

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุลง โดยใช้หลักการ อินเวอร์ทเพนดูลัม

TWO WHEELS BALANCING ROBOT BY INVERTED
PENDULUM



เลขหมู่.....
เลขทะเบียน..... 62332
วัน,เดือน,ปี..... 16 ส.ค. 2549

b. 11621230
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TWO WHEELS BALANCING ROBOT BY INVERTED
PENDULUM**



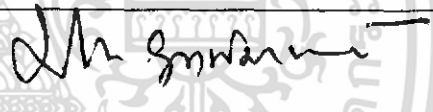
**A THESIS SUBMITTED IN PARTIAL FULLFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

หัวข้อปริญญาโท หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุล โดยใช้หลักการ อินเวอร์ทเพนดูลัม
TWO WHEELS BALANCING ROBOT BY INVERTED PENDULUM
นักศึกษาผู้จัดทำ นายเจษฎา ชนะทัฬหะ รหัสประจำตัว 45010142
นายทัตพล อาจสำอางค์ รหัสประจำตัว 45010302
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2548

อาจารย์ผู้ควบคุมปริญญาโท	ลายมือชื่อ
รศ. ประภาพร อุดคภูมิพันธุ์ ผศ. ดร. ธนิตย์ ตรีสุวรรณวัฒน์	

ภาควิชารับรองแล้ว



(รศ.ประสิทธิ์ จุลเสวีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุลง โดยใช้หลักการ อินเวอร์ทเพนดูลัม
TWO WHEELS BALANCING ROBOT BY INVERTED
PENDULUM
นักศึกษาผู้จัดทำ นายเจษฎา ชนะทัฬห รหัสประจำตัว 45010142
 นายทัตพล อาจสำอางค์ รหัสประจำตัว 45010302
อาจารย์ที่ปรึกษา รศ. ประภาส อุดคกิมานันท์
 ผศ. ดร. ธนิตย์ ตริสุวรรณวัฒน์
ปีการศึกษา 2548

บทคัดย่อ

เทคโนโลยีที่เกี่ยวกับการทรงตัวของวัตถุนั้น ได้มีการคิดค้นและพัฒนาให้ดีขึ้นอยู่ตลอดเวลา ซึ่งหลักการและทฤษฎีต่าง ๆ นั้น ได้ถูกนำมาใช้ประโยชน์มากมาย ไม่ว่าจะเป็น สิ่งปลูกสร้างอาคารบ้านเรือน, หุ่นยนต์ในอุตสาหกรรม หรือ อาจเป็นยานพาหนะต่างๆ เป็นต้น ระบบแบบอินเวอร์ทเพนดูลัม ถือเป็นระบบหนึ่งที่เกี่ยวข้องกับการทรงตัวของวัตถุ ซึ่งได้ถูกนำมาประยุกต์ใช้ในรูปแบบต่างๆ มากมาย ล่าสุดได้มีการนำเอาระบบนี้มาใช้กับยานพาหนะที่เรียกว่า SEGWAY ซึ่งหลักการของระบบอินเวอร์ทเพนดูลัมก็คือ หลักการที่เกี่ยวกับการทรงตัว โดยการรักษาสสมดุลของก้านเพนดูลัม ให้ตั้งฉากกับแกนหรือฐานของเพนดูลัม ซึ่งหลักการนี้ได้ถูกนำมาประยุกต์ใช้ร่วมกับหุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุลง เพื่อรักษาสสมดุลของตัวรถให้ทรงตัวอยู่ได้ตลอด โดยการทรงตัวของก้านนี้จะต้องสัมพันธ์กับส่วนของการทำงานระบบขับเคลื่อนของตัวรถด้วย ซึ่งการควบคุมการทำงานของมอเตอร์นั้นจะใช้ระบบควบคุมแบบออปติมอลคอนโทรล ผ่านตัวประมวลผลไมโครคอนโทรลเลอร์แบบdsPIC ซึ่งเป็นตัวไมโครคอนโทรลเลอร์ความเร็วสูง ความสัมพันธ์ทั้งหมดนี้จะต้องถูกออกแบบอย่างดีที่สุด เพื่อให้หุ่นยนต์สามารถที่จะทรงตัวอยู่ได้แม้จะถูกรบกวนจากแรงภายนอก อีกทั้งยังสามารถควบคุมให้เคลื่อนที่ไป ยังทิศทางที่ต้องการได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Two Wheels Balancing Robot by Inverted Pendulum
Authors Mr. Jedsada Chanatup
Mr. Tutpol Ardsomang
Thesis Advisor Assoc.Prof. Prapat Ukkakimapan
Asst.Prof. Tanit Trisuwanwat
Year 2005

ABSTRACT

Technology of balancing rigid body is developed every time. It is used for many things such as many kinds of Buildings, Robots and Vehicles etc. Inverted Pendulum system is one of theories of balancing robot which is applied and used for many things. And last this system is used for vehicle that is called "SEGWAY". The principle of Inverted Pendulum is about balancing of the rigid body. By keeping the pendulum to be perpendicularly with base. All principle can be applied in The Two Wheel Balancing Robot system for keeping the car in equilibrium. The optimal control system is applied here to design the driving system which operate relatively to the balancing of pendulum and dsPIC, a hi-speed microcontroller, is used to obtain the best performance. All the relations must be designed properly in order to keep the car in equilibrium eventhough the car is disturbed by unexpected force and to control the car to move correctly to the exact destination.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความเมตตาจาก รองศาสตราจารย์ ประภาส อุคคกิมพันธ์ และ ผู้ช่วยศาสตราจารย์ ดร.ธนิตย์ ศรีสุวรรณวัฒน์ ที่ได้ให้คำแนะนำแก่ ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆ ในการทำปริญญาบัตรนี้ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณอาจารย์ไสว พงศ์สวัสดิ์ ที่เอื้อเฟื้ออุปกรณ์ คุณณกฤษณ์ เสมอพิทักษ์ และ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ได้คำแนะนำอันเป็นประโยชน์และให้ความช่วยเหลือตลอดมา

ขอกราบขอบพระคุณคุณพ่อ คุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญาบัตรฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญาบัตรฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	VI
บทที่ 1 บทนำ.....	I
1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	1
1.4 ขั้นตอนการศึกษา.....	1
บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับหลักการควบคุมการทรงตัวของหุ่นยนต์.....	2
2.1 กล่าวนำ	
2.1.1 หลักการทำงานและทฤษฎีต่างๆ.....	2
2.1.2 หลักการออกแบบ.....	2
2.2 หลักการทำงานของคิซิมอเตอร์.....	3
2.3 หลักการของระบบควบคุมที่เหมาะสมที่สุด(Optimal Control System).....	4
2.3.1 หลักการออกแบบระบบควบคุม.....	4
2.3.2 ตัวประมาณค่าสแตต (States Estimator).....	9
2.3.3 ระบบคู่ (Dual System).....	16
2.4 หลักการการทำงานของโรโบติก.....	21
2.4.1 การควบ.....	21
2.4.2 ทฤษฎีและหลักการทำงาน.....	21
2.4.3 ประโยชน์ของโรโบติก.....	23
บทที่ 3 การออกแบบระบบควบคุม ไมโครคอนโทรลเลอร์ วงจรขับมอเตอร์ แบบจำลองทาง	
 คณิตศาสตร์ และ การออกแบบตัวหุ่นยนต์.....	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.1 แบบจำลองทางคณิตศาสตร์ของหุ่นยนต์.....	24
3.1.1 แบบจำลองทางคณิตศาสตร์ของดีซีมอเตอร์.....	24
3.1.2 พฤติกรรมของ Balancing Robot.....	26
3.1.3 หลักการของ Inverted Pendulum.....	28
3.2 การออกแบบระบบควบคุม.....	30
3.3 การออกแบบโปรแกรมบนไมโครคอนโทรลเลอร์.....	33
3.3.1 ข้อมูลเบื้องต้นของไมโครคอนโทรลเลอร์แบบ dsPIC6011.....	33
3.3.2 การออกแบบโปรแกรมในไมโครคอนโทรลเลอร์.....	34
3.4 วงจรควบคุมการทำงานของมอเตอร์.....	36
3.5 การออกแบบตัวหุ่นยนต์.....	38
3.5.1 ลักษณะทั่วไป.....	38
3.5.2 ลักษณะการติดตั้งตัวเซนเซอร์.....	38
3.5.3 จุดสัมผัสพื้น.....	40
3.5.4 ลักษณะการติดตั้งชิ้นส่วนต่างๆ.....	40
บทที่ 4 การทดลอง.....	42
4.1 การทดลองใส่แรงผลัก.....	42
4.2 การทดลองหาผลตอบสนองของหุ่นยนต์.....	43
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	45
บรรณานุกรม.....	46
ภาคผนวก.....	47
ภาคผนวก ก.....	48
ภาคผนวก ข.....	51
ภาคผนวก ค.....	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 แสดงระบบการทำงานของหุ่นยนต์ขับเคลื่อนแบบสองล้อ.....	2
2.2 แสดงการออกแบบตัวควบคุมในแนวตั้งเดิม.....	4
2.3 แสดงถึงสิ่งรบกวนที่เกิดขึ้นในวงควบคุม.....	4
2.4 แสดงการออกแบบตัวควบคุมในแนวใหม่.....	4
2.5 แสดงถึงความสัมพันธ์ระหว่างตัวแปรและสัญญาณ.....	5
2.6 แสดงถึงลักษณะของ Generalized Plant.....	5
2.7 แสดงถึงลักษณะของ Plant.....	5
2.8 แสดงถึงลักษณะของ Weighting Functions.....	6
2.9 แสดงถึงลักษณะของระบบ LQR.....	7
2.10 แสดงถึงลักษณะของ Generalized Plant.....	8
2.11 แสดงถึงลักษณะของ Plant.....	8
2.12 แสดงถึงลักษณะของ Weighting Functions.....	8
2.13 แสดงถึงลักษณะของระบบ LQR รวมทั้ง LQE.....	9
2.14 แสดงถึงลักษณะของระบบ LQE.....	12
2.15 แสดงถึงลักษณะของระบบ LQR.....	17
2.16 แสดงถึงลักษณะของ Pressure Regulator.....	20
2.17 ภาพแสดงการหมุนและทิศทางของแรงที่กระทำกับลูกข่าง.....	21
2.18 แสดงทิศทางการหมุนและแรง.....	22
2.19 แสดงถึงดาวเทียมชี้ไปในทิศทางเดียวกัน.....	23
3.1 แบบจำลองโครงสร้างของมอเตอร์.....	25
3.2 Free body diagram ของล้อ.....	26
3.3 Free body diagram ของโครงรถส่วนที่เป็นก้าน Pendulum.....	28
3.4 Open Loop Poles ของระบบที่ไม่อยู่ในเสถียรภาพ.....	30
3.5 Close Loop Pole ของระบบที่อยู่ในเสถียรภาพ.....	31
3.6 ผลตอบสนองของระบบเมื่อค่าน้ำหนัก $q_{11} = 1$ กับ $q_{33} = 1$ และ $R = 1$	31
3.7 ผลตอบสนองเมื่อทำการเปลี่ยนแปลงค่าน้ำหนัก $q_{11} = 1000$ กับ $q_{33} = 1$ และ $R = 1$	31
3.8 ผลตอบสนองเมื่อค่าน้ำหนักเป็น $q_{11} = 1000$ กับ $q_{33} = 1000$ และ $R = 1/100$	32
3.9 dsPIC30F6010 80-Lead Plastic Thin Quad Flatpack (TQFP).....	33
3.10 แผนภาพโดยรวมของการเขียนโปรแกรม.....	34

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่	หน้า
3.11 วงจรขับเคลื่อนโดยตรงของมอเตอร์.....	37
3.12 ลักษณะทั่วไปของหุ่นยนต์ขับเคลื่อนแบบสองล้อ.....	38
3.13 ลักษณะการติดตั้งตัวเซนเซอร์เมื่อก้านเพนดูลัมอยู่ที่จุดสมดุล.....	39
3.14 เมื่อก้านเพนดูลัมเกิดมีสิ่งรบกวน.....	39
3.15 ลักษณะของจุดสัมผัสพื้น.....	40
3.16 ลักษณะการติดตั้งอื่นๆ.....	40



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

เทคโนโลยีที่เกี่ยวกับการทรงตัวของวัตถุนั้น ได้มีการคิดค้นและพัฒนาให้ดีขึ้นอยู่ตลอดเวลา ซึ่งหลักการและทฤษฎีต่าง ๆ นั้น ได้ถูกนำมาใช้ประโยชน์มากมาย ไม่ว่าจะเป็น สิ่งปลูกสร้างอาคาร บ้านเรือน, หุ่นยนต์ในอุตสาหกรรม หรือ อาจเป็นยานพาหนะต่างๆ เป็นต้น

ระบบแบบอินเวอร์ทเพนดูลัม ถือว่าเป็นระบบหนึ่งที่เกี่ยวข้องกับการทรงตัวของวัตถุซึ่ง ได้ถูกนำมาประยุกต์ใช้ในรูปแบบต่างๆ มากมาย ล่าสุดได้มีการนำเอาระบบนี้มาใช้กับ SEGWAY

SEGWAY เป็นยานพาหนะชนิดหนึ่งที่สามารถรักษาการทรงตัวของมันได้ แม้ว่าจะมีเพียงสองล้อก็ตาม ซึ่งทำให้ผู้วิจัยมีความสนใจเกี่ยวกับการทำงานยานพาหนะนี้ จึงได้ทำการวิจัยและสร้างหุ่นจำลองการขับเคลื่อนแบบ 2 ล้อขึ้นมาโดยใช้หลักการอินเวอร์ทเพนดูลัม

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้เป็นการศึกษา และออกแบบตัวควบคุมหุ่นยนต์ขับเคลื่อนแบบสองล้อ โดยใช้หลักการของ อินเวอร์ทเพนดูลัม เพื่อให้หุ่นยนต์นี้สามารถทรงตัวอยู่ได้โดยไม่ล้ม และเคลื่อนที่ไปยังตำแหน่งที่เราต้องการได้ โดยใช้รีโมทคอนโทรล

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะกล่าวถึงการออกแบบตัวควบคุมแบบออปติมอลคอนโทรล โดยอาศัยหลักการของ อินเวอร์ทเพนดูลัม และ ทฤษฎีของมอเตอร์ เพื่อใช้ในการออกแบบชุดไมโครคอนโทรลเลอร์ dsPIC(Digital Signal Processing PIC) ใช้ไมโครโศป ซึ่งอาศัยหลักการของเพียร์โซอิเล็กทริก ประยุกต์ใช้เป็นตัวเซนเซอร์บอกความเอียง

1.4 ขั้นตอนการศึกษา

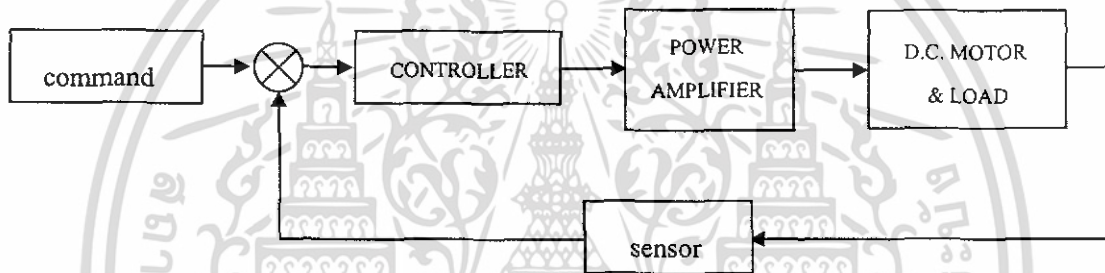
การทำปริญญานิพนธ์นี้เริ่มจากการศึกษาการทำงานของมอเตอร์ หลักการของอินเวอร์ทเพนดูลัม และ ออกแบบระบบควบคุมแบบออปติมอลคอนโทรล โปรแกรมที่ใช้ในการออกแบบเช่น โปรแกรม SolidWork Matlab และใช้ภาษาซี เพื่อออกแบบไมโครคอนโทรลเลอร์แบบ dsPIC 30F เป็นต้น

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับหลักการควบคุมการทรงตัวของหุ่นยนต์ ขับเคลื่อนสองล้อแบบสมดุ

2.1 กล่าวนำ

การควบคุมการทรงตัวและการเคลื่อนที่ของหุ่นยนต์ขับเคลื่อนแบบสองล้อ นั้น เป็นการรวมองค์ประกอบหลายๆองค์ประกอบเข้าด้วยกัน อันได้แก่ หลักการของอินเวอร์ทเพนดูลัม หลักการของดีซีมอเตอร์ หลักการของการควบคุมแบบออปติมอล หลักการทำงานของเซนเซอร์ที่ใช้ หลักการของเพาเวอร์อิเล็กทรอนิกส์ หลักการออกแบบโครงสร้างของรถ เป็นต้น ซึ่งขั้นตอนการทำงานจะแสดงดังแผนภาพต่อไปนี้



ภาพที่ 2.1 แสดงระบบการทำงานของหุ่นยนต์ขับเคลื่อนแบบสองล้อ

จากภาพที่ 2.1 แสดงให้เห็นว่าหุ่นยนต์จะทำงานได้ จำเป็นต้องรู้ถึงหลายองค์ประกอบด้วยกัน ซึ่งจะแบ่งออกเป็น 2 ส่วนใหญ่ดังนี้

2.1.1 หลักการทำงานและทฤษฎีต่างๆ ดังนี้

2.1.1.1 หลักการทำงานของดีซีมอเตอร์

- พฤติกรรมของมอเตอร์ และ โมเดลทางคณิตศาสตร์ของมอเตอร์

2.1.1.2 หลักการของคอนโทรลเลอร์

- หลักการของออปติมอลคอนโทรล

2.1.1.3 หลักการของเซนเซอร์แบบไจโรสโคปและการทำงาน

2.1.1.4 หลักการของวงจรดีซีไดรฟ์มอเตอร์

2.1.2 หลักการออกแบบ ดังนี้

2.1.2.1 การออกแบบโครงสร้างของรถ

- กำเนิดถึงปัจจัยต่างๆ เช่น น้ำหนัก ความสูง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 การออกแบบคอนโทรลเลอร์

- การออกแบบ ในโปรแกรม MATLAB
- การออกแบบการเขียนโปรแกรมในไมโครคอนโทรลเลอร์

2.1.2.3 การออกแบบวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

2.2 หลักการทำงานของดี.ซี.มอเตอร์

ดี.ซี.มอเตอร์เป็นทรานส์ดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษคือ แรงบิดของเพลลาของดี.ซี.มอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์แรงบิดของเพลลาของดี.ซี.มอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์แรงบิดของเพลลาของดี.ซี.มอเตอร์จะได้จากผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการแสดงในรูปแบบที่ 2.4 ในที่นี้กระแสที่ไหลในขดลวดตัวนำจะสร้างฟลักซ์ที่ประกอบด้วยเส้นแรงแม่เหล็ก ϕ และขดลวดตัวนำเหล่านั้นอยู่ห่างจากศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสเท่ากับ

$$T = K\phi I$$

เมื่อ	T	คือ แรงบิดของเพลลา มีหน่วยเป็นนิวตัน – เมตร
	ϕ	คือ เส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์
	I	คือ กระแสเป็นแอมแปร์
	K	คือ ค่าคงที่

ดังนั้นแรงบิดของเพลลาจะเป็นสัดส่วน โดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแสเมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิด โวลต์เตจตกคร่อมตัวมันเอง โวลต์เตจนี้จะ เป็นสัดส่วนกับความเร็วของเพลลาของมอเตอร์และด้านกรไหลของกระแสความสัมพันธ์ระหว่าง โวลต์เตจย้อนกลับนี้และความเร็วของเพลลามอเตอร์คือ

$$E = K\phi\omega$$

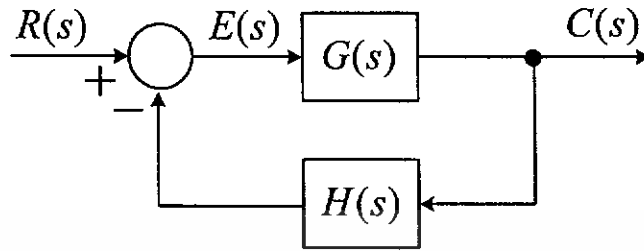
เมื่อ	E	คือ โวลต์เตจย้อนกลับ emf มีหน่วยเป็นโวลต์
	ϕ	คือ เส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์
	ω	คือ ความเร็วของมอเตอร์ มีหน่วยเป็นเรเดียน / วินาที

เอกสาร 2.3 หลักการของระบบควบคุมที่เหมาะสมที่สุด(Optimal Control System) ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หลักการของระบบควบคุมที่เหมาะสมที่สุด(Optimal Control System)

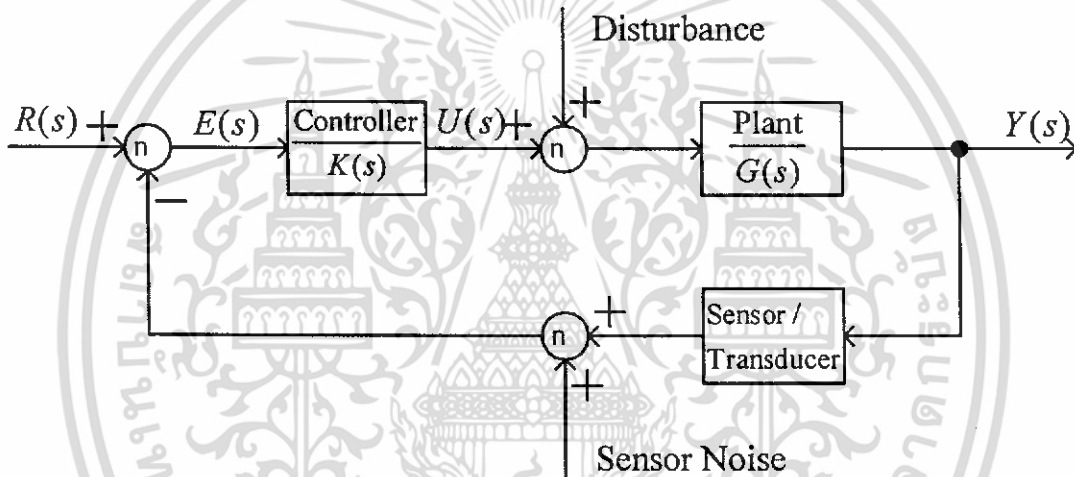
2.3.1 หลักการออกแบบระบบควบคุม

การออกแบบตัวควบคุมในแนวตั้งเดิมนั้น ถ้าเราเขียนแผนภาพกรอบง่ายๆ จะได้



ภาพที่ 2.2 แสดงการออกแบบตัวควบคุมในแนวตั้งเดิม

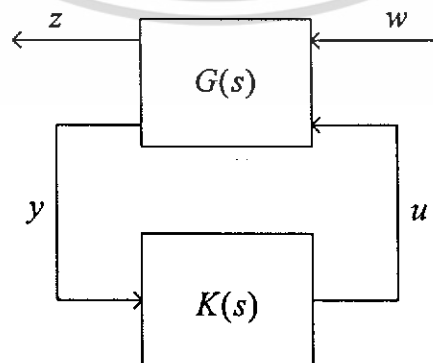
ถ้าพิจารณาว่ามี สิ่งรบกวน ในวงควบคุม เราก็อาจเขียนแผนภาพกรอบใหม่ ได้เป็น



ภาพที่ 2.3 แสดงถึงสิ่งรบกวนที่เกิดขึ้นในวงควบคุม

การออกแบบตัวควบคุมในแนวใหม่ แสดงด้วยแผนภาพกรอบง่ายๆ ที่ประกอบด้วย

- 1) Generalized Plant $G(s)$ และ
- 2) ตัวควบคุม $K(s)$ ที่ต้องการออกแบบ



ภาพที่ 2.4 แสดงการออกแบบตัวควบคุมในแนวใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

z error signal.	w { disturbances, sensor noise, commands.
y measured variables.	u control input.

ภาพที่ 2.5 แสดงถึงความสัมพันธ์ระหว่างตัวแปรและสัญญาณ

$$\text{Generalized Plant; } G(s) \Leftrightarrow \begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix}$$

ภาพที่ 2.6 แสดงถึงลักษณะของ Generalized Plant

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) + B_1w(t) + B_2u(t) \\ z(t) &= C_1x(t) + D_{12}u(t) \\ y(t) &= C_2x(t) + D_{21}w(t) \end{aligned}$$

$$\begin{array}{l} w(t) \rightarrow \\ u(t) \rightarrow \end{array} \begin{array}{l} \frac{d}{dt}x(t) = Ax(t) + B_1w(t) + B_2u(t) \\ z(t) = C_1x(t) + D_{12}u(t) \\ y(t) = C_2x(t) + D_{21}w(t) \end{array} \begin{array}{l} \rightarrow z(t) \\ \rightarrow y(t) \end{array}$$

Generalized Plant $G(s)$ ประกอบไปด้วย

Plant

$$\begin{array}{l} \frac{d}{dt}x(t) = \boxed{A}x(t) + \quad \quad \quad + \boxed{B_2}u(t) \\ \hline y(t) = \boxed{C_2}x(t) + \quad \quad \quad \end{array}$$

ภาพที่ 2.7 แสดงถึงลักษณะของ Plant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

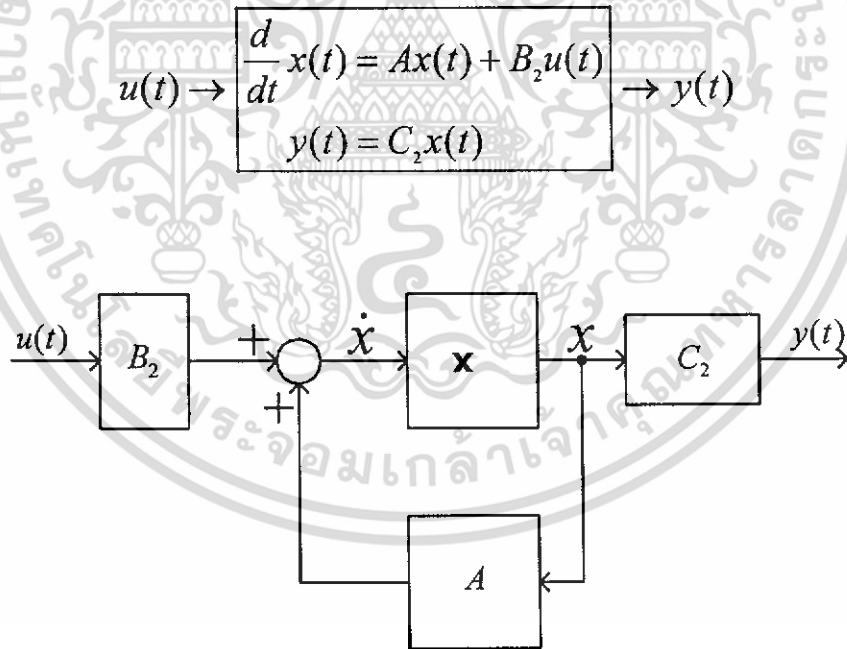
Weighting Functions

$$z(t) = \begin{array}{|c|c|} \hline & \boxed{B_1}w(t) \\ \hline \boxed{C_1}x(t) & + \boxed{D_{12}}u(t) \\ \hline & \boxed{D_{21}}w(t) \\ \hline \end{array}$$

ภาพที่ 2.8 แสดงถึงลักษณะของ Weighting Functions

$$u(t) \rightarrow \begin{array}{|c|c|} \hline \frac{d}{dt}x(t) = Ax(t) + & + B_2u(t) \\ \hline y(t) = C_2x(t) + & \\ \hline \end{array} \rightarrow y(t)$$

Plant Dynamics



ระบบที่ได้รับการปรับ ดัชนีสมรรถนะ (Performance Indexes) ให้มีค่าต่ำที่สุด เรามักเรียก
 กันว่า ระบบควบคุมที่เหมาะสมหรือดีที่สุด
 โดยทั่วไป ดัชนีสมรรถนะ ของระบบเขียนได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$J = \int_0^{t_f} g(x, u, t) dt$$

$u_1 \rightarrow$
 $u_2 \rightarrow$
 \vdots
 $u_m \rightarrow$

Control
system

$\rightarrow x_1$
 $\rightarrow x_2$
 \vdots
 $\rightarrow x_n$

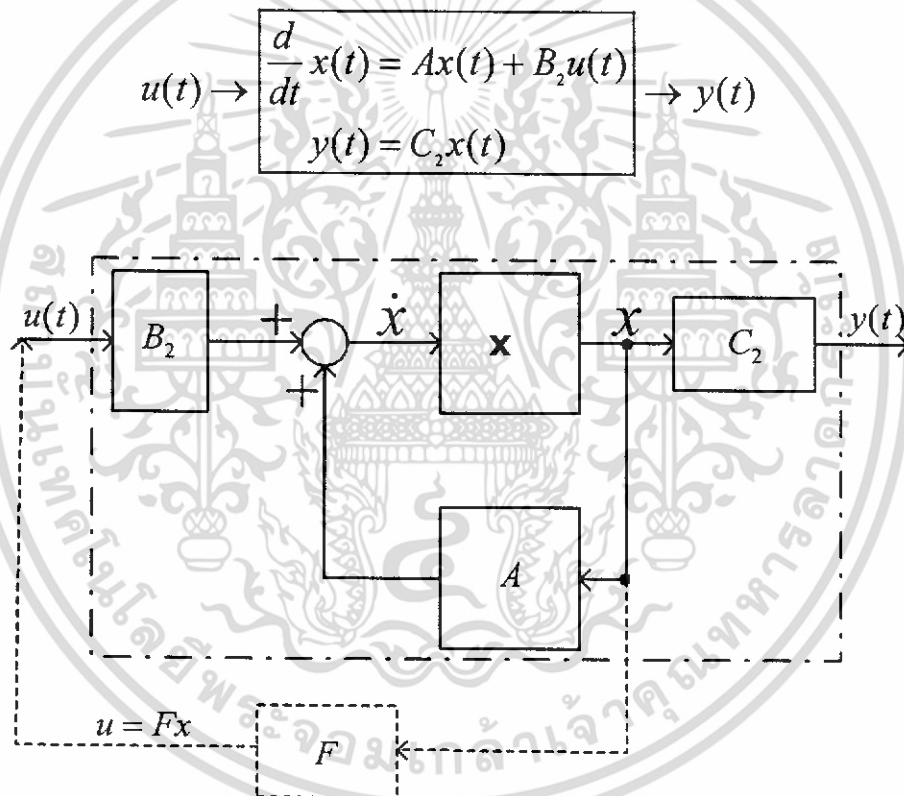
$\left[\begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_m \end{array} \right] =$

$\left[\begin{array}{cccc} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & & & f_{2n} \\ \vdots & & & \vdots \\ f_{m1} & & \cdots & f_{mn} \end{array} \right]$

$\left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right]$

$u = Fx$

การหา $u = Fx$ ที่เป็นฟังก์ชันของ x เพื่อให้ได้ J ต่ำที่สุด



ภาพที่ 2.9 แสดงถึงลักษณะของระบบ LQR

เราจะไปหา P จากสมการริคคาตี (Riccati Equation)

$$A^T P + PA - PB_2 R^{-1} B_2^T P + Q = 0$$

คำตอบ (solutions) ของ P ที่ได้จากการกำหนด Weighting functions Q, R จะเป็นกฎการควบคุมที่เหมาะสมที่สุด

$$u = Fx = \underbrace{-R^{-1} B_2^T P}_{F} x$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณา

$$z(t) = C_1 x(t) + D_{12} u(t) = \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}$$

$$z^T(t) = [C_1 x(t) + D_{12} u(t)]^T = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} C_1 & D_{12} \end{bmatrix}^T$$

$$z^T z = [C_1 x + D_{12} u]^T [C_1 x + D_{12} u] = (C_1^T x^T + D_{12}^T u^T) (C_1 x + D_{12} u)$$

$$= x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u + \underbrace{x^T C_1^T D_{12} u + u^T D_{12}^T C_1 x}_{\text{cross-term}}$$

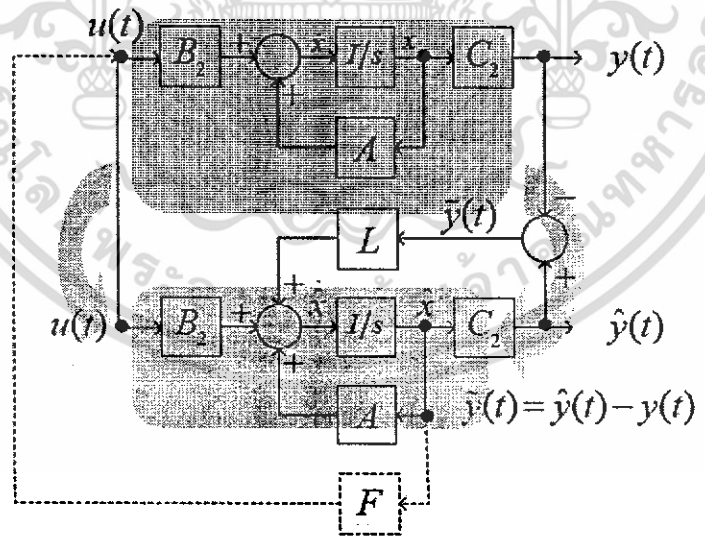
$$= x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u + \underbrace{\left(x^T \underbrace{C_1^T D_{12}}_{=0} u + u^T \underbrace{D_{12}^T C_1}_{=0} x \right)}_{=0}$$

$$z^T z = x^T \underbrace{C_1^T C_1}_{\equiv Q} x + u^T \underbrace{D_{12}^T D_{12}}_{\equiv R} u \equiv x^T Q x + u^T R u$$

$\int_0^{\infty} (z^T z) dt = \|z(t)\|_2^2$

$\|z(t)\|_2^2 \equiv \int_0^{\infty} (x^T Q x + u^T R u) dt = J$

2.3.2 ตัวประมาณค่าสแตต (States Estimator)



ภาพที่ 2.13 แสดงถึงลักษณะของระบบ LQR รวมกับ LQE

เราต้องการสร้างระบบทางพลศาสตร์ หรือที่ทราบกันว่า คือ ตัวสังเกต ที่สามารถประมาณค่าสแตตภายใน (internal state) $x(t)$ จากข้อมูลความรู้ที่ให้มา อันได้แก่ ปริมาณเข้า $u(t)$ และเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริมาณออก $y(t)$ โดยงานนี้สามารถที่จะสำเร็จลุล่วงได้ โดยอาศัยแบบแผนดังที่แสดงในรูปข้างบน

ซึ่งรูปข้างบนแสดงถึงการแยกออกเป็น 2 ส่วน อันได้แก่ ส่วนที่เป็นรูปแบบทางพลศาสตร์ของพลานท์จริงๆ (A, B_2, C_2) และส่วนที่ทำการแก้ไขความคลาดเคลื่อน $L\tilde{y} = L(\hat{y} - y)$ สำหรับเมตริกซ์ L นั้น เราเรียกว่า อัตราขยายของตัวสังเกต เราจะสังเกตได้ว่า ตัวสังเกตจะมี n สเตตภายใน \hat{x} และ สอง ปริมาณเข้า ได้แก่ $u(t)$ และ $y(t)$ เรากำลึงจะแสดงให้เห็นว่า \hat{x} นั้น จะให้ค่าประมาณของทุกสเตต $x(t)$ ถ้า L ถูกเลือกอย่างถูกต้อง ดังนั้น \hat{x} ก็คือ สเตตที่ประมาณ และปริมาณออกของตัวสังเกตนั่นเอง

สมการสเตตของตัวสังเกตคือ

$$\dot{\hat{x}} = A\hat{x} + B_2u + L\tilde{y} = A\hat{x} + B_2u + L(\hat{y} - y)$$

หรือ

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly$$

นี่คือระบบทางพลศาสตร์อันดับที่ n ที่มีสเตตเริ่มต้น (initial state) $\hat{x}(0)$ เท่ากับค่าประมาณเริ่มต้นของสเตต ส่วนเมตริกซ์ L ที่เป็นอัตราขยายของตัวสังเกตนั้น เราจะต้องเลือกค่าของมัน เพื่อให้ว่า ถึงแม้ค่าประมาณเริ่มต้น $\hat{x}(0)$ จะไม่เท่ากับสเตตเริ่มต้นจริง $x(0)$ ครั้นเวลาผ่านไป สเตตที่ประมาณ $\hat{x}(t)$ จะดูเข้าหาสเตตจริง $x(t)$

ปริมาณ $\tilde{y}(t) = \hat{y}(t) - y(t)$ นั้นเราจะเรียกว่า ความคลาดเคลื่อนของการประมาณค่าปริมาณออก (output estimation error) ในการเลือก L เราจะนิยามให้ ความคลาดเคลื่อนของการประมาณค่าสเตต (state estimation error) $\tilde{x}(t) = \hat{x}(t) - x(t) = e(t)$ และเขียนพลศาสตร์ของมันได้ดังนี้

$$\begin{aligned}\dot{\tilde{x}} &= \dot{\hat{x}} - \dot{x} = \dot{e} \\ &= [A\hat{x} + B_2u + L(\hat{y} - y)] - [Ax + B_2u] \\ &= A(\hat{x} - x) + L(\hat{y} - y) \\ &= A(\hat{x} - x) + LC_2\hat{x} - Ly \\ &= A(\hat{x} - x) + LC_2\hat{x} - LC_2x \\ &= A(\hat{x} - x) + LC_2(\hat{x} - x) \\ &= (A + LC_2)(\hat{x} - x)\end{aligned}$$

หรือ

$$\dot{\tilde{x}} = (A + LC_2)\tilde{x} \equiv A_o\tilde{x}$$

เราจะสังเกตได้ว่าปริมาณควบคุมทางเข้า (control input) u ไม่ปรากฏให้เราเห็นเลยจากการหักล้างกันเอง ทั้งนี้ก็เนื่องมาจากว่าปริมาณเข้า u ได้ถูกป้อนเข้าตรงๆ เข้าไปใน ตัวสังเกต โดยเอกสารผ่านเมตริกซ์ B_2 จำนวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการนี้เป็นที่ทราบกันก็คือ ความคลาดเคลื่อนพลศาสตร์ (error dynamics) ซึ่งหลายๆ ที่มาในทฤษฎีระบบควบคุมป้อนกลับ แขนงอยู่บนจุดสำคัญในเรื่องของการวิเคราะห์ความคลาดเคลื่อนพลศาสตร์ เช่นวิชาที่เกี่ยวกับการวัดทางวิศวกรรม (Engineering Measurements) เป็นต้น จากสมการนี้ ($\dot{\tilde{x}} = (A + LC_2)\tilde{x} \equiv A_o\tilde{x}$) เราจะเห็นได้อย่างง่ายดายเลยว่า トラบเท่าที่เราเลือกค่าอัตราขยายของตัวสังเกต L เพื่อที่ว่าเมตริกซ์วงปิดของตัวสังเกต (closed-loop observer matrix) จะให้มีเสถียรภาพแบบเชิงเส้นกำกับ (asymptotically stable) กล่าวคือความคลาดเคลื่อนของการประมาณ (estimation error) $\tilde{x}(t)$ จะเข้าหาศูนย์อย่างเส้นกำกับ (zero asymptotically) เมื่อไรก็ตามที่ความคลาดเคลื่อนเริ่มต้นของการประมาณ (initial estimation error) $\tilde{x}(0) = \hat{x}(0) - x(0)$ ได้เกิดขึ้น

จะว่าไปแล้ว มันก็ไม่ได้เป็นการยากเลย ในการเลือกค่าของ L เพื่อให้ $(A + LC_2)$ มีเสถียรภาพแบบเชิงเส้นกำกับ (A.S.) ถ้าเราเปรียบเทียบปัญหานี้ กับ ปัญหาของการเลือกค่าอัตราขยายของตัวแปรสแตตป้อนกลับ (state variable feedback; SVFB) F เพื่อว่า

$$A_{cl} = (A + B_2F)$$

จะได้มีเสถียรภาพแบบเชิงเส้นกำกับ ปัญหาในการออกแบบตัวสังเกตนั้น การออกแบบเพื่อหาค่าของเมตริกซ์ L จะอยู่ทางด้านซ้ายมือ (ของ C_2 ใน $(A + LC_2)$) ในขณะที่ ปัญหาของการออกแบบค่าอัตราขยายของตัวแปร สแตตป้อนกลับ F นั้น อยู่ทางด้านขวามือ (ของ B_2 ใน $(A + B_2F)$) ฉะนั้นในตอนนี้ เราก็สามารถที่จะมองปัญหาของการหา L ได้ในลักษณะเดียว เช่นกันกับที่ผ่านมา โดยการสลับตำแหน่ง (transposition) ของเมตริกซ์

$$A_o^T = (A + LC_2)^T = A^T + C_2^T L$$

ขณะนี้ สมการข้างบน เมื่อเราดูแล้ว จะเห็นได้ว่ามันก็เหมือนกับปัญหาของการเลือกค่าอัตราขยายของตัวแปรสแตตป้อนกลับ F เพราะปัญหาของการออกแบบหาค่าเมตริกซ์อัตราขยาย L^T ในตอนนี้ มันก็อยู่ทางด้านขวามือของ C_2^T ใน $(A^T + C_2^T L^T)$ เหมือน F ที่อยู่ทางด้านขวามือของ B_2 ใน $(A + B_2F)$ จะอย่างไรก็ตาม เราเห็นได้แล้วว่า ปัญหาของการออกแบบเพื่อเลือกค่าอัตราขยายของตัวแปรสแตตป้อนกลับ F นั้น ใช้ (A, B_2) ในขณะที่ การออกแบบตัวสังเกต ใช้ (A, C_2) จริงๆ แล้ว ทั้งสองปัญหานี้ มีลักษณะเหมือนกัน กล่าวคือ ถ้าเราเปรียบเทียบ การออกแบบอัตราขยายของตัวแปรสแตตป้อนกลับที่เกี่ยวข้องกับ (A, B_2, F) ด้วย (A^T, C_2^T, L^T) ในการออกแบบตัวสังเกต

ดังนั้น การออกแบบตัวสังเกต เพื่อให้ระบบมีเสถียรภาพด้วย วิธีการหนึ่ง จึงอาจทำได้โดย

- เปลี่ยนชื่อจาก (A^T, C_2^T) ไปเป็น (A, B_2)
- ใช้เทคนิคในการออกแบบเพื่อหาค่าอัตราขยายของตัวแปรสแตตป้อนกลับเพื่อให้ระบบ

มีเสถียรภาพ F โดยวิธีการใดวิธีการหนึ่ง เช่น เทคนิคในการวางตำแหน่งโพล (Poles Assignment

ไม่ว่าการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

or Poles Placement) หรือไม่ว่าจะใช้เทคนิคในการหาอัตราขยาย F ที่เหมาะสมที่สุด จากการที่เราได้วางเงื่อนไขไว้แล้ว

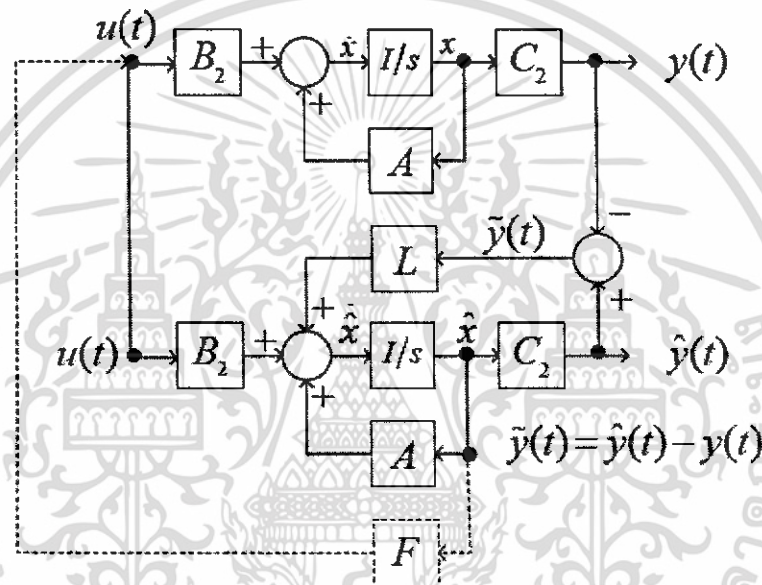
- เปลี่ยนชื่อจาก F^T ไปเป็น L

จากที่ผ่านมา เรามีสมการสเตรตของตัวสังเกตเป็น

$$\dot{\hat{x}} = A\hat{x} + B_2u + L\tilde{y} = A\hat{x} + B_2u + L(\hat{y} - y)$$

หรือ

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly$$



ภาพที่ 2.14 แสดงถึงลักษณะของระบบ LQE

ในการออกแบบตัวควบคุมที่เหมาะสมที่สุด โดยการป้อนกลับด้วยอัตราขยายสเตรต F ไปเป็นปริมาณเข้า $u = Fx$ ให้แก่พลานท์ เพื่อให้ได้ดัชนีสมรรถนะ J มีค่าต่ำสุด เรียกว่า “LQR”

เดิมทีนั้น ยังไม่มีตัวสังเกตหรือตัวประมาณค่าสเตรต เราจึงป้อนกลับจากการวัดสเตรต x ตรงๆ มาบัดนี้ เรามีตัวประมาณค่าสเตรต x นั้นแล้ว ซึ่งค่าประมาณของ x ก็คือ \hat{x} ถ้าหากว่าเราใช้วิธีการหาอัตราขยาย L เช่นเดียวกับที่หา F ใน “LQR” แล้ว เราจะเรียกมันว่า Linear Quadratic Estimator (LQE) และถ้าเราใช้การป้อนกลับด้วยอัตราขยายสเตรต F กับ \hat{x} เพื่อไปเป็นปริมาณเข้า $u = F\hat{x}$ ให้แก่พลานท์ เราจะเรียกตัวควบคุมที่มีทั้ง LQE และ LQR รวมกันนี้ว่า Linear Quadratic Gaussian (LQG) Con-troller ดังนั้น เมื่อเราแทนค่า $u = F\hat{x}$ (โดยที่มันคือปริมาณออกของ ตัวควบคุม LQG) ในสมการสเตรตของตัวสังเกต จะได้

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly$$

$$\dot{\hat{x}} = (A + B_2F + LC_2)\hat{x} - Ly$$

สรุป สมการสแตตและสมการปริมาณออก ของตัวควบคุม LQG คือ

$$\dot{\hat{x}} = (A + B_2F + LC_2)\hat{x} - Ly$$

$$u = F\hat{x}$$

สามารถเขียนในรูปชวเลข (shorthand) ได้เป็น

$$y(t) \rightarrow \left[\begin{array}{c|c} A + B_2F + LC_2 & -L \\ \hline F & 0 \end{array} \right] \rightarrow u(t)$$

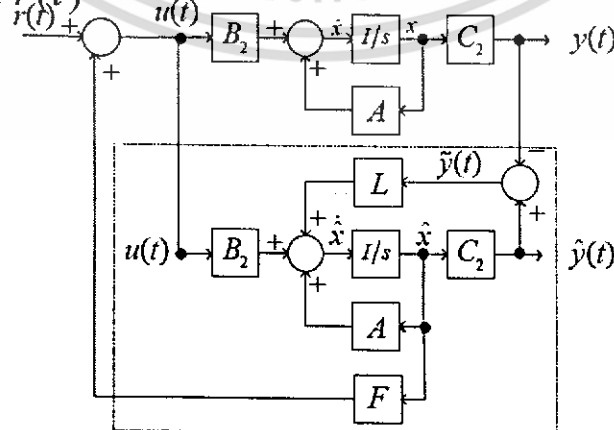
ซึ่งตัวควบคุม LQG นี้ จะออกแบบเพื่อนำไปควบคุมพลานท์

$$u(t) \rightarrow \left[\begin{array}{l} \frac{d}{dt}x(t) = Ax(t) + B_2u(t) \\ y(t) = C_2x(t) \end{array} \right] \rightarrow y(t)$$

ต่อไป ดังนี้

$$y(t) \rightarrow \left[\begin{array}{c|c} A + B_2F + LC_2 & -L \\ \hline F & 0 \end{array} \right] \rightarrow u(t) \rightarrow \left[\begin{array}{c|c} A & B_2 \\ \hline C_2 & 0 \end{array} \right] \rightarrow y(t)$$

ระบบที่มีปริมาณเข้า $r(t)$



ความคลาดเคลื่อนของการประมาณค่าปริมาณออก (output estimation error)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\tilde{y} = \hat{y} - y$$

ความคลาดเคลื่อนของการประมาณค่าสแตต (state estimation error)

$$\tilde{x} = \hat{x} - x = e$$

ดังนั้น อนุพันธ์ ของความคลาดเคลื่อน หรือความคลาดเคลื่อนทางพลศาสตร์

$$\dot{\tilde{x}} = \dot{\hat{x}} - \dot{x} = \dot{e}$$

แต่จากที่สมการสแตตของตัวประมาณ

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly$$

และสมการสแตตของระบบทางพลศาสตร์

$$\dot{x} = Ax + B_2u$$

เราจึงมาถึงตรงที่ว่า

$$\begin{aligned} \dot{\tilde{x}} &= \dot{\hat{x}} - \dot{x} = \dot{e} \\ &= [(A + LC_2)\hat{x} + B_2u - Ly] - [Ax + B_2u] \\ \dot{\tilde{x}} &= A\hat{x} + LC_2\hat{x} - Ly - Ax \\ &= A(\hat{x} - x) + LC_2\hat{x} - Ly \\ &= A(\hat{x} - x) + LC_2\hat{x} - LC_2x \\ &= A(\hat{x} - x) + LC_2(\hat{x} - x) \\ &= (A + LC_2)(\hat{x} - x) \\ &= (A + LC_2)\tilde{x} \end{aligned}$$

เมื่อ $u = r(t) + F\hat{x}$ จะได้

$$\begin{aligned} \dot{x} &= Ax + B_2u; \quad \boxed{u = r(t) + F\hat{x}} \\ &= Ax + B_2r(t) + B_2F\hat{x}; \quad \boxed{\hat{x} = \tilde{x} + x} \\ &= Ax + B_2r(t) + B_2F(\tilde{x} + x) \\ \dot{x} &= (A + B_2F)x + B_2F\tilde{x} + B_2r(t) \end{aligned}$$

ระบบวงปิด ที่ประกอบด้วย พลานท์ และ ตัวควบคุม จะมีสมการสแตต เป็น

$$\begin{aligned} \dot{x} &= (A + B_2F)x + B_2F\tilde{x} + B_2r(t) \\ \dot{\tilde{x}} &= (A + LC_2)\tilde{x} \end{aligned}$$

และถ้าพิจารณาว่าปริมาณออกของพลานท์ เป็นปริมาณออก แล้ว จะได้

$$\begin{aligned}\dot{x} &= (A + B_2 F)x + B_2 F\tilde{x} + B_2 r(t) \\ \dot{\tilde{x}} &= (A + LC_2)\tilde{x} \\ \begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} &= \begin{bmatrix} (A + B_2 F) & (B_2 F) \\ 0 & (A + LC_2) \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} r(t) \\ y &= [C_2 \quad 0] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = C_2 x\end{aligned}$$

ฟังก์ชันถ่ายโอนของพลานท์

$$u(t) \rightarrow \begin{cases} \dot{x}(t) = Ax(t) + B_2 u(t) \\ y(t) = C_2 x(t) \end{cases} \rightarrow y(t)$$

$$U(s) \rightarrow \boxed{C_2 (sI - A)^{-1} B_2} \rightarrow Y(s)$$

ฟังก์ชันถ่ายโอนวงปิด

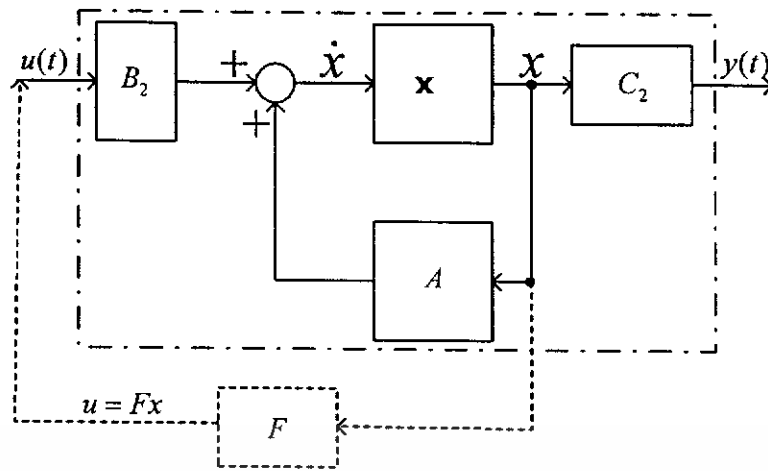
$$\begin{aligned}H(s) &= [C_2 \quad 0] \begin{bmatrix} sI - (A + B_2 F) & -(B_2 F) \\ 0 & sI - (A + LC_2) \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} \\ &= C_2 [sI - (A + B_2 F)]^{-1} B_2\end{aligned}$$

ความเป็นทวิ และความสามารถสังเกต (Duality and Observability)

ถ้าให้ พลานท์ (A, B_2, C_2) มา จะเป็นที่ยูกันเลยว่า พลานท์ (A^T, C_2^T, B_2^T) ก็คือระบบคู่ สำหรับ (A, B_2, C_2) ในระบบคู่ที่ว่านี้ ผลที่เกิดขึ้นเนื่องมาจาก ปริมาณเข้า และปริมาณออก สามารถแลกเปลี่ยนระหว่างกันและกัน ได้เป็นอย่างดี

ระบบ (A, B_2, C_2) จะเรียกได้ว่า เข้าถึงได้ (reachable) ถ้าหากว่าปริมาณควบคุมทางเข้า (control input) u สามารถที่จะเลือกขับ (drive) สเตตเริ่มต้น (initial state) $x(0)$ ใดๆ ไปยัง สเตตสุดท้าย (final state) $x(T)$ ที่เราต้องการ ที่เวลาสุดท้าย (final time) T ได้ เรารู้ว่า ถ้าระบบสามารถเข้าถึงได้แล้ว เราก็สามารถที่จะกำหนดตำแหน่งหรือวาง โพล (poles) ณ ที่ใดๆ ได้ตามอำเภอใจของเรา นอกจากนี้เรายังรู้ดีกว่า ถ้าระบบสามารถเข้าถึงได้แล้ว เราสามารถออกแบบเพื่อเลือกค่าอัตราขยายของตัวแปรสเตตป้อนกลับ F ของปัญหา LQR (Linear Quadratic Regulator) โดยสรุปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.15 แสดงถึงลักษณะของระบบ LQR

LQR เป็นปัญหาของการหา F สำหรับระบบ (A, B_2) ซึ่ง $u = Fx$ เพื่อ
 ที่ว่า $J = \int_0^{\infty} (x^T Q x + u^T R u) dt = \int_0^{\infty} (x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u) dt$ มีค่าต่ำที่สุด
 โดย $F = -R^{-1} B_2^T P$ หาได้จาก P ซึ่งเป็นคำตอบของสมการรีกคาติ

$$A^T P + PA - PB_2 R^{-1} B_2^T P + Q = 0$$

เมื่อ A, B_2 ทราบค่าได้จากสมการสเตตของระบบ ($\dot{x} = Ax + B_2 u$) และ เมตริกซ์ Q, R
 เป็นน้ำหนักที่เราให้ความสำคัญแก่การเบี่ยงเบนของสเตตจากจุดกำเนิด ($x^T Q x$) และ
 ค่าใช้จ่ายในการควบคุม ($u^T R u$) ตามลำดับ

2.3.3 ระบบคู่ (Dual System)

$$(A, B_2, C_2) \Leftrightarrow (A^T, C_2^T, B_2^T)$$

จากที่ว่าในระบบคู่ นั้น ผลที่เกิดขึ้นเนื่องมาจาก ปริมาณเข้า และปริมาณออก สามารถ
 แลกเปลี่ยนระหว่างกันและกันได้เป็นอย่างดี

$$G(s) \begin{cases} \frac{d}{dt} x(t) = Ax(t) + B_1 w(t) + B_2 u(t) \\ z(t) = C_1 x(t) + D_{12} u(t) \\ y(t) = C_2 x(t) + D_{21} w(t) \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Generalized Plant $G(s)$ คือ
$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \boxed{A} & B_1 & \boxed{B_2} \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix}$$

LQR เป็นปัญหาของการหา F สำหรับระบบ (A, B_2) เพื่อที่ว่า

$$J_C = \int_0^{\infty} (x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u) dt \text{ มีค่าต่ำที่สุด}$$

ระบบคู่ของ $G(s)$ คือ $G^T(s)$ หรือ
$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \boxed{A^T} & C_1^T & \boxed{C_2^T} \\ \hline B_1^T & 0 & D_{21}^T \\ B_2^T & D_{12}^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix}$$

LQE เป็นปัญหาของการหา L สำหรับระบบ (A^T, C_2^T) เพื่อที่ว่า

$$J_E = \int_0^{\infty} (x^T B_1 B_1^T x + u^T D_{21} D_{21}^T u) dt \text{ มีค่าต่ำที่สุด}$$

$$G(s) \begin{cases} \frac{d}{dt} x(t) = Ax(t) + B_1 w(t) + B_2 u(t) \\ z(t) = C_1 x(t) + D_{12} u(t) \\ y(t) = C_2 x(t) + D_{21} w(t) \end{cases}$$

LQG เป็นปัญหาของ LQR และ LQE รวมกัน เพื่อที่ว่า ทั้ง

$$\left. \begin{aligned} J_C &= \int_0^{\infty} (x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u) dt \\ J_E &= \int_0^{\infty} (x^T B_1 B_1^T x + u^T D_{21} D_{21}^T u) dt \end{aligned} \right\} \text{มีค่าต่ำที่สุด}$$

$$G(s) \begin{cases} \frac{d}{dt} x(t) = Ax(t) + B_1 w(t) + B_2 u(t) \\ z(t) = C_1 x(t) + D_{12} u(t) \\ y(t) = C_2 x(t) + D_{21} w(t) \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับยืมเพื่อการศึกษาเท่านั้น ไม่ควรนำออกนอกห้องสมุด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการสแตตและสมการปริมาณออก ของตัวควบคุม LQG คือ

$$\dot{\hat{x}} = (A + B_2 F + LC_2) \hat{x} - Ly$$

$$u = F\hat{x}$$

$$y(t) \rightarrow \left[\begin{array}{c|c} A + B_2 F + LC_2 & -L \\ \hline F & 0 \end{array} \right] \rightarrow u(t)$$

สรุป LQR ทหา F ; LQE ทหา L ; LQG ทหาทั้ง F และ L

จากที่เราเคยพิจารณาความสัมพันธ์ระหว่างดัชนีสมรรถนะ J และความคลาดเคลื่อน

$z(t)$

$$z(t) = C_1 x(t) + D_{12} u(t) = \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}$$

$$z^T(t) = [C_1 x(t) + D_{12} u(t)]^T = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} C_1 & D_{12} \end{bmatrix}^T$$

$$z^T z = [C_1 x + D_{12} u]^T [C_1 x + D_{12} u] = (C_1^T x^T + D_{12}^T u^T) (C_1 x + D_{12} u)$$

$$= x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u + x^T C_1^T D_{12} u + u^T D_{12}^T C_1 x$$

cross-term

$$= x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u + \underbrace{(x^T \cancel{C_1^T D_{12}^T} u + u^T \cancel{D_{12}^T C_1} x)}_{=0}$$

$$z^T z = x^T \underbrace{C_1^T C_1}_{\equiv Q} x + u^T \underbrace{D_{12}^T D_{12}}_{\equiv R} u \equiv x^T Q x + u^T R u$$

$$\int_0^{\infty} (z^T z) dt = \|z(t)\|_2^2 \quad \|z(t)\|_2^2 \equiv \int_0^{\infty} (x^T Q x + u^T R u) dt = J$$

เราจะเห็นได้ว่า ตัวควบคุม LQR ซึ่งเป็นส่วนที่ออกแบบเพื่อหาค่า F ของตัวควบคุม LQG นั้น เป็นกรณีพิเศษของ \mathcal{H}_2 Optimal Control เพราะตัด cross-term ของ $z^T z$ ทิ้งไป เราจึงเรียก ตัวควบคุม LQG ว่าเป็น \mathcal{H}_2 Basic Problem วงควบคุมที่ประกอบไปด้วยตัวควบคุม LQG และ ฟลันท์ นั้น จะมีฟังก์ชันถ่ายโอนวงปิดเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(s) = [C_2 \quad 0] \begin{bmatrix} sI - (A + B_2F) & -(B_2F) \\ 0 & sI - (A + LC_2) \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix}$$

$$= C_2 [sI - (A + B_2F)]^{-1} B_2$$

เมื่อระบบวงปิดเป็น

$$\dot{x} = (A + B_2F)x + B_2F\tilde{x} + B_2r(t)$$

$$\dot{\tilde{x}} = (A + LC_2)\tilde{x}$$

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} (A + B_2F) & (B_2F) \\ 0 & (A + LC_2) \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} r(t)$$

$$y = [C_2 \quad 0] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = C_2x$$

จะเห็นได้ว่าเมตริกซ์ระบบ (system matrix) วงปิด คือ

$$A_{LOG} = \begin{bmatrix} (A + B_2F) & (B_2F) \\ 0 & (A + LC_2) \end{bmatrix}$$

ดังนั้น เราสามารถหาโพลวงปิด (closed-loop poles) ได้จาก

$$\Delta(s) = |sI_{2n} - A_{LOG}| = \begin{vmatrix} sI_n - (A + B_2F) & -(B_2F) \\ 0 & sI_n - (A + LC_2) \end{vmatrix}$$

เมื่อ I_n เป็นเมตริกซ์เอกลักษณ์ (identity matrix) ขนาด $n \times n$ แต่เนื่องจากว่า A_{LOG} เป็นเมตริกซ์สามเหลี่ยมบน (upper triangular matrix) ดังนั้น ผลคูณของตัวกำหนด ตามเมตริกซ์เฉียง (diagonal matrix) คือ

$$\Delta(s) = |sI_n - (A + B_2F)| |sI_n - (A + LC_2)|$$

เราจะเห็นได้ว่า โพลวงปิด จะมีจำนวน $2n$ ตัว ถ้าเราออกแบบตัวควบคุมที่มีตัวสังเกต หรือตัวประมาณค่าสเตรวมอยู่ด้วย เพราะโพลในวงปิดจะมีทั้งโพลของตัวสังเกตรวมกันอยู่กับ โพลของพลาตซึ่งเรียกกันว่า “หลักการแยกกัน (Separation Principle)” และถ้าจะว่าไปแล้ว มันเป็นเรื่องที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวใจของทฤษฎีควบคุมสมัยใหม่ (Modern Control Theory) เลยกี่ว่าได้ความหมายของ *Separation Principle* ก็คือว่า มันจะมีอยู่ 2 ขั้นตอนการปฏิบัติเพื่อออกแบบ ตัวควบคุม ได้แก่

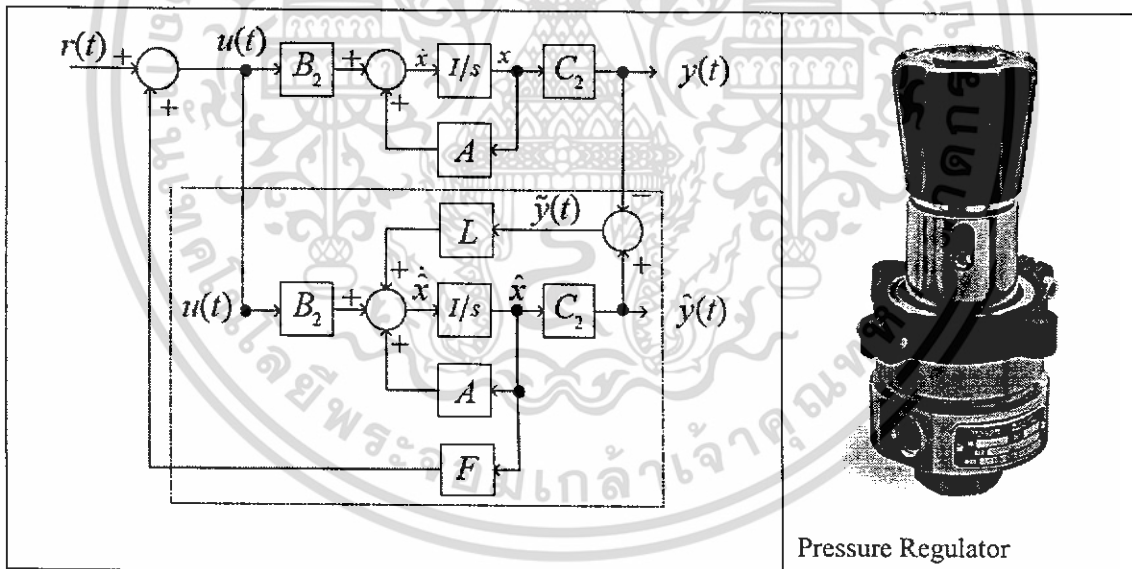
- ใช้เทคนิคอันไหนก็ได้ เพื่อเลือกเมตริกซ์ป้อนกลับ (feedback matrix) F โดยสมมติว่า มันสามารถให้การป้อนกลับด้วย full state ได้

- ออกแบบตัวสังเกต L

โดยที่ พลาที่มีสมการสแตตเป็น $\dot{x} = Ax + B_2u$ ตัวสังเกตมีสมการสแตตเป็น

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly$$

ถ้าหากว่า $r(t)$ ซึ่งเราเคยชินกันว่ามันคือ ปริมาณอ้างอิงทางเข้าหรือคำสั่ง (reference input or command) นั้น มีค่าเท่ากับศูนย์ เราจะได้ $u = r(t) + F\hat{x} = F\hat{x}$ และเรียกปัญหาแบบนี้ว่า “ปัญหาของการรักษาค่า (Regulator Problem)” เช่น เราจะเรียกแหล่งจ่ายแรงดันไฟฟ้ากระแสตรง 5 โวลต์ ที่เราใช้ทดลองวงจรทางดิจิทัล ซึ่งใช้ไอซีประเภทที่ที่แอลว่า ตัวรักษาระดับแรงดันไฟฟ้ากระแสตรง 5 โวลต์ (+5 V.(D.C.) Regulator) ก็หมายถึงว่า ปริมาณออกของแหล่งจ่ายแรงดันไฟฟ้ากระแสตรงจะถูกรักษาให้มีค่าคงที่ที่ 5 โวลต์ อีกตัวอย่างหนึ่งที่เราจะเห็น ได้แก่ ตัวรักษาระดับความดัน (Pressure Regulator) เป็นต้น



ภาพที่ 2.16 แสดงถึงลักษณะของ Presser Regulator

ถ้าจะว่าไปแล้ว “ระบบรักษาค่าอัตโนมัติ (Automatic Regulating System)” ก็คือ ระบบควบคุมป้อนกลับ ที่มี ปริมาณอ้างอิงทางเข้า หรือ ปริมาณออกที่ต้องการ เป็นค่าคงที่ หรือ ไม่ก็เปลี่ยนแปลงไปอย่างช้าๆ กับเวลา โดยที่งานหลักก็เพื่อที่จะคงไว้ซึ่งปริมาณออกจริงๆ ให้อยู่ที่ ค่าที่ต้องการได้ ถึงแม้จะมีสิ่งรบกวน ระบบรักษาค่าอัตโนมัติ ที่มีปริมาณออก หรือตัวแปรที่ถูกควบคุม

เช่น ปริมาณทางกายภาพที่เกี่ยวกับ อุณหภูมิ ความดัน การไหล ระดับของของเหลว และ ความเป็นกรดด่าง (pH) เป็นต้น เราจะเรียกว่า “การควบคุมกระบวนการ (Process Control)”

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 หลักการการทำงานของใจโรสโคป

ใจโรสโคปเป็นอุปกรณ์ที่น่าพิศวงเป็นอย่างยิ่ง เพราะการหมุนของมันค่อนข้างแปลกและคล้ายกับว่ามันทำท่ายกกับแรงโน้มถ่วงได้ คุณสมบัติอันพิเศษนี้สามารถนำไปประยุกต์ใช้ได้ตั้งแต่รถจักรยานจนถึงยานขนส่งอวกาศ เครื่องบินโดยสารทุกประเภทมีใจโรสโคปไว้สำหรับทำเป็นเข็มทิศ และระบบนำร่องอัตโนมัติ สถานีอวกาศ Mir ของรัสเซียใช้ใจโรสโคปจำนวน 11 อัน เพื่อบังคับให้แผงโซลาร์เซลล์หันไปในทิศทางเดียวกับดวงอาทิตย์ตลอดเวลา

การเคลื่อนที่แบบใจโรจะเกิดกับมวลทุกชนิดใน โลกที่มีการหมุน ซึ่งมีการนำมาประยุกต์ใช้ในลักษณะต่างๆมากมายนับไม่ถ้วน ซึ่งพื้นฐานของตัวมันมีดังนี้

2.4.1 การควง



ภาพที่ 2.17 ภาพแสดงการหมุนและทิศทางของแรงที่กระทำกับลูกข่าง

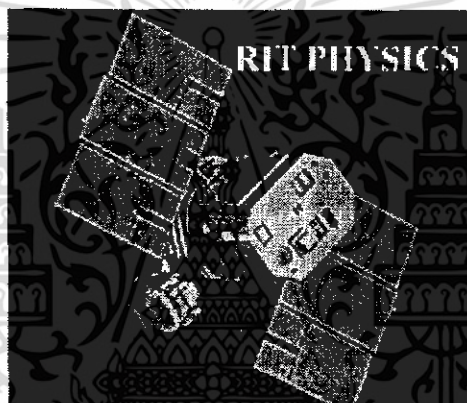
การหมุนของลูกข่างจะมีการหมุน 2 แบบด้วยกัน อย่างแรกคือการหมุนรอบตัวเองของลูกข่าง อย่างที่สองก็คือลูกข่างทั้งลูกจะหมุนรอบแกนกลางอีกแกนหนึ่ง การหมุนแบบนี้เรียกว่า การควง (Precession) โดยลูกข่างไม่ล้ม แต่ถ้าไม่หมุนจะล้ม ซึ่งดูเหมือนกับทำท่ายกกับแรงโน้มถ่วงของโลก

2.4.2 ทฤษฎีและหลักการ

เขวนล้อจักรยานดั่งรูป และหมุนล้อ ตั้งล้อขึ้นให้เอียงดั่งรูป การตั้งล้อในลักษณะนี้ทำให้น้ำหนักของล้อตกลงห่างจากจุดหมุน เกิดการบิดในแนวตั้งฉากกับน้ำหนักของล้อกระทำกับแกนหมุน ดูภาพ

แต่ถ้าเราหมุนหมุนล้อจักรยานหรือใจโร ลองมาดูว่ามีอะไรเกิดขึ้นกับจุดสองจุดบนใจโรบ้าง จากกฎข้อที่หนึ่งของนิวตันที่กล่าวถึงสมดุลของมวลไว้ว่า มวลจะเคลื่อนที่เป็นเส้นตรงด้วยความเร็วคงที่ ถ้าไม่มีแรงภายนอกมากระทำ ดังนั้นจุดบนของใจโรจะถูกแรงบิดจากภายนอก ทำให้เคลื่อนที่ไปทางซ้าย ส่วนจุดล่างจะถูกแรงบิดให้เคลื่อนที่ไปทางขวา ล้อจะเกิดการหมุนแบบควง เมื่อตัวใจโรสโลปหมุนไป 90 องศา จุดบนจะหมุนเปลี่ยนตำแหน่งไป 90 องศา และยังเคลื่อนที่ไปทางซ้าย เช่นเดียวกับจุดล่างเมื่อหมุน ขึ้นมา 90 องศา มันยังคงเคลื่อนที่ไปทางขวา ทำให้ล้อเกิดการหมุนควง ขณะที่จุดบนและจุดล่างเปลี่ยนตำแหน่งไป 90 องศา การเคลื่อนที่ในครั้งแรก จะถูกยกเลิกไปไม่เกิดการพลิกของล้อ ดังนั้นแกนหมุนของใจโรจะเหมือนกับล้ออยู่กับอากาศ ดูเสมือนว่าทำทาบกับแรงโน้มถ่วง ทั้งๆที่เป็นไปตามกฎของฟิสิกส์ทั้งสิ้น

2.4.3 ประโยชน์ของใจโรสโลป

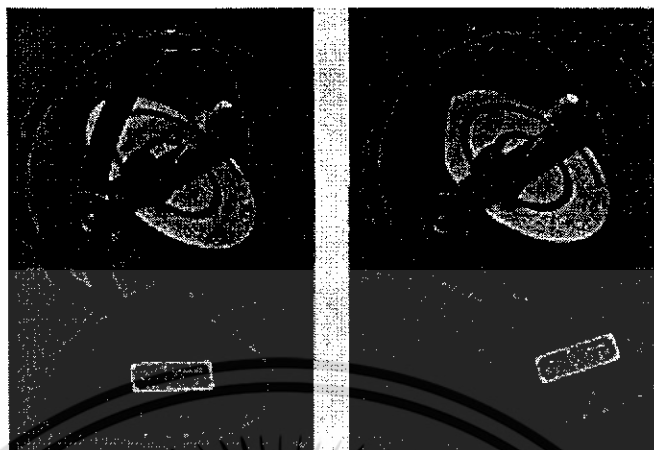


ภาพที่ 2.19 แสดงถึงดาวเทียมซี่ไปโนทิศทางเดียวกัน

ดาวเทียมบางดวงใช้หลักของใจโร เพื่อปรับตำแหน่งของดาวเทียมในอวกาศ ดังภาพที่ 2.19 ภายในดาวเทียมจะประกอบด้วยล้อ 3 อัน ตั้งฉากซึ่งกันและกัน แต่ละอันมีมอเตอร์และเบรกไว้สำหรับควบคุมการหมุน เมื่อล้อเริ่มหมุน ดาวเทียมจะเริ่มหมุนเช่นกันแต่ไปทิศตรงกันข้ามกับล้อ หลังจากที่อยู่ ณ ตำแหน่งที่ต้องการ ก็บังคับให้ล้อหยุดหมุน ดาวเทียมก็จะหยุดหมุนตามไปด้วย เมื่อใช้วิธีนี้เราสามารถที่จะบังคับทิศทางการเคลื่อนที่ได้โดยไม่ต้องใช้เชื้อเพลิงหรือพลังงานมากมายนัก รูปบน ขณะที่ดาวเทียมโคจรอยู่รอบโลก ตำแหน่งของแผงโซลาร์ อาจจะไม่ตรงกับแสงอาทิตย์ ดังนั้นจึงต้องมีการปรับแต่งตำแหน่งของดาวเทียมเสมอ เพื่อให้พลังงานแสงที่ตกลงบนแผงโซลาร์มากที่สุด

เมื่อไม่มีแรงบิดหรือทอร์กกระทำจากภายนอก โมเมนตัมเชิงมุมทั้งขนาดและทิศทางจะคงที่ ดังเช่น เข็มทิศใจโรสโลปที่หมุนอยู่ในเรือ ถ้าไม่มีทอร์กภายนอกไปกระทำมันจะไม่เปลี่ยนทิศทาง แม้ว่าเรือลำนั้นจะเคลื่อนที่รอบโลกไปที่ใดก็ตามดังภาพ ดังนั้นเข็มทิศใจโรสโลปจึงนำไปใช้ในระบับนำร่อง ปัจจุบันมีระบบคอมพิวเตอร์ในการคำนวณ ดังนั้นเมื่อเรือเปลี่ยนตำแหน่งตำแหน่งใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไป คอมพิวเตอร์จะคำนวณเทียบกับตำแหน่งของเข็มทิศใจโร คำนวณพร้อมไปกับข้อมูลจากตัววัด
ความเร่ง คอมพิวเตอร์จะคำนวณหาระยะทางจากจุดเริ่มต้นได้ แม่นยำไม่มีผิดพลาด



สังเกตที่หัวจุกซึ่งเป็นแกนหมุนของงานใจโรสโคป จะไม่เปลี่ยนทิศทางแม้ว่าโครงของ
ใจโร สโคปจะถูกจับให้หมุน เหตุผลเพราะไม่แรงบิดไปกระทำกับงานหมุน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

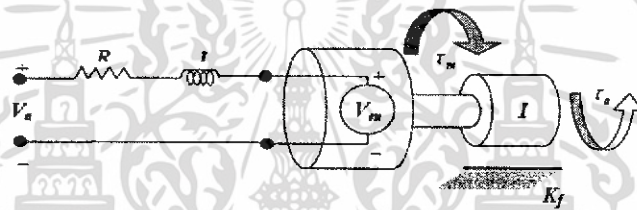
บทที่ 3

การออกแบบระบบควบคุม ไมโครคอนโทรลเลอร์ วงจรมอเตอร์ แบบจำลองทางคณิตศาสตร์ และการออกแบบ ตัวหุ่นยนต์

3.1 แบบจำลองทางคณิตศาสตร์ของหุ่นยนต์

3.1.1 แบบจำลองทางคณิตศาสตร์ของดีซีมอเตอร์

ในเรื่องนี้จะกล่าวถึงพฤติกรรมของ DC Motor โดยจะนำมาเขียนในลักษณะของแบบจำลองทางคณิตศาสตร์(Math Model)ดังนี้



ภาพที่ 3.1 แบบจำลองโครงสร้างของมอเตอร์

เมื่อใส่ Voltage จากภายนอกเข้ามาจะเกิดกระแสขึ้นในขดลวดอาร์เมเจอร์ โดยที่มอเตอร์จะสร้าง torque (T) ขึ้นมาโดยจะเป็นสัดส่วนกับกระแสที่ไหลผ่านขดลวดอาร์เมเจอร์ได้ดังนี้

$$T_m = k_m I \quad (3-1)$$

ความต้านทาน R และขดลวดตัวนำ L ต่ออนุกรมอยู่กับ V_{emf} (Back electromotive force voltage) เกิดขึ้นเนื่องจากขดลวดของมอเตอร์เคลื่อนที่ผ่านสนามแม่เหล็ก โดยสามารถประมาณค่าให้มีความสัมพันธ์เป็นเชิงเส้นกับอัตราเร็วเชิงมุม (ω) ได้ดังนี้

$$V_e = k_e \omega \quad (3-2)$$

ใช้ Kirchoff's Voltage Law (ผลรวมของแรงเคลื่อน ไฟฟ้าภายในเท่ากับศูนย์) ดังนี้

$$V_a - Ri - L \frac{di}{dt} = 0 \quad (3-3)$$

พิจารณาสมการการเคลื่อนที่ของมอเตอร์ โดยประมาณค่าแรงเสียดทานของมอเตอร์ให้เป็นเชิงเส้นกับอัตราเร็วเชิงมุม โดยที่ $I R$ คือ Inertial load of Armature จากกฎของนิวตัน

$$\sum M = \tau_m - k_f \omega - \tau_a = I_x \dot{\omega} \quad (3-4)$$

แทนสมการ (1.1) และ(1.2) ลงใน (1.3)และ(1.4) และจัดใหม่ให้อยู่ในรูปของ Time derivatives

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{d\omega}{dt} = \frac{k_m}{I_R} i - \frac{\tau_o}{I_R} \quad (3-5)$$

$$\frac{di}{dt} = \frac{R}{L} i + \frac{k_e}{L} \omega + \frac{V_a}{L} \quad (3-6)$$

เนื่องจากเป็น DC Motor และเราไม่คิด friction ที่เกิดขึ้นจากสมการที่ (1.5) และ (1.6) จะได้ว่า

$$i = -\frac{k_e}{R} \omega + \frac{1}{R} V_a \quad (3-7)$$

$$\frac{d\omega}{dt} = \frac{k_m}{I_R} i - \frac{\tau_o}{I_R} \quad (3-8)$$

แทนค่าสมการ (3.7) ลงในสมการ (3.8)

$$\frac{d\omega}{dt} = -\frac{k_m k_e}{I_R R} \omega + \frac{1}{I_R R} V_a - \frac{\tau_o}{I_R} \quad (3-9)$$

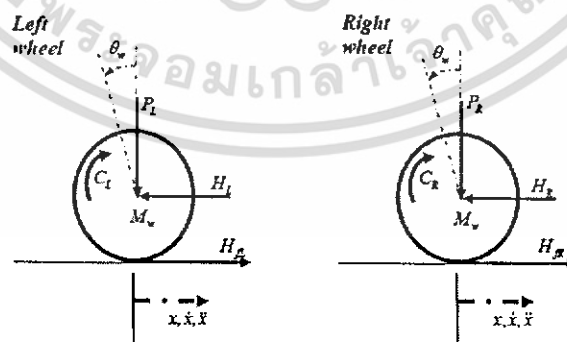
สามารถแสดงเป็น state space model โดยมีตัวแปรที่เราสนใจคือ θ และ ω อินพุตของมอเตอร์คือ Voltage และ Torque

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_m k_e}{I_R R} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_R R} & -\frac{1}{I_R} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_o \end{bmatrix} \quad (3-10)$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \tau_o \end{bmatrix} \quad (3-11)$$

3.1.2 พฤติกรรมของ Balancing Robot

ในการอธิบายพฤติกรรมของ Balancing Robot เราจะทำการแยกพิจารณาออกเป็น 2 ส่วนคือ 1. ส่วนของล้อ 2. ส่วนของ Pendulum โดยทั้งสองส่วนนี้สามารถอธิบายพฤติกรรมของ balancing Robot ได้พิจารณา Body diagram ของล้อ



ภาพที่ 3.2 Free body diagram ของล้อ

ใช้กฎของ Newton โดยพิจารณาผลรวมของแรงแนวแกนนอนในทิศทาง x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}\sum F_x &= Ma \\ M_w \ddot{x} &= H_{Rr} - H_x\end{aligned}\quad (3-12)$$

โมเมนต์รอบจุดศูนย์กลางของล้อ

$$\begin{aligned}\sum M_o &= I\alpha \\ I_w \ddot{\theta}_w &= C_r - H_{Rr}r\end{aligned}\quad (3-13)$$

จาก DC Motor และ Torque สามารถแทนด้วย

$$\tau_m = I_z \frac{d\omega}{dt} + \tau_a \quad (3-14)$$

เปลี่ยนรูปสมการ โดยการแทนค่าตัวแปรของ DC Motor โดย torque ที่ Motor ให้อุปกรณ์คือ

$$C = I_z \frac{d\omega}{dt} = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m V_a}{R} \quad (3-15)$$

แทนค่า (3.15) ใน (3.13)

$$I_w \ddot{\theta}_w = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m V_a}{R} - H_{Rr}r \quad (3-16)$$

$$H_{Rr} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m V_a}{Rr} - \frac{I_w}{r} \ddot{\theta}_w \quad (3-17)$$

นำ (3.17) แทนค่าใน (3.12)

Left wheel

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m V_a}{Rr} - \frac{I_w}{r} \ddot{\theta}_w - H_x \quad (3-18)$$

Right wheel

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m V_a}{Rr} - \frac{I_w}{r} \ddot{\theta}_w - H_x \quad (3-19)$$

เปลี่ยนการเคลื่อนที่เชิงมุมเป็นการเคลื่อนที่เชิงเส้น

Left wheel

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m V_a}{Rr} - \frac{I_w}{r^2} \ddot{x} - H_x \quad (3-20)$$

Right wheel

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m V_a}{Rr} - \frac{I_w}{r^2} \ddot{x} - H_x \quad (3-21)$$

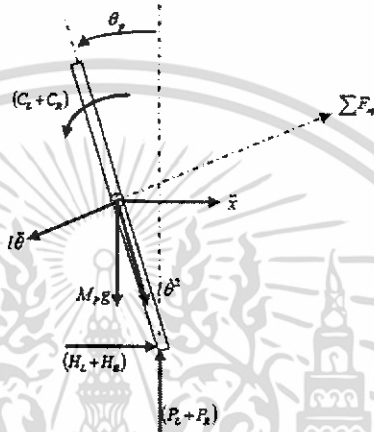
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บวกสมการ (3.20) เข้ากับ (3.21)

$$2\left(M_w + \frac{I_w}{r^2}\right)\ddot{x} = \frac{-2k_m k_s}{Rr^2}\dot{x} + \frac{2k_m V_s}{Rr} - (H_L + H_R) \quad (3-22)$$

3.1.3 หลักการของ Inverted Pendulum

พิจารณาส่วน โครงของรถที่เป็น Inverted Pendulum พิจารณาจาก body diagram



ภาพที่ 3.3 Free body diagram ของ โครงรถส่วนที่เป็นก้าน Pendulum

จาก Newton's law of Motion ผลบวกของแรงตามแนวแกนนอน

$$\begin{aligned} \sum F_x &= M_p \ddot{x} \\ (H_L + H_R) - M_p l \ddot{\theta}_p \cos \theta_p + M_p l \dot{\theta}_p^2 \sin \theta_p &= M_p \ddot{x} \end{aligned} \quad (3-23)$$

ดังนั้น

$$(H_L + H_R) = M_p \ddot{x} + M_p l \ddot{\theta}_p \cos \theta_p - M_p l \dot{\theta}_p^2 \sin \theta_p \quad (3-24)$$

พิจารณาผลบวกของแรงตามแนวแกนตั้งฉากกับ Pendulum

$$\sum F_y = M_p \ddot{x} \cos \theta_p \quad (3-25)$$

$$(H_L + H_R) \cos \theta_p + (P_L + P_R) \sin \theta_p - M_p g \sin \theta_p - M_p l \ddot{\theta}_p = M_p \ddot{x} \cos \theta_p$$

พิจารณาผลบวกของ Moment รอบจุด C.M ของ Pendulum

$$\sum M_o = I \alpha$$

$$-(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p - (C_L + C_R) = I_p \ddot{\theta}_p \quad (3-26)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Torque ที่ให้แก่ pendulum ที่ได้จาก Motor จากสมการ (3-15)

$$C_L + C_R = \frac{-2k_m k_e \dot{x}}{R} + \frac{2k_m V_a}{R}$$

แทนค่า $C_L + C_R$ ในสมการ (3.26)

$$\begin{aligned} & -(H_L + H_R) \cos \theta_p - (P_L + P_R) \sin \theta_p - \left(\frac{-2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} \right) = I_p \ddot{\theta}_p \\ & -(H_L + H_R) \cos \theta_p - (P_L + P_R) \sin \theta_p = I_p \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} \end{aligned} \quad (3-27)$$

คูณ -1 ในสมการ (3.25)

$$\left[-(H_L + H_R) \cos \theta_p - (P_L + P_R) \sin \theta_p \right] + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta}_p = -M_p \ddot{x} \cos \theta_p \quad (3-28)$$

แทนค่า (3.27) ลงใน (3.28) จะได้

$$I_p \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta}_p = -M_p \ddot{x} \cos \theta_p \quad (3-29)$$

จากสมการที่ (3.24) แทนค่าลงใน (3.22)

$$2 \left(M_w + \frac{I_w}{r^2} \right) \ddot{x} = \frac{-2k_m k_e \dot{x}}{Rr^2} + \frac{2k_m V_a}{Rr} - M_p \ddot{x} - M_p l \ddot{\theta}_p \cos \theta_p + M_p l \dot{\theta}_p^2 \sin \theta_p \quad (3-30)$$

จัดสมการ (3.29) และ (3.30) ทั้ง 2 สมการเป็น non-linear equations

$$\left(I_p + M_p l^2 \right) \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} + M_p g l \sin \theta_p = -M_p \ddot{x} \cos \theta_p \quad (3-31)$$

$$\frac{2k_m V_a}{Rr} = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2k_m k_e \dot{x}}{Rr^2} + M_p l \ddot{\theta}_p \cos \theta_p - M_p l \dot{\theta}_p^2 \sin \theta_p \quad (3-32)$$

เมื่อให้ $\theta_p = \pi + \phi$ เมื่อ ϕ คือมุมที่มีการเปลี่ยนแปลงน้อยมาก และมีทิศทางไปด้านหน้า เพราะฉะนั้น

$$\cos \theta_p = -1, \quad \sin \theta_p = -\phi \quad \text{and} \quad \left(\frac{d\theta_p}{dt} \right)^2 = 0$$

ทำสมการให้เป็น สมการเชิงเส้น

$$\left(I_p + M_p l^2 \right) \ddot{\phi} - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} - M_p g l \phi = M_p \ddot{x} \quad (3-33)$$

$$\frac{2k_m V_a}{Rr} = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2k_m k_e \dot{x}}{Rr^2} - M_p l \ddot{\phi} \quad (3-34)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดสมการ (3.33) และ (3.34) ใหม่

$$\ddot{\phi} = \frac{M_p I}{(I_p + M_p I^2)} \ddot{x} + \frac{2k_m k_s}{Rr(I_p + M_p I^2)} \dot{x} - \frac{2k_m}{R(I_p + M_p I^2)} V_a + \frac{M_p g I}{(I_p + M_p I^2)} \phi \quad (3-35)$$

$$\ddot{x} = \frac{2k_m}{Rr \left(2M_x + \frac{2I_x}{r^2} + M_p \right)} V_a - \frac{2k_m k_s}{Rr^2 \left(2M_x + \frac{2I_x}{r^2} + M_p \right)} \dot{x} + \frac{M_p I}{\left(2M_x + \frac{2I_x}{r^2} + M_p \right)} \ddot{\phi} \quad (3-36)$$

นำสมการ (3.35) แทนใน (3.34) และแทนค่า (3.36) ใน (3.33) จะได้ จากนั้นเขียนเป็น state space equation จะได้

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_s (M_p I r - I_p - M_p I^2)}{Rr^2 \alpha} & \frac{M_p^2 g I^2}{\alpha} & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & \frac{2k_m k_s (r\beta - M_p I)}{Rr^2 \alpha} & \frac{M_p g I \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m (I_p + M_p I^2 - M_p I r)}{Rr \alpha} \\ 0 \\ \frac{2k_m (M_p I - r\beta)}{Rr \alpha} \end{bmatrix} V_a \quad (3-37)$$

เมื่อกำหนดให้

$$\beta = \left(2M_x + \frac{2I_x}{r^2} + M_p \right) \quad \alpha = \left[I_p \beta + 2M_p I^2 \left(M_x + \frac{I_x}{r^2} \right) \right]$$

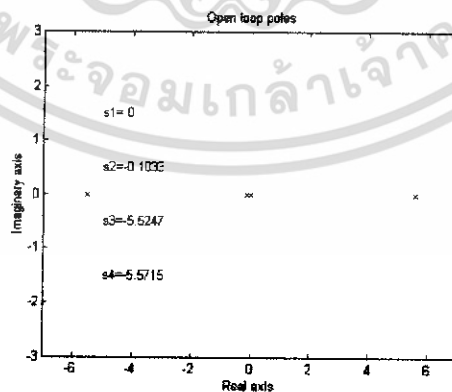
3.2 การออกแบบระบบควบคุม

จากทฤษฎีของออปติมอลคอนโทรลและโมเดลทางคณิตศาสตร์สามารถหาค่าพารามิเตอร์

ต่างๆ ได้ดังต่อไปนี้

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0565 & 2.9476 & 0 \\ 0 & 0.4882 & 30.7856 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad B2 = \begin{bmatrix} 0 \\ 0.0223 \\ 0 \\ -0.1925 \end{bmatrix} \quad C2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

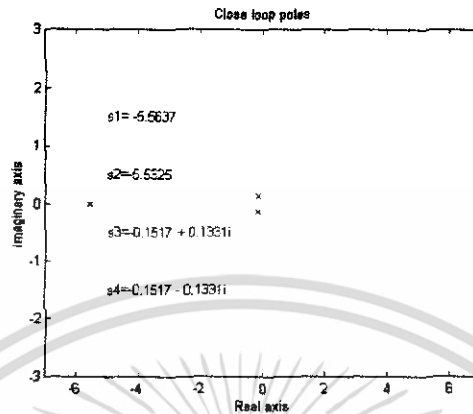
โดยระบบมีตำแหน่งของ Open Loop Poles ดังภาพที่ 3.4



ภาพที่ 3.4 Open Loop Poles ของระบบที่ไม่อยู่ในเสถียรภาพ

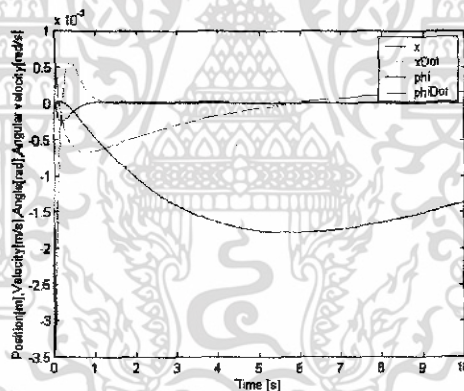
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพ 3.4 จะสังเกตได้ว่าระบบไม่อยู่ในเสถียรภาพ เนื่องจากมีโพลของระบบอยู่ทางซีกขวาของแกนจริง อยู่ที่ตำแหน่งที่ 5.5715 เมื่อทำการออกแบบ LQR และ LQE เพื่อหาค่า Gain F(LQR) และ Gain L(LQE) จะได้ผลของ Close Loop Poles ดังภาพที่ 3.5



ภาพที่ 3.5 Close Loop Pole ของระบบที่อยู่ในเสถียรภาพ

สังเกตตำแหน่งของ Close Loop Poles จะเห็นว่าระบบอยู่ในเสถียรภาพซึ่งผลตอบสนองคือ Impulse เป็นไปดังภาพที่ 3.6



ภาพที่ 3.6 ผลตอบสนองของระบบเมื่อค่าน้ำหนัก $q_{11} = 1$ กับ $q_{33} = 1$ และ $R = 1$

ผลตอบสนองของระบบจะขึ้นอยู่กับค่าการปรับค่าน้ำหนักของเมตริก Q และเมตริก R โดย

$$Q = \begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_{33} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

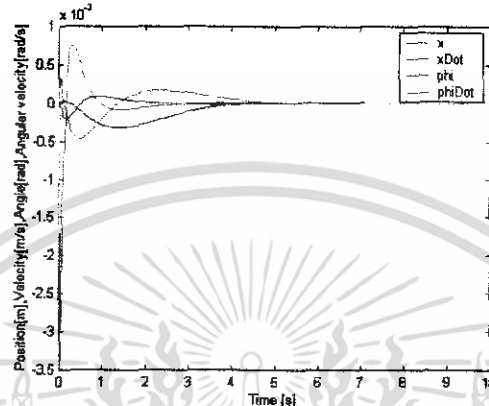
โดยที่ q_{11} จะเป็นตัวน้ำหนักของ X (position)

q_{33} จะเป็นตัวน้ำหนักของ θ (Angular)

$$R = 1$$

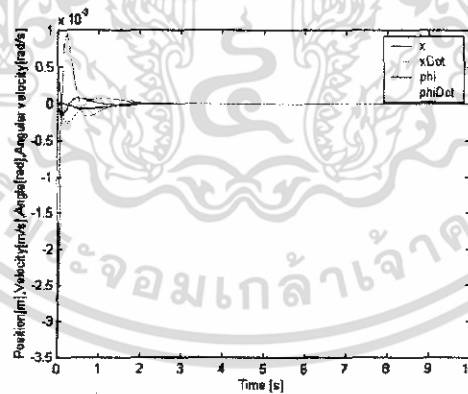
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราทำการเปลี่ยนแปลงค่า Q และ R เป็นค่าต่างจะพบว่าผลตอบสนองของระบบจะเปลี่ยนไป ค่าถ่วงน้ำหนักของเมตริก Q เป็นตัวบ่งชี้ถึงความสำคัญของตัวแปร State เมื่อค่าในเมตริก Q ของตัวแปร State ใดมีค่าสูงแสดงว่าเราให้ความสำคัญต่อผลตอบสนองของตัวแปร State นั้นสูง ส่วนค่าในเมตริก Q ของตัวแปร State ที่มีค่าต่ำบ่งบอกถึงว่าเราให้ความสนใจต่อผลตอบสนองแก่ตัวแปร State นั้นน้อย



ภาพที่ 3.7 ผลตอบสนองเมื่อทำการเปลี่ยนแปลงค่าน้ำหนัก $q_{11}=1000$ กับ $q_{33}=1$ และ $R=1$

จากการเปรียบเทียบกันระหว่างภาพที่ 3.6 กับภาพที่ 3.7 โดยการเพิ่มค่า q_{11} จาก 1 เป็น 1000 โดย q_{11} คือค่าน้ำหนักในเมตริก Q ที่เป็นน้ำหนักของตัวแปร State X(position) จากเส้นกราฟสีน้ำเงินจะเห็นได้ว่า Setting Time มีค่าลดลง



ภาพที่ 3.8 ผลตอบสนองเมื่อค่าน้ำหนักเป็น $q_{11}=1000$ กับ $q_{33}=1000$ และ $R=1/100$

การลดค่าของ R ลงจะทำให้ Setting Time ลดลงเนื่องจากโพลของระบบออกห่างจากแกนจินตภาพ ทำให้ระบบเร็วขึ้น และการเพิ่มค่าน้ำหนักของเมตริก Q ทำให้ Setting Time ลดลงก็เนื่องมาจากเหตุผลเกี่ยวกับการลดค่า R คือทำให้โพลวิ่งออกห่างจากแกนจินตภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบโปรแกรมบนไมโครคอนโทรลเลอร์

เนื่องจากการทำงานของหุ่นยนต์สองล้อนี้ จำเป็นต้องอาศัยการประมวลผลของการคำนวณที่มีความไวค่อนข้างมาก และมีหน่วยความจำให้ใช้มากพอสมควร เหล่านี้ทำให้ต้องเลือกใช้เป็นไมโครคอนโทรลเลอร์แบบดิจิทัล หรือ Digital Signal Processing (dsPIC) ก่อนอื่นจะขอกล่าวข้อมูลเบื้องต้นของไมโครคอนโทรลเลอร์แบบนี้ก่อน

3.3.1 ข้อมูลเบื้องต้นของไมโครคอนโทรลเลอร์แบบ dsPIC

dsPIC คือชื่อของไมโครคอนโทรลเลอร์แบบ 16 บิต มีชื่ออย่างเป็นทางการสำหรับไมโครคอนโทรลเลอร์อนุกรมใหม่นี้ว่า Digital Signal Controller หรือ DSP นั้นหมายความว่า dsPIC เป็นไมโครคอนโทรลเลอร์ที่ได้รับการออกแบบมาเป็นพิเศษเพื่องานประมวลผลสัญญาณดิจิทัลสำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูง

ในการใช้งานนั้นจะเลือกรุ่น dsPIC30F6010 เป็นไมโครคอนโทรลเลอร์ขนาด 80 ขา แบบ Thin Quad Flatpack (TQFP) สาเหตุที่ใช้รุ่นนี้เนื่องจากมีโมดูลของ pulse width modulation ถึง 4 คู่ ซึ่งเท่ากับความต้องการ อีกทั้งหน่วยความจำและ จำนวนพอร์ทัลก็มีให้ใช้มากเพียงพอกับความต้องการอีกด้วย ซึ่งคุณสมบัติต่างๆ จะกล่าวอยู่ในภาคผนวก ค.

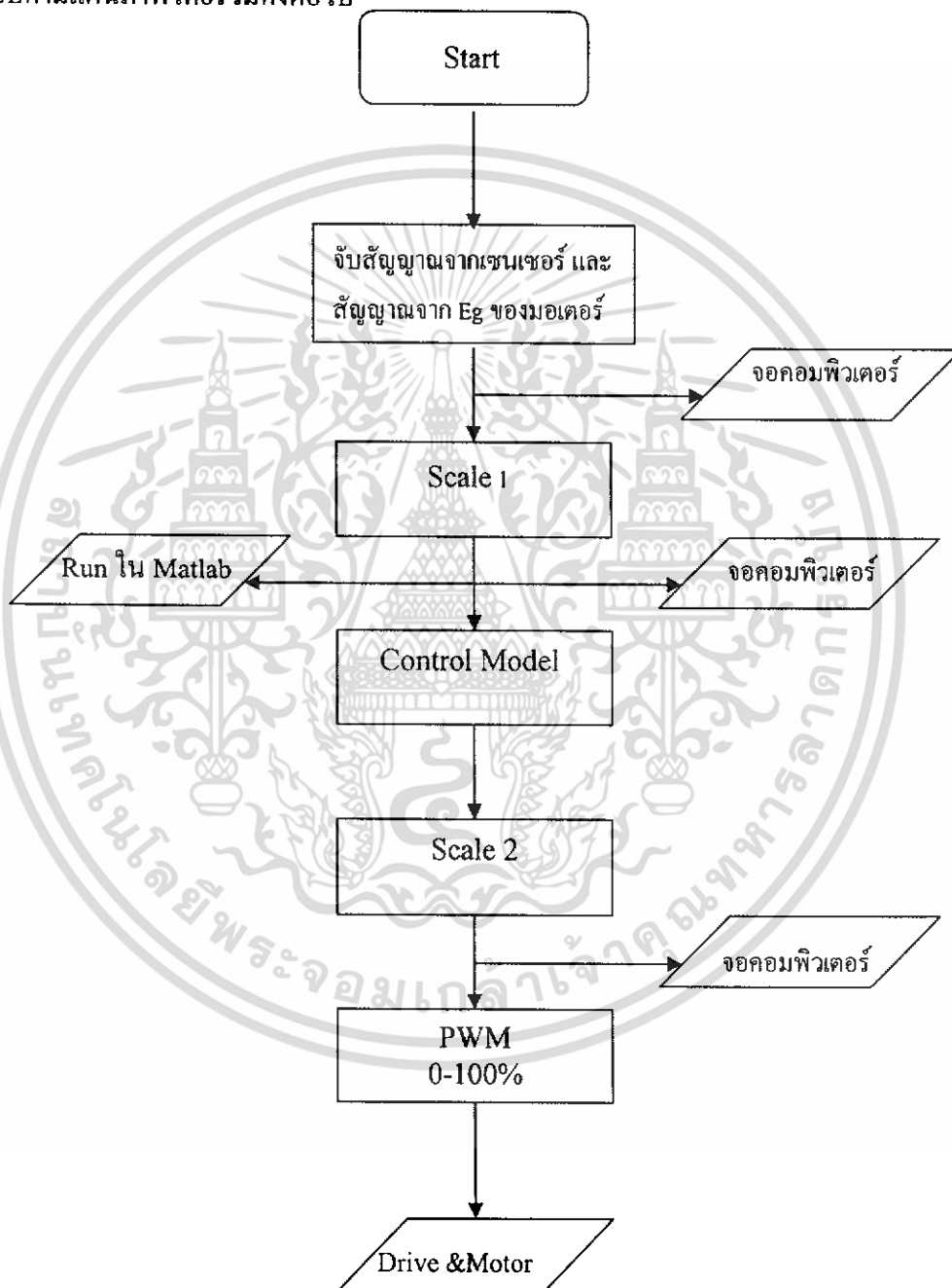


ภาพที่ 3.9 dsPIC30F6010 80-Lead Plastic Thin Quad Flatpack (TQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การออกแบบโปรแกรมในไมโครคอนโทรลเลอร์

ในส่วนของการเขียนโปรแกรมนั้นจะใช้ภาษาซีในการเขียน และใช้โปรแกรม MPLAB IDE เป็นตัวคอมไพล์ การเขียนโปรแกรมทั้งหมดนี้จะอยู่ภายใต้แบบจำลองทางคณิตศาสตร์ที่ได้ศึกษามาแล้วทั้งหมด เพื่อให้การควบคุมมีประสิทธิภาพที่สุด ซึ่งการออกแบบโปรแกรมนั้นจะเริ่มตั้งแต่มีสัญญาณเข้า จำนวนใน โมเดลทางคณิตศาสตร์ จนกระทั่ง ได้สัญญาณเอาต์พุตออกมา ซึ่งจะเป็นไปตามแผนภาพโดยรวมดังต่อไปนี้



ภาพที่ 3.10 แผนภาพโดยรวมของการเขียน โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภาพการทำงานของไมโครคอนโทรลเลอร์ จะเริ่มด้วยการจับสัญญาณจากเซนเซอร์ 2 แหล่ง และสัญญาณจากรีโมทคอนโทรลเลอร์ทั้งหมด 4 ช่องสัญญาณ สัญญาณแรกที่ถูกจับคือสัญญาณที่มาจากรีโมทคอนโทรลเลอร์จากช่องสัญญาณ THRO ซึ่งเป็นสัญญาณพัลส์วิดมอดดูเลชั่นตัวแรกที่ สัญญาณตัวนี้ไว้ใช้สำหรับการบังคับทิศทางเดินหน้า ถอยหลัง ให้กับตัวหุ่นยนต์

สัญญาณที่สอง คือ สัญญาณ AILE ซึ่งเป็นสัญญาณพัลส์วิดตัวที่สองจากรีโมทคอนโทรล สัญญาณนี้เอาไว้ใช้สำหรับบังคับทิศทางเลี้ยวซ้าย ขวาของตัวหุ่นยนต์

สัญญาณตัวที่สาม คือ สัญญาณพัลส์วิดจากเซนเซอร์วัดอัตราเร็วเชิงมุม หรือ ไจโรสโคป ซึ่งจะถูกระดุนการทำงาน หรือ จะทำงานได้เมื่อได้รับสัญญาณ ELEV ซึ่งเป็นสัญญาณพัลส์วิดตัวที่สามจากรีโมทคอนโทรล

สัญญาณที่สี่ คือ สัญญาณ RUDD ซึ่งเป็นสัญญาณพัลส์วิดตัวที่สี่จากรีโมทคอนโทรล สัญญาณนี้เอาไว้ใช้สำหรับการชดเชยสัญญาณ(compensate) ที่เกิดจากความผิดพลาดของเซนเซอร์วัดมุม หรือ สัญญาณ Error จากไจโรสโคป

สัญญาณทั้งสี่นี้จะถูกลำดับการจับสัญญาณไว้แล้ว ด้วยโปรแกรมที่เขียนบนไมโครคอนโทรลเลอร์ สัญญาณทั้งสี่นี้จะมีค่าประมาณ 50Hz หรือ 0.020 ms เพื่อให้มีความละเอียดและความไวต่อการจับระดับสัญญาณจึง Timer Capture 32bit ซึ่งเป็นความเร็วที่สูงที่สุดของการจับสัญญาณ

สัญญาณที่ห้า และ หก คือ แรงดันย้อนกลับของมอเตอร์(Eg)ของมอเตอร์ทั้งสองตัว แต่การวัดสัญญาณ Eg จากมอเตอร์โดยตรงทำได้ยาก ดังนั้นจึงได้นำเครื่องวัดกระแส(Current Sensor)มาเป็นตัววัดกระแส ซึ่งให้สัญญาณออกมาเป็นแรงเคลื่อนทางไฟฟ้า 0-5 โวลต์ สัญญาณนี้จะถูกส่งค่าไปยังช่องของการรับสัญญาณอินพุตอนาล็อก(Analog Input) เพื่อที่จะนำมาคำนวณหาค่าของแรงดันย้อนกลับ และนำมาหาความเร็วเชิงมุมจากโมเดลทางคณิตศาสตร์ของมอเตอร์ ค่าความเร็วนี้จะถูกนำมาคำนวณหาระยะทางที่รถเคลื่อนที่ คอนนี่จะแสดงผลบนหน้าจอคอมพิวเตอร์ เพื่อต่อการสังเกตผล จากนั้นนำตัวเลขของการจับสัญญาณที่ได้ มาทำสเกลให้อยู่ในรูปของความเร็วเชิงมุม (rad/s) และ ระยะทาง(m)

การนำสัญญาณทั้งหมดนี้มาใช้ จะเริ่มจากการหาค่าอ้างอิงของสัญญาณทั้งหมดสัญญาณ โดยจับสัญญาณตอนที่ระบบสมดุล ไม่มีการขยับเขยื้อนตัวหุ่นยนต์ สัญญาณอ้างอิงนี้จะต้องคงที่ตลอดการจับค่าประมาณ 100 ครั้ง จึงจะเสร็จสิ้นการสอบเทียบสัญญาณ (Calibrate) ส่วนการนำค่าตัวเลขจากสัญญาณทั้งหมดมาใช้นั้นกระทำโดย เมื่อป้อนอินพุตให้กับหุ่นยนต์ โดยการบังคับที่รีโมทคอนโทรล หรือ ทำให้ตัวหุ่นยนต์เอียงจากจุดที่เราอ้างอิง สัญญาณที่เข้าไปใหม่จะถูกเปรียบเทียบกับสัญญาณอ้างอิงที่ได้จากขั้นคอนสอบเทียบสัญญาณ สัญญาณที่แตกต่างนี้เองจะถูกนำมาเปลี่ยนค่าเป็นตัวเลขเพื่อเข้าสู่กระบวนการคำนวณต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณตัวที่เจ็ดซึ่งเป็นตัวสุดท้าย คือ สัญญาณที่ใช้วัดโวลต์เตจจากแบตเตอรี่ซึ่งมีค่าประมาณ 0-12 โวลต์ สัญญาณนี้จะต้องถูกแบ่งสัญญาณเป็น 0-5 โวลต์ เพื่อส่งไปให้ช่องรับสัญญาณอะนาล็อกในตัวไมโครคอนโทรลเลอร์

ตัวเลขทั้งหมดที่ได้จากขั้นตอนแรกจะถูกนำมาใช้ในการคำนวณ ในโมเดลทางคณิตศาสตร์ของตัวมอเตอร์ และ ส่วนของออปติมอลคอนโทรลที่ได้จากการออกแบบ โดยอินพุตของคอนโทรลโมเดลจะเป็นค่าตัวเลขขององศาที่เอียงจากจุดอ้างอิง ซึ่งได้มาจากการสัญญาณเซนเซอร์ไจโรสโคป และตัวเลขที่เป็นระยะทางซึ่งได้จากการวัด Eg ของมอเตอร์ จำนวนออกมาเป็นค่าของระยะทาง หลังจากการคำนวณเสร็จสิ้นจะได้ค่า MV (Manipulate Value)

ค่าตัวเลขของค่า MV ที่ได้จะถูกสเกลอีกครั้งให้เป็น 0-100 % เพื่อจะส่งไปให้ส่วนของโปรแกรมที่เขียนเพื่อสร้าง PWM เพื่อให้ได้ duty cycle 0-100% และความถี่ประมาณ 2.5 KHz

ขั้นตอนสุดท้ายของโปรแกรมคือการส่งสัญญาณ PWM 4 คู่ ซึ่งแต่ละคู่จะทำงานเป็นแบบโหมคคอมพลิเมนต์ารี(Complementary mode) คือ มีด้าน high และ ด้าน low ซึ่งจะมีค่าเวลาตายของคู่ PWM (Dead time) มากกว่า 200 นาโนวินาทีขึ้นไป มอเตอร์ 1 ตัวจะใช้ PWM 2 คู่ เพื่อที่จะควบคุมได้ทั้งความเร็วและทิศทางของมอเตอร์ ซึ่งรายละเอียดของวงจรดิจิตอลมอเตอร์จะกล่าวในหัวข้อต่อไป สำหรับโปรแกรมที่เขียนผู้ที่สนใจสามารถดูเพิ่มเติมถึงหลักการเขียนโปรแกรม และ ภาษาที่ใช้เขียน ได้สามารถดูเพิ่มเติมได้ที่ภาคผนวก ก

3.4 วงจรควบคุมการทำงานของมอเตอร์

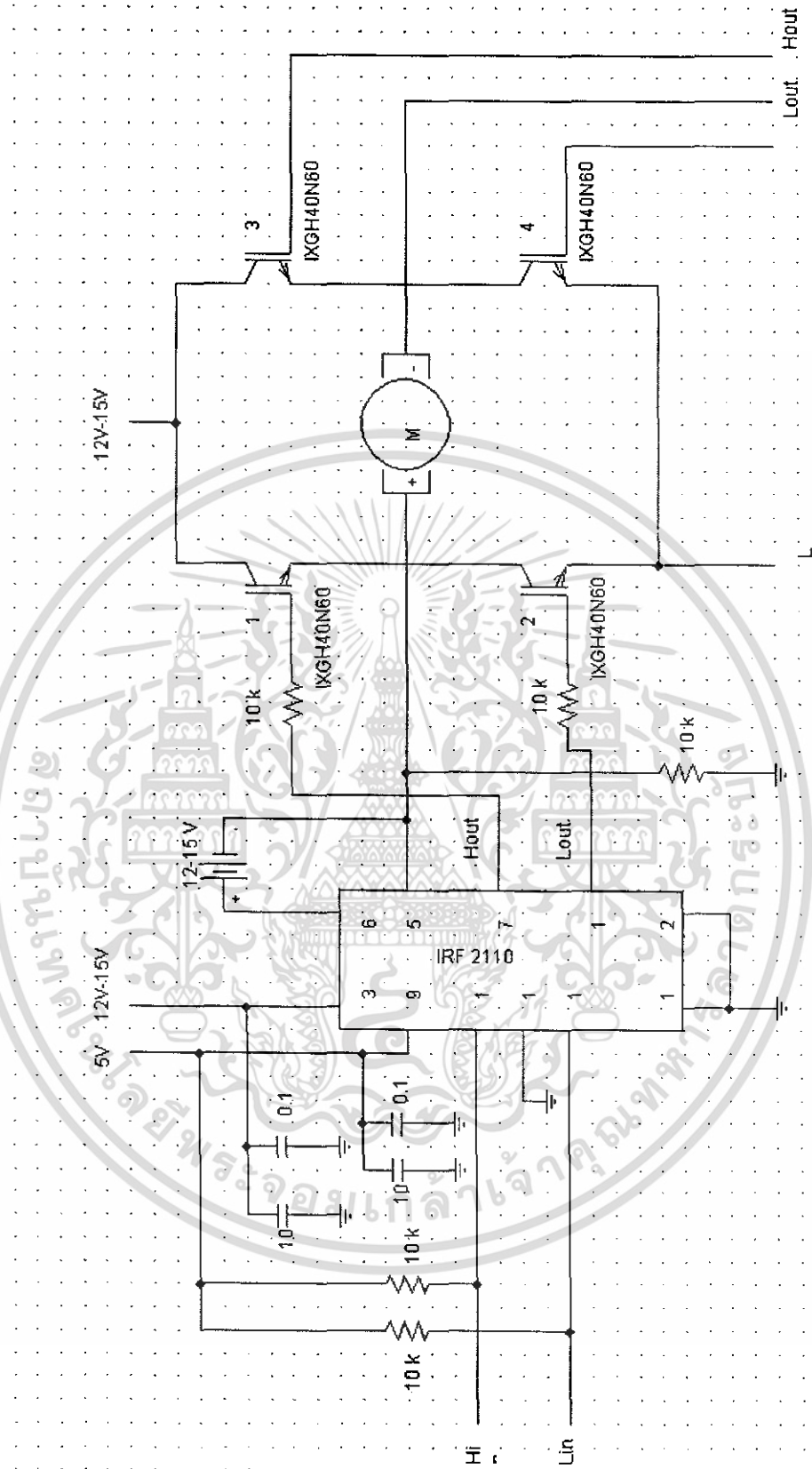
จากภาพ 3.11 ในการออกแบบวงจรขับมอเตอร์อาศัยหลักการขับมอเตอร์แบบ 4 ควอดแรน และคำนึงถึงกระแสกับแรงดันทางด้านเอาต์พุตซึ่งในที่นี้เราจำกัดกระแสไว้ไม่เกิน 15 แอมแปร์ และแรงดันอยู่ที่ -12 ถึง 12 โวลต์ สำหรับการทำงานของวงจรนี้จะรับสัญญาณอินพุต แบบพัลส์วิดมอดดูเลชันความถี่ประมาณ 2.5KHz ซึ่งเป็นสัญญาณที่ได้จากไมโครคอนโทรลเลอร์โดยค่าตัวดีไซเกิลจะขึ้นอยู่กับผลของการคำนวณทางคณิตศาสตร์ในตัวไมโครคอนโทรลเลอร์สัญญาณพัลส์วิดที่ออกมา 1 คู่ จะประกอบไปด้วยสัญญาณแบบ High และ Low ซึ่งสัญญาณทั้งสองนั้นจะเป็นแบบคอมพลิเมนต์ารีมี deattime ประมาณ 200ns

สัญญาณทั้งสองจะถูกส่งเข้าไปที่ตัว Gatedrive เพื่อขยายสัญญาณทางด้านเอาต์พุต จะได้ Hout และ Lout เพื่อไปสั่งการทำงานของตัวสวิทช์ซึ่ง ซึ่งในโปรเจกต์นี้ใช้มอสเฟต เนื่องจากความสามารถในการทนต่อกระแส แรงดัน และสามารถสวิทช์ที่ความถี่สูงๆได้

มอเตอร์ 1 ตัวใช้สัญญาณพัลส์วิด 2 คู่ เพื่อให้สามารถเลือกทิศทางของกระแสได้ แรงดันที่ป้อนให้มอเตอร์นั้นจะอยู่ที่ -12 ถึง 12 โวลต์ ซึ่งตัวที่จะกำหนดขนาดของแรงดันที่จะป้อนให้มอเตอร์ก็คือ สัญญาณพัลส์วิดที่ปรับค่าดีไซเกิลได้ตั้งแต่ 0-100% โดยที่

$$V_a = \text{Ton}/(\text{Toff} + \text{Ton}) * V_{\text{battery}} = \text{duty cycle} * V_{\text{battery}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

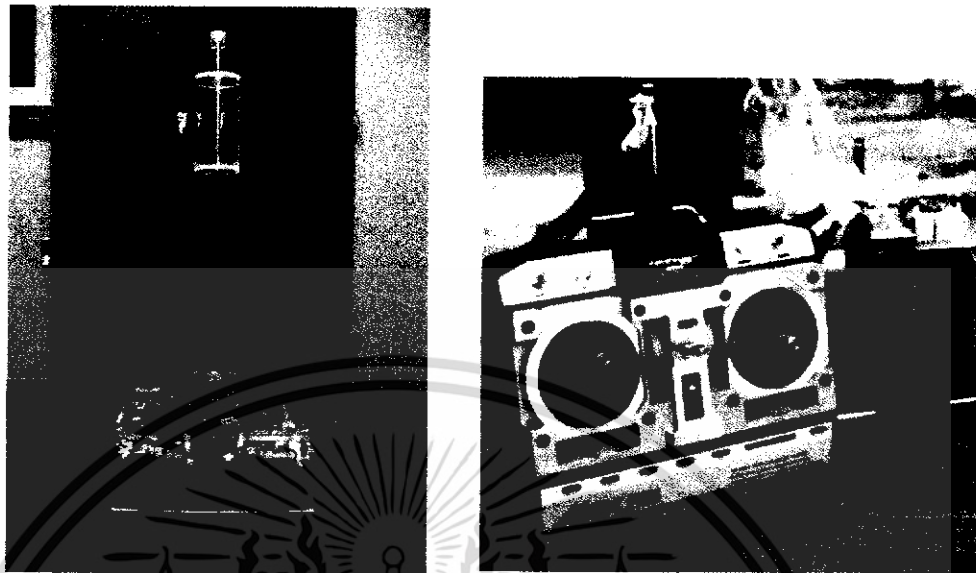


ภาพที่ 3.11 วงจรขับกระแสตรงของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การออกแบบตัวหุ่นยนต์

3.5.1 ลักษณะทั่วไป



ภาพที่ 3-12 ลักษณะทั่วไปของหุ่นยนต์ขับเคลื่อนแบบสองล้อ

เนื่องจากหุ่นยนต์ตัวก่อนมีขนาดเล็กและใช้มอเตอร์ในการควบคุมตำแหน่งของก้านเพนดูลัมแค่หนึ่งตัว จึงได้มีการพัฒนาต่อไปจากเดิม โดยการเพิ่มมอเตอร์เข้าไปอีกหนึ่งตัวในการควบคุมก้านเพนดูลัม ดังนั้นขนาดของตัวหุ่นยนต์จึงต้องมีขนาดใหญ่ขึ้นเนื่องจากการเพิ่มของน้ำหนักที่เกิดจากตัวมอเตอร์ที่มีขนาดใหญ่ขึ้นและแบตเตอรี่ของมอเตอร์ โดยก้านเพนดูลัมทำจากอลูมิเนียมเนื่องจากก้านเพนดูลัมควรมีน้ำหนักมากกว่าโครงสร้างด้านล่างโดยมีอะครีลิก ที่เปรียบเสมือนก้อนน้ำหนักติดอยู่ตรงปลายของก้านเพนดูลัม

การควบคุมทิศทางจะควบคุมผ่านรีโมทคอนโทรลเลอร์จากรูป สัญญาณจากรีโมทจะถูกรับด้วยภาครับสัญญาณแบบ PCM ซึ่งจะให้สัญญาณพัลส์วิดที่ด้กล่าวมาแล้วในส่วนของโปรแกรม การควบคุมทิศทางของตัวหุ่นยนต์นี้เหมือนกับการควบคุมทิศทางของรถบังคับทั่วไป คือเดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา แต่จะแตกต่างตรงที่ผู้บังคับต้องคอยบังคับรีโมทคอนโทรลเพื่อชดเชยสัญญาณที่ผิดพลาดจากตัวเซนเซอร์ ใจ โรส โคป

3.5.2 ลักษณะการติดตั้งตัวเซนเซอร์

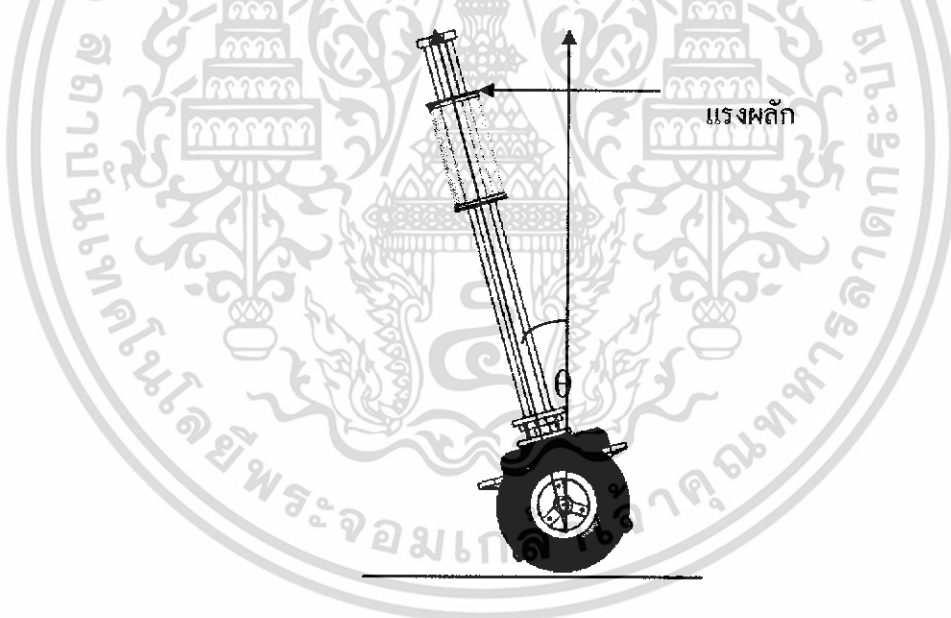
เนื่องจากการรักษาสมดุลของหุ่นยนต์ จะต้องใช้เซนเซอร์วัดมุมที่เอียงไปจากตำแหน่งสมดุลของก้านเพนดูลัมการติดตั้งจึงติดตั้งไว้บนตัวของหุ่นยนต์ เนื่องจากส่วนของแกนหมุนของล้อไม่สามารถ สามารถติดตั้งได้เพราะว่าปัญหาของพื้นที่ที่มีจำกัด แต่จะทำการติดตั้งตรงกลางของ

ตัวหุ่นยนต์เหนือแกนหมุนประมาณ 10 cm. เมื่อก้านเพนดูลัมเกิดมีสิ่งรบกวน ทำให้ก้านเพนดูลัมเปลี่ยนตำแหน่งไปจากจุดสมดุลเซนเซอร์จะวัดมุมที่ก้านเพนดูลัมเปลี่ยนแปลงไป



Gyroscope sensor

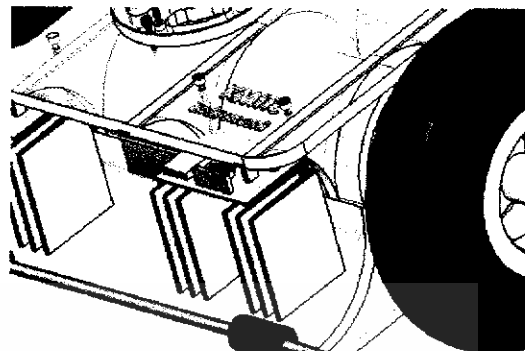
ภาพที่ 3-13 ลักษณะการติดตั้งตัวเซนเซอร์เมื่อก้านเพนดูลัมอยู่ที่จุดสมดุล



ภาพที่ 3-14 เมื่อก้านเพนดูลัมเกิดมีสิ่งรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 จุดสัมผัสพื้น



จุดสัมผัสพื้น

จุดสัมผัสพื้น

ภาพที่ 3-15 ลักษณะของจุดสัมผัสพื้น

จากการคำนวณมีข้อกำหนดว่าก้านเพนดูลัม สามารถยับได้ไม่เกินสิบห้าองศาจึงได้ทำการออกแบบจุดสัมผัสพื้นเพื่อป้องกันการเสียหายที่อาจจะเกิดจากการล้มของตัวหุ่นยนต์ โดยจุดสัมผัสพื้นจะแตะกับพื้น เมื่อก้านเพนดูลัมเอียงไปสิบห้าองศา จะช่วยทำให้ตัวหุ่นยนต์ ไม่กระทบกับพื้นอื่นเนื่องจากเมื่อก้านเพนดูลัม เอียงมากกว่าสิบห้าองศา จุดสัมผัสพื้นจะทำจากยาง ซึ่งมีความยืดหยุ่นได้พอสมควร ช่วยลดแรงกระแทกขณะที่หุ่นยนต์กำลังจะล้ม

3.5.4 ลักษณะการติดตั้งชิ้นส่วนอื่นๆ



ภาพที่ 3-16 ลักษณะการติดตั้งชิ้นส่วนอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบอื่นๆ จากที่กล่าวข้างต้นประกอบด้วย

1. ฐานของก้านเพนคลัม เพื่อให้ก้านเพนคลัมที่มีน้ำหนักไม่น้อยกว่า 1.2 kg ยึดติดกับตัวรถได้ ฐานจึงต้องมีความแข็งแรงพอสมควร ดังนั้นจึงได้ออกแบบฐานเพื่อยึดก้านเพนคลัมถึงสามชั้นดังภาพ

2. ส่วนของชุดไมโครคอนโทรลเลอร์ ชุดรับสัญญาณจากรีโมทคอนโทรลพร้อมแบตเตอรี่ขนาด 4 ก้อนของถ่าน AA และตัวเซนเซอร์แบบใจโรสโคป ซึ่งจะถูกลัดตั้งใกล้กับบริเวณทางขวาของหุ่นยนต์ ดังภาพ การติดตั้งแบบนี้จะลดปัญหาหาของการลדתอนของสัญญาณอันเนื่องมาจากระยะทางของสายไฟ

3. ส่วนของมอเตอร์และชุดเกียร์ แกนหมุนของมอเตอร์นั้นจะเปรียบเสมือนแกนหมุนของทั้งระบบของตัวหุ่นยนต์จะต่อกับส่วนของล้อ

4. ส่วนของที่ชาร์ตแบตเตอรี่ ประกอบด้วยที่ 2 ชุด คือ ที่ชาร์ตของแบตเตอรี่ที่ให้พลังงานกับส่วนของมอเตอร์และที่ชาร์ตแบตเตอรี่ที่ให้พลังงานกับส่วนที่เป็นวงจรไมโครคอนโทรลเลอร์และวงจรดีซีไดรโมเตอร์ ซึ่งจะเขียนบอกอยู่บนแผ่นป้ายที่แปะไว้

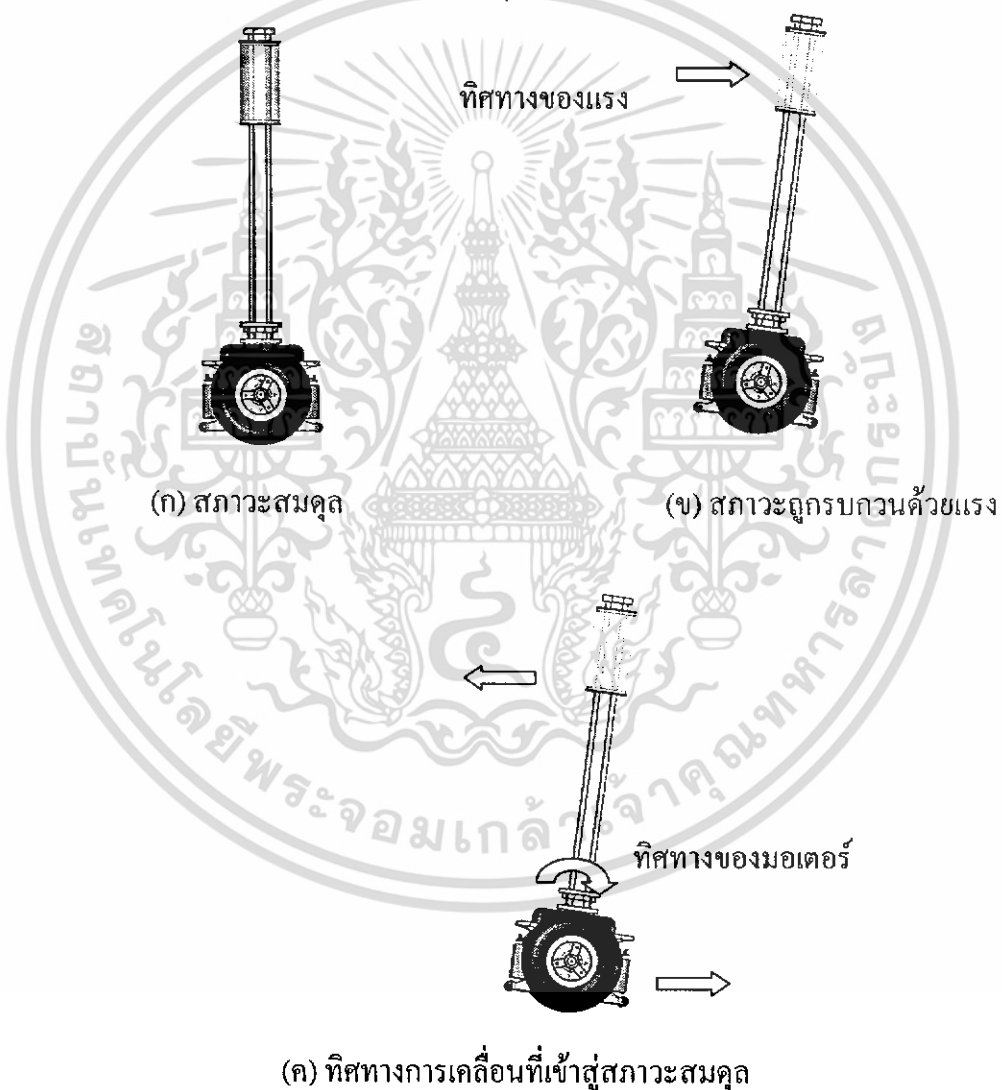
5. ส่วนของวงจรขับเคลื่อน หรือ วงจรดีซีไดรโมเตอร์ ซึ่งจะมีทั้งหมด 4 ชุด แต่ละชุดจะมีตัวระบายความร้อน(Heat zinc) เพื่อระบายความร้อนที่เกิดจากการที่ตัวสวิตซ์แบบมอสเฟต ซึ่งทำงานที่ความถี่สูงๆ เพื่อเพิ่มอายุการใช้งานให้กับตัวมอสเฟต ซึ่งส่วนนี้ทั้งหมดเป็นส่วนที่มีน้ำหนักพอสมควร การติดตั้งจะต้องคำนึงถึงสมดุลตัวตัวหุ่นยนต์ด้วย ดังนั้นจึงทำการติดตั้งด้วย โดยเน้นถึงความสมดุลเป็นหลัก

บทที่ 4

การทดลอง

4.1 การทดลองใส่แรงผลัก

จากภาพที่ 4.1 (ก) เมื่อระบบอยู่ในสภาวะสมดุล ก้านเพนดูลัมจะตั้งตรงดังภาพ แต่เมื่อใส่แรงผลักดันดังภาพ (ข) จะเปรียบเสมือนระบบถูกรบกวนด้วย disturbance สัญญาณจากใจโรสโคปซึ่งเป็นตัววัดความเอียง จะถูกส่งไปยังไมโครคอนโทรลเลอร์ ผ่านกระบวนการควบคุม และส่งสัญญาณออกมาให้มอเตอร์เคลื่อนที่เพื่อรักษาสมดุลให้กับระบบซึ่งมีทิศทางดังภาพ (ค)

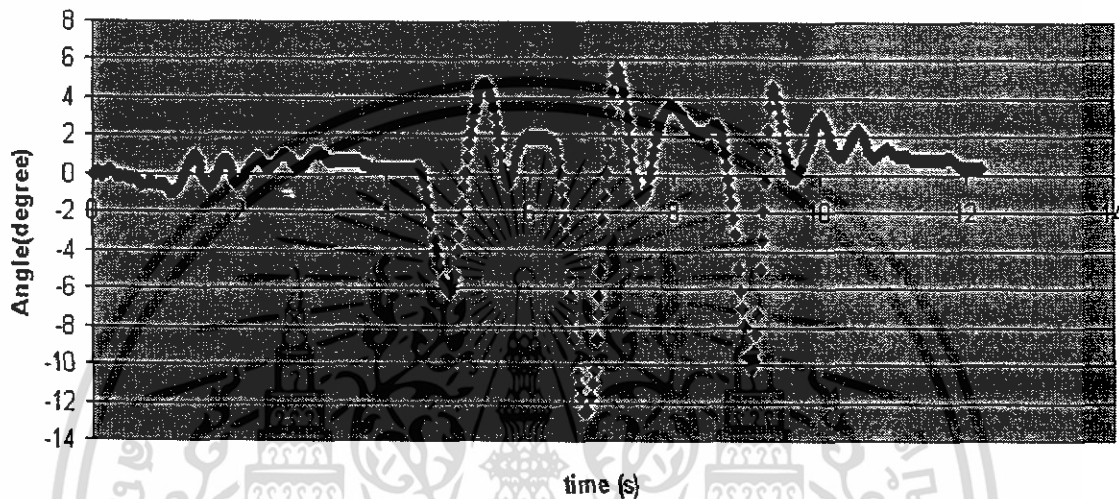


ภาพที่ 4.1 แสดงทิศทางของระบบเมื่อถูกรบกวน และพยายามปรับแก้

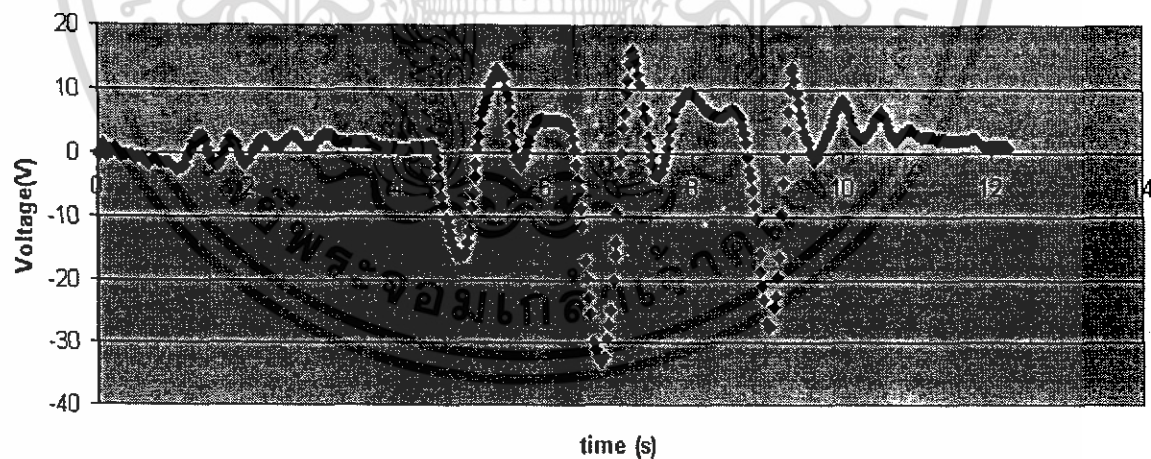
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองหาผลตอบสนองของหุ่นยนต์

การทดลองหาผลตอบสนองของระบบ เมื่อระบบเกิดสิ่งรบกวนขึ้น (หุ่นยนต์เอียงจากสภาวะสมดุล) ระบบจะทำการควบคุมการทำงานของมอเตอร์โดยป้อนค่าแรงดัน เพื่อให้ระบบกลับมาสมดุลอีกครั้ง การอ่านค่าของผลตอบสนองจะทำได้โดยการอ่านค่าสัญญาณของแรงดันที่ จะต้องป้อนให้มอเตอร์ที่มุมเอียงต่าง จากไมโครคอนโทรลเลอร์ผ่านสาย RS 232 เข้าสู่อินเตอร์คอมพิวเตอร์ บันทึกค่า และนำมาพล็อตกราฟซึ่งจะได้ผลการทดลองดังนี้



ภาพที่ 4.2 ภาพแสดงความสัมพันธ์ระหว่างมุมที่เปลี่ยนแปลงต่อเวลา



ภาพที่ 4.3 ภาพแสดงความสัมพันธ์ระหว่างแรงดันที่ป้อนให้กับมอเตอร์ที่เปลี่ยนแปลงต่อเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพแสดงให้เห็นถึงความสัมพันธ์ระหว่างมุมที่เปลี่ยนไปจากระบบที่สมดุล กับค่าของแรงดันไฟฟ้า ที่ส่งให้กับมอเตอร์เพื่อควบคุมให้ระบบกลับสู่สมดุลอีกครั้ง ณ เวลาเดียวกัน เมื่อระบบถูกรบกวน หรือ ถูกทำให้เอียงไปจากสถานะสมดุลมากขึ้นจะทำให้แรงดันที่ต้องป้อนให้กับมอเตอร์มากขึ้นด้วย ซึ่งหุ่นยนต์นี้สามารถเอียงได้สิบห้าองศา

จากภาพจะสังเกตเห็นว่าที่เวลาเริ่มต้นระบบจะสมดุลอยู่ที่มุม 0 องศาแต่เมื่อใส่สิ่งรบกวนเข้าไประบบจะเอียงไม่อยู่ที่ 0 องศา ดังนั้นระบบจะพยายามกลับมารักษาสมดุลอีกครั้ง โดยกลับมาที่มุม 0 องศาในที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง ปัญหาและข้อเสนอแนะ

1. สรุปผลการทดลอง

จากการทดลองเมื่อระบบถูกรบกวนทำให้ก้านเพนดูลัมเปลี่ยนแปลงตำแหน่งไปจากจุดสมดุลตัวเซนเซอร์จะวัดการเอียงไปของก้านเพนดูลัมและวัดระยะทางที่ลื่นเคลื่อนที่ไปจากจุดเริ่มต้นโดยนำสัญญาณการเอียงของมุมของก้านเพนดูลัมและระยะทางที่ตัวหุ่นยนต์ที่เคลื่อนที่ไปจากจุดสมดุลมาคำนวณเพื่อหาค่าของสัญญาณควบคุมไปทำการควบคุมระบบผ่านทางมอเตอร์ โดยที่ก้านเพนดูลัมสามารถเอียงได้ไม่เกินสามสิบองศา ระบบยังสามารถรักษาสมดุลอยู่ได้และจะค่อยๆปรับค่าตำแหน่งของก้านเพนดูลัมเพื่อให้เข้าสู่สมดุลอีกครั้งหนึ่ง

2. ปัญหา

1. ขนาดของมุมของก้านเพนดูลัมที่เปลี่ยนแปลงไปจากจุดสมดุล จะกลับสู่สมดุลได้จะขึ้นอยู่กับกำลังของมอเตอร์ เนื่องจากมีข้อจำกัดของกำลังมอเตอร์มุมของก้านเพนดูลัม จึงสามารถเปลี่ยนแปลงได้แค่สิบห้าองศา

2. ปัญหาที่เกิดจากตัวเซนเซอร์เมื่อปล่อย Gyroscope เอาไว้ตัวมันเองจะเกิดสัญญาณผิดพลาดขึ้นเองเพิ่มขึ้นตามระยะเวลา และไม่สามารถรู้จุดอ้างอิงของตัวเองได้ และปัญหาที่สำคัญที่สุดกับเซนเซอร์ชนิดนี้ คือ ปัญหาเรื่องของการแกว่งของสัญญาณ ทำให้การจับสัญญาณมีความผิดพลาดและส่งผลทำให้การควบคุมผิดพลาดด้วย

3. ชุดเกียร์เกิดมีระยะคลอนทำให้ระบบการควบคุมไม่ดีเท่าที่ควร

4. โครงสร้างเป็นอะคริลิกบอบบางเกินไปเมื่อต้องรับโหลดหนักๆ

3. ข้อเสนอแนะ

1. เปลี่ยนขนาดของตัวมอเตอร์ให้มีกำลังมากขึ้น เพื่อให้ระบบสามารถเปลี่ยนแปลงได้มากขึ้น

2. ปัญหาที่เกิดจากตัว Gyroscope แก้ไขโดย การเพิ่มตัวเซนเซอร์เข้าไป เพื่อเป็นจุดอ้างอิงให้กับตัว Gyroscope หรือใช้ตัวเซนเซอร์ที่มีประสิทธิภาพมากกว่านี้เพื่อแก้ปัญหาของการเกิดความผิดพลาด (Error) ขึ้นของตัว Gyroscope ได้

3. เปลี่ยนชุดเกียร์ให้มีระยะคลอนน้อยที่สุด

4. เปลี่ยนโครงสร้างให้มีความแข็งแรงมากขึ้นหากต้องการรับ โหลดที่มีขนาดมาก

บรรณานุกรม

- [1] ผ.ศ.ดร. ชนิตชัย ศรีสุวรรณวัฒน์ “optimal control system” เอกสารประกอบการเรียนการสอน
- [2] Rich Chi Ooi “Balancing a Two-Wheeled Autonomous Robot” The University of Western Australia,thesis 2003
- [3] Felix Grasser, Aldo D'Arrigo, Silvio Colombi, Alfred Rufer “A Mobile, InvertePedulum” Swiss Federal Institute of Technology Lausanne.
- [4] นคร ภัคดีชาติ, ชัยวัฒน์ ลิ้มพรจิตรวิไล “dsPIC microcontroller: Basic experiment in C programming with MPLAB C30” Innovative Experiment Co.,Ltd.



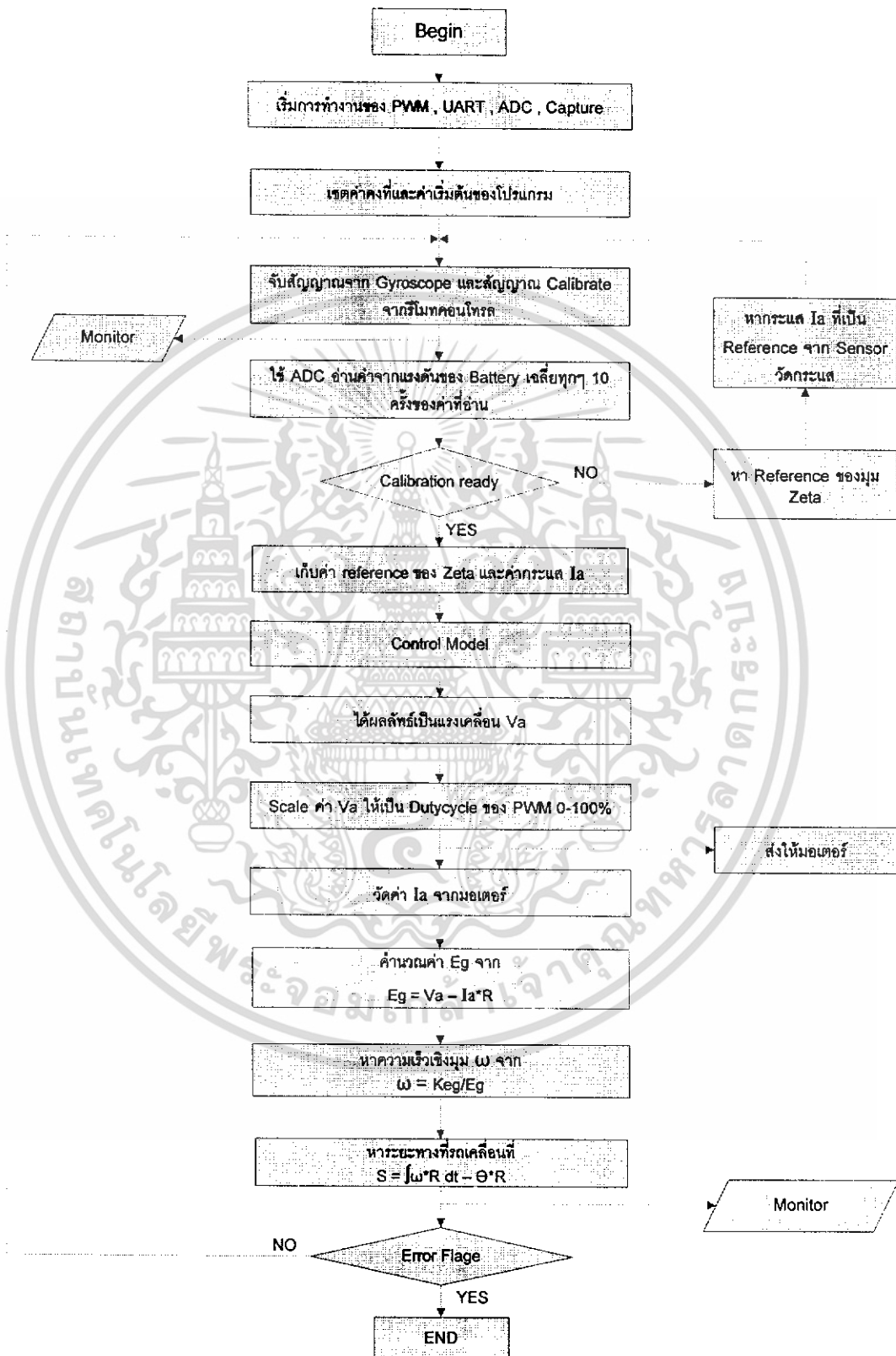
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมที่เขียนบนไมโครคอนโทรลเลอร์ dsPIC



แผนภาพขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/*

Xtal 30 MHz , Oscillator Mode Xt w/PLL 16x

Watchdog Timer Disable

Board Rate Uart1 38400

Capture use Timer23 32bit

: Pin input RD0 or IC1

: input capture 50 HZ 20 ms

: if input pwm = min Timer32bit read

: if input pwm = max

*/

#include<p30f6010.h>

#include<uart.h>

#include<pwm.h>

#include<timer.h>

#include<incap.h>

#include<stdio.h>

#include<print.c>

#include<PWM.c>

#include<Capture.c>

#include<ACD.c>

//include<ports.h>

#define CNTLQ 0x11

#define CNTLS 0x13

#define DEL 0x7F

#define BACKSPACE 0x08

#define CR 0x0D

#define LF 0x0A

int Flag_Capture ; // "0" is no flag , "1" is yes flag

unsigned long Capture_period,zecom,LeRi,Gyro_com ;

unsigned long reference,reference_com,reference_lr,reference_fb,reference_Ia1,reference_Ia2;

unsigned long max_zeta ,min_zeta,max_com,min_com,max_fb,min_fb;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned long max_lr,min_lr,max_Ia1,min_Ia1,max_Ia2,min_Ia2;
unsigned int count,count1,count2,count3,count4,count5,count6,count7,count8 ;
int Error_cap,compensate,Fw_Bw,Left_Right;
float sum;
unsigned long zeta_old,zeta_new;
char calibrate,mm,nn,oo,pp,qq,rr;
char str[12];
float zeta_com;
float total_zeta,zeta_rad;
float adc0,adc1,adc2,adc3;
float xd = 0; // มันคือ x dot
float zd = 0; // sum_zeta dot
float xdd;
float zdd;
float Vbatt;
int i1,i2,i3,i4;
float Ia1,Ia2;
float Ia1_2,Ia1_3,Ia2_2,Ia2_3;
unsigned long sumv;
unsigned long sum_Ia1,sum_Ia2;
float Va1;
float Va2;
float Error1,Error2,Error_Eg,In1,In2,In_Error;
float New_Error_cap,Old_Error_cap;
float New_Error_Eg,Old_Error_Eg,New_Error1,Old_Error1,New_Error2,Old_Error2;
float Eg1,Eg2;
float Biass,Ra1,Ra2,Rate;
float W1,W2;
float velo1,velo2,zeta_dot,old_zeta;
float distance1,distance1_old,distance2,distance2_old,distance_total;
float Va,K1,K2,K3,K4;
float GainE,GainP,GainI,GainI_Eg;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float GainE1,GainP1,GainI1;
float GainE2,GainP2,GainI2;
float Set_Point;
float Ra1_1,Ra1_2;
float Dir;
float F11,F12,F13,F14;
float A22,A42,A23,A43;
float B21,B41;
float L11,L21,L31,L41;
float L12,L22,L32,L42;
float X_X_d,X_dd,X,Zeta,Zeta_,Zeta_d,Zeta_dd;
float U,velo;
unsigned int y; //ใช้เป็นตัวตรวจสอบระบบ

//////////////////////////////////// Capture 1 //////////////////////////////////////
void __attribute__((interrupt)) IC1Interrupt(void) // แคปเจอร์ที่ขา RD0
{
    if(PORTDbits.RD8 == 1)
    {
        WriteTimer23(0);
    }
    if(PORTDbits.RD8 == 0)
    {
        //////////////////////////////////////Begining////////////////////////////////////
        //1.1...function compensate Zeta Or Forward,Bacdward (THRO).....//
        zecom = ReadTimer23();

        //1.2.....Function direct Left or Right (AILE).....//
        while(PORTDbits.RD9 == 0)
            WriteTimer23(0);
        while(PORTDbits.RD9 == 1)
            LeRi = ReadTimer23();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//1.3.....Gyroscope Signal (ELEV).....//
while(PORTDbits.RD10 == 0)
WriteTimer23(0);
while(PORTDbits.RD10 == 1)
Capture_period = ReadTimer23(); //อ่านค่า
Error_cap = reference - Capture_period;

//1.4.....Compensate Gyroscope (Rudder) .....//
while(PORTDbits.RD11 == 0)
WriteTimer23(0);
while(PORTDbits.RD11 == 1)
Gyro_com = ReadTimer23(); //อ่านค่า

//1.5.....Value of Ia1 & Ia2.....//
adc1 = analog(2); // ทา Ia1
adc2 = analog(3); // ทา Ia2

//1.5.....ฟังก์ชันหาเบนด์แก้ป.....//
max_zeta = reference+25;
min_zeta = reference-25;

max_fb = reference_fb + 100;
min_fb = reference_fb - 100;

max_lr = reference_lr + 100;
min_lr = reference_lr - 100;

max_com = reference_com + 100;
min_com = reference_com - 100;

max_Ia1 = reference_Ia1 + 5;
min_Ia1 = reference_Ia1 - 5;
max_Ia2 = reference_Ia2 + 5;
min_Ia2 = reference_Ia2 - 5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//1.6.....เรื่องของการบังคับ.....//
//1.6.1.....หาค่า Forward_Backward.....//
if((zecom > max_fb)||(zecom < min_fb))
{
    Fw_Bw = reference_fb - zecom;
}
if((zecom < max_fb)&(zecom > min_fb))
{
    Fw_Bw = 0;
}
//1.6.2.....หาค่า compensate of Gyroscope.....//
if((Gyro_com > max_com)||(Gyro_com < min_com))
{
    compensate = reference_com - Gyro_com;
}
if((Gyro_com < max_com)&(Gyro_com > min_com))
{
    compensate = 0;
}
//1.6.3.....ฟังก์ชันอินทิเกรตหา Zeta.....//
if((Capture_period>max_zeta)||(Capture_period<min_zeta))//กรณีที่
capture_period มีค่ามากกว่า rangeที่กำหนด
{
    sum = Error_cap + (compensate/50) + sum;
}
if((Capture_period<max_zeta)&(Capture_period>min_zeta))//กรณีที่
น้อยกว่า max แต่มากกว่า min
{
    Error_cap = 0;
    sum = Error_cap + (compensate/75) + sum;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//1.6.4.....หาค่า Left_Right.....//
if((LeRi>max_lr)||(LeRi<min_lr))
{
    Left_Right = reference_lr - LeRi;
}
if((LeRi<max_lr)&(LeRi>min_lr))
{
    Left_Right = 0;
}

//1.6.5.....การหาค่าของกระแส Ia1 & Ia2.....//
if((adc1>max_Ia1)||(adc1<min_Ia1))
{
    Ia1 = adc1 - reference_Ia1;
}
if((adc1<max_Ia1)&(adc1>min_Ia1))
{
    Ia1 = 0;
}
if((adc2>max_Ia2)||(adc2<min_Ia2))
{
    Ia2 = reference_Ia2 - adc2;
}
if((adc2<max_Ia2)&(adc2>min_Ia2))
{
    Ia2 = 0;
}
}

```

```

//1.7.....หาค่าเฉลี่ยของ Vbatt.....//
if(count6 != 10) //เริ่มจาก 0-9 ทั้งหมด 10 ครั้ง
{
    adc3 = analog(1); // ทา Vbatt
    count6++;
    sumv = adc3 + sumv;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(count6 == 10)
{
    Vbatt = (sumv / 2046.0)*2.44; // 0-12 Volts
    sumv = 0;
    count6 = 0;
}
//2.0////////////////////////////////ฟังก์ชันการเบรต Calibrate = 1////////////////////////////////
if(calibrate == 1)
{
    //2.1.1.....หาค่า reference Zeta.....//
    if(mm == 1)
    {
        if(sum != 0.0)
        {
            sum = 0.0;
            reference = reference - Error_cap;
            count = 0;
        }
        if(sum == 0.0)
        {
            count++;
            if(count== 100)
            {
                mm = 0;
            }
        }
    }
}
//2.1.2.....หาค่า reference Gyro_compensate.....//
if(nn==1)
{
    if(compensate != 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    reference_com = Gyro_com;
    count1 = 0;
}
if(compensate == 0)
{
    count1++;
    if(count1==100)
    {
        nn = 0;
    }
}
}
//2.1.3.....หาค่า reference fb.....//
if(oo==1)
{
    if(Fw_Bw != 0)
    {
        reference_fb = zecom;
        ount2 = 0;
    }
    if(Fw_Bw == 0)
    {
        count2++;
        if(count2==100)
        {
            oo = 0;
        }
    }
}
}
//2.1.4.....หาค่า reference lr.....//
if(pp==1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if(Left_Right!=0)
    {
        reference_lr = LeRi;
        count3 = 0;
    }

    if(Left_Right==0)
    {
        count3++;
        if(count3==100)
        {
            pp = 0;
        }
    }
    //2.1.5.....หาค่า reference Ia1.....//
    if(qq==1)
    {
        if(Ia1 != 0)
        {
            reference_Ia1 = adc1;
            count4 = 0;
        }
        if(Ia1==0)
        {
            count4++;
            if(count4==100)
            {
                qq = 0;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//2.1.6.....หาค่า reference Ia2.....//
if(rr==1)
{
    if(Ia2 != 0)
    {
        reference_Ia2 = adc2;
        count5 = 0;
    }
    if(Ia2==0)
    {
        count5++;
        if(count5==100)
        {
            rr = 0;
        }
    }
}
if((mm==0)&(nn == 0)&(oo == 0)&(pp == 0)&(qq == 0)&(rr == 0))
{
    calibrate = 0; // End of calibrate function
    PORTDbits.RD3 = 1; //แสดงสถานะเมื่อคาร์ิบเรตเสร็จ
}

PDC1 = 600.0 ; // (5KHz=0 - 514)(1KHz=0-2570)
PDC2 = 3149.0;
PDC3 = 3149.0;
PDC4 = 600.0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////Calibrate = 0 เริ่มดำเนินการหลัก////////////////////////////////////
if(calibrate==0)
{
//3.4.2.....หาค่า Eg.....

GainP =analog(8)/256.00;
GainI =analog(9)/10.24;*/
count8++ ;
Ra1=96.680;// analog(7)/5.12; //Ra1=96.680
Ra2=92.578;// analog(8)/5.12;//Ra2=92.578
Biass= 600.0;
GainI_Eg=20.00;//analog(9)/5.12;
Rate =0.3;// analog(7)/170.66666667;
Ia1_2=(Ia1)*(Ia1);
Ia1_3=Ia1_2*(Ia1);
K1= -(Ia1_3*0.00000424037087)+(Ia1_2*0.00194762087103)-
(Ia1*0.06698584017707);
K2=
(Ia1_3*0.00000424037087)+(Ia1_2*0.00194762087103)+(Ia1*0.06698584017707);

if( Ia1 > 250 )
{
K1 = 0.0;
}
if( Ia1 < -250)
{
K2 = 0.0;
}

Ia2_2=(Ia2)*(Ia2);
Ia2_3=Ia2_2*(Ia2);

K3= -(Ia2_3*0.00000424037087)+(Ia2_2*0.00194762087103)-
(Ia2*0.06698584017707);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

K4=
(Ia2_3*0.00000424037087)+(Ia2_2*0.00194762087103)+(Ia2*0.06698584017707);

if( Ia2 > 250 )
{
    K3 = 0.0;
}

if( Ia2 < -250)
{
    K4 = 0.0;
}

if(U > 0)
{
    Va1 = (((i1-Biass)/(3749.0-Biass))*Vbatt); //Va1 = Vbatt*Dutycycle1
    Eg1 = (((((i1-Biass)*Vbatt*(Ra1+K1))-((Ia1)*(3749.0-Biass)))/((3749.0-
Biass)*(Ra1+K1))))*0.6457542431;
}
if(U <= 0)
{
    Va1 = (((i2+Biass-3749.0)/(3749.0-Biass))*Vbatt);
    Eg1 = (((((i2+Biass-3749.0)*Vbatt*(Ra1+K2)) - (Ia1)*(3749.0-
Biass))/((3749.0-Biass)*(Ra1+K2))))*1.173752934;
}
if(U > 0)
{
    Va2 = (((i3-Biass)/(3749.0-Biass))*Vbatt); //Va1 = Vbatt*Dutycycle1
    Eg2 = (((((i3-Biass)*Vbatt*(Ra2+K3))-((Ia2)*(3749.0-Biass)))/((3749.0-
Biass)*(Ra2+K3))));
}
if(U <= 0)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Va2 = (((i4+Biass-3749.0)/(3749.0-Biass))*Vbatt);
Eg2 = (((i4+Biass-3749.0)*Vbatt*(Ra2+K4)) - (Ia2)*(3749.0-
Biass))/((3749.0-Biass)*(Ra2+K4));
}

```

```
//3.4.3.....ทวิระยะทาง.....
```

```

W1 = Eg1 / (0.095768*3.0); // Keg1 = 0.0016284 Volt.s/rad
W2 = Eg2 / (0.095768*3.0); // Keg2 = 0.0016284
velo1 = W1*0.245*0.02180 ; //W1*รัศมีล้อ*เวลาใน 1 scantime
velo2 = W2*0.245*0.02180 ; // รัศมีล้อเท่ากับ 0.245
velo=(velo1+velo2)/2.0;
distance1 = distance1+(velo1 + distance1_old)/2.0; //lamp hold signal;
distance1_old = velo1;
distance2 =distance2+ (velo2 + distance2_old)/2.0;
distance2_old = velo2;
distance_total = (distance1+distance2)/2.0;
total_zeta = ((sum/795.0)+(Fw_Bw/2000.00)); // 0-30 degree
zeta_rad = total_zeta*0.0174533 ;// หน่วยเป็น rad 0-0.5236 rad
Zeta=(zeta_rad*(-1.00));
zeta_dot = (Zeta - old_zeta)/0.0218;
old_zeta = Zeta;
X=distance_total;

```

```
//3.2.....การคำนวณคอนโทรลโมเดล.....//
```

```

F11=0.0010;
F12=2.3925;//2.3693;
F13=148.9684;//78.4753;
F14=3.4686;//2.5066;
A22=-0.8682;
A42=-0.9467;
A23=22.5148;
A43=45.9485;
B21=0.7403;
B41=0.8073;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

L11=-5.2029;
L21=-15.9954;
L31=-2.2183;
L41=-36.9359;
L12=-2.2183;
L22=-29.8709;
L32=-24.9131;
L42=-62.7920;

```

```

X_d = L11*(X_-X)+X_d+L12*(Zeta_-Zeta);
X_dd = L21*(X_-X)+A22*X_d+A23*Zeta_+L22*(Zeta_-Zeta)+B21*U;
Zeta_d = L31*(X_-X)+Zeta_d+L32*(Zeta_-Zeta);
Zeta_dd = L41*(X_-X)+A42*X_d+A43*Zeta_+L42*(Zeta_-Zeta)+B41*U;
X_ =X_+(X_d*0.02180);
X_d=X_d+(X_dd*0.02180);
Zeta_ =Zeta_+(Zeta_d*0.02180);
Zeta_d=Zeta_d+(Zeta_dd*0.02180);
U=F11*X+F12*velo+F13*Zeta+F14*zeta_dot;
adc0 =0.0;// analog(4);
GainE1 = 1.906;// 0.938;
GainP1 = 0.668 ;//0.574;
GainI1 = 0.00;//38.086;// 93.164;
GainE2 = 1.906;// 0.700;
GainP2 = 0.668;// 1.033;
GainI2 = 0.00;//38.086;// 2.930;
Dir = Left_Right/(1166.666667*5.0);
/* Set_Point= compensate/(1166.666667*5.0);*/
Error_Eg = ((Eg1)-(Eg2)+Dir);//Error ของความเร็ว
In_Error = In_Error+(Error_Eg*0.02180*0.5);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Error1= ((Set_Point)-(Eg1*GainE1)-(In_Error*GainI_Eg));// Error2=SP2-PV2+Error_Eg
In1=In1+(Error1*0.02180*Rate);
i1 = (((Error1)*GainP1+In1*GainI1)*312.4167)+Biass ;// Close Loop PI controller
if(i1>=3740.0)
{
    i1 = 3740.0;
}
if(i1<=10)
{
    i1 = 10;
}
i2 = 3749.0 + (((Error1)*GainP1+In1*GainI1)*312.4167)-Biass ;
if(i2>=3740.0)
{
    i2 = 3740.0;
}
if(i2<=10)
{
    i2 = 10;
}
Error2= ((U)-(Eg2*GainE2)+(In_Error*GainI_Eg));// Error2=SP2-PV2+Error_Eg
In2=In2+(Error2*0.02180*Rate);
i3 = (((Error2)*GainP2+In2*GainI2)*312.4167)+Biass ;// Close Loop PI controller
if(i3>=3740.0)
{
    i3 = 3740.0;
}
if(i3<=10)
{
    i3 = 10;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i4 = 3749.0 + (((Error2)*GainP2+In2*GainI2)*312.4167)-Biass ;
if(i4>=3740.0)
{
    i4 = 3740.0;
}
if(i4<=10)
{
    i4 = 10;
}
PDC1 = i1; // (5KHz=0 - 514)(1KHz=0-2570)
PDC2 = i2;
PDC3 = i3;
PDC4 = i4;
//3.3.....การหาค่า PWM .....//
/*
    if(Va > Vbatt)
    {
        i1 = 2570;
        i2 = 0;
        i3 = 2570;
        i4 = 0;
    }
    if(Va < (-1.0*Vbatt))
    {
        i1 = 0;
        i2 = 2570;
        i3 = 0;
        i4 = 2570;
    }
    if((Va < Vbatt)&(Va > (-1.0*Vbatt)))
    {
        i1 = 1285.0+(Va/Vbatt*1285.0) ;
        i2 = 1285.0-(Va/Vbatt*1285.0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i3 = i1;
        i4 = i2;
    }
    PDC1 = i2 ; // (5KHz=0 - 514)(1KHz=0-2570)
    PDC2 = i1 ;
    PDC3 = i2 ;*/
}
}
Flag_Capture = 1;
IFS0bits.IC1IF = 0; // เคลียร์อินเตอร์พแคปเจอร์
}
////////////////////////////////// PWM ////////////////////////////////////
void __attribute__((__interrupt__)) _PWMInterrupt(void)
{
    IFS2bits.PWMIF = 0;
}
////////////////////////////////// TX1 RX1 ////////////////////////////////////
void __attribute__((__interrupt__)) _U1TXInterrupt(void)
{
    IFS0bits.U1TXIF=0;
}
void __attribute__((__interrupt__)) _U1RXInterrupt(void)
{
    IFS0bits.U1RXIF=0;
}
////////////////////////////////// Main Program ////////////////////////////////////
int main(void)
{
    char Txdata []="r\nUART test Gyroscope sensor\r\n";
    char i;
    float a,b;
    calibrate = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

reference = 30000;
reference_com = 30000;
reference_lr = 30000;
reference_fb = 30000;

mm=1;
nn=1;
oo=1;
pp=1;
qq=1;
rr=1;

count = 0;
count1 = 0;
count2 = 0;
count3 = 0;
count4 = 0;
count5 = 0;
count6 = 0;
count8 = 0;
total_zeta = 0;
distance1 = 0;
distance2 = 0;
distance_total = 0;
sumv = 0;
xd = 0;
zd = 0;
Va = 0;

zeta_old = 0;
Eg1=0;
Eg2=0;
In1 = 0;
In2 = 0;

In_Error = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Zeta=0;
old_zeta = 0;
Zeta_=0;
Zeta_d=0;
Zeta_dd=0;
X=0;
X_=0;
X_d=0;
X_dd=0;
U=0;
i = 0;
Old_Error_cap =0.0;
Flag_Capture = 0;
uart1_init();
PWM_init();
capture_init();
putsUART1((unsigned int *)Txdata );
printf("\r");
Open_capture();
TRISDbits.TRISD3 = 0;
TRISDbits.TRISD2 = 1;
TRISDbits.TRISD1 = 1;
while(1)
{
    while (Flag_Capture==0); // รอให้มีการ เคปเจอร์ก่อน
    Flag_Capture = 0;
    i++;
    if(i == 2)
    {
        a = count8*0.0218;
        b = total_zeta*(-1.0);
        printf("\r%.4f\t%.4f\t%.3f\n",U,b,a);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i = 0;
    }

}

}

```

ส่วนต่อไปนี้จะ เป็นโปรแกรมย่อยซึ่งเป็นตัวกำหนดครีจิสเตอร์เพื่อกำหนดการทำงาน ทางด้านฮาร์ดแวร์ของตัวไมโครคอนโทรลเลอร์ ซึ่งประกอบไปด้วยส่วนของ PWM, Capture , Analog to digital และ Uart

1. PWM

เป็นส่วนของการกำหนดค่าต่างๆ ของ pulsewidth modulation เช่น การกำหนดความถี่ของ สัญญาณที่ออก การกำหนดโหมคของ PWM เป็น โหมคอิสระ หรือ แบบคอมพลิเมนต์ การ กำหนดค่าของเวลาตาย(dead time) เป็นต้น ซึ่งมีโปรแกรมดังนี้

```

#define FCY 30000000 // 30 MIPS
#define FPWM 200000 // 10 kHz ,if 20 kHz = 40000
#define DEADTIME (unsigned int)(0.000022 * FCY) //เดิมเป็น 0.000002
//#define _DES_FREQ 60 // 60 Hz sine wave is required
#define _DELTA_PHASE (unsigned int)(_DES_FREQ * 65536 / FPWM)
char new;
void PWM_init(void)
{
    TRISE = 0x0100; // PWM pins as outputs, and FLTA as input
    PTPER = 0x0752; // XT x16 5 KHz (ก่อนคือ 6675 = 50Hz)(0100 = 5KHz)()
    OVDCON = 0x0000; // Disable all PWM outputs.
    DTCON1 = 0x3F3F; // (PLLx8,0004 = 202 ns)(PLLx8,0006 = 300ns)(PLLx8,0008 =
    400)(0404)
    DTCON2 = 0x0000;
    PWMCON1 = 0x00FF; // Enable PWM output pins and enable complementary mode 0077
    PDC1 = PTPER; /* 0 Volts on Phase A. This value corresponds to*/
    PDC2 = PTPER; // 0 Volts on Phase B.
    PDC3 = PTPER; // 0 Volts on Phase C.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PDC4 = PTPER; // 0 Volts on Phase D.
IFS2bits.PWMIF = 0; // Clear PWM Interrupt flag
IEC2bits.PWMIE = 1; // Enable PWM Interrupts
OVDCON = 0xFF00; // PWM outputs are controller by PWM module
PTCONbits.PTCKPS = 1; // PWM Time Base Input Clock Prescale Select bits 16
PTCONbits.PTMOD = 0; // Center aligned PWM operation
PTCONbits.PTEN = 1; // Start PWM
return;
}

```

2. ส่วนของ ADC หรือ Analog to Digital

ส่วนนี้จะมีการกำหนดที่สำคัญอยู่ คือ การเซตขาของไมโครคอนโทรลเลอร์ให้เป็นขาที่รับสัญญาณทางอนาล็อก เรื่องของการเริ่มการ conversion และการหยุด conversion ความเร็วในการ sample ของสัญญาณ ซึ่งมีโปรแกรมดังนี้

```

#include<p30f6011.h> // Header file for dsPIC30F6011
#include<lcd6011.h> // Header file for dsPIC30F6011
#include<adc10.h> // Module function for uart
#include<stdio.h>
unsigned int x,a;
unsigned int analog(unsigned char ch)
{
    unsigned int val1;
    ADPCFG &= (!(ch+1)); // all PORTB = Digital; RB0 = analog
    ADCON1 = 0x0000; // SAMP bit = 0 ends sampling ...
    ADCHS = ch; // Connect RB0/AN0 as CH0 input ..
    ADCSSL = 0;
    ADCON3 = 0x000A; // Manual Sample, Tad = internal 2 Tcy
    ADCON2 = 0;
    ADCON1bits.ADON = 1; // turn ADC ON
    ADCON1bits.SAMP = 1; // start sampling ...
    // delay_ms(1);
    // Keep for counter loop
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(a=0;a<102;a++); // Loop for delay 1 millisc per unit(7.3728 MHz@ PLL4x) 816
ADCON1bits.SAMP = 0; // start Converting
while (!ADCON1bits.DONE); // conversion done?
val1 = ADCBUF0; // yes then get ADC value
return(val1);
}

```

3. ส่วนของ Capture หรือ การจับสัญญาณ

เป็นส่วนใช้ในการจับสัญญาณ PWM ของใจโรสโคป ซึ่งต้องกำหนดการทำงานของขาอินพุต ซึ่งในที่นี้ใช้ขา RD0 และกำหนดลำดับความสำคัญของการอินเตอร์รัพ

```

void capture_init()
{
    /* Enable Timer1 Interrupt and Priority to '1' */
    ConfigIntCapture1(IC_INT_PRIOR_1 & IC_INT_ON);
    T2CON = 0x8008;
    // timer 32 bit start
    // timer mode 32 bit
}

void Open_capture()
{
    OpenCapture1(IC_IDLE_STOP & IC_TIMER3_SRC &
        IC_INT_1CAPTURE & IC_EVERY_EDGE);
}

```

4. ส่วนของการแสดงผลผ่านทางช่องสัญญาณอนุกรม หรือ Uart

ในส่วนนี้เป็นการกำหนดค่าเริ่มต้นการใช้งานของ Uart ซึ่งจะเขียนโปรแกรมได้ดังนี้

```

void uart1_init()
{
    int U1ModeValue;
    int U1StatusValue;
    unsigned BaudRateValue = 65; // 38400 when PLL=16x, XTAL1=30MHz
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

U1ModeValue =
    UART_EN                &
    UART_IDLE_CON          &
    UART_TX_ENABLE         & // test
    UART_EN_WAKE           &
    UART_DIS_LOOPBACK     &
    UART_DIS_ABAUD        &
    UART_1STOPBIT         & //test
    UART_NO_PAR_8BIT;

```

```

U1StatusValue =
    UART_INT_TX_BUF_EMPTY &
    UART_TX_PIN_NORMAL   &
    UART_TX_ENABLE        &
    UART_INT_RX_CHAR      &
    UART_ADR_DETECT_DIS  &
    UART_RX_OVERRUN_CLEAR;
OpenUART1( U1ModeValue, U1StatusValue, BaudRateValue );
SetPriorityIntU1RX(3);
EnableIntU1RX;
}

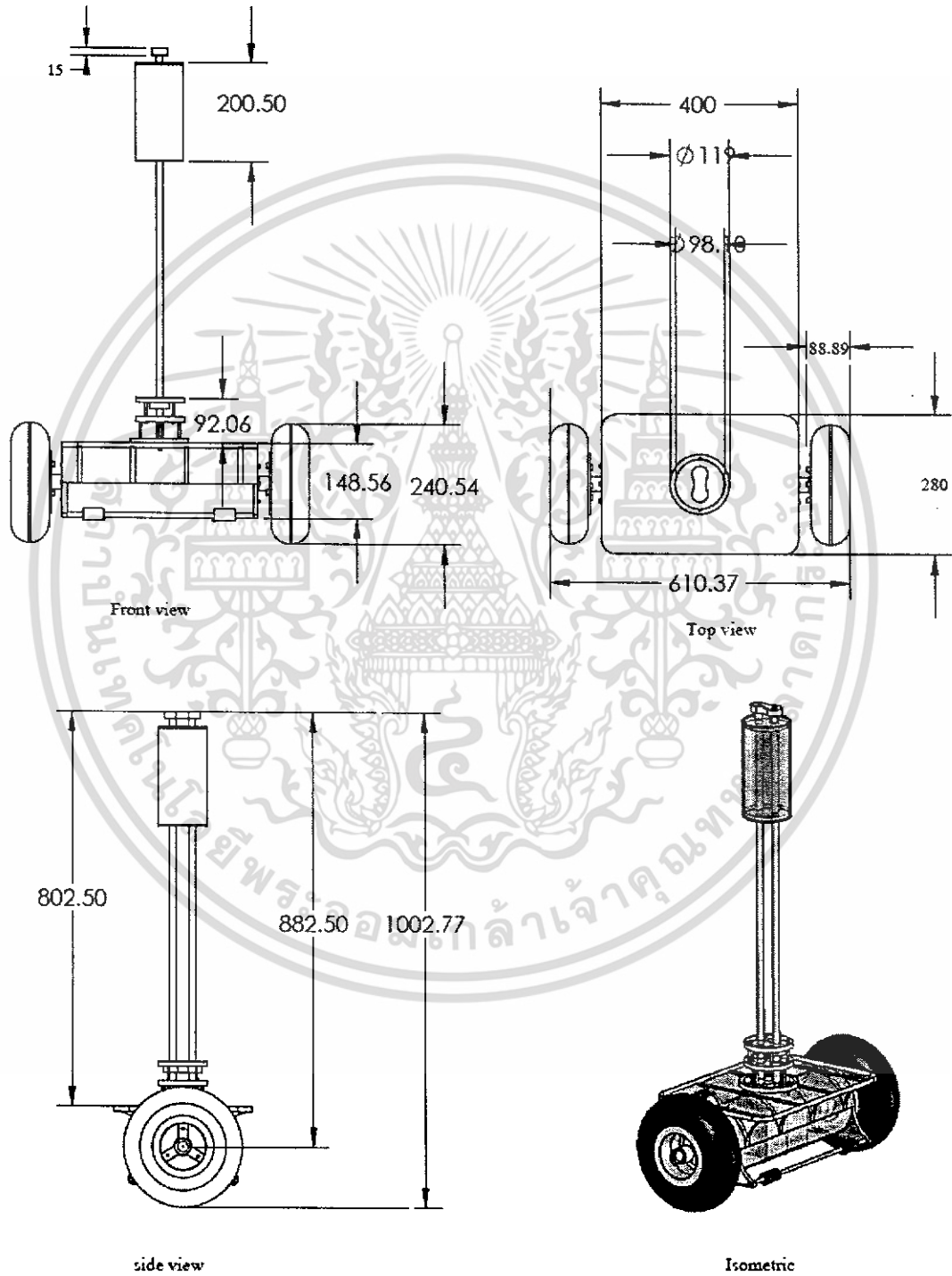
```

ผู้สนใจสามารถนำโปรแกรมนี้มาใช้ได้เลย ซึ่งโปรแกรมน้อยทั้ง 4 นั้น สามารถนำมารวมกับโปรแกรมหลักได้เลย หรือ แยกกันก็ได้ โดยถ้าแยกกันจะต้องกำหนด directory ของโปรแกรมน้อย ใน Include Path ซึ่งอยู่ในส่วนของ Build option

ภาคผนวก ข

โครงสร้างของตัวหุ่นยนต์ขับเคลื่อนแบบสองล้อ

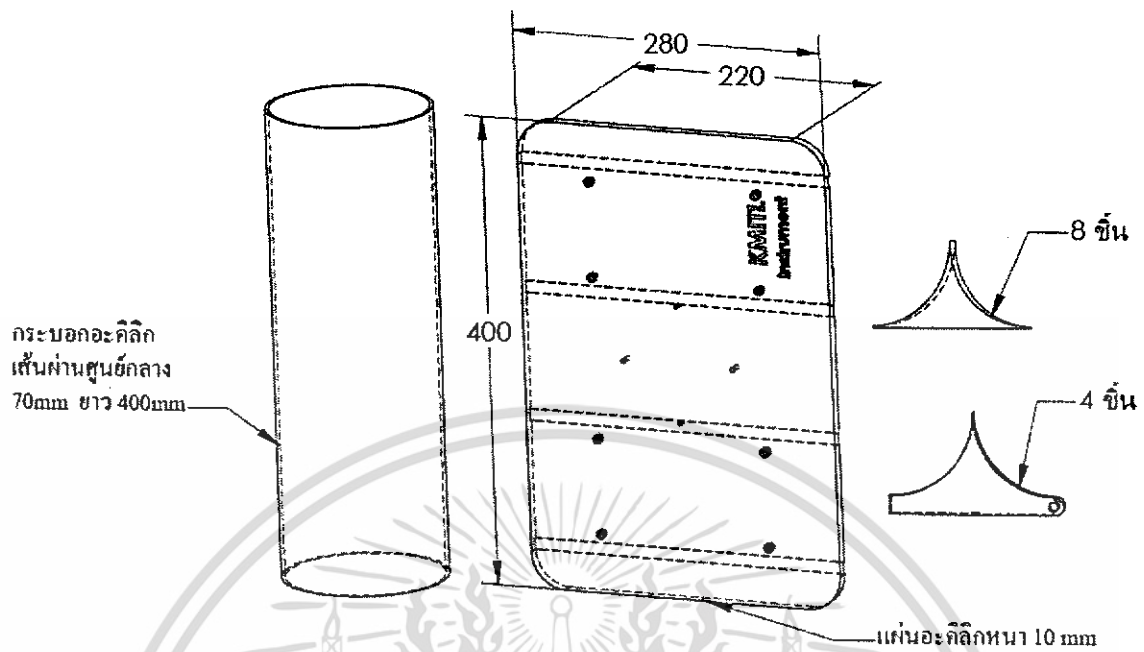
1. Dimension ของตัวรถ



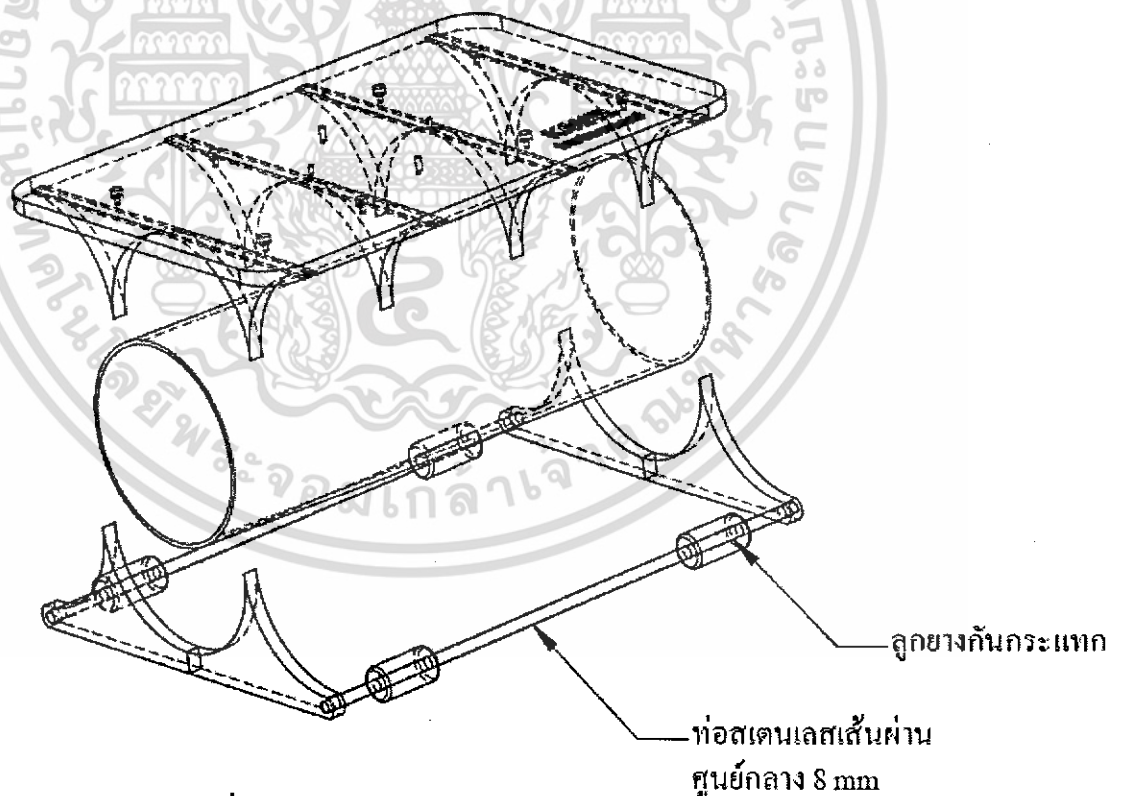
Dimension ของหุ่นยนต์ สักส่วนต่อขนาดจริง 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ชุดประกอบของตัวรถ



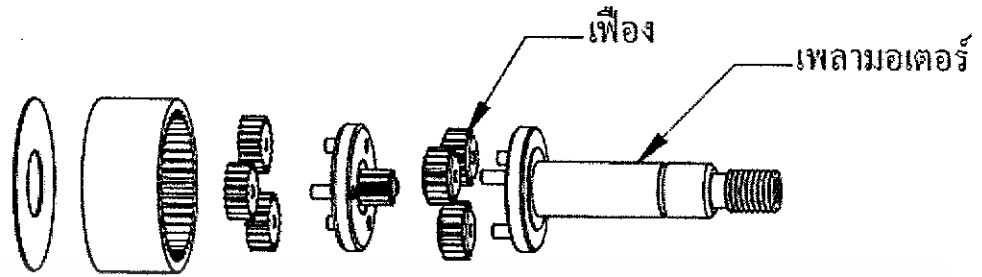
รูปที่ 2.1 ส่วนประกอบหลักของตัวรถ



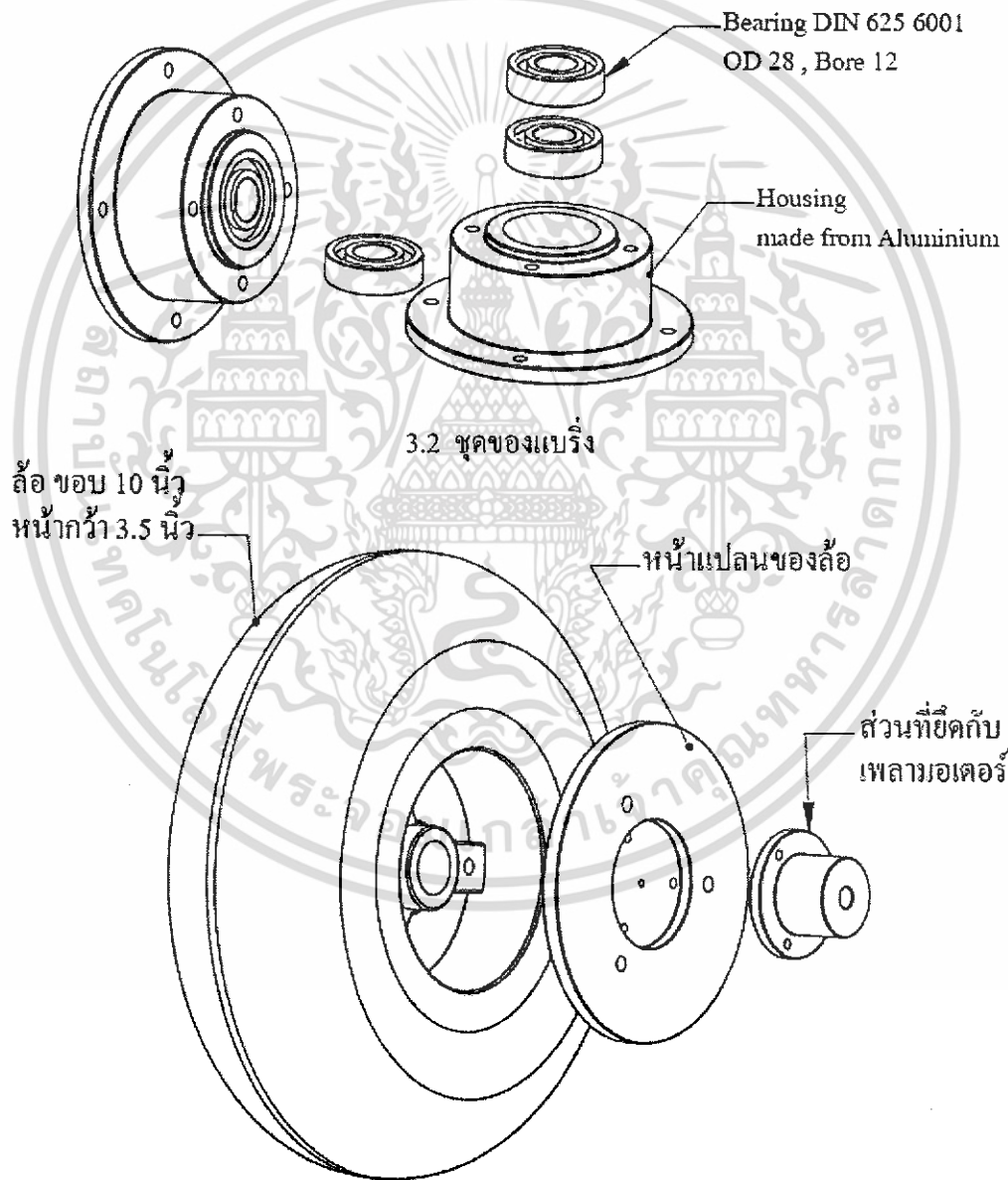
รูปที่ 2.2 แสดงการประกอบส่วนของตัวรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ชุดของระบบเกียร์และล้อ



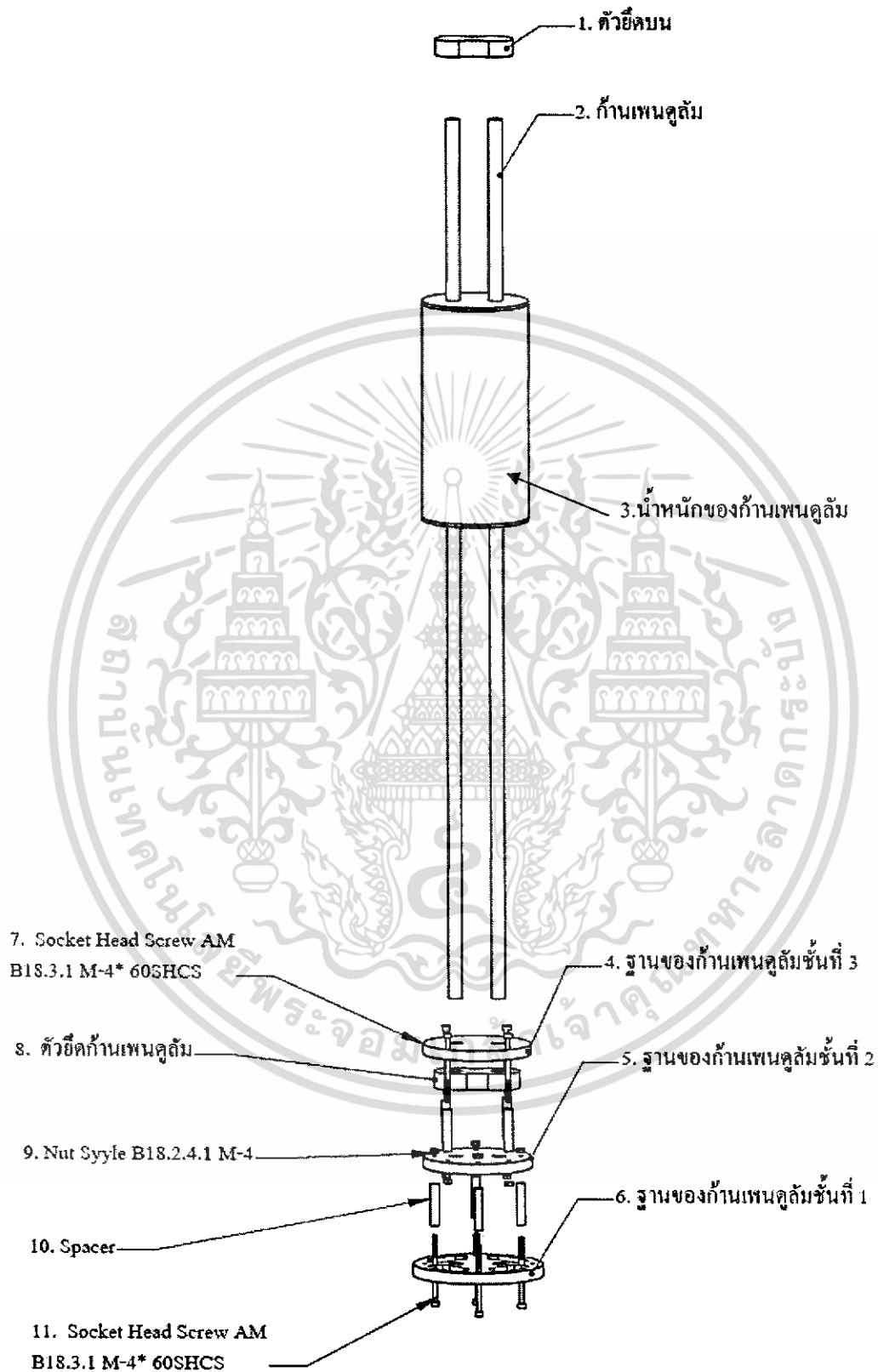
3.1 ชุดเพนเนททารีเกียร์(Planetary Gear) อัตราทด 9:44



3.3 ชุดประกอบของล้อ

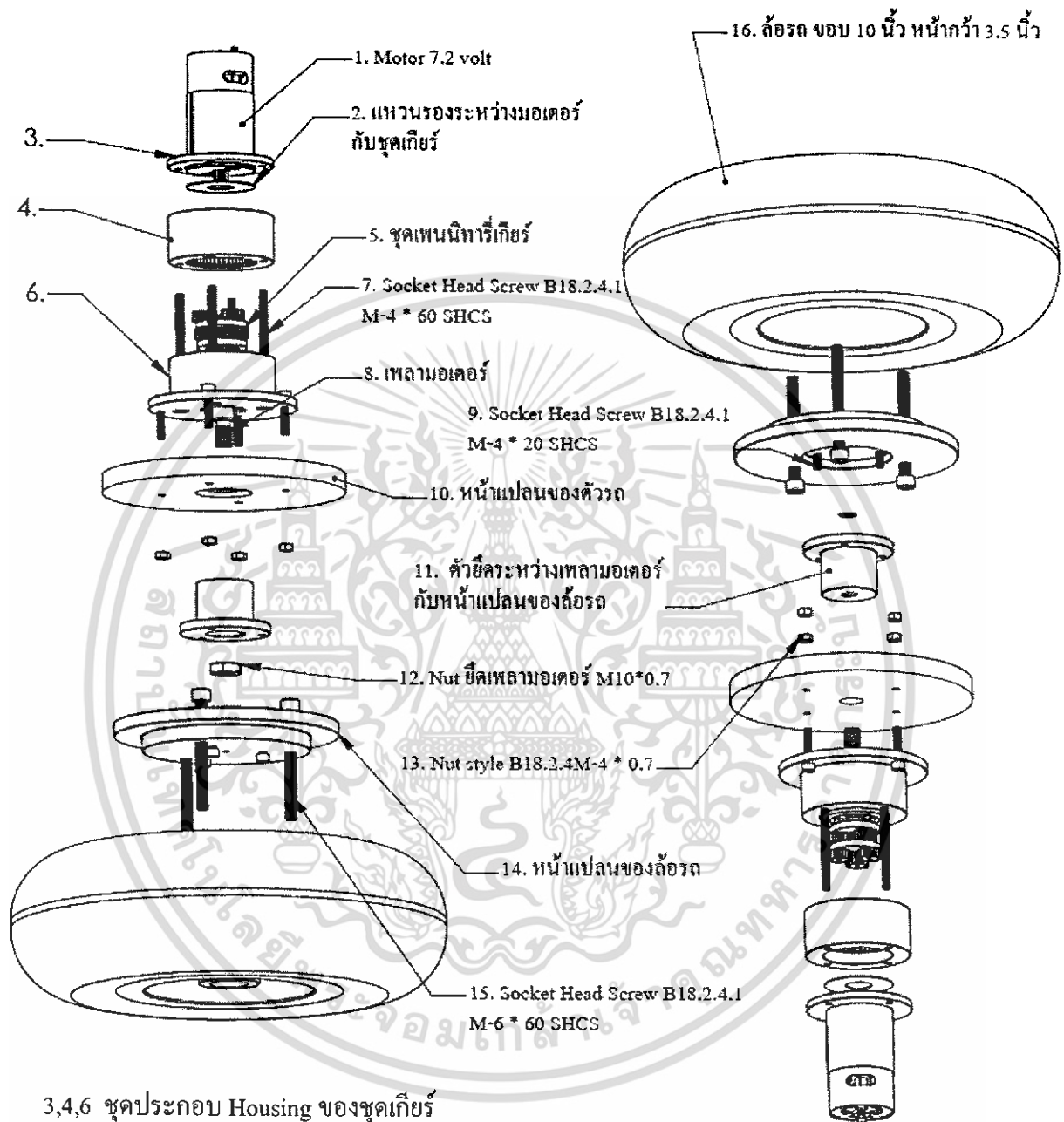
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ส่วนประกอบของก้านเพนคูล์ม



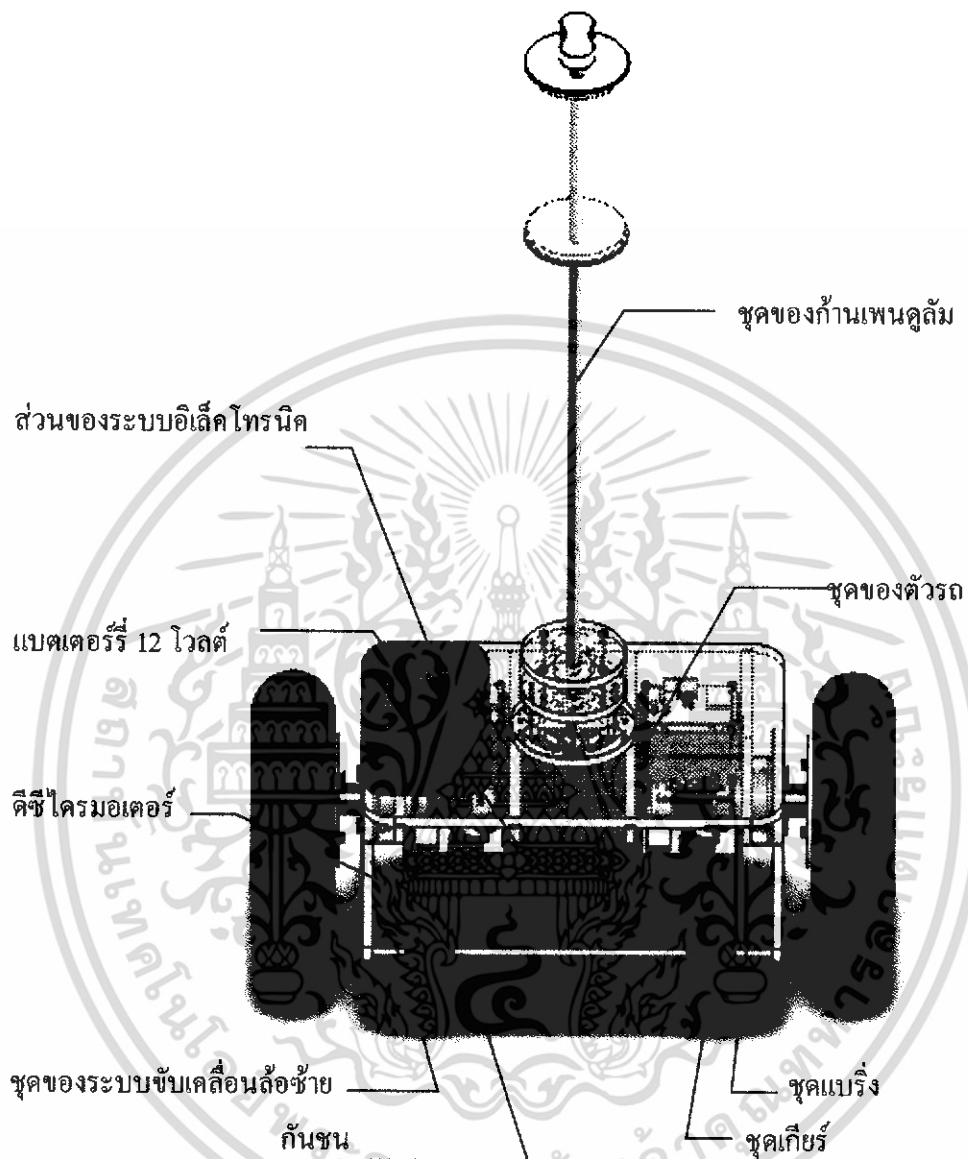
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ส่วนประกอบทั้งหมดของชุดขับเคลื่อน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. รูปร่างลักษณะของหุ่นยนต์ขับเคลื่อนแบบสองล้อ



ภาพจากการออกแบบโดยใช้โปรแกรม Solid Work

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ข้อมูลของไอซีที่ใช้ในปริิญญานิพนธ์

1. dsPIC30F6010(ไมโครคอนโทรลเลอร์)
2. IR2110(gate drive)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



dsPIC30F6010

dsPIC30F6010 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 with flexible Addressing modes
- 84 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 144 Kbytes on-chip Flash program space (Instruction words)
- 8 Kbytes of on-chip data RAM
- 4 Kbytes of non-volatile data EEPROM
- Up to 30 MIPS operation:
 - DC to 40 MHz external clock input
 - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 44 interrupt sources
 - 5 external interrupt sources
 - 8 user selectable priority levels for each interrupt source
 - 4 processor trap sources
- 16 x 16-bit working register array

DSP Engine Features:

- Dual data fetch
- Accumulator write back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single cycle hardware fractional/Integer multiplier
- All DSP instructions single cycle
- \pm 16-bit single cycle shift

Peripheral Features:

- High current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
 - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI™ modules (supports 4 Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 2 CAN modules, 2.0B compliant

Motor Control PWM Module Features:

- 8 PWM output channels
 - Complementary or Independent Output modes
 - Edge and Center Aligned modes
- 4 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead Time control for Complementary mode
- Manual output control
- Trigger for 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F6010

Analog Features:

- 10-bit Analog-to-Digital Converter (A/D) with 4 S/H Inputs:
 - 500 Ksps conversion rate
 - 16 input channels
 - Conversion available during Sleep and Idle
- Programmable Low Voltage Detection (PLVD)
- Programmable Brown-out Detection and Reset generation

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*

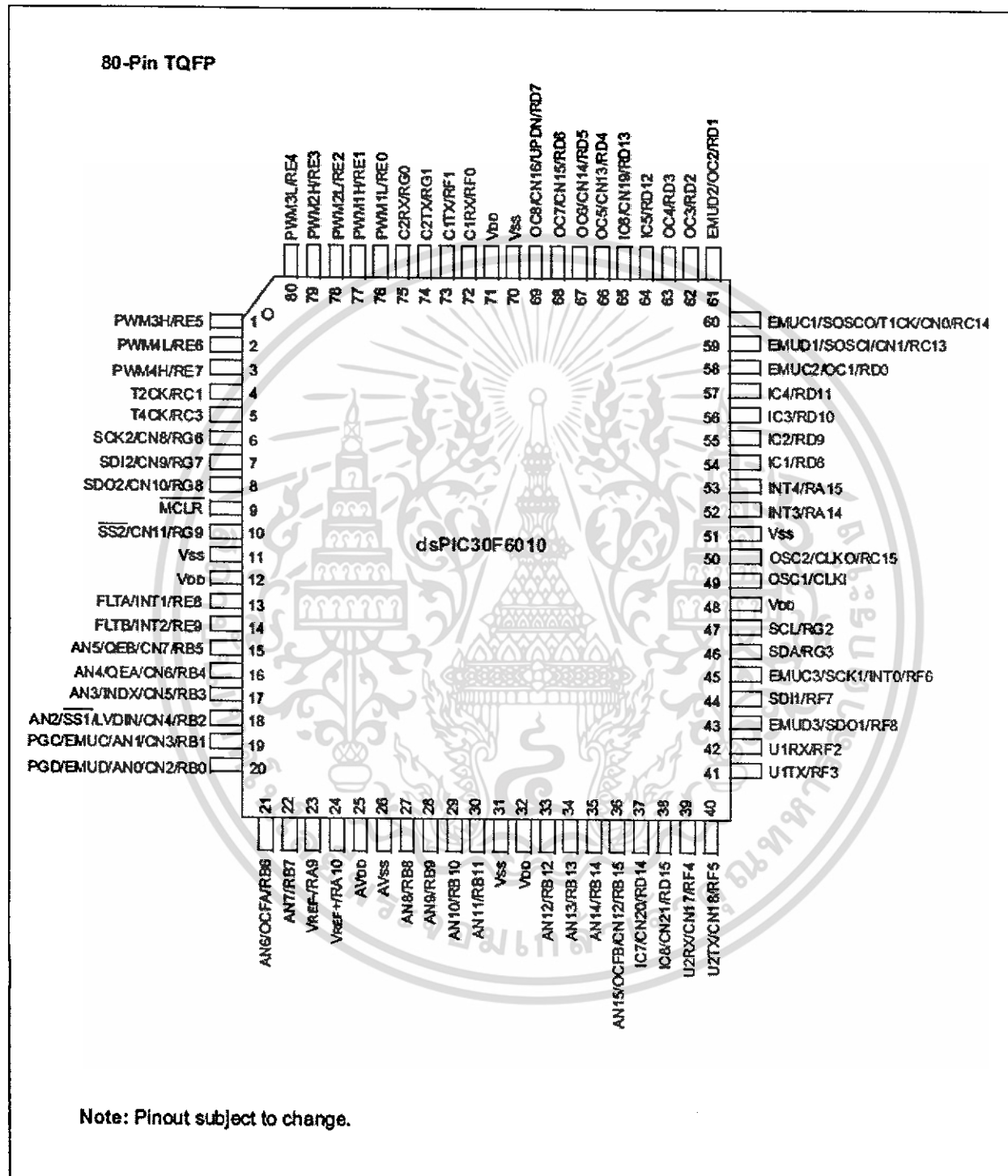
Device	Pins	Program Mem. Bytes/Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Moto Control PWM	A/D 10-bit 500 Ksps	Quad Enc	UART	SPI™	I ² C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	
dsPIC30F4012	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 dsPIC30F6010 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F6010

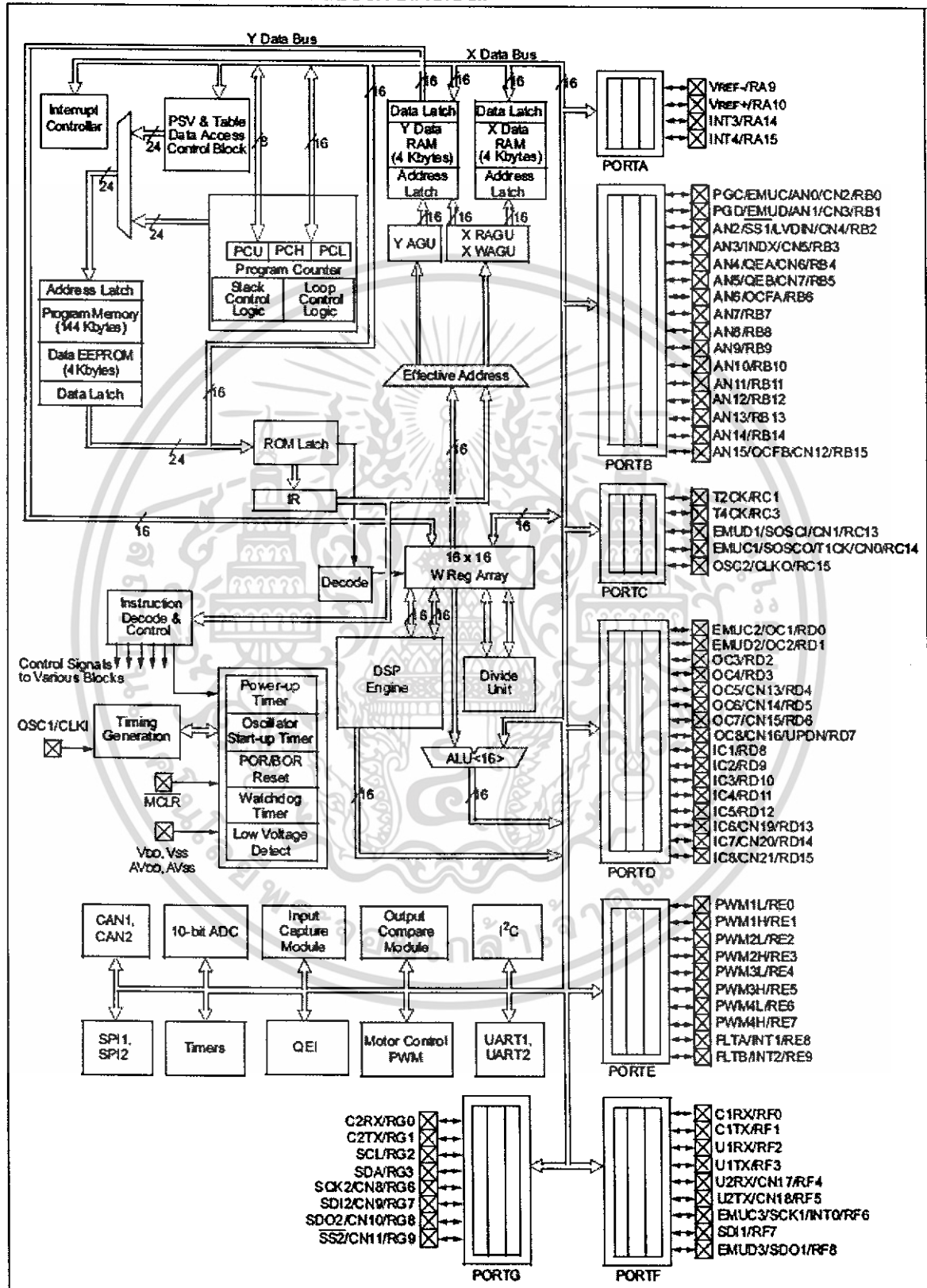
Pin Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F6010

FIGURE 1-1: dsPIC30F6010 BLOCK DIAGRAM



dsPIC30F6010

Table 1-1 provides a brief description of the device I/O pinout and the functions that are multiplexed to a port pin. Multiple functions may exist on one port pin. When multiplexing occurs, the peripheral module's functional requirements may force an override of the data direction of the port pin.

TABLE 1-1: dsPIC30F6010 I/O PIN DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN15	I	Analog	Analog input channels. AN0 and AN1 are also used for device programming data and clock inputs, respectively.
AVDD	P	P	Positive supply for analog module.
AVSS	P	P	Ground reference for analog module.
CLKI	I	ST/CMOS	External clock source input. Always associated with OSC1 pin function.
CLKO	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
CN0-CN23	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
COFS	I/O	ST	Data Converter interface frame synchronization pin.
CSCK	I/O	ST	Data Converter interface serial clock input/output pin.
CSDI	I	ST	Data Converter interface serial data input pin.
CSDO	O	—	Data Converter interface serial data output pin.
C1RX	I	ST	CAN1 bus receive pin.
C1TX	O	—	CAN1 bus transmit pin.
C2RX	I	ST	CAN2 bus receive pin.
C2TX	O	—	CAN2 bus transmit pin.
EMUD	I/O	ST	ICD Primary Communication Channel data input/output pin.
EMUC	I/O	ST	ICD Primary Communication Channel clock input/output pin.
EMUD1	I/O	ST	ICD Secondary Communication Channel data input/output pin.
EMUC1	I/O	ST	ICD Secondary Communication Channel clock input/output pin.
EMUD2	I/O	ST	ICD Tertiary Communication Channel data input/output pin.
EMUC2	I/O	ST	ICD Tertiary Communication Channel clock input/output pin.
EMUD3	I/O	ST	ICD Quaternary Communication Channel data input/output pin.
EMUC3	I/O	ST	ICD Quaternary Communication Channel clock input/output pin.
IC1-IC8	I	ST	Capture inputs 1 through 8.
INDX	I	ST	Quadrature Encoder Index Pulse input.
QEA	I	ST	Quadrature Encoder Phase A input in QE1 mode.
QEB	I	ST	Auxiliary Timer External Clock/Gate input in Timer mode. Quadrature Encoder Phase A input in QE1 mode.
UPDN	O	CMOS	Auxiliary Timer External Clock/Gate input in Timer mode. Position Up/Down Counter Direction State.
INT0	I	ST	External interrupt 0.
INT1	I	ST	External interrupt 1.
INT2	I	ST	External interrupt 2.
INT3	I	ST	External interrupt 3.
INT4	I	ST	External interrupt 4.
LVDIN	I	Analog	Low Voltage Detect Reference Voltage input pin.

Legend: CMOS = CMOS compatible input or output Analog = Analog input
 ST = Schmitt Trigger input with CMOS levels O = Output
 I = Input P = Power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F6010

TABLE 1-1: dsPIC30F6010 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
FLTA	I	ST	PWM Fault A input.
FLTB	I	ST	PWM Fault B input.
PWM1L	O	—	PWM 1 Low output.
PWM1H	O	—	PWM 1 High output.
PWM2L	O	—	PWM 2 Low output.
PWM2H	O	—	PWM 2 High output.
PWM3L	O	—	PWM 3 Low output.
PWM3H	O	—	PWM 3 High output.
PWM4L	O	—	PWM 4 Low output.
PWM4H	O	—	PWM 4 High output.
MCLR	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low Reset to the device.
OCFA	I	ST	Compare Fault A input (for Compare channels 1, 2, 3 and 4).
OCFB	I	ST	Compare Fault B input (for Compare channels 5, 6, 7 and 8).
OC1-OC8	O	—	Compare outputs 1 through 8.
OSC1	I	ST/CMOS	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
OSC2	I/O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLK0 in RC and EC modes.
PGD	I/O	ST	In-Circuit Serial Programming data input/output pin.
PGC	I	ST	In-Circuit Serial Programming clock input pin.
RA9-RA10	I/O	ST	PORTA is a bi-directional I/O port.
RA14-RA15	I/O	ST	
RB0-RB15	I/O	ST	PORTB is a bi-directional I/O port.
RC1	I/O	ST	PORTC is a bi-directional I/O port.
RC3	I/O	ST	
RC13-RC15	I/O	ST	
RD0-RD15	I/O	ST	PORTD is a bi-directional I/O port.
RE0-RE9	I/O	ST	PORTE is a bi-directional I/O port.
RF0-RF8	I/O	ST	PORTF is a bi-directional I/O port.
RG0-RG3	I/O	ST	PORTG is a bi-directional I/O port.
RG6-RG9	I/O	ST	
SCK1	I/O	ST	Synchronous serial clock input/output for SPI™ #1.
SDI1	I	ST	SPI #1 Data In.
SDO1	O	—	SPI #1 Data Out.
SS1	I	ST	SPI #1 Slave Synchronization.
SCK2	I/O	ST	Synchronous serial clock input/output for SPI #2.
SDI2	I	ST	SPI #2 Data In.
SDO2	O	—	SPI #2 Data Out.
SS2	I	ST	SPI #2 Slave Synchronization.
SCL	I/O	ST	Synchronous serial clock input/output for I ² C.
SDA	I/O	ST	Synchronous serial data input/output for I ² C.
SOSCO	O	—	32 kHz low power oscillator crystal output.
SOSCI	I	ST/CMOS	32 kHz low power oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.

Legend: CMOS = CMOS compatible input or output Analog = Analog input
 ST = Schmitt Trigger input with CMOS levels O = Output
 I = Input P = Power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dsPIC30F6010

TABLE 1-1: dsPIC30F6010 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
T1CK	I	ST	Timer1 external clock input.
T2CK	I	ST	Timer2 external clock input.
T3CK	I	ST	Timer3 external clock input.
T4CK	I	ST	Timer4 external clock input.
T5CK	I	ST	Timer5 external clock input.
U1RX	I	ST	UART1 Receive.
U1TX	O	—	UART1 Transmit.
U1ARX	I	ST	UART1 Alternate Receive.
U1ATX	O	—	UART1 Alternate Transmit.
U2RX	I	ST	UART2 Receive.
U2TX	O	—	UART2 Transmit.
VDD	P	—	Positive supply for logic and I/O pins.
VSS	P	—	Ground reference for logic and I/O pins.
VREF+	I	Analog	Analog Voltage Reference (High) input.
VREF-	I	Analog	Analog Voltage Reference (Low) input.

Legend: CMOS = CMOS compatible input or output Analog = Analog Input
 ST = Schmitt Trigger input with CMOS levels O = Output
 I = Input P = Power

ข้อมูลของไมโครคอนโทรลเลอร์รุ่นนี้ที่ได้นำเสนอเป็นเพียงทางด้านคุณสมบัติที่สำคัญเท่านั้น จะไม่กล่าวคุณลักษณะต่างๆ เช่น สถาปัตยกรรม โครงสร้างทางโปรแกรม หน่วยประมวลผลสัญญาณดิจิทัล เป็นต้น ซึ่งข้อมูลเหล่านี้ ผู้สนใจสามารถค้นคว้าหาข้อมูลได้จากเว็บไซต์ของบริษัท ไมโครชิฟ

HIGH AND LOW SIDE DRIVER

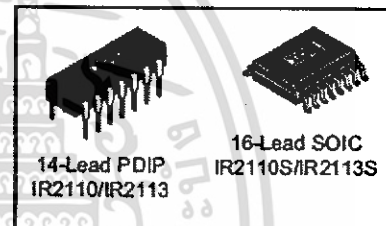
Features

- Floating channel designed for bootstrap operation
Fully operational to +500V or +600V
Tolerant to negative transient voltage
dV/dt immune
- Gate drive supply range from 10 to 20V
- Undervoltage lockout for both channels
- 3.3V logic compatible
Separate logic supply range from 3.3V to 20V
Logic and power ground $\pm 5V$ offset
- CMOS Schmitt-triggered inputs with pull-down
- Cycle by cycle edge-triggered shutdown logic
- Matched propagation delay for both channels
- Outputs in phase with inputs

Product Summary

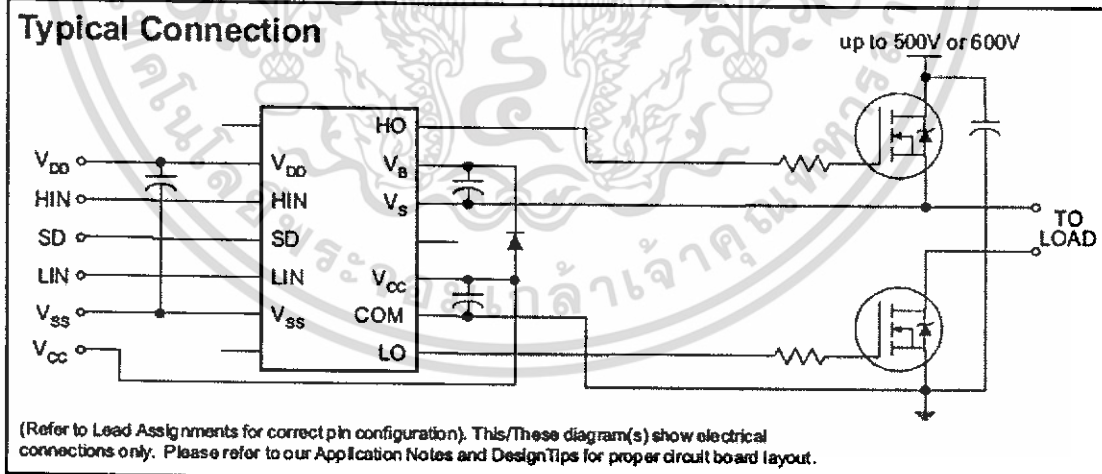
V_{OFFSET} (IR2110)	500V max.
(IR2113)	600V max.
$I_{\text{O}+/-}$	2A / 2A
V_{OUT}	10 - 20V
$t_{\text{on/off}}$ (typ.)	120 & 94 ns
Delay Matching (IR2110)	10 ns max.
(IR2113)	20ns max.

Packages



Description

The IR2110/IR2113 are high voltage, high speed power MOSFET and IGBT drivers with independent high and low side referenced output channels. Proprietary HVIC and latch immune CMOS technologies enable ruggedized monolithic construction. Logic inputs are compatible with standard CMOS or LSTTL output, down to 3.3V logic. The output drivers feature a high pulse current buffer stage designed for minimum driver cross-conduction. Propagation delays are matched to simplify use in high frequency applications. The floating channel can be used to drive an N-channel power MOSFET or IGBT in the high side configuration which operates up to 500 or 600 volts.



IR2110(-1-2)(S)PbF/IR2113(-1-2)(S)PbF

International
IOR Rectifier

Absolute Maximum Ratings

Absolute maximum ratings indicate sustained limits beyond which damage to the device may occur. All voltage parameters are absolute voltages referenced to COM. The thermal resistance and power dissipation ratings are measured under board mounted and still air conditions. Additional information is shown in Figures 28 through 35.

Symbol	Definition	Min.	Max.	Units	
V _B	High side floating supply voltage (IR2110)	-0.3	525	V	
	(IR2113)	-0.3	625		
V _S	High side floating supply offset voltage	V _B - 25	V _B + 0.3		
V _{HO}	High side floating output voltage	V _S - 0.3	V _B + 0.3		
V _{CC}	Low side fixed supply voltage	-0.3	25		
V _{LO}	Low side output voltage	-0.3	V _{CC} + 0.3		
V _{DD}	Logic supply voltage	-0.3	V _{SS} + 25		
V _{SS}	Logic supply offset voltage	V _{CC} - 25	V _{CC} + 0.3		
V _{IN}	Logic input voltage (HIN, LIN & SD)	V _{SS} - 0.3	V _{DD} + 0.3		
dV _S /dt	Allowable offset supply voltage transient (figure 2)	—	50		V/ns
P _D	Package power dissipation @ T _A ≤ +25°C	(14 lead DIP)	—	1.6	W
		(16 lead SOIC)	—	1.25	
R _{THJA}	Thermal resistance, junction to ambient	(14 lead DIP)	—	75	°C/W
		(16 lead SOIC)	—	100	
T _J	Junction temperature	—	150	°C	
T _S	Storage temperature	-55	150		
T _L	Lead temperature (soldering, 10 seconds)	—	300		

Recommended Operating Conditions

The input/output logic timing diagram is shown in figure 1. For proper operation the device should be used within the recommended conditions. The V_S and V_{SS} offset ratings are tested with all supplies biased at 15V differential. Typical ratings at other bias conditions are shown in figures 36 and 37.

Symbol	Definition	Min.	Max.	Units
V _B	High side floating supply absolute voltage	V _S + 10	V _S + 20	V
V _S	High side floating supply offset voltage (IR2110)	Note 1	500	
	(IR2113)	Note 1	600	
V _{HO}	High side floating output voltage	V _S	V _B	
V _{CC}	Low side fixed supply voltage	10	20	
V _{LO}	Low side output voltage	0	V _{CC}	
V _{DD}	Logic supply voltage	V _{SS} + 3	V _{SS} + 20	
V _{SS}	Logic supply offset voltage	-5 (Note 2)	5	
V _{IN}	Logic input voltage (HIN, LIN & SD)	V _{SS}	V _{DD}	
T _A	Ambient temperature	-40	125	

Note 1: Logic operational for V_S of -4 to +500V. Logic state held for V_S of -4V to -V_{SS}. (Please refer to the Design Tip DT97-3 for more details).

Note 2: When V_{DD} < 5V, the minimum V_{SS} offset is limited to -V_{DD}.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dynamic Electrical Characteristics

V_{BIAS} (V_{CC} , V_{BS} , V_{DD}) = 15V, C_L = 1000 pF, T_A = 25°C and V_{SS} = COM unless otherwise specified. The dynamic electrical characteristics are measured using the test circuit shown in Figure 3.

Symbol	Definition	Figure	Min.	Typ.	Max.	Units	Test Conditions
t_{on}	Turn-on propagation delay	7	—	120	150	ns	$V_S = 0V$
t_{off}	Turn-off propagation delay	8	—	94	125		$V_S = 500V/600V$
t_{sd}	Shutdown propagation delay	9	—	110	140		$V_S = 500V/600V$
t_r	Turn-on rise time	10	—	25	35		
t_f	Turn-off fall time	11	—	17	25		
MT	Delay matching, HS & LS turn-on/off	(IR2110) (IR2113)	— —	— —	10 20		

Static Electrical Characteristics

V_{BIAS} (V_{CC} , V_{BS} , V_{DD}) = 15V, T_A = 25°C and V_{SS} = COM unless otherwise specified. The V_{IH} , V_{TH} and I_{IN} parameters are referenced to V_{SS} and are applicable to all three logic input leads: HIN, LIN and SD. The V_O and I_O parameters are referenced to COM and are applicable to the respective output leads: HO or LO.

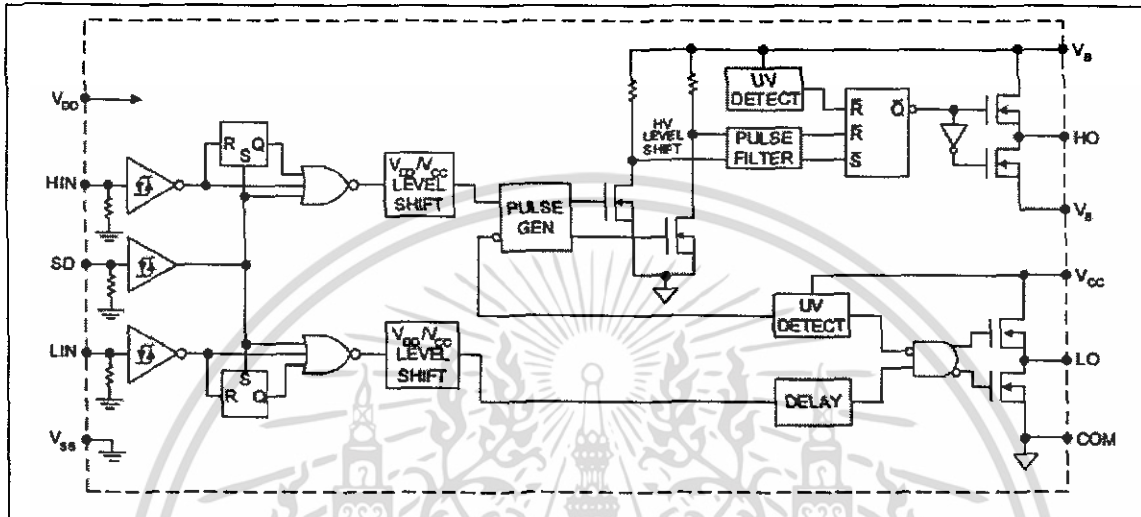
Symbol	Definition	Figure	Min.	Typ.	Max.	Units	Test Conditions
V_{IH}	Logic "1" input voltage	12	9.5	—	—	V	
V_{IL}	Logic "0" input voltage	13	—	—	6.0		
V_{OH}	High level output voltage, $V_{BIAS} - V_O$	14	—	—	1.2		$I_O = 0A$
V_{OL}	Low level output voltage, V_O	15	—	—	0.1		$I_O = 0A$
I_{LK}	Offset supply leakage current	16	—	—	50	μA	$V_B = V_S = 500V/600V$
I_{OBS}	Quiescent V_{BS} supply current	17	—	125	230		$V_{IN} = 0V$ or V_{DD}
I_{OCC}	Quiescent V_{CC} supply current	18	—	180	340		$V_{IN} = 0V$ or V_{DD}
I_{ODD}	Quiescent V_{DD} supply current	19	—	15	30		$V_{IN} = 0V$ or V_{DD}
I_{IN+}	Logic "1" input bias current	20	—	20	40		$V_{IN} = V_{DD}$
I_{IN-}	Logic "0" input bias current	21	—	—	1.0		$V_{IN} = 0V$
V_{BSUV+}	V_{BS} supply undervoltage positive going threshold	22	7.5	8.6	9.7	V	
V_{BSUV-}	V_{BS} supply undervoltage negative going threshold	23	7.0	8.2	9.4		
V_{CCUV+}	V_{CC} supply undervoltage positive going threshold	24	7.4	8.5	9.6		
V_{CCUV-}	V_{CC} supply undervoltage negative going threshold	25	7.0	8.2	9.4		
I_{O+}	Output high short circuit pulsed current	26	2.0	2.5	—	A	$V_O = 0V$, $V_{IN} = V_{DD}$ $PW \leq 10 \mu s$
I_{O-}	Output low short circuit pulsed current	27	2.0	2.5	—		$V_O = 15V$, $V_{IN} = 0V$ $PW \leq 10 \mu s$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IR2110(-1-2)(S)PbF/IR2113(-1-2)(S)PbF

International
IOR Rectifier

Functional Block Diagram



Lead Definitions

Symbol	Description
VDD	Logic supply
HIN	Logic input for high side gate driver output (HO), in phase
SD	Logic input for shutdown
LIN	Logic input for low side gate driver output (LO), in phase
VSS	Logic ground
VB	High side floating supply
HO	High side gate drive output
VS	High side floating supply return
VCC	Low side supply
LO	Low side gate drive output
COM	Low side return

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IR2110(-1-2)(S)PbF/IR2113(-1-2)(S)PbF

International
IR Rectifier

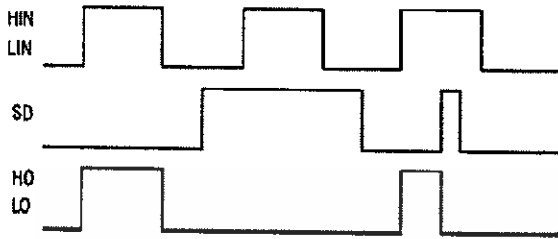


Figure 1. Input/Output Timing Diagram

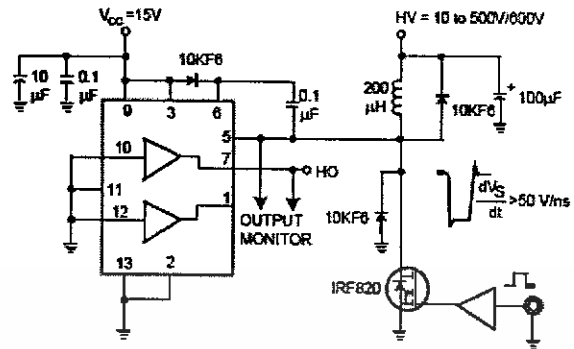


Figure 2. Floating Supply Voltage Transient Test Circuit

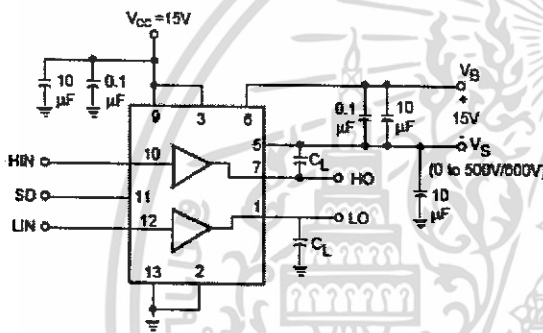


Figure 3. Switching Time Test Circuit

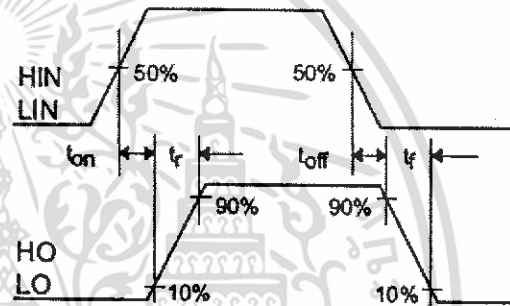


Figure 4. Switching Time Waveform Definition

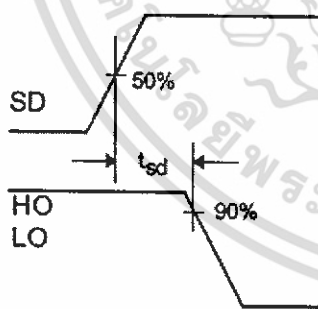


Figure 5. Shutdown Waveform Definitions

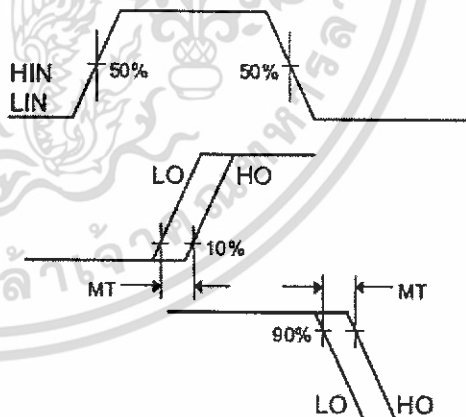


Figure 6. Delay Matching Waveform Definitions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้