของซิลิกาทำให้มีแรงยึดเกาะที่ดีแม้บนพื้นผิวที่เปียก ระบบควบคุมการเลี้ยวบังคับ โดยที่มือจับที่อยู่ด้านซ้ายของที่จับ เพราะว่า Segway HT มีเพียงสองล้อมันจึงสามารถบังคับเลี้ยวโดยรัศมีเป็นศูนย์ได้ ล้อทั้งสองเคลื่อนที่ในทิศตรงกันข้ามในขณะที่ระบบควบคุมของมันยังคงอยู่ในสมดุลย์



รูปที่ 2.8 แสดงล้อและยางของSegway HT

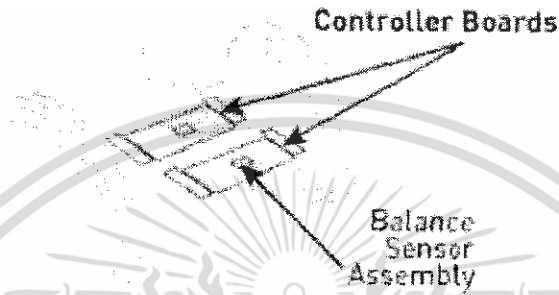
How Stuff Works Inc(2005)กล่าวถึงแบตเตอรี่ที่อัดประจุใหม่ได้สองก้อนที่เป็นแหล่งพลังงานให้กับ Segway HTแสดงในรูป2.9 ในรูปแบบดั้งเดิมจะใช้แบตเตอรี่นิกเกิล-แคดเมียม แต่ในรุ่นใหม่ๆจะเปลี่ยนเป็น นิกเกิลเมทัลไฮดราย หรือ แบตเตอรี่ลิเทียมไอออน92เซลล์ ทั้งสองอย่างได้ออกแบบให้มีแรงดันไฟตรง72โวลต์(Segway Inc 2005) แบตเตอรี่จะถูกตรวจสอบการใช้พลังงานจากวงจรและมิวจรเฉพาะที่สามารถทำให้อัดประจุใหม่ได้จากแรงดันไฟหลักทั่วไป ชนิดของแบตเตอรี่มีผลต่อระยะทางของ Segway มันจะวิ่งได้19กิโลเมตรจากแบตเตอรี่นิกเกิลเมทัลไฮดราย และ39กิโลเมตรเมื่อใช้แบตเตอรี่ลิเทียมไอออน



รูปที่ 2.9 แสดง แบตเตอรี่NI-MH

How Stuff Works Inc(2005) แสดงถึงการควบคุมและการประมวลผลของมันว่าประกอบด้วยวงจรสองส่วน คิดตั้งอยู่ในตัวถังของมันดังแสดงในรูปที่2.10 วงจรทั้งสองจะทำงานแยกออกจากกันอย่างอิสระ หากว่ามีปัญหาเกิดขึ้นวงจรใดวงจรหนึ่งก็จะสามารถควบคุมให้หยุดได้อย่างปลอดภัย วงจรควบคุมทำการตรวจสอบสัญญาณจากเซ็นเซอร์ที่ความถี่100เฮิร์ตซ์ และสัญญาณควบคุมที่ส่งไปที่มอเตอร์มีความถี่1000เฮิร์ตซ์ โดยแต่ละวงจรจะควบคุมขดลวดใน

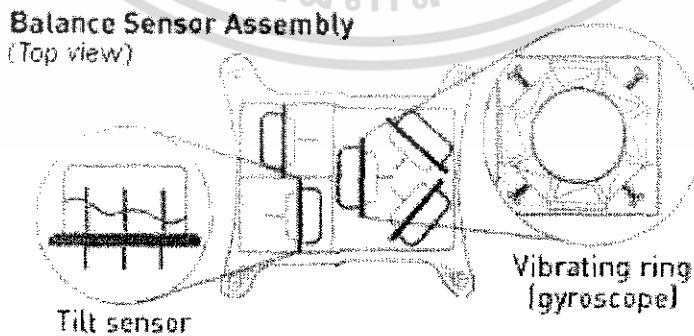
มอเตอร์อย่างละชุด Segway HT ใช้หน่วยประมวลผลสัญญาณดิจิทัล(DSP)ของ Texas Instruments เบอร์ TM320LF2406A ที่ทำงานที่ 40ล้านคำสั่งต่อวินาที มีหน่วยความจำแบบแฟลช32กิโลไบต์ และส่วนต่อเพิ่มอื่นๆอีกมาก ไม่มีบทความที่กล่าวถึงว่าSegway มีระบบการทรงตัวอย่างไร เพราะว่า เป็นข้อมูลที่มีการจดสิทธิบัตรแล้ว อย่างไรก็ตามก็มีการเปิดเผยว่า Segway HT ใช้ หน่วยประมวลผลสัญญาณดิจิทัล(DSP) “เพื่อควบคุมมอเตอร์และการคำนวณการทรงตัวแบบดิจิทัลวงปิด” (Segway Inc 2005)



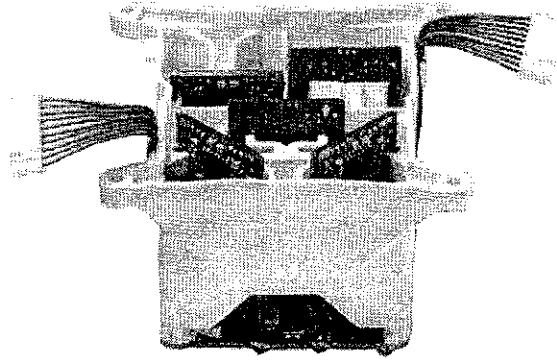
รูปที่ 2.10 แสดงตำแหน่งที่วางวงจรควบคุม

Segway ใช้ไจโรสโคปแบบวงแหวนสี่นิ้ว และเซ็นเซอร์วัดมุมเอียงแบบทำงานโดยของเหลว 2ตัว บนบอร์ดรักษาสมดุลย์ แสดงในรูปที่2.11 และ2.12 เพื่อสามารถทำให้มันตั้งตรงอยู่ได้ (How Stuff Works Inc 2005) ไจโรสโคป ใช้หลักของปรากฏการณ์โคริโอลิส(Coriolis)เพื่อที่จะตรวจวัดความเร็วเชิงมุม แหวนเล็กๆเหล่านี้จะสั่นด้วยแรงกลทางไฟฟ้า ในวิธีที่เมื่อมีการหมุนเกิดขึ้นจะทำให้มีแรงเกิดขึ้นมาและสามารถจะตรวจจับได้โดยอุปกรณ์ตรวจวัดภายใน จากข้อมูลของ Segway Inc (2005) ไจโรสโคปเพียง3ตัวก็เพียงพอต่อการทำงานแล้ว (ไจโรสโคปหนึ่งตัวสำหรับการเคลื่อนที่ในแต่ละแกน) เซ็นเซอร์ที่เหลือมีไว้เพื่อสำรองเท่านั้น เซ็นเซอร์รักษาสมดุลย์จะติดตั้งอยู่ตรงกลางระหว่างแกนดังแสดงในรูป2.13

Segway มีเซ็นเซอร์วัดน้ำหนักรวมอยู่4ตัวติดตั้งอยู่บนแท่นเพื่อที่จะสามารถบอกได้ว่าคนขับกำลังเหยียบลงบนแท่นแล้ว



รูปที่ 2.11 แสดงลักษณะการวางเซ็นเซอร์

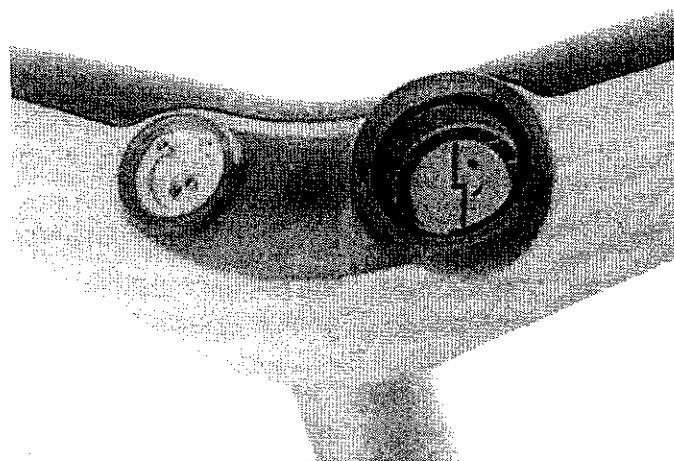


รูปที่ 2.12 แสดงลักษณะการวางเซ็นเซอร์

Segway Inc (2005) ได้อธิบายว่า Segway HT ใช้ระบบกฎเชิงตรรกศาสตร์ แสดงในรูป 2.14 เก็บข้อมูลเข้ารหัส 64 บิต Segway จะไม่เริ่มการทำงานจนกว่าจะได้รับรหัสที่ถูกต้อง กฎเชิงตรรกศาสตร์สามารถเก็บค่าที่กำหนดของพารามิเตอร์ เช่น ความเร็วสูงสุดหรืออัตราการใช้พลังงานเอาไว้ Segway HT มีจอแสดงผลแบบจอผลึกเหลวเพื่อแสดงสถานะของแบตเตอรี่และสถานะทั่วไปของระบบ (จาก How Stuff Works Inc 2005)

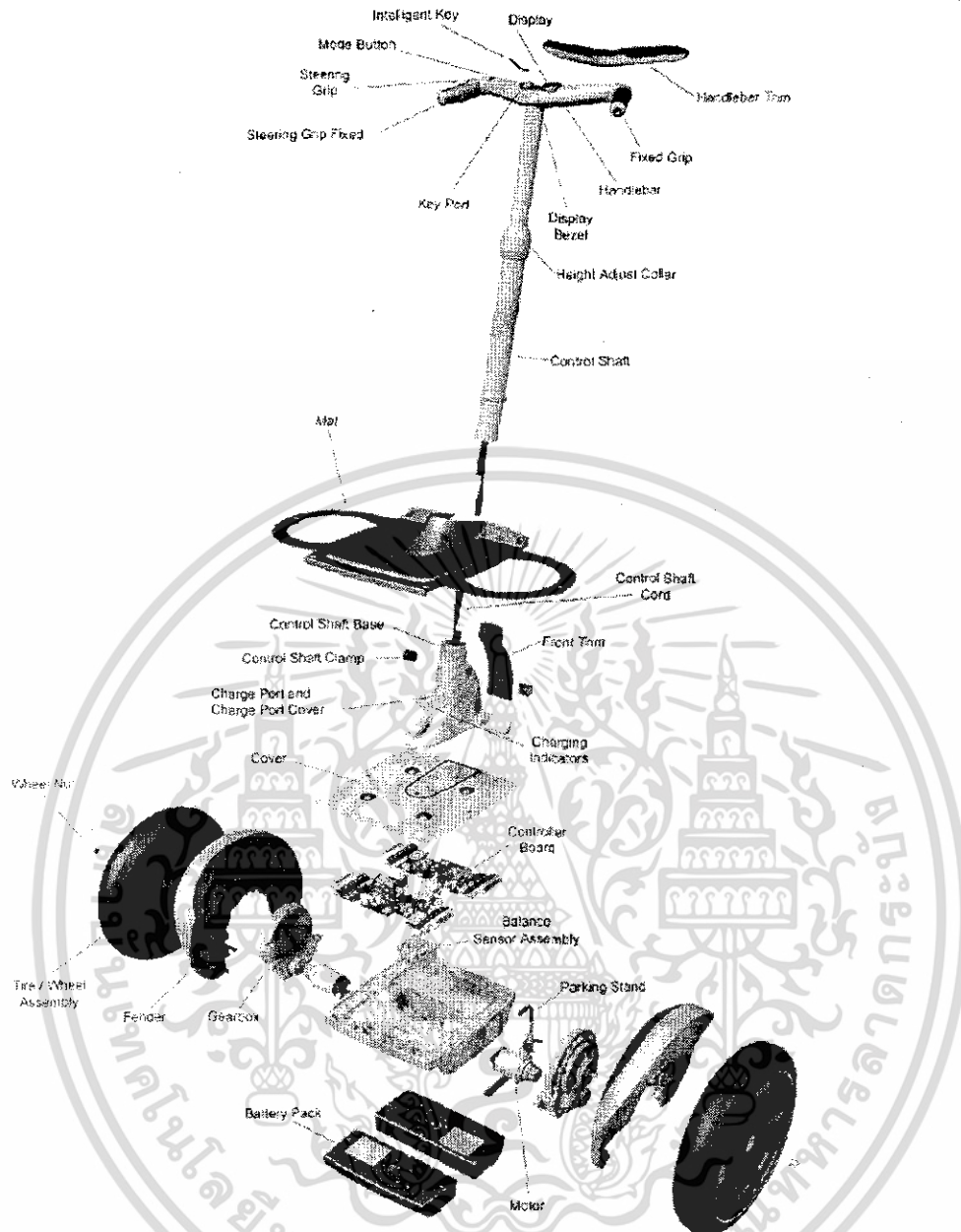
อุปกรณ์อิเล็กทรอนิกส์ที่ละเอียดอ่อนของ Segway HT นั้นจัดเก็บอยู่ในตัวถังอลูมิเนียมหล่อและตกแต่งด้วยพลาสติก Segway Inc (2005) ให้ข้อมูลว่ามันสามารถรับแรงได้ถึง 7 ตัน การตกแต่งทำให้ดูเพรียวและป้องกันรอยขีดข่วน

รูปที่ 2.13 แสดง ตำแหน่งที่วางเซ็นเซอร์



รูปที่ 2.14 แสดงระบบกัญแจและจอแสดงผลLCD

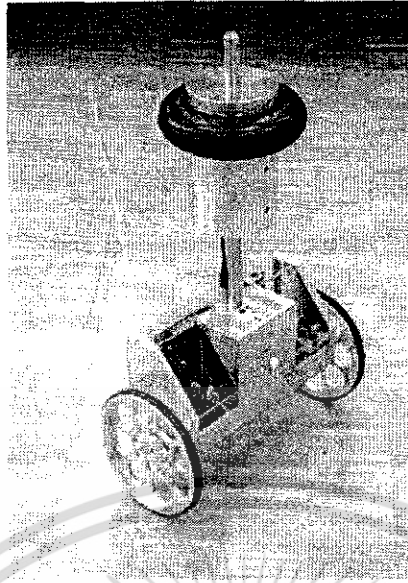




รูปที่ 2.15 แสดงภาพกระจายของ Segway HT

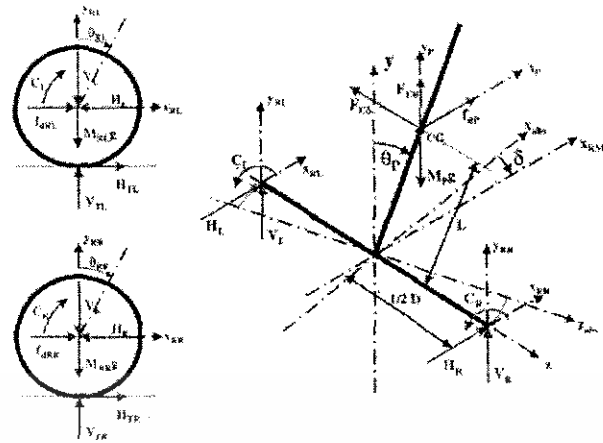
2.2.2 รูปแบบของแกรสเซอร์(The Grasser Model)

ในบทความ JOE : A Mobile Inverted Pendulum, Grasser et al.(2002) นำเสนอบทความของโครงการที่ต้องการจะสร้างพาหนะที่สามารถรักษาสมดุลของผู้ขับขี่บนล้อ 2 ล้อที่อยู่บนแกนเดียวกัน โดยใช้ระบบการควบคุมอัตโนมัติ JOE : A Mobile Inverted Pendulum แสดงในรูปที่ 2.16



รูปที่ 2.16 แสดง JOE : A Mobile Inverted Pendulum, Grasser et al.(2002)

ต้นแบบย่อยส่วนได้ออกแบบให้รองรับค้อนน้ำหนักเพื่อที่จะใช้เลียนแบบผู้ขับแทนที่จะใช้ผู้ขับจริงเนื่องจากเหตุผล 2 ข้อคือ “เพื่อที่จะลดราคาและลดอันตรายของผู้ทดลองขับ”(Grasser et al.2002) และข้อได้เปรียบอีกอย่างของแบบย่อยส่วนก็คือความแน่นอนของสถานะ และสถานะก็สามารถตรวจวัดได้ง่ายกว่าและไม่มี การเปลี่ยนแปลง แต่ก็มีข้อเสียเปรียบของการใช้ต้นแบบย่อยส่วนอยู่เหมือนกัน โดยเฉพาะต้นแบบย่อยส่วนไม่สามารถเลียนแบบโครงการขนาดจริงในขั้นสุดท้ายได้ทั้งหมด แม้ว่าจะสามารถตรวจวัดสถานะได้ง่ายขึ้น แต่มันก็ขาดตัวแปรเนื่องจากน้ำหนักของผู้ขับที่จะต้องนำมาพิจารณาด้วย จึงไม่สามารถพิสูจน์ได้ว่าระบบจะมีการทำงานที่แน่นอนหรือไม่ และมันยังไม่พิจารณาถึงข้อเท็จจริงที่ว่ามนุษย์นั้นไม่ได้อยู่นิ่งเฉยหากแต่มีการเคลื่อนที่ของน้ำหนักในขณะที่ขยับเท้าอยู่บนแท่น หรือว่าการเอียงตัวขณะที่กำลังเข้าโค้ง โครงการนี้ใช้แบบจำลองแบบปริภูมิสถานะ โดยเฉพาะเมื่อต้องการควบคุมให้ได้ความเร็วเชิงเส้นที่ต้องการรวมและถึงอัตราการใช้ด้วย แบบจำลองยังรวมถึงการลดสัญญาณรบกวนสำหรับล้อทั้งสองข้างและสัญญาณรบกวนจากแรงภายนอกและจากน้ำหนักของผู้ขับด้วย จะเห็นได้ว่าแบบจำลองเป็นจุดแข็งของโครงการนี้ และมันสามารถทำให้ผู้แต่งพัฒนาแบบจำลองที่มีความแม่นยำสูงของระบบดังแสดงในรูป 2.17



รูปที่ 2.17 แสดงแผนภาพอิสระ JOE : A Mobile Inverted Pendulum, Grasser et al.(2002)

การพัฒนาาระบบควบคุมจะถูกกล่าวถึงในหัวข้อต่อไป การควบคุมของแบบจำลองเกิดขึ้นจากตัวแปรสถานะ 2 ตัวแยกจากกันสำหรับ pitch และ yaw ตามลำดับ นำมันมารวมกันโดยให้ตัวแปรน้ำหนักที่เหมาะสมเพื่อที่จะให้แรงบิดและแรงดันให้แก่มอเตอร์ ซึ่งเป็นข้อได้เปรียบของแบบจำลองนี้เพราะว่าเราสามารถควบคุมการ pitch และ yaw แยกจากกันโดยอิสระ เมื่อพยายามที่จะเข้าถึงความเร็วเชิงเส้นที่ต้องการรวมถึงความเร็วในการเลี้ยวที่ต้องการ หรือว่าทั้งสองอย่างพร้อมกันก็ได้ สาเหตุสำคัญที่แยกระบบออกจากกันก็เพราะว่าจะทำให้แก้ปัญหาได้แยกกันสองส่วนแทนที่จะแก้ปัญหาในระบบเดียวกันที่มีความเกี่ยวพันกันอย่างซับซ้อน เหตุผลอีกข้อหนึ่งก็คือ จากข้อเท็จจริงที่ว่า มีสัญญาณออกเพียงแค่อ่างเดียวคือสัญญาณที่ส่งไปควบคุมมอเตอร์ และมีสองส่วนที่ผลัดกันทำงานเพื่อควบคุมมอเตอร์ การแยกระบบออกจากกันจะสามารถทำให้เราสามารถกำหนดความสำคัญให้แก่ระบบทั้งสองได้โดยอิสระต่อกัน ในกรณีนี้การรักษาสมดุลมีความสำคัญมากกว่าการบังคับเลี้ยวฉะนั้นเราจึงให้ความสำคัญแก่ระบบรักษาสมดุลมากกว่า

ตัวประมวลผลที่นำมาใช้กับระบบควบคุมในโครงการนี้ประกอบด้วย Shark floating-point DSP, XILINX field-programmable gate array (FPGA) ตัวแปลงสัญญาณดิจิทัลเป็นอนาลอกขนาด 10 บิต 4 ชุด และตัวแปลงสัญญาณอนาลอกเป็นดิจิทัลขนาด 12 บิต 14 ชุด (Grasser et al.2002) นอกจากนี้ในบทความยังมีการอธิบายถึงส่วนประกอบทางฮาร์ดแวร์ที่ใช้ในวงจรการตรวจวัดระบบ, backlash และสัญญาณรบกวนที่เกิดจากชุดเฟืองที่ใช้ ข้อได้เปรียบของการใช้ตัวเข้ารหัส (Encoder) เพื่อที่จะวัดระยะทางและความเร็วเชิงเส้นแทนที่จะใช้ตัววัดความเร็วและอินทิเกรตเพื่อหาความเร็วและการเคลื่อนที่ ซึ่งอาจเกิดการคลาดเคลื่อนได้ ข้อดีอีกอย่างหนึ่งก็คือการกำหนดอัตรา Pitch ที่สร้างขึ้นในตัวควบคุมเพื่อเลียนแบบระบบรักษาความปลอดภัยของพาหนะจริงพร้อมคนขับ หากอัตรา Pitch มีค่าเกินกว่าที่กำหนดไว้ ตัวควบคุมก็จะหยุดการทำงานของระบบควบคุม ส่วนของชุดเกียร์ทำให้เกิด Backlash เป็นจุดอ่อนของ Segway เพราะว่า Encoder ติดตั้งอยู่บนมอเตอร์ไม่ใช้บนล้อจึงไม่สามารถทำให้ตรวจจับปัญหา Backlash ได้

โดยรวมแล้วบทความนี้อธิบายเกี่ยวกับระบบที่ได้รับการออกแบบและคำนวณมาอย่างดี เพื่อที่จะได้แบบจำลองที่มีความแม่นยำสูง รวดเร็วและมีประสิทธิภาพ

2.2.3 รูปแบบของเบลกลเวล(The Blackwell Model)

2.2.3.1 รูปแบบที่ 1

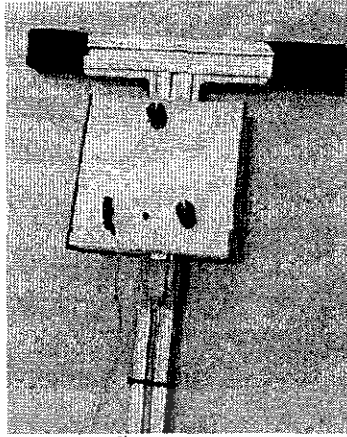
บทความ การสร้างสกุต์เตอร์รักษาสมดุลย์(Blackwell 2005) เป็นตัวอย่างของการพยายามที่สำเร็จในการสร้างพาหนะที่รักษาสมดุลย์ด้วยตนเองที่มีลักษณะคล้ายคลึงกับ Segway HT ในปี 2002 ผู้แต่งบทความได้แสดงการเปรียบเทียบสิ่งประดิษฐ์ของเขา กับ Segway HT และในบทความยังมีรายละเอียดเกี่ยวกับการประกอบอีกด้วย

การสร้างยานพาหนะนี้สามารถสร้างได้ง่ายจากชิ้นส่วนที่หาได้ทั่วไป ข้อได้เปรียบนี้ทำให้การสร้างมีราคาถูกและง่ายต่อการผลิตและประกอบ จากบทความเบลกลเวล(2005)กล่าวว่า “ส่วนประกอบทางกลของมันนั้นง่ายกว่าพาหนะชนิดอื่น ๆ มาก” อย่างไรก็ตามนี่ก็อาจเป็นข้อเสียได้เช่นกัน เพราะว่าทำให้มีรูปร่างที่ไม่สวยงามและให้ความสำคัญกับความสัมพันธ์ระหว่างมนุษย์กับเครื่องกลไม่มากเท่าที่ควร ดังแสดงในรูปที่ 2.18

แท่นยืนทำจากกล่องพลาสติกที่ใช้ใส่อุปกรณ์อิเล็กทรอนิกส์ทั่วไปติดกับเสาที่มีปุ่มหมุนเพื่อการเลี้ยวและการปรับจูนอัตราขยายของระบบควบคุมวงปิดรวมถึงสวิทช์หยุดฉุกเฉินด้วย

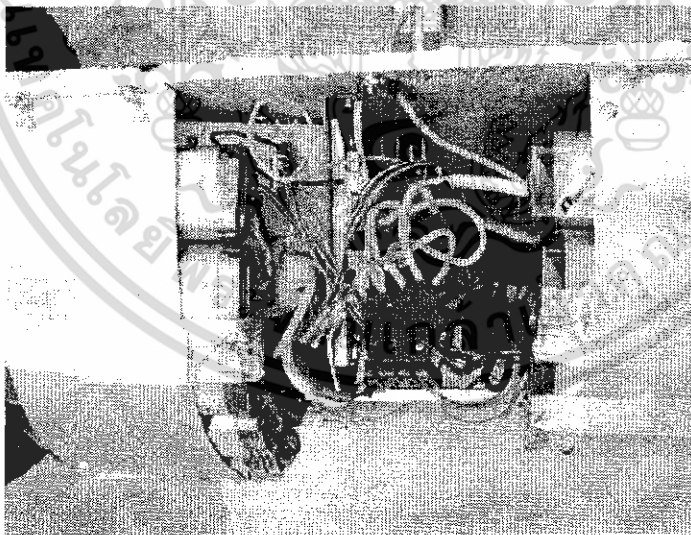


รูปที่ 2.18 แสดงพาหนะของ Blackwell



รูปที่ 2.19 แสดงระบบการเลี้ยวและอุปกรณ์การปรับอัตราขยาย

ส่วนของแหล่งพลังงานประกอบด้วยแบตเตอรี่ของรถบังคับวิทยุ โดยต่อรวมกันแบบขนาน และมีวงจรมอเตอร์เรกติไฟเออร์ ต่อกับแบตเตอรี่ทุกก้อนเพื่อป้องกันกระแสไฟรั่วไหลจากแบตเตอรี่ก้อนหนึ่งไปยังอีกก้อนหนึ่งโดยไม่จำเป็น และแบตเตอรี่แบบนิกเกิลเมทัลไฮไดรไรด์สามารถทนต่อการกระชากของกระแสที่ต้องการ โดยมอเตอร์ได้ จุดอ่อนของการออกแบบที่กล่าวไว้ในบทความนี้ก็คือการติดตั้งแบตเตอรี่ไว้ได้แน่นยึดอย่างหยวบๆ (รูป 2.20) ซึ่งอาจจะทำให้สายไฟเกิดหลุดได้และเป็นสาเหตุให้ระบบพลังงานล้มเหลว จุดอ่อนอีกข้อหนึ่งของตัวอย่างนี้ก็คือแบตเตอรี่มีความจุกระแสน้อย(2500mAh)เปรียบเทียบกับการใช้แบตเตอรี่ก้อนใหญ่ก้อนเดียวซึ่งจะให้กระแสได้มากกว่า



รูปที่ 2.20 แสดงแพ็คเกจแบตเตอรี่ที่อยู่ใต้ชั้น

ระบบควบคุมของพาหนะนี้ทำงานภายใต้ไมโครคอนโทรลเลอร์ขนาด 8 บิตของบริษัท Atmel โดยใช้ตัวควบคุมพีดี ซึ่งรับสัญญาณป้อนกลับจากใจโรสโคปแบบเปียโซอิเล็กทริก จุดอ่อนของ

ระบบนี้ก็คือจะต้องทำการปรับจูนระบบด้วยมือขณะที่กำลังขี่มันอยู่เท่านั้น แทนที่จะใช้ประโยชน์จากระบบจำลองเสมือนจริงซึ่งจะทำให้มีความปลอดภัยมากขึ้น

ระบบควบคุมการเลี้ยวของระบบนี้ทำโดยการเพิ่มและลดกำลังของมอเตอร์เป็นเปอร์เซ็นต์โดยขึ้นอยู่กับความเร็วที่ต้องการขณะนั้น มอเตอร์ถูกควบคุมโดยสัญญาณการมอดูเลตความกว้างพัลส์ (Pulse Width Modulation(PWM)) จากส่วนควบคุมมอเตอร์ PWM ทำให้มอเตอร์สามารถให้แรงบิดสูงสุดในขณะที่ยังมีความเร็วอยู่ภายใต้ความเร็วสูงสุด โดยมันจะจ่ายแรงดันสูงสุดที่มอเตอร์สามารถรับได้ และเปลี่ยนแปลงค่าวัฏจักรการใช้งานของพัลส์ เพื่อที่จะปรับเปลี่ยนความเร็วตามที่ต้องการ

โดยรวมแล้ว บทความนี้คล้ายกับคู่มือการประกอบมากกว่าแต่ว่าแสดงถึงความเป็นไปได้ที่จะสามารถสร้างอะไรที่คล้ายคลึงกับ Segway HT ได้ภายใต้ราคาที่ยอมรับได้

2.2.3.2 รูปแบบที่ 2

บทความนี้เป็นสตูดิโอรักษาสมดุสยแบบที่ 2 ซึ่งพัฒนามาจากตัวต้นแบบตัวแรก และมีการปรับปรุงให้มีสมรรถนะที่ดีหรือดีกว่า Segway HT

รูปแบบใหม่นี้ออกแบบให้มีความคล่องตัวมากกว่า เบากว่า เร็วกว่าและมีพิสัยที่ไกลกว่าแบบแรก มันได้ถูกออกแบบให้พอดีที่จะเคลื่อนที่ผ่านประตูได้มีระยะห่างจากพื้นพื้นมากขึ้นมีระบบการควบคุมเลี้ยวที่ดีขึ้น การปรับปรุงนี้มีส่วนจากการที่ผู้แต่งบทความมีโอกาสได้ขี่ Segway HT และได้รับรู้ถึงจุดด้อยในการออกแบบของตน



รูปที่ 2.21 แสดงพาหนะของ Blackwell แบบที่ 2

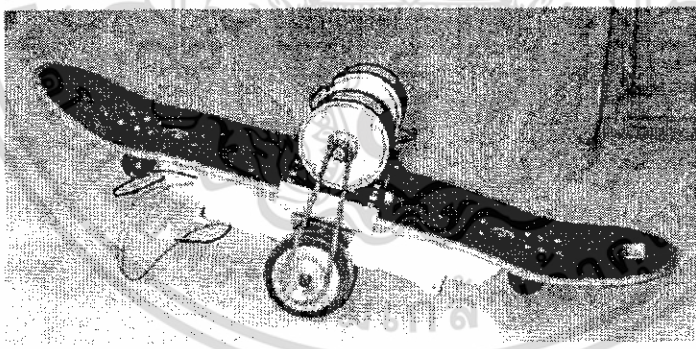
ผู้แต่งได้เปลี่ยนล้อจากล้อ โฟม 14 นิ้ว เป็นล้อจักรยาน 20 นิ้ว ที่มีซี่ล้อเสริมแรงด้วยทำให้น้ำหนักเบาขึ้น ลดแรงเสียดทานและลดแรงบิด ความเฉื่อย และแรงสั่นสะเทือนที่ปกติจะได้จากล้อแบบ

โฟม ผู้แต่งยังเปลี่ยนวงจรขั้วมอเตอร์ให้สามารถจ่ายแรงดัน13-50 โวลต์ที่160แอมป์อย่างต่อเนื่อง หรือ400แอมป์ชั่วขณะและความเร็วในการประมวลผลก็เร็วขึ้นจาก10มิลลิวินาทีเป็น1-2มิลลิวินาที

นอกจากนี้ยังได้เปลี่ยนใจโรสโคปเพียงอย่างเดียว เป็นการใช้ทั้งใจโรสโคปและเซ็นเซอร์วัด ความเร่ง ซึ่งทำให้สัญญาณรบกวนลดลงอย่างมากและสามารถรับมือกับแรงสั่นสะเทือนได้มากขึ้น แบตเตอรี่ก็เปลี่ยนจากขนาดAAเป็นขนาดD 60ก้อนก้อนละ6500mAh สามารถจ่ายแรงดัน36 โวลต์ ที่กระแสสูงสุด200แอมป์แอมป์ เพิ่มเดิมส่วนของสัญญาณบลูทูท และเขียน โปรแกรมให้สามารถ ขยับเคลื่อนได้โดยบังคับจากระยะทางไกล การบังคับก็เปลี่ยนจากแบบเป็นปุ่มกดเป็นตัว ดำเนินการปรับค่าได้เหมือนการบังคับของจรวดยานที่บังคับการเลี้ยวโดยการหมุนคันจับและยังมีการ ปรับปรุงตัวถังเพื่อปรับปรุงให้แข็งแรงและสามารถปกป้องอุปกรณ์ภายในได้

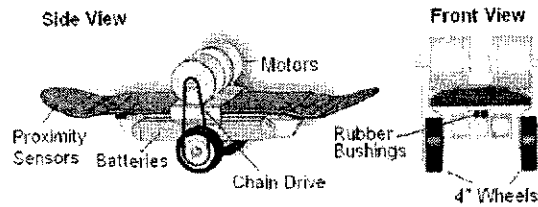
2.2.4 รูปแบบของ ชูจลิกซ์ (The Chudleigh Model)

ในบทความนี้,โครงการเอมมานูเอล โครงการที่ดูเหมือนจะเป็น สเก็ตบอร์ด2ล้อที่รักษา สมดุลย์ได้เอง(Chudleigh et al.2005) ได้กล่าวถึงการดัดแปลงสเก็ตบอร์ดให้มีสองล้ออยู่ในแกน เดียวและวางล้อขนานกันอยู่ใต้จุดศูนย์กลางของสเก็ตบอร์ด (รูป2.22) และการใช้ระบบควบคุม อัตโนมัตินี้ เพื่อควบคุมมอเตอร์ในการรักษาสมดุลย์ ในบทความยังกล่าวถึงข้อคิดเห็นของผู้แต่ง เกี่ยวกับพวหนะขับเคลื่อนได้ด้วยไฟฟ้าแบบใหม่อีกด้วย



รูปที่ 2.22 แสดงรูป The Chudleigh Model

บทความได้อธิบายถึงส่วนประกอบที่ใช้และปัญหาที่เกิดขึ้นในระหว่างการสร้าง อุปกรณ์ที่ใช้ ประกอบด้วยมอเตอร์ไฟฟ้าขนาด 300 วัตต์ 24 โวลต์ โดยแต่ละตัวจะควบคุมล้อโดยส่งแรงผ่าน โซ่ ดังแสดงในรูป2.23 การออกแบบนี้ใช้แบตเตอรี่ลิเทียมไอออนขนาด14.4 โวลต์ต่ออนุกรมกัน จุดเด่น ของการออกแบบนี้คือแบตเตอรี่เพราะว่ามีน้ำหนักเบาและยังสามารถจ่ายกระแสได้มากกว่าที่ ต้องการตามปกติถึง10เท่า



รูปที่ 2.23 แสดงรูป The Chudleigh Model ในแบบ CAD

แทนที่จะใช้เซ็นเซอร์วัดแรงเฉื่อยอย่าง Segway HT พาหนะนี้ใช้เซ็นเซอร์วัดระยะทางอยู่ภายใต้แท่นยืนเพื่อที่จะวัดว่าส่วนปลายของบอร์คอยู่ห่างจากพื้นเท่าไร ตัวเสกต์บอร์ดได้ติดตั้งอยู่บนแหวนยางเพื่อที่จะสามารถเอียงบอร์คทางด้านข้างได้เล็กน้อยเพื่อที่จะให้เซ็นเซอร์อินฟราเรดที่ติดอยู่ด้านข้างทั้งสองตรวจจับการบังคืบเลี้ยวได้ เซ็นเซอร์ทั้ง 2 แสดงในรูปที่ 2.24 ข้อมูลที่ได้จากเซ็นเซอร์จะถูกส่งไปยังไมโครคอนโทรลเลอร์ PIC (Programmable Integrate Circuit) เพื่อทำการเพิ่มหรือลดพลังงานที่ส่งไปยังมอเตอร์เพื่อการควบคุมการเลี้ยวคล้ายคลึงกับของแบลคเวลท์ที่กล่าวมาตอนต้น ข้อได้เปรียบของรูปแบบนี้ก็คือ การใช้ไมโครคอนโทรลเลอร์ เซ็นเซอร์วัดระยะทางและเซ็นเซอร์อินฟราเรด มีราคาถูกกว่ามาก ส่วนข้อด้อยก็คือทั้งอินฟราเรดเซ็นเซอร์และเซ็นเซอร์วัดระยะทางทำให้เกิดสัญญาณรบกวนทางการวัดมากดังนั้นจึงทำให้ระบบควบคุมที่ใช้มีเสถียรภาพไม่มากนัก

ชวลีจ (Chudleigh et al 2005) ใช้ระบบควบคุมแบบคลาสสิกพีดี ที่ถูกนำไปทดลองใน MATLAB's Simulink toolbox และแบบจำลองเสมือนจริง

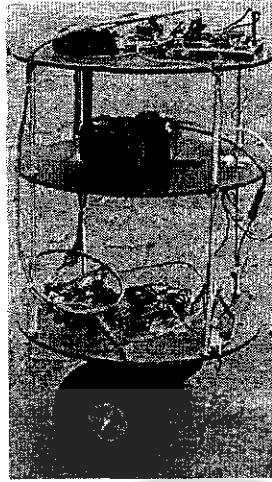
โดยรวมแล้ว บทความนี้นำเสนอโครงงานไปทางด้านวงจรไฟฟ้ามากกว่า การออกแบบมีราคาถูกมีประสิทธิภาพและมีความคิดสร้างสรรค์ ข้อด้อยก็คือการใช้เซ็นเซอร์ที่ทำให้เกิดสัญญาณรบกวนมาก



รูปที่ 2.24 แสดงเซ็นเซอร์วัดระยะทางและเซ็นเซอร์อินฟราเรด

2.2.6 รูปแบบของลาตัน (The Larson Model)

ในบทความนี้ โครงงานหุ่นรักษาสมดุล เป็นเคส ลาตัน 2005 (bender Larson 2005) ผู้แตงนำเสนอรายละเอียดของกรรมวิธีที่เขาใช้สร้างหุ่นยนต์รักษาสมดุลโดยอัตโนมัติ (autonomous self-balancing robot)



รูปที่ 2.25 แสดงรูปแบบของถาดล้น

โครงการนี้หุ่นยนต์เริ่มต้นจากขนาดที่ใหญ่ก่อน มีส่วนประกอบทางกลต่างๆประกอบด้วย กรอบอลูมิเนียมที่ใช้เป็นที่ติดตั้งมอเตอร์และระบบขับเคลื่อน และล้อยพลาสติกPVC ที่มีความสูง และมีสมดุคัลย์ที่ดี ทำให้มีความแข็งแรงและน้ำหนักเบาอีกด้วย มอเตอร์ที่ใช้ในโครงการนี้มีตัว เข้ารหัสที่มีความละเอียดสูง ทำให้สามารถนำมาหาตำแหน่งและความเร็วของหุ่นได้อย่างแม่นยำ

หน่วยประมวลผลหลักใช้ไมโครคอนโทรลเลอร์เบอร์PIC18F452ของบริษัทไมโครชิพ รวมถึง หน่วยประมวลผลรองที่เอาไว้รับข้อมูลจากตัวเข้ารหัสโดยเฉพาะ เพื่อที่หน่วยประมวลผลหลักจะได้ ไม่ต้องเสียเวลาในการประมวลผลข้อมูลดิบ แบตเตอรี่ถูกนำไปติดตั้งไว้ในที่สูงเพื่อช่วยในการ สมดุลย์ เซ็นเซอร์ที่ใช้ในการรักษาสมดุคัลย์คือ ไจโรสโคป Tonkin CG16DO ซึ่งใช้หลักการทำงาน ของเปียโซอิเล็กทริก ร่วมกับข้อมูลที่ได้จาก เซ็นเซอร์วัดความเร่งแบบ2แกน MXD2125-GW เพื่อที่จะได้ข้อมูลที่แม่นยำและรวดเร็ว

ข้อดีของโครงการนี้ก็คือ สามารถประกอบและคิดแปลงได้ง่าย และการใช้ทั้งไจโรสโคปและ เซ็นเซอร์วัดความเร่งร่วมกันเพื่อความแม่นยำของข้อมูล

บทที่ 3

การออกแบบระบบการทำงาน

ในโครงการนี้มีความจำเป็นที่จะต้องใช้การประมวลผลแบบตามเวลาจริง(Real time) เพราะว่าเป็นการทำงานที่เกี่ยวกับสถานะนั้นๆในช่วงเวลาหนึ่งเท่านั้น หากประมวลผลช้าเกินไปจะทำให้ตอบสนองต่อสถานะนั้นๆผิดพลาดไปจากที่ต้องการ โครงการนี้จึงต้องใช้หน่วยประมวลผลที่เร็ว ประกอบกับมีการคำนวณทางคณิตศาสตร์ร่วมด้วย คณะผู้จัดทำจึงเลือกใช้หน่วยประมวลผลสัญญาณดิจิทัลเบอร์ dsPIC30F2010 ของบริษัทไมโครชิพเป็นหน่วยประมวลผลหลักของโครงการ

3.1 คุณสมบัติของ dsPIC30F2010

dsPIC30F2010 เป็น MCU ซึ่งใช้การประมวลผลข้อมูลแบบ 16 บิต จากค่าย Microchips ซึ่งมีจุดเด่นในด้านของความสามารถในการประมวลผลข้อมูลสัญญาณแบบดิจิทัล สำหรับนำไปประยุกต์ใช้ในงานควบคุมต่างๆ โดยโครงสร้างภายในจะเป็นการผสมผสานระหว่างไมโครคอนโทรลเลอร์ (MCU) และวงจร DSP (Digital Signal Processing) รวมเข้าไว้ด้วยกัน หรืออาจเรียก MCU ตระกูล dsPIC ว่าเป็น DSC หรือ Digital Signal Controller ก็ได้

คุณสมบัติด้านการประมวลผล

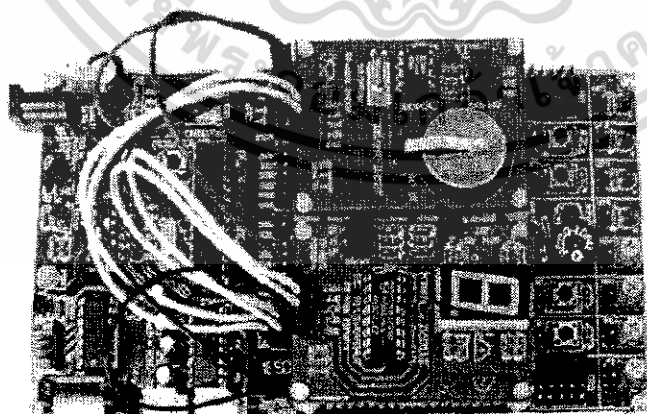
- ใช้สถาปัตยกรรมแบบ RISC โดยมี 84 คำสั่งมาตรฐาน รองรับการอ้างตำแหน่งแอดเดรสแบบต่างๆได้โดยอิสระ โดยรูปแบบโครงสร้างการจัดผังหน่วยความจำจะแตกต่างจากสถาปัตยกรรมแบบ Harvard
- ชุดคำสั่ง ใช้การอ้างแอดเดรสแบบ 24บิต และการอ้างถึงข้อมูลขนาด 16บิต
- มีหน่วยความจำโปรแกรมแบบ Flash ขนาด 12KByte (4KWord) สามารถทำการลบและโปรแกรมซ้ำใหม่ได้กว่า 100,000 ครั้ง พร้อมระบบป้องกันการอ่าน
- มีหน่วยความจำ RAM ขนาด 512 Byte
- มีหน่วยความจำข้อมูลถาวรแบบ EEPROM ขนาด 1KByte สามารถลบและเขียนซ้ำได้กว่า 1,000,000 ครั้ง และสามารถเก็บรักษาข้อมูลไว้ได้แม้ไม่ได้จ่ายไฟเลี้ยงให้ MCU
- มีรีจิสเตอร์ขนาด 16 บิต ให้ใช้งานจำนวน 16 ชุด
- สามารถประมวลผลด้วยความเร็วสูงสุดที่ 30 MIPS (30 ล้านคำสั่งต่อวินาที)
- รองรับสัญญาณนาฬิกาจากแหล่งกำเนิดภายนอก 0-40 MHz
- รองรับการใช้งานกับแหล่งกำเนิดความถี่แบบ XTAL ค่า 4-10 MHz

- มีวงจรถนอมความถี่ภายในแบบ Phase-Lock-Loop โดยสามารถกำหนดค่าอัตราการถนอมความถี่ได้ 3 ระดับ คือ 4 เท่า, 8 เท่า และ 16 เท่า
- รองรับการ Interrupt ได้ถึง 27 แหล่ง พร้อมสัญญาณ Interrupt จากภายนอก 3 แหล่ง และสามารถจัดระดับความสำคัญของการ Interrupt ได้ 8 ระดับ

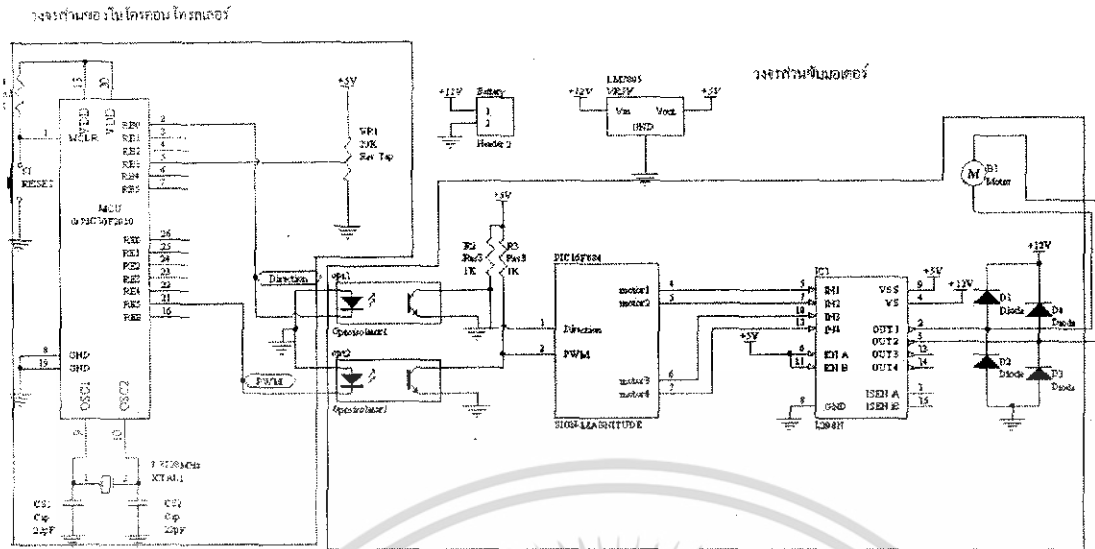
คุณสมบัติของ Peripheral I/O

- ขาสัญญาณ I/O สามารถจ่ายกระแส(Source) และ รับกระแส (Sink) ได้มากถึง 25mA
- มี Timer ขนาด 16 บิต จำนวน 3 ชุด และสามารถโปรแกรมใช้งานเป็น Timer แบบ 32 บิต ได้โดยใช้ Timer 16 บิต 2 ช่องรวมกัน
- มี Input Capture ขนาด 16 บิต จำนวน 4 ช่อง
- มี Output Compare/PWM ขนาด 16บิต จำนวน 2 ช่อง
- มีวงจรถ่ายโอนข้อมูลแบบ SPI จำนวน 1 ช่อง
- มีวงจรถ่ายโอนข้อมูลแบบ I2C จำนวน 1 ช่อง
- มีวงจรถ่ายโอนข้อมูลแบบ UART จำนวน 1 ช่อง
- มีวงจรถ่ายโอนข้อมูลแบบ DCPWM สำหรับใช้ควบคุมมอเตอร์ 3 ช่อง
- มีวงจรถอดรหัสแบบ QEIM ขนาด 16บิต จำนวน 1 ช่อง
- มีวงจรถ่ายโอนข้อมูลแบบ A/D ขนาด 10บิต จำนวน 6 ช่อง

เพื่อความสะดวกและรวดเร็วในการพัฒนาโครงการ คณะผู้จัดทำได้เลือกที่จะใช้บอร์ดทดลองสำเร็จรูปของบริษัท อีทีที จำกัด เบอร์ ET- dsPIC30F2010 TRAINING KIT V1.0 เพราะว่ามีส่วนต่อเชื่อมที่จำเป็นเพียงพอต่อการใช้งาน และมีส่วน โปรแกรมได้ในวงจร (in circuit programming)

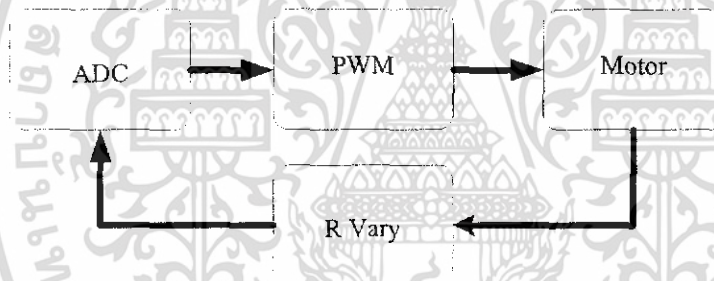


รูปที่ 3.1 แสดงบอร์ดทดลองเบอร์ ET- dsPIC30F2010 TRAINING KIT V1.0



รูปที่ 3.2 แสดงวงจรส่วนไมโครคอนโทรลเลอร์และส่วนขับเคลื่อนมอเตอร์

3.2 บล็อกไดอะแกรมวงจรการควบคุมการทำงานของหุ่นยนต์รักษาสมดุลย์



รูปที่ 3.3 แสดงบล็อกไดอะแกรมวงจรการควบคุมการทำงานของหุ่นยนต์รักษาสมดุลย์

จากรูปที่ 3.3 สามารถอธิบายการทำงานของบล็อกไดอะแกรมได้ดังนี้ ตัวต้านทานปรับค่าได้ (R vary) ทำหน้าที่เป็นเซ็นเซอร์ในการตรวจจับว่าขณะนี้หุ่นยนต์เอียงไปทางใดมากเท่าใด โดยให้สัญญาณออกมาเป็นสัญญาณอนาล็อก ซึ่งจะถูกแปลงโดยหน่วยแปลงสัญญาณอนาล็อกเป็นดิจิตอล (ADC) ขนาด 10 บิตที่อยู่ในตัว dsPIC อยู่แล้ว จากสัญญาณดิจิตอลที่ได้จะนำไปเข้าหน่วยกำเนิดสัญญาณการมอดูเลตความกว้างพัลส์ (Pulse Width Modulation (PWM)) ไปควบคุมแรงบิดของมอเตอร์ ซึ่งจะขับเคลื่อนมอเตอร์ด้วยความเร็วที่เหมาะสมและค่าที่ได้จากตัวต้านทานปรับค่าได้จะเปลี่ยนแปลงไปและเริ่มวงจรใหม่เช่นนี้วนไปเรื่อยๆ ในส่วนต่อไปจะกล่าวถึงรายละเอียดในการใช้งานไมโครคอนโทรลเลอร์ ในหน่วยต่างๆที่ใช้ในโครงการนี้โดยหัวข้อที่ 3.2.1 จะกล่าวถึงรายละเอียดของหน่วยแปลงสัญญาณอนาล็อกเป็นดิจิตอล และหัวข้อที่ 3.2.2 จะกล่าวถึงรายละเอียดของหน่วยกำเนิดสัญญาณการมอดูเลตความกว้างพัลส์

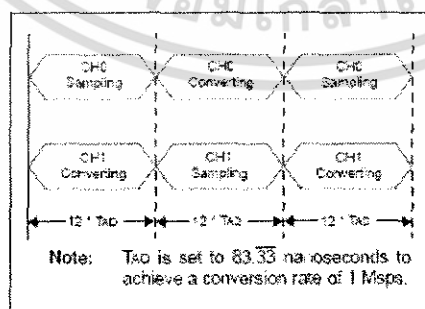
3.2.1 หน่วยการแปลงสัญญาณอนาลอกเป็นดิจิทัล

เนื่องจากโครงการนี้ใช้ข้อมูลจากเซ็นเซอร์เป็นสัญญาณอนาลอกเพื่อที่จะนำไปประมวลผลจึงจำเป็นต้องมีการแปลงสัญญาณอนาลอกเป็นดิจิทัลเสียก่อนเป็นอันดับแรก ดังนั้นการแปลงสัญญาณอนาลอกเป็นดิจิทัลให้รวดเร็วจึงมีความสำคัญมาก เพราะว่าหากเกิดการล่าช้าขึ้นก็ย่อมจะส่งผลให้การประมวลผลต่อไปผิดพลาดเนื่องจากได้ข้อมูลที่ไม่เป็นจริงที่เวลานั้นๆ คณะผู้จัดทำจึงศึกษาการใช้หน่วยแปลงสัญญาณอนาลอกเป็นดิจิทัลโดยมีจุดมุ่งหมายให้ทำการแปลงสัญญาณได้เร็วที่สุดเท่าที่หน่วยประมวลผลจะรองรับได้

ไมโครคอนโทรลเลอร์ dsPIC30F2010 มีหน่วยแปลงสัญญาณอนาลอกเป็นดิจิทัลขนาด 10 บิต ซึ่งสามารถทำการแปลงค่าได้สูงถึง 1 ล้านตัวต่อวินาที ต่อไปนี้จะเป็นการอธิบายว่าเราจะกำหนดค่าอย่างไรให้ dsPIC สามารถแปลงค่าได้เร็วอย่างนั้นได้

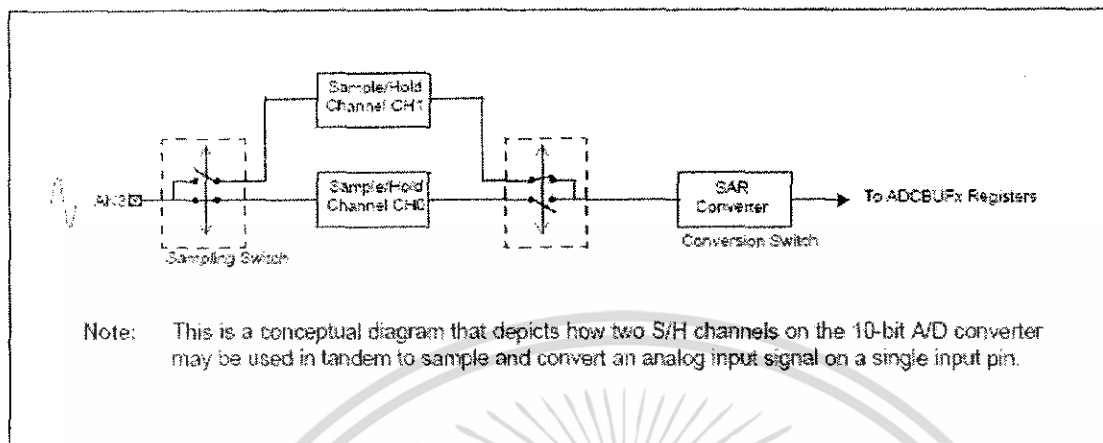
หน่วยแปลงสัญญาณอนาลอกเป็นดิจิทัลบน dsPIC นั้นมีช่องซั๊กตัวอย่างและคงค่า (sample and hold (S/H)) อยู่ 4 ช่องคือ CH0, CH1, CH2, CH3 ขาที่เป็นสัญญาณเข้าอนาลอกจะเชื่อมต่อกับช่องรักษาตัวอย่างและคงค่าเหล่านี้ผ่านทางระบบของมัลติเพล็กซ์ โดยปกติขาที่เป็นสัญญาณเข้าอนาลอกจะเชื่อมต่อกับช่องซั๊กตัวอย่างและคงค่า CH0, CH1, CH2 หรือ CH3 1 ถึง 2 ช่อง ซึ่งการเลือกว่าจะใช้ขาใดโดยรีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register (SFR)) ชื่อ ADCHS และ บิต CHPS (ADCON2<9:8>) จะใช้กำหนดว่าจะเปิดช่องซั๊กตัวอย่างและคงค่าช่องใดให้ทำงานบ้าง หากกำหนดค่า '01' ให้แก่ CHPS ช่องซั๊กตัวอย่างและคงค่าช่อง CH0 และ CH1 ก็จะทำเป็นต้น

หน่วยแปลงสัญญาณอนาลอกเป็นดิจิทัลสามารถกำหนดให้ช่องซั๊กตัวอย่างและคงค่า 2 ช่อง หรือ 4 ช่องรับสัญญาณอนาลอกของขาสัญญาณเข้าได้พร้อมๆ กัน หรือติดต่อกันก็ได้ (สัญญาณหนึ่งต่อจากอีกสัญญาณหนึ่ง) การทำงานในรูปแบบของการรับสัญญาณแบบติดต่อกันจะสามารถทำให้เกิดค่าการแปลงที่เร็วที่สุดได้ การเซ็ตรูปแบบของการรับสัญญาณแบบติดต่อกันสามารถทำได้โดยเซ็ตค่าบิต SIMSAM (ADCON1<3>) ให้เป็นศูนย์ แสดงแผนผังการทำงานดังรูปที่ 3.4



รูปที่ 3.4 แผนผังการทำงานของการใช้ช่องซั๊กตัวอย่างและคงค่า 2 ช่องในรูปแบบของการรับสัญญาณแบบติดต่อกัน

เพื่อให้ทำการแปลงสัญญาณให้ได้ 1 ล้านครั้งต่อวินาที ควรกำหนดให้ใช้ขาสัญญาณเข้าอนุภาค 1 ขาต่อช่องซัดตัวอย่างและคงค่า 2 ช่องเช่น CH0 และ CH1 ดังรูป 3.5 ต่อไปจะกล่าวถึงขั้นตอนการกำหนดค่าต่างๆเพื่ออัตราการแปลงสัญญาณ 1 ล้านครั้งต่อวินาที



รูปที่ 3.5 แสดงแผนผังการต่อเชื่อมเพื่อให้ค่าอัตราการสุ่มมากที่สุด

ขั้นตอนในการเซตค่า

1. เปิดการทำงาน การซัดตัวอย่างอัตโนมัติ

$$\text{ASAM}(\text{ADCON1} \langle 2 \rangle) = '1'$$

2. เปิดการทำงาน การคงค่าอัตโนมัติ

$$\text{SSRC}(\text{ADCON1} \langle 7:5 \rangle) = '111'$$

3. เปิดการทำงาน รูปแบบของการรับสัญญาณแบบติดต่อกัน

$$\text{SIMSAM}(\text{ADCON1} \langle 3 \rangle) = '0'$$

4. เปิดการทำงาน ช่องซัดตัวอย่างและคงค่า 2 ช่อง

$$\text{CHPS} \langle 1:0 \rangle (\text{ADCON2} \langle X: Y \rangle) = '01'$$

5. กำหนดค่าความสำคัญของอินเตอร์ปต์ให้มากกว่า 1

$$\text{SMPI} \langle 3:0 \rangle (\text{ADCON2} \langle 5: 2 \rangle) = '0000'$$

6. เลือกค่าการซัดตัวอย่างต่ำสุด

$$\text{SAMC} \langle 4:0 \rangle (\text{ADCON} \langle 12:8 \rangle) = '00010'$$

7. เลือกค่า T_{AD} ต่ำสุด

$$\text{ADCS} \langle 5:0 \rangle (\text{ADCON3} \langle 5:0 \rangle)$$

ค่า $T_{AD} > 83.3$ จึงจะได้อัตราการแปลง 1 ล้านตัวอย่างต่อวินาที

8. ตั้งค่ารีจิสเตอร์ ADCHS เพื่อกำหนดขาสัญญาณเข้าอนุภาคให้แก่ CH0 และ CH1

ขั้นตอนที่นี้เป็นขั้นตอนที่สำคัญมากเพราะว่าหน่วยแปลงสัญญาณอนุภาคเป็นดิจิตอลจะเขียนผลลัพธ์ที่ได้เริ่มตั้งแต่ ADCBUF0 ในทุกครั้งที่เกิดการอินเตอร์ปต์ หากจะใช้ช่องซัดตัวอย่างและคง

ส่วนประกอบหลักของโมดูลนี้คือส่วนกำเนิดสัญญาณ PWM ซึ่งได้ค่าฐานเวลามาจาก รีจิสเตอร์ PTMR และ PTPER ในโมดูล MCPWM นี้สามารถกำหนดค่าวัฏจักรหน้าที่ (Duty cycle) ในส่วนกำเนิดสัญญาณ PWM แต่ละส่วนที่เป็นอิสระต่อกัน นอกจากนี้ยังสามารถกำหนดการทำงานของขาพอร์ตเอาต์พุตของโมดูล MCPWM โดยตรงผ่านทางรีจิสเตอร์ OVDCON

สัญญาณที่ออกจากโมดูล MCPWM จะมีขาพอร์ต 2 ขาต่อช่องนั้นคือ ขาเอาต์พุตด้านแรงดันสูง PWMxH และขาเอาต์พุตด้านแรงดันต่ำ PWMxL หรือเรียกว่าคู่เอาต์พุต

ฐานเวลา PWM ในโมดูล MCPWM สามารถกำหนดให้ทำงานได้ 4 โหมดคือ

- โหมดเปลี่ยนแปลงค่าอิสระ
- โหมดทำงานครั้งเดียว
- โหมดนับค่าขึ้นหรือลงอย่างต่อเนื่อง
- โหมดนับค่าขึ้นหรือลงอย่างต่อเนื่องพร้อมการอินเตอร์รัปต์เพื่อปรับปรุ่ค่า

การเลือกโหมดทำได้โดยการกำหนดค่าที่บิต PTMOD1 และ PTMOD0 ซึ่งเป็นบิต 1 และ 0 ในรีจิสเตอร์ PTCON ซึ่งในโครงการนี้จะเลือกใช้โหมดเปลี่ยนแปลงค่าอิสระ ในโหมดนี้ค่าฐานเวลาจะเพิ่มค่าขึ้นจนกระทั่งตรงกับค่าในรีจิสเตอร์ PTER จากนั้นรีจิสเตอร์ PTMR จะรีเซ็ตและทำการนับค่าเพิ่มขึ้นต่อเนื่องไปอีกตราบเท่าที่บิต PTEN ยังคงเซตเป็น '1' อยู่

การเปลี่ยนค่าวัฏจักรหน้าที่สัญญาณ PWM ของโมดูล MCPWM กำหนดได้จากรีจิสเตอร์ PDC1-PDC4 ซึ่งต่างก็มีบัฟเฟอร์เพื่อป้องกันสัญญาณรบกวนเมื่อมีการปรับปรุ่ค่าของสัญญาณ PWM โดยวัฏจักรหน้าที่ของสัญญาณ PWM จะถูกปรับปรุ่ค่าตามข้อมูลที่เขียนลงในรีจิสเตอร์ PDCx จากนั้นค่าจากรีจิสเตอร์ PDCx จะถูกส่งไปยังบัฟเฟอร์เพื่อทำการเปรียบเทียบ เมื่อมีการเปลี่ยนแปลงข้อมูลในรีจิสเตอร์ PDCx เรียบร้อย ข้อมูลนั้นจึงถูกส่งไปบัฟเฟอร์ข้อมูลนั้นจึงถูกส่งไปยังบัฟเฟอร์เพื่อทำงานต่อไป ทำให้ไม่เกิดการติดขัดหรือเกิดความผิดพลาดในขณะที่เปลี่ยนค่าวัฏจักรหน้าที่ เมื่อฐานเวลา PWM ทำงานในโหมดเปลี่ยนแปลงค่าอิสระหรือ โหมดทำงานครั้งเดียว ค่าวัฏจักรหน้าที่ของสัญญาณ PWM จะถูกปรับปรุ่เมื่อค่าของรีจิสเตอร์ PTMR เท่ากับ PTPER และเมื่อรีจิสเตอร์ PTMR เกิดการรีเซ็ตเป็นศูนย์

นอกจากนั้นในโมดูล MCPWM ยังมีความสามารถพิเศษอีกประการหนึ่งคือ การป้องกันการเปลี่ยนค่าวัฏจักรหน้าที่ ทำได้โดยเซตบิต UDIS ซึ่งเป็นบิต 2 ของรีจิสเตอร์ PWMCON2

ตัวอย่างการตั้งค่าการใช้งานเพื่อกำเนิดสัญญาณสี่เหลี่ยมมีค่าวัฏจักรหน้าที่ 50%

```
#define __dsPIC30F6010__
#include <p30fxxxx.h>
#include <pwm.h>
void __attribute__((__interrupt__)) _PWMInterrupt(void)
```

ตัวอย่างการตั้งค่าการใช้งานเพื่อกำเนิดสัญญาณสี่เหลี่ยมมีค่าวัฏจักรหน้าที่50% (ต่อ)

```
{
IFS2bits.PWMIF = 0;
}
int main()
{
/* กำหนดค่าการอินเทอร์รัปต์ของPWM */
    unsigned int config;
/* กำหนดค่าที่จะโหลดให้ค่าวัฏจักรหน้าที่ */
    unsigned int period;
/* กำหนดค่าให้รีจิสเตอร์ของสัญญาณกระตุ้นพิเศษ */
    unsigned int sptime;
/* กำหนดค่าการใช้งานPWM */
    unsigned int config1;
/* กำหนดค่าที่จะโหลดให้ รีจิสเตอร์ PWMCON1 */
    unsigned int config2;
/* กำหนดค่าโพสดีสเคลและวัฏจักรหน้าที่ของสัญญาณกระตุ้นพิเศษ */
    unsigned int config3;
/* กำหนดค่าที่จะส่งไปยัง รีจิสเตอร์ PDCx */
    unsigned int dutycyclereg;
    unsigned int dutycycle;
    unsigned char updatedisable;
/* กำหนดค่าความสำคัญของอินเทอร์รัปต์ */
    config = (PWM_INT_EN & PWM_FLTA_DIS_INT & PWM_INT_PR1
& PWM_FLTA_INT_PR0
& PWM_FLTB_DIS_INT & PWM_FLTB_INT_PR0);
    ConfigIntMCPWM( config );
/* Configure PWM to generate square wave of 50% duty cycle */
    dutycyclereg = 1;
```

```
dutycycle = 0x3FFF;
updatedisable = 0;
```

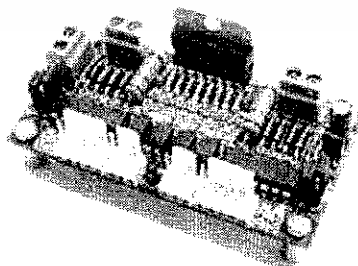
ตัวอย่างการตั้งค่าการใช้งานเพื่อกำหนดสัญญาณสี่เหลี่ยมมีค่าวัฏจักรหน้าที่ 50% (ต่อ)

```
SetDCMCPWM(dutycyclereg,dutycycle,updatedisable);
period = 0x7fff;
sptime = 0x0;
config1 = (PWM_EN & PWM_PTSIDL_DIS & PWM_OP_SCALE16
           & PWM_IPCLK_SCALE16 & PWM_MOD_UPDN);
config2 = (PWM_MOD1_COMP & PWM_PDIS4H & PWM_PDIS3H &
           PWM_PDIS2H & PWM_PEN1H & PWM_PDIS4L &
           PWM_PDIS3L & PWM_PDIS2L & PWM_PEN1L);
config3 = (PWM_SEVOPS1 & PWM_OSYNC_PWM & PWM_UEN);
OpenMCPWM(period,sptime,config1,config2,config3);
while(1);
}
```

เมื่อเราสามารถควบคุมการค่าวัฏจักรหน้าที่จากโมดูลMCPWMได้แล้วต่อไปก็จะขอกล่าวถึงการใช้บอร์ดขับมอเตอร์ที่ใช้ในการควบคุมมอเตอร์ในโครงการนี้

วงจรมอเตอร์

ในโครงการนี้ใช้บอร์ดขับมอเตอร์ Dual Motor Driver(DMD2A) ของบริษัทเอนนี่คอนโทรล จำกัดซึ่ง DMD2A เป็นบอร์ดขับมอเตอร์ไฟฟ้ากระแสตรงมีช่องเอาต์พุตสำหรับมอเตอร์ 2 ช่องและสามารถขับกระแสได้ 2 แอมป์แปรต่อช่อง สามารถเลือกรับสัญญาณควบคุมความเร็วมอเตอร์เป็นสัญญาณมอดูเลตความกว้างพัลส์(PWM) หรือควบคุมความเร็วด้วยสัญญาณอนาลอก รูปที่ 3.8 แสดงรูปบอร์ดขับมอเตอร์ DMD2A



รูปที่ 3.8 แสดงรูปบอร์ดขับมอเตอร์ DMD2A

ลักษณะพิเศษ

- แหล่งจ่ายแรงดันไฟตรง 9-24 โวลต์
- เอาต์พุต 2 ช่อง กระแสเอาต์พุต 2 แอมแปร์ต่อช่อง
- ป้องกันกระแสเกินพิกัด
- ความถี่สัญญาณPWM สูงสุด 25KHz
- Isolated Digital Input and Non-isolated Analog Input

การทำงานของวงจร

เริ่มจากไมโครคอนโทรลเลอร์ PIC16F864 ทำหน้าที่เป็นหน่วยประมวลผลรองรับข้อมูลสัญญาณ PWM และ ทิศทางจากไมโครคอนโทรลเลอร์มาแล้วส่งสัญญาณผ่านตัวแบ่งแยกทางแสง เพื่อที่จะกันสัญญาณดิจิทัลออกจากส่วนของสัญญาณอนาล็อก ไปกระตุ้นขาของไอซีเบอร์ L298D ซึ่งเป็นไอซีขับเคลื่อนมอเตอร์ชนิด H-Bridge สามารถขับกระแสเอาต์พุตได้ 2 แอมแปร์ต่อช่อง 2 ช่อง ซึ่งหน่วยประมวลผลรองจะเป็นตัวตัดสินใจว่าจะส่งสัญญาณไปยังขาไหนบ้างจึงจะได้ทิศทางและความเร็วที่ต้องการซึ่งได้รับมาจากหน่วยประมวลผลหลัก



บทที่ 4

ส่วนประกอบทางกล

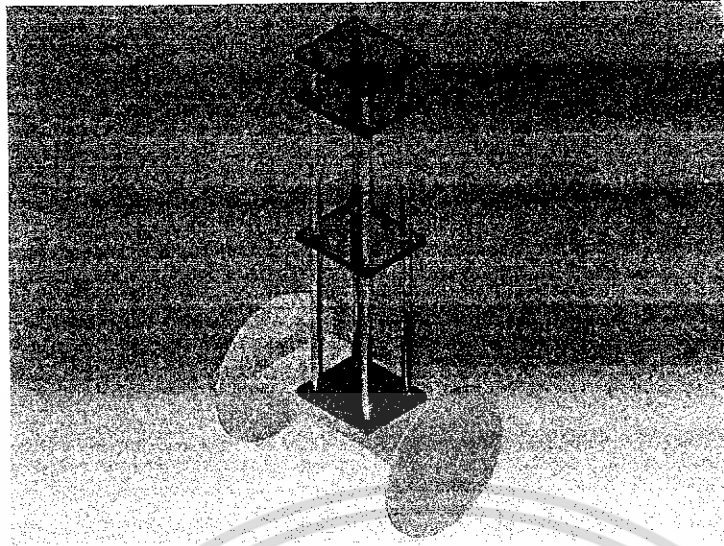
จากที่กล่าวมาในบทที่ 2 เราได้ศึกษาข้อดีข้อเสียของรูปแบบต่างๆที่มีผู้ที่ทำมาก่อนหน้านี้แล้ว มาจำนวนหนึ่งทั้งที่กล่าวอยู่ในบทความนี้และบางส่วนที่ไม่ได้กล่าวถึง พบว่า โครงสร้างหลักๆที่ทำให้หุ่นยนต์สามารถรักษาสมดุลได้ก็คือ หุ่นยนต์ต้องมีความสูงที่เหมาะสม, น้ำหนักเบา, มีระบบขับเคลื่อนที่รวดเร็วพอที่จะเปลี่ยนทิศทางบ่อยครั้งได้ โครงสร้างโดยรวมต้องมีความแข็งแรงไม่โยกเยก รวมถึงเซ็นเซอร์ที่ใช้ตรวจสอบลักษณะของหุ่นยนต์เมื่อสัมผัสกับพื้นต้องมีความแม่นยำ มีสัญญาณรบกวนต่ำและมีความเร็วในการตอบสนองต่อการเปลี่ยนแปลงสูง คณะผู้จัดทำจึงเห็นว่าการที่จะมีการออกแบบโดยโปรแกรมประเภท CAD ก่อน ซึ่งอย่างน้อยที่สุดก็จะได้เห็นลักษณะของหุ่นยนต์ก่อนและมั่นใจได้ว่าชิ้นส่วนแต่ละชิ้นจะสวมต่อเข้าด้วยกันได้พอดี ทำให้มีความแข็งแรงมีระเบียบเรียบร้อย และมองเห็นข้อบกพร่องได้ก่อนที่จะลงมือทำ

โปรแกรมประเภท CAD ที่นำมาใช้คือโปรแกรม CATIA ของบริษัท Dassault system จำกัด เพราะว่าเป็น โปรแกรมที่ใช้งานง่าย และเป็นการศึกษานอกเหนือจากวิชาที่เรียนในมหาวิทยาลัยอีกด้วย

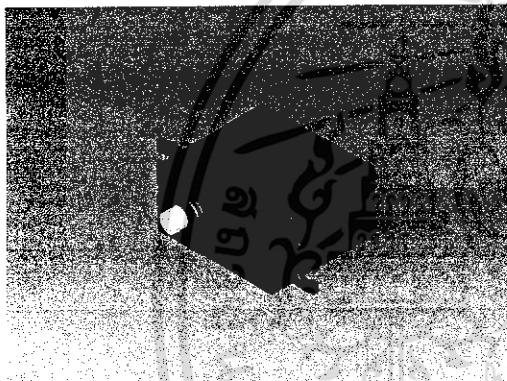
ในปีการศึกษาที่ 2 (พ.ศ.2548-2549)นี้ คณะผู้จัดทำได้ออกแบบ ปรับปรุง และสร้างหุ่นยนต์ทั้งหมดระบบ ซึ่งต่อไปนี้จะอธิบายถึงข้อเสียของหุ่นยนต์แต่ละแบบพร้อมทั้งรูปที่ได้จากการออกแบบโดยโปรแกรม CATIA

4.1 รูปแบบที่ 1

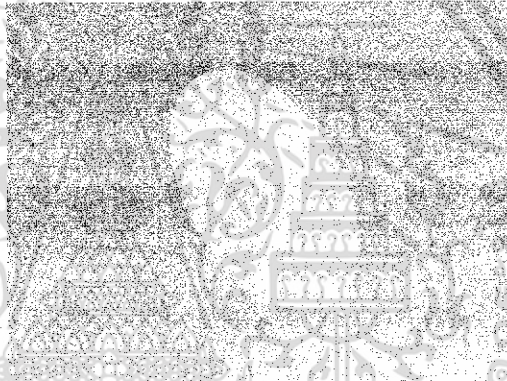
รูปแบบที่นี้เป็นแบบแรกที่คณะผู้จัดทำได้สร้างขึ้น โดยมีระบบขับเคลื่อนแยกกันทั้ง 2 ล้อ ใช้ Servo motor เป็นตัวขับเคลื่อนล้อที่ใช้ตัดแปลงมาจากแผ่น CD เพราะที่ต้องการให้ล้อมีเส้นผ่าศูนย์กลางมากที่สุดเพื่อที่จะชดเชยข้อด้อยของเซอร์โวมอเตอร์ที่มีความเร็วต่ำ รูปที่ 4.1-4.5 แสดงรูปส่วนประกอบของหุ่นยนต์รูปแบบที่ 1



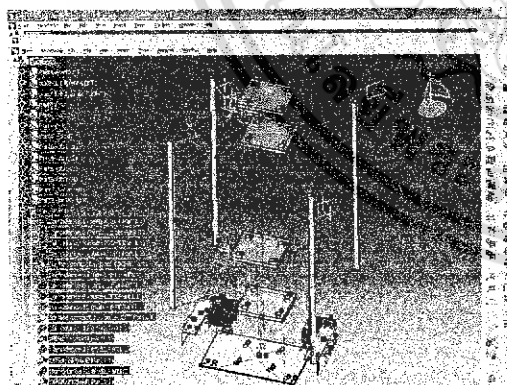
รูปที่ 4.1 หุ่นยนต์รูปแบบที่ 1



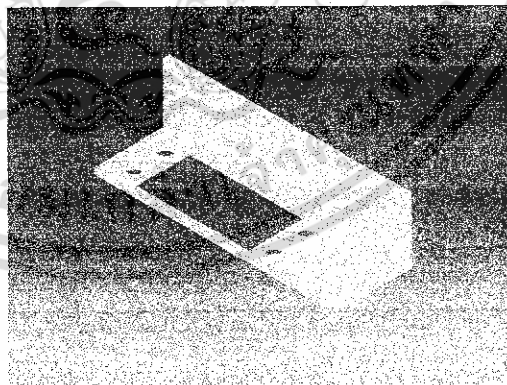
รูปที่ 4.2 แสดงเซอร์โวมอเตอร์



รูปที่ 4.3 แสดงล้อที่ตัดแปลงจากแผ่นCD



รูปที่ 4.4 แสดงหน้าต่างการทำงาน



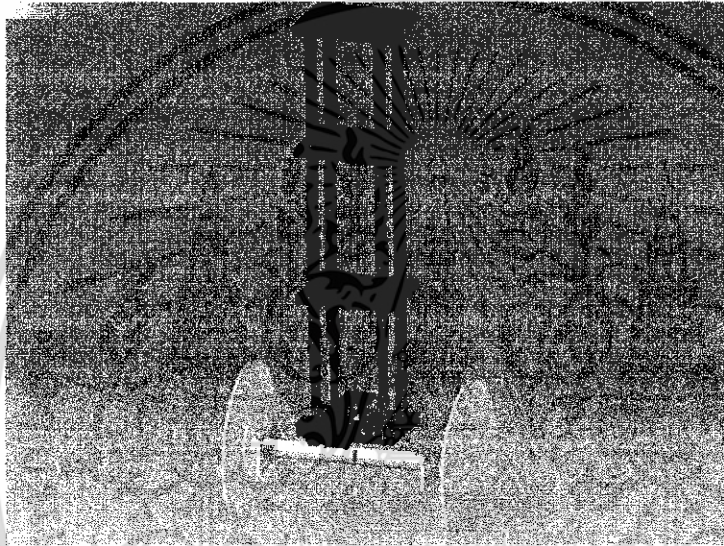
รูปที่ 4.5 แสดงแผ่นติดตั้งเซอร์โวมอเตอร์

ข้อเสียของรูปแบบนี้ก็คือเซอร์โวมอเตอร์มีความเร็วต่ำเกินไปแม้จะชดเชยโดยใช้ล้อที่มีเส้นผ่านศูนย์กลางมากก็ตามผลตอบสนองที่ได้ก็ยังไม่เร็วพออยู่ดี ประกอบกับภายในเซอร์โวมอเตอร์ใช้

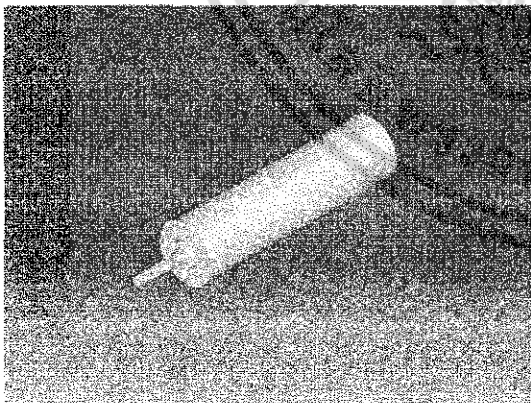
เฟืองที่มีขนาดเล็กทำให้เมื่อมีการเปลี่ยนทิศทางบ่อยทำให้เฟืองเกิดความเสียหาย ไม่สามารถใช้ต่อไปได้

4.2 รูปแบบที่ 2

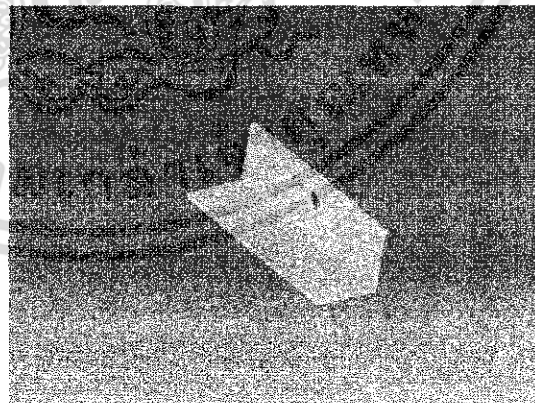
ในรูปแบบนี้ได้ปรับปรุงโดยการเปลี่ยนมอเตอร์ใหม่ แทนที่เซอร์โวมอเตอร์ โดยการใช้มอเตอร์กระแสตรงมีอัตราทดสูงในตัวซึ่งประกอบด้วยเฟืองเหล็กมี Backlash ต่ำ และเนื่องจากเปลี่ยนมอเตอร์จึงต้องมีการเปลี่ยนแผ่นติดตั้งมอเตอร์ด้วย รูปที่ 4.6-4.8 แสดงรูปของหุ่นยนต์รูปแบบที่ 2 รูปมอเตอร์กระแสตรงที่นำมาแทนที่เซอร์โวมอเตอร์ เป็นติดตั้งมอเตอร์ที่ปรับปรุงใหม่



รูปที่ 4.6 แสดงรูปของหุ่นยนต์รูปแบบที่ 2



รูปที่ 4.7 แสดงมอเตอร์ไฟตรง



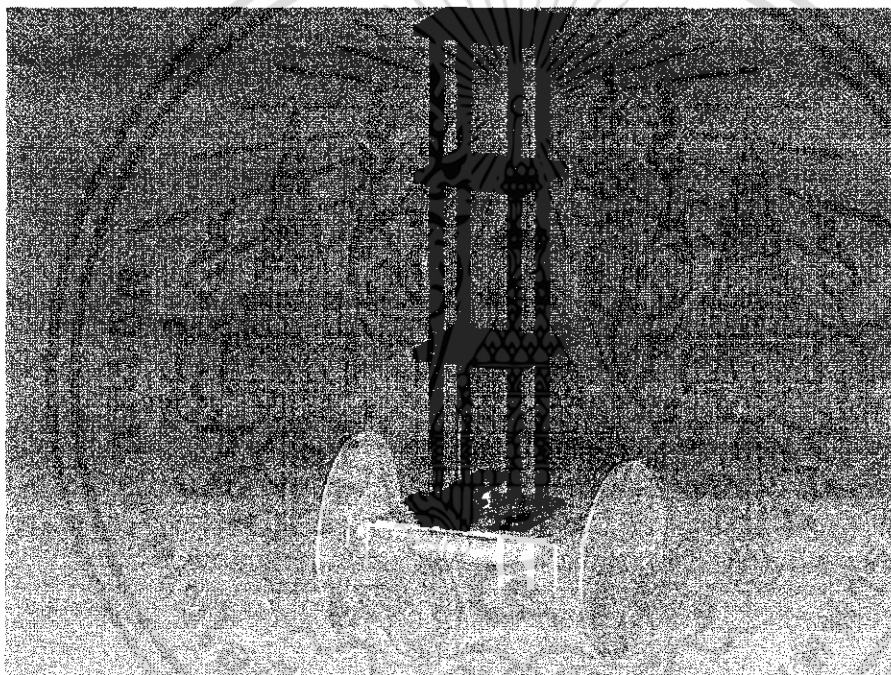
รูปที่ 4.8 แสดงแป้นมอเตอร์

ข้อเสียของระบบนี้ก็ยังคงอยู่ที่มอเตอร์เช่นเดิมในขั้นแรกที่ทำให้การทดลองพบว่ามอเตอร์มีความเร็วต่ำเกินไปซึ่งก็แก้ปัญหาโดยการเพิ่มแรงดันจากเดิม 9 โวลต์เป็น 12 โวลต์ ปัญหาต่อมาคือมอเตอร์ทั้ง

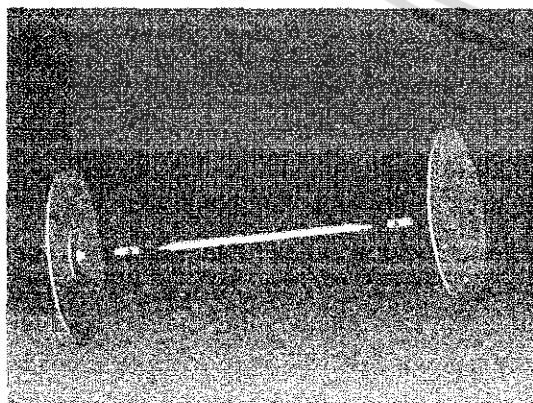
สองตัวหมุนด้วยความเร็วรอบที่ไม่เท่ากันและมีผลตอบสนองไม่เท่ากันทำให้หุ่นยนต์ไม่สามารถที่จะอยู่นิ่งๆ ได้ ด้วยข้อเสียนี้ทำให้ต้องมีการปรับปรุงต่อไป

4.3 รูปแบบที่3

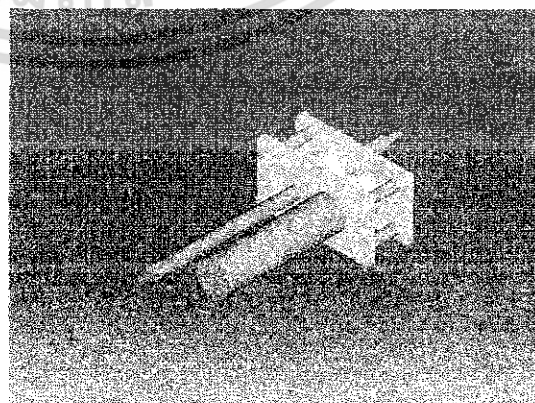
จากปัญหาความไม่สมมาตรของมอเตอร์ที่เกิดขึ้นในรูปแบบที่2ในรูปแบบนี้จึงทำการขุดมอเตอร์ที่ใช้งานให้เหลือเพียงอันเดียว โดยการต่อเพลาร่วมระหว่างล้อทั้งสองและต่อกับมอเตอร์โดยใช้ชุดเฟืองที่มีอัตราทดเท่ากับ1 ทำให้จับล้อทั้งสองได้พร้อมกันและไม่มีปัญหาเรื่องผลตอบสนองไม่เท่ากันอีก รูปที่4.9-4.11 แสดงโครงสร้างหุ่นยนต์ในรูปแบบที่3 การต่อเพลาร่วมกับล้อทั้งสอง และชุดเกียร์ที่ทำขึ้น



รูปที่ 4.9 แสดงหุ่นยนต์รูปแบบที่3



รูปที่ 4.10 แสดงการต่อเพลาร่วม

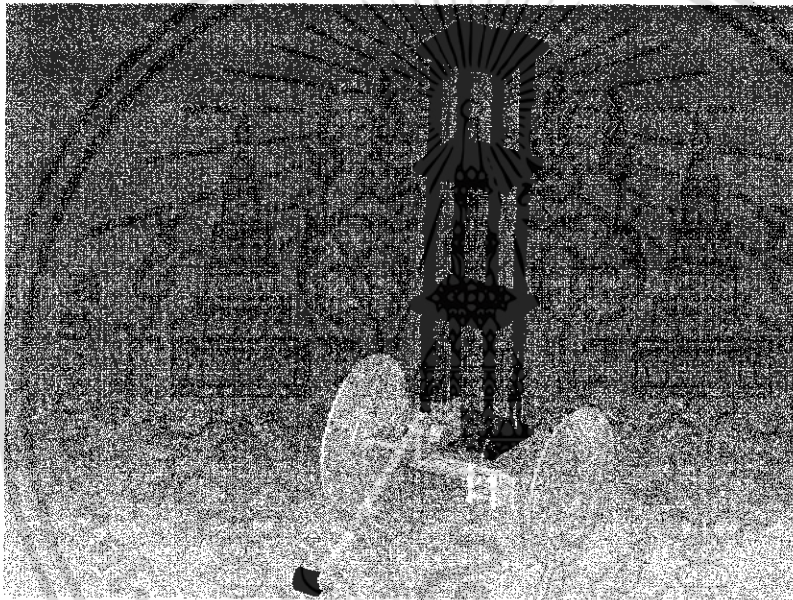


รูปที่ 4.11 แสดงชุดเกียร์ที่ทำขึ้นพร้อมมอเตอร์

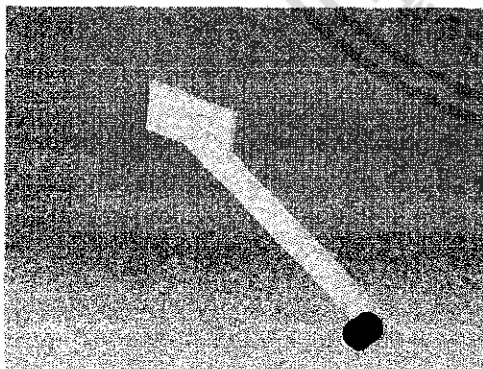
ปัญหาที่เกิดขึ้นกับรูปแบบนี้ก็คือเซ็นเซอร์วัดระยะทางที่ใช้งานอยู่ให้ข้อมูลที่ช้าและไม่แม่นยำพอที่จะตรวจจับการเคลื่อนที่ขนาดเล็กๆ ได้ โครงสร้างทางกลโดยรวมแล้วถือว่าใช้งานได้ดี

4.4 รูปแบบที่ 4

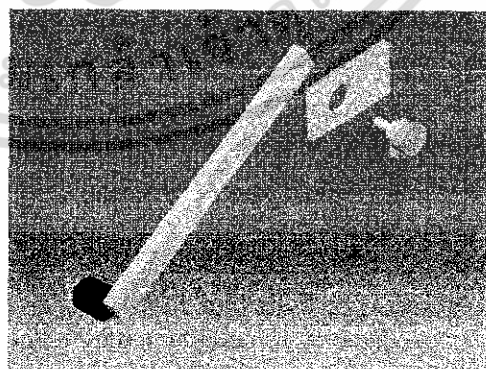
เนื่องจากปัญหาของเซ็นเซอร์ที่ได้กล่าวมาแล้วในตอนแรกก็พยายามที่จะแก้ไขโดยการปรับปรุงทางด้านโปรแกรม ภายหลังจากตรวจสอบพบว่าแท้จริงแล้วเซ็นเซอร์ที่ใช้ให้ข้อมูลออกมาไม่มีการแกว่งตลอดเวลา จึงปรับเปลี่ยนมาใช้ตัวด้านทานปรับค่าได้โดยนำไปติดกับเสาขาที่ยื่นออกไปและกับพื้นมาเป็นเซ็นเซอร์แทนเพราะว่าให้อาชีพทุกที่เป็นสัญญาณอนาลอกเหมือนกันทำให้ไม่เกิดผลกระทบต่อส่วนอื่นมากนักเว้นแต่ค่าอัตราการขยายจะเปลี่ยนไป รูปที่ 4.12-4.14 จะแสดงรูปของหุ่นยนต์รูปแบบที่ 4 เซ็นเซอร์ที่นำมาใช้แทนเซ็นเซอร์วัดระยะทางเดิม



รูปที่ 4.12 แสดงหุ่นยนต์รูปแบบที่ 4



รูปที่ 4.13 แสดงตัวด้านทานปรับค่าได้ยึดติดกับเสาที่ยื่นออกมา

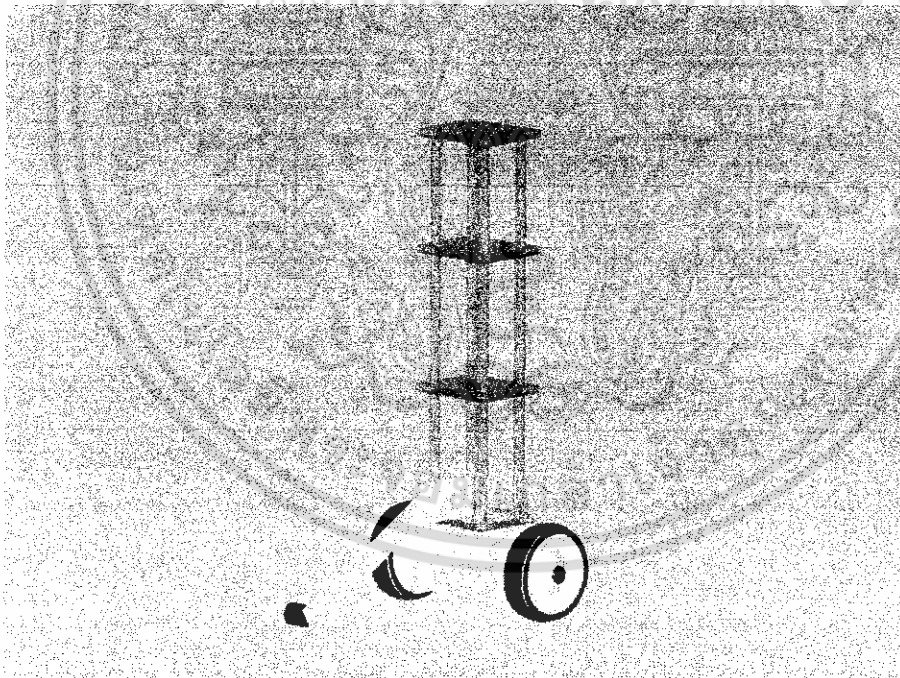


รูปที่ 4.14 แสดงส่วนประกอบของชุดเซ็นเซอร์

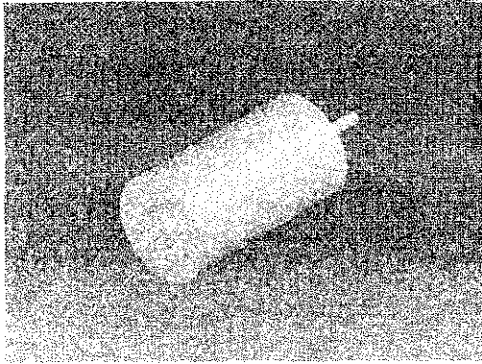
ปัญหาที่เกิดขึ้นของรูปแบบนี้คือส่วนของเพลลาที่ต่อกับล้อนั้นออกแบบมาไม่ดี มีส่วนจับยึดน้อยไปทำให้ล้อหลุดออกได้ง่าย และไม่สามารถหาซื้อเฟืองแบบโลหะที่จะใช้เป็นส่วนยึดกับแกนเพลลาได้ ทำให้ต้องเปลี่ยนไปใช้เฟืองพลาสติกที่มีความหนาของฟันหลายๆแทนแต่ก็ไม่สามารถทำการสร้างเกียร์เพื่อยึดกับแกนเพลลาได้เนื่องจากเฟืองพลาสติกนั้นมีความอ่อนตัวและบางเกินไป จึงใช้การยึดติดกับแกนเพลลาด้วยกาวอีพ็อกซี่แรงยึดสูงแทนซึ่งใช้ได้ผลดีในตอนแรก แต่เมื่อโดนแรงบิดในทิศทางกลับไปที่กลับมาเรื่อยๆทำให้กาวทนแรงไม่ไหว ประกอบกับเพลลาที่ออกแบบมาไม่

4.5 รูปแบบที่ 5

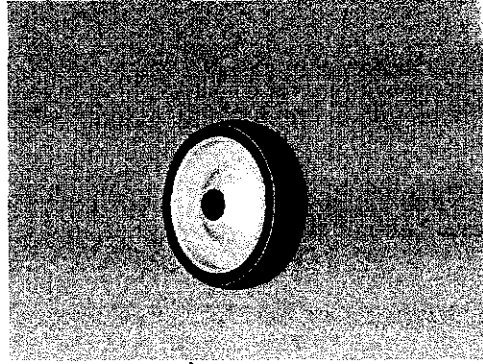
ในโครงสร้างแบบนี้ได้เลือกใช้มอเตอร์เพลลาคู่ซึ่งเป็นมอเตอร์ไฟตรงแรงดัน 12 โวลต์ ไม่มีการทดรอบ มีเพลลาขึ้นออกมาทั้งสองข้างเหมาะแก่การจับล้อพร้อมกันเพราะไม่ต้องมีอุปกรณ์อื่นใดเพิ่มเติมในการขับเคลื่อนซึ่งอาจเกิดปัญหา back lash และมีการเสียหายในภายหลังได้ และเนื่องจากมอเตอร์นี้เป็นมอเตอร์ที่มีรอบสูงอยู่แล้วจึงทำการเปลี่ยนล้อให้มีขนาดเส้นผ่านศูนย์กลางน้อยลงและมีความกว้างของล้อมากขึ้นด้วยเพื่อให้หุ่นยนต์มีเสถียรภาพดีขึ้น รูปที่ 4.15-4.20 แสดงหุ่นยนต์รูปแบบที่ 5, มอเตอร์เพลลาคู่, ล้อ ตัวยึดล้อและแป้นยึดมอเตอร์



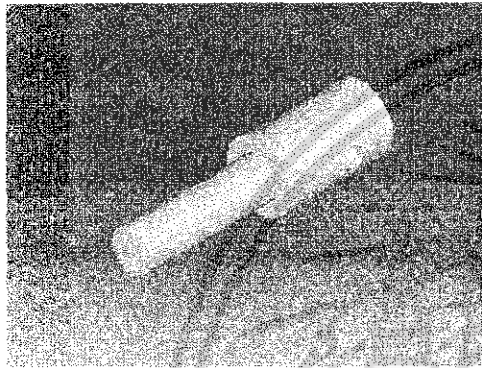
รูปที่ 4.15 แสดงหุ่นยนต์รูปแบบที่ 5



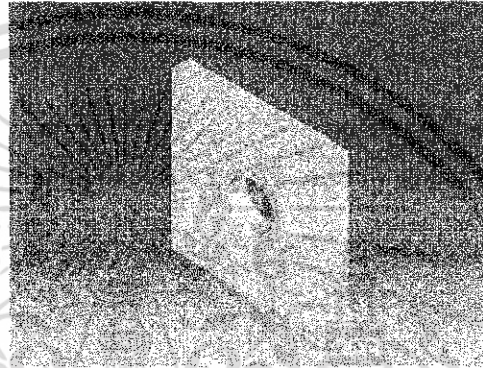
รูปที่ 4.16 แสดงมอเตอร์เพลาคู่



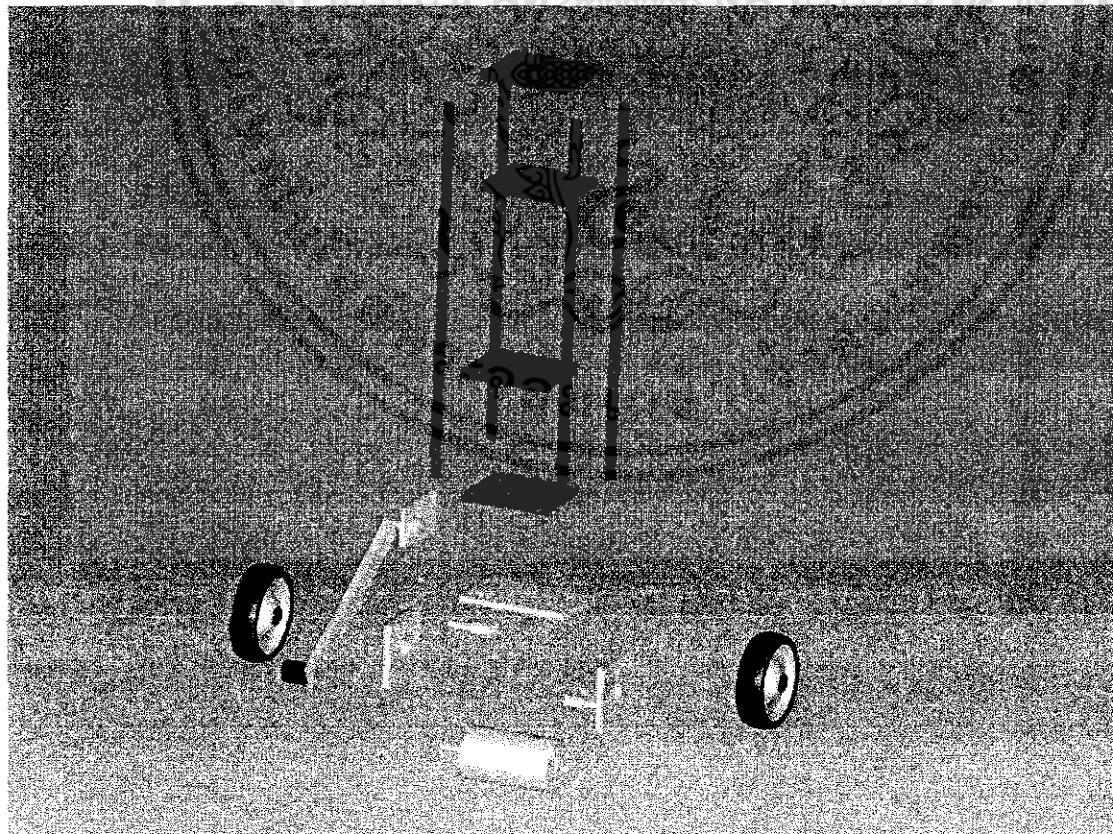
รูปที่ 4.17 แสดงล้อ



รูปที่ 4.18 แสดงตัวยึดล้อ

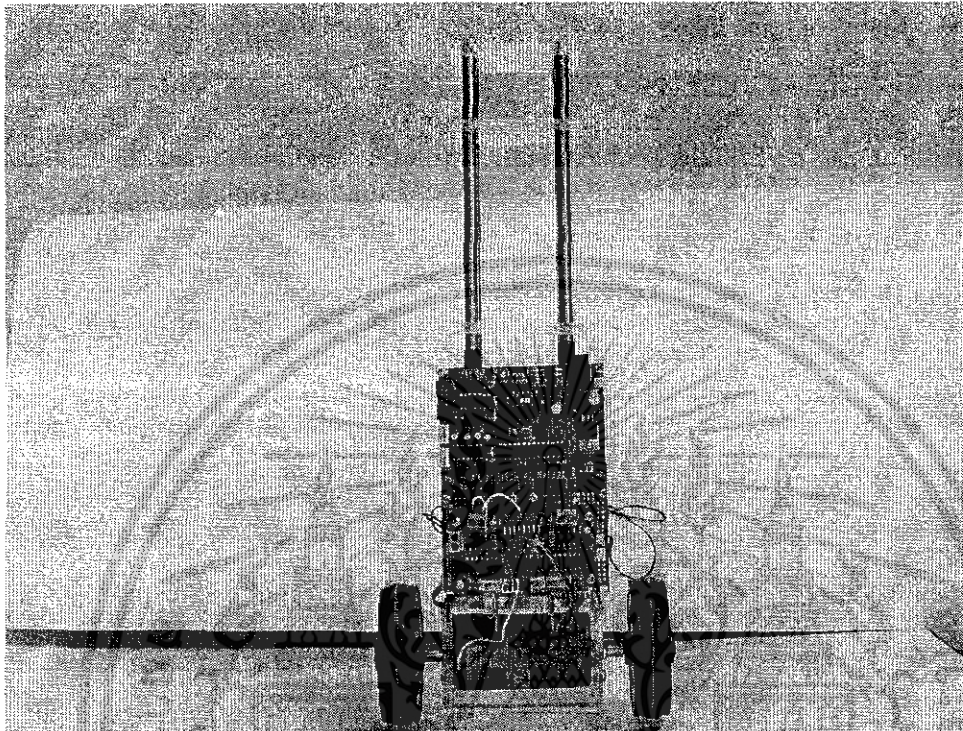


รูปที่ 4.19 แสดงแป้นยึดมอเตอร์

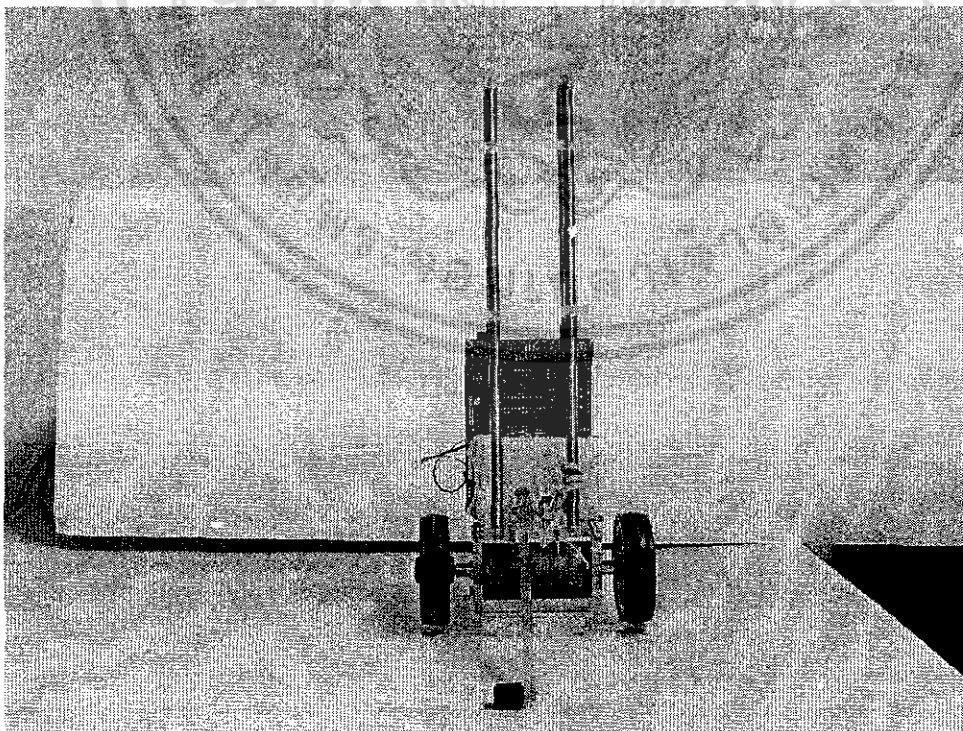


รูปที่ 4.20 แสดงหุ่นยนต์รูปแบบที่ 5 ด้วยการกระจายออก

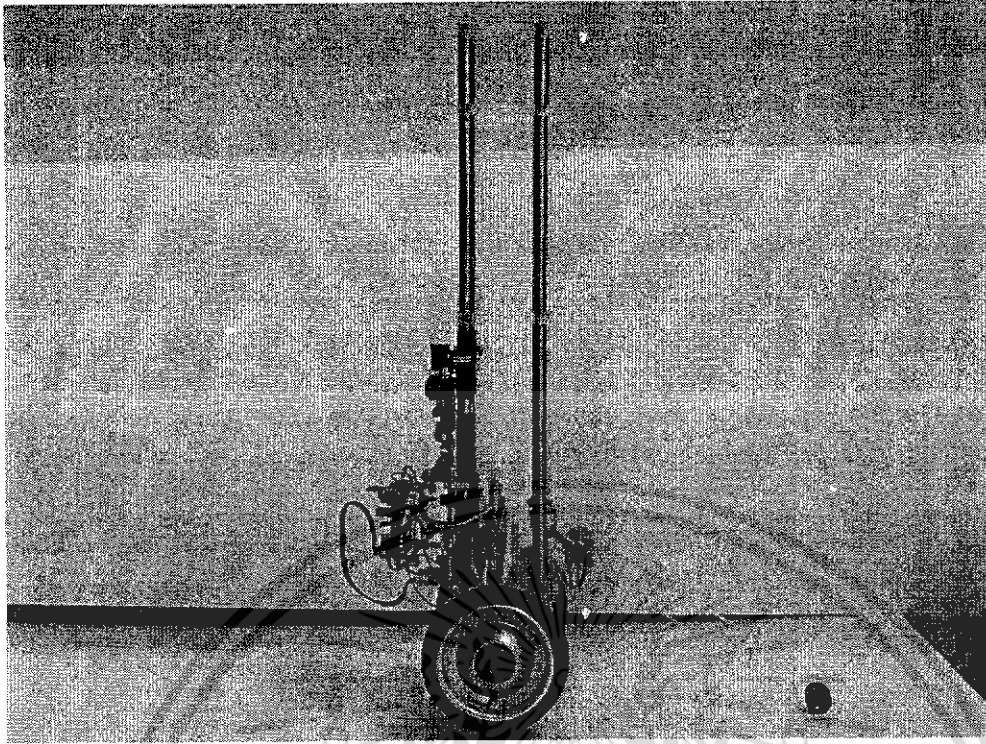
รูปแบบนี้เป็นรูปแบบสุดท้ายในการทำโครงงานนี้ซึ่งนับได้ว่าเป็นแบบที่ประกอบได้ง่าย
 ทนทานและมีปัญหาน้อยมาก



รูปที่ 4.21 แสดงภาพจริงด้านหน้าของหุ่นยนต์



รูปที่ 4.22 แสดงภาพจริงของด้านหลังหุ่นยนต์



รูปที่ 4.23 แสดงภาพจริงด้านข้างของหุ่นยนต์



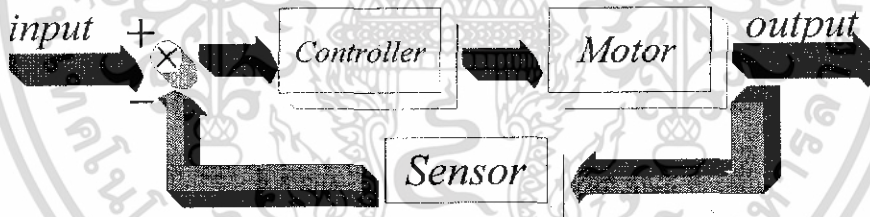
บทที่ 5

หลักการในการควบคุมหุ่นยนต์และทฤษฎีของระบบควบคุม

หลักการในการควบคุมหุ่นยนต์และระบบควบคุมจะกล่าวถึง หลักการในการควบคุมหุ่นยนต์รักษาสมดุลได้ด้วยตนเองเพื่อให้รู้จักฝั่งการทำงานก่อน จากนั้นจะเป็นเรื่องของการศึกษาตัวควบคุมที่คาดว่าจะสามารถทำให้หุ่นยนต์สามารถรักษาสมดุลได้

5.1 หลักการในการควบคุมหุ่นยนต์

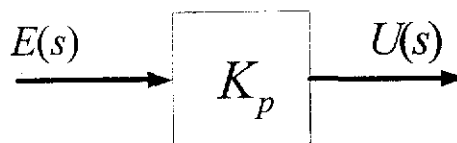
หลักในการควบคุมให้ระบบรักษาสมดุลอยู่ได้ จะต้องควบคุมมอเตอร์เพื่อบังคับล้อให้เคลื่อนที่ไปยังทิศทางและความเร็วที่ต้องการ เพื่อควบคุมให้จุดศูนย์กลางมวล เคลื่อนที่เข้าสู่ค่าที่อ้างอิงไว้ ซึ่งการควบคุมดังกล่าว เกิดจากการทำงานร่วมกันระหว่างไมโครคอนโทรลเลอร์ ชุดควบคุมมอเตอร์ และเซนเซอร์ โดยเซนเซอร์จะวัดค่ามุมของหุ่นยนต์ ในขณะนั้น จากนั้นจะส่งค่าไปยังหน่วยประมวลผล(controller) เพื่อให้ หน่วยประมวลผล(controller) ส่งสัญญาณควบคุมมอเตอร์ไปยังชุดควบคุมมอเตอร์เพื่อควบคุมให้ระบบเคลื่อนที่ได้ตำแหน่งของมุมที่ต้องการ การทำงานโดยรวมของระบบควบคุมแสดงดังรูป



รูปที่ 5.1 แสดงการทำงานโดยรวมของระบบ

5.2 ตัวควบคุมแบบสัดส่วน(P Controller)

ตัวควบคุมแบบสัดส่วนจะประกอบไปด้วยพารามิเตอร์เดียว คือค่าอัตราส่วนหรืออัตราขยาย โดยมีสัญญาณอินพุต เป็นสัญญาณแตกต่าง $E(s)$ (error signal) ระหว่างสัญญาณอ้างอิง (set-point , reference signal) กับสัญญาณป้อนกลับจากเอาต์พุตของระบบ $U(s)$ (actuating signal)



P-controller

รูปที่ 5.2 แสดงแผนผังการทำงานของตัวควบคุมแบบสัดส่วน

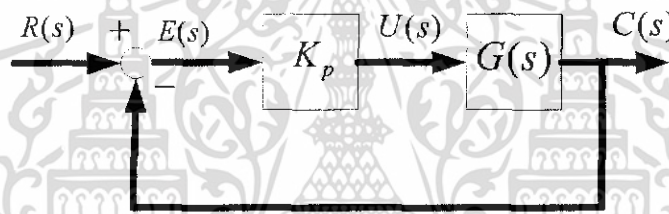
ความสัมพันธ์ระหว่างสัญญาณเอาต์พุตและสัญญาณแตกต่างที่ส่งเข้าไปในระบบ สามารถเขียนได้ดังนี้

$$u(t) = K_p e(t) \quad (5.1)$$

สามารถเขียนเป็นฟังก์ชันถ่ายโอนของตัวควบคุมแบบสัดส่วนได้ดังนี้คือ

$$\frac{U(s)}{E(s)} = K_p \quad (5.2)$$

โดยที่ K_p คือ ค่าอัตราขยายของตัวควบคุมแบบสัดส่วน เมื่อนำตัวควบคุมแบบสัดส่วนนี้ไปเป็นตัวควบคุมในระบบวงปิด จะมีลักษณะดังรูป (5.3)



รูปที่ 5.3 แสดงระบบวงปิดของการใช้งานตัวควบคุมแบบสัดส่วน

จากระบบในรูป(5.3) สามารถเขียนฟังก์ชันถ่ายโอนได้ดังนี้

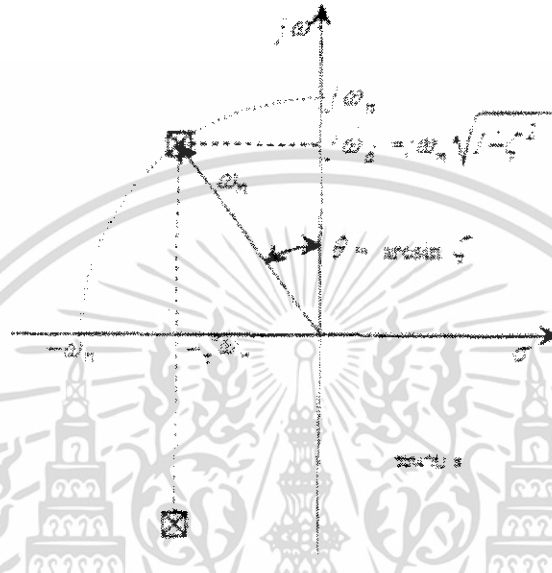
$$\frac{C(s)}{R(s)} = \frac{K_p G(s)}{1 + K_p G(s)} \quad (5.3)$$

จากสมการที่(5.3) ตำแหน่งของโพลวงปิดจะขึ้นกับค่า K_p เพราะค่า K_p ที่เปลี่ยนไปนี้ จะทำให้รากของสมการคุณลักษณะเปลี่ยนไปได้ การวิเคราะห์ว่าเส้นทางรากของสมการคุณลักษณะ หรือตำแหน่งโพลในระนาบ s ของระบบวงปิดจะเปลี่ยนแปลงไปในทิศทางใดเมื่อค่า K_p เปลี่ยนไป เรียกว่า การวิเคราะห์ทางเดินราก (Root Locus Method)

ในการออกแบบเลือกค่า K_p ที่เหมาะสมนั้น เราต้องดูว่าเส้นทางเดินรากลนั้นเดินไปในทิศทางใด โดยทั่วไป มักจะปรับค่า K_p จนกระทั่งโพลของระบบวงปิด มาอยู่ในตำแหน่งที่มีค่า อัตราส่วนการหน่วง(damping ratio) ตามที่ต้องการ แต่เนื่องจากอัตราส่วนการหน่วงของระบบจะมีความสัมพันธ์กับค่าเปอร์เซ็นต์ค่าพุ่งเกิน(percent overshoot) ดังนั้น ในบางครั้ง การบอกข้อกำหนดการออกแบบ

อาจจะบอกในรูป ค่าเปอร์เซ็นต์ค่าพุงเกินของระบบวงปิดมาให้ และเราต้องปรับค่า K_p จนกระทั่งโพลของระบบวงปิดมีค่าอัตราส่วนการหน่วงสัมพันธ์กับค่าเปอร์เซ็นต์ค่าพุงเกินนั้น

ความสัมพันธ์ระหว่างตำแหน่งโพลสังยุค(conjugate poles) ของระบบอันดับที่ 2 ที่ไม่มีซีโร กับค่าอัตราส่วนการหน่วงสามารถแสดงได้ดังรูป(5.4)



รูปที่ 5.4 แสดงความสัมพันธ์ระหว่างตำแหน่งโพลกับค่าอัตราส่วนการหน่วง

ผลตอบสนองของระบบวงปิด ที่มีโพลมากกว่า 2 โพลจะสามารถประมาณลักษณะตอบสนองได้ด้วยโพลเด่น(dominant pole) 2 ตัว ในกรณีที่ส่วนจริงของโพลอื่นๆ นั้นอยู่ไกลจากแกนจินตภาพ เป็นระยะทางมากกว่า 10 เท่า(หรือ 6 เท่า)ของระยะทางของโพลเด่นจากแกนจินตภาพ

ถ้าข้อกำหนดเกี่ยวกับ โพลเด่นนี้เป็นจริง จะสามารถหาผลตอบสนองของระบบต่อสัญญาณอินพุตแบบขั้นบันได ได้เท่ากับ

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \cos(\omega_n \sqrt{1-\zeta^2} t + \phi) \quad (5.4)$$

โดยที่

$$\phi = \tan^{-1} \frac{\zeta}{\sqrt{1-\zeta^2}}$$

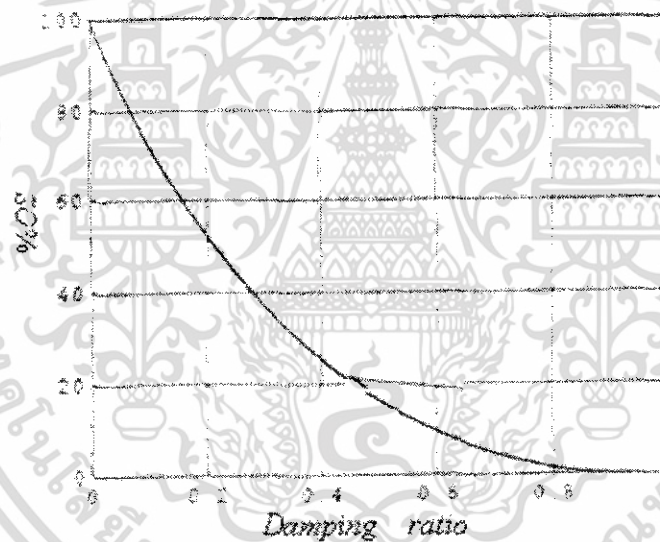
เปอร์เซ็นต์ค่าพุ่งเกินของสมการผลตอบสนองที่(6.4) หาได้จากนิยามค่าเปอร์เซ็นต์ค่าพุ่งเกินดังนี้

$$\%OS = \frac{c_{\max} - c_{\text{final}}}{c_{\text{final}}} \times 100 \quad (5.5)$$

ค่าอัตราส่วนความหน่วงที่สัมพันธ์กับเปอร์เซ็นต์ค่าพุ่งเกินที่ต้องการมีค่าเท่ากับ

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \quad (5.6)$$

ความสัมพันธ์ในสมการ(5.6) จะสามารถพล็อตได้ดังรูป(5.5)



รูปที่ 5.5 แสดงความสัมพันธ์ระหว่างความหน่วงที่สัมพันธ์กับเปอร์เซ็นต์ค่าพุ่งเกิน

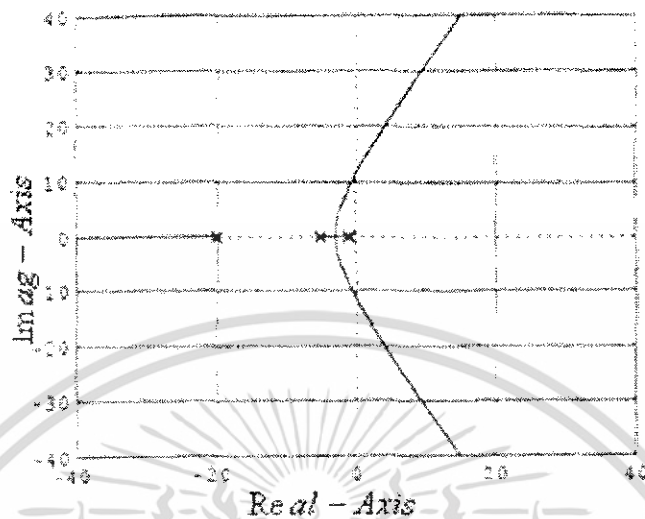
เส้นทางรากของระบบวงปิด สามารถหาได้จากสมการลักษณะเฉพาะ

$$1 + K_p G(s) = 0 \quad (5.7)$$

เมื่อจัดสมการลักษณะเฉพาะให้เป็น

$$1 + K_p \frac{p(s)}{q(s)} = 0 \quad (5.8)$$

จากสมการ(5.8) จะสามารถหาเส้นทางเดินรากเมื่อกำหนดค่า K_p เปลี่ยนไป ได้ดังรูป(5.6)



รูปที่ 5.6 แสดงตัวอย่างเส้นทางราก

เมื่อโพลของระบบวงปิดขยับไปบนเส้นทางรากโดยปรับค่า K_p จนกระทั่งโพลของระบบวงปิดอยู่ในเส้นอัตราส่วนการหน่วง ที่เราต้องการ(จากการคำนวณด้วยสมการที่(5.6)) ผลตอบสนองของระบบวงปิดนั้นจะมีเปอร์เซ็นต์ค่าพุงเกินได้ตามต้องการ

5.3 ตัวควบคุมแบบสัดส่วนบวกอินทิกรัล(PI Controller)

ในการออกแบบตัวควบคุมแบบสัดส่วนนั้น ตัวควบคุมแบบสัดส่วนจะสามารถปรับปรุงผลตอบสนองทรานเซียนท์ได้บางส่วน(ช่วงผลตอบสนองทรานเซียนท์ คือ ช่วงที่ผลตอบสนองเริ่มตอบรับสัญญาณอินพุทก่อนเข้าสู่สถานะอยู่ตัว) แต่ระบบยังมีปัญหาของค่าผิดพลาดที่สถานะคงตัว (steady-state error) อยู่ ซึ่งเกิดขึ้นเนื่องมาจากตัวควบคุมแบบสัดส่วนเอง ซึ่งถ้าต้องการจะลดค่าผิดพลาดที่สถานะอยู่ตัวนั้น สามารถทำได้โดยเพิ่มอัตราขยาย K_p ให้มีค่าสูงขึ้น แต่ในความเป็นจริงแล้วอาจจะทำไม่ได้เนื่องจากมีข้อจำกัดของฮาร์ดแวร์ ดังนั้น โดยส่วนมากแล้วปัญหาของค่าผิดพลาดที่สถานะอยู่ตัวจะไขว้วิธีแก้ไขได้โดยใช้ตัวควบคุมแบบสัดส่วนบวกอินทิกรัล

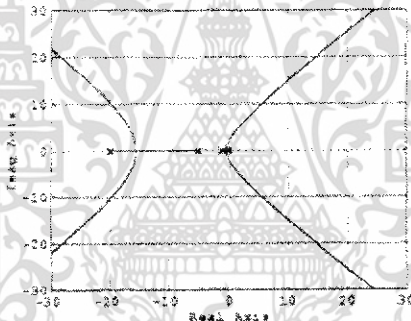
ในการเพิ่มอินทิเกรเตอร์นั้นสามารถทำได้โดยให้ทรานสเฟอร์ฟังก์ชันของ ตัวควบคุมเป็นแบบ

$$\frac{K}{s}$$

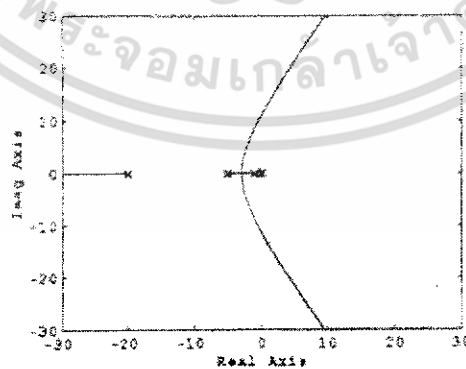


รูปที่ 5.7 แสดงแผนผังการทำงานของตัวควบคุมแบบสัดส่วนบวกอินทิกรัล

แต่การเพิ่มอินทิเกรเตอร์เปล่านั้นเป็นการเพิ่มโพลที่จุดกำเนิดของระนาบ s จะทำให้รูปแบบเส้นทางรากของระบบผิดเพี้ยนไป จะเป็นการส่งผลกระทบต่อผลตอบสนองทรานเซียนท์ เพราะเส้นทางรากจะเดินออกจากโพลวิ่งเข้าหาซีโร ถ้ามีโพลเพิ่มขึ้นมาที่จุดกำเนิด จะเป็นการเพิ่ม system type ให้กับระบบเส้นทางเดินรากจะต้องรวมผลของโพลตัวนี้ด้วย ดังนั้นเพื่อให้การเพิ่ม system type ด้วยการเพิ่มโพลที่จุดกำเนิดนี้ไม่ส่งผลกระทบต่อผลตอบสนองทรานเซียนท์ จึงจำเป็นต้องใส่ซีโรใกล้เคียงกับโพลที่จุดกำเนิดนี้ด้วยเพื่อไม่ให้เส้นทางรากเดิม และผลตอบสนองทรานเซียนท์ผิดเพี้ยนไปมากนัก



รูปที่ 5.8 แสดงเส้นทางเดินรากของระบบ type 0 หลังจากการเพิ่ม โพลแล้ว



รูปที่ 5.9 แสดงเส้นทางเดินรากหลังจากมี โพลที่จุดกำเนิดและซีโรที่ $s = -0.1$

จากรูปที่ 5.9 ฟังก์ชันถ่ายโอนวงเปิดเป็น type 1 เพราะมีโพลที่จุดกำเนิด ทำให้มีค่าความผิดพลาดที่สถานะคงตัว สำหรับสัญญาณอ้างอิงแบบขั้นบันได แต่ในขณะเดียวกันเส้นทางเดินรากก็ไม่ได้ผิผื่นไปจากกรณีการออกแบบตัวควบคุมแบบสัดส่วนมากนัก ทำให้ยังคงสามารถกำหนดลักษณะของการตอบสนองทรานเซียนท์ได้ดังเดิม

ดังนั้นการออกแบบตัวควบคุมแบบสัดส่วนบวกอินทิกรัล จึงไม่ใช่เพียงแค่ใส่ อินทิเกรเตอร์เพิ่มเข้าไปเพียงอย่างเดียว จากการวิเคราะห์ข้างต้นพบว่าเราต้องใส่ซีโรเข้าไปด้วย แล้วจึงค่อยเปลี่ยนค่าอัตราขยายถ่ายโอนของตัวควบคุมแบบสัดส่วนบวกอินทิกรัลสามารถเขียนได้เป็น

$$\frac{U(s)}{E(s)} = K_p \frac{s + \alpha}{s} \quad (5.9)$$

โดยที่ $s = -\alpha$ คือ ตำแหน่งของซีโรที่ต้องใส่เข้าไปพร้อมกับโพลที่จุดกำเนิด

5.4 ตัวควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์ (PD Controller)

การควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์ คือ การควบคุมที่ใช้ ตัวควบคุมแบบสัดส่วน และตัวควบคุมแบบอนุพันธ์ร่วมกัน ซึ่งตัวควบคุมแบบอนุพันธ์นั้นจะช่วยเพิ่มค่าความหน่วง (damping) ให้กับระบบที่ต้องการจะควบคุม นั่นคือ ทำให้ระบบมีเสถียรภาพมากขึ้น แต่โดยทั่วไปแล้วตัวควบคุมเชิงอนุพันธ์จะไม่ทำให้ค่าความผิดพลาดในสถานะอยู่ตัวมีค่าเป็นศูนย์ได้ เนื่องจากสัญญาณเอาต์พุตที่ได้มาจากตัวควบคุมเชิงอนุพันธ์นั้นเกิดจากการหาอนุพันธ์ของสัญญาณผิดพลาด $\frac{d(e(t))}{dt}$ (เมื่อ $e(t)$ คือสัญญาณผิดพลาดของระบบ) ดังนั้นถ้าค่าสัญญาณผิดพลาดนี้มีค่ามาก สัญญาณเอาต์พุตที่ออกมาจากตัวควบคุมแบบอนุพันธ์นี้จะกระเพื่อม (fluctuate) สูง เนื่องจากค่าความชันของสัญญาณเปลี่ยนแปลงค่อนข้างมาก

จากที่กล่าวมาข้างต้นเห็นได้ว่าถ้าใช้ตัวควบคุมแบบอนุพันธ์เพียงอย่างเดียวนั้นอาจทำให้ระบบไม่มีเสถียรภาพได้ จึงจำเป็นต้องใช้ร่วมกับตัวควบคุมแบบสัดส่วน ถ้าการควบคุมแบบสัดส่วนบวกอินทิกรัลปัญหาของค่าผิดพลาดที่สถานะคงตัวลดน้อยลง การควบคุมแบบอัตราส่วนบวกกับแบบอนุพันธ์ก็เป็นการช่วยให้ช่วงผลตอบสนองทรานเซียนท์ของระบบวงปิดมีความรวดเร็วยิ่งขึ้น

ทรานสเฟอ์ฟังก์ชันของตัวควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์นั้นสามารถแสดงได้ดังนี้

$$G_c(s) = K_p(s + z_c) \quad (5.10)$$

จากสมการเห็นได้ว่าตัวควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์เป็นการเพิ่มซีโรที่ตำแหน่ง $s = -z_c$ ให้กับวงเปิด เพื่อปรับรูปของเส้นทางเดินรากให้เดินผ่านจุดที่ต้องการ

5.5 ตัวควบคุมแบบพีไอดี(PID Controller)

การใช้ตัวควบคุมแบบพีไอนั้น จะเป็นการช่วยชดเชยให้ผลตอบสนองทรานเซียนท์ดีขึ้น แต่อาจจะมีค่าผิดพลาดที่สถานะอยู่ตัว จึงจำเป็นต้องใช้ตัวควบคุมแบบอินทิกรัล ในการแก้ปัญหาที่สถานะอยู่ตัว ซึ่งตัวควบคุมแบบนี้จะรวมเรียกว่า ตัวควบคุมแบบพีไอดี

ดังนั้นตัวควบคุมแบบพีไอดี คือตัวควบคุมที่รวมผลการควบคุมทั้งของส่วนของการควบคุมแบบสัดส่วน การควบคุมอินทิกรัล รวมทั้งการควบคุมแบบอนุพันธ์ เพื่อให้ผลตอบสนองของระบบวงปิดมีค่าความผิดพลาดที่สถานะอยู่ตัวเป็นศูนย์ มีผลตอบสนองทรานเซียนท์ที่เร็ว โดยที่ตัวควบคุมพีไอดีมีทรานเฟอร์ฟังก์ชันดังนี้

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (5.11)$$

หรือ

$$G_c(s) = \frac{K_p s + K_i + K_d s^2}{s} = \frac{K_d \left(s^2 + \frac{K_p s}{K_d} + \frac{K_i}{K_d} \right)}{s} \quad (5.12)$$

บทที่ 6

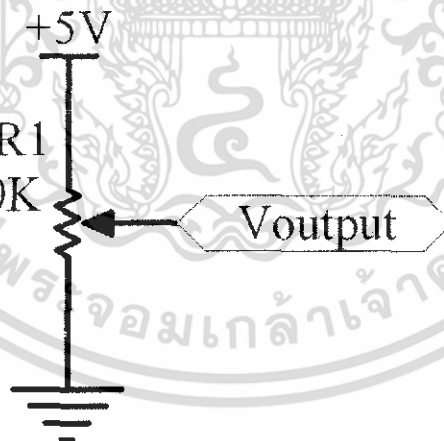
การทดลองและผลการทดลอง

จากระบบควบคุมหุ่นยนต์รักษาสมดุลที่ได้ออกแบบและได้ทำการประกอบขึ้นมา ในบทนี้จะอธิบายถึงการทดลองต่างๆที่ได้จากการทดลองจริง เพื่อศึกษาผลตอบสนอง และทดสอบเสถียรภาพของระบบ

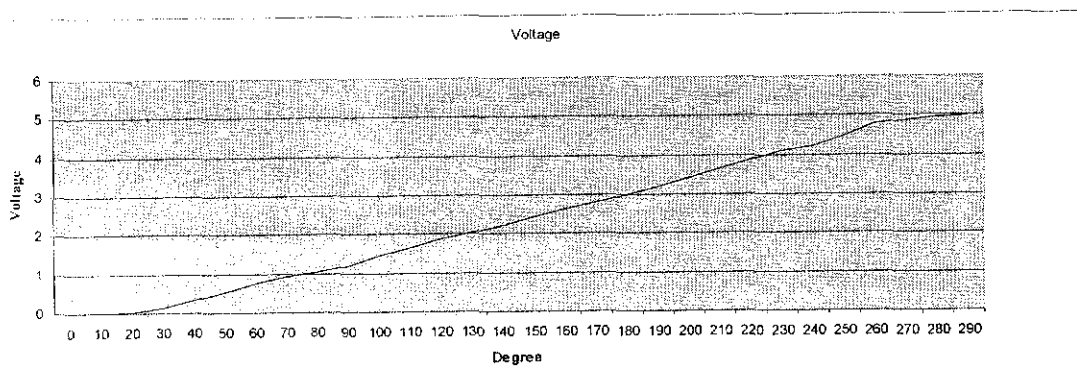
6.1 การทดลองระหว่างองศาที่เปลี่ยนไปของตัวต้านทานปรับค่าได้กับแรงดันไฟฟ้า

โครงงานนี้ใช้ตัวต้านทานปรับค่าได้เป็นเซ็นเซอร์ในการตรวจวัดองศาของหุ่นยนต์เทียบการระนาบของพื้น ดังนั้นจึงจำเป็นที่จะต้องรู้ว่าตัวต้านทานปรับค่าได้ที่ใช้ในโครงงานนี้มีช่วงการใช้งานเป็นมุมเท่าใด และแต่ละช่วงการใช้งานมีค่าผลลัพธ์ที่ได้เป็นเช่นไรบ้าง

ในการทดลองใช้การเพิ่มมุมของตัวต้านทานปรับค่าได้ครั้งละ 10 องศา และวัดค่าแรงดันเอาต์พุตที่ได้ แล้วนำมาพล็อตกราฟเพื่อแสดงความสัมพันธ์ระหว่างมุมที่เปลี่ยนไปของตัวต้านทานปรับค่าได้และแรงดันเอาต์พุต วงจรที่ใช้ในทดลองแสดงดังรูปที่ 6.1 ผลการทดลองที่ได้แสดงในรูปที่ 6.2 และตารางที่ 1



รูปที่ 6.1 แสดงวงจรต่อทดลอง



รูปที่ 6.2 แสดงกราฟระหว่างแรงดันเอาต์พุตและมุมของตัวต้านทานปรับค่าได้

องศา (Deg)	แรงดัน (โวลต์)	องศา (Deg)	แรงดัน (โวลต์)
0	0	150	2.43
10	0.01	160	2.62
20	0.04	170	2.81
30	0.13	180	2.99
40	0.34	190	3.2
50	0.542	200	3.42
60	0.76	210	3.65
70	0.92	220	3.89
80	1.06	230	4.1
90	1.2	240	4.23
100	1.45	250	4.48
110	1.65	260	4.77
120	1.86	270	4.88
130	2.03	280	4.96
140	2.22	290	4.98

ตารางที่ 1 แสดงผลลัพธ์ที่ได้จากการทดลอง

จากการทดลองแสดงให้เห็นว่าตัวต้านทานปรับค่าได้มีมุมการใช้งานตั้งแต่ 0 องศา ถึง 290 องศา และแรงดันเอาต์พุตในช่วง 70 องศา ถึง 230 องศา สามารถประมาณให้เป็นเชิงเส้นได้

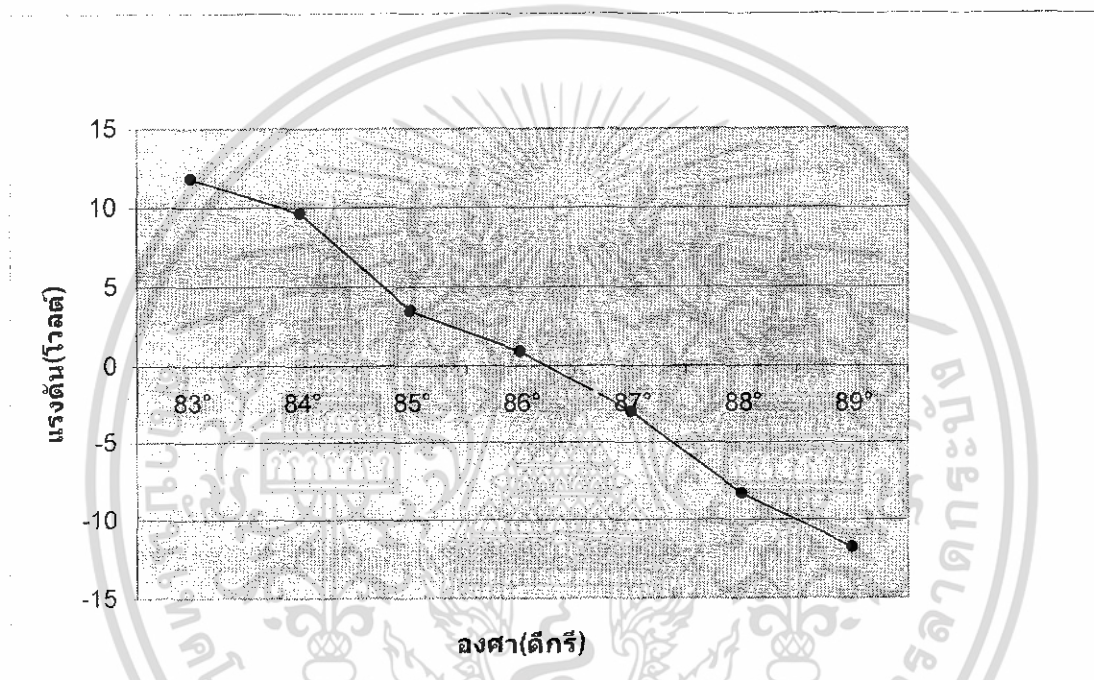
6.2 การทดลองหาความสัมพันธ์ระหว่างองศาที่เปลี่ยนแปลงไปจากสมมุติและแรงดันที่จ่ายให้มอเตอร์

เพื่อให้ทราบได้ว่าหุ่นยนต์รักษาสมมุติที่สร้างขึ้นสามารถมีการเปลี่ยนแปลงทางมุมได้เท่าใด จึงออกแบบการทดลองเพื่อหาความสัมพันธ์ระหว่างองศาที่เปลี่ยนแปลงไปจากสมมุติและแรงดันที่จ่ายให้มอเตอร์ซึ่งแรงดันนี้จะแปรผันตรงกับกับความเร็วของมอเตอร์ซึ่งได้ทดลองโดยวัดมุมของหุ่นยนต์ขณะที่อยู่ในสมมุติพร้อมกับค่าแรงดันที่จ่ายให้มอเตอร์ จากนั้นก็เพิ่มมุมที่หุ่นยนต์ทำมุมกับพื้นเพิ่มขึ้นหรือลดลงจากมุมที่สมมุติแล้วจึงวัดค่าแรงดันว่ามีการเปลี่ยนแปลงอย่างไร หาก

ค่าแรงดันถึงจุดที่สามารถจ่ายให้มอเตอร์ได้สูงสุดแล้ว แสดงว่ามุมของหุ่นยนต์ขณะนั้นเป็นมุมมากที่สุดที่จะสามารถรักษาสมดุลย์อยู่ได้ หากทำมุมเอียงเพิ่มมากกว่าค่านี้มอเตอร์จะไม่สามารถให้ความเร็วที่เพียงพอที่จะรักษาสมดุลย์อยู่ได้ ผลการทดลองแสดงดังตารางที่2 และรูปที่6.3

องศา((Deg))	83	84	85	86	87	88	89
แรงดัน(โวลต์)	11.8	9.62	3.46	0.89	-3.04	-8.36	-11.8

ตารางที่2 แสดงผลการทดลองหาความสัมพันธ์ระหว่างองศาที่เปลี่ยนแปลงไปจากสมดุลย์และแรงดันที่จ่ายให้มอเตอร์



รูปที่6.3 แสดงกราฟระหว่างองศาและแรงดันที่ได้จากการทดลอง

จากผลการทดลองเห็นได้ว่าองศาที่หุ่นยนต์ทำมุมกับระนาบเป็นมุม86องศาจะมีค่าแรงดันน้อยที่สุดและเป็นมุมที่หุ่นยนต์กำลังอยู่ในสมดุลย์ และเมื่อมีค่ามุมเพิ่มขึ้นจากจุดที่สมดุลย์แรงดันที่มอเตอร์จะมีค่าเป็นลบซึ่งหมายความว่ามอเตอร์กำลังหมุนย้อนทิศทางเพื่อที่จะรักษาค่ามุมของหุ่นให้กลับสู่ค่ามุมที่สมดุลย์ และในทางกลับกันหากค่าของมุมมีค่าลดลงจากจุดที่สมดุลย์แรงดันของมอเตอร์จะมีค่าเป็นบวก เพื่อที่จะรักษาค่ามุมให้กลับสู่สมดุลย์เช่นกัน

จากกราฟในรูปที่4.3 ได้เห็นว่ามอเตอร์จะได้รับแรงดันสูงสุด+12โวลต์ และ-12โวลต์ที่มุม83องศาและ89องศาตามลำดับซึ่งเป็นมุมน้อยที่สุดและสูงสุดที่หุ่นยนต์ยังสามารถกลับเข้าสู่สถานะสมดุลย์ได้

6.3 การทดลองหาค่าอัตราขยายที่เหมาะสม

เนื่องจากโพรแกรมนี้ใช้ตัวควบคุมแบบพีไอดี จึงต้องมีการหาค่าอัตราขยายที่เหมาะสมเพื่อให้ระบบมีเสถียรภาพมากที่สุด ซึ่งในโพรแกรมนี้จะใช้วิธีลองผิดลองถูกในการหาค่าอัตราขยาย แนวทางการทดลองคือสุ่มเลือกค่าอัตราขยายขึ้นมาใช้ในระบบควบคุมเรื่อยๆจนได้ค่าที่สามารถทำให้หุ่นยนต์สามารถรักษาสมดุลย์อยู่ได้ แล้วนำค่าที่ใกล้เคียงกับค่านั้นมาทำการทดลองอีกครั้ง เพื่อที่จะทดสอบว่าค่าที่ใช้อยู่นั้นสามารถทำให้ระบบมีเสถียรภาพที่ดีที่สุดแล้วหรือไม่ ซึ่งทดสอบ โดยการใส่แรงกระทำที่รู้ค่าแก่ระบบที่กำลังอยู่ในสมดุลย์และเพิ่มปริมาณของแรงขึ้นเรื่อยๆจนกว่าระบบจะไม่สามารถกลับเข้าสู่สมดุลย์ได้

ในการทดลองนี้ได้ควบคุมปริมาณของแรงโดยใช้ลูกตุ้มเหวี่ยงซึ่งจะมีปริมาณของแรงแปรผันตามมุมที่เพิ่มขึ้นเรื่อยๆ รูปของการทดลองแสดงดังรูปที่ 6.4



รูปที่ 6.4 แสดงรูปการทดลอง

การทดลองนี้จะแยกออกเป็นสองส่วนคือการทดลองที่ 6.3.1 การหาค่าอัตราขยายที่ใช้ในตัวควบคุมแบบพีดี และ การทดลองที่ 6.3.2 การหาค่าอัตราขยายที่ใช้ในตัวควบคุมแบบพีไอดี

6.3.1 การทดลองหาค่าอัตราขยายที่ใช้ในตัวควบคุมแบบพีดี

เนื่องจากระบบที่ใช้เดิมมีค่าอัตราขยายของตัวควบคุมพีดีคือ 4 และ 2 ตามลำดับ ในการทดลองนี้เป็นการทดลองเพื่อหาค่าอัตราขยายอนุพันธ์ เพียงอย่างเดียวเท่านั้น และจะตัดสินใจว่าอัตราขยายที่ทำให้ระบบเข้าสู่สมดุลย์ที่ระยะทางที่น้อยที่สุดและที่แรงกระทำมากที่สุด เป็นอัตราขยายที่ดีที่สุด ซึ่งมีผลการทดลองดังแสดงในตารางที่ 3

ตารางที่ 3 แสดงผลการทดลองหาค่าอัตราขยายอนุพันธ์

K_D	องศา (Deg)	ระยะทาง (cm)
1	10	10
	20	23
	30	28
	40	33
	50	59
	60	-
	70	-
1.5	10	20
	20	26
	30	34
	40	43
	50	50
	60	-
	70	-
2	10	6
	20	17
	30	22
	40	27
	50	42
	60	60
	70	-
2.5	10	7
	20	7
	30	13
	40	33
	50	37
	60	-
	70	-
3	10	4.5
	20	12
	30	17
	40	18
	50	23
	60	-
	70	-

จากการทดลองเห็นได้ว่าค่าอัตราขยายที่ดีที่สุดคือ 2 เนื่องจากสามารถรักษาสมดุลย์ขณะที่มีแรงเนื่องจากมุม 60 องศา มากระทำได้ในขณะที่อัตราขยายอื่นๆ อนุพันธ์ไม่สามารถรักษาสมดุลย์ได้

6.3.2 การหาค่าอัตราขยายที่ใช้ในตัวควบคุมแบบพีไอดี

หลังจากได้ทดลองหาค่าอัตราขยายสำหรับตัวควบคุมแบบพีดีมาแล้วในการทดลองที่ 4.3.1 ต่อไปเป็นการทดลองหาค่าอัตราขยายตัวควบคุมแบบพีไอดีเพื่อที่จะนำมาเปรียบเทียบความ

แตกต่างระหว่างระหว่างตัวควบคุมทั้งสองแบบว่าตัวควบคุมแบบไหนจะสามารถทำให้ระบบมีเสถียรภาพมากกว่า และจากการทดลองที่ 4.3.1 พบว่าอัตราขยายอนุพันธ์เท่ากับ 2 สามารถทำให้ระบบมีเสถียรภาพที่ดีที่สุด ในการทดลองนี้จึงกำหนดให้อัตราขยายอนุพันธ์มีค่าเท่ากับ 2 และทำการเปลี่ยนแปลงค่าอัตราขยายอินทิกรัลเท่านั้น ซึ่งการหาค่าอัตราขยายอินทิกรัลที่ใช้ในระบบควบคุมแบบพีไอดี ได้ผลการทดลองดังนี้

ตารางที่ 4 แสดงผลการทดลองการหาค่าอัตราขยายอินทิกรัล

K_I	องศา(Deg)	ระยะทาง(cm)
1/550	10	2
	20	56
	30	60
	40	80
	50	-
	60	-
	70	-
1/575	10	2
	20	2
	30	4
	40	10
	50	15
	60	18
	70	30
1/600	10	3
	20	5
	30	6
	40	14
	50	20
	60	29
	70	-
1/625	10	3
	20	6
	30	8
	40	15
	50	20
	60	-
	70	-

จากผลการทดลองเห็นได้ว่าค่าอัตราขยายอินทิกรัลที่ดีที่สุดคือ 1/575 เพราะสามารถทำให้ระบบมีเสถียรภาพที่ดีที่สุดเนื่องจากสามารถรักษาสมดุลขณะที่มีแรงจากมุมกระทำ 70 องศาสามารถทำได้ ในขณะที่อัตราขยายอื่นจะทำให้หุ่นยนต์ล้มหรือไม่สามารถรักษาสมดุลได้

บทที่ 7

สรุปและวิเคราะห์ปัญหา

7.1 สรุปผลการดำเนินงาน

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอหุ่นยนต์รักษาสมดุลด้วยตนเอง ให้สามารถรักษาสมดุลย์ได้ แม้เมื่อมีการรบกวนจากแรงภายนอก โดยมีหลักการทำงานแบบป้อนกลับ โดยใช้ตัวต้านทานปรับค่าได้เป็นเซ็นเซอร์ในการวัดมุมที่หุ่นยนต์ทำมุมอยู่กับระนาบพื้นในขณะนั้น โดยแรงดันที่ได้จากเซ็นเซอร์จะถูกนำไปแปลงเป็นสัญญาณดิจิตอลและทำการคำนวณเพื่อที่จะส่งสัญญาณออกไปควบคุมมอเตอร์ การออกแบบโครงสร้างของหุ่นยนต์ในโครงการนี้ได้มาจากการศึกษารายงานและบทความของโครงการที่มีความเกี่ยวข้องกับการรักษาสมดุลเช่น ระบบอินเวอร์ทเพนดูลัม และหุ่นยนต์รักษาสมดุลที่มีผู้เคยทดลองมาก่อนหน้านี้ จึงได้ข้อสรุปว่าหุ่นยนต์ควรมีความสูงที่เหมาะสมและ มอเตอร์ที่ใช้ควรมีแรงบิดสูงเพื่อที่จะสามารถให้ผลตอบสนองที่รวดเร็ว วงจรที่ใช้ในการควบคุมควรมีความเร็วในการประมวลผลสูงเพื่อให้เพียงพอต่อการคำนวณและการแปลงค่าอนาลอกเป็นดิจิตอลให้อยู่ในช่วงเวลาจริง ไม่เช่นนั้นแล้วจะไม่สามารถตอบสนองได้ทันกับความ เป็นไปของระบบ ณ เวลานั้น เมื่อได้ทำการสร้าง และประกอบ โครงสร้างของหุ่นยนต์ รวมถึงวงจร เชนส์จสมบูรณ์

เนื่องจากต้องการรักษาเสถียรภาพของหุ่นยนต์โดยใช้ระบบควบคุมแบบพีดี และแบบพีไอดี จึงทำการทดลองหาค่าอัตราขยายของระบบแบบพีดี และพีไอดี โดยวิธีการลองผิดลองถูกจนได้ค่าอัตราขยายที่ทำให้เกิดเสถียรภาพในขอบเขตที่ยอมรับได้ แล้วจึงทำการทดลองเพื่อหาค่าอัตราขยายที่เหมาะสมที่สุด โดยการทดลองใส่แรงที่ได้จากการแกว่งลูกตุ้มที่มีองศาเริ่มต้นที่แตกต่างกัน ซึ่งจะแปรผันตรงกับแรงที่กระทำต่อหุ่นยนต์ ซึ่งจากการทดลองนี้ทำให้ได้ค่าอัตราขยายที่สามารถทำให้หุ่นยนต์มีเสถียรภาพที่ดีที่สุด

7.2 ปัญหาและแนวทางแก้ไข

เนื่องจากไม่ได้ศึกษาการหาแบบจำลองทางคณิตศาสตร์ ดังนั้นเมื่อมีการเปลี่ยนแปลง โครงสร้างบ่อย จึงทำให้เสียเวลาในการสร้างมากขึ้น และทำให้ไม่สามารถทำการศึกษาและ ออกแบบระบบขณะที่กำลังทำการเปลี่ยนแปลงโครงสร้าง ในขณะเดียวกันเมื่อมีการเปลี่ยนแปลง โครงสร้าง อัตราการขยายต่างๆในระบบย่อมมีการเปลี่ยนแปลง ทำให้ต้องมีการหาค่าอัตราขยายใหม่ทุกครั้งที่มีการเปลี่ยนแปลงโครงสร้าง ดังนั้นแนวทางการแก้ไข ควรจะทำการศึกษาหาแบบจำลองทางคณิตศาสตร์และนำไปออกแบบระบบควบคุมก่อนที่จะมีการสร้างหุ่นยนต์ขึ้นจริง

7.3 แนวทางการพัฒนาในอนาคต

1.) เปลี่ยนตัวควบคุมไปใช้ตัวควบคุมแบบอื่นซึ่งมีการพัฒนาขึ้น มีความสามารถมากขึ้น ซึ่งมีตัวควบคุมที่น่าสนใจอยู่มากมาย โดยอาจใช้โมเดลของระบบเดิม หรือมีการเปลี่ยนโมเดลใหม่ให้ดูน่าสนใจมากขึ้น

2.) เปลี่ยนเซ็นเซอร์ที่ใช้งาน เนื่องจากเซ็นเซอร์ที่สามารถนำมาใช้ในการวัดมุมเอียงนั้นมีอยู่มากมายเช่นการใช้ ไจโรสโคปพร้อมกับเซ็นเซอร์วัดความเร่งและเซ็นเซอร์วัดระยะทาง เป็นต้น



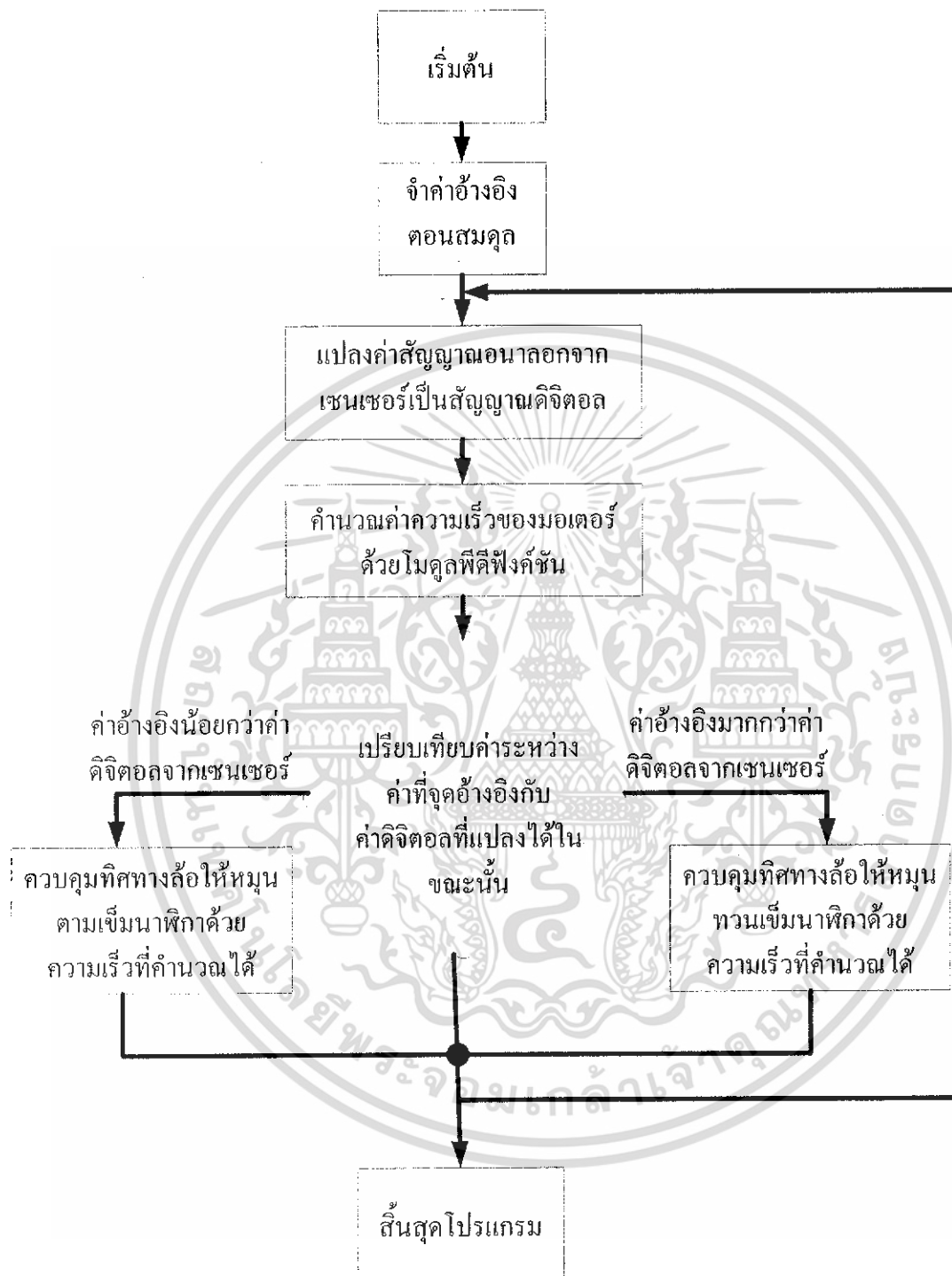
อ้างอิง

1. ทรงกรด ชिरาชัย , ทัตยา ปุคคะฉนันทน์ , นพดล จันทร์คุณภาส ปริญยานิพนธ์เรื่อง “รถอินเวอร์ทเพนดูลัม ” ปีการศึกษา 2541 : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2. ชวลิต ศรีณคุปต์ , สงกรานต์ ศรีปัญญา , ปริญยานิพนธ์เรื่อง “อินเวอร์ทเพนดูลัม ” ปีการศึกษา 2542 : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
3. ชัชววรรณ สมัยสงค์ , ศิริพร ประสิทธิ์มีสมใจ ปริญยานิพนธ์เรื่อง “ การศึกษาระบบไม่มีเสถียรภาพ ” ปีการศึกษา 2547 : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4. วิบูลย์ แสงวีระพันธุ์ศิริ “ การควบคุมระบบพลศาสตร์ ” , สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย , 2548
5. Beckwith, B., Desjardins, E., Howard, C., Murphy, J., Ugancz, M. & Woolley, J. (2004), HTV Project Final Report, Camosun College, Canada.
6. Blackwell, T. (2005), How To Build A Self Balancing Scooter, 2005.
URL: <http://tlb.org/scooter.html>
7. Chudleigh, M., Clarke, D. & Lemire-Elmore, J. (2005), Project Emanual: The Almost Self Balancing Two Wheeled Electric Skateboard, 2005.
URL: <http://www.ebikes.ca/Projects/Emanual/Index.html>
8. Grasser, F., D'Arrigo, A., Colombi, S. & Rufer, A. (2002), JOE: A Mobile, Inverted Pendulum, Vol. 49, IEEE Transactions on Industrial Electronics.
9. HowStuffWorks Inc (2005), How Segways Work Website, 2005.
URL: <http://travel.howstuffworks.com/ginger.htm>
10. Larson, T. (2005), Balancing Robot Project “ Bender ”, 2005.
URL: <http://www.tedlarson.com/robots/balancingbot.htm>
11. Segway Inc (2005), Segway Website, Last accessed October 2005.
URL: <http://www.segway.com>



ภาคผนวก ก

แผนผังการทำงานของโปรแกรมโดยรวม



โปรแกรมตั้งค่าเริ่มต้นสำหรับการควบคุมมอเตอร์

โปรแกรมตั้งค่าเริ่มต้นสำหรับการควบคุมมอเตอร์นั้น เขียนด้วยภาษาซี โดยใช้คอมไพเลอร์คือ MPLAB ICD2 รายละเอียดของโปรแกรมสามารถแสดงได้ดังนี้

```
// Program           : Library for Motor control
// Description       : Include for motor control system
//                   : operate clock 7.3738 MHz at PLL 16x
// Filename          : mc.h
// C compiler        : C30 Compiler by Microchip Technology
```

```
-----
#include<pwm.h>           // Module function for PWM
#include<qei.h>           // Module function for PWM
#include<adc10.h>        // Module function for 10 bit ADC
#define PERIOD 4608      // Define Period 10 ms(100 Hz) FSZE
```

```
-----
//----- Interrupt service routine for PWM -----
//-----
```

```
void _ISR_PWMInterrupt(void)
{
    IFS2bits.PWMIF = 0; // Clear PWM interrupt flag
}
```

```
-----
//----- Interrupt service routine for QEI -----
//-----
```

```
void _ISR_QEIInterrupt(void)
{
    POSCNT = 0; // Reset position counter
    IFS2bits.QEIIIF = 0; // Clear QEI interrupt flag
}
```

```
----- Function for initial module PWM -----
```

```

void pwm_init(void)
{
    unsigned int config;    // Holds the PWM interrupt configuration value
    unsigned int sptime;   // Holds the value to be loaded into special event compare
    unsigned int config1;  // Holds PWM configuration value
    unsigned int config2;  // Holds the value be loaded into PWMCON1 register
    unsigned int config3;  // Holds the value to configure the special event trigger
                          // postscale and duty cycle

    sptime = 0x0;

    config1 = (    PWM_EN &           // Enable PWM module
                 PWM_OP_SCALE16 & // Output post scaler select 1:1
                 PWM_IPCLK_SCALE16 & // Input prescaler select 1:1
                 PWM_MOD_FREE);     // PWM free running mode

    config2 = (    PWM_MOD1_IND & // 3th channel in independent mode
                 PWM_MOD2_IND & // 2th channel in independent mode
                 PWM_MOD3_IND & // 1th channel in independant mode
                 PWM_PEN3H &     // H of channel 3 works as PWM
                 PWM_PDIS3L &    // L of channel 3 works as IO
                 PWM_PDIS2H &    // H of channel 2 works as IO
                 PWM_PDIS2L &    // L of channel 2 works as IO
                 PWM_PDIS1H &    // H of channel 1 works as IO
                 PWM_PDIS1L );    // L of channel 1 works as IO

    config3 = (    PWM_SEVOPS1 & // Special event post scaler 1:1
                 PWM_OSYNC_PWM & // over ride synchronised with PWM
                                     clk
                 PWM_UEN);        // Update of PDCs 1and PTPER
                                     enabled
}

```

```

    OpenMCPWM(PERIOD,sptime,config1,config2,config3); // Setup parameter for PWM
module

}

//----- Function drive DC motor Forward -----
//-----
void forward(char pow)
{
    unsigned int dutycycle,updatedisable;
    dutycycle = pow*(PERIOD/50); // Keep duty cycle
    updatedisable = 0;           // Disable update
    LATBbits.LATB0 = 1; // Bit1 control direction drive motor forward for module 1

    SetDCMCPWM(3,dutycycle,updatedisable);
}

//----- Function drive DC motor Backward -----
//-----
void backward(char pow)
{
    unsigned int dutycycle,updatedisable;
    dutycycle = pow*(PERIOD/50); // Keep duty cycle
    updatedisable = 0;           // Disable update
    LATBbits.LATB0 = 0; // Bit1 control direction drive motor backward for module 1
    SetDCMCPWM(3,dutycycle,updatedisable);
}

//-----

```

```
//-----Function stop motor -----  
//-----  
  
void stop_motor(void)  
{  
    LATEbits.LATE5 = 0; // Bit1 control stop motor for module 1  
  
}  
  
void mc_init(void)  
{  
    TRISBbits.TRISB0 = 0; // Configuration RB0 as output port  
    pwm_init(); // Initial module PWM  
}
```



โปรแกรมควบคุม โมดูลแปลงอนาลอกเป็นดิจิทัล

โปรแกรมควบคุม โมดูลแปลงอนาลอกเป็นดิจิทัล นั้น เขียนด้วยภาษาซี โดยใช้คอมไพเลอร์คือ MPLAB ICD2 รายละเอียดของโปรแกรมสามารถแสดงได้ดังนี้

```

/*
* ADDITIONAL NOTES:
* This file contains two functions - ADC_Init() and _ADCInterrupt().
*
// C compiler          : C30 Compiler by Microchip Technology
*****
#include <p30f2010.h>
#include "ADCme.h"
unsigned int ADResult1;
unsigned int ADResult2;
unsigned int ADResult3;
unsigned int ADResult4;
unsigned int ADResult5;
unsigned int ADResult6;

//Functions and Variables with Global Scope:

//Functions:
//ADC_Init() is used to configure A/D to convert 16 samples of 1 input
//channel per interrupt. The A/D is set up for a sampling rate of 1MSPS
//Timer3 is used to provide sampling time delay.
//The input pin being acquired and converted is AN3.
void ADC_Init(void)
{
    //ADCON1 Register
    //Set up A/D for Automatic Sampling
    //Use internal counter (SAMC) to provide sampling time
    //Set up A/D conversion results to be read in 1.15 fractional

```

```

//number format.

//Set up Sequential sampling for multiple S/H amplifiers

//All other bits to their default state
ADCON1bits.FORM = 0;
ADCON1bits.SSRC = 7;
ADCON1bits.ASAM = 1;
ADCON1bits.SIMSAM = 0;

//ADCON2 Register
//Set up A/D for interrupting after 2 samples get filled in the buffer
//Set up to sample on 2 S/H amplifiers - CH0 and CH1
//All other bits to their default state
ADCON2bits.SMPI = 9;
ADCON2bits.CHPS = 1;
    //ADCON2bits.VCFG = 3; //Ideally use external references

//ADCON3 Register
//We would like to set up a sampling rate of 1 MSPS
//Total Conversion Time= 1/Sampling Rate = 125 microseconds
//At 29.4 MIPS, Tcy = 33.9 ns = Instruction Cycle Time
//The A/D converter will take 12*Tad periods to convert each sample
//So for ~1 MSPS we need to have Tad close to 83.3ns
//Using equaion in the Family Reference Manual we have
//ADCS = 2*Tad/Tcy - 1
ADCON3bits.SAMC = 0;
ADCON3bits.ADCS = 15;

//ADCHS Register
//Set up A/D Channel Select Register to convert AN3 on Mux A input
//of CH0 and CH1 S/H amplifiers
ADCHS = 0x0023;

```

```

//ADCSSL Register
//Channel Scanning is disabled. All bits left to their default state
ADCSSL = 0x0000;

//ADPCFG Register
//Set up channels AN7 as analog input and configure rest as digital
//Recall that we configured all A/D pins as digital when code execution
//entered main() out of reset
ADPCFG = 0xFFFF;
ADPCFGbits.PCFG3 = 0;

//Clear the A/D interrupt flag bit
IFS0bits.ADIF = 0;

//Set the A/D interrupt enable bit
IEC0bits.ADIE = 0;

//Turn on the A/D converter
//This is typically done after configuring other registers
ADCON1bits.ADON = 1;
}

//_ADCInterrupt() is the A/D interrupt service routine (ISR).
//The routine must have global scope in order to be an ISR.
//The ISR name is chosen from the device linker script
void __attribute__((__interrupt__)) _ADCInterrupt(void)
{
    ADResult1 = ADCBUF0;
    ADResult2 = ADCBUF1;
    ADResult3 = ADCBUF2;
    ADResult4 = ADCBUF3;
}

```

```
ADResult5 = ADCBUF4;  
ADResult6 = ADCBUF5;  
IFS0bits.ADIF = 0; //Clear the A/D Interrupt flag bit or else the CPU will  
//keep vectoring back to the ISR  
}
```



โปรแกรมควบคุมกำลังและทิศทางการหมุนของมอเตอร์

โปรแกรมควบคุมกำลังและทิศทางการหมุนของมอเตอร์นั้น เขียนด้วยภาษาซี โดยใช้คอมไพเลอร์คือ MPLAB ICD2 รายละเอียดของโปรแกรมสามารถแสดงได้ดังนี้

```
// Program           : Motor control
// Description       : Control position motor by PWM module
//
//
//                   operate clock 7.3738 MHz at PLL 16x
// Filename          : motor_control.c
// C compiler        : C30 Compiler by Microchip Technology
//-----
#include <p30f2010.h> // Header file for dsPIC30F2010
#include <ADCme.h>
#include <mc.h>      // Module function motor control
#define con 4
#define con2 2
#define con3 590
#define t 1
unsigned int ADResult ,ADshow;
//----- Function For drive motor-----
unsigned int adc_val,adc_val2,i,sum,max,min, old ,new,diff[10],n,e[4];

unsigned char power;           // value for controlmotor

_FOSC(CSW_FSCM_OFF & XT_PLL16); //Run this project using an external crystal
//routed via the PLL in 16x multiplier mode
//For the 7.3728 MHz crystal we will derive a
//throughput of  $7.3728e+6 * 16/4 = 29.4 \text{MIPS} (F_{cy})$ 
//,~33.9 nanoseconds instruction cycle time( $T_{cy}$ ).
_FWDT(WDT_OFF);             //Turn off the Watch-Dog Timer.
```

```

_FBORPOR(MCLR_EN & PWRT_OFF); //Enable MCLR reset pin and turn off the
//power-up timers.
_FGS(CODE_PROT_OFF);

```

```

void PD(void)//PD Function

```

```

{
    diff[1]= abs(adc_val2 - ADshow);    // P
    diff[2]= (diff[1])/t;                // D
    power = con*diff[1] + con2 * diff[2] ;
    e[2] = ADshow;                       //old ADC Value
    if(power > 100)
    {
        power = 100 ;
    }
}

```

```

void read_adc2 (unsigned int t0) //read ADC value

```

```

{
    int i,j;
    unsigned int d_adc,old_adshow;
    ADResult1 = ADCBUF0;
    ADResult2 = ADCBUF1;
    ADResult3 = ADCBUF2;
    ADResult4 = ADCBUF3;
    ADResult5 = ADCBUF4;
    ADResult6 = ADCBUF5;
    ADResult=(ADResult1+ADResult2+ADResult3+ADResult4+ADResult5)/5 ;
    ADshow = ADResult;
    delay_ms(t0);
}

```

```

void mem_balanc(void) // memory setpoint

```

```

{
    LATCbits.LATC14    = 1;
    delay_ms(200);
    read_adc2(300);
    adc_val2 = ADshow ;
    delay_ms(200);
    LATCbits.LATC14    = 0;
}

//----- Function drive motor Forward -----//

void pos_fd(unsigned int step)//control motor forward
{
    forward(step); // Call function process drive motor
}

//----- Function drive motor Backward -----//

void pos_bk(unsigned int step)//control motor backward
{
    backward(step); //function process drive motor
}

//----- Main Program -----//

int main(void)
{
    unsigned int count=1,step; // Keep position count
    mc_init(); // initial for module PWM
    ADC_Init(); //Initialize the A/D converter
    adc_val2 = con3 ; //memory setpoint

while(1)

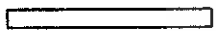
```

```
{  
    read_adc2(10); //read ADC Value for 10 ms  
    PID();  
    if (ADshow < adc_val2 ) // if ADC Value more than setpoint  
    {  
        pos_fd(power);  
    }  
    if (ADshow > adc_val2 ) // if ADC Value less than setpoint  
    {  
        pos_bk(power);  
    }  
    return 0;  
}
```



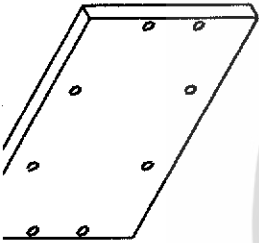


ภาคผนวก ข



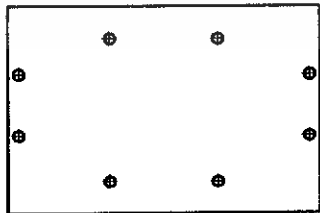
Front view

Scale: 1:2



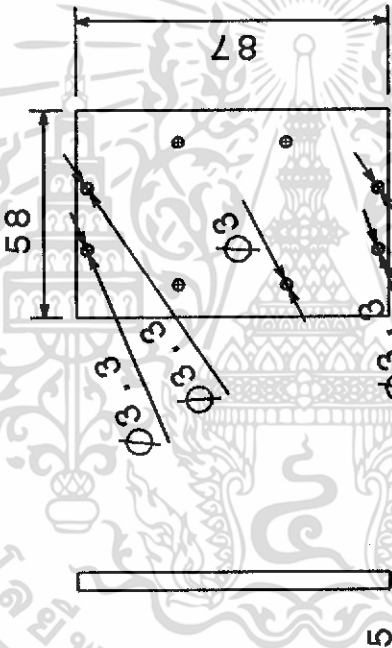
Isometric view

Scale: 1:2



Top view

Scale: 1:2



Left view

Scale: 1:2



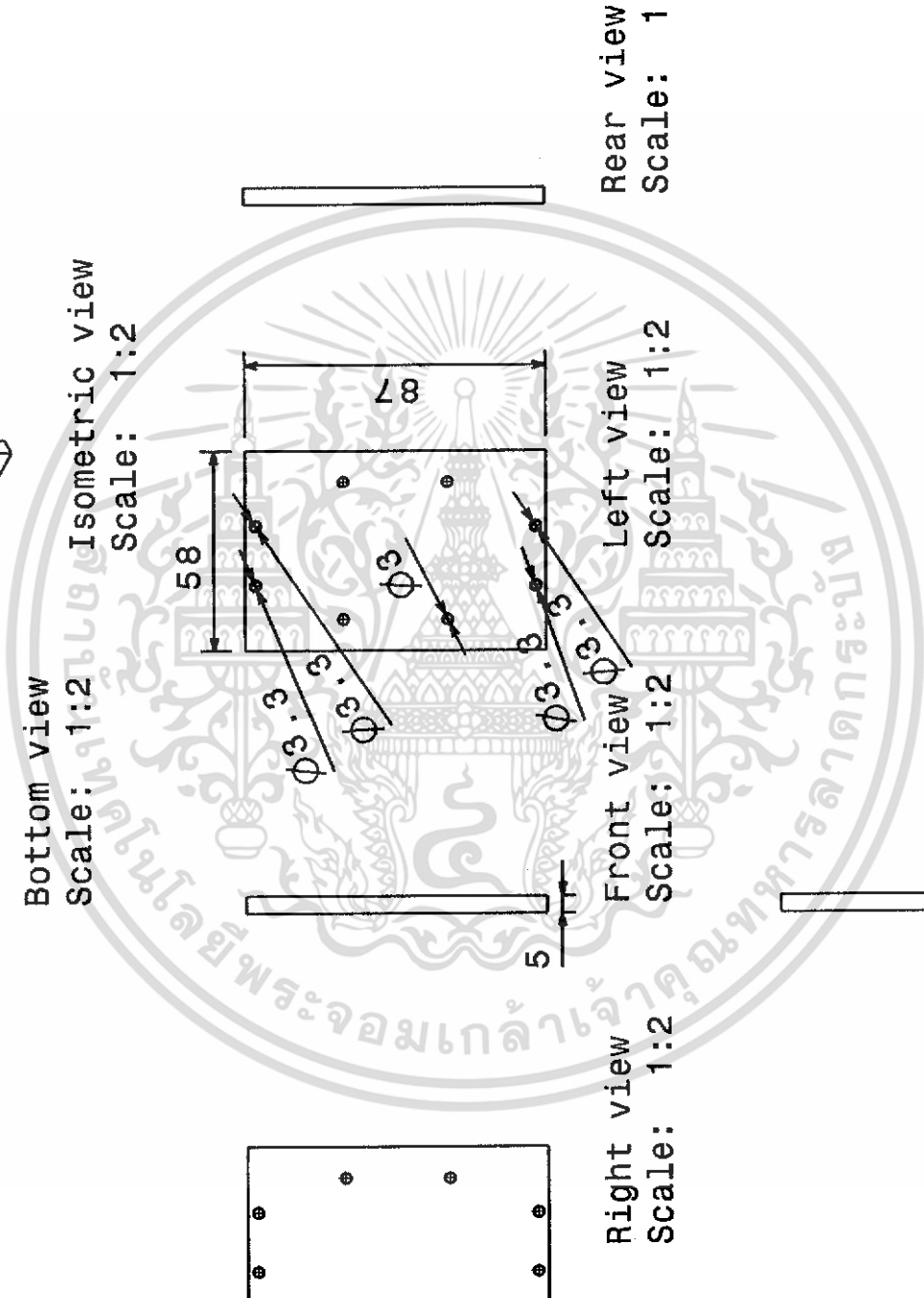
Right view

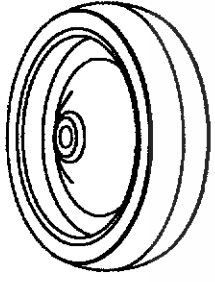
Scale: 1:2



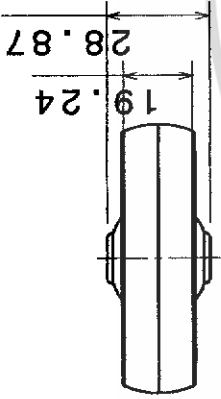
Rear view

Scale: 1:2

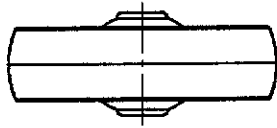




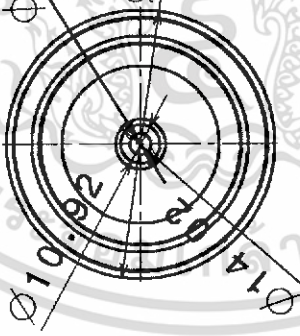
Isometric view
Scale: 1:2



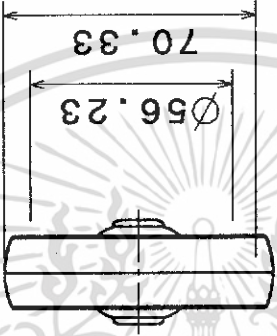
Bottom view
Scale: 1:2



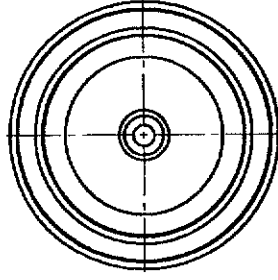
Right view
Scale: 1:2



Front view
Scale: 1:2



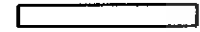
Left view
Scale: 1:2



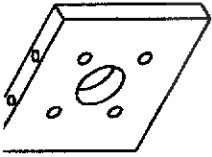
Rear view
Scale: 1:2



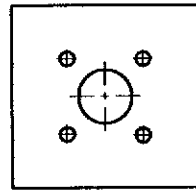
Top view
Scale: 1:2



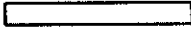
Bottom view
Scale: 1:2



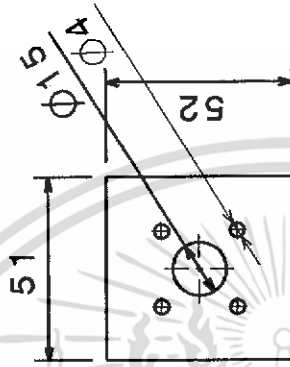
Isometric view
Scale: 1:2



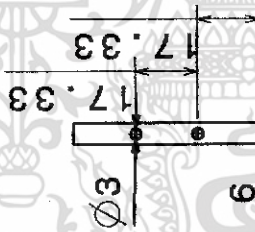
Right view
Scale: 1:2



Rear view
Scale: 1:2



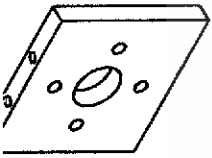
Left view
Scale: 1:2



Front view
Scale: 1:2

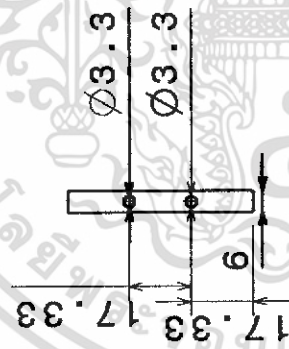


Top view
Scale: 1:2

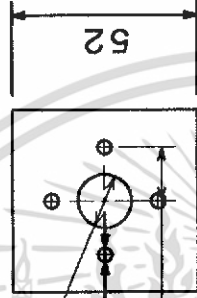


Bottom view
Scale: 1:2

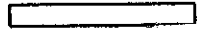
Isometric view
Scale: 1:2



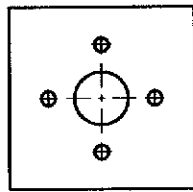
Front view
Scale: 1:2



Left view
Scale: 1:2



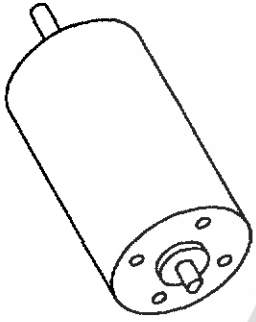
Rear view
Scale: 1:2



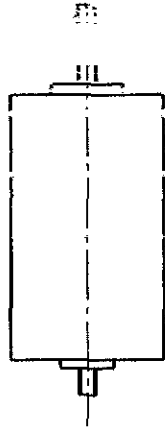
Right view
Scale: 1:2



Top view
Scale: 1:2

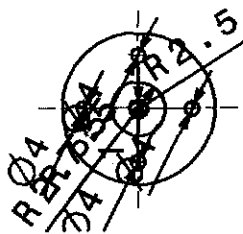
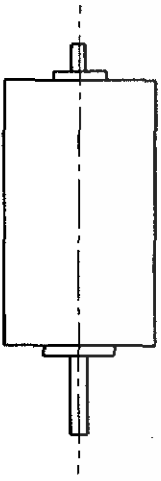


Isometric view
Scale: 1:2

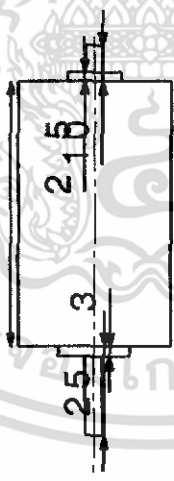


Rear view
Scale: 1:2

Bottom view
Scale: 1:2



Right view
Scale: 1:2



Front view
Scale: 1:2



Left view
Scale: 1:2



Top view
Scale: 1:2



Isometric view
Scale: 1:2



Bottom view
Scale: 1:2

$\varnothing 3$



Front view
Scale: 1:2



R3

Right view
Scale: 1:2

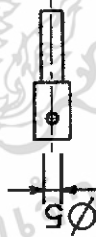


R5

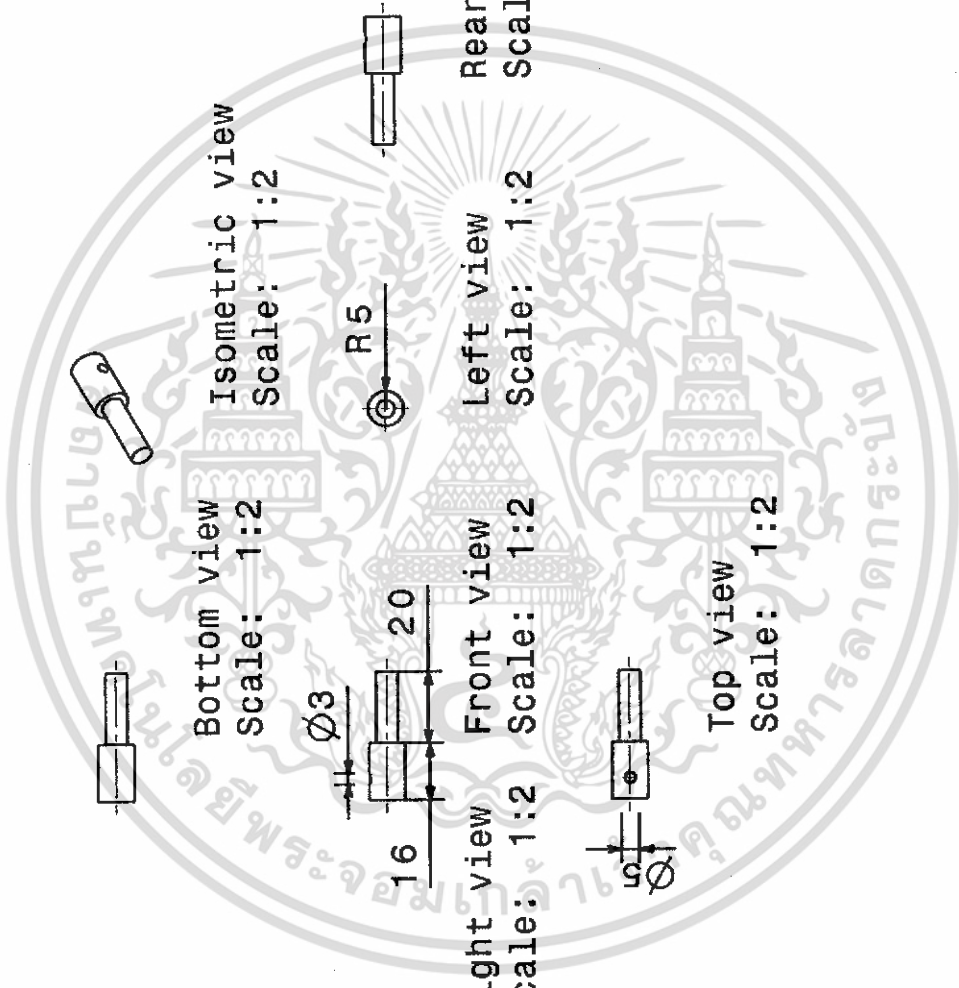
Left view
Scale: 1:2



Rear view
Scale: 1:2



Top view
Scale: 1:2





ภาคผนวก ค

GP2D12/GP2D15

General Purpose Type Distance Measuring Sensors

■ Features

1. Less influence on the color of reflective objects, reflectivity
2. Line-up of distance output/distance judgement type
 Distance output type (analog voltage) : **GP2D12**
 Detecting distance : 10 to 80cm
 Distance judgement type : **GP2D15**
 Judgement distance : 24cm
 (Adjustable within the range of 10 to 80cm)
3. External control circuit is unnecessary
4. Low cost

■ Applications

1. TVs
2. Personal computers
3. Cars
4. Copiers

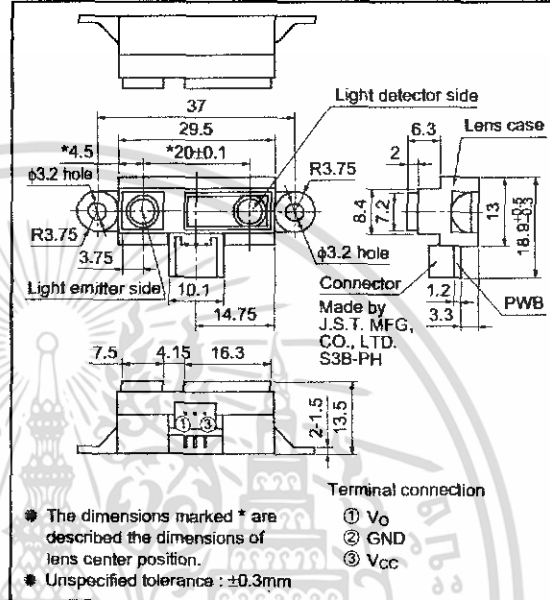
■ Absolute Maximum Ratings

(Ta=25°C, Vcc=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	Vcc	-0.3 to +7	V
Output terminal voltage	Vo	-0.3 to Vcc+0.3	V
Operating temperature	Topt	-10 to +60	°C
Storage temperature	Tsig	-40 to +70	°C

■ Outline Dimensions

(Unit : mm)



Notice In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.
 Internet Internet address for Electronic Components Group <http://www.sharp.co.jp/ecg/>

■ Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Operating supply voltage	V _{CC}	4.5 to +5.5	V

■ Electro-optical Characteristics

(T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Distance measuring range	ΔL	*1 *3	10	-	80	cm
Output terminal voltage	GP2D12	V _O L=80cm *1	0.25	0.4	0.55	V
	GP2D15	V _{OH} Output voltage at High *1	V _{CC} - 0.3	-	-	V
	GP2D15	V _{OL} Output voltage at Low *1	-	-	0.6	V
Difference of output voltage	GP2D12	ΔV _O Output change at L=80cm to 10cm *1	1.75	2.0	2.25	V
Distance characteristics of output	GP2D15	V _O *1 *2 *4	21	24	27	cm
Average Dissipation current	I _{CC}	L=80cm *1	-	33	50	mA

Note) L : Distance to reflective object.

*1 Using reflective object : White paper (Made by Kodak Co. Ltd. gray cards R-27 : white face, reflective ratio ; 90%).

*2 We ship the device after the following adjustment : Output switching distance L=24cm±3cm must be measured by the sensor.

*3 Distance measuring range of the optical sensor system.

*4 Output switching has a hysteresis width. The distance specified by V_O should be the one with which the output L switches to the output H.

Fig.1 Internal Block Diagram

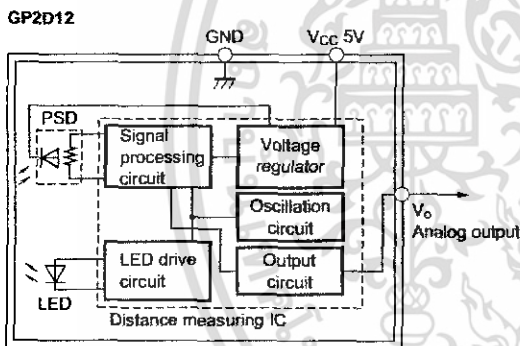


Fig.2 Internal Block Diagram

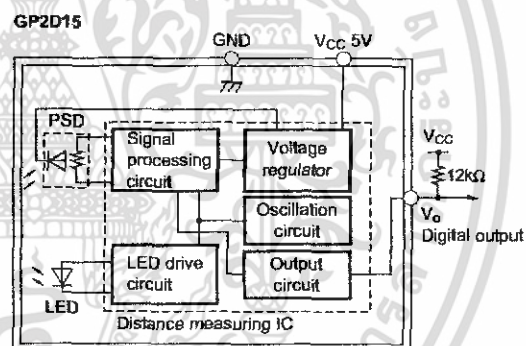


Fig.3 Timing Chart

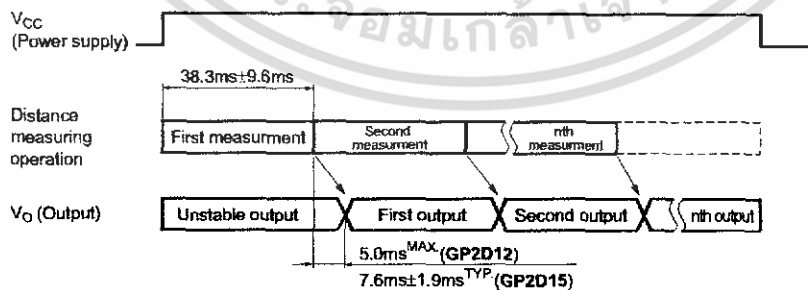


Fig.4 Distance Characteristics

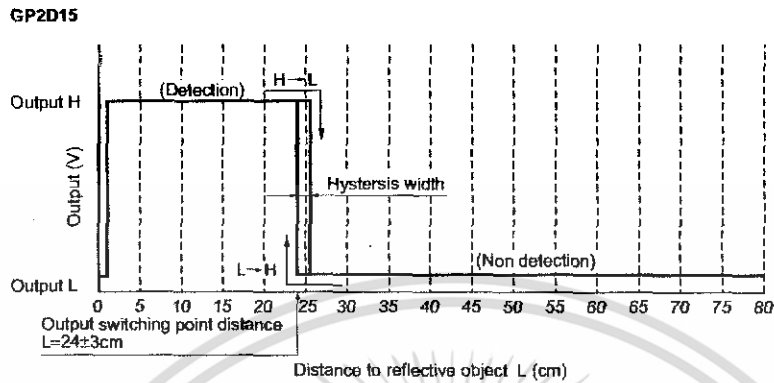


Fig.5 Analog Output Voltage vs. Surface Illuminance of Reflective Object

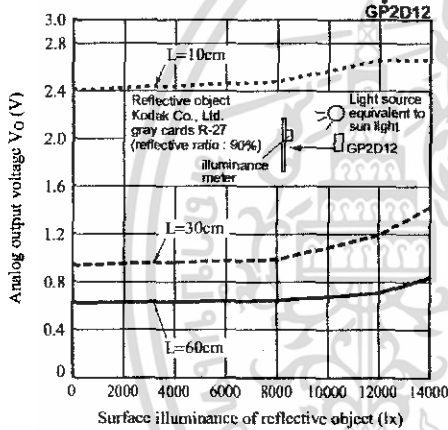


Fig.6 Analog Output Voltage vs. Distance to Reflective Object

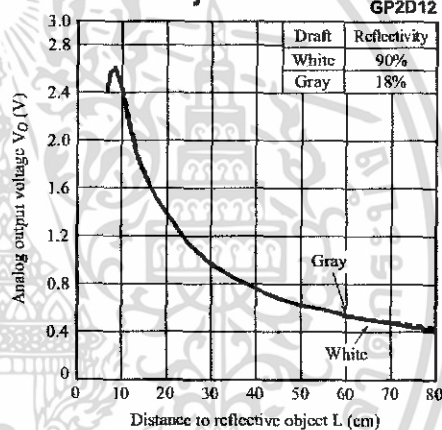


Fig.7 Analog Output Voltage vs. Ambient Temperature

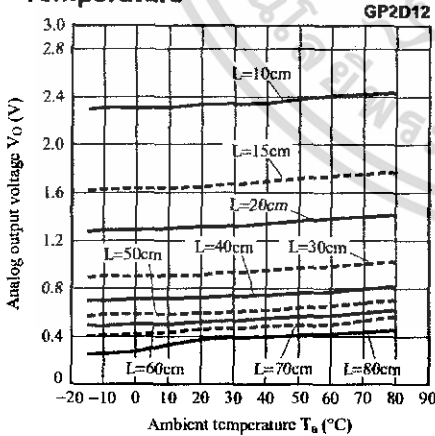
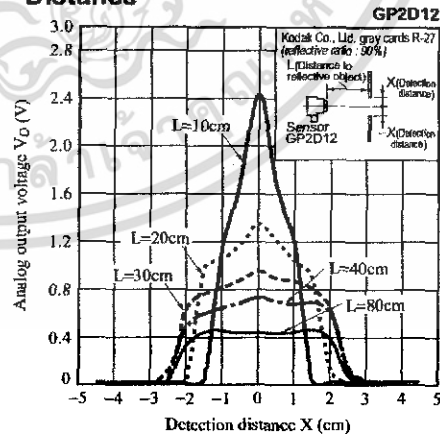


Fig.8 Analog Output Voltage vs. Detection Distance



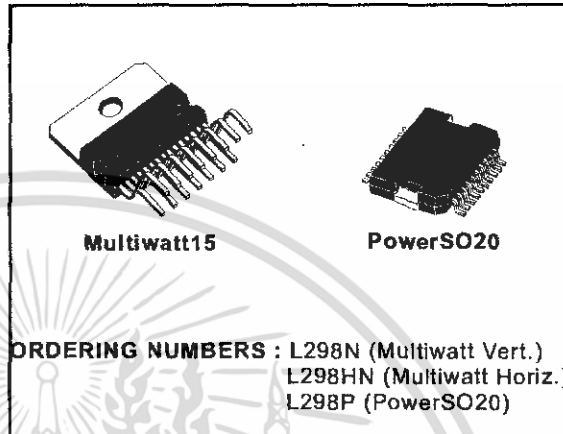


DUAL FULL-BRIDGE DRIVER

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

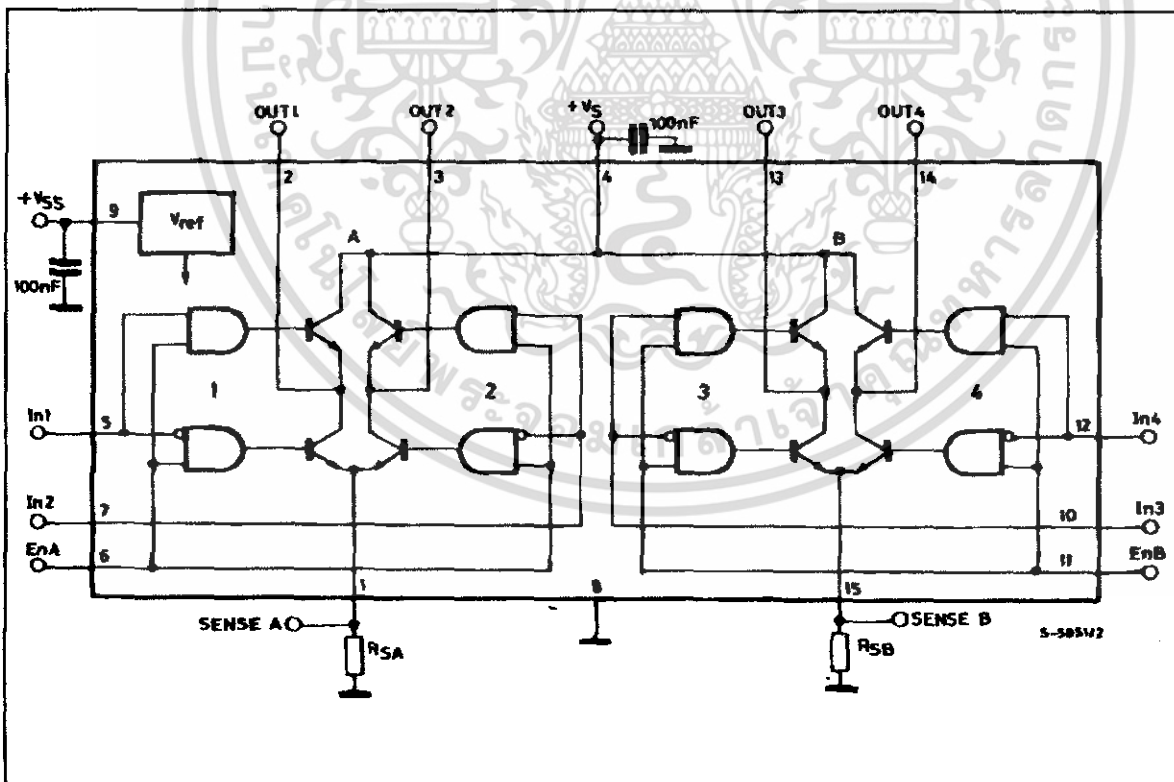
DESCRIPTION

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



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

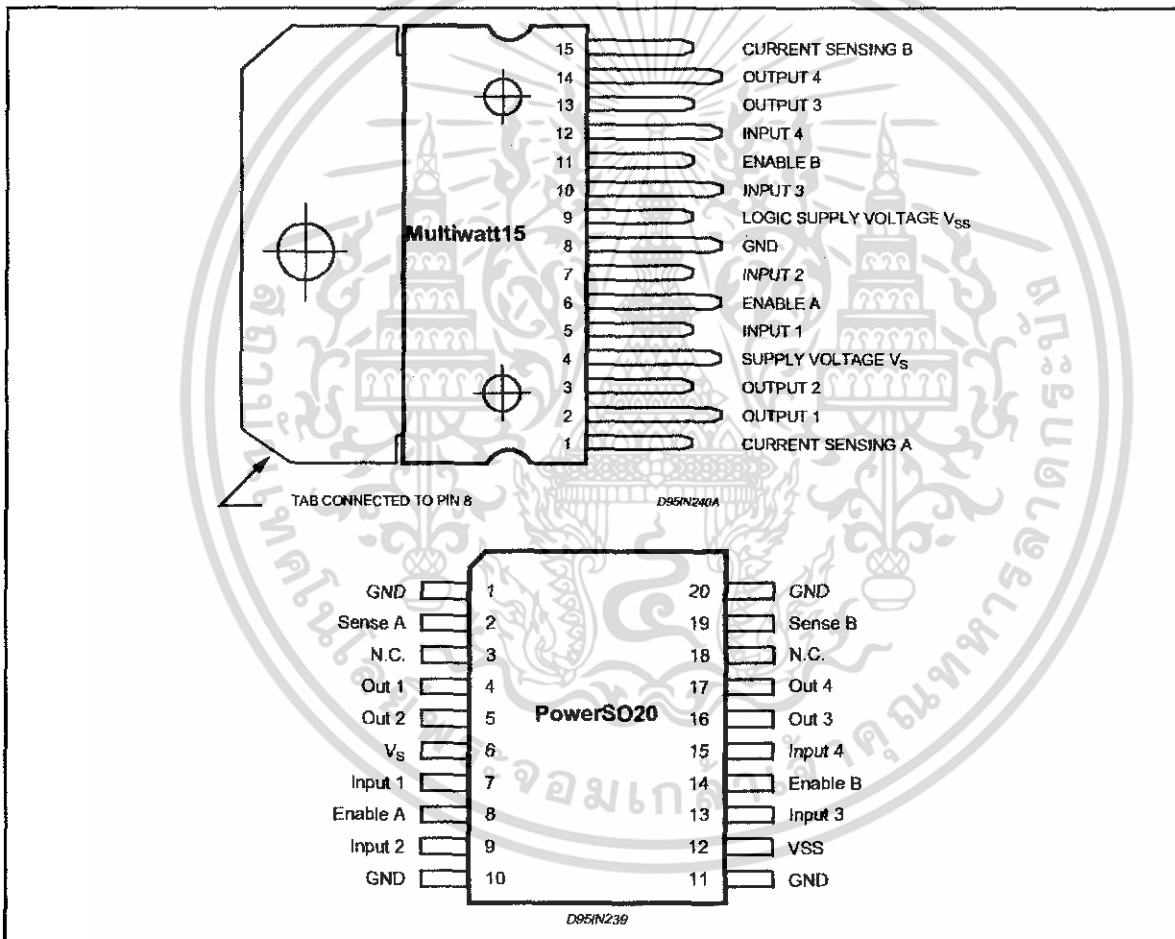
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _S	Power Supply	50	V
V _{SS}	Logic Supply Voltage	7	V
V _I , V _{En}	Input and Enable Voltage	-0.3 to 7	V
I _O	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100µs)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	-DC Operation	2	A
V _{sens}	Sensing Voltage	-1 to 2.3	V
P _{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T _{op}	Junction Operating Temperature	-25 to 130	°C
T _{stg} , T _J	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
R _{th(jc)case}	Thermal Resistance Junction-case	Max.	3	°C/W
R _{th(ja)amb}	Thermal Resistance Junction-ambient	Max.	13 (*)	°C/W

(*) Mounted on aluminum substrate



PIN FUNCTIONS (refer to the block diagram)

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

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_J = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _i = L V _i = H		13 50	22 70	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = L V _i = L V _i = H V _i = X		24 7	36 12	mA
V _L	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _H	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			-10	μA
I _{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} - 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} - 0.6V		30	100	μA
V _{CEsat} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V
V _{CEsat} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V
V _{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V



dsPIC30F2010

28-pin dsPIC30F2010 Enhanced Flash 16-bit Digital Signal Controller

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *dsPIC30F Family Reference Manual* (DS70046). For more information on the device instruction set and programming, refer to the *dsPIC30F Programmer's Reference Manual* (DS70030).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture
- 84 base instructions with flexible addressing modes
- 24-bit wide instructions, 16-bit wide data path
- 12 Kbytes on-chip Flash program space
- 512 bytes on-chip data RAM
- 1 Kbyte non-volatile data EEPROM
- 16 x 16-bit working register array
- Up to 30 MIPS operation:
 - DC to 40 MHz external clock input
 - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 27 interrupt sources
- Three external interrupt sources
- 8 user selectable priority levels for each interrupt
- 4 processor exceptions and software traps

DSP Engine Features:

- Modulo and Bit-Reversed modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single cycle hardware fractional/integer multiplier
- Single cycle Multiply-Accumulate (MAC) operation
- 40-stage Barrel Shifter
- Dual data fetch

Peripheral Features:

- High current sink/source I/O pins: 25 mA/25 mA
- Three 16-bit timers/counters; optionally pair up 16-bit timers into 32-bit timer modules
- Four 16-bit Capture input functions
- Two 16-bit Compare/PWM output functions
 - Dual Compare mode available
- 3-wire SPI™ modules (supports 4 Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- Addressable UART modules with FIFO buffers

Motor Control PWM Module Features:

- 6 PWM output channels
 - Complementary or Independent Output modes
 - Edge and Center Aligned modes
- 4 duty cycle generators
- Dedicated time base with 4 modes
- Programmable output polarity
- Dead time control for Complementary mode
- Manual output control
- Trigger for synchronized A/D conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

Analog Features:

- 10-bit Analog-to-Digital Converter (A/D) with:
 - 500 Ksps (for 10-bit A/D) conversion rate
 - Six input channels
 - Conversion available during Sleep and Idle
- Programmable Brown-out Detection and Reset generation

dsPIC30F2010

Special Microcontroller Features:

- Enhanced Flash program memory:
 - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
 - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with on-chip low power RC oscillator for reliable operation
- Fail-Safe clock monitor operation

- Detects clock failure and switches to on-chip low power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes
 - Sleep, Idle and Alternate Clock modes

CMOS Technology:

- Low power, high speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption

dsPIC30F Motor Control and Power Conversion Family*

Device	Pins	Program Mem. Bytes/Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	A/D 10-bit 500 Ksps	Quad Enc	UART	SPI™	I ² C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	-
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

* This table provides a summary of the dsPIC30F2010 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

Pin Diagrams

28-Pin SDIP and SOIC

MCLR	1	28	AVDD
EMUD3/AN0/REF+/CN2/RB0	2	27	AVSS
EMUC3/AN1/REF-/CN3/RB1	3	26	PWM1L/RE0
AN2/SS1/CN4/RB2	4	25	PWM1H/RE1
AN3/INDX/CN5/RB3	5	24	PWM2L/RE2
AN4/QEA/IC7/CN6/RB4	6	23	PWM2H/RE3
AN5/QEB/IC8/CN7/RB5	7	22	PWM3L/RE4
VSS	8	21	PWM3H/RE5
OSC1/CLKI	9	20	VDD
OSC2/CLKO/RC15	10	19	VSS
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13	11	18	PGC/EMUC/U1RX/SDI1/SDA/RF2
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	12	17	PGD/EMUD/U1TX/SDO1/SCL/RF3
VDD	13	16	FLTA/INT0/SCK1/OCFA/RE8
EMUD2/OC2/IC2/INT2/RD1	14	15	EMUC2/OC1/IC1/INT1/RD0

28-Pin QFN

EMUC3/AN1/REF-/CN3/RB1	28	21	PWM2L/RE2
EMUD3/AN0/REF+/CN2/RB0	27	20	PWM2H/RE3
MCLR	26	19	PWM3L/RE4
AVDD	25	18	PWM3H/RE5
AVSS	24	17	VDD
PWM1L/RE0	23	16	VSS
PWM1H/RE1	22	15	PGC/EMUC/U1RX/SDI1/SDA/RF2
AN2/SS1/CN4/RB2	1		
AN3/INDX/CN5/RB3	2		
AN4/QEA/IC7/CN6/RB4	3		
AN5/QEB/IC8/CN7/RB5	4		
VSS	5		
OSC1/CLKIN	6		
OSC2/CLKO/RC15	7		
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13	8		
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	9		
VDD	10		
EMUD2/OC2/IC2/INT2/RD1	11		
EMUC2/OC1/IC1/INT1/RD0	12		
FLTA/INT0/SCK1/OCFA/RE8	13		
PGD/EMUD/U1TX/SDO1/SCL/RF3	14		